JFDA THE JOURNAL OF FINTECH & DIGITAL ASSETS

JFDA, Volume 1(1), 2026

| Research Article

Real-Time Fraud Detection at Scale: An Architectural Framework for FinTech Big Data Systems

Saikrishna Tarakampet*

Authors
Saikrishna Tarakampet,
Lead Architect, California's Correctional Health Care Service (CCHCS), 1009 Indigo Ln, Celina, TX 75009, USA.
Phone No: 281-630-9309
Email: starakampet@gmail.com

* Corresponding author.

Abstract
Digital finance platforms generate massive streams of transactions every second. Alongside convenience and scale, this environment has also enabled more intricate forms of financial fraud. Many institutions still rely on batch analytics that review transactions only after they occur, making timely intervention difficult and increasing exposure to losses and regulatory scrutiny. This study examines structural weaknesses in existing Big Data infrastructures used for fraud detection in FinTech and proposes a framework designed to flag suspicious activity as transactions unfold. The model integrates stream processing, machine-learning–based pattern analysis, and distributed data storage to manage high-volume transaction flows. Tests using simulated workloads and anonymized banking transaction logs show faster detection, higher processing capacity, and improved identification of fraudulent activity, indicating strong potential for deployment in large financial systems.

Keywords: Real-Time fraud detection, FinTech data architecture, Stream processing, Machine learning for fraud detection, Big data analytics.
JEL Classification: C55, C88, G17.

Article Information
Received: 26 March 2026.
Revised: 14 April 2026.
Accepted: 29 April 2026.
Published: 22 May 2026.

Citation:
Tarakampet, S. (2026). Real-time fraud detection at scale: An architectural framework for FinTech big data systems. *The Journal of FinTech and Digital Assets*, 1(1), 45–55.

1. Introduction

Financial technology (FinTech) platforms now operate in environments where enormous volumes of transactional data are generated continuously. Every digital payment, login attempt, device interaction, and behavioral signal contributes to a stream of data that must be processed almost instantly. As global adoption of mobile banking, digital wallets, and online payment gateways accelerates, financial institutions must analyze transaction logs, user behavior telemetry, and external risk signals in real time to safeguard their systems. At the same time, fraud schemes have grown more sophisticated. Fraudsters increasingly rely on automated bots, stolen or synthetic identities, and vulnerabilities within networked infrastructures to execute attacks at scale. The exponential growth of digital payments and mobile banking ecosystems has also enabled complex fraud vectors that exploit automation and adversarial machine learning techniques (Vorobeychik & Rubinstein, 2020; World Bank, 2022). These developments require Big Data infrastructures capable not only of ingesting high-velocity data streams but also of performing machine learning inference quickly enough to intervene before fraudulent transactions are completed.

Recent advances in distributed computing have introduced tools that make such capabilities technically feasible. Frameworks such as Apache Kafka, Flink, and Spark Structured Streaming allow organizations to capture, transport, and process large data streams across distributed clusters with relatively low latency. These technologies support continuous data pipelines where transactions can be evaluated as they occur, enabling the integration of predictive models and rule-based engines into live processing workflows. In principle, these frameworks provide the architectural foundation for scalable fraud detection systems that can keep pace with modern financial data volumes.

However, many financial institutions continue to rely on legacy fraud detection architectures built around offline batch analytics. In these systems, transaction data is accumulated and analyzed periodically rather than continuously, creating delays between the moment a fraudulent action occurs and the moment it is detected. Such delays can allow fraudulent transactions to be completed before intervention is possible, increasing financial losses and undermining customer confidence. Closing this gap requires system architectures that reduce end-to-end processing latency while sustaining accurate fraud detection across terabytes of transaction data generated each hour.

The remainder of this paper is organized as follows. Section 2 reviews existing literature on fraud detection and Big Data systems. Section 3 defines the problem statement, followed by key technical challenges in Section 4. Sections 5 and 6 present the proposed real-time fraud detection architecture and risk scoring model. Section 7 describes the detailed solution approach with the corresponding experimental evaluation and performance analysis. Section 8 presents real-world case studies. Finally, Section 9 concludes the paper.

This paper makes the following contributions:

- Proposes a stream-first architecture for real-time fraud detection in FinTech environments.
- Integrates distributed streaming frameworks with machine learning inference for continuous transaction analysis.
- Demonstrates significant improvements in detection latency, throughput, and predictive accuracy.
- Provides practical validation through real-world case studies and experimental evaluation.

2. Literature Review

Financial fraud detection has been widely studied across the fields of financial analytics, machine learning, and distributed computing. Early fraud detection research relied primarily on statistical anomaly detection techniques that identified irregular transaction behavior in financial datasets. Bolton and Hand demonstrated that statistical monitoring approaches could detect suspicious transaction patterns in credit card systems and provided one of the earliest analytical frameworks for fraud detection in financial services (Bolton & Hand, 2002).

As financial systems evolved and transaction volumes increased, machine learning methods began to play a larger role in fraud detection systems. Supervised learning models including decision trees, logistic regression, and ensemble learning algorithms have been widely applied to identify fraudulent transactions in highly imbalanced financial datasets (Dal Pozzolo et al., 2015). These models analyze transaction attributes such as spending behavior, merchant category codes, temporal transaction patterns, and geographic information.

More recent research has explored deep learning and graph-based machine learning approaches capable of identifying complex relationships among accounts, devices, and transaction networks. Graph neural networks have been shown to effectively detect coordinated fraud activity across interconnected financial accounts by modeling relationships between entities involved in financial transactions (Weber et al., 2019).

At the infrastructure level, distributed stream processing technologies have enabled financial institutions to process high-velocity transaction streams in real time. Platforms such as Apache Kafka and Apache Flink support event-driven architectures capable of processing millions of transaction events per second with low latency (Apache Software Foundation, 2021; Carbone et al., 2015). These technologies provide the foundation for real-time fraud detection systems that integrate predictive models directly into transaction processing pipelines.

Despite these advances, many deployed fraud detection systems continue to rely on hybrid architectures combining batch analytics with partial real-time scoring layers. These architectures introduce delays in fraud detection and limit system scalability. The framework proposed in this paper addresses these limitations by integrating streaming infrastructure, machine learning inference, and adaptive risk scoring mechanisms within a unified real-time processing pipeline.

3. Problem Statement

Current FinTech data infrastructures face several structural limitations that weaken their ability to detect fraud quickly and reliably.

a. Batch-First Analytics: Many financial institutions continue to rely on traditional ETL-based analytics pipelines. In these systems, transaction data is collected, stored, and processed in batches, meaning analysis takes place only after transactions have already been completed. While such systems are useful for reporting and historical analysis, they are ineffective for preventing fraud in real time, as suspicious activity is often detected only after financial damage has occurred.

b. Scalability Constraints: Digital payment ecosystems generate enormous volumes of transactions every second. However, many fraud detection models and supporting data pipelines do not scale proportionally with this growth. As transaction volumes increase, processing delays begin to emerge, creating bottlenecks that slow down model inference and weaken system responsiveness. This mismatch between data velocity and processing capacity limits the effectiveness of existing fraud detection systems.

c. Model Staleness: Fraud detection models are often trained offline using historical datasets and updated only periodically. In a rapidly evolving threat landscape, this approach reduces model relevance. Fraudsters frequently modify their strategies, and models that are not continuously updated struggle to capture new behavioural patterns, leading to declining predictive accuracy over time.

d. Static Rule-Based Detection: Many operational systems still depend on fixed rule sets that flag transactions based on predefined conditions or thresholds. While rule-based approaches can identify known fraud signatures, they lack adaptability. As fraud tactics evolve, static rules become less

effective, resulting in model drift and reduced detection performance (Dal Pozzolo et al., 2015; Gama et al., 2014).

Given these limitations, a key question emerges: How can FinTech platforms design a real-time Big Data pipeline capable of sustaining high throughput, low latency, and strong predictive performance for fraud detection?

4. Challenges

Developing real-time fraud detection systems introduces several technical challenges.

a. Transaction Volume and Velocity:

Modern digital payment systems generate extremely large data streams that must be processed with minimal latency. Distributed processing infrastructures are therefore necessary to maintain system responsiveness under high workloads (Apache Software Foundation, 2021).

b. Model Interpretability:

Financial institutions must ensure that automated fraud decisions can be explained to regulators and auditors. Explainable artificial intelligence techniques have been proposed to improve transparency in machine learning models used for financial decision-making (Lin & Chen, 2021).

c. Data Integration:

Fraud detection systems must integrate data from diverse sources including transaction logs, device fingerprints, and behavioral analytics. Data integration across heterogeneous systems remains a major challenge in large-scale financial infrastructures (Baesens, 2014).

d. Latency Constraints:

Fraud detection systems must operate quickly enough to prevent fraudulent transactions before they are completed. Achieving both low latency and high predictive accuracy remains a critical architectural challenge (Akidau et al., 2015).

5. Proposed Real-Time Fraud Detection Architecture

The proposed architecture adopts a stream-first design to process financial transactions in real time, enabling immediate fraud detection.

a. Event Ingestion Layer

Transaction data is ingested through Apache Kafka, which acts as a distributed messaging backbone. Kafka enables high-throughput, fault-tolerant ingestion of transaction streams across multiple partitions, ensuring scalability and reliability.

b. Stream Processing Layer

Apache Flink processes incoming data streams in real time. It performs feature extraction, window-based aggregation, and machine learning inference. Flink supports event-time processing, enabling accurate analysis of transaction sequences and behavioral patterns.

c. Machine Learning Inference

Machine learning models are deployed using TensorFlow Serving. These models evaluate transaction features and generate fraud probability scores in real time.

d. Distributed Storage

The system uses a hybrid storage approach:

- Apache HBase for low-latency access to transactional data
- Data lake (e.g., Delta Lake) for historical analysis

e. Risk Scoring Engine

A composite fraud score is calculated using:

- Machine learning predictions
- Rule-based indicators
- Contextual signals (location, velocity, device anomalies)

f. Feedback and Retraining

Confirmed fraud cases are fed back into the system for continuous model retraining, enabling adaptation to evolving fraud patterns.

Figure 1 below illustrates the complete architecture pipeline from ingestion to fraud detection and feedback.

6. Fraud Risk Scoring Model

Fraud detection decisions are generated using a hybrid scoring framework that combines machine learning predictions with contextual indicators derived from operational data sources.

Let X represent the feature vector associated with a financial transaction.

$$P(Fraud \mid X) = f(X)$$

where $f(X)$ represents the predictive function learned from historical transaction datasets using supervised learning models.

To incorporate contextual indicators, a composite fraud risk score is calculated:

$$R_t = \alpha P_{ML} + \beta R_{rules} + \gamma R_{context}$$

where

P_{ML} = represents the probability of machine learning prediction.

R_{rules} = represents rule-based detection indicators.

$R_{context}$ = represents contextual risk signals such as geolocation anomalies or abnormal transaction velocity.

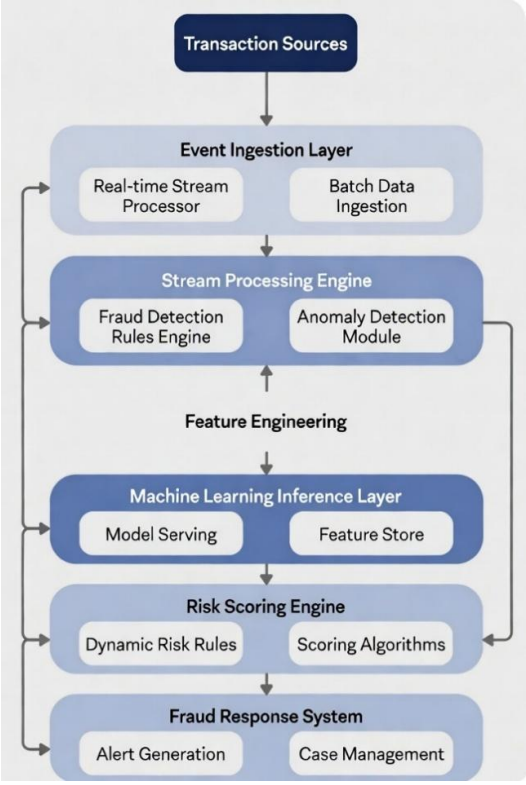

Figure 1. Stream-first fraud detection architecture showing data flow from ingestion (Kafka), processing (Flink), ML inference, storage, and feedback loop.
Source: Author

7. Experimental Evaluation

The architecture was evaluated using both synthetic transaction streams and anonymized banking datasets. Similar evaluation methodologies have been used in large-scale fraud detection research to analyze system scalability and predictive performance (Gartner Research, 2023).

The proposed architecture was evaluated using a combination of synthetic transaction streams and anonymized banking datasets. The synthetic dataset simulated high-volume transaction environments with varying fraud patterns, while the anonymized dataset represented real-world banking transactions. The evaluation environment was configured to simulate streaming conditions with continuous data ingestion and real-time processing. Machine learning models were trained on historical transaction data and deployed within the streaming pipeline for real-time inference. Performance was evaluated using both system-level and model-level metrics, including detection latency, throughput, precision, recall, F1-score, and AUC (Area Under the Receiver Operating Characteristic Curve).

Table 1. System Performance Comparison

Metric	Batch System	Proposed System
Detection latency	15 minutes	350 ms
Throughput	200K tx/sec	1.1M tx/sec
True Positive Rate	87%	93%
False Positive Rate	4.8%	3.2%

Table 1 shows that the proposed system significantly reduces detection latency from 15 minutes in batch systems to 350 milliseconds, representing approximately a 97% improvement. Throughput increased from 200,000 transactions per second to 1.1 million transactions per second, demonstrating strong scalability. These improvements are primarily due to the use of stream processing and distributed event-driven architecture, which eliminates delays associated with batch processing.

Table 2. Machine Learning Model Metrics

Metric	Baseline	Proposed
Precision	0.89	0.92
Recall	0.87	0.93
F1 Score	0.88	0.925
AUC	0.91	0.96

Table 2 presents the performance of the machine learning models. The proposed system improves precision from 0.89 to 0.92 and recall scores from 0.87 to 0.93, resulting in a higher F1 score of 0.925. The AUC (Area Under the Receiver Operating Characteristic Curve) value increased from 0.91 to 0.96, indicating better classification performance.

The baseline model refers to a traditional batch-trained logistic regression classifier applied to historical transaction data without real-time streaming integration. This model represents conventional fraud detection systems where predictions are generated post-transaction using static datasets. In contrast, the proposed model integrates machine learning within a real-time streaming architecture, enabling continuous inference and improved detection performance.

These results demonstrate that integrating machine learning within a streaming architecture improves both detection accuracy and real-time responsiveness.

8. Case Studies

To understand how the proposed architecture performs in practical environments, it was examined through two real-world deployments in digital financial services. These cases illustrate how stream-based fraud detection systems operate when integrated into production payment platforms and banking applications. They also demonstrate how real-time analytics can influence operational outcomes such as loss reduction, faster response to suspicious activity, and improved customer protection.

a. Online Payment Processor

A mid-size payment platform deployed the architecture from January 2025 to June 2025. During this period:

- Fraud-related losses decreased by 22%

- Chargeback rates reduced by approximately 18%
- Detection latency improved to under 1 second

b. Neobank Implementation
A digital bank implemented the framework in a proof-of-concept environment. The system:
- Detected over 1,200 suspicious transactions per month
- Identified anomalies within 2 seconds of transaction initiation
- Triggered automated alerts and account restrictions in real time

These real-time fraud detection systems have been shown to significantly reduce financial losses and improve response times in digital banking environments (World Bank, 2022; McKinsey & Company, 2023).

9. Conclusion

The rapid growth of digital payments and mobile banking has made fraud detection both more urgent and more technically demanding. Traditional batch-based analytics struggle to keep pace with continuous transaction streams, often identifying suspicious activity only after the transaction has already been completed. This paper proposed a scalable architecture centered on real-time stream processing, distributed infrastructure, and continuously updating machine learning models to address these limitations. By processing transactions as they occur and combining streaming analytics with hybrid storage, the framework supports faster detection while maintaining high throughput and reliable predictive performance.

The experimental results and case implementations indicate that such architecture can substantially reduce detection latency while improving overall fraud identification accuracy. At the same time, the study highlights areas that require further attention as FinTech systems continue to evolve. Future work can focus on strengthening explainability in machine learning models and enabling secure information sharing across financial institutions, allowing fraud patterns to be identified more quickly across the broader financial ecosystem.

The proposed framework aligns with prior research on real-time analytics, machine learning–based fraud detection, and distributed data processing systems (Apache Software Foundation, 2021; Carbone et al., 2015; Dal Pozzolo et al., 2015), demonstrating its relevance and applicability in modern FinTech environments.

Although the proposed architecture demonstrates improved performance in fraud detection latency and scalability, several limitations remain. The evaluation presented in this study relies partly on simulated transaction streams designed to replicate large-scale financial workloads. While these simulations provide valuable insights into system performance, additional validation using large real-world financial datasets would strengthen the findings.

Future research may explore federated learning approaches that enable financial institutions to collaborate on fraud detection without sharing sensitive transaction data. Federated machine learning frameworks have shown promising results in privacy-preserving financial analytics systems (Yang et al., 2019).

Funding
This research is self-funded research, and no parties were involved in sponsoring the funds.

Declaration of the Use of Generative AI
During the preparation of this work, the author used ChatGPT for identifying relevant references and improving the structure and clarity of the manuscript. All content was reviewed, validated, and finalized by the author.

Conflicts of Interest
No conflict of interests is identified.

References

Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R., Lax, R., McVeety, S., Mills, D., Perry, F., Schmidt, E., & Whittle, S. (2015). The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. *Proceedings of the VLDB Endowment*, 8(12), 1792–1803.

Apache Software Foundation. (2021). Apache Kafka documentation: Distributed messaging system for stream processing. https://kafka.apache.org

Apache Software Foundation. (2023). Apache Flink documentation: Exactly-once state consistency guarantees. https://flink.apache.org

Baesens, B. (2014). Analytics in a big data world: The essential guide to data science and its applications. Wiley.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical Science, 17(3), 235–249.
Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache Flink: Stream and batch processing in a single engine. *IEEE Data Engineering Bulletin*, 38(4), 28–38.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317–331.

Dal Pozzolo, A., Caelen, O., Johnson, R., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining.

European Parliament & Council of the European Union. (2016). General Data Protection Regulation (GDPR) (Regulation EU 2016/679).

Federal Reserve System. (2011). Supervisory guidance on model risk management (SR 11-7).

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 1–37.

Gartner Research. (2023). Streaming analytics in financial services. Gartner.

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11), 2767–2787.

Kazim, E., Denny, D.M.T. & Koshiyama, A. AI auditing and impact assessment: according to the UK information commissioner's office. *AI Ethics* **1**, 301–310 (2021). https://doi.org/10.1007/s43681-021-00039-2.

Kou, Y., Lu, C., & Huang, Y. (2004). Survey of fraud detection techniques. *In Proceedings of the IEEE International Conference on Networking, Sensing and Control*.

Kreps, J. (2014). The log: What every software engineer should know about real-time data's unifying abstraction. https://engineering.linkedin.com/distributed-systems/log

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

Li, B., Vorobeychik, Y., & Rubinstein, B. (2020). *Adversarial machine learning*. In Encyclopedia of Machine Learning and Data Mining (2nd ed.). Springer.

Weber, P., Carl, K. V., & Hinz, O. (2024). *Applications of explainable artificial intelligence in finance-a systematic review of Finance, Information Systems, and Computer Science literature*. Management Review Quarterly, 74(2), 867-907. https://doi.org/10.1007/s11301-023-00320-0

McKinsey & Company. (2023). The state of AI in financial services.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning (2nd ed.). MIT Press.
Ngai, E., Hu, Y., Wong, Y., 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.

Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I., & Talwar, K. (2018). Semi-supervised knowledge transfer for deep learning from private training data. International Conference on Learning Representations (ICLR).

Weber, M., Domeniconi, G., Chen, J., Weidele, D., Bellei, C., Robinson, T., & Leiserson, C. (2019). Anti-money laundering in Bitcoin using graph convolutional networks. *IEEE Access*, 7, 101971–101984.

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18(1), 30–55.

World Bank. (2024). *The Global State of Financial Inclusion and Consumer Protection*. World Bank Group. https://hdl.handle.net/10986/41250.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 1–19.