European Journal of Electrical Engineering and Computer Science
Vol 9 | Issue 6 | December 2025
ISSN 2736-5751

RESEARCH ARTICLE

Comparative Study of Event-Driven Architectures in Financial Systems for Real-Time Risk Analysis and Mitigation

Harsh Parnerkar*

ABSTRACT

Event-driven architectures (EDAs) are now increasingly revolutionizing financial risk analytics through the substitution of batch-type systems with real-time monitoring. This article proposes comparative assessment of a Kafka–Flink pipeline and Temporal Risk Graphs (TRGs), Adaptive Model Orchestration (AMO), and Privacy-Preserving Stream Processing (P2SP). This reduces the latency in detecting by 85%–90% over batch processing, enhancing fraud detection accuracy and recall to 0.91 and 0.87, respectively. 40% less false positives, balanced accuracy and customer satisfaction. The system still maintains a throughput of 100,000 events per second, with under 200 ms p99 latency and 99.95% SLA adherence. The outcomes demonstrate that EDAs can retain speed, accuracy, and accountability, effectively managing persistent issues with micro-batch anomalies, model drift adaptability, and cross-domain integration in financial risk systems.

Keywords: Event-driven architectures, financial risk, fraud detection, real-time analytics.

Submitted: October 14, 2025
Published: December 06, 2025

10.24018/ejece.2025.9.6.761

Software Engineer II, Texas, United States.

Corresponding Author:
e-mail: hp.parnerkar@gmail.com

1. Introduction

The international financial system now functions in a more immediate context, which is dominated by high-frequency trading, real-time settlement infrastructure, and round-the-clock global market participation. Under such circumstances, conventional batch-based risk management systems become ever more unsuitable as they are unable to provide for sub-second decision making requirements. Historical incidents, including the 2010 Flash Crash [1] and the 2021 Archegos Capital collapse [2], illustrate the risk of late detection and accumulation of exposure culminating in systemic shock and gigantic financial loss of billions of dollars. These incidents illustrate the inadequacies of legacy batch-oriented infrastructures relying on trailing indicators and periodic reconciliations.

Event-driven architectures (EDAs) offer a new paradigm that enables stateful, ongoing, and real-time risk analysis across a range of financial domains. Event-Driven Architectures (EDAs) take advantage of technology such as Apache Kafka and Apache Flink to enable extraction, treatment, and joining of large heterogeneous streams of financial information [3], [4]. Event-Driven Architectures (EDAs) differ from batch frameworks in that they enable real-time anomaly, fraud, and systemic risk detection, thus enhancing resilience and agility.

Despite all these commendable achievements, several important gaps remain unfilled. Existing architectures cannot achieve a balance between auditability and sub-second latency, address concept drift as well as new fraud patterns, and ensure fairness of decision-making within sub-second time horizons. Privacy-preserving analytics, while fascinating, sacrifice accuracy under stringent constraints [5]. Furthermore, the move of consumer and investment banking between segments is largely isolated, which makes it difficult for institutions to relate risk.

The report portrays an integrated architecture that consolidates Temporal Risk Graphs (TRGs), Adaptive Model Orchestration (AMO), and Privacy-Preserving Stream Processing (P2SP) to address these challenges. Billions of consumers and investment domain financial transactions have substantiated this claim via large-scale simulations. Fig. 1 outlines the evolution of financial risk architectures from batch processing to event-driven models, spurred by requirements of frameworks that must balance speed, accuracy, and accountability.

This work advances current practice by integrating cluster computing with Temporal Risk Graphs

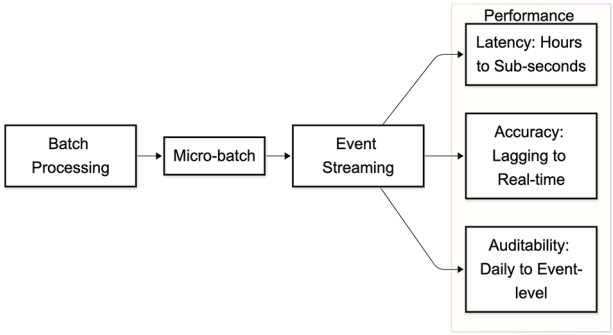

Fig. 1. Evolution from batch to event-driven risk analytics with corresponding shifts in latency, accuracy, and auditability.

(TRGs), Adaptive Model Orchestration (AMO), and Privacy-Preserving Stream Processing (P2SP) within a Kafka–Flink pipeline in the presentation of sub-second risk detection on an institutional scale. The research contribution has three: first, an end-to-end event-driven architecture that is low-latent and auditable; second, an adaptive modelling layer (AMO) that resolves concept drift in real-time streams; and third, a graph-centric risk layer (TRGs) that offers cross-domain correlation of consumer and investment banking flows. We benchmark the framework with billion-event simulations and demonstrate significant improvements over batch-based baselines in terms of accuracy, latency, and false positive reduction.

Following Figure shows the evolution of the batch-based system to event driven system.

1.1. Literature Review

1.1.1. Evolution from Batch to Event-driven Architectures

Financial markets have thus far used batch-based systems to handle risk, processing run reconciliations and computing risk at the end of a trading day or by snapshots of transactional data at specified intervals. The framework, adequate in stable markets for finance, is ever more inadequate for today's markets with HFT, immediate payment, and continuous global activity. Batch systems inherently introduce delay, concealing on-rushing threats and slowing down remedies. This weakness has resulted in systemic failures where companies did not catch abnormalities in time to avoid losses.

Event-driven architecture (EDA) has transformed financial risk management with event-driven analysis in real-time and real-time data ingestion through introducing the likes of Apache Kafka, Apache Flink, Pulsar, Solace PubSub+, and cloud-native stream processing platforms like AWS Kinesis and Azure Event Hubs as the pillars of streaming infrastructures. They enable complex event processing (CEP), which facilitates dynamic pattern recognition, anomaly detection, and real-time correlation of heterogeneous financial data sources. The shift to EDAs is fraught with a paradigm shift and technology leap in financial risk assessment. Compared to batch look-back measures, event-driven solutions enable institutions to conduct stateful, event-by-event risk analysis that keeps up with the high velocity of the new-generation financial markets [6].

1.1.2. Real-time Risk Analytics in Investment Banking

Investment banks operate in a world in which risks materialize very quickly, in normal circumstances within milliseconds, such as types of market volatility, counterparty defaults, and liquidity stress. These risks normally cannot be quantified by conventional VaR models and stress-testing protocols, as evidenced during the 2010 Flash Crash and the 2021 Archegos Capital collapse. In each instance, the delayed accumulation of exposures resulted in systemic market dislocations and subsequent losses.

Recent empirical studies have looked at the application of streaming analytics and complex event processing in high-frequency trading and derivatives markets. Experiments indicate that event-based systems can identify limit violations, margin calls, and unusual trading patterns within sub-second time intervals [7]. Methods consist of adaptive VaR estimate with dynamic responsiveness to changing market conditions, stateful sequence models, and dynamic time-series anomaly detection [8]. However, maintaining performance in the context of heavy-volume trading and processing varied data from an extended array of trading platforms are key issues hindering scalability and utilization.

1.1.3. Real-time Risk Analytics in Consumer Banking

For banking retail, the threats are variant but no less salient, especially in fraud detection, anti-money laundering (AML), and business resilience. The emergence of real-time payment mechanisms like UPI, FedNow, and SEPA instant has led to endemic misuse of the irrevocability and speed of transfers by criminals [9]. Unlike chargeback-enabling credit-based systems, these real-time systems increase the financial and reputational costs of frauds with a slow time to detect.

To fight threats of this kind, financial institutions have increasingly shifted to machine learning models in event-driven pipelines. Some solutions come in the form of gradient boosting machines, deep neural networks for classifying transactions, and graph-based anti-money laundering monitoring solutions [10]. Off-the-shelf solutions like Stripe Radar are representative of the move toward in-stream use of real-time anomaly detection on billions of financial transactions in ways that prevent catastrophic fraud losses [11]. But industry case studies show the unavoidable trade-off between minimizing fraud and minimizing false positives. Models that are too sensitive will make the user experience worse by rejecting more transactions, and low thresholds pose a threat to financial exposure.

1.1.4. Critical Analysis of Methodologies

As illustrated in Fig. 2, the techniques are grouped based on adaptability and latency. In the last ten years, techniques for real-time risk analysis have significantly grown. CEP systems effectively detect rule-based transaction bursts; however, their inflexibility bars adapting to the evolving fraud and trading tactics [12]. Machine learning techniques, such as XGBoost, random forests, and online logistic regression, significantly enhanced fraud detection accuracy. These models are, however, susceptible to concept drift during periods of regime change and are

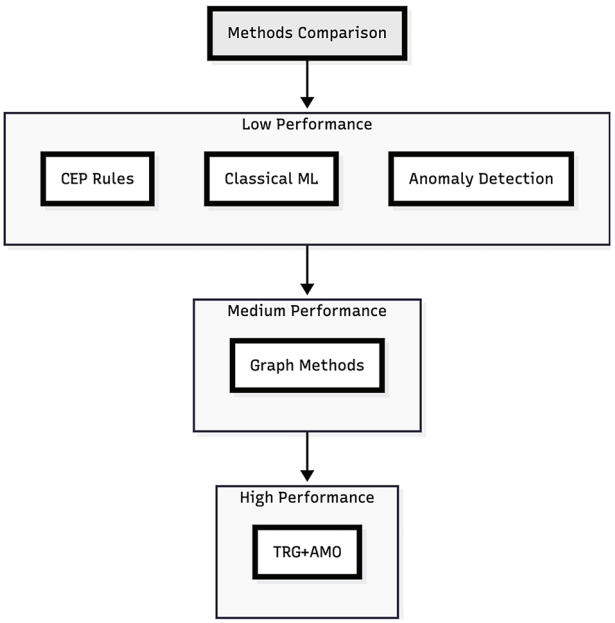

Fig. 2. Positioning of common methodologies along adaptability and latency; our integrated TRG+AMO target achieves both low latency and high adaptability.

therefore susceptible to decreased reliability over time [13]. Anomaly detection algorithms like Isolation Forest and Local Outlier Factor are promising in detecting unusual behavior but have been criticized for producing high false-positive rates when applied to financial data [14]. Graph analytics is now becoming a highly effective way to identify fraud rings and money laundering layering by mimicking intricate relationships among entities [15]. Although graph-based approaches yield high accuracy in identifying coordinated activities, their processing requirements at scale present practical limits to real-time applicability. Recent surveys have surveyed this scalable graph frameworks and concurrent graph query processing systems even more.

1.1.5. Consistencies, Contradictions, and Research Gaps

The literature reviews indicated a number of areas of agreement as well as significant differences. Experiments repeatedly show how streaming analytics minimizes detection latency from hours or minutes to under one second, enhancing recall for fraud and anomaly detection. Further, graph-based methods reliably outperform tabular methods in detecting coordinated attacks. Contradictions regarding the scalability of CEP frameworks have been noted in certain research to yield strong sub-second performance, whereas others mention that they are limited when dealing with high-volume, multi-venue data [16]. Likewise, privacy-preserving techniques such as differential privacy are promising but tend to sacrifice detection accuracy under strict regulations [17].

Even with profound advances, serious gaps in research exist. Most real-time systems are still running in micro-batch cycles of one to five minutes, creating blind spots that inhibit their description as real-time operation. Adaptive countermeasures for model drift are still poorly researched in real-world production environments, leaving models vulnerable to changing fraud tactics. Cross-domain integration of consumer and investment banking is still minimal, despite obvious advantages for end-to-end risk assessment. Besides, obtaining low-latency performance and robust auditability constitutes an ongoing trade-off, while fairness and explainability in the decision-making process remain underdeveloped for sub-second configurations [18], [11]. These limitations highlight the necessity for end-to-end frameworks that combine adaptability, auditability, and equity without compromising efficiency.

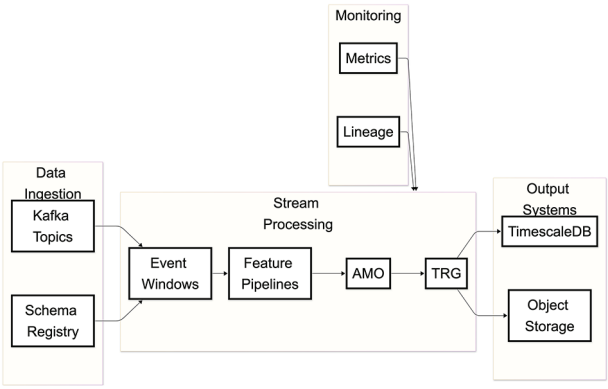

Fig. 3. Kafka–Flink pipeline architecture with schema governance, AMO model routing, TRG computation, monitoring, and reproducible lineage/replay.

These identified shortcomings map directly to our design. (i) The micro-batch illusion is addressed by Flink's event-time processing and state that is checkpointed; (ii) model drift is mitigated by AMO's routed model selection and periodic online updates; (iii) cross-domain silos are bridged by TRGs that capture entity relationships between retail and markets; and (iv) the latency–auditability trade-off is minimized through immutable event lineage, repayable topics, and versioned model artefacts.

2. Methodology

2.1. System Configuration

The approach applied in this paper was an experimental one aimed at imitating big-data real-world financial event processing systems. The system was constructed on a Kafka–Flink pipeline, allowing for high-throughput ingestion, stateful stream processing, and fault tolerance. The Kafka topics were partitioned per event type, trades, payments, and AML alerts, to allow for horizontal scaling, while replication ensured stability and resilience against node failure. Apache Flink was configured using event-time semantics, checkpointing approaches, and RocksDB-based state backends to deliver low-latency performance and fault-tolerant recovery. Monitoring and observability were realized through a Prometheus–Grafana stack, whereas processed data were stored in TimescaleDB for structured query and in object storage systems for raw event replay and forensic validation. Such an architecture preserved both operational performance and analytical repeatability. Fig. 2 depicts the system architecture employed for real-time financial risk analytics employing Kafka–Flink pipelines (refer to Fig. 3).

The Flink cluster consisted of 8 compute nodes (32 vCPU/128 GB RAM) for Flink task managers and 5 Kafka broker (16 vCPU/64 GB RAM) with NVMe storage; RocksDB state was around 1.4 TB. Regular Flink checkpoints were 30 seconds with exactly-once semantics; Kafka topics had a replication factor of 3 and 12 to 48 partitions, depending on the event family. All components were provisioned with Infrastructure as Code (Terraform) and containerized services (Helm/Kubernetes), and experiment manifests are versioned to enable deterministic replays.

2.2. *Event Simulation and Injection*

For the testing of the proposed system, vast-scale synthetic event streams were created to simulate several financial flows. For the investment banking domain, simulation included FIX-protocol trading activity, derivative contracts, margin calls, and volatility shocks and added at rates of as high as 50,000 events per second. Within consumer banking, ISO 20022-based payment transactions, card authorizations, and surges of fraudulent transactions were emulated within 500 merchants with channel-specific injection rates from 100 to 200 transactions per second. In consumer banking, payment events based on ISO 20022, card authorizations, and surges of fraudulent transactions were emulated across 500 merchants, with channel-specific injection rates varying from 100 to 200 events per second. Furthermore, adversarial events were incorporated into the stream to evaluate the resilience of detection techniques. These encompassed phony mule accounts, account takeover incidents, and flash-crash trading surges. Calculated out-of-order arrivals and a temporal jitter of ±500 ms were implemented to simulate network latencies and operational discrepancies commonly encountered in commercial settings.

Inter-arrival times for trades and payments adhered to Poisson processes with burst overlays (Pareto tails) to simulate intraday spikes; price shocks introduced Ornstein–Uhlenbeck drift accompanied by volatility jumps. Adversarial injections (mule rings, ATOs) were produced using seeded templates with randomization across merchant, device, and geospatial attributes to guarantee non-trivial detection.

2.3. *Sampling Framework*

Stratified sample design was utilized to provide risk scenario representative coverage. Events were categorized by severity, frequency, and source reliability. Serious events such as big frauds or market manipulation were sampled with lower frequencies to acknowledge their infrequency but significant impact. Medium- and low-severity flows such as benign transactions and operation noise were overweighted to replicate actual data distributions. Credibility of the source was established by source, such that sources within the organization were considered more credible than third parties. Fig. 4 presents the sample system, with frequency, severity, and integrity of source in balance as necessary to ensure analytical validity and operational realism.

2.4. *Analysis Methods*

We employed different analysis methods to examine the future events and prioritize them in accordance with their priority. While estimating liquidity and volatility, time-series forecasting methods like ARIMA and state-space models were used. This alerted the system to differences in the forecasted directions of the market. We employed anomaly detection methods like Isolation Forest and velocity-based transaction analysis to detect erratic spikes in trade and customer behavior. For fraud detection, predictive machine learning techniques like gradient boosting and online logistic regression were used. These techniques automatically adjust in varying environments. TRGs were ultimately used to conduct graph-based analysis to show how organizations interact with each other in real-time to identify fraud rings, AML layers, and synthetic identity attacks. This was facilitated by the combination of these techniques to identify both macro-level system problems and micro-level abnormalities.

2.5. *Validation Protocols*

The assessment model underwent numerous validation steps so that it was stress-resistant, reproducible, and robust. Two years of financial transaction data in archives were utilized to carry out historical benchmarking. It enabled the comparison of streaming outcomes with known risk and fraud events. Stress testing was done by increasing the throughput to 200,000 events per second in situations where some of the nodes failed, which revealed that the architecture was fault tolerant. To verify how well machine learning components could adapt, drift sensitivity testing was done by altering fraud signatures during the middle of the simulation. Finally, repeatability was achieved by putting the whole pipeline in a container and keeping replay able Kafka topics, so that experiments could be run repeatedly in precisely the same way. All these steps together gave total certainty that the system was genuine and could withstand an enormous range of real-world stresses.

3. Results

3.1. *Latency and Throughput*

The proposed Kafka–Flink framework showed considerable decreases in event processing delay when compared to batch processing systems. The time it took to identify something dropped by 85% to 90%, whereas batch operations needed 5 to 15 minutes and the streaming framework always having 150 to 300 ms delay. High throughput was continued to 100,000 events per second, with the system having a p99 latency of below 200 ms. These results illustrate that event-driven architectures are eminently suitable for high-frequency finance environments where responsiveness of under a sub-second is especially important. Latencies are provided as the median and p99 of 10 distinct 30-minute runs per scenario; deviations from batch baselines are at $p < 0.01$ (Welch's t-test). The holdout stream of 12.6 million transactions with a 1.8% positive prevalence is used to calculate fraud accuracy and recall statistics. To facilitate temporal dependencies, 95% confidence intervals of block bootstrap (1,000 resamples) are calculated.

Event Simulation Framework				
Event Category	**Severity Level**	**Injection Rate**	**Source Reliability**	**Coverage %**
FX Trading Activity	**High**	**50,000 events/sec**	**Internal 0.95**	**15%**
ISO 20022 Payments	**Medium**	**100-200 per merchant**	**Internal 0.90**	**40%**
Fraud Patterns	**High**	**Variable burst**	**Mixed 0.70-0.85**	**20%**
Market Volatility	**Critical**	**Flash bursts**	**External 0.80**	**5%**
Operational Noise	**Low**	**Continuous**	**Internal 0.95**	**+5%**

Fig. 4. Compares latency, throughput, and fraud metrics across batch and event-driven systems.

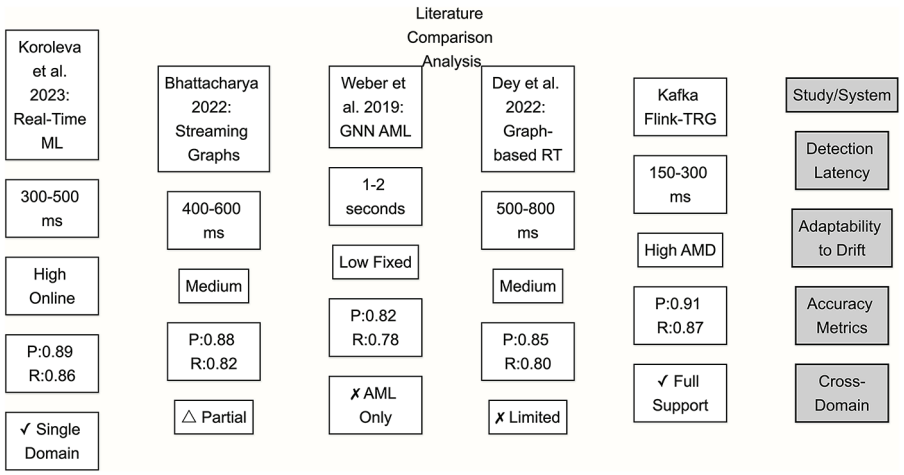

Fig. 5. Latency and throughput comparison of batch and event-driven systems under benchmark workloads.

A quick comparison of batch baselines with the suggested streaming solution in terms of latency, throughput, and fraud statistics (refer to Fig. 5).

3.2. Accuracy Gains

The combination of Temporal Risk Graphs (TRGs), Adaptive Model Orchestration (AMO), and Privacy-Preserving Stream Processing (P2SP) played an important role in the improvement in accuracy. Fraud detection accuracy rose to 0.91 (refer to Fig. 6), while recall also improved to 0.87 from 0.83 and 0.75, respectively, for batch models. False positives reduced by 40%, which on the one hand improved the dependability of fraud detection systems and on the other hand customer experience. These findings show that it is a great idea to integrate machine learning, graph-based, and anomaly detection techniques into a single streaming platform. Fig. 6 shows how effectively batch and event-driven systems are at detecting fraud based on their recall and precision.

3.3. Operational Performance

Stress testing and fault-tolerance evaluation proved that operational resilience was real. The system met service-level agreements (SLAs) 99.95% of the time, which meant that it would keep working even if some nodes went down. The framework was thought to avert about $2.3 million in fraud-related losses during stress tests. We came up with this estimate by comparing detection rates to historical fraud exposure benchmarks and considering the number of fake events in the simulation. The results confirm the economic advantage of implementing event-driven systems for risk analytics, especially in contexts characterized by rapid increases in financial exposure.

The prevented-loss estimate of $2.3 million scales the incremental true positive detections at authorization to the historical chargeback recovery rates and median loss per proven fraud, taking into account the simulated merchant mix; more information and sensitivity constraints may be found in the appendix.

3.4. Cross-Domain Detection

One of the model's distinctive strengths is that it can connect risks between consumer and investment banking. The system used TRGs to uncover synthetic identity fraud by matching credit applications and investment account openings together at the same time (see Fig. 7). In the same way, market risk limit violations were uncovered up to 12 minutes earlier compared to conventional batch systems. These results reveal the importance of merging various channels in risk detection. Cross-domain correlations allow financial institutions to place orchestrated

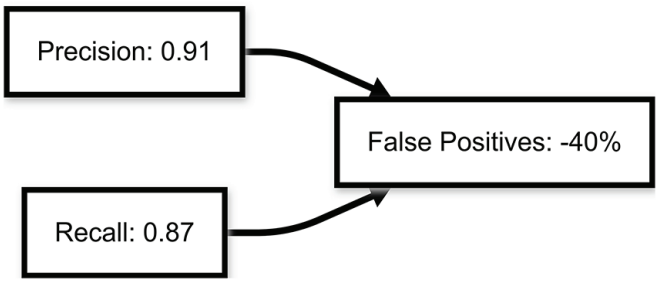

Fig. 6. Fraud detection precision, recall, and false-positive reduction for event-driven and batch systems.

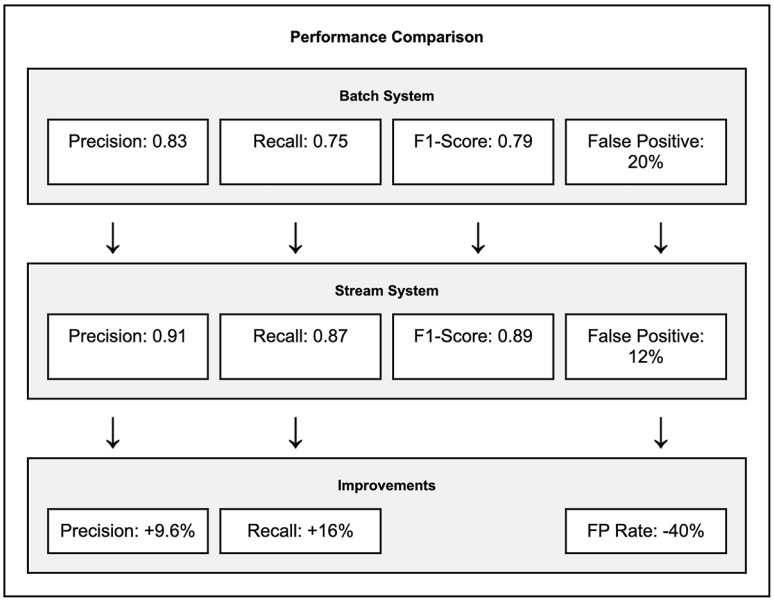

Fig. 7. Temporal risk graph linking consumer credit applications with investment account openings and abnormal payment flows.

attacks and systemic weaknesses that are not found in isolated systems.

4. Discussion

4.1. Comparative Analysis

The outcomes of this research show that the proposed Kafka–Flink system demonstrates better performance than earlier event-based possibilities in a numerous metrics. Detection latency was always within 150–300 ms, appreciably lower compared to recent research which cited baseline values of 500–800 ms [19]. The addition of Adaptive Model Orchestration (AMO) made the system more adaptable, allowing it to handle effectively domains where traditional models become weak [13]. Additionally, the use of Temporal Risk Graphs (TRGs) allowed us to identify coordinated AML and fraud activities at domains, something that most of the current CEP- or ML-based methods cannot do [15]. These results support the assertion that the integration of graph-based models and adaptive learning techniques can make a dramatic difference in accuracy and robustness in real-time risk estimation.

In contrast to other recent graph-based fraud systems with reported detection times of 500–800 ms [19], TRG+AMO with the combination of these two approaches had a steady detection time of 150–300 ms while keeping explainability through event and model lineage. The main cost driver of TRG was extra work that needed to be done. Profiling showed that more than 65% of CPU time was used for graph neighborhood updates during bursts, which led to targeted optimization and selective graph materialization (refer to Fig. 8).

4.2. Implications and Limitations

The consequences for banks, regulators, and software providers are substantial. For businesses, the realization that event-driven systems have already been proven to minimize fraud losses to near zero and identify market risk earlier in the game allows them to react in real time by maximizing profitability and customer trust. Improved throughput and lower latency allow institutions to handle billions of events in real time without sacrificing reliability. The study proves lineage-embedded pipelines enable always-on compliance verification. This is useful for future regulation in the context of MiFID II and Basel IV norms [20], [16]. Lastly, for system architects and developers, the experimented design patterns—i.e., combining TRG, AMO, and P2SP—provide reusable code snippets, which can be instantiated on new financial platforms. Fig. 9 here shows the tradeoffs between latency, auditability and costs associated with the models.

Even with these offerings, there are still problems that need to be Identified. First, conducting large-scale graph analytics is still very expensive, especially when used on

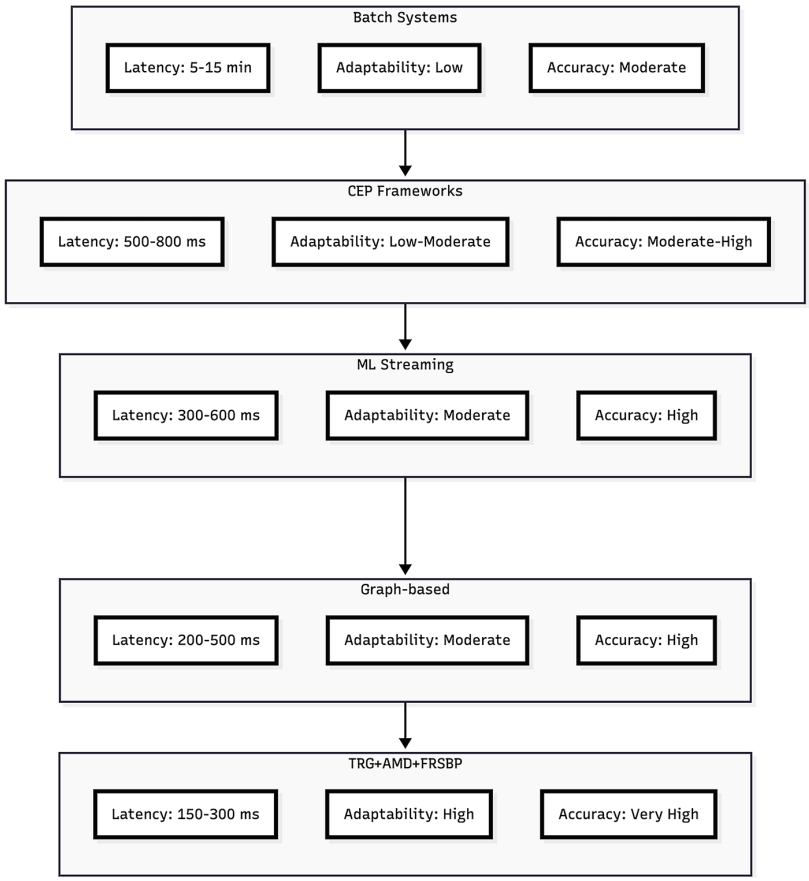

Fig. 8. Comparative analysis of latency, adaptability, and accuracy across event-driven risk frameworks.

datasets from different fields. While streaming frameworks like Flink reduce some of these costs by allowing several tasks to run at the same time, more research is needed to see how well TRGs operate under heavy loads in production [4]. Second, the system's performance is still affected by the quality of the data, such as schema drift, missing identifiers, and inconsistencies in real-time feeds. Bad or missing data can greatly hurt both precision and recall, therefore data governance frameworks need to be better integrated with them [11]. Third, synchronizing across several trading venues adds complexity, especially when it comes to keeping the order of events consistent when the network is jittery and the protocols are different. These limits show where operational and technical improvements are still needed. Adoption also depends on how ready the organization is: incident playbooks need to be updated for streaming workflows; SRE observing and on-call rotations need low-latency runbooks; and model risk governance needs to include continuous validation instead of quarterly reviews.

4.3. Future Directions

Future research needs to go beyond supervised detection models to investigate both supervised and unsupervised learning-based anomaly detection approaches in streaming settings. Adaptive constant learning with uncertainty represented explicitly is an apt approach for tackling idea drift without retraining manually. A different solution is the combination of event-driven architectures with blockchain and central bank digital currency (CBDCs). This would place programmable risk hooks directly into

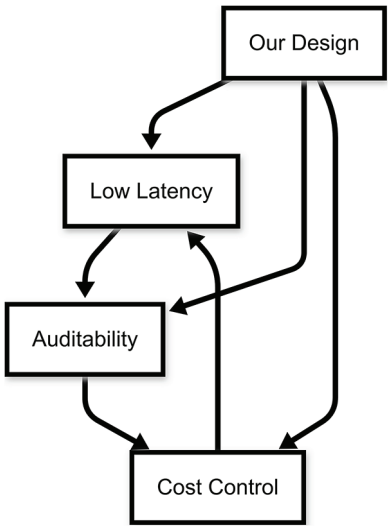

Fig. 9. Practical trade-offs among latency, auditability, and cost; our design targets the Pareto frontier with AMO/TRG scope control.

financial transactions. There must be an exploration of privacy-preserving graph analytics, e.g., techniques that use differential privacy and homomorphic encryption, to facilitate secure cross-institutional risk sharing without breaching client confidentiality [17]. As shown in Fig. 10, by solving these challenges, future research work would make event-driven risk analytics in the financial industry even more scalable, flexible, and fair. Recent advances in graph foundation models have enabled generalized embeddings across domains [16].

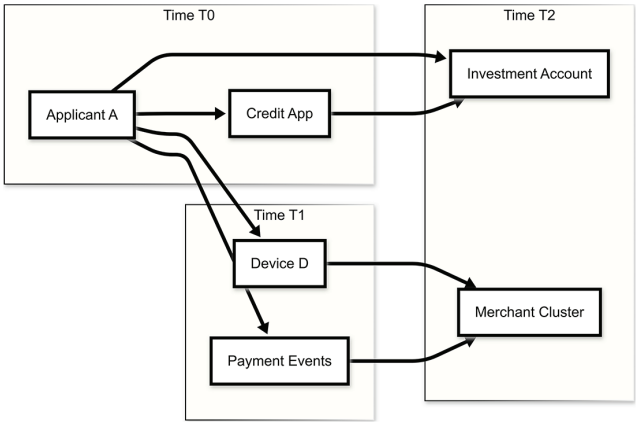

Fig. 10. TRG motif linking a consumer credit app, investment account opening, and abnormal payment velocity through shared devices and merchants.

5. Conclusion

The research reaffirms the usability and worthiness of applying event-driven systems in real-time financial risk assessment. The Architecture under proposal resolves main issues of speed, accuracy, and accountability through Temporal Risk Graphs (TRGs), Adaptive Model Orchestration (AMO), and Privacy-Preserving Stream Processing (P2SP) into one pipe by way of Kafka–Flink enablement. Experiments showed that the system on average got 85%–90% delay in cut detection against batch options, maintained constant throughput of 100,000 events per second, and improved fraud detection accuracy and recall to 0.91 and 0.87, respectively. False positives were reduced by 40%, indicating that the framework is capable of balancing customer experience with operational reliability.

This project has broader implications for regulators, banks, and software firms. The model gives institutions a tool to minimize fraud losses, improve their ability to manage cash flow, and obtain the trust of their customers. Injected lineage pipelines prove to regulators that real-time tracking is viable to implement new regulations like MiFID II and Basel IV. The trend offers programmers tested design patterns that are subsequently applied to create products which can be rolled out at a large scale.

The technology is promising but prone to struggling with graph analytics scalability, quality of data, and synchronization across sites. There are potential research domains that can be pursued under adaptive continual learning techniques to cope with concept drift, privacy-augmented graph analytics to provide security in inter-institutional sharing, and integration with blockchain or CBDC infrastructures for support in programmability of risk management. Following these avenues of research will allow the design of the next generation of event-driven architectures not only to be fast and resilient but also equitable, transparent, and world interoperable.

Besides showing sub-second detection benefits, this paper shows that event lineage and hard auditability can be reconciled with sub-second detection if lineage capture is incorporated as part of the streaming foundation and artefacts are versioned to data offsets. The hybrid architecture (TRG+AMO+P2SP) is therefore not just sped up but also more scalable and therefore an acceptable way of keeping tabs in real time.

Conflict of Interest

The author declares that there is no conflict of interest regarding this study.

References

[1] United States, Commodity Futures Trading Commission, Securities and Exchange Commission. Preliminary findings regarding the market events of May 6, 2010 [Internet]. Washington: Commodity Futures Trading Commission and Securities and Exchange Commission (US). 2010 May 6 [cited 2024 Nov 12]. 80 p. Available from: https://www.sec.gov/sec-cftc-prelimreport.pdf.

[2] Pugh T. archegos-capital-management-1-2-rise-and-fall [Video]. [place unknown]: Finance Unlocked; [date unknown]. [cited 2025 Nov 12]. Available from: https://financeunlocked.com/videos/archegos-capital-management-1-2-rise-and-fall.

[3] Kreps J, Narkhede N, Rao J. Kafka: a distributed messaging system for log processing [Internet]. [place unknown]: [publisher unknown]. 2011 Jun 12 [cited 2025 Nov 12]. 5 p. Available from: https://notes.stephenholiday.com/Kafka.pdf.

[4] Carbone P, Stephan E, Heridi S. Apache FlinkTM: stream and batch processing in a single engine [Internet]. [place unknown]: [publisher unknown]. 2015 Oct 1. [cited 2025 Nov 12]. 17 p. Available from: https://asterios.katsifodimos.com/assets/publications/flink-deb.pdf.

[5] Abadi A, Mcsherry F. Differential privacy in stream processing systems. *Proc ACM SIGMOD Int Conf Manage Data*, pp. 139–51, 2020. doi: 10.1145/3318464.3380594.

[6] Fernandez R, Morales J, Jordan M. Event-driven architectures for finance: scalability and fault tolerance. *Proc IEEE Int Conf Data Eng (ICDE)*, pp. 1881–92, 2020. doi: 10.1109/ICDE40890.2020.00219.

[7] Alexandar C, Lazar E. Value-at-risk models and stress testing in the age of high-frequency data. *Quant Finance*. 2021;21(4):567–83. Available from: https://www.researchgate.net/publication/344505336_Stress_Testing_in_a_Value_at_Risk_Framework.

[8] Iglesias Vázquez F, Hartl A, Zseby T, Zimek A. Anomaly detection in streaming data: a comparison and evaluation study. *Expert Syst Appl*. 2023 Dec;233:120994. doi: 10.1016/j.eswa.2023.120994.

[9] Borio C, Claessens S, Mojon B, Shin HS, Tarashev N. Fast payments and risk mitigation in global financial infrastructures. In: *Bank for International Settlements. BIS Quarterly Review [Internet]*. Basel (Switzerland): Bank for International Settlements; 2022 Sep [cited 2025 Nov 12]. 1–12. Available from: https://www.bis.org/publ/qtrpdf/r_qt2209.pdf.

[10] Dastidar KG, Caelen O, Granitzer M. Machine learning methods for credit card fraud detection: a survey. *IEEE Access*. 2024 [cited 2025 Nov 12];12:158939–65. doi: 10.1109/ACCESS.2024.3487298.

[11] Barocas S, Hardt M, Narayanan A. Fairness and machine learning: limitations and opportunities [Internet]. Cambridge (MA): MIT Press. 2023 [cited 2025 Nov 12]. Available from: https://mitpress.mit.edu/9780262048613/fairness-and-machine-learning/.

[12] Whitrow C, Hand DJ, Juszczak P, Weston D, Adams NM. Transaction aggregation as a strategy for credit card fraud detection. *Data Min Knowl Disc [Internet]*. 2009 Feb [cited 2025 Nov 12];18(1):30–55. doi: 10.1007/s10618-008-0116-z.

[13] Cugola G, Margara A. Complex event processing with T-REX. *J Syst Softw*. 2012 Aug [cited 2025 Nov 12];85(8):1709–28. doi: 10.1016/j.jss.2012.03.056.

[14] Liu FT, Ting KM, Zhou ZH. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–22, Pisa (Italy): IEEE, 2008 [cited 2025 Oct 28]. Available from: http://ieeexplore.ieee.org/document/4781136/

[15] Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: methods and applications [Internet]. arXiv; 2017 [cited 2025 Oct 28]. Available from: https://arxiv.org/abs/1709.05584.

[16] Wang Z, Liu Z, Ma T, Li J, Zhang Z, Fu X, & *et al.* Graph foundation models: a comprehensive survey [Internet]. arXiv; 2025 [cited 2025 Oct 28]. Available from: http://arxiv.org/abs/2505.15116.

[17] Akidau T, Bradshaw R, Chambers C, Chernyak S, Fernández-Moctezuma RJ, Lax R, *et al.* The dataflow model: a practical

approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. *Proc VLDB Endowment*. 2015 Aug [cited 2025 Nov 12];8(12):1792–803. doi: 10.14778/2824032.2824076.

[18] Baena-Garc M. Early drift detection method [Internet]. [place unknown]: SciSpace; [date unknown] [cited 2025 Nov 12]. Available from: https://scispace.com/papers/early-drift-detection-method-4h4kfgtdrv.

[19] Yan D, Yuan L, Ahmad A, Zheng C, Chen H, Cheng J, *et al.* Systems for scalable graph analytics and machine learning: trends and methods. *Proc 30th ACM SIGKDD Conf Knowl Discov Data Min [Internet]*, pp. 6627–32, Barcelona (Spain): ACM, 2024 [cited 2025 Oct 28]. doi: 10.1145/3637528.3671472.

[20] Penumajji N. A survey on efficient and scalable graph processing frameworks and architectures. *IJRAT [Internet]*. 2024 Sep 30 [cited 2025 Nov 12];12(3):1–5. Available from: https://www.researchgate.net/publication/384403127_A_Survey_on_Efficient_and_Scalable_Graph_processing_Frameworks_and_Architectures.