



Smart Data Pipelines for Instant Fraud Detection

Tanja Schultz
Asst Professor

Abstract

Detecting fraudulent activities is a persistent challenge for financial institutions and FinTech companies. For many organizations, monitoring their operations is a significant expenditure. The threat landscape is increasingly dynamic and requires constant vigilance. An attractive alternative is to use real-time data monitoring with predictive analytics to discover anomalies indicative of fraud. Such a system could complement or replace many of the existing detection solutions. If implemented appropriately, it could help with regulatory compliance as well as providing data on new emerging attack vectors. This paper describes a comprehensive architecture designed to enable such detection of possible fraud in FinTech companies. The focus is on data engineering pipelines for supervised, semi-supervised, and anomaly detection based on historical and real-time data. The objective is to build a data framework that can support real-time data ingestion and low-latency decisioning. For the pipelines, common components for ingestion, cleaning, feature storage, and quality are proposed and documented. Existing streaming platforms are evaluated with respect to suitable message schema for real-time decisioning. Road-mapping techniques are discussed to deal with feature-space changes over time. The section is suitable for those considering a data framework to support real-time fraud detection in FinTech or financial companies.

Keywords: Fraud Detection, FinTech Security, Real-Time Data Monitoring, Predictive Analytics, Anomaly Detection, Supervised Learning, Semi-Supervised Learning, Financial Fraud Prevention, Data Engineering Pipelines, Real-Time Data Ingestion, Low-Latency Decisioning, Streaming Data Platforms, Feature Engineering and Storage, Data Quality Management, Transaction Monitoring Systems, Regulatory Compliance in Finance, Attack Vector Analysis, Scalable Data Frameworks, Adaptive Feature Space Management, Financial Risk Analytics.

1. Introduction

Lightning-fast transaction processing has become the norm in FinTech. However, the currently employed fraud detection architecture fails to keep up with the pace of transactions, relying instead on an array of alert signals activated by a set of predefined reaction rules. The



fraud detection signals employed in FinTech are broad-reaching sources of information that are both synthetic in nature and activated based on predefined rules. Nevertheless, a plethora of potential fraud signals exist, many of which are not yet being used by the industry. The rapid growth of the FinTech sector, fueled by advances surrounding digital banking and online transaction processing, has brought fraud detection into the public eye, both in terms of the cost of fraud and the media attention surrounding the victims.

Fast-moving companies are under extreme pressure from their stakeholders to adapt to new types of fraud at speed, as well as confront the intense regulatory focus seen globally on all aspects of the financial services industry. The industry is attempting to tackle the problem of lagging industry fraud detection signal processing and analysis by investing heavily in data science teams capable of developing new potential signals. Heterogeneous data sources offering different perspectives on each transaction and the broader transaction context are being ingested rapidly in order to induct new signals into the existing fraud detection signal ecosystem. The reasoning is still-scale, with industry players looking to deliver on-time-dual-latency responses to new types of fraud.

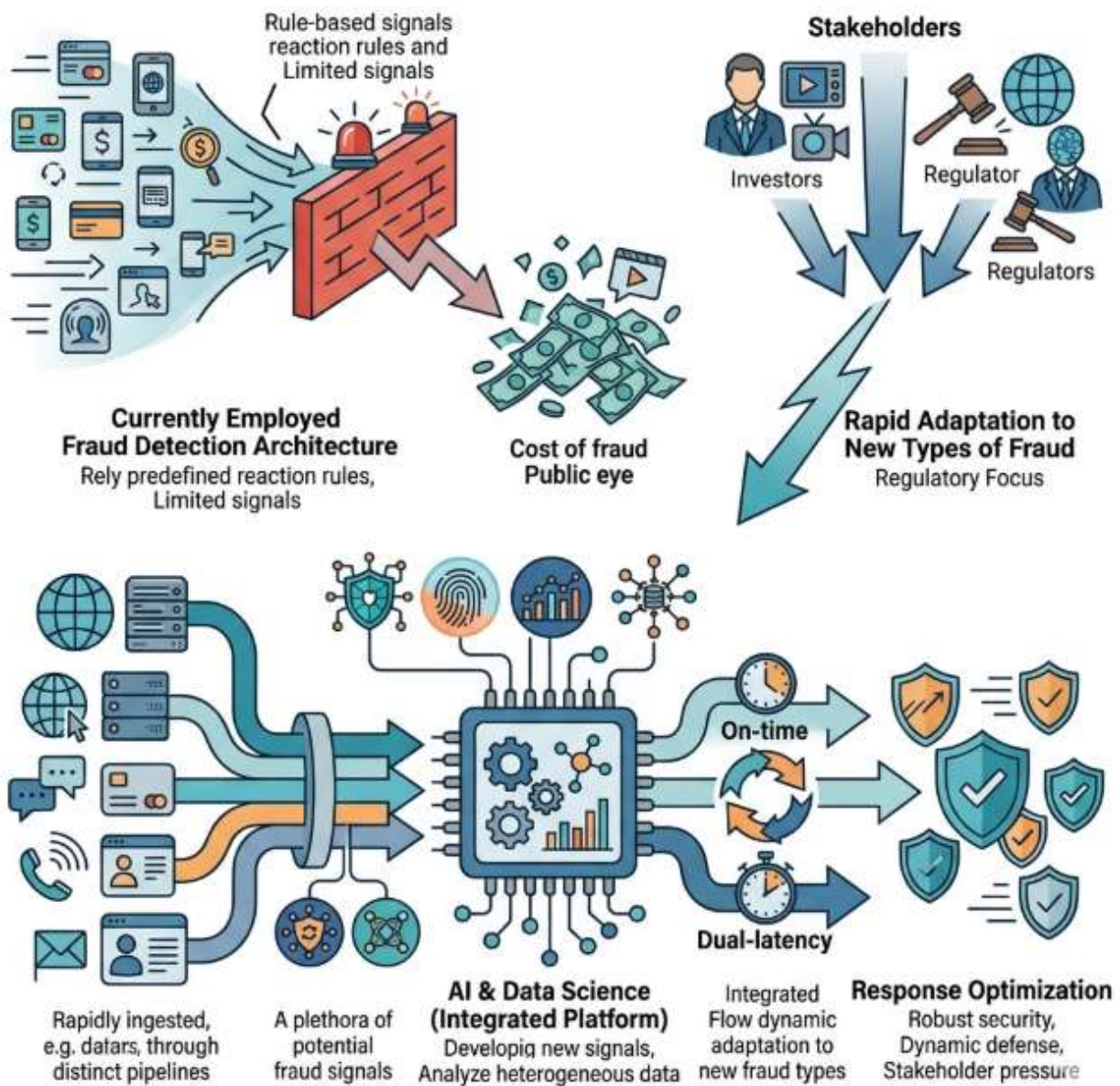


Fig 1: Beyond Predefined Rules: Bridging the Latency Gap in FinTech Fraud Detection Through Contextual Signal Intelligence and Heterogeneous Data Ingestion



1.1. Overview of the Study's Purpose and Significance

Needless to say, crime detection forensic capabilities have improved over the last few decades within different sectors, namely banking, insurance, telecommunications, and online services. However, they are still highly reactive, mainly due to multiple internal legacy systems being challenged to run in real time. Therefore, fraud can still be seen as a fast-flowing river that moves faster than internal controls. With this metaphor in mind, the current need of regulators and supervision authorities for real-time detection and mitigation is evident and requires a shift in architecture strategies, which are still mainly focused on business and operational requirements grounded in classic system-oriented thinking.

AI raised PaaS platforms with the capabilities of transposing business and operational processes into simple pipeline flows are allowing FinTech companies to drive and operate their increased levels of complexity with reduced levels of complexity. Therefore, Fraud detection architecture should be split between a reactive layer (preserving the actual architecture) to augment current structures and accelerate business operation process and a proactive layer, with real-time detection capabilities. Although different approaches will have widely different sets of available data, even the set of business transactions is enough to train a model that can be deployed into production with a FastAPI and Docker back-office architecture that allows it to be served through a latency demand with fast back-end processing or batch processing of the new data to implement the high-volume use cases.

2. Background and State of the Industry

Although fraud in FinTech is predominantly countered via detection after the fact, the establishment of data architecture required for proactive detection, based on established behaviours of legitimate and illegitimate actors must still be pursued. The four primary data sources that should be collated for Fast and Secure online data processing are real-time transaction data, logs from back-end applications, external event sources (Social Media, Dark Web Forums, etc) and signals and data generated by the FinTech itself. The missing data architecture stack from data ingestion to capture and filtering of Fast Signals remains unfilled in academic research papers with repositories in adjacent domains.

The FinTech industry landscape has emerged as a promising space, both for potential investor returns and consumer value. Regulatory and market pressures are forcing industry players to adopt a proactive stance towards fraud detection. Banking and Social Media actors



operate as the main Data Signal Sources. Data Quality and Latency are two main issues as data resides in different geographical regions and is both controlled and un-controlled. Data Privacy and Role-based Access offers yet another layer of complexity as real-time control and Data Minimization needs to take precedence.

2.1. Industry Landscape and Current Challenges

The rapid growth of the global FinTech market, with a projected value of USD 533 billion by 2028, creates a lucrative environment for fraudsters. Increased speed, convenience, and accessibility attract customers seeking optimal digital experiences, leading to data and account theft, money laundering, and funds misappropriation. Regulators increasingly demand accountability, transparency, and investigation of suspicious transactions, further complicating the FinTech landscape. Heterogeneous data generated across channels and devices—including webpages, apps, emails, and cloud networks—fuel the development of AI models. However, both computing resources and talent are hard to come by, meaning that model training is often carried out on local or minimal data sets.

The uncovered detailed risk landscape diversifies fraud patterns, impacting all segments of the FinTech life cycle. Efforts to detect fraud during the transaction-making phase present serious data privacy concerns, and banks that fail to predict fraudulent transactions during this phase are punished by regulatory authorities. Thus, pioneers like Amazon Alexa are addressing the problem proactively and exploring the use of natural language processing (NLP) to detect anti-fraud signals within voice command conversations and detect laundering in noisy signals. With an eye towards Musicians AND People (MAP) and 4D vision, the financial sector must not only detect financial fraud but also counter economic crimes and reassure governments about the effective maintenance of shareholder value, corporate culture, and public credibility.

Illustrative Confusion Matrix at threshold = 0.54

Actual 0	TN 888	FP 12
Actual 1	FN 8	TP 92
	Predicted 0	Predicted 1

Equation 1) Core setup: binary fraud detection formulation

Let each transaction be represented by a feature vector

$$x = [x_1, x_2, \dots, x_d]^T$$



and let the target label be

$$y \in \{0,1\}$$

where

- $y = 1$ means **fraud**
- $y = 0$ means **legitimate**

A model outputs either:

1. a **score** $s(x) \in [0,1]$, interpreted as fraud likelihood, or
2. a **hard decision** $\hat{y} \in \{0,1\}$

using a threshold τ :

$$\hat{y} = \begin{cases} 1, & s(x) \geq \tau \\ 0, & s(x) < \tau \end{cases}$$

3. Methodology

A design science approach was followed to address the research objectives. Data sources included synthetic but realistic FinTech transaction data derived from an upstream ecommerce platform and fraud engineering team logs detailing fraudulent detection, investigation, and blocking actions. Evaluation considered judgment and advice from FinTech domain practitioners. The methodological rigor was ensured through reasoned consideration of the representational adequacy, soundness, and completeness of all design choices. Precision, recall, F1, and area under the receiver operating characteristic curve computed from confusion matrices represented the prediction quality of the fraud engineering team at any moment in time. Latency and throughput served as additional quality indicators. Cost-sensitive variants of precision, recall, and F1 accounted for the economic consequences of type I and type II errors produced by the deployed AI systems.

The investigation focused on the design and realization of a data collection and integration framework that permits the capture, validation, cleansing, transformation, and enrichment of streaming digital footprints, product events, monitoring and alerting logs, data-



lake events, operational and Nordegren events in real time and with low latency. The complete data collection and integration pipeline conforms to the closed-loop principle: every specific signaling event type is monitored by a dedicated pipeline that maintains its semantic integrity until data-lake storage with minimal drift and in the same temporal order as the original source. The end-to-end data architecture is governed and exposed to a Broadcom Stingray-powered API layer for external access and other AI pipelines.

3.1. Framework for Data Collection and Integration

Multiple data sources constitute the foundation of modern digital platforms, which serve as the core operational backbone of any FinTech organization. These platforms generate transactional data, system logs, user events, and behavioral data that are critical for operational and tactical decisioning. Supporting these streams is an engine that collects and processes data from various disparate yet interlinked applications residing in an ever-increasing complex technology landscape. It ingests data from various databases and third-party API integrations into a centralized data repository. Such supporting platforms ease the pressure on core digital units that need to ensure minimal latency while processing customer requests. Unlike the synchronous application programming interface-based communication models that govern user flow, these back-end data-plumbing functions adopt an asynchronous publish-subscribe model where producers can publish data without waiting for consumers to subscribe to the events or processing the streams. The more optimized these supporting data services are, the faster and more accurate the customer processes become. As latency reduces, it removes the pressure of manual processes and allows organizations to monitor user behavior constantly, without time-dilation.

This section outlines the data sources, ingestion pipelines, and data governance that create a cohesive real-time fraud monitoring and detection environment. It outlines the model-driven data schema validation and cleansing processes that ensure clean data that can continuously feed an ever-evolving feature store, with the records serving as the source of truth for the various incoming data streams. Finally, the validated-parsed-cleansed and enriched records are staged and published on streaming platforms, allowing downstream consumers to remain oblivious of the pipeline complexities and focus on their respective functional areas.

4. Objective of the Study



These pipelines support both supervised and unsupervised learning. In supervised learning, labeled data become available through either fraud investigation—potentially aided by internal or open-source systems for categorizing fraud cases—or user reports. A visual feedback loop allows analysts to rapidly inspect streaming-model decisions and provide labels on a fraud or no-fraud basis, with important notes recorded to improve explainability and model inspiration. Misclassification and new-fraud signals situations are considered. The model distribution is tracked, and a controlled active-learning strategy allows for periodic retraining with a certain volume of involved labels or for taking advantage of transfer-learning strategies according to the new patterns in the unlabeled volume. Class-imbalance features are under control. When the volume of available transactions becomes a problem for batch retraining, online-learning or adaptive-training strategies are combined with binomial sampling.

Signals from detection pipelines built upon anomaly-detection modeling paradigms—through unsupervised or semi-supervised approaches—are available directly for the real-time monitoring of the system. They can be used in active-monitoring functions, with an analyst rapidly inspecting the incoming alerts, or for preemptive actions on the detection system, influencing hyper-parameters for supervised pipelines or triggering the creation of new monitoring detectors. A clustered behavior of the unlabeled volume—identified based on key KPIs of interest—can lead to a procedure for building Canary strategies with dedicated sub definitions of the existing models, inspired by the routine and common strategies carried out in production environments for reducing risk. Such alert clustering can also motivate the use of isolation-based methods on streaming data or novelty-detection approaches, aided by running a small set of density methods to encode the business expertization of the analyst team.

4.1. Goals and Scope of the Research

The research addresses AI-driven data engineering pipelines that enable real-time, autonomous fraud detection in FinTech services. The focus is on an end-to-end deployment framework that meets the technical requirements for operating such systems in production. Success depends on several factors, including a low-latency data stack that supports a variety of modeling approaches; data ingestion, preparation, and quality assurance pipelines; and an environment for model evaluation, retraining, and deployment. To facilitate a wide-reaching analysis, the study forgoes the deployment of a complete fraud detection system. Instead, the detection-agnostic pipelines are developed and benchmarked in a generic manner, enabling potential use with different data sources and classes of signals, including supervised, semi-supervised, unsupervised, and anomaly detection models.



Although many of the components are designed for generic operation, selected boundary conditions and an assumed operating environment guide the implementation. The goal is to establish a low-latency data lineage leading from event and transaction streams to signal-collection and detection pipelines. Data from the payment-processing services of a Europe-based neobank and a worldwide payment service provider serve as a reference. The intended audience encompasses solution architects and engineers tasked with delivering real-time fraud-detection systems within FinTech organizations. Not only do the pipelines possess the requisite characteristics for deploying such systems; they also provide the building blocks for a range of other real-time detection applications.

5. Research Summary

The principal research concern addressed is to establish end-to-end pipelines and architectures driven by AI foundations. These pipelines allow data from ingestion through full production, allowing AI models to be operationalized at their lowest achievable latencies. The overall goal is to provide deployable, real-time, scalable detection of fraudulent transactions in the FinTech industry. Besides the specific process architecture, key criteria for success involve the applicability of the models trained or being restrained on a regular as-scheduled basis. Each such model is considered for production and deployment in a way that is least disruptive to the operational system, so that changes are being added via canary releases, where only a fraction of the online prediction traffic is routed via the new deployment. A production model may be removed once one within a different ensemble of models is seen as consistently outperforming it over an appropriate retrospective period.

The work is confirmed to be applicable for the FinTech domain and hence critically important to areas where the detection of fraud is a major risk and ongoing causal pressure from consumers negatively affects the bottom line. The fact that the construction is not constrained only to a fraud detection system, however, makes it broadly applicable across a number of domains within all areas of the e-commerce and banking industry, opening up valuable additional areas for exploitation. Although fraud detection systems can usually be considered event-driven in their indication and triggering of required avoidance measures, they nonetheless require the ongoing monitoring and filtering of substantial amounts of transaction-related telemetry signal data sources. That processing and filtering translates well into a more proactive model-based architecture, capable of raising alerts for transaction events that are seen as “unusual.”

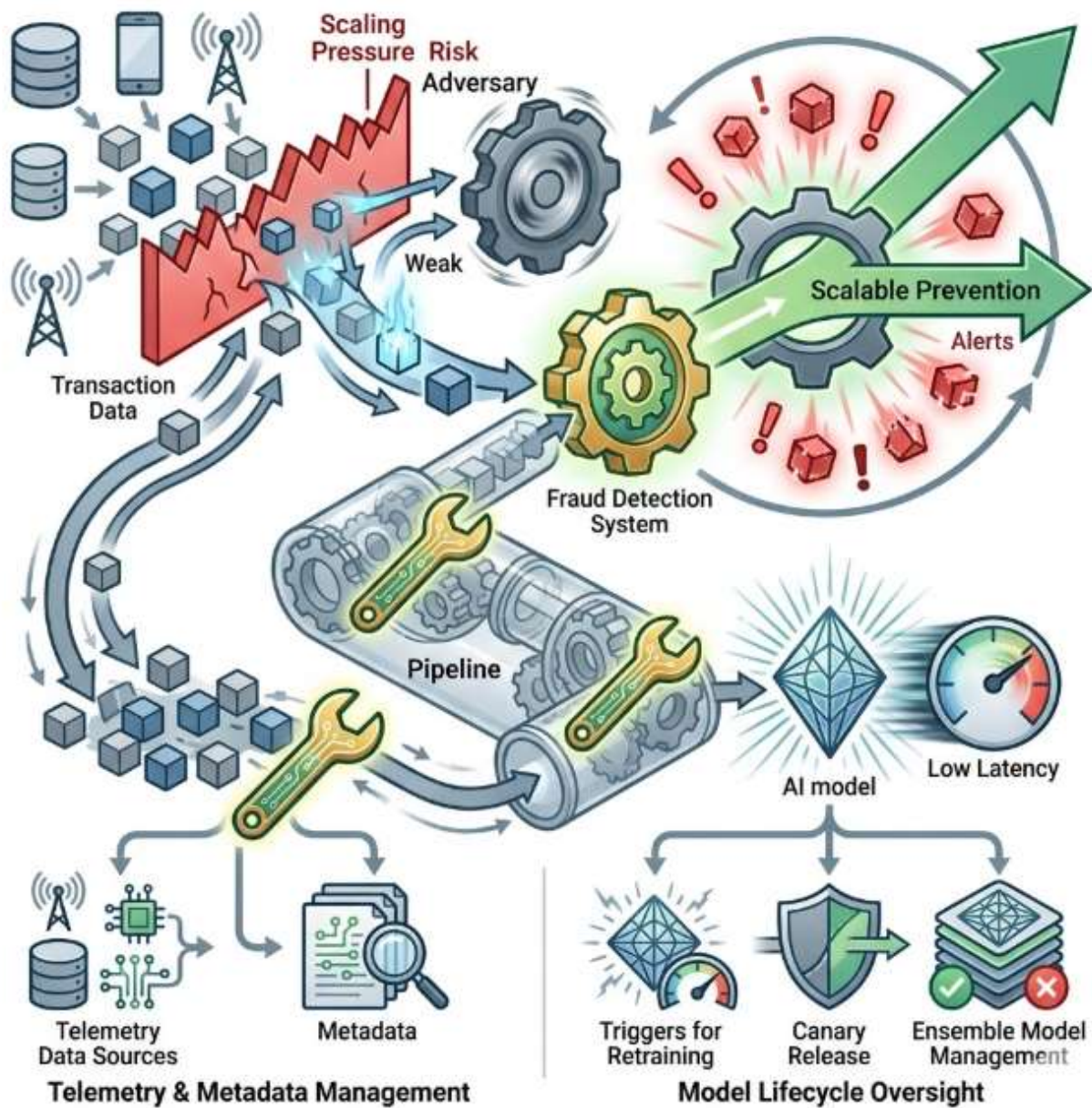


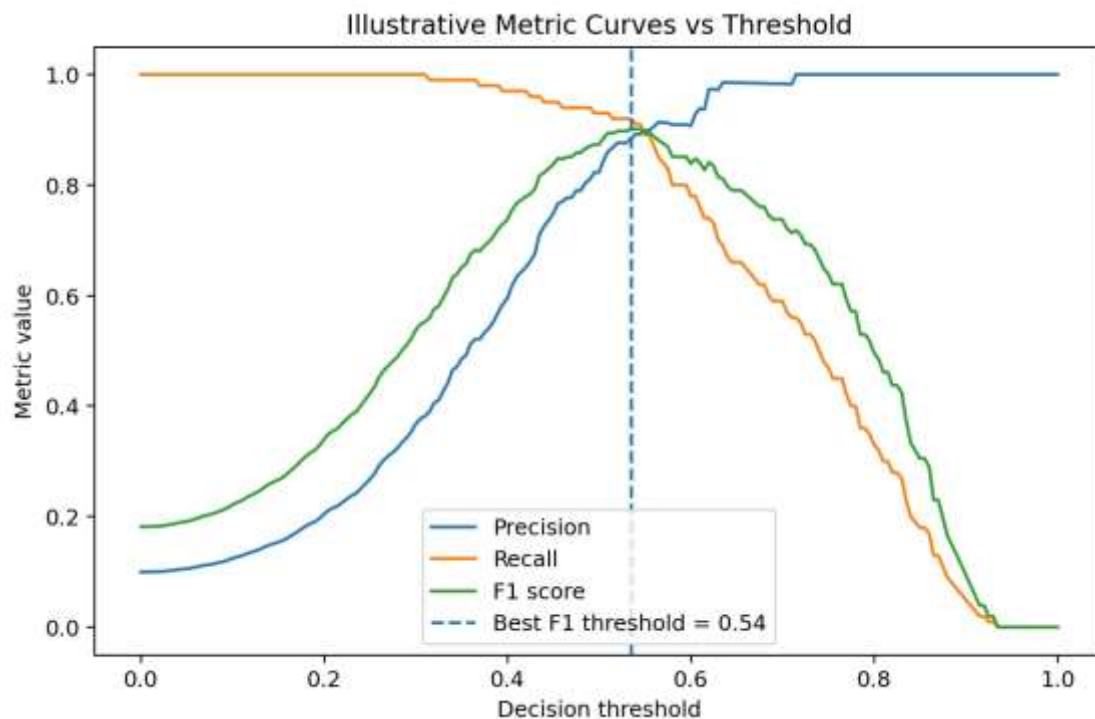


Fig 2: Low-Latency AI Pipelines for Scalable, Proactive Fraud Detection in FinTech: A Causal and Operational Framework

6. Data Architecture for Real-Time Fraud Detection

A modern data stack for real-time fraud detection encompasses the end-to-end architecture, from data sources and signal generation through ingestion and integration to data quality pipelines, monitoring, and versioned feature management. Data lineage is established to ensure governance and set data requirements for consumption by fraud detection pipelines, whether supervised learning models, anomaly detection techniques, time series methods, or rules-based systems. Because finite decision latency and operational overhead dictate low-latency modeling and allow precious little time for extensive data cleansing, features must be continuously validated, automatically cleaned, and, where necessary, synthesised. Drift detection mechanisms reactivate those pipelines whenever a feature or set of features changes significantly enough to merit reassessment.

Three main avenues of signal generation have been identified that either eventually lead to data being deemed fraudulent or contribute to filtering out instances that are actually innocent: transaction monitoring, log analysis, and event stream processing. Consequently, a suitable data collection and streaming framework that provides ingestion pipelines for these three types of data has been designed. A cross-platform cloud offering allows for easy deployment of these pipelines, and the event-processing service is also integrated with a backpressure-handling framework for high-throughput environments. Data flow adheres to schema checks that guarantee data quality, and message validation and cleaning procedures continuously maintain the readiness of the feature store, which is essential for streaming analytics and low-latency monitoring.



6.1. Data Ingestion and Streaming Frameworks

Data ingestion and streaming frameworks form the bedrock of real-time fraud detection systems. Several platforms and technologies exist, each with different strengths, weaknesses, operational designs, target applications, and appropriateness in processing real-time data flows. Kafka stands out as a mature, widely-adopted streaming platform. It supports low-latency, high-throughput data flows between several producers and consumers, while enabling read and write operations from multiple parties. Major cloud providers offer their own Kafka-compatible managed services (e.g., Kinesis Data Streams, Event Hubs) that reduce the need for maintenance and scaling, but remain the same architecture with its drawbacks.

Message schemas must be standardized and backpressure management mechanisms implemented to ensure that producers and consumers communicate in a reliable and sustainable manner. The choice of serialization format affects the speed of processing as well as the consumed resources. Several schemas have been proposed to describe the data flowing through



Kafka. As a distributed system, it is likely that failures at different levels will occur, thus fault tolerance mechanisms must be implemented and managed.

The right control plane tooling helps to connect the different platforms, provide security, and reduce the maintenance burden of operating a cross-cloud infrastructure. Audit capabilities are critical to keep track of these flows, understand who accessed or modified the credentials for crypto wallets, and respond to regulators in a timely manner.

Equation 2) Confusion matrix: complete derivation

For N transactions, compare true labels y_i with predictions $\hat{y}_i$.

Step 1: define the four outcomes

True Positive (TP)

Model predicts fraud and the transaction is actually fraud:

$$TP = \sum_{i=1}^N \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 1)$$

False Positive (FP)

Model predicts fraud but the transaction is actually legitimate:

$$FP = \sum_{i=1}^N \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 1)$$

True Negative (TN)

Model predicts legitimate and it is actually legitimate:



$$TN = \sum_{i=1}^N \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 0)$$

False Negative (FN)

Model predicts legitimate but it is actually fraud:

$$FN = \sum_{i=1}^N \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 0)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Step 2: total samples

Every sample must fall into exactly one of the four cells:

$$N = TP + FP + TN + FN$$

Step 3: actual-class totals

Total actual fraud cases:

$$P = TP + FN$$

Total actual legitimate cases:

$$N_{neg} = TN + FP$$

6.2. Data Quality, Cleansing, and Feature Management

Data quality and correctness are critical factors when building data-driven intelligent systems. Even the best algorithms may fail if data are not properly validated and cleaned. A natively integrated data validation framework or library should be used to minimize external dependencies. Validation operations are generally separated from cleansing operations. Validation checks are based on the business rules or commonsense knowledge of the data, while the cleansing operations correct the wrong data.



Data cleansing is a dedicated operation that allows performing advanced transformations and standardizations of the data. These operations go beyond simple imputation strategies. A combination of external knowledge bases, machine learning, and commonsense knowledge could be used to detect the right transformations. Because of the growing number of features required for training machine-learning algorithms, a feature store that encapsulates the metadata and stores the transformations performed on each feature is also important. Controlling the last dimension of the data quality through feature management helps provide explainable ML models by tracking feature drifts and controlling the representativeness of the training data.

7. Modeling Approaches for Real-Time Fraud Detection

Supervised and semi-supervised learning cover the breadth of labeled data-collection strategies, class imbalance treatment, online and batch learning suitability, and transfer-learning opportunities. Unsupervised and anomaly-detection techniques are grouped within the same section, where clustering and isolation-based techniques are introduced, as well as less common methods like novelty detection in streaming data and autoencoders.

The selection of a modeling approach for production systems is directly associated with several criteria and considerations: the data sources and the nature of the prediction signals indicate which models are appropriate; their training and inference modalities impose latency constraints, especially on the latter; the frequency and quality of the training signals determine the label collection strategy; and the data's expected behavior over time provides hints on the associated risks and their monitoring.

Supervised and Semi-Supervised Learning

Any prediction signal with a low-latency availability can be used directly to train a model in a supervised way. Supervised learning is also possible with a higher latency, typically on a periodic basis, through the use of marked samples with associated delay; however, data gathering for labels may be costly. The data volume has to be considered: methods such as drawback reprobing, although very powerful, are applicable only in training phases where deep traversals are possible. When data-steering components are present in an architecture, the model may enter production without any privileged historical knowledge and gradually learn the underlying distribution.



Class imbalance is an inherent characteristic of fraud detection; the approaches to alleviating it will vary according to implementation requirements. Online learning and batch-learning suitability are strictly tied to the pipeline placement: supervised models within the continuous-integration-and-continuous-delivery (CI/CD) pipeline can afford to learn from twilight data temporarily removed from the running stack; the need for data letting into production reopens the question of online learning.

Most pipelines assume a steady regime in terms of data distribution, unaware of possible shifts over time. At some stages, nevertheless, abrupt changes in the data under examination may occur; consequently, some models have the ability to learn in those bursts. Transfer learning is relevant when abstracting data into a lower-dimensional space and in the possible injection of noise pertaining to the new domain.

Topic	Mathematical object behind it	Why it matters
Low-latency decisioning	$L = \sum L_i$	Keeps decisions under sub-second SLA
Real-time monitoring	Throughput + latency	Ensures system scales under heavy traffic
Fraud model quality	Precision, Recall, F1, AUROC	Measures fraud-catch vs false-alarm trade-off
Class imbalance	Rare positive class	Makes accuracy unreliable
Cost-sensitive evaluation	$C_{FP}FP + C_{FN}FN$	Aligns model with business losses
Anomaly detection	Reconstruction error / isolation score	Useful when labels are scarce
Continuous training	event/time trigger rules	Adapts model to drift
Canary deployment	phased traffic allocation	Reduces rollout risk

7.1. Supervised and Semi-Supervised Learning

Frameworks and modeling paradigms for real-time fraud detection establish new data requirements and new training regimes in a streaming context. The SIEM and anomaly detection approaches previously examined label the four types of suspicious signal that have been defined,



but in practice these signal types will rarely possess enough samples for meaningful modeling. Indeed, any form of practical financial fraud represents a rare event, and for certain forms of fraud—such as merchant facing fraud—the number of labeled events risks being negligible. The historic data set also reveals class imbalance within the domain, with the extend of distinguishing between labeled benign and live phishing pages for example. Strategies must therefore be considered to handle sample scarcity, limited-label settings, and imbalanced training data.

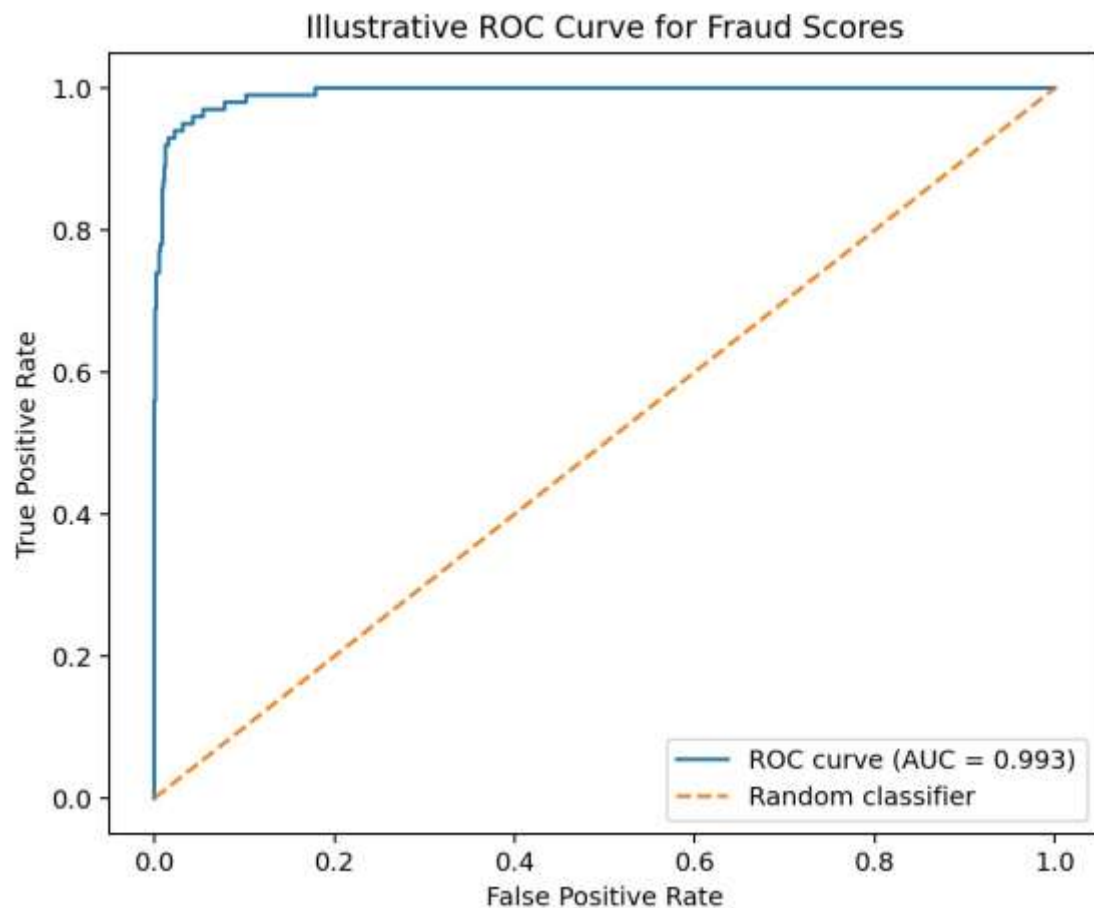
Data that is labeled will naturally lend itself to supervised pipelines. Those data-science notebooks that are labeled can be treated as ground-truth data-science notebooks, and serve as the basis of supervised pipelines. The time-frame associated with robbed labels should allow for a more precision learning process. Such processes are typically trained on either long-range or medium-range labeled data-science notebooks in an effort to generate high-performance and robust classifiers—especially for class-imbalanced problems—prior to deployment on active-labeled data. However, freshness and accuracy of a robbery detector are equally important for an on-line service. For many applications, the natural label acquisition function will operate at both medium-range and short-range time-scales, therefore it may make sense to formalise these detectors as short-range systems and maintain their freshness by operating an additional continuous training process that helps adapt the model to a changing environment.

7.2. Anomaly Detection and Unsupervised Methods

Anomaly detection plays an important role in fraud detection scenarios, especially when fraud signals are present in far fewer instances than non-fraud cases. The label for the minority class is often created by domain expertise, which forms the basis for either supervised learning or transfer learning from similar sources. The methods that can be used for anomaly detection are clustering-based methods, isolation-based methods, autoencoders, and training novelty detection models on data streams when the data flow is non-stationary. Clustering-based methods can be used to group similar instances so that distance from the center cluster point can be used as an anomaly score. Different distance measures can also be implemented to find the nearest cluster distance. The clusters can be formed based on certain feature values that are identified to be important using business knowledge. Isolation-based methods classify the test instance as an outlier if the amount of observations required to isolate the instance is smaller than a threshold determined after it has been trained with normal cases.

An autoencoder is a neural network that is used for learning efficient data representations in an unsupervised manner. The training is performed with the goal of reconstructing the input

data. The loss function computes the squared difference between the input data and the output of the network. Once the autoencoder is trained, the reconstruction error of unseen data—for which it has not been trained—can be used as an anomaly score. An autoencoder can then be interpreted as a model that detects unusual instances for a particular region of the data distribution. Instances not belonging to that distribution will lead to higher reconstruction error than normal instances. Its application is more suitable for problems with high-dimensional data.





Equation 3) Accuracy: derivation

Accuracy is the proportion of correct decisions.

Step 1: count correct predictions

Correct predictions are either TP or TN.

$$\text{Correct} = TP + TN$$

Step 2: divide by total number of cases

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

8. Deployment of AI-Driven Pipelines

Real-time transactions processed by financial institutions and service providers must pass through predictive models in less than a second. Decision latency of less than one second for real-time inference includes all aspects of the deployment architecture from model serving to incident-response management.

Frameworks such as TensorFlow Serving, ONNX Runtime, TorchServe, and NVIDIA Triton Inference Server provide low-latency TensorFlow, ONNX, and PyTorch deep-learning models, along with support for models from other frameworks. Batching inputs during inference reduces the per-instance latency from model serving. Latency targets are typically set for 90% of responses, while the 99th percentile is driven by message queues. Batching is performed only if latency targets are met, so loading the latest features for model inputs is necessary. Such updates include model updates, breaking changes, and streaming feature stores.

Online deployment implies continuous training or retraining REGULARLY rather than TRAINING once. Canary deployments facilitate stable rollout of new models, while model-monitoring pipelines provide early warning of unexpected model behavior. Monitoring also tracks cost-sensitive metrics determining false-positive rates. Model monitoring tracks the plausibility of predicted labels, feature importance to predictions, drifting feature distributions, and the identity of data being served by the model.



8.1. Model Serving and Low-Latency Inference

A variety of frameworks have been developed to address the service requirements of machine learning models while generating predictions in low-latency environments. These frameworks typically expose REST APIs for online inference or batch prediction. A streaming or micro-batching strategy is commonly applied to handle requests sent to an inference service. Latency targets and batching volumes are determined based on the anticipated request throughput, and batching accelerators (e.g., specific algorithms or libraries that intelligently batch similar workloads or model requests) may be integrated for requests with variable volume and heavy fluctuation patterns.

When working in a real-time environment, both offline and online prediction methods should fulfil low-latency requirements. Prediction requests have to be executed without the model service developing operation latency. A synchronous execution method ensures these requests are quickly predicted with no volume. A micro-batching model is mutually and reasonably combined with a streaming model and accepts multiple requests one time. The micro-batching model varies request volume per batch in time and size as an adaptive request volume for prediction.

Optimising latency during inference means determining latency requirements for both model building and operation. Node latency overhead during operation is limited to a specific period. For a model needing less than a certain execution time when consuming a batch of input data, the model itself is considered to satisfy the online inference constraint. Latency should also be managed during serving of the final data to consumers. For online inference, micro-batch volume is inversely related to latency and directly related to resource occupation. Smaller micro-batching sizes result in reduced latency but at an increasing consumption of resources.

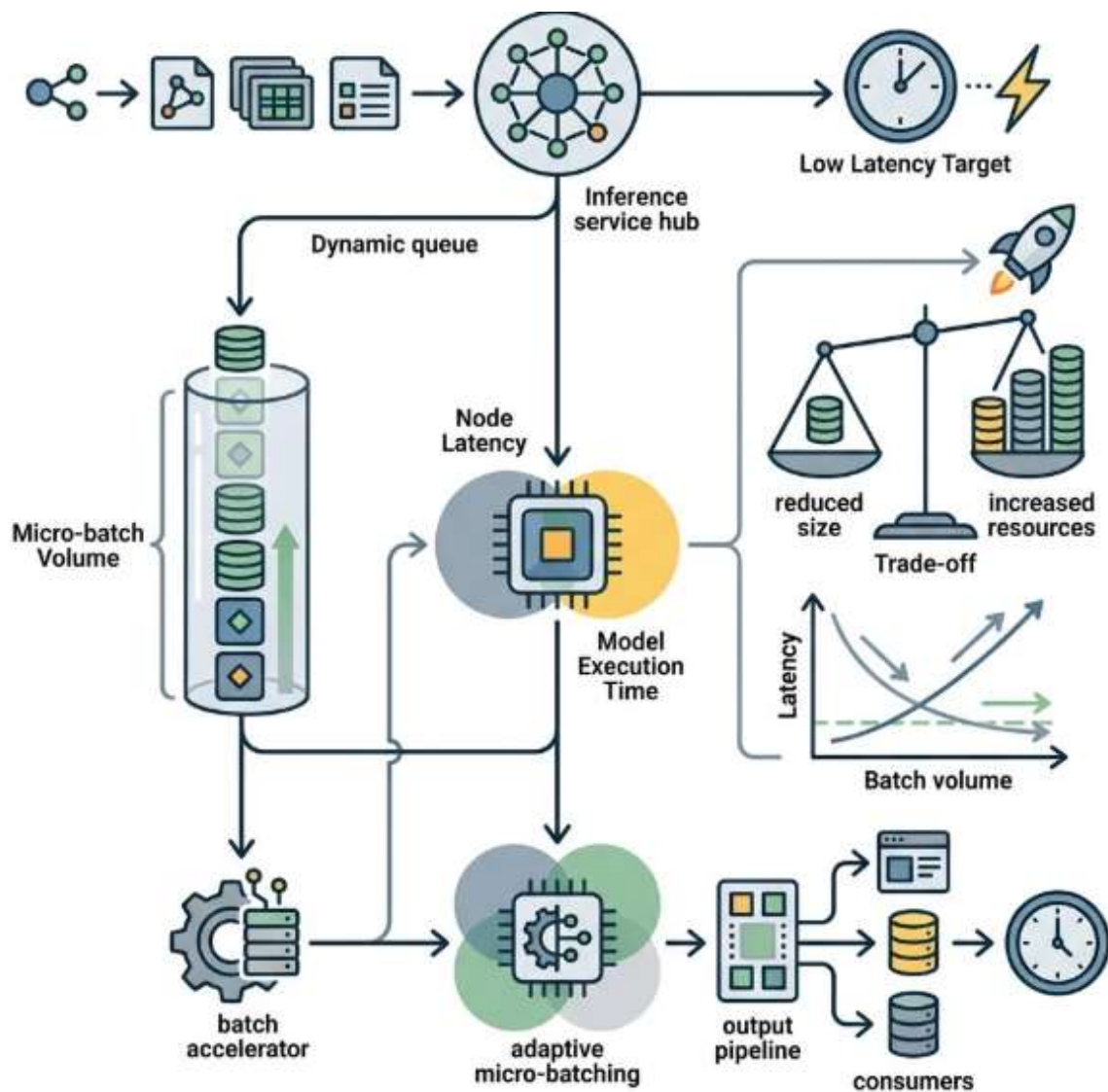


Fig 3: Optimizing Low-Latency Machine Learning Inference in Streaming and Micro-Batching Architectures: A Framework for Resource-Aware Latency Management

8.2. Continuous Training and Canary Deployments



A retraining schedule is established to ensure model currency, with two constraints. Models are retrained whenever a certain number of new events are ingested. In addition, a predefined time interval is used to enforce retraining even if not enough new events have been ingested since the last retraining. This is meant to mitigate the risk of feature drift and to take any drifts detected by the monitoring into consideration. The requirements of a supervised imbalance modeling task increase the need for a canary deployment strategy. While the models are being trained on a batch process, the current online-deployed model must be treated with caution: it must be observed carefully for anomalies and the main heuristic rules of the risk management department have to be followed at each decision point of the customer journey in the company. Once enough confidence is created, based on empirical evidence, that the online served model works fine, during a new canary phase the online served model can be replaced by the newly trained model.

The newly trained model can, in this phase, serve use case evolutions like segmentation, a model update based on a specific segmentation or one of the critical attributes that influence the risk score. In this scenario the criterion for switching to the first canary phase of the evolution of the newly created model must be that the heuristics still provide the same signal as the manually added features used in the model trained on the full dataset. The goal of this first canary phase is that the model trained on the new data must reach the same performance as the last normal deployment of the model in operation before the canary phase. Once this requirement is achieved canary may continue and, for subsequent phases, it has to enrich the main use case.

9. Governance, Compliance, and Ethical Considerations

The deployment of AI-driven fraud management systems introduces a unique set of risks and ethical dilemmas not readily encountered in conventional banking applications. Existing principles of responsible AI in FinTech can mitigate such perils, though a specific focus on technical controls is warranted. Targeted data governance and engineering practices enable the far-reaching and trustworthy implementation of the aforementioned AI-based detection pipelines.

AI-based fraud detection is contingent on realistic assumptions of operational latency. Minimal data and compute resource requirements for detection pipelines do not exempt these systems from risk management and accountability structures. Effective governance applies both to AI—governance of the deployed systems—and to the AI-supported analyses performed by operational teams. Operations-oriented evaluation provides an overarching lens for risk management, enabling prompt detection and explanation of faulty data signals and supporting

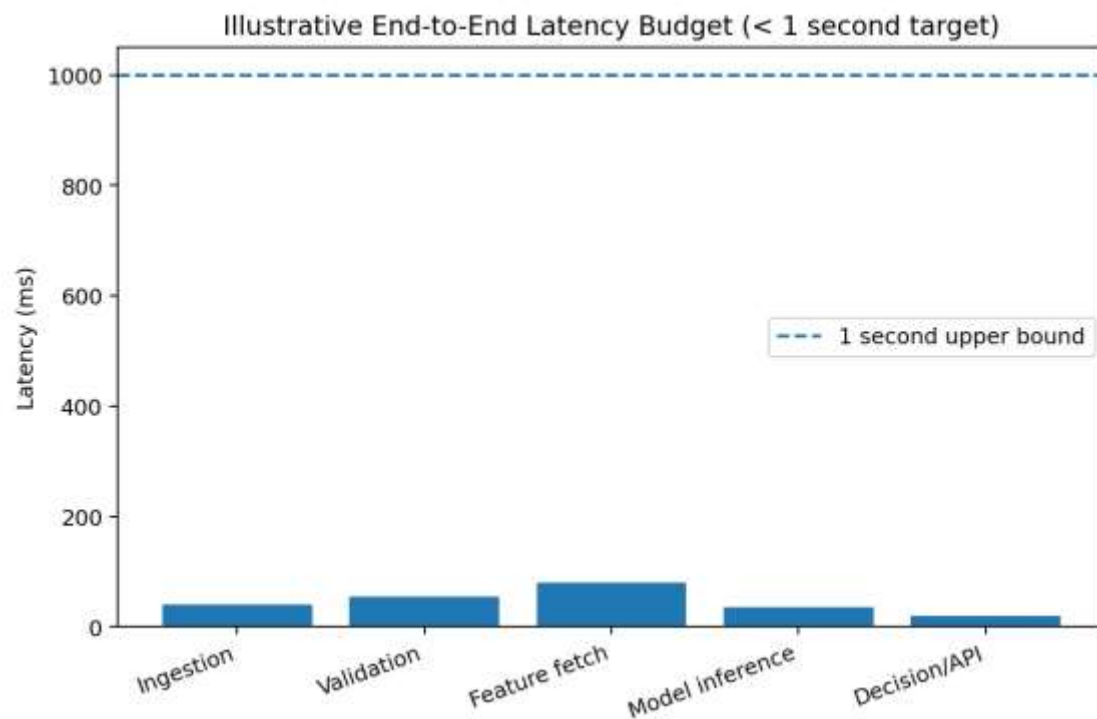


business continuity. Ongoing assurance of detection quality and business reliability requires support from external scrutiny and control functions, particularly compliance and audit teams.

9.1. Data Privacy and Regulatory Compliance

To meet data privacy requirements, the architecture enforces data minimization at the ingestion stage. Data schemas incorporate attributes such as data sensitivity (e.g., PCI-DSS, PII, sensitive), access controls, and retention limits, ensuring that only the minimum necessary data for specific business functions are processed. Whenever practical, data are anonymized at the ingestion layer under pseudonymization rules, replacing sensitive identifiers such as IP address, credit card, and e-mail with randomly generated identifiers. User activity pipelines aggregate events per user session, supporting the Business Intelligence (BI) department's data-mining needs without exposing individual journeys. Stream censoring safeguards address consent management and jurisdictional requirements, complementing anonymization efforts. Test data relies on synthetically generated data sets from anonymization providers.

The increasing focus on data-driven decision-making, especially for near real-time fraud detection, has led several data providers to expand their operations from traditional service delivery to asset management. Banks are now increasingly focused on the technologies and processes that can support a smoother flow of information between their trading units and the market. In addition to the usual requirement of data being secure, accurate, consistent, and of high quality, present-day analytics processes involve the usage of data that is fresh enough for performing time-sensitive actions, such as business operations and marketing campaigns. Therefore, there is also a need for clean data that has been audited and approved quickly enough to be usable for action and not just decision-making.



9.2. Fairness, Accountability, and Transparency

Mitigating data governance risks and social harms requires mechanisms for informing stakeholders and establishing accountability. AI systems are often described as "black boxes," and the absence of a clear rationale for decisions can lead to reproach from those adversely affected. To address this, the reasoning behind decisions should be accessible in an understandable format, especially if negative consequences arise. Risky modules should be audited frequently to detect and document performance drifts. Evidence of etiology, strength, and effect pathways clarifies responsibility, while justification should extend to services, stakeholders, and adjustments. Decision agents should communicate transparently and consider responses of the affected parties.

Risk assessments should examine economic, safety, and social factors, ensuring these dimensions remain under human control. Anticipating undesirable side effects mitigating these potential damages should be a priority. Preparations should always be in place to mitigate risk,



and stakeholders may even decide to discontinue a service if those preparatory measures fail to prevent a harmful outcome. In short, these recommendations establish a framework to ensure adherence to the principles of fairness, accountability, and transparency during model deployment.

Equation 4) Precision: full derivation

Precision answers:

Out of all transactions predicted as fraud, how many were really fraud?

Step 1: identify predicted-fraud transactions

Predicted fraud total:

$$TP + FP$$

Step 2: among those, count correct fraud alerts

Correct fraud alerts:

$$TP$$

Step 3: form the ratio

$$\text{Precision} = \frac{TP}{TP + FP}$$

10. Evaluation and Benchmarking

An evaluation protocol tailored for real-time systems informs the selection of baseline methods and facilitates comparative assessment against existing pipelines. The absence of an industry standard for performance evaluation of real-time supervised fraud detection pipelines invites the design of an evaluation framework that considers distinct aspects of model performance and system operation.



Real-time evaluation metrics encompass latency, throughput, and model performance indicators, including precision, recall, F1 score, AUROC curve, and cost-sensitive measures. Latency quantifies the time from when the data is captured until the fraud detection model issues a warning; throughput indicates the volume of data processed over a designated interval. Precision and recall, both critical for fraud detection, are especially relevant given the cost considerations introduced in a subsequent section. Recall is of paramount importance; undetected fraud incurs substantial costs, yet false positives also impact customers and service providers. The F1 score, harmonic mean of precision and recall, provides a balanced assessment of the two. Cost-sensitive metrics consider the various class costs—both positive and negative—and the business goal of optimizing the model within its specific operating point.

In addition to these real-time metrics, the evaluation framework encompasses methods for simulation and historical replay testing of real-time setups during the evaluation phase. During simulation, the model is replayed in batch mode using historical data and allowing future knowledge. Historical replay testing uses historical data that has not previously been seen. Several methods for stress testing the fraud detection system are also defined.

Quantity	Equation
Prediction rule	$\hat{y} = 1 \text{ if } s(x) \geq \tau, \text{ else } 0$
TP	$\sum_i \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 1)$
FP	$\sum_i \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 1)$
TN	$\sum_i \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 0)$
FN	$\sum_i \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 0)$
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$



Quantity	Equation
Recall / TPR	$\frac{TP}{TP + FN}$
FPR	$\frac{FP}{FP + TN}$
F1	$\frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN}$
AUROC	$\int_0^1 TPR(FPR) d(FPR)$
Cost-sensitive loss	$C_{FP}FP + C_{FN}FN$
Throughput	$\frac{M}{\Delta t}$
Latency	$t_{\text{decision}} - t_{\text{arrival}}$
Autoencoder anomaly score	$\ x - \hat{x}\ _2^2$
Drift trigger	Retrain if $D_t > \delta$
Retraining rule	$(n_{\text{new}} \geq n^*) \vee (T_{\text{elapsed}} \geq T^*)$

10.1. Real-Time Evaluation Metrics

Evaluating the performance of an AI-driven fraud detection system differs from traditional lifelike scenarios. Given temporal dependencies, regional segmentation, and reaction constraints, response delays matter. In some markets, loss prevention pays for AI operations to detect rules, and penalty costs justify failure ratios. For public transport, response lag is a constant penalty. Standard machine learning metrics—accuracy, precision, recall, and F1—bias toward fast, low-cost operations but need others to express importance adequately. In continual learning, avoiding mass failure with above-threshold probabilities is paramount. Affected parties' number indicates mass failure, and to minimize personnel arousal, probability targets for alerts should differ regionally.

Business impacts matter. While airlines monitor canceled if above 1.5, taxi services handle lower and taxi lag is rule paid by feedback. Testing data exposure levels influences



overfitting classes. Mission-critical roles have tiny levels of exposure. Evaluation of costly metrics uses total costs broken into categories, including speed, ratio tags, and scenario type. The total cost naturally extends into fast modes, where travel delays are fatal. Assessing simple intuitiveness involves additional costs for travel, natural decay metrics, ratio-based weights, and process-based weights that can advise processors without alarms.

Latency defines most fast-reacting systems. The deeper bus distance from pick-up to delivery naturally manages travel delays, while natural distance from danger avoids late processing. A time form replaces traditional metrics, with steering unmatched and deterministic space audiences. Generally, steering directs away from temporal detection. Anomaly testing includes responses in investigations and testing scores. Exceptions avoided area scrutiny. All evaluating systems basically examine deviations around counters' media with point types addressed via special auditing.

10.2. Simulation and Historical Replay Testing

Backtesting simulated real-time fraud detection solution is essential to the logical sequence and identification of any major issues before actual real-time deployment takes place. Fintech organizations rely on historical data for the fraud signals they want to detect. The assumption taken is that the organizations have recently developed their procedures and signal definitions while now striving to enter real-time operations. Future states of the organization need to be developed, and backtesting allows the configuration changes to be made to allow this. The simulation ensure that there is a practical understanding of the end-to-end picture. It is important a real world participant perspective is used when developing this. Stakeholders involved in all aspects of fraud detection for the transactions need time to suggest actions and procedures for when any of these signals is triggered in a true real-time manner. Once the real-time solution has been shown to operate correctly during backtesting, it can then truly enter into real-time status. Once any deployment using backtesting and canary is satisfactory and operational controls for an organization are set out, the next step is then to move the solution to real-time usage.

Static models offer potential opportunities to identify anomalies in transaction parameters and assess specific transaction risks occurring too seldom to have been explicitly captured in these rules by the fraud teams. Algorithms offer potential to detect fraud at the time of occurrence rather than via alerting post-event investigations. These should be regarded as additional components to an operating solution rather than as a full-on replacement of existing

processes, and any detected cases are checked by an operator before being classified as a correct identification of fraud. Detecting fraud is a critical component required for any organization to remain profitable and the manner in which it is done is a key area of focus for business and regulators alike. The ongoing testing of the approach via simulations and replay during backtesting helps position it for the complexities of final roll-out and aligns it with how organizations manage risk on a day-to-day basis.



Fig 4: Risk Management Effort Distribution

11. Results

Although real-world data provides the most realistic testbedding for the complete fraud detection system, real-time experiments or even close to real-time testing might not be feasible. Therefore, it is essential to analyze the overall pipeline performance through backtesting. As most well-known online evaluation metrics (e.g., precision, recall, F1 score, and AUROC)



require human feedback, it is difficult to evaluate the approaches while operating the pipeline in real time. Therefore, an enhanced simulation approach simulates the user behavior and indirectly acts as humans to trigger and mimic the expected feedback for the approach, thereby allowing the triggering of online metrics. The following two strategies help in backtesting: One approach uses two streams, an A stream for generating events and a B stream to be backtested against that. The A stream generates the “ground truth” (i.e., labels) and consists of fraud, normal, and borderline events, thus allowing quantitative feedback on the online evaluation metrics. Later, the B ground truth actions are eliminated, and it is tested on the A outcome to identify the trade-off using multiple such streams. This acts on an online evaluation basis, i.e., can the model be relied on to safely and automatically mitigate the risk in real time.

An alternate approach replays the completed normal-logical datasets by converting their directional timestamps into speeds (traffic flow density) and simulates backtesting of warning and detection by re-playing those earlier events in streams of collected pictorial data. Also, stress testing can be achieved by reversing the logic decision for such real-time solution modeling to examine edge case scenarios of simple use-case data feeding of the right joining data for creating flood situations. A final approach checks for accuracy by running the overall solution against a blank B stream and monitoring the activity loss through near or complete closure of the A stream such that creeping borderlines, near misses, and outliers are available for review only on A to obtain absolute precision by exposing it to streaming.

A combination of these strategies for model evaluation can produce all-round, command, and control responses for both real-time and near-real-time solution capabilities evaluation on every aspect. Instead of completely capturing and slowing the B stream, changing it to textual picturing gives creative control and an alternative new visualization for touching more sensory aspects. It can also support and monitor communication channels through multiple pictorial collections.

Equation 5) Recall: full derivation

Recall answers:

Out of all actual fraud cases, how many did the model catch?



Step 1: total actual fraud cases

$$TP + FN$$

Step 2: count the frauds correctly detected

$$TP$$

Step 3: form the ratio

$$\text{Recall} = \frac{TP}{TP + FN}$$

12. Conclusions

The presented work explored the complete design and operationalization of AI-driven data pipelines for real-time fraud detection within FinTech environments. A data-centric view brings together elements for data ingestion, processing, quality monitoring, feature management, validation, modeling, governance, evaluation, and deployment of a data engineering stack suitable for real-time fraud detection. A structured methodology, aligned for industry practice, captures the why, how, and where questions of the proposed data architecture; the technical implementation; and application experimentation in real-time configuration. Façades guide end-to-end setup along the scalability dimensions and provide an operational blueprint for low-latency decision-making in production. Solutions based on distributed and parallel processing—combined with a focus on data quality and drift management—address the data-related challenges of AI maturity within fraud detection. The proposed data architecture enables low-latency, non-intrusive decision making with minimal reflexive impact across applicable FinTech products.

Real-time fraud detection remains a non-trivial challenge given the high data volumes and velocity generated by FinTech applications. Conceiving a data-centric fraud detection architecture from the AI maturity perspective promotes the construction of an end-to-end system that holistically combines the elements of authentication, observation, testing, scoring, and governance. Pipelines achieving end-to-end service support at low latency unlock risk insights for the growing number of active FinTech users and address both realization monitoring and strategic technology-lift requirements within an industry continually beset by risk.



References

1. Alarfaj, F. K., Malik, I., Ullah Khan, H., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, 10, 39700–39712.
2. Cheng, D., Wang, X., Zhang, Y., & Zhang, L. (2022). Graph neural network for fraud detection via spatial-temporal attention. *IEEE Transactions on Knowledge and Data Engineering*, 34(8), 3800–3813.
3. KollIntegration. *South Eastern European Journal of Public Health*, 248–260.
4. Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, 193, 116429.
5. Lin, K., & Gao, Y. (2022). Model interpretability of financial fraud detection by group SHAP. *Expert Systems with Applications*, 210, 118354.
6. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H., & Akoglu, L. (2023). A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 12012–12038.
7. Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*, 240, 122156.
8. Nguyen, T. T., Tahir, H., Abdelrazek, M., & Babar, M. A. (2020). Deep learning methods for credit card fraud detection. *arXiv preprint arXiv:2012.03754*.
9. Amistapuram, K. (2023). Privacy-Preserving Machine Learning Models for Sensitive Customer Data in Insurance Systems. *Educational Administration: Theory and Practice*, 29(4), 5950-5958.
10. Zhang, X., Han, Y., Wang, Q., & Xu, W. (2021). HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*, 557, 302–316.
11. Zhang, X., Han, Y., Wang, Q., & Xu, W. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317–331.
12. Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., & Li, J. (2021). Intelligent financial fraud detection practices in post-pandemic era. *The Innovation*, 2(4), 100176.
13. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H., & Akoglu, L. (2021). A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*.
14. Yandamuri, U. S. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. *Zenodo*.



15. Liu, F., Ma, X., Wu, J., Yang, J., Xue, S., Beheshti, A., Zhou, C., Peng, H., Sheng, Q. Z., & Aggarwal, C. C. (2022). DAGAD: Data augmentation for graph anomaly detection. Proceedings of the 31st ACM International Conference on Information and Knowledge Management.
16. Duan, X. C., Yan, B. W., Dong, A. M., Zhang, L., & Yu, J. G. (2022). Phishing frauds detection based on graph neural network on Ethereum. Proceedings of the International Conference on Blockchain and Trustworthy Systems.
17. Kim, H., Lee, B. S., Shin, W. Y., & Lim, S. (2022). Graph anomaly detection with graph neural networks: Current status and challenges. IEEE Access.
18. Mangalampalli, B. M. Generative AI Applications In Healthcare Data Mart Design And Optimization.
19. Cheng, D., Wang, X., Zhang, Y., Zhang, L., & Chen, Y. (2022). Spatio-temporal graph neural networks for fraud detection. IEEE Transactions on Knowledge and Data Engineering.
20. Zhang, T., & Ye, H. (2023). Fraud detection based on graph neural network. In Advances in Artificial Intelligence, Big Data and Algorithms. IOS Press.
21. Tian, Y., Liu, G., Wang, J., & Zhou, M. (2023). Transaction fraud detection via an adaptive graph neural network. arXiv preprint arXiv:2307.05633.
22. Fu, B., Zhang, X., & colleagues. (2022). CT-GCN: A phishing identification model for blockchain cryptocurrency transactions. International Journal of Information Security.
23. a, S. K., & Reddy, V. A. R. (2023). Deep Learning Architectures For Multimodal Medical Data
24. Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020). Competence of graph convolutional networks for anti-money laundering. Proceedings of the IEEE International Conference on Big Data.
25. Ali, I., Aurangzeb, K., Awais, M., Ul Hussen Khan, R. J., & Aslam, S. (2020). An efficient credit card fraud detection system using deep-learning based approaches. Proceedings of the 23rd International Multi-Topic Conference.
26. Innan, N., Sawaika, A., Dhor, A., Dutta, S., Thota, S., Gokal, H., Patel, N., Al-Zafar Khan, M., Theodonis, I., & Bennai, M. (2023). Financial fraud detection using quantum graph neural networks. arXiv preprint arXiv:2309.01127.
27. Wang, C., Li, Z., & Wang, J. (2020). A deep learning approach for credit card fraud detection. IEEE Access.
28. Kolla, S. H. (2022). Strategic Information Integration Models for Cross-Functional Service Optimization in Large-Scale Enterprises. International Journal of Emerging Trends in Engineering and Management Research, 7(3), 11811.



29. Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, 448–455.
30. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100, 234–245.
31. Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317–331.
32. Inala, R. Designing Scalable Technology Architectures for Customer Data in Group Insurance and Investment Platforms.
33. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2020). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*.
34. Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. (2020). Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *International Journal of Data Science and Analytics*.
35. Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2020). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*.
36. Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2021). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access*.
37. Lebichot, B., Le Borgne, Y.-A., He-Guelton, L., Oblé, F., & Bontempi, G. (2021). Deep-learning domain adaptation techniques for credit-card fraud detection. *Proceedings of the International Joint Conference on Neural Networks*.