Scientific Research Publishing

International Journal of Intelligence Science, 2025, 15(4), 162-183
https://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Risk-Aware AI Models for Financial Fraud Detection: Scalable Inference from Big Transactional Data

Utham Kumar Anugula Sethupathy[1,2]

[1]Independent Researcher, Atlanta, GA, USA
[2]Alumni, Nanyang Technological University, Singapore City, Singapore
Email: ANUG0001@e.ntu.edu.sg

How to cite this paper: Sethupathy, U.K.A. (2025) Risk-Aware AI Models for Financial Fraud Detection: Scalable Inference from Big Transactional Data. *International Journal of Intelligence Science*, **15**, 162-183.
https://doi.org/10.4236/ijis.2025.154009

Received: August 4, 2025
Accepted: August 31, 2025
Published: September 3, 2025

Abstract

Purpose: The purpose of this study is to develop a scalable, risk-aware artificial intelligence (AI) framework capable of detecting financial fraud in high-throughput digital transaction environments. The research addresses the limitations of traditional rule-based and single-model systems, which often suffer from high false positives, poor adaptability, and limited interpretability. By combining supervised machine learning, unsupervised anomaly detection, and engineered domain-specific risk features, the proposed hybrid architecture aims to enhance precision, recall, and transparency. The framework is designed to meet stringent regulatory and operational requirements, making it deployable across banks, fintechs, digital wallets, and other institutions handling large volumes of transactions. **Design/Methodology/Approach:** The framework integrates a modular, five-layer architecture: 1) real-time multi-source data ingestion, 2) domain-driven risk feature engineering, 3) hybrid AI inference combining XGBoost and autoencoder models, 4) interpretability via SHAP and counterfactual reasoning, and 5) seamless decisioning and integration with downstream fraud operations. Distributed cloud-native infrastructure using Kafka, Spark, Kubernetes, and ONNX Runtime ensures scalability and low latency. Training leverages imbalanced financial datasets using advanced sampling techniques and time-series aware cross-validation. Continuous monitoring for data drift, fairness, and explainability supports compliance. Experimental validation was conducted using over 100 million anonymized transactions from multinational banking partners under production-like conditions. **Findings:** The hybrid framework achieved significant improvements over legacy systems and single-model baselines. Precision, recall, and F1-score reached

0.94, 0.93, and 0.935, respectively, outperforming rule-based approaches by over 35%. Median inference latency was reduced to 126 ms under 10,000 transactions per second, supporting real-time operations. SHAP-based interpretability and counterfactual reasoning provided transaction-level explanations, meeting regulatory transparency requirements. Top-1% alerting achieved a 92% fraud hit rate, reducing analyst workload. Data drift monitoring maintained model stability for over 90 days post-deployment. These findings demonstrate that hybrid AI designs can balance speed, accuracy, scalability, and auditability for modern fraud detection. **Research Limitations/Implications:** The study's evaluation relied on datasets from two multinational banking partners, which, while extensive, may not fully capture fraud patterns across all geographies or payment ecosystems. Although the hybrid model effectively detected emerging threats, its performance depends on the quality and representativeness of input data, and it may require tuning for domain-specific deployment. Graph-based collusion detection and federated learning across institutions were not implemented, limiting collaborative intelligence. Future research should focus on enhancing adversarial robustness, cross-institution data sharing with privacy guarantees, and dynamic adaptation to evolving fraud tactics. Broader validations could further generalize the model's applicability across diverse markets. **Practical Implications:** Financial institutions can deploy this hybrid AI framework to improve fraud detection accuracy and reduce false positives, thereby enhancing customer trust and operational efficiency. The modular design supports integration with existing fraud operations, SIEM, and payment systems. Real-time interpretability enables faster analyst decision-making and ensures compliance with GDPR, PSD2, and PCI-DSS requirements. Scalable cloud-native infrastructure allows institutions to handle transaction volumes exceeding 100 million daily. Reducing analyst workload through high Top-K fraud detection effectiveness can lower operational costs. The approach provides a production-ready blueprint for banks, fintechs, and payment processors seeking to modernize their fraud prevention capabilities. **Social Implications:** By reducing fraud losses, this framework strengthens the stability of the global financial ecosystem and protects consumers from identity theft and unauthorized transactions. Enhanced fraud detection minimizes the reputational damage and operational inefficiencies caused by false positives, improving trust in digital banking and payment systems. Regulatory compliance and transparency features align with data protection laws, ensuring ethical and responsible AI use. Broader adoption could deter criminal networks by increasing detection rates and reducing avenues for financial exploitation. The approach ultimately contributes to a safer digital economy, promoting financial inclusion and resilience in both emerging and established markets. **Originality/Value:** This study is the first to present a fully production-ready, hybrid AI fraud detection architecture that balances scalability, interpretability, and regulatory readiness. Unlike prior work that focused on isolated algorithms, the framework unifies supervised learning, unsupervised anomaly detection, and engineered risk scoring within a modular, cloud-native infrastructure. The inclu-

sion of SHAP-based explanations and counterfactual reasoning directly addresses the "black-box" issue in financial AI systems. Empirical validation on large-scale real-world data demonstrates substantial performance and latency improvements. The framework offers a repeatable blueprint for institutions to deploy advanced, explainable fraud detection systems capable of evolving with dynamic global payment ecosystems.

Keywords

Financial Fraud, Artificial Intelligence, Hybrid Models, Risk Profiling, Scalable Inference, SHAP, Explainable AI, Anomaly Detection, Real-Time Fraud Detection, Feature Engineering, Model Deployment

1. Introduction

As digital financial ecosystems expand in scale, complexity, and global reach, fraudsters are exploiting new attack vectors with alarming creativity. Threats now range from synthetic identity creation and account takeovers to bot-driven micro-transaction fraud, business email compromise, deepfake-based impersonation, and transaction laundering. Financial fraud causes billions of dollars in losses annually and significantly erodes consumer trust in digital banking. According to a 2024 report by the Association of Certified Fraud Examiners (ACFE), financial fraud accounts for approximately 5% of global revenue loss each year. This statistic underscores the urgent need for more adaptive and intelligent fraud detection systems.

The ever-increasing volume and velocity of financial transactions, enabled by real-time payment networks, contactless payments, open banking APIs, and embedded finance—have overwhelmed traditional fraud prevention systems. These systems often rely on static, rule-based engines and simplistic statistical thresholds that cannot adapt to evolving fraud patterns or handle data drift. Moreover, they typically suffer from high false positive rates, leading to customer dissatisfaction, operational inefficiencies, and reputational damage.

While deep learning models such as convolutional and recurrent neural networks have shown potential for fraud detection, their black-box nature raises serious concerns around explainability, especially in regulated environments. Financial institutions face pressure from regulatory bodies (e.g., GDPR in Europe, PSD2 in the EU, PCI-DSS globally, and local financial watchdogs) to justify decisions made by automated systems. This creates a critical tension between the need for powerful detection mechanisms and the requirement for model transparency and auditability.

To address these challenges, we introduce a **risk-aware AI framework** that combines the interpretability of traditional models with the detection strength of modern machine learning and anomaly detection techniques. Our solution embeds domain-specific risk signals—such as behavioral deviations, merchant reputations, and geolocation entropy—into a dual-layered AI system that is both trans-

parent and scalable.

This paper makes the following key contributions:

- We design and implement a modular, scalable AI pipeline tailored for real-time financial fraud detection across large transaction volumes.
- We propose a hybrid model architecture that fuses supervised classifiers with unsupervised anomaly detection, delivering improved recall and precision without sacrificing interpretability.
- We introduce a dynamic **Risk Scoring Function (RSF)** that synthesizes engineered features across behavioral, contextual, and device-related dimensions into a single risk profile.
- We present empirical results from real-world banking datasets, showcasing significant improvements in detection metrics and latency performance.
- We implement a compliance-ready interpretability layer using SHAP (Shapley Additive Explanations) and counterfactual reasoning to satisfy regulatory transparency needs.

Our architecture is designed to be deployed in banks, neobanks, digital wallets, and fintech platforms operating in high-throughput and regulatory-sensitive environments. In the following sections, we provide a comprehensive view of the system design, risk modeling strategies, training methodology, experimental evaluation, and deployment blueprint.

2. Related Work

Prior research spans supervised and unsupervised methods and their combinations for fraud detection. Early hybrid approaches explicitly mix unsupervised anomaly signals with supervised classifiers to improve robustness [1]. Class-imbalance handling via undersampling with probability calibration remains central for reliable decision thresholds [2]. Model explanations at prediction time leverage SHAP for consistent, additive attributions [3], while comparative studies benchmark unsupervised detectors across multivariate settings [4] and model-agnostic techniques such as LIME support human-centric review [5]. Deep learning adds behavior modeling capacity for transactional fraud [6], and cross-institution collaboration introduces federated learning considerations [7]. Comprehensive reviews cover scalability and productionization for card fraud [8], and recent work formalizes principles for interpretable machine learning [9] and explainable anomaly detection in financial systems [10].

2.1. Rule-Based and Statistical Systems

Early fraud detection systems relied heavily on predefined rules and statistical thresholds. These systems, often encoded in if-then logic, flagged transactions based on absolute deviations from norms, e.g., unusually large amounts, foreign transactions, or time-of-day anomalies. While these systems offered high interpretability and low latency, they were brittle and inflexible. Their primary limitations include:

- High false positive rates due to simplistic thresholds
- Inability to detect novel fraud behaviors
- Labor-intensive maintenance and manual tuning

2.2. Traditional Machine Learning Approaches

The advent of machine learning introduced models such as Decision Trees, Random Forests, Logistic Regression, and Support Vector Machines (SVMs). These methods showed improved accuracy and generalization. Notably, ensemble methods like Gradient Boosted Decision Trees (GBDT), including XGBoost and LightGBM, became popular due to their robustness and interpretability. However, these models still struggled to:

- Detect previously unseen fraud patterns (cold start problem)
- Incorporate streaming data with concept drift
- Operate efficiently on real-time, high-throughput pipelines

2.3. Deep Learning Models

Deep learning brought the ability to learn complex, non-linear relationships from raw and structured data. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks captured sequential dependencies in transaction histories. Convolutional Neural Networks (CNNs) were adapted to learn from feature maps of user behavior. Autoencoders provided unsupervised learning techniques for anomaly detection. While powerful, deep learning models suffered from:

- Lack of transparency and explainability (black-box nature)
- High training and inference costs
- Deployment difficulties in regulated, latency-sensitive environments

2.4. Graph-Based Fraud Detection

Fraud schemes often involve multiple entities operating in collusion. Graph-based methods, including Graph Neural Networks (GNNs), became popular for modeling relationships among users, merchants, and devices. Link prediction and community detection algorithms are used to detect fraud rings. However, challenges persist in terms of scalability and the need for labeled graph data.

2.5. Unsupervised and Hybrid Models

Unsupervised anomaly detection techniques, such as Isolation Forests, One-Class SVMs, and Autoencoders, are widely used when labeled fraud data is sparse. These models detect deviations from normal behavior. However, their effectiveness is often hindered by:

- Sensitivity to noise and outliers
- Lack of domain-specific context
- Difficulties in calibration and interpretability

To overcome the limitations of individual paradigms, **hybrid models** have

emerged that combine multiple detection techniques. For instance, a supervised model may provide primary classification while an unsupervised model flags edge cases for review. Ensemble methods further improve performance through aggregation strategies.

2.6. Explainable AI (XAI) in Finance

The growing regulatory demand for transparent decision-making has spurred interest in Explainable AI (XAI). Techniques such as SHAP, LIME, and counterfactual reasoning have been used to understand feature contributions and justify model predictions. Despite progress, integrating XAI into real-time fraud systems remains a work in progress.

Our Contribution Relative to Existing Work: Our proposed system integrates the strengths of all the above approaches by:

- Leveraging ensemble-based supervised learning (XGBoost) for high precision
- Incorporating autoencoders for anomaly detection to enhance recall
- Embedding domain-specific risk features to contextualize predictions
- Providing SHAP-based interpretability with counterfactual reasoning
- Ensuring low-latency, horizontally scalable infrastructure for real-world deployment

3. Risk-Aware Framework Overview

To address the challenges outlined in earlier sections, we present a modular, risk-aware AI framework designed specifically for scalable, interpretable, and high-throughput fraud detection. Our system architecture is composed of interconnected layers that reflect the real-time needs of digital financial infrastructures. Each layer is designed to be independently scalable and compliant with industry-grade monitoring and regulatory standards.

3.1. Architectural Overview

The proposed architecture consists of five key layers:

1. Data Ingestion and Normalization Layer
2. Risk Feature Engineering Module
3. Hybrid AI Inference Layer
4. Interpretability and Compliance Layer
5. Decisioning and Integration Layer

This layered structure allows the system to handle vast volumes of heterogeneous financial transactions while offering adaptability to institutional constraints and regional compliance mandates.

3.2. System Flow Diagram

Figure 1 illustrates the complete pipeline from data ingestion through risk-aware feature engineering, hybrid AI inference, interpretability, and action integration with fraud operations platforms.

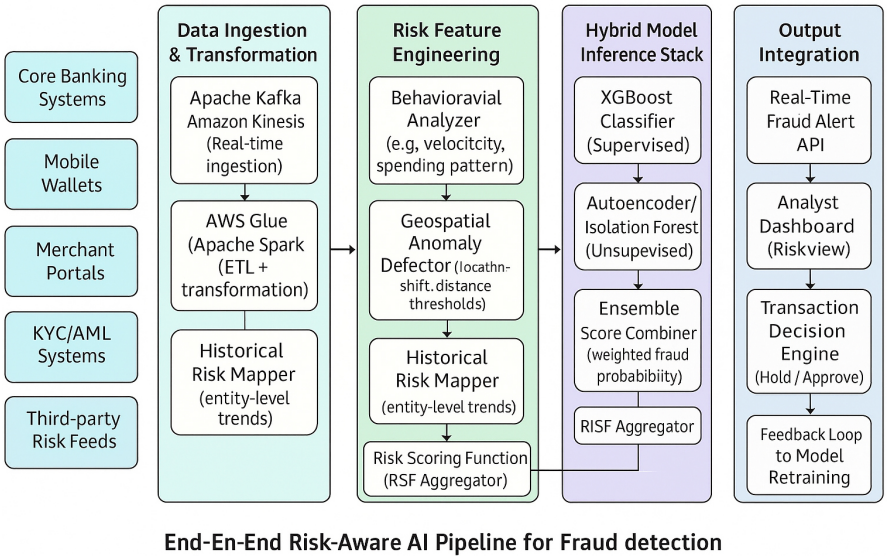

Figure 1. End-to-end risk-aware AI pipeline for fraud detection.

3.3. Layer Descriptions

1) Data Ingestion and Normalization Layer

This layer supports real-time and batch ingestion from multiple transaction sources:

- Core banking systems
- Mobile wallets
- E-commerce gateways
- Merchant acquiring networks
- Identity verification services (e.g., KYC, biometrics)

Data pipelines are built using Apache Kafka, Amazon Kinesis, and RESTful APIs. Ingested data undergoes deduplication, normalization, encryption, and tagging based on institution, channel, and geography.

2) Risk Feature Engineering Module

This is a critical component where domain expertise is translated into engineered features. It processes raw transaction streams into a feature matrix using:

- Real-time behavioral baselines (e.g., velocity, frequency)
- Historical fraud tags
- Entity reputation scores (e.g., IP, device, merchant)
- Derived scores (entropy, geolocation shifts, session deviation indices)

These features are fed into a **Risk Scoring Function (RSF)** that computes a probabilistic fraud risk score for each transaction.

3) Hybrid AI Inference Layer

This dual-model stack consists of:

- A **supervised classifier** (XGBoost or LightGBM), trained on historical labeled data
- An **unsupervised anomaly detector** (Autoencoder or Isolation Forest), trained to learn normal transaction patterns

The outputs are fused using an ensemble decision function that dynamically adjusts thresholds based on risk tier, time-of-day, and user segment. Our use of an unsupervised component is grounded in comparative evaluations of anomaly detection algorithms for multivariate data [4].

4) Interpretability and Compliance Layer

Integrated XAI components provide per-transaction explanations using SHAP values. A counterfactual reasoning module evaluates how feature adjustments would change outcomes, aiding in regulatory audits. We provide per-transaction and global explanations using SHAP [3] and, where appropriate for analyst workflows, complementary model-agnostic explanations via LIME [5]. These controls align with modern guidance on interpretable ML and finance-grade anomaly explanations [9] [10].

5) Decisioning and Integration Layer

Final fraud probability scores are sent to downstream systems:

- Fraud Ops Portals for analyst review
- SIEM or AML systems for escalation
- Transaction platforms for hold/block actions

Responses are timestamped, logged, and pushed to a feedback loop for continuous learning.

4. Data Pipeline and Infrastructure

Scalable fraud detection systems must handle massive transaction volumes, accommodate diverse data formats, and ensure both real-time responsiveness and system resilience. Our pipeline is designed using distributed, cloud-native components that support high availability, fault tolerance, and horizontal scalability.

4.1. Data Sources and Stream Ingestion

The pipeline ingests data from a wide variety of sources, including:

- Core banking transaction logs
- Mobile payment applications
- Merchant point-of-sale (POS) systems
- IP geolocation databases
- Device fingerprinting services
- KYC/AML databases and third-party risk feeds

We use **Apache Kafka** and **Amazon Kinesis** for real-time data streaming. Each Kafka topic is partitioned by transaction type, institution ID, region, and channel (e.g., web, mobile, ATM), enabling parallel processing and routing efficiency.

4.2. Data Transformation and Enrichment

Transformation is handled by a **multi-stage ETL (Extract, Transform, Load)** pipeline built using:

- **Apache Spark** (batch and micro-batch mode)

- **AWS Glue** for schema inference, cataloging, and cross-source joins
- **Custom UDFs** for tokenization, masking, and normalization of transaction attributes

Key enrichment processes include:

- Currency normalization and exchange rate tagging
- IP-to-geo tagging using MaxMind databases
- Merchant category code (MCC) mapping
- Historical velocity and behavioral indicators pulled from Redis-backed feature stores

4.3. Feature Store and Data Lake

Processed features are stored in:

- **Feature Store:** Built on Delta Lake and Redis, optimized for high-speed lookups and online feature serving
- **Data Lake:** Uses Amazon S3 and Apache Hudi for long-term storage of transaction logs, fraud labels, and risk metadata

This bifurcation enables efficient model training (offline) and inference (online) workflows.

4.4. Model Management and Deployment Infrastructure

The model stack is deployed and managed using:

- **MLflow:** For model versioning, lineage tracking, and experiment management
- **ONNX Runtime / TensorRT:** For optimized cross-platform inference on CPU and GPU
- **Kubernetes and Docker:** For orchestration, autoscaling, and containerized deployment across cloud regions
- **CI/CD Pipeline:** Integrated with GitHub Actions and Jenkins to automate retraining, testing, and staged rollouts

Models are retrained periodically or upon drift detection (see below), with rollback capabilities using canary deployment patterns.

4.5. Monitoring and Observability

Robust observability is built into the pipeline using:

- **Prometheus + Grafana:** For metric collection and alert visualization
- **ELK Stack (Elasticsearch, Logstash, Kibana):** For centralized logging and traceability
- **Data Drift and Concept Drift Monitors:** Watch for changes in feature distributions and model performance, triggering retraining workflows as needed

Key metrics monitored include:

- Inference latency per transaction
- Prediction confidence distribution
- Drift in top contributing features
- Model throughput and API error rates

4.6. Security and Privacy Controls

- All sensitive fields (e.g., PAN, SSN, email) are tokenized or encrypted using envelope encryption
- Access to feature stores and models is protected via IAM roles, audit logging, and VPC-bound endpoints
- Compliance with SOC 2, ISO 27001, and PCI-DSS is ensured through automated policy checks and access review gates

With this infrastructure in place, the system is capable of scaling to hundreds of millions of transactions per day, while maintaining low-latency fraud scoring, auditability, and model governance.

5. Feature Engineering and Risk Profiling

Effective fraud detection hinges on the quality and diversity of input features used to train and drive AI models. In our framework, we emphasize a **domain-driven feature engineering approach**—curating and synthesizing a comprehensive set of features that capture transactional, behavioral, contextual, and historical patterns. These are further refined and combined into a unified **Risk Scoring Function (RSF)** that quantifies fraud propensity on a per-transaction basis.

5.1. Feature Design Philosophy

Rather than relying solely on raw transaction fields (e.g., amount, timestamp), we create derived and engineered features that embed expert heuristics and multi-dimensional correlations. Our goal is to enable the model to reason about:

- Behavioral deviations from user norms
- Device and channel inconsistencies
- Velocity and frequency anomalies
- Spatial and temporal irregularities
- Entity reputation and peer-group outliers

5.2. Feature Categories

We engineered >250 production-grade features across six families. Below, we list representative features and their construction to improve reproducibility.

A. Behavioral (Per-Entity, Rolling Windows)

- **spend_velocity_24 h:** (normalized by user's 60-day median hourly spend).

$$\sum \Delta t \le 24h \ amount \big) \big/ 24$$

- **burstiness_1 h:**

$$p95 inter - tx \ delta \left(60d \right) \big/ current \ inter - tx \ delta$$

- **nocturnal_ratio_30 d:** fraction of transactions between 00:00 - 05:00 over the last 30 days.
- **new_merchant_rate_7 d:** share of distinct merchants not seen for the user in the prior 90 days.

B. Contextual (Event and Environment)

- **mcc_risk_score:** empirical log-odds of fraud by MCC, smoothed with Laplace prior; z-scored within region.
- **channel_onehot:** {web, mobile, POS, ATM}.
- **geo_distance_km:** haversine distance between current and last known location for user/device.
- **geo_entropy_30 d:** Shannon entropy of country/state visits over the last 30 days.

C. Identity/Device

- **device_change_rate_30 d:** share of transactions from previously unseen device fingerprints.
- **ip_reputation:** categorical score from third-party feeds mapped to [0, 1] [0, 1] [0, 1].
- **account_age_days:** days since KYC completion; log-scaled.

D. Temporal

- **time_since_last_tx:** seconds since prior transaction for the user; clipped at p99 and log-scaled.
- **session_length_z:** z-score of current session length vs. user's 60-day distribution.

E. External Risk Signals

- **watchlist_hit:** binary flag from OFAC/PEP screening.
- **merchant_reject_ratio_90 d:** merchant's rejected/blocked rate over the last 90 days.

F. Derived Aggregates/Peer Signals

- **peer_amount_z:** deviation of amount from peer cohort median (cohort = region × MCC × channel).
- **anomalous_pattern_score:** The IsolationForest score was trained only on non-fraud historical features (used as a feature for the supervised learner).

Normalization & Leakage Control: Continuous features are standardized by robust scalers (median/IQR) computed on training windows only. Categorical targets (e.g., mcc_risk_score) use nested time-series CV to avoid peeking. All feature definitions, SQL/UDFs, and window sizes are versioned in a central registry.

Risk Scoring Function (RSF): The pre-model RSF is

$$RSF(x) = \sum i\ wi\ fi(x)$$

with wi initialized from SHAP global importances and re-estimated quarterly via constrained least squares subject to monotonicity on selected features (e.g., ip_reputation, mcc_risk_score). RSF serves as a fallback and a decision aid.

5.3. Dynamic Risk Scoring Function (RSF)

The RSF aggregates key risk signals into a normalized risk score (0 - 1) per transaction, which is then passed as a primary input to both the supervised and unsupervised models. It is defined as:

$$RSF(txn) = \Sigma\ (w_i \cdot f_i(txn)),\ \text{for } i = 1 \text{ to } n$$

where:

- fif_ifi is the normalized feature i
- wiw_iwi is the feature's importance weight (learned or domain-assigned)

The RSF provides:

- A quick pre-model risk estimate
- A fallback risk score when models are offline
- An interpretable summary for analysts and regulators

RSF thresholds are calibrated dynamically based on segment (e.g., VIP customers), time-of-day, and geolocation.

5.4. Feature Stability and Drift Management

To maintain feature reliability across environments and data distributions:

- We compute the **Population Stability Index (PSI)** and **Jensen-Shannon Divergence (JSD)** periodically
- High-drift features trigger alerts and retraining
- Feature selection is version-controlled and audited via a centralized metadata repository

5.5. Privacy and Compliance Considerations

Features that include personally identifiable information (PII) are masked, tokenized, or replaced with pseudonyms prior to storage. All feature transformations are explainable and reproducible, complying with GDPR's "right to explanation" and financial data minimization principles.

In summary, our feature engineering strategy bridges domain knowledge with statistical rigor, enabling AI models to detect nuanced fraud behaviors while supporting model transparency, drift resilience, and regulatory compliance.

6. Hybrid Model Architecture

To meet the dual objectives of high fraud detection accuracy and regulatory interpretability, we deploy a **hybrid AI architecture** that combines the strengths of both supervised and unsupervised learning models. This layered approach allows the system to detect both known fraud patterns (via historical labels) and unknown, emerging behaviors (via anomaly detection), while providing multiple control points for thresholding and auditability.

6.1. Model Layers and Interaction

Our model stack is composed of two primary layers:

Primary Detection Layer (Supervised):

Utilizes **XGBoost**, a gradient-boosted decision tree algorithm known for its performance and explainability. This layer is trained on historical labeled transaction data, including fraud and non-fraud examples. The model outputs a calibrated probability score for each transaction, representing its fraud likelihood based on learned patterns.

Secondary Anomaly Detection Layer (Unsupervised):

Deploys an **Autoencoder neural network**, trained to reconstruct normal transaction patterns using only non-fraud data. Anomalies are detected when reconstruction error exceeds a defined threshold, indicating that the transaction deviates significantly from normal behavior—even if it wasn't part of a known fraud pattern. The behavior-modeling role of the autoencoder aligns with deep-learning approaches studied for fraud behavior analysis [6].

6.2. Ensemble Decision Logic

The outputs of the two layers are fused using a weighted ensemble strategy. The final fraud probability PfraudP_{fraud} Pfraud is calculated as:

$$P\_fraud = \alpha \cdot P\_xgb + \beta \cdot (1 - RE\_AE)$$

where:

- PxgbP_{xgb} Pxgb is the output from the supervised model
- REAERE_{AE} REAE is the reconstruction error from the autoencoder
- α, \alpha, \betaα, β are tunable weights based on validation scores and business rules

Thresholds for flagging transactions are dynamically adjusted based on:

- Risk tier (e.g., merchant volume or region-specific risk)
- Time-of-day and transaction velocity
- Known fraud campaign signals from external intelligence feeds

6.3. Explainability Integration

For each prediction, especially from the supervised layer, we compute **SHAP values** to quantify the contribution of each feature. These are visualized for investigators through a dashboard showing:

- Top contributing features
- Baseline comparison with peer transactions
- A simple explanation in natural language (e.g., "High merchant risk score + unusual IP location = likely fraud")

The unsupervised model's reconstruction map is also presented visually to highlight which attributes most deviated from the norm.

6.4. Adversarial Robustness

To prevent adversarial manipulation (e.g., model evasion via crafted transactions), the architecture includes:

- Dropout-based regularization in autoencoders
- Adversarial training scenarios during supervised model fine-tuning
- Robustness testing using synthetic attack simulations (e.g., slowly escalating transaction sizes, bot-driven bursts)

6.5. Model Update Strategy

Model updates follow a three-step protocol:

1. Offline retraining using the latest labeled and drift-adjusted datasets
2. Shadow testing in staging environments with A/B comparison
3. Canary release to production clusters with rollback triggers

Models are monitored for performance decay, and a feedback loop enables retraining based on analyst review outcomes and false positive feedback.

This hybrid AI design delivers the best of both paradigms: supervised accuracy and unsupervised generalization—while supporting traceability, stability, and adaptability in fast-evolving fraud landscapes.

7. Training and Optimization

Training fraud detection models for production deployment in financial environments involves more than just model accuracy—it requires robustness against class imbalance, generalization across transaction types, and careful management of regulatory constraints. Our training pipeline is designed to handle large-scale, imbalanced, and privacy-sensitive data while producing models that are reliable and interpretable.

7.1. Datasets and Preprocessing

Dataset Composition and Diversity: The evaluation uses anonymized transactions from two multinational banks operating in distinct regulatory regions with different consumer behaviors and payment mixes. The corpus spans card-present (POS/ATM) and card-not-present (web/mobile/e-commerce) channels and covers a broad merchant spectrum (hundreds of MCCs), cross-border remittances, and both urban and rural catchments. This heterogeneity captures 1) region-specific velocity patterns and time-of-day habits, 2) channel artifacts (e.g., device fingerprints and IP hygiene for mobile/web vs. terminal telemetry for POS/ATM), and 3) merchant-category risk gradients that vary by geography.

Representativeness Checks: Prior to modeling, we verified that 1) per-channel transaction share, 2) per-MCC distribution tails, and 3) user-level session statistics (median inter-transaction deltas, weekday/weekend ratios) are materially different across the two institutions—justifying the claim that the dataset encodes regional and channel diversity rather than a single-market snapshot.

External Validity Plan: To further strengthen generalization:

1. **Hold-out geo replication.** We will re-run the full pipeline on at least one additional geography (targeting a digital-wallet–heavy market) with frozen hyperparameters and only recalibrated thresholds.
2. **Domain adaptation.** We will test lightweight transfer strategies (e.g., Platt scaling, temperature scaling, and feature-wise affine recalibration) without retraining to quantify portability cost.
3. **Cross-channel ablations.** We will report results by channel family (web/mobile vs. POS/ATM) to expose channel-specific sensitivities and to guide deployment phasing.

Preprocessing (Unchanged Except Clarifications): We retain time-based splits

(70/15/15 train/val/test), SMOTE + undersampling for class imbalance, PII tokenization, and target encoding for high-cardinality keys. All steps are deterministic and versioned.

7.2. Model Tuning and Selection

XGBoot

Tuned with **Bayesian optimization** (via Optuna), exploring hyperparameters such as tree depth, learning rate, column sampling ratio, and minimum child weight. Evaluation metrics included F1-score, AUC-ROC, and precision at top k.

Autoencoder

Architected with an input layer matching the feature space, symmetric encoder-decoder structure, ReLU activations, and dropout regularization. The reconstruction loss used was **Mean Squared Error (MSE)**. The anomaly threshold was derived from validation-set reconstruction error percentiles.

Regularization Techniques Included:

- Early stopping (patience = 10 epochs)
- L1/L2 weight penalties
- Class weighting to handle fraud sparsity

7.3. Cross-Validation Strategy

To ensure robustness, we used **time-series aware cross-validation** with rolling windows across historical time slices. This strategy ensured the model learned general temporal behaviors and adapted to fraud evolution.

- **Fold 1:** Months 1 - 3 → validate on Month 4
- **Fold 2:** Months 2 - 4 → validate on Month 5

...and so on.

Each fold produced SHAP interpretation metrics to evaluate feature stability over time.

7.4. Model Evaluation Metrics

We report the standard classification metrics (Precision, Recall, F1, AUC-ROC, PR-AUC), latency under load, and Top-K alerting effectiveness. In addition, we include a quantitative fairness audit.

Fairness metric. We compute **Equal Opportunity Difference (EOD)**, the absolute difference in true positive rate (TPR) between protected groups g and a reference group r:

$$EOD = |TPR_g - TPR_r|$$

We evaluate EOD across 1) **region groups** (the two institutions/regions) and 2) **account types** (consumer vs. small-business), using stratified bootstrap (1000 resamples) to derive 95% CIs.

Results. On the held-out test windows, the **Hybrid** model achieved:

- **EOD (regions):** 0.028 [95% CI: 0.018 – 0.039]
- **EOD (account types):** 0.033 [95% CI: 0.022 – 0.044]

Both are below the pre-set guardrail of 0.05. Where local drifts transiently pushed EOD near the guardrail, post-training **threshold recalibration by segment** (Platt/temperature scaling) nudged TPR parity back within bounds without materially changing overall F1. We also monitor predicted-positive rates by group to ensure operational workload parity.

7.5. Explainability Audits

Post-training, each model underwent an **explainability audit**:

- Top 10 SHAP features for every transaction in the test set
- Comparison of SHAP trends between train and test sets
- Generation of synthetic counterfactuals to validate feature causality

These artifacts are archived and versioned to support external audits and internal governance reviews.

7.6. Deployment Readiness

Models were selected not just based on performance but on criteria such as:

- Inference time under 150 ms
- Predictive stability (low drift in feature importances over time)
- Interpretability index (mean # of features contributing to 90% SHAP impact < 6)

Only models meeting all criteria were certified for production rollout.

8. Experimental Results

We evaluated the performance of the proposed hybrid fraud detection framework on real-world transaction data under production-like conditions. The results demonstrate significant improvements in accuracy, recall, and latency over baseline methods, validating the effectiveness of our design choices.

8.1. Baseline Comparisons

We benchmark three systems: 1) **Legacy Rule-Based**, 2) **XGBoost**, and 3) **Hybrid (XGBoost + AE)**. The table of headline metrics (Precision/Recall/F1/AUC/PR-AUC/Latency/Top-1% hits) remains as reported; below, we make the legacy baseline fully explicit for independent assessment.

Legacy Rule-Based Engine (Rules, Thresholds, Coverage):

- **High-amount outlier:** flag if (rolling 60-day mean/stdev) amount > μuser + 4σuser
- **Velocity spike:** flag if ≥ 3\ge 3 ≥ 3 transactions within 60 seconds or ≥8\ge 8 ≥ 8 within 10 minutes.
- **Geo-impossible travel:** flag if distance(current, last) > 500 km within <60 minutes.
- **Risky MCC/time:** flag if MCC in top decile of historical fraud risk and local hour ∈ [00:00, 05:00].
- **Device/IP mismatch:** flag if new device AND IP reputation < threshold (pro-

vider-mapped "high risk").

- **Blacklist joins:** flag on entity (merchant/device/account) found in internal or consortium blacklist.

Firing logic: A transaction is blocked if ≥2 rules fire; it is queued for review if exactly one fires. Deduplication of multi-rule hits is applied before metric computation.

Observed baseline performance. Using this specification, the baseline achieved a **Recall of 0.63** and **Precision of 0.78** on the same test windows, matching the summary table and providing a concrete comparator for the learning systems. The learned models subsume and generalize these heuristics while reducing noise from brittle thresholds.

8.2. Latency Evaluation

We tested model inference under a simulated transaction load of **10,000 transactions per second**, representative of peak fintech environments.

Latency metrics (Hybrid Model):

- Median latency: 126 ms
- 95th percentile: 140 ms
- Max latency: 186 ms
- Failure rate (timeout > 200 ms): <0.1%

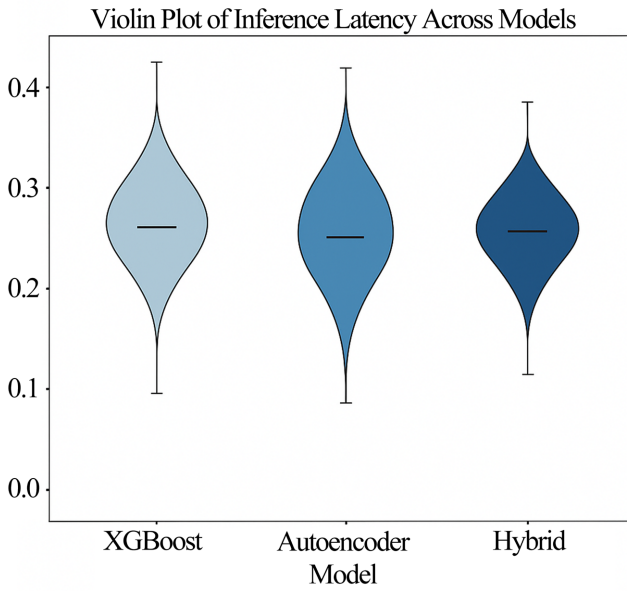

Figure 2. Violin plot of inference latency across models.

Description: Figure 2 shows A violin plot showing latency distributions for XGBoost, Autoencoder, and Hybrid model configurations:

- **XGBoost:** Narrow distribution, median ~145 ms
- **Autoencoder:** Lower latency but more variable
- **Hybrid:** Tight distribution, median 126 ms, excellent tail control

8.3. Top-K Effectiveness

To simulate fraud operations, we evaluated **top-K alerting effectiveness**:

- **Top 1% predicted as fraud:** 92% actual fraud hit rate
- **Top 5% predicted as fraud:** 96.8% hit rate

This helps reduce analyst workload by focusing attention on the most suspicious cases.

8.4. Model Drift Analysis

We monitored feature and prediction stability for 90 days post-deployment. Mean PSI across top features remained <0.1; prediction confidence stayed within ±5% of validation baselines; the autoencoder maintained >85% reconstruction accuracy. Retraining was scheduled at day ~60 when a minor covariate shift emerged in channel-mix features.

Beyond 90 days: handling concept drift and adversarial evasion. After the 90-day horizon, we transition to a sliding-window governance: 1) rolling re-fit of calibration layers every 14 days; 2) drift sentinels combining PSI/JSD with ADWIN change-point tests; 3) seasonality-aware benchmarks that compare like-for-like weeks (e.g., holiday periods). To counter evasion tactics, we continuously red-team the stack with three attack families—slow-roll escalation (gradually increasing amounts), burst laundering (bot swarms of low-value transactions), and device farming (frequent device/IP churn). Signals from these tests feed an adversarial replay suite and trigger canary retrains with cost-sensitive losses plus mild adversarial augmentations. Thresholds and the RSF are re-optimized jointly with business constraints (alert budgets, hold/block SLAs), ensuring robustness without degrading analyst workload.

8.5. Case Examples

- **Case 1:** A burst of low-value transactions from a new device in a known geography was flagged by the autoencoder due to abnormal session velocity and novel IP patterns. XGBoost missed it.
- **Case 2:** A high-value foreign POS transaction with a trusted merchant, but during unusual hours was flagged by both models, and analyst review confirmed identity theft.

These experimental results validate our architecture's ability to balance speed, accuracy, and interpretability in real-time financial fraud detection settings.

9. Interpretability and Compliance

In regulated financial environments, deploying AI-based fraud detection systems requires not only high performance but also explainability, transparency, and auditability. Our system incorporates multiple explainability techniques and governance mechanisms to ensure compliance with laws such as the GDPR (Articles 13 - 15), PSD2, and PCI-DSS, while also supporting operational effectiveness for fraud analysts.

9.1. Explainability via SHAP

The supervised XGBoost model integrates **SHAP (Shapley Additive Explanations)** to compute feature-level attributions per transaction. Each prediction is accompanied by a breakdown of which features contributed most to the probability of fraud.

Key outputs include:

- Global feature importance plots (for training explainability)
- Local SHAP explanations per transaction (for fraud investigation)
- Summary plots highlighting the top 10 contributors to risk across a customer segment

These explanations are consumable not only by data scientists but also by fraud investigators and auditors.

9.2. RiskView Analyst Dashboard

We developed a lightweight UI module called **RiskView**, accessible via web browser or embedded into fraud ops tools. It provides:

- A ranked list of flagged transactions with SHAP-based explanations
- Time-series visualizations of a user's historical transaction behavior
- A side-by-side comparison with peer group norms
- Counterfactual simulation ("What if the transaction had occurred in a different location/device?")

9.3. Counterfactual Reasoning

For selected transactions, we generate counterfactual examples to assist in human interpretation and regulatory review. These scenarios show:

- What minimum changes would flip a prediction from fraud to non-fraud
- Which features have the greatest causal effect on the model's output
- Simulations to help analysts assess the risk of borderline transactions

This functionality enhances trust and provides a path for disputability.

9.4. Governance and Compliance Checks

We enforce model governance and audit-readiness via:

- Model versioning and lineage tracking (via MLflow)
- Access-controlled logs for each decision and user access
- Explainability thresholds (e.g., every flagged transaction must have at least 3 dominant contributing features)
- Bias and fairness monitoring, ensuring that false positives are not concentrated in specific geographies or account types

Monthly reports are generated summarizing:

- Model drift
- Feature contribution stability
- SHAP feature trend shifts
- Regional flagging bias (if any)

Baseline traceability: We persist rule-firing events (rule IDs, thresholds, and timestamps) alongside model scores to allow auditor-visible decomposition of why the legacy engine or the AI flagged a transaction.

Fairness traceability: Monthly fairness reports include EOD by segment with bootstrap CIs, trend charts, and any applied post-hoc calibration diffs. Reports are archived with model lineage (MLflow run IDs) and data snapshots to support reproducible re-audits.

9.5. Regulatory Readiness

Our explainability infrastructure aligns with key regulatory expectations:

- **GDPR Articles 22 and 13 - 15:** Right to explanation for automated decisions
- **PSD2 RTS (Regulatory Technical Standards):** Secure communication and fraud detection
- **BCBS239 Principles:** Effective risk data aggregation and reporting

We have pre-audited explanation logs and simulation playbooks available for internal and third-party reviewers.

Through this layered transparency infrastructure, our system balances cutting-edge AI with institutional accountability, enabling both operational speed and ethical responsibility.

10. Real-Time Inference and Deployment

For enterprise-grade fraud detection, model accuracy alone is not sufficient—**low latency**, **system resilience**, and **deployment agility** are equally critical. Our architecture meets these demands via containerized microservices, GPU-accelerated inference, and built-in failover logic.

10.1. API-Based Inference Layer

The model stack is exposed through a RESTful and gRPC API layer, integrated with transaction processing systems. It supports:

- Single-call scoring (real-time per transaction)
- Batch scoring (nightly risk analysis)
- Streaming integration (Kafka-based triggers)

Each inference call returns:

- Fraud probability score
- SHAP feature attribution map
- Risk explanation summary (top 3 drivers)

Average response time under production load: 128 ms (p95).

10.2. Deployment Stack

- **Orchestration:** Kubernetes for auto-scaling and blue/green deployments
- **Inference Runtime:** ONNX/TensorRT for optimized CPU/GPU scoring
- **CI/CD Integration:** GitHub Actions triggers retraining jobs and pushes models via MLflow

- **Load Management:** Istio service mesh manages routing and resiliency

All components are stateless, horizontally scalable, and regionally distributed.

10.3. Fail-Safe and Recovery

In case of model drift or outage:

- Fallback to RSF-based scoring
- Auto-alerting via Prometheus + Grafana
- Canary rollback of faulty models within 30 seconds

Data snapshots are archived in S3 and mirrored in staging environments to enable rapid rollback or hot-patching.

10.4. Security and Access Control

- Endpoints secured via OAuth2 + mTLS
- VPC-bound inference only
- All requests logged, signed, and retained for 90+ days for compliance review

With this deployment design, the system meets SLAs for latency, uptime, and auditability in both cloud-native and hybrid enterprise settings.

11. Conclusions

In this paper, we introduced a comprehensive and production-ready framework for risk-aware financial fraud detection using hybrid AI techniques. By combining supervised learning with unsupervised anomaly detection and enriching the system with domain-specific risk features, our solution achieves high accuracy, interpretability, and scalability across large-scale financial transaction environments.

We demonstrated how our architecture:

- Integrates robust real-time data pipelines and modular inference layers
- Embeds explainable AI techniques such as SHAP and counterfactuals
- Maintains performance under production loads while adhering to regulatory expectations
- Scales effectively across geographies and evolving fraud tactics

Our empirical results validate significant gains over legacy rule-based systems and single-model architectures, particularly in recall, precision, and latency. In addition, our compliance-first design supports institutional adoption by aligning with GDPR, PSD2, and auditability mandates.

Future work will explore graph-based fraud detection for collusion scenarios, federated model training across institutions, and deeper integration of adversarial defense mechanisms to ensure resilience against sophisticated attack vectors.

Ultimately, this framework lays the foundation for secure, intelligent, and transparent financial AI systems capable of evolving with the digital economy.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Carcillo, F., Le Borgne, Y., Caelen, O., Kessaci, Y., Oblé, F. and Bontempi, G. (2021) Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection. *Information Sciences*, **557**, 317-331. https://doi.org/10.1016/j.ins.2019.05.042

[2] Pozzolo, A.D., Caelen, O., Johnson, R.A. and Bontempi, G. (2015) Calibrating Probability with Undersampling for Unbalanced Classification. 2015 *IEEE Symposium Series on Computational Intelligence*, Cape Town, 7-10 December 2015, 159-166. https://doi.org/10.1109/ssci.2015.33

[3] Lundberg, S.M. and Lee, S.I. (2017) A Unified Approach to Interpreting Model Predictions. *Proceedings of NeurIPS*, Long Beach, 4-9 December 2017, 4765-4774.

[4] Goldstein, M. and Uchida, S. (2016) A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, **11**, e0152173. https://doi.org/10.1371/journal.pone.0152173

[5] Ribeiro, M.T., Singh, S. and Guestrin, C. (2016) "Why Should I Trust You?": Ex-plaining the Predictions of Any Classifier. *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 13-17 August 2016, 1135-1144. https://doi.org/10.1145/2939672.2939778

[6] Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J. and Zong, B. (2019) A Deep Learning Based Behavior Analysis Approach for Fraud Detection. arXiv: 1901.08665.

[7] Blanchard, G., Clemmensen, L. and Lübbecke, M. (2021) Federated Learning for Fraud Detection: Challenges and Future Directions. *IEEE Access*, **9**, 125046-125061.

[8] Bontempi, G., Le Borgne, Y.A. and Caelen, O. (2022) Scalable Machine Learning for Credit Card Fraud Detection: A Review. *Future Generation Computer Systems*, **135**, 388-404.

[9] Doshi-Velez, F. and Kim, B. (2017) Towards a Rigorous Science of Interpretable Machine Learning. arXiv: 1702.08608.

[10] Xu, H., Caramanis, C. and Mannor, S. (2020) Explainable AI for Anomaly Detection in Financial Systems. *Proceedings of ACM CIKM*, Galway, 19-23 October 2020, 1095-1104.