International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I3P102

Volume 74 Issue 3, 7-13, March 2026

Original Article

A Theoretical Framework for Hybrid Rule-Based and Machine-Learning Fraud Detection

Sumit Asthana

Independent Applied AI Researcher, PA, USA.

Corresponding Author : sumitasthana14@gmail.com

Received: 21 January 2026 Revised: 26 February 2026 Accepted: 11 March 2026 Published: 28 March 2026

Abstract *- This paper introduces a practical approach to modern fraud detection by combining rule-based systems with machine learning models. Traditional systems often work in isolation, rely heavily on static rules, and struggle to identify emerging or complex fraud patterns. To address these challenges, we propose a hybrid framework that blends the clear logic of rules with the adaptive strengths of machine learning. The solution includes a two-layer model: a first layer using supervised techniques to predict known fraud types, and a second layer using unsupervised methods to validate for errors, both missed frauds and false alerts. A shared feature store ensures consistent data usage across training and real-time prediction, while explainability tools help analysts understand model decisions.*

Keywords *- Fraud Detection, Hybrid Models, Machine Learning, Rule-Based Systems, Anomaly Detection, Explainable AI.*

1. Introduction

Fraud in the financial world is moving at a breakneck pace, making it harder than ever for banks and other organizations to stay ahead. For example, in the early months of 2020, one financial partner saw the number of fraud cases in their digital channels skyrocket overnight as the pandemic changed the way things normally worked. Their current rule-based defenses fell apart, sending out a lot of useless alerts and not much real protection. This experience shows a long-standing problem: most detection platforms are separate and only watch one payment channel or product at a time. Because of this, risky behavior that affects more than one account or service often goes unnoticed, putting institutions at risk.

Scammers are quick to take advantage of any weakness in today's world, and the tools we use often do not keep up with real-world threats. Rule engines are clear and predictable, but they are too rigid on their own. Machine learning models can change quickly, but they can also feel like black boxes. It is not common for institutions to use both methods in a way that makes each one stronger.

This paper presents a path forward—a hybrid detection framework that integrates the precision of human-generated rules with the adaptability of machine learning. We present a multi-layered plan: a supervised model identifies known patterns, while an unsupervised model searches for new and unexpected threats. Using this method, businesses can both increase their detection rates and cut down on those annoying false positives that damage trust. In short, we are suggesting a plan for making fraud prevention smarter and stronger so that it can handle whatever new twists financial crime throws at it.

2. The Strategic Imperative for a Hybrid Fraud Detection Ecosystem

In today's financial world, criminals are getting smarter and faster, but many businesses are still using old defences. Rule-based systems that do not change cannot keep up. To stay ahead, fraud prevention needs to change from using separate tools to a more connected, flexible system that uses machine learning to see what will happen next and rules to make things clear.

This paper delineates the deficiencies of conventional methodologies, elucidates the significance of a hybrid model, and presents the case for a cohesive fraud management strategy.

2.1. The Limitations of Legacy and Siloed Systems

Today, most banks and other financial institutions use a mix of fraud prevention systems that have been built up over many years. These systems do not often talk to each other. They came out one by one as banks dealt with new threats, bought other companies, or just added new ways to pay. It is not uncommon to see teams in a fraud prevention department working on six, seven, or even more separate platforms at the same time. One system checks credit card fraud, another checks debit transactions, a third checks digital payments, and so on.

This fragmentation of architecture leads to big problems with operations and a lack of expertise. It can take team members a long time, sometimes up to six months, to learn how to use each platform well enough to do their jobs. When some tools are too unstable for consistent operational deployment, system reliability problems make these challenges even worse. As a result, analysts are always in reactive workflows, switching between different interfaces and manually using backup systems when their main tools fail.

The siloed structure makes things even worse by not giving everyone a clear view of customer risk. Transaction data that is spread out across systems for each channel makes it hard to connect customer activities in a meaningful way, which means that complex fraud patterns go unnoticed. Think about an account takeover that lets bad transactions happen across many payment channels. When the necessary data is stored in separate technological and organizational compartments, these kinds of complicated situations stay hidden. This problem gets worse when institutions rely on proprietary scoring algorithms from vendors that are hard to understand, cannot be changed, limit the ability to respond quickly, and make it hard to do meaningful audits.

Most detection systems are based on strict rules, which makes them especially bad at catching fraud that changes over time. As criminal tactics get better, static rules quickly become useless. This means that manual updates are always behind new threats, which costs a lot of money to maintain. This combination of high operational costs and a lack of ability to adapt to new fraud patterns makes businesses more vulnerable, and traditional architecture cannot handle it well enough.

2.2. The Combination of Rules and Machine Learning: A Hybrid Paradigm

Banks have learned the hard way that neither pure rules nor pure machine learning work. What works? Using both at the same time. Every method has its strengths and weaknesses, but when you combine them, they always beat either one on its own. Fraud teams see this happen every day, so it is not just a theory.

Rules-based systems are still the most important tools for stopping fraud. Why? Because there are times when you need to be completely sure. Think about these situations:

- The clear cases: Are there transactions from North Korea? Blocked. Someone trying to buy twenty things in thirty seconds? Marked. These patterns are so clear that a complicated analysis would be too much.
- Rules for compliance: Regulators will not accept "our AI thought it was probably okay" as an excuse. There are strict limits on how much money can be sent, some countries require mandatory holds, and auditors need to be able to check that you know your customer.
- Efficiency in operations: You do not need to do a lot of research on your customer's daily Starbucks purchase. Rules instantly approve routine transactions, which frees up processing power for things that are truly suspicious.

Machine Learning Models work on a completely different problem. Fraud changes too quickly for rules written by people to keep up. This is where ML makes its money:

- The shape-shifters: Fraudsters today are always pushing the limits. They will change the number of transactions, the timing of them, and the devices they use. ML picks up on these small changes that strict rules do not see.
- Too much information: Each transaction creates hundreds of data points, such as the location, device ID, purchase history, typing patterns, and network characteristics. It is impossible for a person to write rules that cover every possible interaction. ML naturally understands this complexity.
- Different shades of gray: It is not always easy to spot real fraud. ML gives probability scores that tell the difference between "definitely suspicious" (send to the investigation team) and "slightly unusual" (maybe just ask for more proof).

Most successful implementations are like triage in an emergency room. Rules take care of the easy cases first, like letting Grandma go grocery shopping once a week or stopping a purchase with a stolen card number. Everything else flows to ML models that examine the murky middle ground where fraud often hides. It does not look nice, but it works.

2.3. From Fragmentation to a Unified Strategy

A feature store is an important part of any fraud detection system that can grow because it connects raw data to useful information. At its core, it is a structured storage space where features like transaction speed, device fingerprints, and beneficiary network centrality are defined once and then used over and over again for both model training and real-time scoring. Without a system like this, teams often rebuild the same transformations in separate areas, which can cause training-serving skew, where models act differently in production than they did in development. This reliability is non-negotiable in fraud detection, where even small differences can lead to missed cases or a lot of false alerts. A well-designed feature store also makes sure that access is quick, so that transactions can be compared to historical behavior in real time on a large scale. This is something that ad-hoc pipelines do not often do. In addition to being technically efficient, it adds discipline and governance by versioning, documenting, and checking the quality of each feature. This stops people from doing the same thing twice and lowers the risk of problems in the business. The feature store becomes the system's single source of truth (SSoT) by bringing together fragmented data pipelines into a single managed layer. This gives data scientists, engineers, and fraud analysts a consistent view of customer and transaction

behavior. This coming together not only makes it easier to find things, but it also makes people in the organization more confident in the models.

3. Proposed Solution Architecture: A Multi-Layered Modelling Approach

The proposed architecture is a complex, multi-layered modeling engine that was made to solve the problem of low event rates in financial fraud detection, where fake transactions are very rare compared to real ones. This makes datasets that are very unbalanced, which can make traditional models less accurate. The system goes beyond just one model and uses a hybrid, sequential framework that mixes supervised and unsupervised learning to improve accuracy and keep up with changing financial crime. This multi-layered approach makes sure that both known fraud patterns and new anomalies are found, resulting in a system with high recall and high precision.

3.1. Data Ingestion and Enrichment

The architecture is built on a single data pipeline that takes in and changes data from many sources to make a complete picture.

3.1.1. Source Data

The system gets data from a lot of different places, such as customer data (demographics), account data (savings, loans, credit cards), transaction data (transfers, online banking), network data (beneficiaries), and risk data (historical behavior).

3.1.2. Data Transformation and Enrichment

The process starts with a set of "Enriched Data Features" that come from the source data pool, not with raw data. There are three main categories for these features: Transaction Behavior, Customer Profile, and Network Topology. This first step in feature engineering is very important because it gives the models a multi-faceted view of each event.

3.1.3. The Modeling Engine: A Two-Layered System

A two-layer modeling engine uses the enriched data to find possible fraud and then improves those predictions to make them more accurate. The system is made up of two steps that happen one after the other.

Layer 1: High-Confidence Detection with Supervised Classification

The first layer is an important first step in the screening process; it acts as a high-throughput triage engine.

This layer uses supervised learning models, such as Gradient Boosting Decision Trees (GBDT), XGBoost, and LightGBM, which are based on decision trees and boosting. These models are best for classifying structured data and are very popular in the finance industry. They can quickly find complex, non-linear relationships and can be fine-tuned to fix severe class imbalance using parameters like scale_pos_weight.

Function: The first job of this system is to quickly sort through all incoming transactions. This layer first scores each transaction and gives it one of two labels based on patterns it has learned from past data.

Positive Label (+ve): Given to transactions that are very likely to be fake. Layer 2 checks these and sends them to the positive path.

Negative Label (-ve): Given to transactions that do not meet the high-confidence standards for fraud. These are sent to Layer 2's negative path for a second look.

Strategic Goal (Getting the Most Recall): The main goal of this layer is not to be perfectly accurate, but to get the highest recall (True Positive Rate). It is set up to be very sensitive, which means it will catch as many real fraudulent transactions as possible, even if it means getting more false positives. This is a key part of risk management because the cost of missing a fraud (false negative) is usually much higher than the cost of looking into a false alarm (false positive).

Layer 2: Using unsupervised anomaly detection to check predictions again

The second layer is the most creative part of the architecture because it uses a complex system for correcting mistakes and refining decisions. Its job is to "double check wrong predictions and rethink decisions." There are two separate paths in this layer that use an unsupervised learning method.

A key new idea is to combine supervised and unsupervised learning. Layer 1 is supervised, which means it can easily spot patterns of fraud that it already knows about. Layer 2 uses autoencoders, which are a type of unsupervised learning that is great at finding anomalies because they learn the natural structure of data without using labels that have already been set.

Fixing Errors in a Symmetrical Way: The Dual-Path Design

The system makes two separate, parallel processing paths to deal with certain types of possible errors from Layer 1.

Path 1: The Positive Label (+ve) Stream (Reducing False Positives)

a) Function

This path receives all transactions that the Layer 1 classifier flagged as suspicious and is dedicated to reducing false positives.

b) Technology

It employs a novel sequence of Principal Component Analysis (PCA) and an Autoencoder.

c) PCA Dimension Reduction

PCA is applied in a targeted manner only to this high-risk subset. Its purpose is noise reduction—separating the core fraud signal from the random variations of false positives—and feature space transformation, creating a simpler, orthogonal feature space for the autoencoder.

d) Autoencoder (Positive Path)

This autoencoder is trained exclusively on a curated dataset of confirmed, labeled fraudulent transactions. It learns the "normal" structure of a true positive. When a new suspicious transaction is processed, a low reconstruction error confirms it fits the fraud pattern, while a high reconstruction error indicates it is an outlier even among suspicious events—a likely false positive.

e) Output

Final classification of Fraud (low error) or re-evaluation as a likely False Positive (high error).

Path 2: The Negative Label (-ve) Stream (Reducing False Negatives)

a) Function

This path receives the vast stream of transactions deemed safe by Layer 1 and acts as a safety net to find novel or subtle fraud patterns that were missed.

b) Technology

It uses a dedicated Autoencoder for anomaly detection.

c) Autoencoder (Negative Path)

This model is trained exclusively on a large corpus of normal, non-fraudulent transactions. It becomes an expert at recognizing the patterns of legitimate customer behavior.

During operation, if a transaction is reconstructed with a low reconstruction error, the initial "Not Fraud" decision is confirmed. If it produces a high reconstruction error, it deviates from the learned norm and is flagged as a potential false negative.

d) Output

Confirmation as "Not Fraud" (low error) or flagging as a potential False Negative (high error). This symmetrical design is a hallmark of a mature risk management system, providing a defense against both known fraud types and novel tactics designed to evade traditional supervised models.

3.2. Decisioning and Output

The final decision is a synthesis of the outputs from both layers.

3.2.1. Fraud

A transaction is classified as Fraud if it receives a positive label from either Layer 1 or Layer 2.

3.2.2. Not Fraud

A transaction is classified as Not Fraud only if it receives a negative label from Layer 1 and a subsequent negative label from Layer 2.

4. Data and Feature Engineering Foundations

The performance of any fraud detection system is ultimately determined by the quality of its data and the predictive power of its features.

4.1. The Unified Feature Store: A Single Source of Truth

"Training-serving skew" is a common failure mode in ML deployments. This happens when the features used for offline training are different from those used for live inference, which makes the model work much worse. This is often because there are separate data pipelines. These problems with the data pipeline have a huge effect on how the business runs. A real-world fraud detection program lost an astonishing 51% of known fraud transactions when it tried to map transactions between the core banking and fraud monitoring systems. The model did not do well at first because it could not be trained on a full set of fraudulent examples due to a data mismatch.

A Feature Store is the answer. It is a central place where all features are stored and managed throughout their entire lifecycle. It is a single source of truth that makes sure that both offline training and online inference have the same interface. This means that training-serving skew is not possible. Formalizing this on a strong cloud base is an important part of a modern ML platform and is necessary to get past data systems that are too fragmented.

4.2. A Comprehensive Taxonomy of Fraud Detection Features

A strong model needs a deep, multi-dimensional view of transactional entities. There are different types of features:

4.2.1. Customer Profile and History Features

These give you a starting point for knowing who the customer is and how they have acted over time. They move slowly and have static attributes like age and income, account tenure, product penetration, and historical aggregates like Recency, Frequency, and Monetary.

4.2.2. Real-Time Transactional Behavior Features

These record the details of the transaction and need to be calculated in real time. They include things like transaction attributes (amount, currency), velocity counts (transactions in the last 24 hours), deviation from norms (transaction amount vs. customer average), and time deltas (time between

transactions). Tracking transactions to new beneficiaries in real time is one of the strongest signs of fraud.

4.2.3. Session and Device Intelligence Features

These features are very important for finding account takeovers (ATOs) because they look at the user's digital session. These include channel and device attributes, indicators of device changes, IP and geolocation analysis, and analysis of session behavior.

4.2.4. Analysis of Beneficiaries and Counterparties

Fraudsters often move money to new or suspicious counterparties. These features keep track of how beneficiaries interact with each other, such as new beneficiary indicators and how they act after a transaction.

4.2.5. Network Graph Features

These are some of the most advanced features. They use a graph to show how different entities are related to each other, which helps find organized fraud rings. They include centrality measures (Degree, Eigenvector) to find important nodes like mule accounts, clustering coefficients to find fraud rings that are very close to each other, and community detection algorithms to find groups of accounts that are working together.

5. Advanced Model Validation, Performance, and Novelty

Successfully operationalizing a model requires a robust framework for validation, governance, and continuous improvement that moves beyond simple offline metrics.

5.1. Novelty Assessment

The newness of this architecture does not come from the individual algorithms, like decision trees, boosting, PCA, and autoencoders, which are all well-known. Instead, it comes from how they are put together and how they are used in certain situations.

5.2. Sophisticated Hybrid Stacking

The design is a two-stage stacked model that is very advanced. A primary supervised classifier serves as an initial filter, directing data to a secondary tier of specialized, unsupervised models for error rectification. The clear dual-path structure for both positive and negative predictions is a very complete implementation.

5.3. Targeted, Post-Classification Application of PCA

The application of PCA is arguably the most important new idea. It is not used as a global pre-processing step; instead, it is used as a targeted post-classification tool only on the high-risk data stream. This non-standard use is a specialized way to clean up the data and improve the feature space, which makes it easier for the next autoencoder to tell the difference between true positives and false positives. This is a unique design choice to use a filtered data stream with Classifier -> PCA -> Autoencoder.

5.4. Symmetrical and Specialized Error Re-evaluation

The architecture's dedication to "double-checking" both positive and negative predictions with separate, purpose-built autoencoders creates a strong and symmetrical error-correction system. This makes a system that is fast (high-throughput Layer 1), accurate (error-correction in Layer 2), and able to change to new ways of committing fraud (unsupervised Layer 2 models).

5.5. A Multi-Layered Testing Strategy

5.5.1. Backtesting

A model must be rigorously backtested against a historical dataset with known outcomes to simulate its performance and estimate business impact before it can be considered a challenger. This is the first line of defense.

5.5.2. Shadow Testing

Shadow Testing is a method that combines backtesting and live A/B testing. A challenger model is put into production and gets a copy of live data. It makes predictions that are logged for later analysis but not used to make business decisions. This lets you test stability and performance on live data at full scale without any risk.

5.5.3. A/B Testing (The Champion-Challenger Framework)

This is the standard way in the industry to test live models.

The current production model, called the "Champion," handles most of the traffic. A small, statistically significant portion of the traffic is sent to one or more "Challenger" models. We keep an eye on performance in real time against important operational KPIs. If a Challenger keeps beating the Champion, they get promoted.

Benefit

This cycle keeps the organization safe as it changes its defenses against new types of fraud**.**

5.6. Operational KPIs: Measuring What Matters

While academic metrics like AUC are useful in development, operational management requires KPIs that reflect business value.

This table illustrates the performance of a fraud detection model during its initial six-month deployment, revealing two key trends. First, a significant data integrity issue is evident when comparing "Actual Fraud" to "Actual Fraud Mapped," as a substantial number of fraudulent transactions failed to reach the model for analysis each month. Second, despite this data challenge, the model's core effectiveness shows clear and consistent improvement over time.

The Model Recall—which measures the percentage of mapped frauds the model successfully identified—steadily

increased from 20% in Month 1 to over 36% by Month 6. This suggests that while the model's overall impact was hampered by upstream data issues, its learning algorithm was progressively adapting and becoming more accurate on the data it was able to analyse.

Table 1. Operational Metrics for Evaluating Fraud Detection Models

Metric	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
Actual Fraud	47	71	23	17	72	42
Actual Fraud Mapped	25	31	18	16	49	33
Model Fraud Identified	5	7	4	4	17	12
Model Recall	20.00%	22.50%	22.22%	25.00%	34.69%	36.36%

6. Ensuring Transparency through Explainable AI (XAI)

A significant barrier to adopting sophisticated ML models in finance is the "black box" problem. Explainable AI (XAI) is a set of techniques designed to provide human-understandable explanations for ML predictions, bridging the gap between accuracy and transparency.

6.1. The Need for Explainability

The need for explainability is driven by both regulation and business needs. Regulations like GDPR are increasingly interpreted as establishing a "right to explanation" for automated decisions, making an inability to explain why a transaction was blocked a significant compliance risk. Internally, XAI is crucial for building trust with fraud analysts and business leaders, ensuring they will rely on a system whose logic can be interrogated.

6.2. A Practical Toolkit for XAI: SHAP and LIME

Two model-agnostic methods have become industry standards:

- SHAP (SHapley Additive exPlanations): Based on game theory, SHAP assigns each feature an "importance value" for a particular prediction.
    - Global Explanations: Aggregating SHAP values shows the most important features for the model overall, which is invaluable for validation and high-level justification to auditors.
    - Local Explanations: For any single transaction, SHAP can generate a "force plot" that visualizes exactly how each feature contributed to pushing the risk score higher or lower.
- LIME (Local Interpretable Model-agnostic Explanations): LIME explains a complex prediction by training a simple, interpretable model (like linear regression) on a small dataset generated around that single prediction. Its output is a simple list of the top features that influenced the decision, which is highly intuitive for a human analyst.

6.3. Operationalizing XAI

The true value of XAI is realized when it is integrated directly into the analyst's workflow. Instead of just a risk score, an analyst's case management queue can be enriched with a LIME-generated explanation like:

- Reason for Alert:
    - Transaction Amount ($12,500) contributed +0.40 to risk score.
    - Beneficiary Added < 10 mins ago contributed +0.35 to risk score.
    - Login from New Device contributed +0.15 to risk score.
    - Customer Tenure (> 5 years) contributed -0.10 to the risk score.

This immediately tells the analyst why the system flagged the transaction, enabling faster, more confident decisions and transforming the model into a collaborative assistant.

7. Conclusion

This study presents a hybrid framework that combines rule-based logic with machine learning to address the limitations of traditional fraud detection systems. By introducing a layered architecture with supervised and unsupervised models, supported by a unified feature store and explainability tools, the framework offers a scalable and adaptive solution for detecting both known and emerging fraud patterns. It balances high fraud capture with reduced false positives, aiming to improve both operational efficiency and customer trust. The proposed approach also provides a clear roadmap for implementation, including model validation and ongoing monitoring. This study contributes to building more resilient financial systems and paves the way for safer digital transactions, ultimately benefiting society through reduced financial losses and enhanced trust in digital banking.

Conflicts of Interest

The author declares that there is no conflict of interest.

References

[1] Vishnu Vardhan et al., "Anomaly Detection in Credit Card Transactions using Autoencoders," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 13, no. 3, pp. 128-133, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Fabrizio Carcillo et al., "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection," *Information Sciences*, vol. 557, pp. 317-331, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sergei Krutikov et al., "Challenging Gradient Boosted Decision Trees with Tabular Transformers for Fraud Detection at Booking.com," *arXiv Preprint*, pp. 1-9, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] V. Venu Madhav, and K. Aruna Kumari, "Analysis of Credit Card Fraud Data using PCA," *IOSR Journal of Engineering*, vol. 10, no. 1, pp. 10-14, 2020. [Publisher Link]

[5] Ela Guven, "*A Comparative Analysis of Autoencoders for Credit Card Fraud Detection*," Tilburg University Repository, 2024. [Google Scholar] [Publisher Link]

[6] Yisong Chen et al., "Year-Over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review," *arXiv Preprint*, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]