International Journal of Science and Technology (IJST)
ISSN: 3049-1118, Volume- 3, Issue- 4 (October - December 2026)

Trust and Fraud Prevention in Autonomous AI Payments: Integrating Behavioral Anomaly Detection with a Mandate-Based Authorization Framework

Krish Sharma[1] & Harsh Pradhan[2]
[1]Cloud Administrator, Rackspace Technology, India
[2]Senior Analyst, Deloitte India

Abstract

Autonomous AI agents are beginning to initiate payments for people and firms, yet fraud controls and Know-Your-Customer rules still assume that a human initiates each transaction. We propose a Know-Your-Agent (KYA) architecture that combines mandate-based authorization, modeled on the Agent Payments Protocol (AP2) and x402, with behavioral fraud scoring in which an autoencoder anomaly score is fused into an XGBoost classifier. Because no public data on agent-initiated payments exist, we built a seeded simulator of 250,000 payments from 4,000 agent-principal pairs, with a 3.0% fraud rate distributed across five agent-specific attacks, and evaluated every model on pairs held out from training. Behavioral scoring raised F1 from 0.303 for mandate checks alone to 0.774, although the autoencoder added nothing measurable over XGBoost alone (F1 0.772). Removing the five mandate-context features lowered F1 to 0.63. In leave-one-attack-type-out tests, supervised recall on an unseen attack fell to a range between 0.3% and 27%, whereas the autoencoder used as a separate channel still detected 95% of unseen token replays and 46% of unseen prompt-injection payments. On the synthetic PaySim and Bank Account Fraud (BAF) benchmarks, a simulated mandate overlay doubled F1 on BAF only because it is generated from the fraud label, and zero-shot transfer between the two datasets was weak with the overlay and weaker without it. We conclude that mandate rules, supervised scoring and anomaly detection should run as parallel channels, and we set out the liability rules, monitoring thresholds and consent-record standards that a KYA framework requires.

Keywords: agentic payments, know your agent, fraud detection, anomaly detection, mandate-based authorization, synthetic data, leave-one-attack-out evaluation

1. Introduction

Payments have always had a person near the point of initiation: someone taps a card, approves a transfer or clicks to buy. That assumption is weakening. Software agents built on large language models can now search, compare offers and pay on behalf of a person or firm that may not see the transaction until later. The International Monetary Fund describes this as a shift in who controls the payment decision and argues that the outcome depends on institutional

Published: 09 October 2026
DOI: https://doi.org/10.70558/IJST.2026.v3.i4.241335
Copyright © 2026 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

design and governance (Davidovic & Tourpe, 2026). Google introduced the Agent Payments Protocol (AP2) in September 2025 in collaboration with more than sixty partners (Parikh & Surapaneni, 2025), Coinbase released the x402 protocol for stablecoin payments over HTTP in May 2025 (Reppel et al., 2025), and Mastercard announced agent-specific payment tokens in April 2025 (Mastercard, 2025). Advisory firms now list agentic AI among the main forces reshaping payments in 2026 (BDO USA, 2026).

Know-Your-Customer (KYC) and Anti-Money-Laundering (AML) regimes assume that the party initiating a transaction is a natural or legal person who can be identified and held liable. An agent strains that assumption. It acts on delegated, time-limited authority; it can be manipulated by instructions hidden in the content it reads (Greshake et al., 2023), and its identity is better described as a chain of attestations linking it to a principal (South et al., 2025a). Mandate-based protocols such as AP2 bind each agent action to a signed record of what the principal authorized. A mandate states what an agent may do, but it cannot show that a payment inside those limits is legitimate: mandates can be obtained by fraud, authorizations can be replayed, and an agent can be steered to spend its allowance on purposes the principal never intended.

This paper asks whether behavioral monitoring placed alongside mandate verification can detect these failures, and how much of that claim current data can support. No public dataset of real agent-initiated payments exists, so our primary evidence comes from a simulator that we document in full. We also use two public benchmarks, PaySim (Lopez-Rojas et al., 2016) and the Bank Account Fraud (BAF) suite (Jesus et al., 2022). Both are synthetic, so they test the modeling pipeline on data we did not generate but do not provide evidence of performance on real fraud.

The study has four objectives: to specify an architecture that combines mandate verification with behavioral scoring, in which an autoencoder's reconstruction error is fused into a gradient-boosted classifier; to build a seeded simulation containing five agent-specific attacks and deliberately ambiguous cases; to evaluate it on agent-principal pairs never seen in training, with temporal, leave-one-attack-type-out and mandate-feature ablation tests; and to analyze the fraud economics, liability and regulatory gaps of this setting and propose a Know-Your-Agent (KYA) framework. Section 2 reviews related work, Section 3 describes the architecture, Section 4 sets out the methodology, Section 5 reports results, Section 6 discusses economics, liability, regulation and limitations, and Section 7 concludes.

2. Literature Review

2.1.Agentic Payments and Mandate-Based Authorization

AP2 centers on the mandate, a tamper-evident, cryptographically signed record of a user's instructions. An Intent Mandate captures the request and its conditions, and a Cart Mandate records the exact items and price approved before payment (Parikh & Surapaneni, 2025; Google, 2025). x402 revives the HTTP 402 "Payment Required" status code so that a client, including an autonomous agent, can pay a quoted price in stablecoins within the same request (Reppel et al., 2025); Cloudflare and Coinbase announced a foundation to govern it as an open standard in September 2025 (Cloudflare, 2025). These protocols ground authorization in

verifiable consent. They say little about payments that are properly authorized yet fraudulent, which is the gap addressed here.

2.2.Agent Identity and Delegated Authority

South et al. (2025a) propose extending OAuth 2.0 and OpenID Connect with agent-specific credentials for narrowly scoped delegation, and an OpenID Foundation white paper surveys the open authentication and authorization problems that agents raise (South et al., 2025b). Guidance on digital identity for customer due diligence defines assurance levels for natural and legal persons but not for software acting on their behalf (FATF, 2020). The threat model also differs: indirect prompt injection lets planted instructions redirect a genuine agent holding genuine credentials (Greshake et al., 2023), which identity checks alone cannot catch.

2.3.Fraud Detection and Evaluation

Statistical fraud detection is long established (Bolton & Hand, 2002), and work on card fraud stresses class imbalance, drift and delayed labels, which call for time-ordered evaluation (Dal Pozzolo et al., 2018). XGBoost is a strong default for tabular data (Chen & Guestrin, 2016), and reconstruction-based autoencoders are the usual unsupervised alternative because they can flag patterns absent from labeled data (Pang et al., 2021). Hybrids pair the two: Ding et al. (2024) used an autoencoder with LightGBM on card data, Shanaa and Abdallah (2025) combined supervised models with autoencoders and isolation forests, and Hari Vinayak and Jarin (2025) paired a stacked autoencoder with XGBoost for intrusion detection. None of these studies isolates whether the unsupervised part helps with attacks the supervised part never saw.

Evaluation design matters as much as the model. Leakage inflates results (Kaufman et al., 2012), grouped validation is needed when many records share an entity (Roberts et al., 2017), and precision-recall analysis is more informative than ROC under imbalance (Saito & Rehmsmeier, 2015). Public fraud benchmarks are also synthetic. PaySim comes from an agent-based simulator calibrated on mobile-money logs, and its public release contains fraud produced by simulated fraudulent agents (Lopez-Rojas et al., 2016; Lopez-Rojas, 2017). BAF was generated by CTGAN models trained on noised, privacy-protected bank data (Jesus et al., 2022). Finally, this literature models human behavior using devices, typing and location, none of which applies to an agent running a task specification on cloud infrastructure.

2.4.Regulatory Treatment of Non-Human Principals

The U.S. GENIUS Act treats permitted payment stablecoin issuers as financial institutions under the Bank Secrecy Act and sets civil penalties of up to $100,000 per day, with up to a further $100,000 per day for knowing participation, reaching institution-affiliated parties as well as issuers (Guiding and Establishing National Innovation for U.S. Stablecoins Act, 2025). Treasury's advance notice of proposed rulemaking asked how issuers would block, freeze and reject impermissible transactions (U.S. Department of the Treasury, 2025). Neither addresses a payment initiated by a software agent acting under a scoped mandate.

3. System Architecture

The system has three layers (Figure 1): mandate issuance and verification, a behavioral signal collector and a fraud-scoring engine, followed by a decision layer and an audit trail.

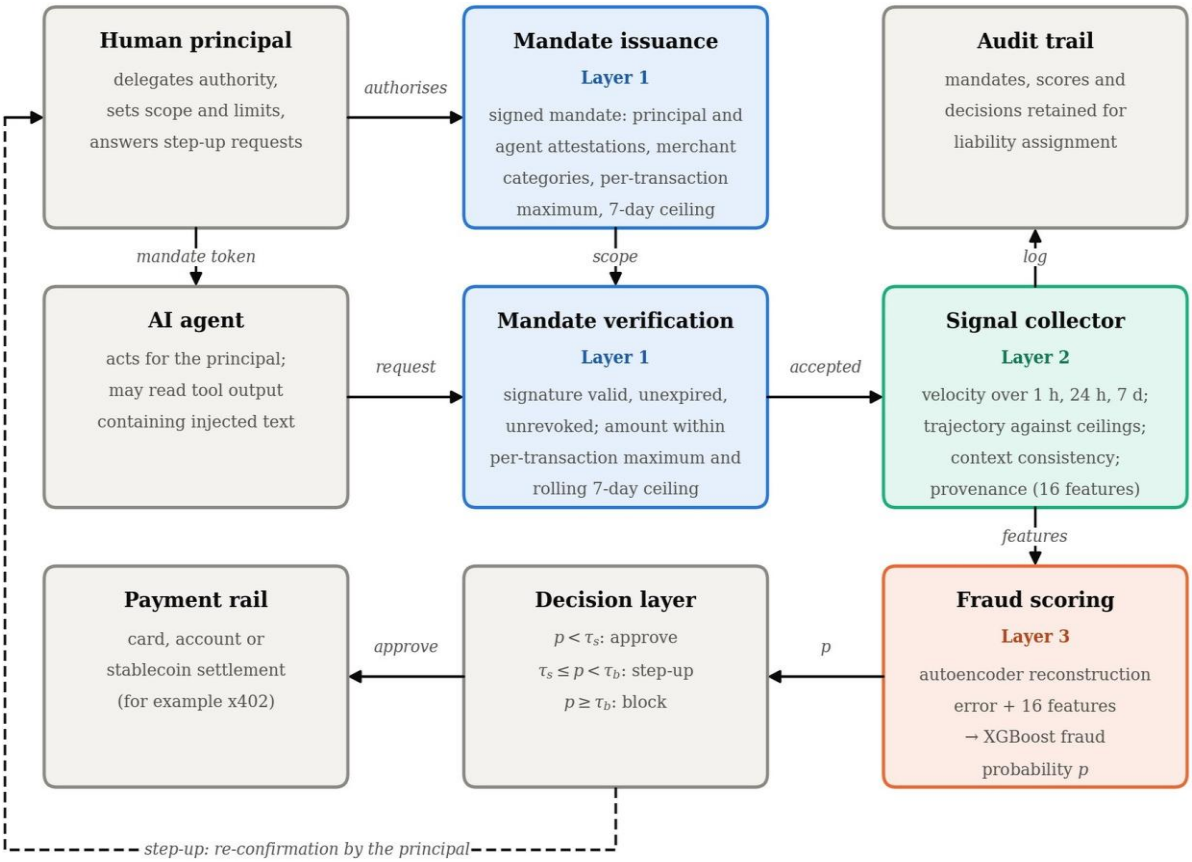


Figure 1. Architecture of the proposed Know-Your-Agent trust system. Layer 1 issues and verifies mandates, Layer 2 computes sixteen behavioral and mandate-context features for each accepted payment, and Layer 3 produces a fraud probability p. The decision layer approves the payment, requests step-up confirmation from the principal or blocks it, and every decision is written to the audit trail.

3.1.Mandate Issuance and Verification

The issuance service gives each agent a signed mandate containing principal and agent attestations, permitted merchant categories, a per-transaction maximum, a rolling seven-day ceiling, a validity period and a revocation endpoint. The verifier rejects any request with an invalid, expired or revoked token, or one that exceeds either limit, following the mandate-binding approach of AP2 (Parikh & Surapaneni, 2025). Only accepted payments reach the subsequent layers, and the simulator is built the same way.

3.2.Behavioral Signal Collector

For each accepted payment, the collector computes sixteen features in four families: velocity (payment counts over the last hour, day and week, and spending against the declared budget), trajectory (amount relative to the pair's history and progress toward the ceiling), context consistency (match between the declared task, the merchant and the session's intent) and

mandate context (attestation, nonce uniqueness, authorization reuse, ceiling proximity and identity proofing). Section 4.2 defines them.

3.3.Fraud-Scoring Engine and Decision Layer

An autoencoder trained only on legitimate payments produces a reconstruction-error anomaly score. An XGBoost classifier (Chen & Guestrin, 2016) takes the sixteen features plus this score and outputs a fraud probability p. The intent is that the classifier handles known attacks while the anomaly score carries information about unfamiliar ones; Section 5.3 tests this. Payments with p below a step-up threshold τ_s are approved, those with $\tau_s \leq p < \tau_b$ are sent to the principal for confirmation, and those with $p \geq \tau_b$ are blocked. Every mandate, score and decision is written to an append-only audit trail, which the liability framework in Section 6 relies on.

4. Methodology

4.1.Simulated Agentic-Payment Dataset

The simulator is written in Python and driven by a single NumPy generator with seed 42, so the dataset can be regenerated exactly. It produces 250,000 accepted payments from 4,000 agent-principal pairs over 90 days: 242,500 legitimate and 7,500 fraudulent (3.0%), with 1,500 per attack type. Held-out test sets have slightly different rates because they are drawn by pair; those rates are reported with each result.

Of the pairs, 3,900 are legitimately onboarded and 100 are created under fabricated identities. Each pair is a recurring-purchase (40%), procurement (30%) or personal-finance (30%) agent. Amounts follow log-normal distributions with archetype log-means of 3.4, 5.8 and 4.3 and log-standard deviations of 0.35, 0.70 and 0.90, shifted per pair by N(0, 0.4Z) and scaled by U(0.8, 1.2). Activity weights are 1.0, 1.5 and 1.2 per archetype with a log-normal pair multiplier (σ = 0.5). The per-transaction maximum is $\exp(\mu_i + 2.3\sigma_i) \times U(1.2, 2.0)$, and the seven-day ceiling is seven times the declared daily budget × U(1.8, 3.0). Identity-proofing and attestation baselines are drawn per pair from Beta(12, 2) and Beta(25, 1.5), with small per-payment noise. Legitimate context and provenance scores come from Beta(8, 2) for task-merchant match, Beta(9, 2) for intent similarity, Beta(30, 1) for nonce uniqueness and Poisson(0.02) for reuse count.

Fraud is injected by changing how payments are generated (Table 1). Each attack instance has a strength s equal to 1, except for 15% near-miss instances with s from U(0.2, 0.5) (1,129 fraudulent payments in the final data). Conversely, 2% of legitimate payments become hard negatives of five kinds: near-ceiling purchases, new merchant categories, network retries, identity re-verification and unusually small purchases (4,724 in the final data). The simulator generates surplus candidates (257,050 legitimate and 3,000 per attack), passes them in time order through a verifier that enforces both mandate limits (3.7% were rejected), and then samples exactly 242,500 legitimate payments and 1,500 per attack. Scope-creep agents size each payment to between 85% and 99% of their remaining headroom.

Table 1. Attack scenarios injected by the simulator. s is the attack strength (1 for 85% of instances, U(0.2, 0.5) for near-miss instances); a is a legitimate amount drawn for the pair, and M is its per-transaction maximum.

Attack	Mechanism	Generation rule	Pairs	Payments
Mandate scope creep	A compromised agent spends up to the edge of its authority without breaching it	7-day episode; target amount (1- s)a + s·U(0.55, 0.98)M, capped at U(0.85, 0.99) of the remaining 7-day headroom	250	1,500
Prompt-injected re-authorization	Instructions planted in tool output redirect payments during a session	3-hour session; amount ×(1 + 0.3s); task-merchant match (1 - s)Beta(8, 2) + s·Beta(4, 4); intent similarity (1 - s)Beta(9, 2) + s·Beta(4, 3); attestation reduced by s·U(0, 0.12)	300	1,500
Attack	Mechanism	Generation rule	Pairs	Payments
Mandate token replay	A captured signed authorization is presented again	Copy of an earlier legitimate payment of the pair, 1 to 30 minutes later, same amount; nonce uniqueness (1 - s)Beta(30, 1) + s·Beta(6, 4); reuse count round(s(1 + Poisson(1)))	150	1,500
Synthetic-identity onboarding	An agent is registered under a fabricated principal to obtain a mandate	New pair, all payments fraudulent, 10-day window; identity baseline (1 - s)Beta(12, 2) + s·Beta(5, 3); attestation baseline (1 - s)Beta(25, 1.5) + s·Beta(9, 3); amount ×(1 + 0.8s)	100	1,500

Low-and-slow draining	Many small payments over weeks, each unremarkable on its own	14- to 21-day episode of extra payments; amount ×((1 - s) + s·U(0.10, 0.35))	60	1,500

4.2.Feature Definitions

Table 2 defines the sixteen features: eleven behavioral and five mandate-context. History windows count only payments made strictly earlier of the same pair, except that spending windows include the current amount, which is known at authorization. The amount deviation is shrunk toward the declared archetype prior (pseudo-count 5). No feature uses the label, attack type or strength.

Table 2. Definitions of the sixteen features. The first eleven are behavioral; the last five form the mandate-context family that is removed in the ablation and simulated in the PaySim and BAF overlays.

Family	Feature	Definition
Velocity	v_count_1h	Number of the pair's payments in the preceding hour
Velocity	v_count_24h	Number of the pair's payments in the preceding 24 hours
Velocity	v_count_7d_ratio	Payments in the preceding 7 days divided by 7 × the declared daily payment rate
Velocity	v_spend_24h_ratio	Spending in the 24 hours up to and including this payment divided by the declared daily budget
Velocity	v_log_gap	ln(1 + seconds since the pair's previous payment); 7 days for a pair's first payment
Trajectory	log_amount	ln(amount)

Family	Feature	Definition
Trajectory	amount_z	(ln amount - m)/√v, where m and v are the running mean and variance of ln amount over the pair's earlier payments, shrunk toward the archetype prior with pseudo-count 5
Trajectory	t_cum_ceiling_ratio	Spending in the 7 days up to and including this payment divided by the 7-day ceiling
Trajectory	t_ceiling_approach_rate	t_cum_ceiling_ratio now minus its value 24 hours earlier
Context	c_task_merchant_match	Similarity in [0, 1] between the declared task and the merchant category
Context	c_session_intent_sim	Similarity in [0, 1] between the session's initial intent and the current action
Mandate context	m_ceiling_proximity	Amount divided by the per-transaction maximum
Mandate context	m_attestation_validity	Validity score in [0, 1] of the principal-agent attestation chain
Mandate context	m_nonce_uniqueness	Nonce uniqueness score in [0, 1]; 1 means the nonce has never been seen
Mandate context	m_reuse_count	Number of earlier presentations of the same signed authorization
Mandate context	m_identity_proofing	Identity-proofing score of the principal in [0, 1]

4.3.Data Splits and Leakage Controls

The primary split is principal-wise. Within each pair category (legitimate-only or attack type), 20% of pairs form the test set: 801 pairs and 50,425 payments, of which 1,474 (2.92%) are fraudulent. Of the remaining pairs, 15% form a validation set (29,458 payments) and the rest the fitting set (170,117). No pair appears in more than one set, so a model cannot recognize a fraudulent pair instead of fraudulent behavior (Roberts et al., 2017). For comparison, we also report a temporal split (fit on days 0 to 50, validate on days 50 to 60, test on days 60 to 90) and a random transaction-level split.

Further controls against leakage (Kaufman et al., 2012) are as follows: Budgets, ceilings and priors come from registration data, not from the dataset. All scalers and models are fitted on the fitting set only. The autoencoder scores used to train the hybrid model are cross-fitted: each of three groups of fitting pairs is scored by an autoencoder trained on the other two. Hyperparameters are tuned by grouped three-fold cross-validation within the training pairs, and thresholds are chosen on the validation set.

4.4.Models and Evaluation Protocol

We compare a rule-based mandate check with fixed thresholds set before any data were generated, logistic regression on the behavioral features, a random forest, XGBoost, the hybrid model and the autoencoder alone. Table 3 gives all settings, the tuned XGBoost parameters, the thresholds and the seeds. The blocking threshold τ_b maximizes F1 on validation data, and the step-up threshold τ_s is the 99th percentile of hybrid scores on legitimate validation payments.

Table 3. Models, hyperparameters, thresholds and random seeds.

Component	Setting
Random seeds	42 for the simulator, all splits, the bootstrap and every model; 43 and 44 for the PaySim and BAF mandate overlays
Rule-based mandate check	Flag if m_ceiling_proximity ≥ 0.90, t_cum_ceiling_ratio ≥ 0.90, m_attestation_validity < 0.60, m_nonce_uniqueness < 0.50 or m_reuse_count ≥ 2; fixed before the data were generated
Logistic regression	11 standardized behavioral features; L2 penalty, C = 1.0, balanced class weights, L-BFGS solver, up to 2,000 iterations
Random forest	300 trees, √d features per split, at least 2 samples per leaf
XGBoost (standalone and hybrid)	Grid over depth {4, 6}, learning rate {0.05, 0.10} and trees {200, 400} by grouped 3-fold cross-validation on training pairs; selected depth 6, learning rate 0.05, 400 trees; subsample 0.8, column sample 0.8, minimum child weight 1, histogram tree method
Autoencoder	Layers 16-12-6-12-16 (11-12-6-12-11 in the behavioral-only ablation), ReLU, Adam with learning rate 0.001, batch size 1,024, L2 weight 0.0001, at most 100 epochs with early stopping (10% internal split, patience 5); inputs standardized on legitimate fitting payments; score is the mean squared reconstruction error; 3-fold

	grouped cross-fitting; for PaySim and BAF, trained on at most 200,000 legitimate fitting rows
Blocking threshold τ_b	Maximum-F1 point on the validation set: hybrid 0.407, XGBoost 0.350, random forest 0.366, logistic regression 0.882, autoencoder 1.887 (reconstruction error)
Step-up threshold τ_s	99th percentile of hybrid scores on legitimate validation payments: 0.1686

We report precision, recall, F1, AUC-PR (average precision) and the false-positive rate, with 95% confidence intervals for F1 from 500 bootstrap resamples of test pairs. Five-fold grouped cross-validation checks stability. In leave-one-attack-type-out testing, each attack type is removed from the fitting and validation sets, models are retrained, and recall is measured on test pairs containing only legitimate payments and the held-out attack. The ablation retrains XGBoost and the hybrid model on the eleven behavioral features alone. Latency is the median time to score one payment on a single CPU thread over 1,000 calls. Experiments ran on a two-core Linux machine with 8 GB of memory using Python 3.11, scikit-learn 1.8.0, XGBoost 3.2.0, NumPy 2.4.4 and pandas 3.0.2.

4.5.External Benchmarks: PaySim and BAF

The PaySim release (Lopez-Rojas, 2017) contains 6,362,620 simulated transactions over 743 hourly steps, of which 8,213 are fraudulent and all are of type TRANSFER or CASH_OUT. We modeled all 2,770,409 transactions of those two types without sampling and used the remaining 3,592,211 only to count earlier payments received by each destination account. The split is temporal: steps 1 to 350 for fitting, 351 to 400 for validation and 401 to 743 for testing. The BAF Base variant (Jesus et al., 2022) contains 1,000,000 synthetic account applications over eight months, 11,029 (1.10%) fraudulent; following its authors, we fitted on months 0 to 4, validated on month 5 and tested on months 6 and 7. PaySim features are amounts, balances, balance errors, hour and destination velocity; BAF features are its 30 released attributes other than the label and month.

Neither dataset has mandate information, so we added a simulated overlay with the same five mandate features and distributions as the simulator, assigning each fraudulent row a random attack type. Because the overlay is generated from the label, every external result is reported with and without it. For cross-dataset testing, we defined eight proxy features computable from both datasets (amount, exposure, velocity at three horizons, identity inconsistency, counterparty or device anomaly, and tenure), rank-transformed within each dataset using only its unlabeled training rows. A model trained and thresholded on one dataset was applied unchanged to the other's test period and compared with a natively trained model. The two overlays were generated independently (seeds 43 and 44).

5. Results

5.1.Model Comparison on Held-Out Principals

Table 4 and Figure 2 compare the models on the principal-wise test set. The rule-based mandate check detects 31.8% of fraud with a precision of 0.288 while flagging 2.36% of legitimate payments, which confirms that checking whether a payment is within scope does not separate fraud from legitimate agent activity. Logistic regression reaches an F1 of 0.412. The tree-based models are close together: XGBoost reaches an F1 of 0.772 (AUC-PR 0.832), the hybrid model 0.774 (0.829) and the random forest 0.765 (0.820). The difference between the hybrid and XGBoost lies well inside the bootstrap intervals. The autoencoder score shifts the operating point rather than the ranking, trading recall (0.695 against 0.728) for precision (0.873 against 0.821). It has the second-largest gain importance (0.137, after the one-hour payment count at 0.420), so it carries signal, but that signal largely duplicates the engineered features. Median single-payment latency was below 1 ms for all models except the random forest (15.1 ms), excluding feature retrieval.

Table 4. Model comparison on the principal-wise test set (50,425 payments from 801 held-out pairs; 1,474 fraudulent, 2.92%). Thresholds were chosen on validation pairs. FPR is the false-positive rate; latency is the median single-payment inference time on one CPU thread.

Model	Precision	Recall	F1	AUC-PR	FPR (%)	Latency (ms)
Rule-based mandate check	0.288	0.318	0.303	0.112*	2.36	0.004
Logistic regression (behavioral features)	0.443	0.385	0.412	0.387	1.46	0.096
Random forest	0.834	0.707	0.765	0.820	0.42	15.149
XGBoost	0.821	0.728	0.772	0.832	0.48	0.468
Hybrid: autoencoder + XGBoost	0.873	0.695	0.774	0.829	0.30	0.841
Autoencoder score alone	0.740	0.303	0.429	0.445	0.32	-

Note. *The rule-based check produces a binary flag, so its AUC-PR is the average precision of a single operating point. F1 95% confidence intervals from a pair-level bootstrap: XGBoost 0.735 to 0.801; hybrid 0.737 to 0.804.

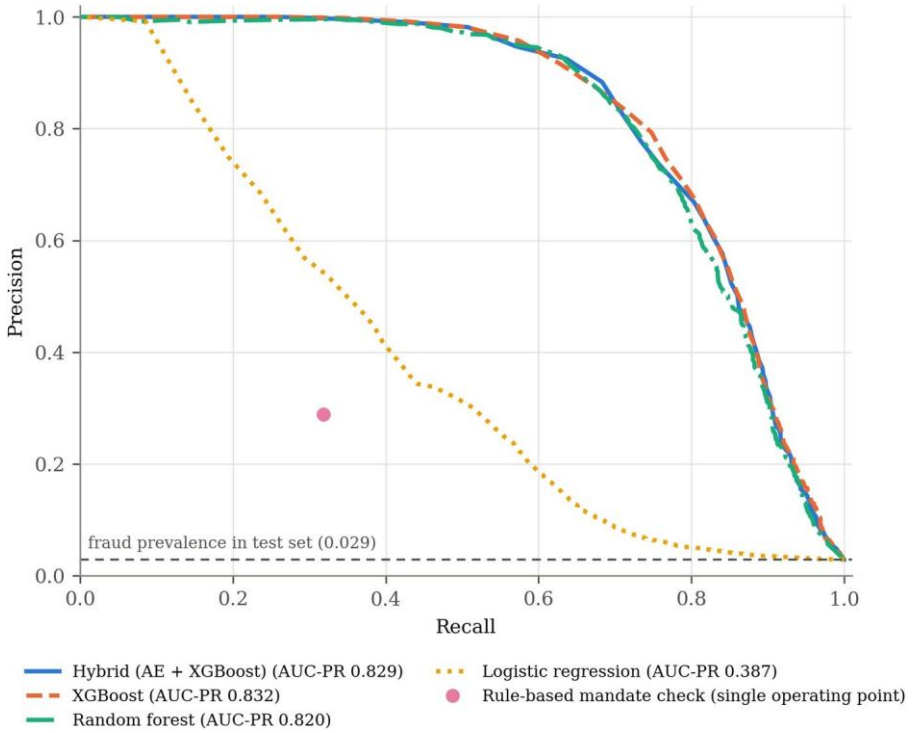


Figure 2. Precision-recall curves on the principal-wise test set (801 held-out agent-principal pairs). The rule-based mandate check produces a single operating point. The dashed horizontal line marks the fraud prevalence in the test set.

5.2.Effect of the Split Design

Moving from a random to a principal-wise split lowers F1 by about 0.015 for XGBoost and

0.018 for the hybrid model, and the temporal split gives similar or slightly lower values (Table 5). The effect is modest because most attack signal in this simulator lies in individual payments, but its direction is the one expected from leakage. Five-fold grouped cross-validation gives F1 values of 0.778 ± 0.006 for XGBoost and 0.775 ± 0.008 for the hybrid model, so the two cannot be separated on this benchmark.

Table 5. Effect of the split design. Test sizes are given with the fraud rate in parentheses. The random split lets payments from the same pair fall on both sides; the temporal split keeps time order but reuses pairs; the principal-wise split holds out whole pairs.

Split	Test payments	Rule F1	XGBoost F1	XGBoost AUC-PR	Hybrid F1	Hybrid AUC-PR
Random, transaction-level	50,000 (3.00%)	0.282	0.787	0.843	0.792	0.842

Temporal (days 60 to 90)	83,019 (2.86%)	0.269	0.760	0.829	0.763	0.824
Principal-wise (primary)	50,425 (2.92%)	0.303	0.772	0.832	0.774	0.829
Principal-wise, 5-fold grouped CV (mean ± SD)	250,000 over 5 folds	-	0.778 ± 0.006	0.833 ± 0.006	0.775 ± 0.008	0.827 ± 0.005

5.3.Detection by Attack Type and Leave-One-Attack-Type-Out Testing

With all attacks in training, the hybrid model detects 95.1% of token replays, 83.8% of synthetic-identity payments, 70.7% of low-and-slow payments and 67.3% of prompt-injection payments, but only 32.5% of scope creep (Table 6). Scope-creep payments stay inside every limit and resemble a heavy legitimate user; the fixed cumulative-ceiling rule catches more of them (47.5%), at a high false-positive rate.

When an attack type is withheld from training, recall on it collapses for both supervised models: from 0.3% (synthetic identity) to 26.6% (token replay) for XGBoost, and from 0.6% to 20.9% for the hybrid model. Fusing the anomaly score into the classifier does not help, because without examples the classifier learns to discount the score where the unseen attack occurs. The autoencoder on its own still detects 95.1% of held-out token replays and 45.5% of held-out prompt-injection payments, but little of the individually plausible attacks (3.4% of scope creep, 10.4% of synthetic identities, 5.9% of low-and-slow).

We therefore tested, as a post hoc variant, flagging a payment if either the hybrid score or the autoencoder score exceeds its own validation threshold. Held-out recall rose to 95.8% for token replay and 48.2% for prompt injection (Figure 3), while the false-positive rate rose from 0.20%–0.42% to 0.51%–3.28%. With all attacks in training, this rule gave an F1 of 0.740, slightly below the hybrid alone. It still missed most held-out scope creep, synthetic identities and low-and-slow payments, for which identity proofing and mandate rules remain the main defenses.

Table 6. Recall (%) by attack type on principal-wise test pairs. "Seen" models were trained on all five attack types; "held out" models were retrained without the attack type in that row (leave-one-attack-type-out). The rule-based check uses no labels; the autoencoder is trained without labels, and the value shown uses a threshold set on validation data that exclude the attack in that row.

Attack	Rule	AE	XGB seen	XGB held out	Hybrid seen	Hybrid held out	Hybrid or AE held out

Mandate scope creep	47.5	3.4	35.3	2.4	32.5	3.4	6.4
Prompt-injected re-authorization	24.1	45.5	74.3	8.9	67.3	6.3	48.2
Mandate token replay	68.8	95.1	97.0	26.6	95.1	20.9	95.8
Synthetic-identity onboarding	22.0	10.4	86.4	0.3	83.8	0.6	10.7
Low-and-slow draining	2.3	5.9	73.0	4.9	70.7	5.9	11.2

Note. AE = autoencoder alone; XGB = XGBoost. "Hybrid or AE" flags a payment if the hybrid score exceeds its threshold or the autoencoder score exceeds its own validation threshold. False-positive rates on legitimate test payments in the held-out runs: hybrid 0.20% to 0.42%; hybrid or AE 0.51% to 3.28%.

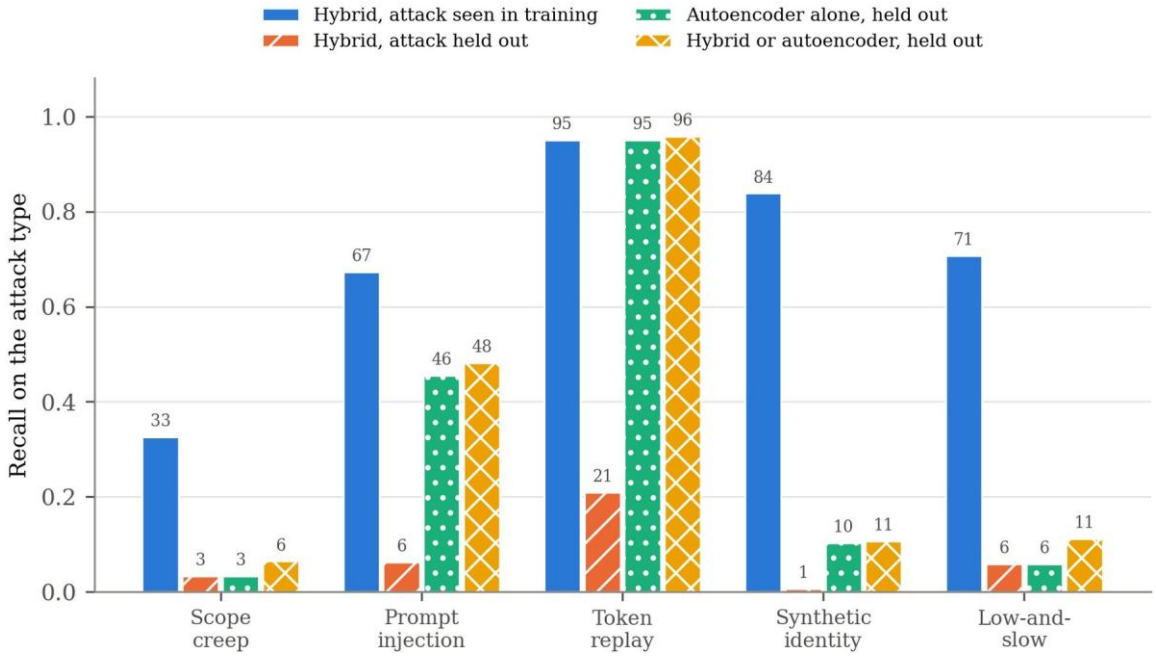


Figure 3. Recall by attack type when the attack was present in training (first bar) and when it was withheld from training (other bars). The last bar flags a payment if either the hybrid score or the autoencoder score exceeds its own validation threshold. Numbers above the bars are percentages.

5.4.Mandate-Feature Ablation

Removing the five mandate features lowers F1 from 0.772 to 0.629 for XGBoost and from 0.774 to 0.631 for the hybrid model (Table 7). The loss falls on the credential-based attacks: hybrid recall drops from 95.1% to 65.4% for token replay and from 83.8% to 42.4% for synthetic identity. The mandate features alone give an F1 of 0.379. The two feature groups are complementary, and neither is sufficient alone.

Table 7. Mandate-feature ablation on the principal-wise test set. The last two columns give recall (%) on the two attack types that act mainly through mandate and provenance signals.

Features	Model	Precision	Recall	F1	AUC-PR	Token replay	Synthetic identity
All 16	XGBoost	0.821	0.728	0.772	0.832	97.0	86.4
All 16	Hybrid	0.873	0.695	0.774	0.829	95.1	83.8
11 behavioral only	XGBoost	0.638	0.620	0.629	0.699	69.6	46.9
11 behavioral only	Hybrid	0.687	0.583	0.631	0.696	65.4	42.4
5 mandate only	XGBoost	0.488	0.309	0.379	0.354	87.8	56.6

5.5.External Benchmarks and Cross-Dataset Transfer

PaySim is almost perfectly separable with its own features: the hybrid model reaches an F1 of 0.995 on steps 401 to 743, detecting 3,702 of 3,736 fraudulent transactions with 7 false positives (Table 8). It relies mostly on the origin balance after the transaction and the origin balance error (gain importance 0.746 and 0.145), which reflect how PaySim generates fraud by emptying accounts rather than the strength of the model. BAF is much harder: F1 0.265 and AUC-PR 0.192 with its own features. Adding the overlay roughly doubles F1 to 0.516. That gain comes entirely from label-derived features, which is why overlay results cannot validate the mandate layer.

Table 8. Native temporal evaluation on the two public synthetic benchmarks, with and without the simulated mandate overlay. PaySim test: steps 401 to 743, 240,873 transactions, 3,736 fraudulent. BAF test: months 6 and 7, 205,011 applications, 2,878 fraudulent.

Dataset	**Model**	**F1, domain + overlay**	**AUC-PR, domain + overlay**	**F1, domain only**	**AUC-PR, domain only**
PaySim	Rule-based check (overlay features)	0.287	0.161	-	-

PaySim	Logistic regression (domain features)	-	-	0.975	0.994
PaySim	XGBoost	0.979	0.999	0.965	0.999
PaySim	Hybrid	0.994	1.000	0.994	1.000
BAF	Rule-based check (overlay features)	0.288	0.158	-	-
BAF	Logistic regression (domain features)	-	-	0.214	0.153
BAF	XGBoost	0.512	0.531	0.263	0.189
BAF	Hybrid	0.516	0.536	0.265	0.192

Note. The rule-based check uses only the five overlay features, and logistic regression uses only the domain features. Test prevalence: PaySim 1.55%, BAF 1.40%.

The cross-dataset results do not support a claim of transferable fraud signal (Table 9, Figure 4). With the proxies and the shared overlay, a model trained on BAF reaches an AUC-PR of 0.372 on PaySim, compared with 0.997 natively, and a model trained on PaySim reaches 0.038 on BAF, compared with 0.391 natively. Without the overlay, PaySim-to-BAF transfer is at chance (AUC-PR 0.017 at a prevalence of 1.40%), and BAF-to-PaySim keeps only a weak signal (0.132 at a prevalence of 1.55%, against 0.991 natively). Several proxies move in opposite directions in the two domains: the identity-inconsistency proxy rises for BAF fraud but falls for PaySim fraud.

Table 9. Cross-dataset transfer on the common proxy schema. Each row names the training dataset and the test dataset; zero-shot models receive no data or labels from the test dataset except its unlabeled training rows, which are used only for rank normalization.

Feature set	Train → test	XGBoost F1	XGBoost AUC-PR	Hybrid F1	Hybrid AUC-PR
8 proxies + 5 simulated mandate features	PaySim → PaySim (native)	0.932	0.994	0.933	0.997
8 proxies + 5 simulated mandate features	BAF → PaySim (zero-shot)	0.384	0.365	0.393	0.372
8 proxies + 5 simulated mandate features	BAF → BAF (native)	0.410	0.390	0.410	0.391

8 proxies + 5 simulated mandate features	PaySim → BAF (zero-shot)	0.075	0.193	0.051	0.038
8 proxies only	PaySim → PaySim (native)	0.896	0.995	0.866	0.991
8 proxies only	BAF → PaySim (zero-shot)	0.155	0.075	0.229	0.132
8 proxies only	BAF → BAF (native)	0.116	0.057	0.117	0.056
8 proxies only	PaySim → BAF (zero-shot)	0.014	0.016	0.005	0.017

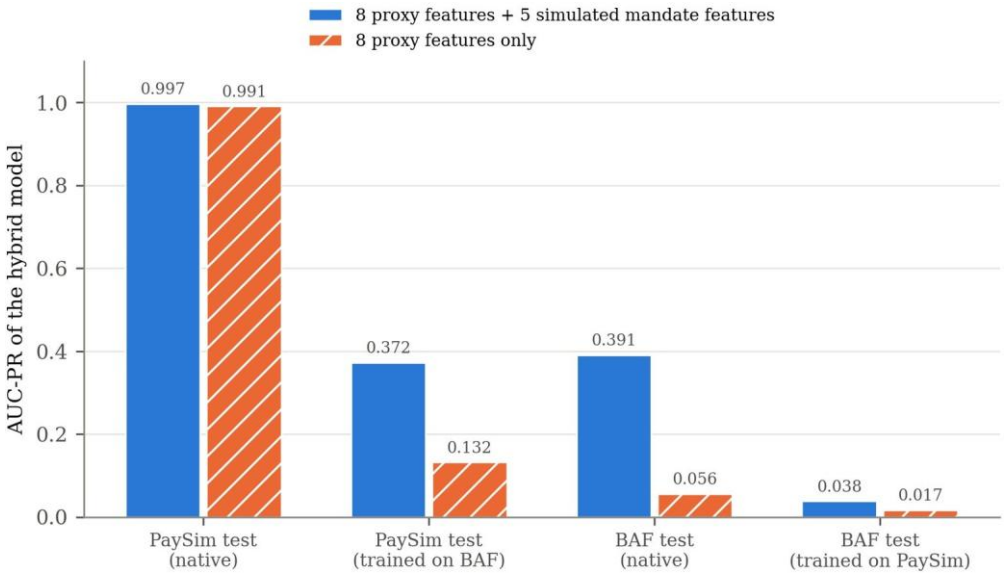


Figure 4. AUC-PR of the hybrid model on the PaySim and BAF test periods when trained on the same dataset (native) or on the other dataset (zero-shot), with and without the five simulated mandate features. Test prevalence is 1.55% for PaySim and 1.40% for BAF.

6. Discussion

6.1.Interpretation

Behavioral scoring adds a great deal to mandate verification, lifting recall from under a third to about 70% at a false-positive rate below 0.5%. Fusing the autoencoder into the classifier did not deliver robustness to novel attacks; the anomaly score helped only as a separate channel. Evidence for generalization beyond the simulator is weak, since both benchmarks are synthetic and transfer between them is poor. In practice, this argues for mandate rules, a supervised

model and an anomaly detector running in parallel, each able to trigger step-up verification, and for evaluations that hold out principals and attack types.

6.2.Fraud Economics

For N agent payments per day at a fraud rate f and mean fraudulent amount v, a detector with recall r and false-positive rate q, and friction cost c per legitimate payment stepped up or blocked, the expected daily fraud loss and friction cost are:

$$L_f = \mathrm{N} \times \mathrm{f} \times (1 - \mathrm{r}) \times \mathrm{v} \qquad (1)$$

$$L_c = \mathrm{N} \times (1 - \mathrm{f}) \times \mathrm{q} \times \mathrm{c} \qquad (2)$$

Table 10 uses N = 100,000, f = 3.0%, v = $210.10 (the simulated mean) and an assumed c =

$5.00, with r and q measured on the test set. The mandate check alone leaves $429,744 of daily fraud loss. Blocking at τ_b reduces it to $192,423 with $1,476 of friction, and adding the step-up tier (0.9% of payments) reduces it to $148,807 with $4,726 of friction, counting a step-up as effective only when a genuine principal is asked. The step-up tier lowers total loss when friction stays below about $67 per affected payment at f = 3.0%, or about $6.50 at f = 0.3%, so τ_s should be set from each platform's own fraud rate and friction cost.

Table 10. Illustrative expected daily loss for N = 100,000 agent payments per day, fraud rate f = 3.0%, mean fraudulent amount v = $210.10 (the simulated mean) and friction cost c =

$5.00 per legitimate payment that is stepped up or blocked. Recall and false-positive rate are measured on the principal-wise test set.

Configuration	Recall (%)	FPR (%)	Fraud loss	Friction cost	Total
Rule-based mandate check	31.8	2.36	$429,744	$11,463	$441,207
Hybrid, block at τ_b	69.5	0.30	$192,423	$1,476	$193,899
Hybrid, step-up at τ_s and block at τ_b	76.4	0.97	$148,807	$4,726	$153,533

6.3.Liability Allocation

A mandate system creates a chain of delegated authority from principal to agent, tool and payment rail, and current law says little about how losses move along it. We propose that liability follow the point of failure: the tool or runtime provider when a properly scoped agent is deceived, the issuing service when a mandate is obtained with a fabricated identity or replayed, and the principal when the scope granted far exceeds the task. The GENIUS Act offers a partial precedent in penalties that reach institution-affiliated parties and distinguish knowing participation, but it does not contemplate an agent between principal and rail. The October 2015 EMV liability shift is a closer analogy: it placed counterfeit card-present losses on whichever party, issuer or merchant, had not adopted chip technology (U.S. Payments

Forum, 2017). Its record is mixed. Card-present fraud rates for some debit cards did not fall and losses shifted toward merchants and cardholders (Hayashi, 2025), and merchant antitrust claims over the networks' parallel adoption of the rules ended in $231.7 million of settlements approved in April 2026 (*B&R Supermarket, Inc. v. Visa, Inc.*, 2026). A liability-shift rule is only as sound as the evidence of which party failed, so an immutable audit trail of mandates, scores and decisions (Section 3.3) is a precondition for this framework.

6.4.Regulatory Gaps and a Proposed KYA Framework

Three gaps stand out. Identity-proofing standards cannot verify an agent separately from its principal (FATF, 2020), and synthetic identity was among the attacks our learned models missed when unseen. Transaction-monitoring thresholds were calibrated for human rhythms and fit machine-speed activity poorly, and Treasury's rulemaking does not yet address non-human initiators (U.S. Department of the Treasury, 2025). No retention standard exists for records of delegated authority and consent. We therefore propose mandatory disclosure of mandate scope at agent registration; a standardized attestation chain extending OAuth 2.0 and OpenID Connect (South et al., 2025a, 2025b); layered monitoring with thresholds calibrated for agent traffic, combining fixed rules, supervised scoring and an anomaly channel; and minimum retention of consent and decision records scaled to transaction value.

6.5.Limitations

The primary results come from a simulator whose attacks we designed, so absolute detection rates reflect how strongly each attack perturbs our features and are not estimates for real agentic fraud. Context features enter as noisy scores that a real system would compute, for example, from text embeddings, with unknown noise. The models score payments one at a time, whereas sequence models might catch scope creep and low-and-slow draining better. The external benchmarks are synthetic, the overlay is label-derived, and the cross-dataset proxies are approximations. Latency excludes feature retrieval, the 3.0% fraud rate exceeds production rates, the liability analysis is not legal advice, and AP2 and x402 are still evolving.

7. Conclusion

We tested whether behavioral monitoring can close the gap between what a mandate authorizes and what an agent does. In a documented simulation evaluated on unseen agent-principal pairs, behavioral scoring raised F1 from 0.303 for mandate checks alone to 0.774, and removing the mandate features lowered it to about 0.63. The autoencoder fused as a feature did not improve on gradient boosting, and both supervised models largely failed on attack types withheld from training, while the autoencoder alone still caught 95% of unseen token replays. On the synthetic PaySim and BAF benchmarks, transfer between datasets was weak, and much of it came from the shared simulated mandate features. Mandate verification, supervised scoring and anomaly detection should therefore run as parallel channels, and claims of generalization should rest on principal-disjoint and attack-disjoint evaluation. For policy, agent identity proofing, monitoring suited to machine-speed activity and durable consent records are the gaps a KYA framework most needs to close. The next step is validation on real agent-initiated payment data as it becomes available.

Data and Code Availability

The simulator, feature pipeline and experiment scripts that produce every table and figure in this paper are available from the corresponding author upon reasonable request. With the seeds listed in Table 3 and the software versions given in Section 4.4, they regenerate the simulated dataset exactly and reproduce the reported results, up to small numerical differences that can arise between hardware and library builds. The PaySim and BAF datasets are publicly available from their original distributors (Lopez-Rojas, 2017; Jesus et al., 2022). No proprietary, personal or institution-specific data were used.

Ethics Statement

This study used only simulated data and publicly released synthetic datasets. It involved no human participants and no personal data, and the simulated mandate and provenance features do not describe any real agent, principal or institution. Any production deployment of a KYA system should be audited for fairness across principal and agent populations before use. Identity-proofing scores and step-up friction can fall more heavily on people with thin credit or banking histories, a concern that the BAF suite was designed to help study (Jesus et al., 2022).

References

B&R Supermarket, Inc. v. Visa, Inc., No. 1:17-cv-02738 (E.D.N.Y. Apr. 28, 2026) (order granting final approval of class settlements). https://fraudliabilityshiftlitigation.com/

BDO USA. (2026, January 9). 2026 fintech industry predictions. https://www.bdo.com/insights/industries/fintech/2026-fintech-industry-predictions

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science, 17*(3), 235–255. https://doi.org/10.1214/ss/1042727940

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Cloudflare. (2025, September 23). *Cloudflare and Coinbase will launch x402 Foundation* [Press release]. https://www.cloudflare.com/press/press-releases/2025/cloudflare-and-coinbase-will-launch-x402-foundation/

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems, 29*(8), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643

Davidovic, S., & Tourpe, H. (2026). *How agentic AI will reshape payments* (IMF Note No. 2026/004). International Monetary Fund. https://doi.org/10.5089/9781513533308.068

Ding, L., Liu, L., Wang, Y., Shi, P., & Yu, J. (2024). An AutoEncoder enhanced light gradient boosting machine method for credit card fraud detection. *PeerJ Computer Science, 10*, e2323. https://doi.org/10.7717/peerj-cs.2323

Financial Action Task Force. (2020). *Guidance on digital identity*. FATF. https://www.fatf-gafi.org/en/publications/Financialinclusionandnpoissues/Digital-identity-guidance.html

Google. (2025). *Agent Payments Protocol (AP2)* [Protocol specification]. https://ap2-protocol.org/

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security* (pp. 79–90). Association for Computing Machinery. https://doi.org/10.1145/3605764.3623985

Guiding and Establishing National Innovation for U.S. Stablecoins Act, Pub. L. No. 119-27, 139 Stat. 419 (2025). https://www.govinfo.gov/app/details/PLAW-119publ27

Hari Vinayak, M. V., & Jarin, T. (2025). A hybrid model for detecting intrusions using stacked autoencoders and extreme gradient boosting. *Computers & Security, 150*, Article 104212. https://doi.org/10.1016/j.cose.2024.104212

Hayashi, F. (2025, February 12). *Did card-present fraud rates decline in the United States after the migration to chip cards?* (Payments System Research Briefing). Federal Reserve Bank of Kansas City. https://www.kansascityfed.org/research/payments-system-research-briefings/did-card-present-fraud-rates-decline-in-the-united-states-after-the-migration-to-chip-cards/

Jesus, S., Pombal, J., Alves, D., Cruz, A., Saleiro, P., Ribeiro, R., Gama, J., & Bizarro, P. (2022). Turning the tables: Biased, imbalanced, dynamic tabular datasets for ML evaluation. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track*. https://proceedings.neurips.cc/paper_files/paper/2022/file/d9696563856bd350e4e7ac5e5812f23c-Paper-Datasets_and_Benchmarks.pdf

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), Article 15. https://doi.org/10.1145/2382577.2382579

Lopez-Rojas, E. A. (2017). *Synthetic financial datasets for fraud detection* [Data set].

Kaggle. https://www.kaggle.com/datasets/ealaxi/paysim1

Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. In *Proceedings of the 28th European Modeling and Simulation Symposium (EMSS 2016)* (pp. 249–255).

https://www.msc-les.org/proceedings/emss/2016/EMSS2016_249.pdf

Mastercard. (2025, April 29). *Mastercard unveils Agent Pay, pioneering agentic payments technology to power commerce in the age of AI* [Press release]. https://www.mastercard.com/us/en/news-and-trends/press/2025/april/mastercard-unveils-agent-pay-pioneering-agentic-payments-technology-to-power-commerce-in-the-age-of-ai.html

Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys, 54*(2), Article 38. https://doi.org/10.1145/3439950

Parikh, S., & Surapaneni, R. (2025, September 16). *Powering AI commerce with the new Agent Payments Protocol (AP2)*. Google Cloud Blog. https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol

Reppel, E., Caspers, R., Leffew, K., Organ, D., Kim, D., & Dalal, N. (2025). *x402: An open standard for internet-native payments* [White paper]. Coinbase. https://www.x402.org/x402-whitepaper.pdf

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F., & Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography, 40*(8), 913–929. https://doi.org/10.1111/ecog.02881

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE, 10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Shanaa, M., & Abdallah, S. (2025). A hybrid anomaly detection framework combining supervised and unsupervised learning for credit card fraud detection (Version 2). *F1000Research, 14*, 664. https://doi.org/10.12688/f1000research.166350.2

South, T., Marro, S., Hardjono, T., Mahari, R., Whitney, C. D., Greenwood, D., Chan, A., & Pentland, A. (2025a). *Authenticated delegation and authorized AI agents* (arXiv:2501.09674). arXiv. https://doi.org/10.48550/arXiv.2501.09674

South, T., Nagabhushanaradhya, S., Dissanayaka, A., Cecchetti, S., Fletcher, G., Lu, V., Pietropaolo, A., Saxe, D. H., Lombardo, J., Maligehalli Shivalingaiah, A., Bounev, S., Keisner, A., Kesselman, A., Proser, Z., Fahs, G., Bunyea, A., Moskowitz, B., Tulshibagwale, A., Greenwood, D., … Pentland, A. (2025b). *Identity management for agentic AI: The new frontier of authorization, authentication, and security for an AI agent world* [White paper]. OpenID Foundation. https://doi.org/10.48550/arXiv.2510.25819

U.S. Department of the Treasury. (2025, September 19). GENIUS Act implementation: Advance notice of proposed rulemaking. *Federal Register, 90*, 45159. https://www.federalregister.gov/documents/2025/09/19/2025-18226/genius-act-implementation

U.S. Payments Forum. (2017). *Understanding the U.S. EMV liability shifts* (Version 2.0) [White paper]. https://www.uspaymentsforum.org/wp-content/uploads/2017/07/EMV-Fraud-Liability-Shift-WP-FINAL-July-2017.pdf