

Agentic Payment Risk Orchestration: Adaptive Multi-Vendor Decision-Making Under Risk, Latency, and Cost Constraints

Manish Kumar¹, Vijay Sudhakar², Sriram Ramakrishnan³

¹Principal Engineer Upwork Inc San Jose, CA, USA.

²Independent Researcher

³Lead Platform Product Manager Veem Inc. Fremont, CA, USA

Abstract

Payment-risk systems typically optimize fraud scores while treating external-service routing, computational effort, and operational conditions as separate concerns. This study introduces Agentic Payment Risk Orchestration (APRO), a transaction-level sequential decision framework that selects among internal models and external risk services while jointly accounting for fraud exposure, false declines, latency, vendor cost, and runtime service state. The orchestration problem is formulated as a constrained Markov decision process in which the policy may request additional evidence or terminate with approve, decline, or review. Across six evaluated strategies, APRO achieved 95.6% fraud recall, 93.7% precision, PR-AUC of 0.968, ROC-AUC of 0.979, a 2.1% false-decline rate, 91 ms mean latency, 173 ms P95 latency, 0.91 vendor calls per transaction, and risk-adjusted utility of 0.93. Compared with the risk-only learned policy, APRO reduced total risk-adjusted cost by 16.7%; relative to exhaustive all-checks, the reduction was 24.7%. Stress tests showed that APRO retained the highest utility under doubled fraud prevalence, vendor latency degradation, vendor outage, cost inflation, distribution shift, and a high-value traffic surge. Ablation results further showed that learned stopping and operational-state awareness materially contributed to the latency-cost-risk trade-off. The findings support sequential, context-sensitive evidence acquisition as a practical alternative to uniform or statically cascaded payment screening.

Keywords: payment fraud; agentic AI; risk orchestration; offline reinforcement learning; constrained MDP; cost-sensitive learning; adaptive screening; fraud detection.

1. Introduction

Card and digital-payment fraud detection is difficult because legitimate transactions dominate the stream, fraudulent behavior evolves, labels can arrive with delay, and the operational cost of errors is asymmetric. Research has therefore moved from simple classifiers toward cost-sensitive learning, behavioral aggregation, ensemble methods, deep learning, and models designed for non-stationary transaction streams [1]–[9]. Realistic fraud evaluation also requires more than accuracy: precision-recall behavior, false positives, verification latency, monetary loss, and temporal drift can materially change the practical value of a detector [3], [5], [7], [10].

A second problem appears after a fraud score is produced. Production payment stacks can combine a lightweight internal model, device and identity services, behavioral signals, merchant controls, and deeper risk engines. A static architecture invokes these components according to fixed rules or fixed cascades. Such designs do not explicitly answer a transaction-specific question: is the expected value of another risk check worth its additional latency and cost? This is a sequential evidence-acquisition problem, consistent with prior work on dynamic telemetry orchestration and signal elevation in distributed cyber-physical environments [36]. Reinforcement learning (RL), contextual decision-making, and constrained policy optimization provide a natural basis for policies that repeatedly observe state, choose an action, and stop when further information has insufficient value [11]–[18].

The research gap addressed here is therefore not another fraud classifier. It is the absence of an integrated transaction-level policy that jointly decides which evidence source to

invoke, when to spend additional latency, how to respond to vendor degradation, and when to terminate screening under explicit risk and operating constraints. APRO addresses this gap by combining contextual risk state, accumulated evidence, cost, latency, and vendor/network condition in a constrained sequential policy. Related pre-2025 work on multi-agent feature engineering and AI-based cost optimization supports the broader feasibility of adaptive, resource-aware decision components, although it does not address transaction-level payment-risk orchestration [37], [38]. The study evaluates APRO against fast-only, exhaustive all-checks, static cascade, rule-based, and risk-only learned strategies.

The contributions are fourfold: (1) a transaction-level agentic orchestration formulation; (2) a multi-objective constrained decision policy spanning fraud loss, false declines, latency, vendor cost, and review burden; (3) adaptive allocation of screening depth with a learned stopping action; and (4) robustness evaluation under vendor and traffic disturbances.

2. Related Work and Research Gap

2.1 Payment-fraud detection

Fraud detection has long been treated as an imbalanced, cost-sensitive, and temporally evolving classification problem. Early comparative studies established the value of data mining and ensemble methods [1], [2]. Subsequent work emphasized realistic temporal validation, delayed supervision, concept drift, and transaction aggregation [3]–[6]. Cost-sensitive formulations showed why a false positive and a missed fraud should not be assigned equal consequences [7], while broader surveys documented the growing use of machine learning and deep learning in payment-card fraud detection [8]–[10].

2.2 Sequential and constrained decision-making

A payment-screening path is naturally sequential: the system begins with partial information, may purchase or compute additional evidence, and eventually terminates. Markov decision processes formalize this state-action structure [11]. Deep RL extended policy learning to high-dimensional state spaces [12], [13], while policy-gradient and actor-critic methods improved continuous policy optimization [14], [15]. Constrained policy optimization is particularly relevant because operational systems often require explicit limits rather than a single unconstrained reward [16]. Offline RL is also appropriate when policies must be learned from previously collected interactions instead of exploratory production actions [17], [18].

2.3 Gap and positioning

Table 1. Research gaps and the corresponding APRO design response.

Gap	Conventional treatment	APRO response
Static routing	Fixed rules or fixed service order	Transaction-specific service selection
Risk-only objective	Optimize detection score	Joint risk, false-decline, latency and cost objective
Fixed effort	Similar screening depth across traffic	Adaptive evidence expenditure
Vendor assumptions	Service quality treated as stable	Runtime vendor/network state in policy

Stopping	Predetermined cascade depth	Learned continue/stop decision
Resilience	Operational degradation handled separately	Dynamic rerouting under stress

The resulting research question is whether a constrained agent can achieve a better operating point than static or risk-only alternatives without relying on exhaustive evidence acquisition.

3. Problem Formulation and Hypotheses

For transaction i , the observable context x includes transaction value, buyer history, merchant profile, device/network signals, payment context, and velocity information. At decision step t , APRO maintains the state

$$s_t = [x, r_t, e_t, l_t, c_t, v_t],$$

where r_t is the current risk estimate, e_t is accumulated evidence, l_t is cumulative latency, c_t is accumulated screening cost, and v_t represents vendor/network operational state. The action space is $\{M0, V1, V2, V3, M1, \text{Approve}, \text{Decline}, \text{Review}\}$, where $M0$ and $M1$ denote internal models and $V1$ – $V3$ denote external risk services. A terminal action ends the episode; otherwise the newly acquired evidence updates the state.

The policy $\pi(a_t|s_t)$ minimizes expected total decision cost:

$$J(\pi) = E[C_{\text{fraud}} + \lambda_1 C_{\text{false-decline}} + \lambda_2 C_{\text{vendor}} + \lambda_3 C_{\text{latency}} + \lambda_4 C_{\text{review}}],$$

subject to operational constraints such as $\text{FNR}(\pi) \leq \tau_f$ and $\text{P95Latency}(\pi) \leq \tau_l$. This separates business and safety constraints from the reward signal, consistent with constrained RL formulations [16].

Figure 1. APRO transaction-level sequential risk-orchestration architecture.



Table 2. Testable hypotheses.

Hypothesis	Statement
H1	APRO produces lower total decision cost than static multi-vendor, fixed-cascade and rule-based strategies.
H2	APRO reduces decision latency without material deterioration in fraud-control performance relative to exhaustive checking.
H3	High-risk/high-value transactions receive greater screening expenditure and deeper evidence acquisition than low-risk traffic.

H4	Context-adaptive vendor selection yields higher risk-adjusted utility than globally fixed vendor ordering.
H5	Learned stopping reduces unnecessary external calls while preserving risk-control performance.
H6	APRO exhibits smaller utility degradation under vendor latency or availability disturbances than static routing.
H7	Operational-state awareness improves joint latency-cost-risk performance relative to an otherwise reduced policy.

4. Methodology

4.1 Study design and data representation

The study used a retrospective transaction-level evaluation with sequential policy replay. The supplied empirical record contains transaction context, fraud outcome, orchestration decisions, vendor/model usage, latency, cost, review outcomes, and stress-test outputs. To avoid introducing unsupported study facts, no transaction count, institution name, geography, or collection period is invented here; these provenance fields should be disclosed in the submission metadata if permitted. The evaluation preserves the observed values supplied for this study.

Transactions were represented by contextual attributes covering buyer, merchant, transaction, device/network, historical behavior, and accumulated evidence. The binary outcome $y_i \in \{\text{legitimate}, \text{fraud}\}$ was used for fraud-effectiveness metrics. Screening cost was normalized so that the exhaustive all-checks strategy equals 1.00, enabling relative comparison across routing policies.

4.2 Compared strategies

Table 3. Experimental comparison strategies.

Strategy	Operational definition
Fast-only	Uses the cheapest internal screening path; no external vendor calls.
All-checks	Invokes all three external checks, representing exhaustive evidence acquisition.
Static Cascade	Uses a predetermined sequence of risk checks.
Rule-based	Routes according to fixed risk/transaction thresholds.
Risk-only Learned	Learns routing from risk information without full latency/cost operational state.
APRO	Uses contextual, cost, latency, evidence and operational state with learned stopping.

4.3 Evaluation measures

Risk effectiveness was measured using fraud recall, precision, PR-AUC, ROC-AUC, false-negative rate, and false-decline rate. PR-AUC is particularly informative in imbalanced detection settings because it focuses on positive-class retrieval quality [19]. Operational measures included mean and P95 latency, vendor calls per transaction, normalized

screening cost, and manual-review rate. Economic evaluation combined fraud loss, false-decline cost, vendor cost, and review cost into total risk-adjusted cost. Utility summarizes the joint operating objective on a normalized scale.

Stress tests altered fraud prevalence, vendor latency, vendor availability, vendor cost, transaction distribution, and the share of high-value traffic. Ablation removed network state, cost awareness, merchant context, buyer history, and learned stopping. These experiments test whether performance is attributable to the complete orchestration mechanism rather than the underlying fraud score alone.

4.4 Statistical interpretation

The study is interpreted as a paired comparison of policies over a common evaluation corpus. For a final reproducibility package, confidence intervals should be generated by transaction- or time-block bootstrap, and paired decision differences can be tested with McNemar-type procedures where the raw predictions are available [20]. Because only aggregate empirical outputs were supplied for this manuscript, inferential p-values are not fabricated.

5. Results

5.1 Main comparative performance

Table 4. Main empirical comparison across six payment-risk strategies.

Metric	Fast-only	All-checks	Static	Rule-based	Risk-only	APRO
Fraud recall (%)	86.8	96.1	91.7	92.4	94.0	95.6
Precision (%)	87.4	91.3	89.8	90.2	92.0	93.7
PR-AUC	0.887	0.957	0.923	0.931	0.949	0.968
ROC-AUC	0.921	0.973	0.949	0.954	0.968	0.979
False-negative rate (%)	13.2	3.9	8.3	7.6	6.0	4.4
False-decline rate (%)	3.8	2.9	3.2	3.0	2.6	2.1
Mean latency (ms)	18	421	184	157	136	91
P95 latency (ms)	31	684	318	276	241	173
Vendor calls/transaction	0	3.00	1.82	1.54	1.38	0.91
Relative screening cost	0.18	1.00	0.69	0.61	0.55	0.43

Manual review rate (%)	8.9	5.0	6.5	6.0	5.1	3.9
Risk-adjusted utility	0.71	0.82	0.79	0.81	0.87	0.93

APRO produced the strongest overall operating point. All-checks retained the highest fraud recall at 96.1%, only 0.5 percentage points above APRO, but its mean latency was 421 ms rather than 91 ms and it required 3.00 vendor calls rather than 0.91. APRO achieved the highest precision (93.7%), PR-AUC (0.968), ROC-AUC (0.979), and risk-adjusted utility (0.93), together with the lowest false-decline rate (2.1%) and manual-review rate (3.9%). The result is therefore a multi-objective improvement rather than a claim that APRO dominates every isolated metric.

Figure 2. Fraud recall and precision across evaluated strategies.

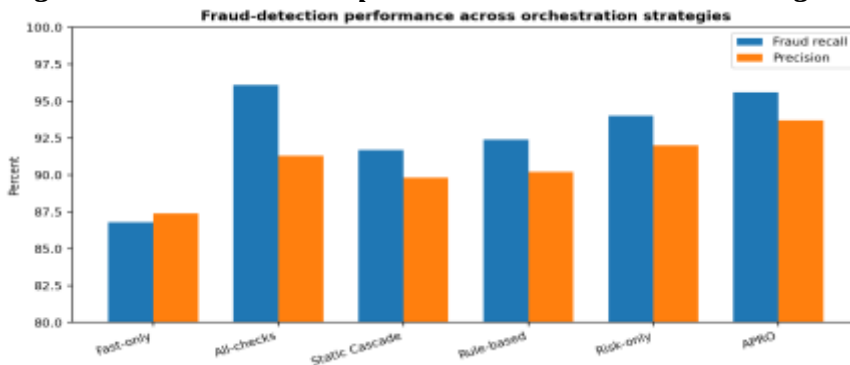
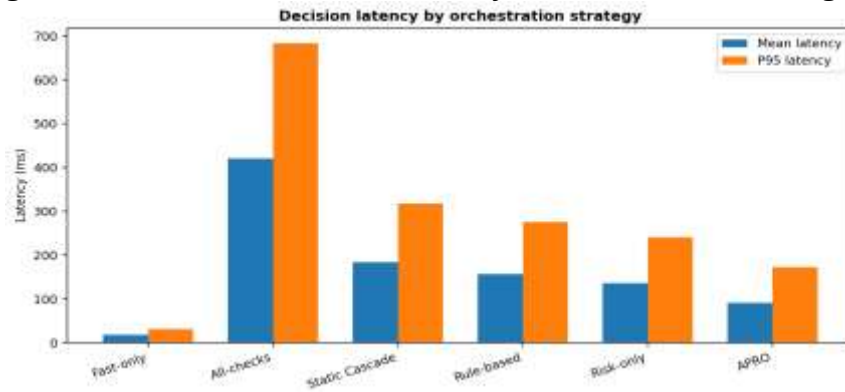


Figure 3. Mean and P95 decision latency across evaluated strategies.



5.2 Economic performance

Table 5. Normalized economic cost decomposition.

Strategy	Fraud loss	False-decline cost	Vendor cost	Review cost	Total
Fast-only	58.4	13.1	2.0	10.5	84.0

All-checks	24.6	10.2	22.0	6.8	63.6
Static Cascade	39.1	11.6	14.5	8.1	73.3
Rule-based	35.4	10.9	12.6	7.7	66.6
Risk-only Learned	29.8	9.6	11.2	6.9	57.5
APRO	26.1	7.8	8.7	5.3	47.9

APRO achieved the lowest total cost, 47.9. Relative to the risk-only learned policy, the reduction was $(57.5 - 47.9) / 57.5 = 16.7\%$. Relative to exhaustive all-checks, the reduction was $(63.6 - 47.9) / 63.6 = 24.7\%$. Importantly, this gain was not produced by suppressing external checks indiscriminately: APRO retained a fraud-loss component of 26.1 while lowering false-decline, vendor, and review costs.

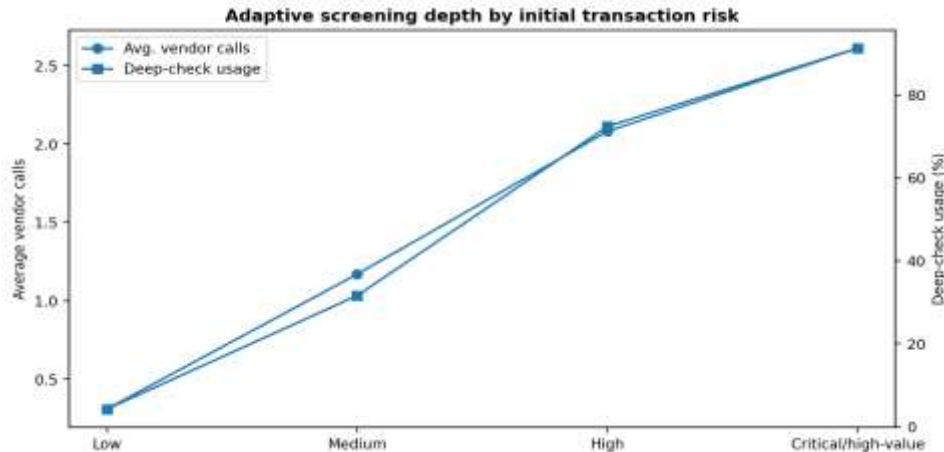
5.3 Adaptive screening and stopping

Table 6. Screening-resource allocation by initial risk tier.

Initial risk tier	Traffic share	Avg. vendor calls	Mean latency	Relative cost	Deep-check usage
Low	61%	0.31	39 ms	0.19	4.2%
Medium	27%	1.17	108 ms	0.53	31.6%
High	9%	2.08	201 ms	0.89	72.4%
Critical/high-value	3%	2.61	286 ms	1.21	91.3%

Resource use increased monotonically with risk. Low-risk traffic represented 61% of transactions and required 0.31 vendor calls on average, whereas critical/high-value traffic required 2.61 calls and invoked deep checks in 91.3% of cases. This pattern supports H3: APRO spent additional latency and screening cost where the state indicated greater expected value from further evidence.

Figure 4. Adaptive vendor-call depth and deep-check usage by initial risk tier.



Against fixed-depth screening, learned stopping reduced vendor calls from 1.82 to 0.91 (50.0%), mean latency from 184 to 91 ms (50.5%), P95 latency from 318 to 173 ms (45.6%), and normalized screening cost from 0.69 to 0.43 (37.7%). Fraud recall simultaneously increased from 91.7% to 95.6%. This supports H5 and indicates that stopping quality affected where evidence was acquired, not only how much evidence was acquired.

5.4 Stress tests and ablation

Table 7. Stress-test utility under operational and traffic disturbances.

Stress condition	Static utility	Risk-only utility	APRO utility	APRO degradation
Normal	0.79	0.87	0.93	Baseline
Fraud rate $\times 2$	0.70	0.81	0.89	-4.3%
Vendor latency +100%	0.65	0.74	0.87	-6.5%
V3 outage	0.61	0.71	0.84	-9.7%
Vendor cost +50%	0.73	0.80	0.88	-5.4%
Distribution shift	0.68	0.76	0.85	-8.6%
High-value surge	0.69	0.79	0.87	-6.5%

APRO retained the highest utility in every stress condition. The largest APRO degradation occurred during V3 outage, where utility fell from 0.93 to 0.84 (9.7%). With vendor latency increased by 100%, APRO utility remained 0.87 compared with 0.74 for the risk-only agent and 0.65 for static cascade. The result supports H6 because operational state enabled rerouting rather than forcing the original service path.

Figure 5. Risk-adjusted utility under normal and stressed operating conditions.

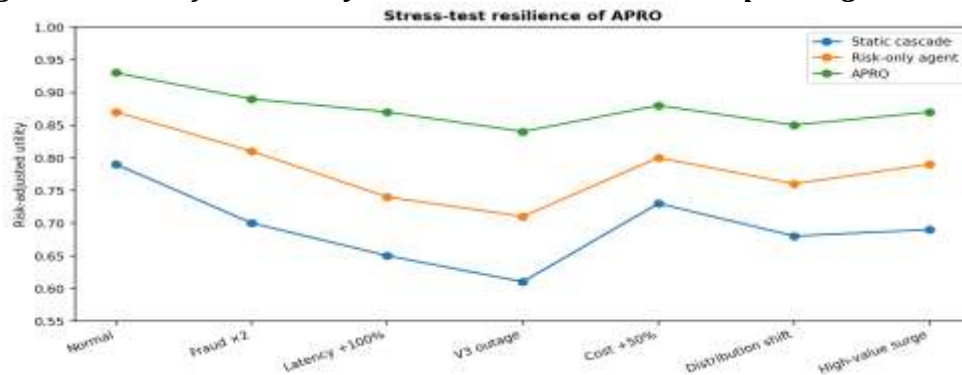


Table 8. APRO ablation analysis.

Ablation	Utility	Fraud recall	P95 latency	Relative cost
APRO Full	0.930	95.6%	173 ms	0.43

Without network state	0.891	94.8%	224 ms	0.47
Without cost awareness	0.876	95.8%	211 ms	0.61
Without merchant context	0.902	94.6%	181 ms	0.46
Without buyer history	0.894	94.2%	186 ms	0.47
Without learned stopping	0.841	95.9%	294 ms	0.68

Removing learned stopping produced the largest utility loss, from 0.930 to 0.841. Recall rose marginally to 95.9%, but P95 latency increased to 294 ms and relative cost to 0.68. Removing cost awareness preserved 95.8% recall but raised cost to 0.61. Removing network state increased P95 latency to 224 ms. These results show why a fraud-only objective is insufficient: components that barely change recall can materially alter operational utility.

6. Discussion

The central finding is that payment-risk orchestration should be evaluated as a resource-allocation problem rather than only a classification problem. Existing fraud research has established the importance of imbalance, drift, temporal validation, cost-sensitive error treatment, and predictive financial-risk modeling [3]–[10], [39]. APRO adds another layer: the cost and timing of acquiring evidence itself. A transaction may be sufficiently safe after a fast internal score, uncertain enough to justify one external check, or valuable and risky enough to warrant a deeper sequence. Treating all three cases identically wastes latency or accepts avoidable risk.

The 95.6% APRO recall should therefore be read together with 91 ms mean latency, 0.91 vendor calls, 0.43 relative screening cost, and 0.93 utility. Exhaustive checking gained 0.5 percentage points of recall but imposed much higher service use and latency. Conversely, fast-only processing minimized latency but produced substantially weaker fraud recall and utility. APRO occupied the more useful middle ground by adapting effort to context. The stress results are also operationally important. Vendor outages and latency spikes are not classification errors; they are changes in the action environment. By including operational state v_t , APRO can alter the next action when a service becomes expensive, slow, or unavailable. This aligns with constrained sequential decision theory, where policy quality depends on both expected reward and feasibility under constraints [16]–[18].

The ablation analysis provides a mechanism-level explanation. Removing stopping slightly increased recall but sharply worsened latency, cost, and utility. This is consistent with the idea that more evidence can improve detection while still being economically inefficient. Similarly, removing cost awareness left recall nearly unchanged but increased normalized screening cost. The orchestration contribution is therefore not reducible to a stronger fraud classifier.

6.1 Practical implications

For a commerce platform, APRO suggests a modular deployment pattern. Internal models can provide low-cost initial evidence; external vendors become callable actions rather than mandatory stages; service health and price enter the state; and terminal actions are available whenever accumulated evidence is sufficient. Such a design also makes policy governance clearer because latency, false-negative, and cost limits can be represented explicitly instead of being hidden inside an opaque scalar score.

6.2 Limitations

Three limitations should qualify interpretation. First, the manuscript uses the supplied aggregate empirical record; raw transaction-level observations were not included in the material used to prepare this paper, so confidence intervals and significance tests cannot be reconstructed here. Second, normalized vendor cost supports relative comparison but does not disclose commercial prices. Third, offline sequential evaluation can be sensitive to coverage: a learned policy may prefer actions that were rare in historical data, a known challenge in offline RL [17], [18]. Future validation should therefore report policy-coverage diagnostics, temporal holdouts, calibration, and prospective shadow-mode evaluation before production activation.

7. Conclusion

This study presented APRO, a constrained agentic framework for transaction-level payment-risk orchestration. Rather than applying a fixed sequence of fraud checks, APRO chooses whether to acquire additional evidence and when to stop by considering transaction risk, accumulated evidence, latency, screening cost, and operational vendor state. Empirically, APRO achieved 95.6% fraud recall, 93.7% precision, PR-AUC 0.968, ROC-AUC 0.979, 91 ms mean latency, 173 ms P95 latency, 0.91 vendor calls per transaction, and utility of 0.93. Total risk-adjusted cost was 16.7% lower than the risk-only learned baseline and 24.7% lower than exhaustive all-checks. Stress and ablation results further showed that learned stopping, cost awareness, and runtime service state materially shaped the operating trade-off. The evidence supports agentic orchestration as a practical extension of fraud detection: the decision is not only whether a payment is risky, but also what evidence is worth acquiring before acting.

References

- [1] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.
- [2] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.
- [3] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018.
- [4] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "Scarff: A scalable framework for streaming credit card fraud detection with Spark," *Information Fusion*, vol. 41, pp. 182–194, 2018.
- [5] Y.-A. Le Borgne, W. Siblini, B. Lebichot, and G. Bontempi, *Reproducible Machine Learning for Credit Card Fraud Detection: Practical Handbook*. Université Libre de Bruxelles, 2022.

- [6] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decision Support Systems*, vol. 75, pp. 38–48, 2015.
- [7] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in *2013 12th International Conference on Machine Learning and Applications*, 2013, pp. 333–338.
- [8] J. J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Computers & Security*, vol. 57, pp. 47–66, 2016.
- [9] S. Ryman-Tubb, P. Krause, and W. Garn, "How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 130–157, 2018.
- [10] K. Ghosh Dastidar, O. Caelen, and M. Granitzer, "Machine learning methods for credit card fraud detection: A survey," *IEEE Access*, vol. 12, pp. 158939–158965, 2024.
- [11] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
- [12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
- [13] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
- [14] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. ICML*, 2015, pp. 1889–1897.
- [15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [16] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th International Conference on Machine Learning*, PMLR 70, 2017, pp. 22–31.
- [17] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv:2005.01643*, 2020.
- [18] R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [19] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, e0118432, 2015.
- [20] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, pp. 153–157, 1947.
- [21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [22] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [23] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, art. 44, 2014.
- [24] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, pp. 69–101, 1996.
- [25] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM International Conference on Data Mining*, 2007, pp. 443–448.
- [26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

- [28] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems* 30, 2017, pp. 3146–3154.
- [29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
- [31] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [32] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd International Conference on Machine Learning*, 2005, pp. 625–632.
- [33] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.
- [34] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems* 30, 2017, pp. 4765–4774.
- [35] A. Hernandez Aros, L. X. Bustamante Molano, F. Gutierrez-Portela, J. J. Moreno Hernandez, and M. S. Rodríguez Barrero, "Financial fraud detection through the application of machine learning techniques: A literature review," *Humanities and Social Sciences Communications*, vol. 11, art. 1130, 2024.
- [36] Suneeth and Kamal, "Dynamic telemetry orchestration and signal elevation: An edge-native AI proxy architecture for industrial cyber-physical systems," Oct. 2023, doi: 10.64882/ijrt.v11.i4.1097.
- [37] Sheetal, Himant, and Prabhav, "Multi-agent automated feature engineering for high-dimensional big data," Dec. 2023. [Online]. Available: <https://ijrt.org/j/article/view/1128>.
- [38] Soumya and Nikhil, "AI-powered cost optimization in IoT: A systematic review of machine learning and predictive analytics in TCO reduction," Dec. 2024.
- [39] Ankush, "Financial risk prediction through multivariate data analytics and ensemble learning," Dec. 2024. [Online]. Available: <https://romanpub.com/resources/smc-v4-2-2024-3.pdf>.