
From Research to Practice: Operational Deployment of Temporal Drift Detection in Financial Fraud Prevention

Thabo Fathy¹, Karim Daniels²

¹The University of Jordan, JORDAN

²United Arab Emirates University, UNITED ARAB EMIRATES

ABSTRACT

Temporal drift detection is well established in research, but fraud teams that deploy it face practical questions that research benchmarks rarely answer: how often the tests will raise false alarms, how large a shift they can see, whether they detect changes in fraud itself, and how quickly they can respond when confirmed labels arrive late. This study takes the temporal-drift framework of Zuberi et al. [1] Population Stability Index (PSI), Kolmogorov–Smirnov (KS) and Jensen–Shannon divergence (JSD) tests on monitored features, combined with prevalence-shift z-tests — and evaluates it as an operational system on a simulated mobile-money transaction stream. Four controlled experiments examined false-alarm rates on stable data, detection power for population shifts, sensitivity to drift confined to fraudulent transactions, and supervised detection delay under label delays of 0–28 days. KS significance tests were unsuitable as stand-alone alarms: under ordinary week-to-week variation, the share of monitoring windows with a KS $p < 0.001$ alarm rose from 12% to 95% as window size grew from 1,000 to 80,000 transactions. PSI thresholds and the combined rule of Zuberi et al. produced almost no false alarms, but detected a shift in transaction amount only once it approached half a standard deviation. Feature-level tests missed a new fraud pattern until fraudulent transactions reached about 20% of traffic, whereas the prevalence z-test detected an accompanying rise in fraud rate in 81% of weeks at 1% prevalence. Supervised detection delay was dominated by label delay: a sequential Page–Hinkley test reached detection one day after labels arrived, against five days for a weekly test, but raised a false alarm in 41% of 120-day stable periods against 5%. The results translate into a deployment lifecycle — calibration, shadow mode, tiered response and periodic review — and into operating rules that treat feature drift as information, prevalence and performance shifts as triggers, and label latency as the main lever for faster response.

Keywords: Concept Drift; Fraud Prevention; Drift Detection; Population Stability Index; False Alarms; Label Delay; Model Monitoring; MLOps

Introduction

Machine-learning fraud models are deployed into environments that change continuously. Fraudsters adapt to controls, customers adopt new payment channels, and the volume and mix of transactions follow economic and seasonal cycles [2]–[4]. The resulting concept drift degrades detection unless it is monitored and acted upon [5], [6]. Research has produced a large toolbox of drift detectors [7]–[9], and recent fraud-specific work has shown how these can be combined into a rigorous measurement framework [1].

The step from such a framework to a running operational system is not trivial. Engineering studies of production machine learning have shown that most of the work and risk sits in monitoring, data dependencies and feedback loops rather than in the model itself [10], [11]. For drift detection specifically, a monitoring team must decide which statistics to compute, on how many transactions, at which thresholds and how often; how to handle the delay before investigated fraud labels are available [12]; and what to do when an alarm fires. A detector with a high false-alarm rate erodes trust and wastes analyst time; a detector with low power

gives false reassurance; and in regulated institutions every retraining it triggers must be validated and approved [13].

This study addresses that research-to-practice gap. We take the temporal-drift framework of Zuberi *et al.* [1], which was developed and evaluated on a 48-hour card-transaction benchmark, and examine how its components behave when deployed as continuous monitoring. The contributions are:

- 1.a calibration of false-alarm rates for PSI, KS, JSD and the combined rule of the adopted framework [1] under both stationary data and realistic week-to-week variation, across monitoring windows of 1,000 to 80,000 transactions;
- 2.power curves showing the size of population shift each statistic can detect at each window size;
- 3.evidence on the detectability of drift confined to fraudulent transactions, comparing feature-level tests with the prevalence-shift z-test of the framework;
- 4.an analysis of supervised detection delay and false alarms for weekly, daily and sequential tests under label delays of 0–28 days, with an cost model;
- 5.a deployment lifecycle and tiered response playbook derived from these results.

Background

Drift statistics used in practice

Three statistics dominate operational drift monitoring. The Population Stability Index compares binned distributions and is widely used in credit and fraud model governance, with conventional thresholds of 0.10 for a moderate and 0.25 for a major shift [14]. The two-sample Kolmogorov–Smirnov test measures the largest gap between empirical distribution functions and provides a p-value [15]. The Jensen–Shannon divergence is a bounded, symmetric measure of the difference between two distributions [16]. Supervised monitors instead track model errors over time, using sequential tests such as CUSUM and Page–Hinkley [17]–[19] or windowed tests such as DDM and ADWIN [7], [8].

A well-known difficulty with significance tests on large data is that, as sample size grows, trivially small differences become statistically significant [20], [21]. In drift monitoring this matters because transaction volumes are large and because real streams are never perfectly stationary: day-of-week mix, promotions and minor product changes produce small, harmless shifts. Rabanser *et al.* [9] showed in a broad empirical study that the choice of test and its sample size strongly affects whether shift is detected, and that univariate tests on informative features are often competitive with more complex methods.

The temporal-drift framework adopted in this study

Zuberi *et al.* [1] proposed a robust machine-learning framework for detecting temporal drift in financial fraud prevention. The framework enforces a chronological train–validation–test split with all preprocessing fitted on training data only, selects features by SHAP importance [22], measures drift between partitions with PSI, KS and JSD on the selected features, tests for prevalence shift with z-tests, and tunes decision thresholds to maximise F2. On the European card benchmark it found measurable but modest drift and stressed that, because the data covered only 48 hours, its drift estimates were a lower bound for production settings [1]. This study deploys the framework’s drift battery and prevalence test as continuous monitors and asks how they behave over operational time scales. For the combined alarm rule, a feature is flagged when $\text{PSI} \geq 0.10$ with $\text{KS } p < 0.001$, or when $\text{PSI} \geq 0.25$, and an alarm requires at least two flagged features.

Methods

Data and monitored features

Experiments used a simulated mobile-money transaction stream in the tradition of PaySim [23]. Each transaction is described by ten features, including log amount, transfer and cash-out flags, the share of the sender's balance transferred, whether the destination account is new, the destination's 24-hour in-degree, hour of day, sender account age and the sender's 24-hour transaction count. Legitimate transactions follow stable distributions; two fraud patterns are simulated — account-takeover cash-outs and a mule-network pattern of smaller transfers to new, high-in-degree accounts. Six features were monitored: the six with the largest SHAP importance in a gradient-boosting model trained on the stream, following the feature-selection step of the adopted framework [1].

“Benign variation” was modelled by redrawing, for every monitoring window, small random perturbations of the legitimate population: a shift in mean log amount with standard deviation 0.03, and shifts of about one percentage point in the transfer share and night-time share. These changes are harmless to a fraud model but are typical of week-to-week variation in real payment traffic.

Experiments

Table 1. Experimental design

Experiment	Question	Design	Replications
E1 False alarms	How often do tests fire when nothing harmful has changed?	Reference and current windows of 1,000–80,000 transactions; stationary vs benign variation; 6 monitored features	300 per cell
E2 Power	How large a population shift can each test detect?	Shift of 0–0.5 SD in log amount; windows of 1,000, 5,000 and 20,000	150 per cell
E3 Minority drift	Do tests detect a change in fraud itself?	60% of fraud switches to the mule pattern and fraud prevalence rises 1.5-fold; fraud share 0.1–20%; 10,000 per window	150 per cell
E4 Supervised delay	How fast do label-based monitors detect a drop in detection?	Recall of the deployed model falls from 0.85 to 0.46 at day 120; about 9 labelled frauds per day; label delay 0–28 days; weekly, daily and Page–Hinkley tests	300 per cell

In E4 the weekly and daily tests are one-sided z-tests ($z > 3$) of the observed fraud-capture rate against the pre-drift rate of 0.85, computed on non-overlapping weekly windows or on a 7-day window evaluated daily. The Page–Hinkley test accumulates, fraud by fraud, the difference between the expected and observed capture rate (allowance $\delta = 0.05$, threshold $\lambda = 6$) [17], [18]. A detection before day 120 was counted as a false alarm. Detection delay is measured from drift onset and includes the label delay, since a drop cannot be observed before the labels arrive [12].

Results

False alarms on stable data

On truly stationary data (Figure 1a), only the conventional KS test at $p < 0.05$ raised frequent alarms — in 18%–23% of windows, the expected consequence of testing six features at a 5% level — and the KS effect-size rule $D \geq 0.05$ fired in 47% of the smallest windows, where sampling noise alone produces gaps of that size. PSI thresholds, JSD and the combined rule of the adopted framework produced almost no alarms.

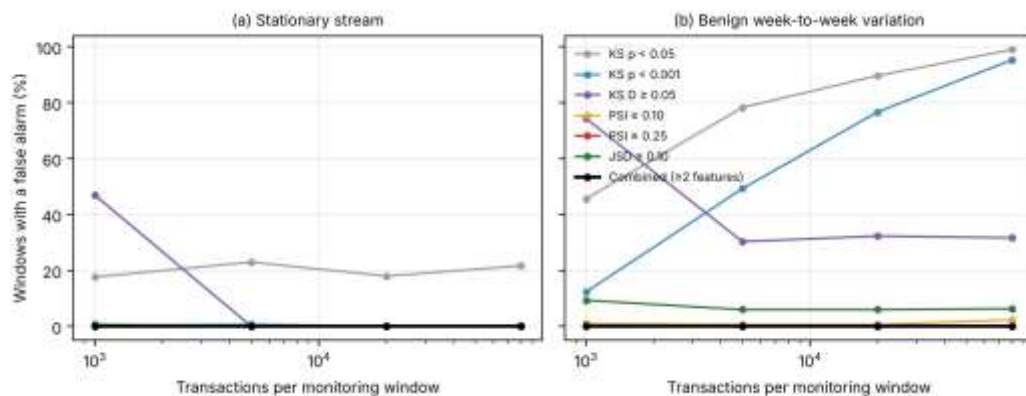


Figure 1. Share of monitoring windows with at least one false alarm, by rule and window size, on (a) a stationary stream and (b) a stream with benign week-to-week variation.

Benign variation changed the picture for significance tests (Figure 1b, Table 2). The share of windows in which $KS\ p < 0.001$ raised an alarm climbed from 12% at 1,000 transactions to 49% at 5,000, 77% at 20,000 and 95% at 80,000. Larger windows make KS more sensitive to harmless shifts — the large-sample p-value problem [20] in operational form. In contrast, $PSI \geq 0.10$ fired in at most 2.3% of windows, $PSI \geq 0.25$ never fired, $JSD \geq 0.10$ in 6%–9%, and the combined rule of Zuberi *et al.* [1] — which requires both a meaningful PSI and a significant KS result on at least two features — produced no false alarms in any configuration.

Table 2. False-alarm rate per monitoring window under benign variation, by window size (transactions per window)

Rule	1,000	5,000	20,000	80,000
$KS\ p < 0.05$	45.7%	78.3%	89.7%	99.0%
$KS\ p < 0.001$	12.3%	49.3%	76.7%	95.3%
$KS\ D \geq 0.05$	74.3%	30.3%	32.3%	31.7%
$PSI \geq 0.10$	1.0%	0.7%	0.7%	2.3%
$PSI \geq 0.25$	0.0%	0.0%	0.0%	0.0%
$JSD \geq 0.10$	9.3%	6.0%	6.0%	6.3%
Combined (≥ 2 features)	0.0%	0.0%	0.0%	0.0%

Note. An alarm is counted when any of the six monitored features meets the rule (the combined rule requires at least two features).

Detection power

Figure 2 shows the other side of the trade-off. With 20,000 transactions per window, $KS\ p < 0.001$ detected a shift of only 0.05 standard deviations in log amount in 81% of trials — a change too small to matter for most fraud models. $PSI \geq 0.10$ needed a shift of about 0.5 SD to fire reliably (100% at 20,000 transactions; 12% at 0.3 SD with 5,000), and $PSI \geq 0.25$ detected even 0.5 SD in only 51% of small windows. $JSD \geq 0.10$

sat between the two, detecting a 0.3 SD shift in 99% of trials. Unlike KS p-values, PSI and JSD behaved almost identically across window sizes once windows exceeded a few thousand transactions, which makes their thresholds portable between portfolios of different volume.

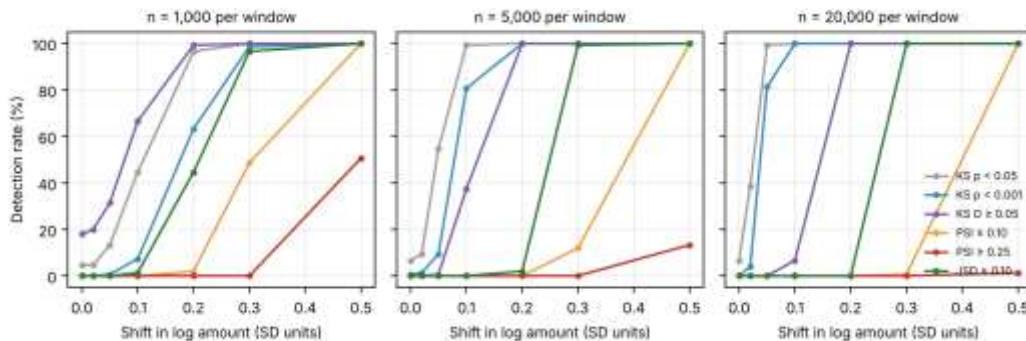


Figure 2. Detection rate as a function of the size of a population shift in log amount, for three window sizes.

Drift confined to fraudulent transactions

Figure 3 addresses the question that matters most for fraud prevention: whether the tests detect a change in fraud itself. When 60% of fraud switched to the mule-network pattern, the feature battery did not raise an alarm in any trial at fraud shares from 0.1% to 10% of traffic. The largest single-feature PSI — always on destination in-degree, the feature that characterises the new pattern — was 0.018 at 1% fraud and 0.050 at 5%, rose to 0.12 at 10% and exceeded the alarm rule only at a 20% fraud share (mean PSI 0.31, two features flagged on average). Real fraud rates are one to two orders of magnitude lower [12].

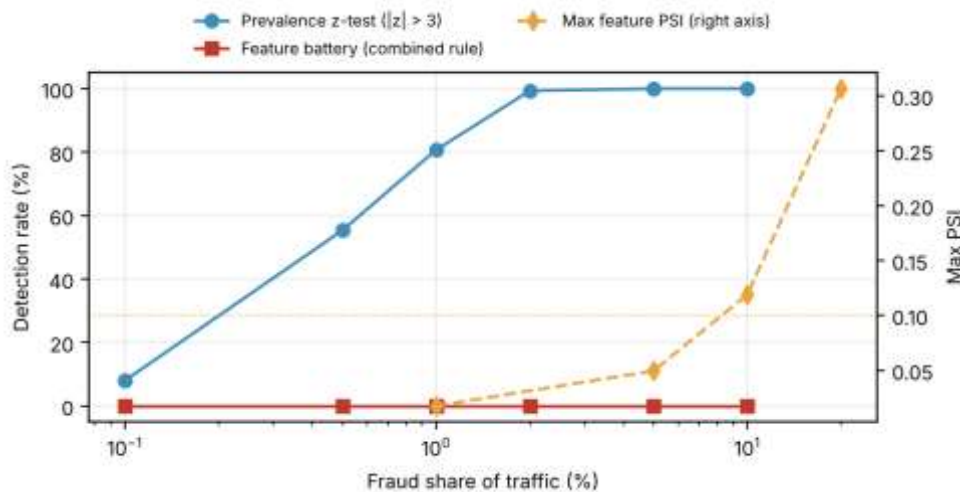


Figure 3. Detection of a new fraud pattern by the feature battery and by the prevalence z-test, as a function of the fraud share of traffic; dashed line shows the largest single-feature PSI.

The prevalence z-test of the adopted framework [1], applied once labels were available, behaved very differently. A 1.5-fold rise in fraud rate was detected in 8% of weeks at 0.1% prevalence, 55% at 0.5%, 81% at 1% and 99% at 2%. Its power is limited only by the number of fraud cases per window, so it improves with volume and with longer reference periods. Where a new fraud pattern also changes how much fraud occurs, the label-based prevalence test is therefore a far more sensitive alarm than any feature-level statistic.

Supervised detection delay

Figure 4 and Table 3 report how fast label-based monitors detected a fall in the model's fraud-capture rate from 0.85 to 0.46. With immediate labels, the sequential Page–Hinkley test detected the drop after a median of 1 day, the daily 7-day test after 2 days and the weekly test after 5 days. With a 14-day label delay these became 15, 16 and 19 days, and with 28 days 29, 30 and 33 days. In every case the label delay accounted for most of the total: the choice of test changed detection by at most four days, whereas the label delay changed it by four weeks.

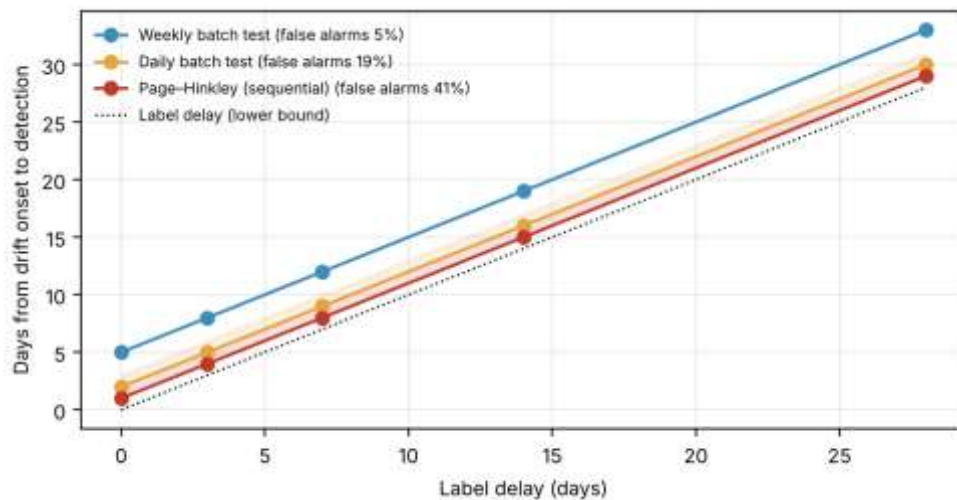


Figure 4. Median days from drift onset to detection (shaded to the 90th percentile) by label delay for three supervised monitoring tests; legend shows the share of 120-day stable periods with a false alarm.

Speed came at the price of false alarms. Over a 120-day stable period before the drift, the weekly test raised a false alarm in 5% of replications, the daily test in 19% and the Page–Hinkley test in 41%. Evaluating a 7-day window every day multiplies the number of tests, and a sequential test tuned for fast response accumulates noise from individual fraud outcomes.

Table 3. Days from drift onset to supervised detection: median (90th percentile)

Label delay (days)	Weekly batch test	Daily batch test	Page–Hinkley (sequential)
0	5 (5)	2 (3)	1 (2)
3	8 (8)	5 (6)	4 (5)
7	12 (12)	9 (10)	8 (9)
14	19 (19)	16 (17)	15 (16)
28	33 (33)	30 (31)	29 (30)
False alarms per 120 stable days	5%	19%	41%

Cost of Monitoring Choices

To show how these trade-offs combine, we computed an expected annual cost for each supervised test. The calculation assumes three harmful drift events a year, an average loss of US\$300 per missed fraud, and the extra 3.5 frauds per day missed while the capture rate is degraded, plus a fixed cost per false alarm for investigation. These parameters are assumptions for illustration, not estimates. When a false alarm costs US\$2,000, the faster tests are cheaper at every label delay (for example 6 versus 16 thousand dollars a year with immediate labels). When a false alarm costs US\$20,000 — closer to the cost of a full retraining and

validation cycle — the Page–Hinkley test becomes the most expensive option (31 versus 20 thousand dollars with immediate labels). Under both assumptions, expected cost rises steeply with label delay (Figure 5).

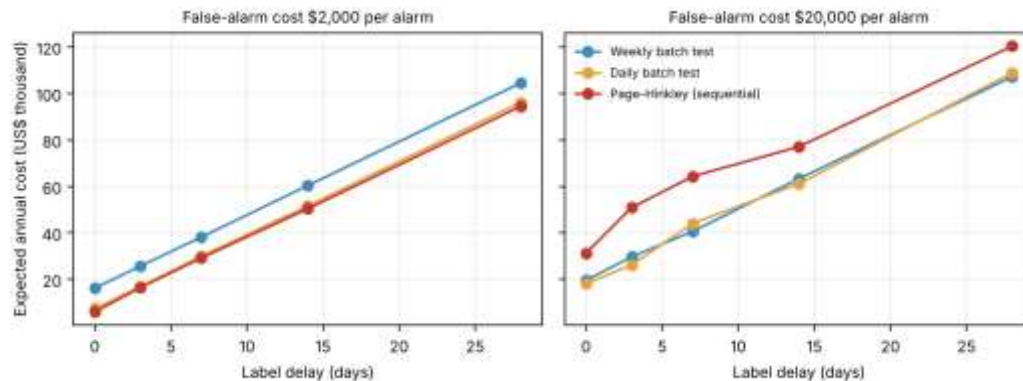


Figure 5. expected annual cost of three supervised monitoring tests by label delay, for a low and a high cost per false alarm (assumptions stated in the text).

From Research to Practice: A Deployment Playbook

The results support a staged deployment of the adopted framework (Figure 6). In the calibration stage, each statistic is run on several months of historical data believed to be stable, to measure its false-alarm rate at the intended window size before any threshold is set; this study shows that the answer depends strongly on the statistic and on volume. In shadow mode, the monitors run in production and their alarms are logged and reviewed without triggering action, so that thresholds can be adjusted to the institution's own benign variation. Only then are alarms connected to a tiered response.

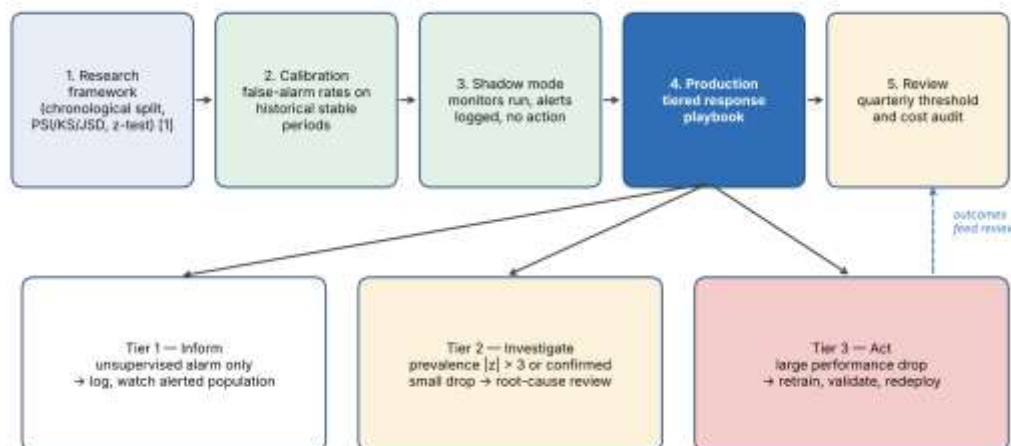


Figure 6. Deployment lifecycle for the adopted temporal-drift framework, with a three-tier response playbook.

Table 4. Recommended operating configuration derived from the experiments

Component	Recommended setting	Evidence in this study
Feature drift statistic	PSI and JSD with effect-size thresholds; KS used only together with PSI, never alone	KS p-value alarms rise to 95% under benign variation (Table 2)

Component	Recommended setting	Evidence in this study
Alarm rule	Combined rule of Zuberi <i>et al.</i> [1]: $PSI \geq 0.10$ with KS $p < 0.001$, or $PSI \geq 0.25$, on ≥ 2 features	No false alarms in any configuration (Figure 1)
Role of feature alarms	Tier 1 (inform): log and inspect the alerted population; do not retrain	Feature tests miss new fraud patterns below ~20% of traffic (Figure 3)
Prevalence monitoring	Weekly z-test on labelled data with a multi-week reference; Tier 2 (investigate) at $ z > 3$	Detects a 1.5-fold rise in 81% of weeks at 1% prevalence
Performance monitoring	Weekly or daily windowed test on fraud capture rate; Tier 3 (act) on a large confirmed drop	Within 1–4 days of sequential tests at far fewer false alarms (Table 3)
Label pipeline	Reduce investigation-to-label latency; prioritise labelling of high-score and disputed cases	Label delay dominates detection time and cost (Figures 4 and 5)
Governance	Document thresholds, owners and actions; review false-alarm and miss rates quarterly	Model-risk guidance [13]

Three operating principles summarise the playbook. First, feature drift is information, not a trigger: the combined rule of the adopted framework [1] keeps false alarms near zero, but its alarms should prompt inspection rather than retraining, because they mostly reflect changes in legitimate behaviour. Second, prevalence and performance shifts are the triggers: they are measured on labels, and they are the only monitors in this study that responded to a change in fraud itself. Third, label latency is the main lever: shortening the time from transaction to confirmed label by one week shortens detection by one week, whereas moving from a weekly to a sequential test saves at most four days and multiplies false alarms.

Discussion

The study confirms, under continuous operation, two observations that Zuberi *et al.* [1] made on a short benchmark: drift can be measured rigorously with standard statistics, and prevalence shift deserves explicit testing alongside feature shift. It adds three findings that only appear over operational time scales. Significance-based tests become less useful as volume grows, because benign variation is always present; this is consistent with broader evidence on large-sample inference [20] and on dataset-shift detection [9]. Feature-level monitoring has a structural blind spot for fraud, because fraud is too small a part of the traffic to move population distributions — a limitation that applies to any unsupervised monitor of the whole population. And the limiting factor in supervised detection is not the statistical test but the operational process that produces labels [12].

For institutions, the practical implication is that a drift-detection deployment should be judged by its false-alarm rate on the institution's own stable data and by its end-to-end time from drift to action, not by the sophistication of its statistics. For research, the results suggest evaluating drift detectors with benign variation present, reporting false-alarm rates alongside power, and studying monitors that focus on the high-score or alerted population, where fraud is concentrated.

Limitations

The experiments use simulated data, which make the presence and size of each shift known but cannot reproduce the full variety of real payment traffic [23]. Benign variation was modelled with small random

perturbations whose size real portfolios may exceed. Only univariate statistics and a single combined rule were studied; multivariate and classifier-based shift detectors [9] may behave differently. The supervised analysis assumes a constant labelling rate and a single step change in model performance, and the cost model rests on stated assumptions rather than measured losses.

Future Work

Validation on production monitoring logs with documented incidents would show how closely these false-alarm and delay estimates hold in practice. Monitors restricted to the alerted, high-score population, or to analyst feedback, could address the minority-drift blind spot. Finally, joint optimisation of thresholds, cadence and label prioritisation under an explicit loss function would turn the playbook into a decision rule tailored to each institution.

Conclusion

Deploying the temporal-drift framework of Zuberi *et al.* [1] in continuous operation shows that its value depends on how each component is used. The combined PSI-KS rule is a robust, low-false-alarm signal of change in the transaction population, but it cannot see new fraud patterns; KS p-values alone are unsuitable for large-volume monitoring; the prevalence z-test is the most sensitive monitor of change in fraud; and label latency, more than any choice of statistic, determines how quickly a fraud model can be corrected. Treating feature drift as information, prevalence and performance shifts as triggers, and label latency as the main lever turns a research framework into an operational practice.

References

- [1] Zuberi, N., Rahman, M. M., Rashid, S. U., Ahsan, M. N., Rahman, M. S., & Mollah, M. H. O. R. (2026). A robust machine learning framework for detecting temporal drift in financial fraud prevention. *Scientific Reports*, 16, Article 27776.
- [2] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235–255.
- [3] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, 57, 47–66.
- [4] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10), 4915–4928.
- [5] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), Article 44.
- [6] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12), 2346–2363.
- [7] Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Advances in Artificial Intelligence – SBIA 2004* (Lecture Notes in Computer Science, Vol. 3171, pp. 286–295). Springer.
- [8] Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (pp. 443–448). Society for Industrial and Applied Mathematics.

- [9] Rabanser, S., Günnemann, S., & Lipton, Z. C. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 1396–1408.
- [10] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28, 2503–2511.
- [11] Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. In *Proceedings of the 2017 IEEE International Conference on Big Data* (pp. 1123–1132). IEEE.
- [12] 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.
- [13] Board of Governors of the Federal Reserve System & Office of the Comptroller of the Currency. (2011). *Supervisory guidance on model risk management* (SR Letter 11-7).
- [14] Yurdakul, B. (2018). *Statistical properties of population stability index* [Doctoral dissertation, Western Michigan University]. ScholarWorks at WMU.
- [15] Massey, F. J., Jr. (1951). The Kolmogorov–Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), 68–78.
- [16] Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145–151.
- [17] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2), 100–115.
- [18] Hinkley, D. V. (1971). Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3), 509–523.
- [19] Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application*. Prentice Hall.
- [20] Lin, M., Lucas, H. C., Jr., & Shmueli, G. (2013). Too big to fail: Large samples and the p-value problem. *Information Systems Research*, 24(4), 906–917.
- [21] Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.
- [22] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.
- [23] 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* (pp. 249–255).
- [24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.