Adaptive Machine Learning for Financial Fraud Detection Under Concept Drift: A Diagnosis-Gated Framework for Model Reliability and Explainability

[1*]Nielsen Keller, Clara Jonas[2]

[1] Technical University of Denmark, DENMARK
[2] ETH Zurich, SWITZERLAND

ABSTRACT

Financial fraud detection is a non-stationary learning problem in which changes in customer behavior, fraud strategies, payment channels, and label availability can silently undermine a model trained on historical transactions. This article develops a diagnosis-gated adaptive framework that separates four functions that are routinely collapsed into a single automatic trigger in practice: drift sensing across inputs, scores, and confirmed outcomes; diagnosis of whether an alert reflects real behavioral change, a data-pipeline fault, or a labeling artifact; selection of a proportionate adaptation response; and post-adaptation validation on temporally later data. The framework is built on the leakage-safe, chronologically evaluated procedure of Zuberi et al., which it implements as its evaluation backbone, and it is tested empirically on the public European credit-card dataset. A model trained on day one was deployed across eight three-hour windows of day two with a one-window label delay, and a data-pipeline fault (the top-ranked feature, V14, written as zero for every record) was injected into window 5. Four policies were compared: no adaptation, retraining whenever a population stability index crossed 0.25, scheduled retraining, and the diagnosis-gated cycle. The naive PSI alarm fired in every eligible window, mainly because of time-of-day composition, so alarm-triggered retraining became indistinguishable from scheduled retraining and raised total cost from EUR 16,450 to EUR 17,436. The gated policy made 2 adaptations instead of 6, identified the injected fault in the window it occurred, and routed scoring to a pre-validated fallback model, recovering recall in that window from 0.00 to 0.73 and lowering total cost to EUR 11,601. Outside the fault window, however, the gate later re-read the quarantined fault window as a performance decline and promoted an unnecessary retrain that raised alert volume without catching more fraud, so the static model was cheapest there. The results support the framework's central claim that diagnosis should precede adaptation, while showing that windows diagnosed as faults must be explicitly quarantined from later outcome checks and promotion tests; they come from a single 48-hour public dataset with an injected fault and should be confirmed on longer, multi-institution data.

Keywords: Financial Fraud Detection; Concept Drift; Adaptive Machine Learning; Model Monitoring; Label Latency; Class Imbalance; Explainable AI; Model Governance

Introduction

A financial institution's fraud-detection model is almost always trained on a finite window of historical, confirmed transactions and then deployed into a live stream whose generating process does not stand still. Customers adopt new payment channels, shift spending patterns in response to economic conditions, and change how they respond to friction such as step-up authentication; fraud operators, for their part, adapt deliberately and quickly once a given pattern begins to be caught, because every successful control changes the incentive landscape the next attack is designed against. The result is a working distinction that is easy to state but harder to operationalize: a model that discriminates well on a fixed, historical test set is not the same thing as a model that remains reliable as the transaction-generating process continues to evolve after deployment. A model can report an unchanged, even improving, aggregate accuracy on a rolling sample of recent transactions while its true detection capability against the specific fraud patterns now active in the

population has quietly degraded, simply because the population of transactions the aggregate metric is computed over has itself shifted in ways that mask the degradation.

The deeper difficulty is that the relevant kind of change is not simply that the marginal distribution of transaction features moves over time. If it were only that, a sufficiently frequent recalibration of score thresholds against the new distribution would largely resolve the problem. The harder and more consequential case is that the relationship between observed transaction features and the fraud label itself can change: a feature combination that was overwhelmingly associated with legitimate activity in the training window can become, within weeks, a leading indicator of a new fraud typology, while a feature that previously carried real discriminative weight can lose it as fraud operators learn to avoid triggering it. The concept-drift literature treats this broad family of phenomena, distinguishing virtual drift in the input distribution from real drift in the conditional relationship between inputs and labels, as a non-stationary learning problem that requires explicit monitoring and adaptation rather than a one-time training exercise [1]. A closely related formalization, covariate shift adaptation, treats the narrower case in which the input distribution changes while the conditional label-generating relationship is assumed to remain stable, and makes explicit the reweighting machinery needed to correct for it; it is a useful limiting case precisely because fraud detection rarely satisfies its stability assumption, which is itself informative about why many importance-weighting corrections developed for covariate shift under-perform when applied naively to fraud streams [2].

Zuberi *et al.*[3] make an important and underappreciated methodological contribution to this problem by demonstrating, on a benchmark credit-card dataset, the value of chronological splitting, train-only preprocessing, feature selection, resampling, and validation-tuned decision thresholds for detecting temporal drift without the optimistic bias that leaks future information into a model's training or tuning process. Their framework is a genuinely useful starting point for evaluating whether a given model is sensitive to drift at all, and it corrects a methodological error, information leakage across the temporal boundary that separates training from deployment, that has inflated reported performance in a meaningful share of published fraud-detection studies. It is, however, an evaluation framework rather than an adaptation policy: it is well suited to establishing, after the fact, that drift occurred and that a particular modeling pipeline is or is not robust to it, but it does not by itself answer the operational questions an institution faces the moment a drift signal actually fires in production. Which signals, among the many that a monitoring system can in principle compute, should be treated as sufficient grounds for intervention? How should the false-alarm rate inherent to any statistical change-detection procedure be controlled so that the fraud team is not permanently chasing noise? How long should a model be allowed to continue operating once an alert has fired but before confirmed outcome labels are available to validate whether the alert was real? And, perhaps most consequentially from a governance perspective, what evidence is sufficient to justify the cost, risk, and validation burden of retraining a production model, as opposed to a lighter-weight response such as recalibration or a temporary threshold adjustment? This article addresses exactly these questions through a decision-oriented adaptive framework rather than a further evaluation methodology.

The practical stakes of getting this balance wrong are not merely theoretical. Dal Pozzolo and colleagues, writing from a practitioner's vantage point on deployed credit-card fraud systems, observe that real fraud-detection pipelines operate under a verification latency in which confirmed outcome labels for a given transaction routinely arrive days or weeks after the transaction itself, making the timing of labels an intrinsic part of the learning problem rather than a secondary data-engineering inconvenience to be abstracted away in an evaluation study[4]. A later companion analysis extends this practitioner framing into an explicit realistic-modeling and learning-strategy proposal for environments that are simultaneously non-stationary, severely imbalanced, and subject to delayed labels, underscoring that these three complications compound rather than occurring independently[5]. More recently, a multi-institution validation study of drift-resilient

early-warning models, spanning both SME loan distress and transaction fraud across a sample of U.S. community financial institutions, provides direct field evidence that the practical challenge of building monitoring and adaptation infrastructure that holds up outside a single benchmark dataset is an active and unresolved industry concern rather than a narrow academic curiosity[6]. Taken together, this body of work establishes both that concept drift is a first-order threat to deployed fraud models and that no single existing contribution specifies a complete, governed decision process for responding to it once detected; closing that specific gap is the purpose of this article.

Research Objectives and Contribution

This article develops an adaptive fraud-detection architecture that explicitly separates four functions that, in many deployed systems, are either conflated into a single automatic trigger or left to ad hoc human judgment with no documented decision criteria: drift sensing across input features, prediction scores, and confirmed outcomes; diagnosis of the source and persistence of an observed signal before any model change is contemplated; selection of a proportionate adaptation response from an explicit menu of options; and post-adaptation validation against temporally later, previously unseen data. Three contributions follow from this separation. First, the article offers a taxonomy of response options, ranging from intensified monitoring with no model change through probability recalibration, threshold adjustment, bounded retraining, ensemble updating, and rollback to a validated fallback model, together with an account of the risk profile each option carries and the class of evidence that should be required before it is selected. Second, it specifies a controlled adaptation cycle, structured as a diagnostic sequence, that is designed to prevent a transient logging fault, a temporary label-reporting delay, or an isolated seasonal fluctuation from triggering a disruptive and costly retraining event that a less-structured monitoring system would otherwise invite. Third, it proposes a set of evaluation criteria, encompassing detection delay, false-alert burden, adaptation frequency, post-adaptation performance across multiple rolling windows, and rollback safety, that extend beyond the purely predictive metrics, such as precision, recall, and the area under the precision-recall curve, that dominate most fraud-detection evaluations and that, on their own, say nothing about how well an adaptive system is actually being governed over time. Fourth, it implements the full cycle on the public European credit-card dataset and compares it with static, alarm-triggered and scheduled policies, so that the first and third propositions in the Analytical Synthesis and Testable Propositions section are tested rather than only stated.

Two prior contributions are treated in this article not merely as supporting citations but as implemented components of the proposed architecture. The chronological-splitting, train-only preprocessing, feature-selection, resampling, and validation-tuned thresholding procedure that Zuberi *et al.* demonstrate for leakage-safe drift evaluation on a benchmark credit-card dataset[3] is adopted directly, not merely referenced, as the evaluation backbone specified in the Evaluation Protocol section below: an institution implementing the present framework is, in effect, also implementing that chronological-evaluation discipline rather than choosing it as one option among several. Likewise, the phased, multi-institution validation design that Rashid and Sattiraju field across SME loan distress and transaction-fraud pilots at a sample of U.S. community financial institutions[6] is adopted as the recommended rollout sequence set out in the Recommendations section: establishing the baseline registry and diagnostic sequence on an already-deployed model before extending governance to a newly introduced one deliberately mirrors the staged validation path their field study reports as both feasible and informative, rather than being an independently invented rollout procedure. The first of these components is also implemented in code in the Empirical Evaluation section, where the Zuberi *et al.* procedure supplies the initial model, the feature selection and the threshold rule for every policy compared. Both components are made explicit here so that a reader attempting to operationalize the framework can trace each governance component back to a specific, previously validated procedure rather than to an unexplained design choice.

Concept Drift and Adaptive Learning in Fraud Detection

Concept Drift as a Family of Distributional Changes

Gama *et al.*[1] distinguish concept drift from the broader and more loosely used notion of simply learning under a changing data stream, and the distinction matters for designing a monitoring system because different members of the concept-drift family call for different detection strategies and different urgency. In retail-fraud prevention specifically, a gradual drift may reflect a slow, population-level shift in customer habits, such as the ongoing migration from in-person to card-not-present spending; an abrupt drift more often follows the sudden appearance of a new attack strategy, such as a coordinated synthetic-identity ring that begins exploiting a previously unremarkable feature combination within a matter of days; and a recurring, or seasonal, drift is typically associated with calendar-linked demand patterns, such as holiday-period transaction surges, that a system has in principle seen before and should be able to anticipate rather than treat as novel each time it recurs. A monitoring architecture that relies on a single detector tuned to one of these patterns, most commonly abrupt change, risks being systematically insensitive to the other two: a detector calibrated to flag sudden jumps will tend to miss a slow drift until it has already accumulated into a large cumulative effect, while a detector tuned to gradual trends may be too slow to flag a sudden attack before material loss has occurred. Figure 1 sketches the idealized shape of each of these three patterns schematically, for expository purposes only; the panels are not derived from any observed dataset, and no two real drift episodes trace an identical path.

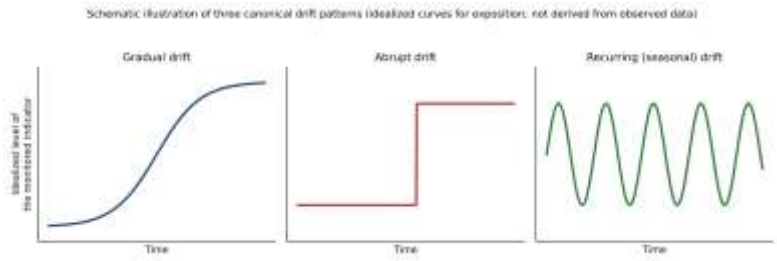

Figure 1. Schematic illustration of three canonical drift patterns (gradual, abrupt, and recurring), shown as idealized curves for exposition only and not derived from observed data.

A further complication, easy to understate without a formal vocabulary for it, is that covariate shift, in which the marginal distribution of the input features changes while the conditional relationship between those features and the fraud label remains fixed, is analytically distinct from the harder case in which that conditional relationship itself changes[2]. The practical significance of the distinction is that a system experiencing pure covariate shift can, in principle, be corrected by reweighting or recalibrating against the new input distribution without altering the underlying decision function at all, whereas a system experiencing a genuine change in the input-label relationship cannot be fully repaired this way and may require retraining on data that reflects the new relationship. Distinguishing the two in practice is difficult because both produce a visible shift in the distribution of model scores; the diagnostic sequence proposed later in this article is designed in part to make exactly this distinction before a response is chosen, rather than assuming that any observed score-distribution shift implies the more disruptive, retraining-requiring case. The monitoring system must also be able to distinguish all of these genuine behavioral phenomena from a quite different and mundane class of explanation: a data-pipeline fault that corrupts or truncates a feature, a delayed or incomplete batch load, a change in how an upstream system encodes a categorical field, or a revision to the institution's own fraud-confirmation process that changes which transactions receive a positive label without any change in underlying customer or fraud behavior at all. Conflating a pipeline fault with genuine drift is

a recurring and costly failure mode, because it can lead an institution to retrain a model specifically to compensate for what is, in reality, a data-quality incident that should instead be fixed at its source.

From Static Models to Periodic and Online Learning

Institutions confronting this landscape have, broadly, three governance postures available to them, and each carries a different balance of operational simplicity against responsiveness. A static model, trained once and left in production without further updates, is the easiest posture to govern: its behavior is fixed, its validation was completed once, and its outputs are, in principle, fully reproducible for as long as it remains in service. The cost of that simplicity is that a static model becomes progressively stale as the environment it was trained on drifts away from the environment it now scores, with no mechanism internal to the model itself that can arrest or even detect that staleness. Periodic retraining, in which the model is refreshed on a fixed cadence, such as monthly or quarterly, offers a middle path by introducing some responsiveness without requiring continuous online learning infrastructure, but a fixed schedule is, by construction, poorly matched to the actual arrival pattern of drift: it can waste validation effort retraining a model that has not meaningfully changed, while leaving an abrupt attack to run unaddressed for up to a full retraining cycle if the attack happens to begin immediately after the most recent scheduled refresh.

Online or near-online adaptation, in which the model is updated continuously or very frequently as new labeled outcomes become available, can in principle react far more quickly to genuine change, but it introduces its own distinct risks that a static or periodic posture does not share to the same degree: noisy or delayed labels can be incorporated into the model before their accuracy is confirmed, feedback loops can emerge in which the model's own past decisions shape the population of transactions it is subsequently trained on, and an institution can lose the ability to point to a single, stable, previously validated model version if updates are frequent and only loosely governed. Dal Pozzolo and colleagues emphasize precisely this point from an applied standpoint: because real fraud systems operate under meaningful label-verification latency and a continuously evolving transaction stream, the timing with which labels become available, and the delay before a model can be credibly evaluated against them, is itself part of the learning problem that a credible adaptive architecture must model explicitly rather than treat as an implementation detail to be resolved later[5]. The choice of base learner interacts with this posture question but does not resolve it: ensemble methods such as random forests[7] or gradient-boosted trees are commonly used in production fraud scoring precisely because they tend to be robust to moderate noise and to the kind of nonlinear feature interactions that fraud patterns typically exhibit, but no choice of base learner, however robust, substitutes for an explicit governance posture on when and how that learner is updated. A further option, situated conceptually between periodic and fully online adaptation, is an adaptive-windowing approach in which the system continuously monitors a sliding window of recent observations and automatically shrinks the window, effectively discarding older data, when a change-detection statistic computed over it exceeds a calibrated threshold[8]. Adaptive windowing of this kind is attractive because it ties the effective training horizon directly to detected change rather than to an arbitrary fixed schedule, but it still requires the same diagnostic discipline this article argues for: a window-shrinking event triggered by a data-pipeline anomaly rather than genuine drift is no more desirable than an unnecessary scheduled retraining would be.

Class Imbalance, Thresholds, and Operational Utility

Fraud labels are rare by nature, commonly representing well under one percent of transactions in retail-banking and card-payment contexts, and the costs of the two error types a classifier can make are sharply asymmetric: a missed fraud case can result in direct financial loss and, where it recurs, reputational and regulatory exposure, while a false alert consumes investigation capacity and, if it reaches the customer, can damage the relationship through unwarranted friction. Precision-recall analysis is particularly informative

under this kind of severe imbalance because it focuses attention on the performance that actually matters operationally, namely how many of the transactions flagged for review are genuinely fraudulent and how large a share of true fraud the flagged set actually captures, rather than on an aggregate accuracy figure that a trivial always-predict-legitimate classifier can satisfy almost perfectly by construction[9]. The receiver operating characteristic curve and its associated area-under-curve summary remain informative for some purposes, most notably for comparing the ranking quality of two models independent of a specific decision threshold, but Fawcett's foundational treatment of ROC analysis is itself explicit that the curve's apparent reassurance can be misleading under severe class imbalance, because a large population of true negatives can make a model's false-positive rate look small in relative terms even when the absolute number of false positives it implies, and therefore the investigation burden it would impose, is operationally unmanageable[10]. The two tools are complementary rather than substitutes, and an adaptive system's monitoring dashboard should report both rather than treating either alone as sufficient.

The Matthews correlation coefficient offers a further complementary summary because it incorporates all four cells of the confusion matrix, true positives, true negatives, false positives, and false negatives, into a single balanced figure that is markedly less sensitive to the class-imbalance distortions that can inflate accuracy, and in some circumstances even F1, when the negative class dominates the data as heavily as it does in fraud detection[11]. Synthetic Minority Over-sampling Technique, and the broader family of resampling methods it inspired, has long been a standard response to class imbalance during model training, generating synthetic minority-class examples by interpolating between existing fraud cases and their nearest neighbors in feature space so that a classifier is not simply overwhelmed by the majority class during optimization[12]. Resampling of this kind interacts with drift in a way that is easy to overlook: synthetic examples generated by interpolating within an older fraud distribution can become a poor representation of the current fraud population once that population has genuinely shifted, so that a resampling strategy calibrated once at training time can itself become a source of staleness that a naive periodic-retraining schedule may not fully correct if the resampling step is not revisited alongside the model it feeds. An adaptive system should therefore not be retrained merely because an aggregate metric such as F1 or accuracy moves by some small amount; the change must be weighed against the metric's own sampling uncertainty, the expected monetary and reputational loss the observed shift implies, and the institution's available investigation capacity, since a decision threshold that is entirely appropriate during a low-transaction-volume period can generate an unmanageable alert queue during a high-volume period even if the underlying model's discriminative quality has not changed at all.

Explainability as Diagnostic Evidence During Adaptation

SHapley Additive exPlanations, commonly abbreviated SHAP, provide a theoretically grounded way of attributing an individual model prediction to the input features that produced it, drawing on cooperative game theory to allocate credit for a prediction across features in a manner that satisfies a specific set of consistency properties[13], and a later tree-specific algorithmic refinement made exact SHAP computation practical at the scale and latency a production scoring system requires, which is part of why SHAP has become a default explainability layer for gradient-boosted and random-forest fraud models in particular[14]. Within an adaptive architecture, SHAP's most valuable role is not to generate a single global explanation of the model but to support a specific diagnostic use: comparing the ranked importance of individual features before and after a monitoring alert, so that an analyst can see, concretely, which feature's rising or falling importance is associated with an observed change in model behavior, rather than being left with only an aggregate metric that has moved without any visibility into why.

That diagnostic use must, however, be exercised with real caution, because an explanation shift is evidence to be weighed rather than proof of a causal change in the underlying fraud-generating process. Lipton's widely

cited caution about the mythos of model interpretability is directly relevant here: an explanation method can appear to offer a transparent account of a model's behavior while in fact obscuring the degree to which that account is itself an artifact of the specific model class, the specific explanation technique, and the correlational structure of the features used, none of which is guaranteed to track the true causal mechanism an analyst is implicitly trying to recover[15]. Correlated predictors can redistribute attributed importance between themselves from one scoring period to the next for reasons that have nothing to do with any change in the underlying relationship, simply because the correlation structure between two historically redundant features has itself shifted; and a change in a feature's apparent importance can equally reflect a change in that feature's own distribution, a case of covariate shift in the narrower technical sense described above[2], rather than any change in how strongly that feature is genuinely associated with fraud once its distribution is accounted for. The practical implication for the architecture proposed in this article is that an adaptation decision should triangulate explanation-based evidence with data-quality checks, distributional drift indicators, and mature-label performance rather than treating a SHAP-based explanation shift, on its own, as sufficient grounds for any specific response.

Proposed Adaptive Framework

The architecture proposed here organizes an institution's fraud-model governance into six interacting modules: a baseline model registry that records every model version together with its training window, feature schema, and validation record; a data-quality and drift-sensing layer; a label-maturity management function that tracks which outcomes are confirmed, provisional, or still pending; a diagnosis and impact-assessment stage; a controlled adaptation stage that executes a chosen response; and a post-change validation stage. These six modules form a feedback loop in the sense that validated outcomes from one cycle inform the baseline against which the next cycle's drift signals are interpreted, but the loop is explicitly not a single automatic trigger: a model change is gated by the accumulation of converging evidence across the drift-sensing, label-maturity, and diagnosis modules, and it is not initiated automatically by any single statistic crossing a threshold, however that threshold was chosen.

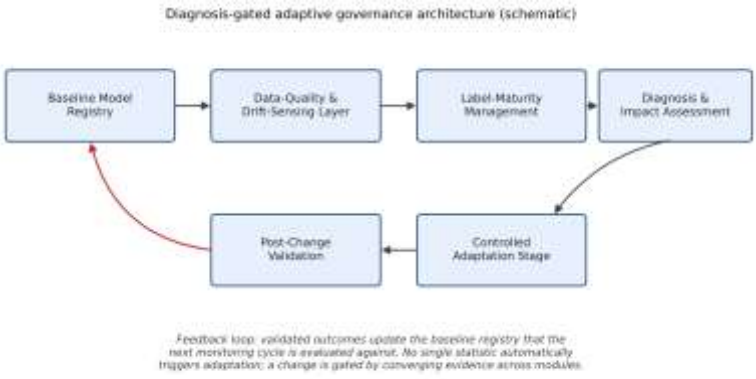

Figure 2. Schematic depiction of the six-module diagnosis-gated adaptive-governance architecture and its feedback loop.

Drift Sensing Across Inputs, Scores, and Outcomes

Effective monitoring must cover more than one layer of the system, because drift that is visible at one layer can be invisible at another and vice versa. Input-feature monitoring tracks the distribution of each feature the model consumes, independent of what the model does with it; prediction-score monitoring tracks the

distribution of the model's own output, which can shift even when no individual input feature has moved appreciably, if the model's internal combination of features has effectively changed how it weights them; prevalence monitoring, where the true fraud rate can be estimated or bounded even before full label confirmation, tracks whether the underlying rate of the event itself is moving; and outcome-based performance monitoring, which can only be computed once labels mature, tracks the actual precision, recall, and related metrics the model is achieving in production. The population stability index compares two binned distributions, typically a current scoring window against a reference baseline, and summarizes the divergence between them as a single number, $PSI = \Sigma_b (q_b - p_b) \ln(q_b / p_b)$, where p_b and q_b are the proportions of observations falling into bin b of the reference and current distributions respectively; it is simple to compute and widely used in credit-risk and fraud monitoring, but it is sensitive to the binning scheme chosen and says nothing by itself about whether an observed shift affects the model's predictive performance. The Kolmogorov-Smirnov statistic offers a complementary, binning-free comparison for continuous features, based on the maximum vertical distance between two empirical cumulative distribution functions, while Jensen-Shannon divergence provides a symmetric, bounded measure of distributional distance that behaves more gracefully than some alternatives when one of the two distributions being compared assigns very low probability to regions the other considers common. These indicators answer related but genuinely different questions, and a monitoring dashboard that reports only one of them risks a blind spot that a different indicator would have caught.

An adaptive-windowing mechanism of the kind proposed by Bifet and Gavaldà, in which the effective size of the training window used for monitoring or retraining is automatically reduced when a statistical test detects a significant change between the older and newer portions of the window, offers one concrete way of operationalizing continuous drift sensing without committing in advance to a fixed lookback period[8]. Thresholds for all of these indicators, whichever subset an institution chooses to deploy, should be calibrated against the institution's own historical variability and the operational consequences of a false alert, rather than adopted wholesale as universal constants imported from a different dataset or a different institution's published benchmark, because the natural volatility of a given feature or score in one transaction population can be entirely different from its volatility in another, and a threshold tuned for one will be systematically too sensitive or too insensitive for the other.

Diagnosis Before Adaptation

An alert generated by any of the drift-sensing indicators described above should initiate a structured diagnostic sequence before any adaptation response is selected, precisely because the same statistical signature, a shift in a feature's distribution or in the model's score distribution, can be produced by causes that call for completely different responses. The first diagnostic step is to verify the basic integrity and availability of the input data feeding the alert, ruling out a pipeline fault, a truncated batch, or a schema change at the source system before treating the signal as behavioral at all; a surprising share of production drift alerts, in practice, trace back to exactly this kind of data-quality incident rather than to any genuine change in customer or fraud behavior. The second step, once data integrity is confirmed, is to determine whether the observed shift is localized to a specific feature, a particular customer segment, a specific channel such as a newly launched payment method, or a bounded time period, as opposed to being diffuse across the whole population; a localized shift is both easier to diagnose further and more amenable to a targeted response than a diffuse one. The third step is to check, to the extent mature labels are already available, whether confirmed outcome-based performance metrics have actually moved in a way consistent with the input-level alert, since a distributional shift that leaves outcome performance unaffected is a materially different and generally lower-urgency situation than one that coincides with a genuine drop in precision or recall. The fourth step is to assess whether the shift, once confirmed, appears to be persistent or merely transient, for

instance by checking whether it reverses within a short subsequent window, since a transient fluctuation that is about to self-correct does not warrant the same response as a persistent structural change. Only once this sequence has been worked through should the institution select a specific intervention; following it in order is what reduces the risk, highlighted repeatedly throughout this article, of retraining a production model to compensate for what later proves to have been a temporary logging error, a short-lived seasonal blip, or a data-quality incident that should instead have been remediated at its source. Figure 3 summarizes this four-step sequence.

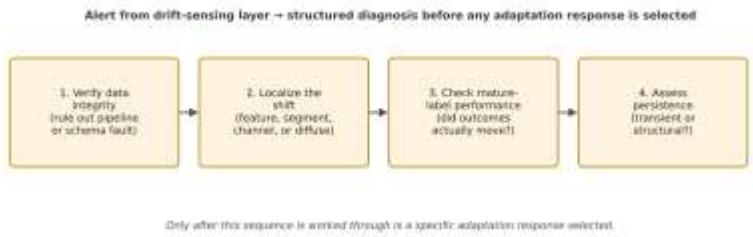

Figure 3. The four-step diagnostic sequence applied to any drift alert before an adaptation response is selected.

Adaptation Options and Their Risk Profiles

Once a diagnosis has established that a response is genuinely warranted, the architecture proposes a graded menu of response options rather than a binary retrain-or-ignore choice, because the available responses differ substantially in cost, disruption, and in the specific failure mode each one can and cannot address. Retaining the current model while intensifying monitoring, for instance by shortening the review cadence for the specific indicators that triggered the alert, is the least disruptive response and is appropriate when the diagnostic sequence has not yet produced conclusive evidence either way. Recalibrating the model's output probabilities, without altering its underlying ranking function, is appropriate when the evidence suggests the model's relative ordering of risky transactions remains sound but its absolute probability estimates have drifted, for example because the true fraud prevalence has shifted; recalibration is comparatively cheap and low-risk precisely because it leaves the model's decision boundary over feature space unchanged. Adjusting the decision threshold is similarly lightweight and is the correct response when the score distribution has shifted in a way that is well explained by a change in transaction volume or population composition rather than by any change in the underlying ranking quality, but a threshold adjustment cannot repair a genuinely changed ranking function, and applying it to a situation that actually calls for retraining will mask rather than resolve the underlying problem. Retraining on a recent window of data is the appropriate response once the diagnosis indicates a real change in the input-label relationship that a purely post-hoc adjustment to an unchanged model cannot address, but retraining carries real risks of its own: it can cause the model to forget genuinely useful patterns learned from historical data that remain valid, and it can inadvertently fit noise in labels that have not yet been fully confirmed if the label-maturity module has not been consulted first. Updating an ensemble, by adding a new weak learner tuned to recent data while retaining the existing ensemble members, offers an intermediate option between full retraining and no model change, and reverting to a previously validated fallback model is the appropriate response when a recent adaptation is itself shown, through post-adaptation validation, to have degraded performance rather than improved it.

Table 1. Mapping from monitoring evidence to candidate adaptation responses.

Monitoring evidence	Likely interpretation	Candidate response	Risk if misapplied
Input-distribution shift only; outcome performance unchanged; isolated to one feature or segment	Localized covariate shift or a benign population change not yet affecting predictive validity	Intensify monitoring of the affected feature or segment; no model change yet	Under-reaction if the shift is an early leading indicator of a larger change still to come
Score distribution shifts broadly; outcome performance stable once labels mature	Change in population composition or volume rather than in ranking quality	Recalibrate output probabilities or adjust the decision threshold	Masking a genuine ranking-function problem if the diagnosis was incomplete
Outcome-based precision or recall degrades on mature labels; shift persists beyond a short window; not attributable to a data or labeling fault	Real change in the input-label relationship (concept drift proper)	Retrain on a recent, label-mature window; validate before full rollout	Overfitting to noisy or still-provisional labels; forgetting valid historical patterns
A newly deployed model's post-adaptation validation shows degraded performance relative to the prior validated model	The most recent adaptation was itself unwarranted or poorly specified	Roll back to the last validated fallback model; re-diagnose	Repeated rollback cycling if the fallback itself has also become stale

Post-Adaptation Validation

Any model that has been updated through retraining, ensemble modification, or a material recalibration should be evaluated on data that were not used, directly or indirectly, to fit or to select it, including data that were not used to tune the specific threshold or hyperparameters the diagnosis stage might have adjusted alongside the model itself. The evaluation should report performance separately across multiple chronological windows both before and after the adaptation event, rather than collapsing the comparison into a single before-after summary statistic, because a single summary can obscure whether an apparent improvement is concentrated in a short period immediately following the change, which is suggestive of overfitting to recent, possibly still-provisional labels, or is sustained across several subsequent windows, which is better evidence of a genuine improvement. The reported metrics should include precision, recall, the area under the precision-recall curve, the Matthews correlation coefficient, calibration quality, the resulting alert volume at the institution's chosen operating threshold, and a measure of predictive uncertainty where the modeling approach supports one. Critically, a model should never be declared improved merely because it fits the most recent data better than its predecessor did; the relevant test is whether it generalizes acceptably to observations that arrive after the adaptation, since a model retrained specifically to accommodate the most recent window can achieve an attractive-looking immediate improvement that does not survive contact with the subsequent window, a pattern that is itself a leading indicator of exactly the overfitting risk the diagnosis stage is designed to guard against.

Evaluation Protocol

Evaluating an adaptive fraud-detection architecture credibly requires a protocol that simulates deployment rather than one that evaluates a static, retrospectively fixed model against a held-out test set in the conventional way. At each evaluation cutoff, the model in force is restricted to information that would genuinely have been available before that cutoff, and it is then evaluated on the subsequent time window using outcomes that have, by the time of evaluation, had sufficient time to mature. Where labels arrive with a known or estimable delay, the simulation protocol must explicitly prevent the evaluation from training on, or being credited with knowledge of, outcomes that would not yet have been confirmed at the relevant point in simulated time; this constraint is especially consequential because the most recently observed transactions in any rolling window are disproportionately likely to be the ones still awaiting label confirmation, so a naive evaluation that ignores label latency will systematically overstate how much of the most recent data could actually have informed a real-time decision. As stated in the Research Objectives and Contribution section, this chronological-cutoff discipline is not a novel proposal of this article but a direct implementation of the leakage-safe splitting, train-only preprocessing, and validation-tuned thresholding procedure that Zuberi *et al.* demonstrate empirically[3]; the protocol below should be read as operationalizing their procedure for the specific case of a diagnosis-gated, multi-response adaptive system rather than the single-model benchmark they evaluate.

Predictive Performance Metrics

Let TP, FP, TN, and FN denote the standard confusion-matrix counts of true positives, false positives, true negatives, and false negatives respectively at a given operating threshold. Precision is then TP/(TP+FP), the share of flagged transactions that are genuinely fraudulent, and recall is TP/(TP+FN), the share of genuine fraud that is actually flagged; an F-score can be weighted to emphasize recall over precision, or vice versa, when the asymmetric cost of a missed fraud case relative to a false alert justifies doing so. The Matthews correlation coefficient, $MCC = (TP \cdot TN - FP \cdot FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]}$, incorporates all four confusion-matrix counts into a single figure and is particularly useful under the severe imbalance characteristic of fraud detection, where accuracy and even F1 can be misleadingly favorable[11]. The area under the precision-recall curve should be reported alongside the area under the receiver operating characteristic curve rather than in its place, because a very large population of legitimate transactions can make ROC-based performance look reassuring even while the precision an institution would actually experience at its chosen alert volume remains operationally poor[9]; reporting both curves, in the spirit of Fawcett's broader treatment of what ROC analysis can and cannot tell an analyst under imbalance[10], gives a more complete picture than either alone.

Adaptation-Specific Measures

Beyond these predictive-quality metrics, the evaluation protocol should report a distinct family of measures specific to how well the adaptive system is being governed over time, since two systems with identical predictive performance at a single point in time can differ substantially in how disruptively and how reliably they reach that performance. Detection delay measures the time elapsed between the onset of a meaningful, confirmed change and the system's first alert; false-alert rate measures how often the system signals a change that subsequent diagnosis does not confirm as genuine; the proportion of alerts that lead to a justified intervention, as opposed to a diagnosis that concludes no action is warranted, indicates how well-calibrated the alerting layer is; adaptation frequency and rollback frequency track how often the model is actually changed and how often a change is later reversed; and recovery time measures how long it takes a system's measured performance to return to an acceptable level once a genuine drift event has begun. Operational utility can in principle be expressed as a single expected-cost figure that weights false alerts and missed fraud

by their respective costs, but the cost coefficients used in such a calculation must be estimated from the institution's own loss and investigation data, or explicitly elicited from the relevant business stakeholders, rather than invented for the purpose of the evaluation; where credible cost estimates are genuinely unavailable, the evaluation should instead report sensitivity analyses across a plausible range of cost assumptions so that a reader can judge how the conclusions depend on that range, rather than presenting a single point estimate that conceals the underlying uncertainty.

Reproducibility and Governance Requirements

Each model version produced by the adaptation cycle should be associated, in the baseline model registry described earlier, with its training window, its feature schema, the specific preprocessing pipeline applied to produce its inputs, the decision threshold in force, the full post-adaptation validation results that justified its deployment, and a documented approval record identifying who authorized the change and on what evidence. The system should at all times retain the previous validated model as an available fallback, rather than treating the newest model as an irreversible replacement, precisely so that the rollback response described in the adaptation-options taxonomy above is actually available when post-adaptation validation calls for it. Governance also requires a defined owner for every alert the monitoring layer raises and a standing procedure for documenting, for each alert, whether it was ultimately attributed to genuine drift, a data-quality incident, a label-reporting delay, or some other cause; without this documentation discipline, an institution loses the ability to learn, over successive cycles, which of its monitoring indicators are actually informative in its own operating environment and which are generating noise that the diagnostic sequence is repeatedly, and expensively, having to screen out.

Empirical Evaluation on the European Credit-Card Stream

Data, Backbone Model and Deployment Simulation

The evaluation uses the public European credit-card dataset: 284,807 transactions over two days, 492 of them fraudulent[16]. Day one (the first 86,400 seconds) supplied the initial training data; its last 20 percent served as the validation window. Day two was divided into eight consecutive three-hour deployment windows containing 48, 21, 10, 17, 26, 37, 40, 12 frauds respectively (211 in total). The backbone followed Zuberi *et al.*[3]: features were standardised on training data only, 15 inputs were selected by mean absolute SHAP value from a class-balanced gradient-boosting surrogate fitted on training data, SMOTE was applied to the training data only[12], a LightGBM classifier was fitted[17], and the alert threshold was the F2-optimal value on the validation window. The raw elapsed-time field was excluded as a candidate input because it is a monotonic clock and is always outside its training range at deployment.

Labels were treated as maturing with a one-window delay: when window w is scored, outcomes are known only for windows up to w - 2. To test diagnosis, a data-pipeline fault was injected into window 5 by writing the top-ranked feature, V14, as zero for every record, imitating an upstream null-fill error. The fault is synthetic; everything else is the original data. Cost was measured as EUR 5 per alert (an assumed analyst review cost) plus the amount of every missed fraud.

Four policies were compared. The static policy never changes the model. The alarm-triggered policy retrains on all mature data whenever any monitored feature in the most recent window has a PSI of 0.25 or more against training. The scheduled policy retrains before every window from window 3 onward. The diagnosis-gated policy applies the sequence of the Proposed Adaptive Framework section: (i) a label-free data-integrity screen that flags any input where more than half the records share a single value although fewer than 5 percent did in the reference; (ii) input drift measured against a seasonally matched reference, meaning day-one transactions from the same hours of the day; (iii) score drift; and (iv) mature-label PR-AUC on the most

recent mature window. An integrity failure routes scoring to a pre-validated fallback model trained without the affected input. A mature-label PR-AUC more than 0.15 below the validation value triggers a bounded retrain, which is promoted only if it is no worse than the incumbent on the most recent mature window. Input and score shift together without label evidence trigger intensified monitoring only.

Results

Table 2 Summarises the four policies over the eight windows and Figure 4 shows cumulative cost.

Table 2. Policy comparison over eight day-two windows (cost in EUR; fault injected in window 5).

Policy	Model changes	Alerts	Frauds caught	Recall	Missed fraud value	Total cost	Cost excl. fault window
Static	0	230	145/211	0.687	15,300	16,450	9,438
Alarm-triggered retraining	6	228	146/211	0.692	16,296	17,436	10,424
Scheduled retraining	6	228	146/211	0.692	16,296	17,436	10,424
Diagnosis-gated	2	340	164/211	0.777	9,901	11,601	9,858

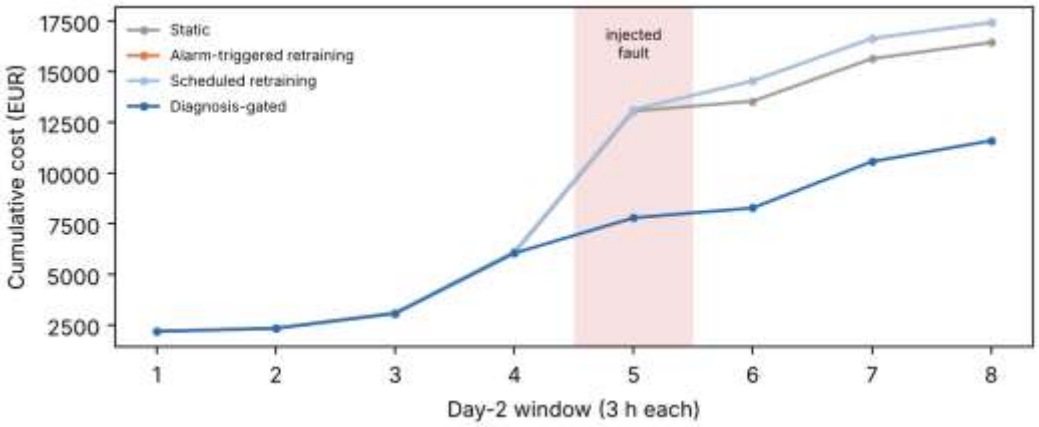

Figure 4. Cumulative cost by policy across the eight day-two windows; shaded window contains the injected fault [authors' analysis].

Naive drift sensing was uninformative. Measured against the whole training period, the largest feature PSI was between 5.21 and 8.26 in every window, so the alarm-triggered policy retrained before every eligible window and produced exactly the same models and costs as the scheduled policy. Against the seasonally matched reference the largest PSI fell to between 1.28 and 1.51 outside the fault window, which shows that most of the apparent drift was time-of-day composition. Retraining on this signal did not pay: both retraining policies ended with a higher total cost than the static model (EUR 17,436 against EUR 16,450).

The injected fault exposed the main weakness of policies without diagnosis. With V14 set to zero, the static model raised 0 alerts and missed all 26 frauds in window 5 (cost EUR 7,012). The retraining policies did no better (EUR 7,012), because retraining cannot repair corrupted inputs at scoring time. The gated policy's integrity screen flagged V14 in the same window using no labels, diagnosed a data-pipeline fault, and routed

scoring to the fallback model, which caught 19 of the 26 frauds (recall 0.73) at a cost of EUR 1,743. Almost all of the gated policy's overall advantage comes from this window.

Outside the fault window the gated policy was not the cheapest, and the reason is instructive. Before window 7 the outcome check evaluated the incumbent on the most recent mature window, which was window 5, the fault window itself, where mature-label PR-AUC was 0.309 (validation value 0.783) because the inputs had been corrupted. The gate therefore read the same fault a second time, now as a performance decline, triggered a bounded retrain, and validated the candidate on that corrupted window, where it passed and was promoted. The promoted model raised 76 and 64 alerts in windows 7 and 8 (static model: 38 and 18) while catching the same number of frauds. Excluding the fault window, total cost was EUR 9,438 for the static model, EUR 9,858 for the gated policy and EUR 10,424 for both retraining policies.

The lesson is specific. A window diagnosed as a data fault must be quarantined: it should be excluded from later outcome checks, from promotion tests and from training data, and the diagnosis log must carry that status forward until the fault is repaired. The cycle described in the Diagnosis Before Adaptation section implies this, but the evaluation shows it must be stated as an explicit rule, together with a promotion test that requires confirmation on more than one clean mature window.

Analytical Synthesis and Testable Propositions

Two analytical points follow from the architecture developed above. First, adaptation is valuable only when the sensitivity of the detection signal and the urgency of the resulting response are properly aligned with each other. A drift detector tuned to be sensitive to any distributional movement at all will tend to trigger costly interventions in response to harmless seasonal or population-composition changes that do not actually affect predictive validity, consuming governance capacity that would be better spent elsewhere; a detector that insists on waiting for fully mature, confirmed labels before raising any alert will, by construction, respond too slowly to an abrupt attack that is doing active damage in the interim. The correct balance between these two failure modes is not a universal constant; it depends on an institution's risk tolerance, on the specific time its labeling process requires to confirm a fraud outcome with confidence, and on the investigation capacity it has available to absorb a higher rate of lower-confidence alerts if it chooses to lean toward greater sensitivity.

Second, and following directly from the response taxonomy set out above, retraining is not, and should not be treated as, the default or only response to a confirmed drift signal. If a model's underlying ranking of transactions by risk remains sound but its output probability calibration has shifted, for instance because the true fraud prevalence has changed, a comparatively simple recalibration may be entirely sufficient and considerably less risky than a full retraining cycle. If the observed score distribution has shifted primarily because transaction volume or population composition has changed, a review of the operating threshold or of the institution's investigation-capacity plan may be the more appropriate response. If, by contrast, the underlying conditional relationship between features and the fraud label has genuinely changed, a case properly diagnosed as concept drift in the stricter sense discussed earlier[1], a new model or an updated feature representation may indeed be required, and an mapping from this kind of diagnostic evidence to the corresponding candidate response, together with the specific risk each mapping carries if misapplied, is set out in Table 1. The diagnosis stage described in the Proposed Adaptive Framework section exists specifically to help an institution select the least disruptive intervention that is actually consistent with the available evidence, rather than defaulting reflexively to the most disruptive one out of an understandable but ultimately counterproductive abundance of caution.

Testable Propositions

Three propositions follow from this architecture. The first proposition, P1, is that a diagnosis-gated adaptation policy will reduce the rate of unnecessary retraining events, relative to an automatic policy that retrains whenever a single distributional statistic crosses a fixed threshold, without a correspondingly large increase in detection delay for genuine drift events. The evaluation above supports P1 on this dataset: the gated policy made 2 model changes against 6, and it responded to the injected fault in the window where it occurred, whereas the threshold policy never responded to it. One of the gated policy's two changes was itself unnecessary, so the reduction is real but not complete. The second proposition, P2, is that explicitly incorporating label-maturity information into the monitoring layer, so that outcome-based performance estimates are weighted or filtered according to how confirmed the underlying labels actually are, will improve the reliability of recent performance estimates relative to treating all recent labels as equally trustworthy; this proposition follows directly from the label-latency concerns raised by Dal Pozzolo and colleagues[4][5]. It could not be tested here, because the public dataset records no label-arrival times and the one-window delay was imposed by design. The third proposition, P3, is that combining distributional, input-level signals with outcome-based signals, rather than relying on either family alone, improves the trade-off between detection delay and false-alert burden. The evaluation gives partial support: the label-free integrity screen caught the fault with no delay, the seasonally matched input statistics removed most of the spurious drift signal, and the outcome-based check was the only signal sensitive to performance, but because it was fed the quarantined fault window it also produced the one unnecessary retrain.

Discussion

The framework developed in this article treats adaptive fraud detection as a governed control process rather than as a purely algorithmic, model-selection problem. The predictive model itself is one component of a considerably wider system that also includes the data pipelines feeding it, the label-verification process that confirms its outcomes, the investigation capacity available to act on its alerts, the institution's risk appetite, and the change-approval process that governs when and how the model itself may be modified. This framing is consistent with the realistic, latency-aware operational perspective that Dal Pozzolo and colleagues bring to credit-card fraud detection[5] and with the broader concept-drift literature's insistence that non-stationary learning problems require monitoring and adaptation infrastructure, not merely an occasionally refreshed model[1]. It is also consistent with the direction suggested by recent multi-institution field evidence: a validation study spanning SME loan distress and transaction fraud across a sample of U.S. community financial institutions found that building monitoring and adaptation infrastructure robust enough to generalize beyond a single institution's own data is itself a substantial and unresolved practical challenge, which is exactly the kind of challenge a governed, explicitly diagnosed adaptation process is intended to address[6].

The framework also pushes back against a recurrent overclaim that is easy to slip into when discussing adaptive machine learning in applied settings: demonstrating that a system can detect drift is not the same thing as demonstrating that the system is resilient to drift. Resilience can only be demonstrated through sustained performance over time, measured honestly against temporally later data the system has not seen, and through the quality and proportionality of the response the system produces once a change is detected, not merely through the sensitivity of its detection layer in isolation. Establishing resilience in this fuller sense requires prospective or rolling-origin evaluation of the kind described in the Evaluation Protocol section, together with careful, explicit accounting for delayed and provisional labels rather than treating every recent label as equally trustworthy. A short benchmark evaluation, however methodologically careful, of the kind Zuberi *et al*. provide[3], can meaningfully support methodological development and can establish that a given modeling pipeline is not obviously broken under drift, but it cannot, by itself, establish long-term resilience in a live production environment, which is a claim that only sustained, governed operation over an extended

period can ultimately support. The present evaluation shares that limitation: it covers 48 hours of one public dataset and one injected fault, and it shows that the diagnostic sequence behaves as designed, not that it will deliver resilience in production.

Conclusion

An adaptive fraud-detection system should not react to every statistical change it observes with automatic retraining. It should instead detect a possible change, diagnose its source and persistence before committing to any specific response, determine whether the change actually affects predictive utility as measured on mature, confirmed outcomes, select a response that is proportionate to the diagnosed evidence from an explicit menu of options rather than defaulting to the most disruptive available action, and validate the resulting model on temporally later data that played no part in producing it. This process combines the strengths of distributional drift monitoring, rare-event-aware evaluation, explainability used as corroborating rather than conclusive evidence, and explicit model governance, while limiting the institution's exposure to overreacting to transient or non-actionable statistical noise. The contribution this article offers is a diagnosis-gated adaptation cycle, built around explicit monitoring of detection delay, false-alert burden, label maturity, post-adaptation performance across multiple rolling windows, and rollback safety, together with a taxonomy of graded response options matched to the evidence the diagnostic sequence actually produces. Implemented on the European credit-card stream, the cycle avoided 4 of the 6 retraining events a threshold policy would have made, contained an injected data-pipeline fault in the window it occurred, and lowered total cost from EUR 16,450 to EUR 11,601; it also showed that a window diagnosed as a fault must be quarantined from later outcome checks and promotion tests, or the same fault will be read twice. Future work should compare the proposed framework against both a fixed-schedule periodic-retraining policy and an automatic, single-threshold drift-triggered retraining policy, using long-duration, temporally ordered, and institutionally diverse data of the kind that multi-institution validation efforts are beginning to make available[6], so that the three propositions advanced in this article can be tested under conditions that approximate the governance problem a real institution actually faces.

References

[1] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44.

[2] Sugiyama, M., & Kawanabe, M. (2012). Machine learning in non-stationary environments: Introduction to covariate shift adaptation. MIT Press.

[3] Zuberi, N., Rahman, M. M., Rashid, S. U., Ahsan, M. N., Rahman, M. D., & Mollah, M. H. O. R. (2026). A robust machine learning framework for detecting temporal drift in financial fraud prevention. *Scientific Reports*.

[4] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2015). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915-4928.

[5] 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.

[6] Rashid, S. U., & Sattiraju, S. R. (2025). Multi-Institution Validation of Drift-Resilient Early-Warning Models for SME Loan Distress and Transaction Fraud: Evidence from US Community Financial Institution Pilots. *International Journal of Modern Computing*, *8*(1), 40-54.

[7] Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.

[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).

[9] 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), Article e0118432.

[10] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

[11] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21, Article 6.

[12] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.

[13] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

[14] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2, 56-67.

[15] Lipton, Z. C. (2018). The mythos of model interpretability. Communications of the ACM, 61(10), 36-43.

[16] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence* (pp. 159-166). IEEE.

[17] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30* (pp. 3146-3154).