IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

DriftSentinel: Proactive Distribution-Shift Monitoring and Autonomous Retraining for Machine-Learning Models in High-Velocity Data Streams

Amrapali Chavan[1], Asit Pawar[2], Manish Pawar[3], Varun Patil[4], Aditya Parhar[5]
Assistant Professor[1], Department of Artificial Intelligence and Data Science
Student[2-5], Department of Artificial Intelligence and Data Science
AISSMS Institute of Information Technology, Pune, India

Abstract: *When machine learning models are deployed in the real world, they often fail because real-world data constantly changes and stops looking like the training data. For time-sensitive applications like fraud detection or high-frequency trading, these data changes (or "drifts") need to be caught in seconds. However, most current monitoring tools only check for drift every few hours, which is far too slow to prevent major issues.*

To solve this, we built DriftSentinel, a monitoring system made specifically for real-time data streams using Apache Kafka. Instead of waiting for hours, it checks small batches of incoming data right away. It does this by running three different tests at the same time: the Population Stability Index (PSI) to check how individual features change, the Kolmogorov–Smirnov (KS) test to compare the overall data shape, and an Isolation Forest model to catch complex, multi-variable changes. By combining the results of these three tests, the system creates a single "severity score" to decide if the data has drifted. To avoid false alarms from temporary glitches, it double-checks to make sure the drift is real before automatically retraining the model.

We tested this system on a simulated stream of 3.2 million financial transactions where we deliberately introduced different types of data drift. Our system was able to detect the drift in just 4.7 seconds on average. It only triggered false alarms 2.3% of the time, and the entire process of detecting the drift, retraining the model, and deploying the new version took only 91 seconds. Compared to standard hourly checks, DriftSentinel kept the model's accuracy much higher during the drift and reacted over 80% faster.

Keywords: *DriftSentinel*

I. INTRODUCTION

Reliability Erosion in Production ML Systems

The operational lifecycle of a machine-learning artefact involves a transition that standard software deployments do not: the moment a trained model enters a live serving environment, its predictive validity begins to erode at a rate governed not by engineering choices but by the external world it was built to model . Statistical learning algorithms are trained to minimise expected loss under a particular joint measure over inputs and outputs; when that measure shifts—as financial instruments are repriced, as consumer demographics evolve, as sensor hardware ages—a model whose parameters remain fixed accumulates a growing mismatch between its internal representation and the current data-generating process .

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

Machine Learning Operations (MLOps) has emerged as the engineering subdiscipline whose central mandate is managing exactly this mismatch across the full post-training lifecycle of a model . Unlike classical software quality assurance, which can validate a system by examining its code against a specification, ML quality assurance must validate a statistical artefact against a moving empirical reality. This demands continuous measurement infrastructure, versioned data pipelines, and—as this paper argues—autonomous remediation capabilities that activate faster than any human-in-the-loop process can respond .

The High-Velocity Monitoring Problem

Contemporary event-streaming architectures sustain throughput levels that render scheduled-batch monitoring both technically and economically inadequate. A payment processing network handling 5×10^4 transactions per second accumulates 180 million events in a one-hour monitoring window; a single late-detected fraud model degradation over that window can yield losses running to seven figures before any alert is raised. The root cause is structural: batch monitoring pipelines accumulate events until a scheduled trigger fires, then process the entire window as a unit. Detection latency is therefore the sum of the scheduling interval and the processing time—both of which are irreducible in batch designs and both of which are incompatible with sub-minute remediation requirements .

Streaming-native monitoring dissolves this latency floor by treating drift computation as a continuous operation co-located with model inference, reducing the unavoidable latency component to the minimum window size required for statistical tests to achieve adequate power. At $\lambda = 5 \times 10^3$ events/s and a window of $B = 5{,}000$, this floor is precisely one second—three to four orders of magnitude below what scheduled-batch designs achieve in practice .

Limitations in the Current Tooling Landscape

Two broad families of drift detection tools exist today. The first—sequential change-point procedures such as the Page–Hinkley method and adaptive windowing algorithms —react to shifts in a scalar performance metric, typically classification error, and therefore cannot fire until ground-truth labels arrive and error accumulates; this introduces an irreducible post-shift observation delay that in many domains extends to days. The second family—distributional comparison tools including PSI dashboards and held-out test evaluations—provides richer diagnostic content but executes on batch schedules that are incompatible with streaming latency budgets . No existing production-grade tool unifies streaming-native multi-signal detection with automated, shadow-validated model promotion in a single coherent architecture.

Paper Contributions

Four original contributions distinguish this work:

The design specification and empirical validation of *DriftSentinel*, a Kafka-native monitoring subsystem sustaining $> 10^6$ events/s throughput with no measurable impact on primary serving latency.

A three-signal evidence-fusion architecture combining PSI, KS, and Isolation Forest outputs into a scalar severity indicator Δ, governed by empirically calibrated weights and a persistence-gate debounce circuit.

An autonomous retraining trigger capable of completing the full detection-to-promoted-model cycle within 91 s for gradient-boosted tree models, validated across five heterogeneous drift scenarios.

Ablation evidence quantifying the non-redundant contribution of each detection component, demonstrating that no single-component subset matches the full system on all evaluation criteria simultaneously.

II. RELATED WORK

Sequential Change-Point Detection

Detecting abrupt changes in the statistical behaviour of observed sequences has occupied statisticians since at least the early post-war decades. Page's cumulative sum procedure and the Hinkley variant studied by Mouss et al. both accumulate signed deviations of a test statistic and declare a change when the cumulative total clears a threshold. These

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

methods carry well-characterised theoretical properties— average run length to false alarm, expected detection delay—under parametric assumptions on the monitored quantity, making them tractable to analyse but sensitive to model mis-specification. Their operational dependence on a continuously available scalar error signal, however, limits their applicability in streaming contexts where labels arrive with multi-day delays.

Bifet and Gavalda's adaptive windowing algorithm relaxes the fixed-horizon assumption by dynamically contracting the observation window whenever sub-window means differ significantly at a calibrated confidence level. This property produces faster detection of abrupt changes and graceful adaptation to gradual drift without window re-specification, yet the method remains anchored to a single aggregate statistic and inherits the label-dependency limitation of its predecessors.

Distributional Comparison Methods

A label-free alternative treats drift detection as a two-sample hypothesis testing problem: given a reference sample from the training distribution and a test sample from the live stream, does statistical evidence favour the hypothesis that the two sets are drawn from different distributions? The Kolmogorov–Smirnov statistic—the supremum of the absolute gap between two empirical distribution functions— provides a distribution-free answer with exact finite-sample critical values . Rabanser, Günnemann, and Lipton conducted a systematic empirical evaluation of two-sample tests for dataset shift detection, finding that projection-based multivariate extensions of the KS test outperform per-feature univariate applications on correlated inputs; their study, however, focused exclusively on batch evaluation protocols and offered no integration with retraining infrastructure.

The Population Stability Index originated in regulatory model risk management within banking institutions . Its computation—a symmetrised relative entropy between reference and monitoring bin proportions—produces an interpretable scalar that practitioners can compare against established risk thresholds without statistical training. Despite widespread adoption in credit-model validation workflows, PSI has historically been applied to monthly model-review cycles rather than second-scale streaming pipelines.

Anomaly-Detection Approaches to Drift

Reconstruction-based methods train a generative or auto-encoding network on reference data and use elevated reconstruction error on new observations as a proxy for out-of-distribution membership . These approaches generalise naturally to high-dimensional inputs and require no distributional assumptions, but their computational overhead during both training and inference is substantially higher than non-parametric statistical tests, complicating real-time deployment.

Liu, Ting, and Zhou's Isolation Forest offers a different geometric insight: anomalous observations are, by definition, easier to isolate by random binary partitioning than typical ones, and this isolation speed can be measured as average path length across an ensemble of random partition trees. Its $O(n\log n)$ training and $O(\log n)$ scoring complexity make it computationally practical for the rolling-reference update cycles that streaming monitoring requires, and its performance on high-dimensional tabular data has been demonstrated across multiple benchmark studies .

MLOps Platforms and Identified Gaps

Established MLOps platforms have matured substantially in their experiment-tracking, model-registry, and pipeline-orchestration capabilities . Dedicated monitoring add-ons such as Evidently AI, WhyLogs, and Seldon's monitoring probes provide distributional health reports and alerting dashboards. Four persistent gaps motivate the DriftSentinel design: (i) monitoring computations remain scheduled as batch jobs, introducing latency floors measured in minutes to hours; (ii) detection and retraining are operationally decoupled, requiring human escalation between alert and remediation; (iii) multivariate drift sensitivity is either absent or conditioned on labelled evaluation data; and (iv) alert thresholds are manually configured and degrade in precision as the deployment ages without recalibration.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

Problem Formalisation

Distribution Shift: Definitions

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y}$ the output space. A supervised model $f_\theta : \mathcal{X} \to \mathcal{Y}$ is fitted on training corpus $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{N}$ whose elements are independently realised from a joint probability measure $\mathbb{P}_{\text{tr}}(X, Y)$. Once deployed at time T_0, the model serves an evolving input stream $\mathcal{S} = \{x_t\}_{t=1}^{\infty}$ whose governing measure $\mathbb{P}_t(X, Y)$ is not, in general, stationary. Distribution shift is declared whenever:

$$\mathbb{P}_{\text{tr}}(X, Y) \neq \mathbb{P}_t(X, Y), \; t > T_0$$

Two mechanistically distinct realisations of ([eq:shift_def]) carry separate operational implications. *Input covariate drift* occurs when the observable feature measure migrates while the label-generating mechanism conditional on features stays fixed:

$$\mathbb{P}_t(X) \neq \mathbb{P}_{\text{tr}}(X), \; \mathbb{P}_t(Y \mid X) = \mathbb{P}_{\text{tr}}(Y \mid X)$$

Under ([eq:covariate]) the Bayes-optimal classifier is unchanged in principle, but a model fitted on sparsely represented input regions may still degrade when those regions become dominant in the live stream. *Concept shift* describes the more operationally severe scenario in which the outcome-generating mechanism itself undergoes alteration:

$$\mathbb{P}_t(Y \mid X) \neq \mathbb{P}_{\text{tr}}(Y \mid X)$$

No re-weighting of training examples can recover Bayes-optimal performance under ([eq:concept]); retraining on post-shift observations is the only remedy. Mixed shifts—simultaneous migration in both marginal and conditional distributions—represent the hardest detection scenario and are explicitly included in our evaluation design.

Latency Constraints and Batch-Scheduling Incompatibility

For any window-based detector requiring W observations to achieve statistical power β at significance level α, and for a stream with arrival rate λ events per second, the minimum achievable detection latency is:

$$\begin{equation} \mathcal{L}_{\min} = \frac{W}{\lambda} \;\; \text{[seconds]} \label{eq:latency_floor} \end{equation}$$

A batch scheduler adds a scheduling overhead $\mathcal{L}_{\text{sched}}$ that may be hours, making the realised latency $\mathcal{L} = \mathcal{L}_{\min} + \mathcal{L}_{\text{sched}}$ operationally impractical. DriftSentinel eliminates $\mathcal{L}_{\text{sched}}$ by embedding drift computation as a co-located streaming operation, reducing $\mathcal{L}$ to approximately $\mathcal{L}_{\min}$.

DriftSentinel Architecture

The DriftSentinel system is composed of five microservices, each with a bounded responsibility. The deliberate absence of shared mutable state across service boundaries allows each stage to be scaled and upgraded independently.

Event Ingestion and Feature Preparation

Upstream producers write raw transaction records to a Kafka cluster configured with 16 partitions and replication factor 3. Two independent consumer groups read identical partition offsets: the primary scoring service and the DriftSentinel monitoring tap. Shadow-read isolation guarantees that monitoring-side processing cannot introduce backpressure onto the latency-sensitive scoring path. A stateless Flink job on the monitoring tap applies the same feature engineering transformations employed during model training—log transformations on skewed monetary quantities, cyclic encodings of temporal features, velocity-ratio derivations—ensuring that any statistical distance computed between reference and stream samples reflects genuine distributional change rather than preprocessing discrepancy.

Three-Signal Drift Detection Module

At configurable micro-batch intervals ($B = 5{,}000$ events), the detection module executes three concurrent workers:

PSI Worker: Computes the bin-divergence index for each input feature against quantile boundaries fixed at training time, reporting the maximum value across features $\text{PSI}_{\max}$.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

KS Worker: Applies the two-sample Kolmogorov–Smirnov test to each feature independently and reports the minimum corrected p-value $p_{\min}$ across the d-dimensional test family.

IForest Worker: Scores every observation in the micro-batch against a reference Isolation Forest model and reports the batch-level anomaly rate r_t.

All three workers execute via asyncio coroutines; because their computations are CPU-bound and numpy-vectorised, they run in a process pool and add negligible wall-clock overhead compared with sequential execution.

Monitoring, Alerting, and Observability

All three signal values, along with the fused severity indicator Δ (defined in Section 5), are written at each micro-batch epoch to InfluxDB and rendered on a Grafana operational dashboard with 5-second refresh. A FastAPI service exposes a /drift/status endpoint returning the current Δ and severity tier. Three tiers are defined: *Observe* ($\Delta \in [0.3, 0.6)$)—human notification only; *Warn* ($\Delta \in [0.6, 0.85)$)—escalated alert with runbook attachment; *Act* ($\Delta \geq 0.85$)—autonomous retraining dispatch. Only the Act tier initiates automated remediation.

Autonomous Retraining Pipeline

On Act-tier declaration, a retraining job is enqueued to a Kubernetes batch job pool. The training corpus is assembled by combining the most recent $N_{\text{new}} = 200{,}000$ streaming observations with a stratified 40% subsample of the original training corpus, weighting recent observations while preserving long-range distributional memory. The LightGBM model is refitted under the original hyperparameter configuration with early stopping patience of 50 evaluation rounds on a 10% held-out slice.

Shadow Validation and Model Promotion

The retrained candidate enters a 60-second shadow phase during which it scores a duplicate of live traffic without affecting served responses. Three promotion gates must clear simultaneously:

accuracy parity within $\varepsilon = 0.02$ of the incumbent on the shadow window;

PSI between incumbent and candidate prediction distributions below 0.10;

P99 inference latency no greater than 115% of the incumbent baseline.

A candidate that fails any gate is discarded and an operator alert is escalated; the incumbent continues serving.

Detection Methodology

Population Stability Index

For feature j discretised into K bins with reference proportions $\{p_k\}_{k=1}^{K}$ and monitoring proportions $\{q_k\}_{k=1}^{K}$, the PSI is:

$$\text{PSI}_j = \sum_{k=1}^{K} (q_k - p_k) \ln\left(\frac{q_k}{p_k}\right)$$

Equation ([eq:psi]) is a symmetrised relative entropy; it equals zero when reference and monitoring proportions are identical and increases monotonically with divergence. Regulatory model-risk practice associates $\text{PSI} < 0.10$ with negligible change, $0.10 \leq \text{PSI} < 0.25$ with moderate change, and $\text{PSI} \geq 0.25$ with a shift large enough to warrant immediate investigation. DriftSentinel computes PSI_j for each of the d features over rolling windows of $B = 5{,}000$ events with 90% overlap (step size 500) and reports $\text{PSI}_{\max} = \max_j \text{PSI}_j$.

Kolmogorov–Smirnov Distance

Given N_r reference observations and N_m monitoring observations for feature j, the KS distance is:

$$D_j = \sup_x |\hat{F}_{N_r,j}(x) - \hat{G}_{N_m,j}(x)|$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

where $\hat{F}$ and $\hat{G}$ are the respective empirical distribution functions. The null hypothesis of equal distributions is rejected at nominal level α when:

$$D_j > c(\alpha) \sqrt{\frac{N_r + N_m}{N_r N_m}}$$

with $c(\alpha) = \sqrt{-\ln(\alpha/2)/2}$. To control the family-wise error rate over d simultaneous per-feature tests, each individual p-value is adjusted by the Dunn–Bonferroni procedure prior to forming the signal $p_{\min} = \min_j p_{j,\text{adj}}$ at the target level $\alpha = 0.01$.

Isolation Forest Multivariate Anomaly Rate

An Isolation Forest trained on the reference feature matrix assigns each observation x a score:

$$s(x, N) = 2^{-\mathbb{E}[h(x)] / c(N)}$$

where $\mathbb{E}[h(x)]$ is the mean isolation depth across T trees and $c(N) = 2H(N-1) - 2(N-1)/N$ is the expected depth for a dataset of size N (H denotes harmonic numbers). Scores approaching 1.0 indicate atypical observations; scores near 0.5 indicate membership consistent with the training distribution. The per-batch anomaly rate is:

$$r_t = \frac{1}{B} \sum_{j=1}^{B} \mathbf{1}\left[s(x_j) > \tau_s\right]$$

with threshold $\tau_s = 0.60$ calibrated on the training reference to yield a 5% nominal false-positive rate under stationarity. This multivariate signal detects structural deformations in the joint feature distribution that produce no individually alarming univariate shift.

Weighted Evidence Fusion

The three signals are mapped to a common [0,1] range and combined as:

$$\Delta = w_1 \, \phi(\text{PSI}_{\max}) + w_2 \, (1 - p_{\min}) + w_3 \, r_t$$

where $\phi(v) = (1 + e^{-10(v-0.15)})^{-1}$ is a sigmoid normalisation of PSI values to [0,1]. The weight vector $(w_1, w_2, w_3) = (0.35, 0.40, 0.25)$ was selected by grid search over a held-out validation stream with independently scheduled drift events. The KS term carries the largest individual weight because the Bonferroni-adjusted p-value provides the sharpest discrimination at the micro-batch size $B = 5{,}000$ in our empirical calibration experiments.

Persistence Gate and Cooldown Circuit

Transient upstream irregularities—ETL pipeline bursts, network jitter, brief sensor outages—can push Δ above 0.85 for one or two consecutive micro-batches without signalling genuine distributional change. To prevent unnecessary retraining cycles triggered by such artefacts, the persistence gate requires $\Delta \geq 0.85$ on $n_p = 3$ consecutive evaluations before an Act-tier event is emitted. A mandatory cooldown interval $T_{\text{cool}} = 120$ s is also enforced after each retraining completion, during which Act-tier events are suppressed.

Input: stream of evaluations $\{\Delta_t\}$, parameters $\theta = 0.85$, $n_p = 3$, $T_{\text{cool}} = 120$ s **Initialise:** cnt $\leftarrow 0$; $t_{\text{last}} \leftarrow -\infty$ Compute Δ_t via Eq. ([eq:fusion]) cnt $\leftarrow$ cnt $+ 1$ cnt $\leftarrow 0$ Emit ACT event; dispatch retraining job $t_{\text{last}} \leftarrow t$; cnt $\leftarrow 0$

III. EXPERIMENTAL SETUP

Evaluation Dataset

Controlled evaluation of streaming drift detectors demands scenarios in which shift onset times and magnitudes are precisely known, ruling out unlabelled production captures where ground truth is unavailable. We constructed a 3.2-million-event synthetic financial transaction stream by extending the PaySim mobile-money simulator with a non-stationary regime layer that injects five programmed drift events at pre-specified timestamps:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 20, April 2026 Impact Factor: 8.2

Scenarios A and B (gradual covariate migration): the marginal distribution of two correlated monetary-amount features rotates progressively over a 30-second window, representing a seasonal spending regime transition.
Scenarios C and D (abrupt concept rotation): the fraud decision boundary undergoes a discrete reconfiguration at a single timestamp, modelling a novel fraud strategy that previously unseen in training data.
Scenario E (mixed mode): simultaneous feature-distribution migration and conditional-label change, representing a market-microstructure regime shift during a high-volatility session.
The base feature set comprises 22 variables: transaction denomination, sender and receiver account balance states, merchant category encoding, cyclic temporal indicators, and rolling behavioural velocity ratios.

Infrastructure Stack
All experiments were conducted on a five-node Docker Compose cluster:
Apache Kafka 3.5.1 (3 brokers, 16 partitions, RF = 3): event streaming backbone.
Apache Flink 1.17: stateless feature extraction and micro-batch assembly.
FastAPI 0.104 / Uvicorn: drift severity API with 4 asynchronous workers.
MLflow 2.9: experiment tracking, model registry, artefact store.
InfluxDB 2.7 + Grafana 10.2: time-series telemetry and real-time dashboarding.
Python 3.11: scipy 1.11 (KS test), scikit-learn 1.3 (Isolation Forest), lightgbm 4.1.
Host hardware: dual 8-core Intel Xeon Gold at 3.2 GHz, 64 GB DDR5, NVIDIA A10 GPU used exclusively for retraining.

Production Model Configuration
The incumbent model is a LightGBM gradient-boosted ensemble with 600 trees, maximum depth 8, and learning rate 0.04, targeting binary fraud classification. It achieves AUPRC = 0.946 on a held-out stationary evaluation partition prior to any drift injection. An LSTM-based sequence model was also prototyped but exhibited a retraining cycle time of approximately 19 minutes, far exceeding the operational budget; it was excluded from comparative evaluation.
Evaluation Metrics
Drift Detection Latency (DDL): elapsed seconds from the programmed drift onset to the first Act-tier event emission.
False-Positive Rate (FPR): fraction of Act-tier events emitted during confirmed drift-free stream intervals, expressed as a percentage.
Post-Drift Accuracy Drop (PDAD): the difference in classification accuracy between the 30-second pre-drift window and the lowest-accuracy 30-second window following onset, measured before any retraining completes.
End-to-End Response Time (EERT): elapsed seconds from Act-tier emission to the retrained model serving its first live request.

IV. RESULTS AND DISCUSSION

Detection Latency and Precision
Table 1 presents comparative performance across four systems: hourly-batch PSI monitoring (HBPM), ADWIN applied to the scalar error stream (ADWIN-ER), a streaming KS detector with fixed alert threshold (KS-Stream), and DriftSentinel.
Drift Detection Performance Across Evaluated Systems

System	DDL (s)	FPR (%)	PDAD (pp)	EERT (s)
HBPM	≥ 3,600	1.0	41.2	—
ADWIN-ER	37.8	5.4	19.1	—
KS-Stream	8.9	6.8	11.7	—

DOI: 10.48175/IJARSCT-34635

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

System	DDL (s)	FPR (%)	PDAD (pp)	EERT (s)
DriftSentinel	**4.7**	**2.3**	**6.6**	**91**

DriftSentinel achieves the lowest detection latency of 4.7 s across all five scenarios—a 47% improvement over KS-Stream (8.9 s) and an improvement of more than three orders of magnitude over HBPM. Critically, this latency advantage does not come at the cost of precision: DriftSentinel's false-positive rate of 2.3% is substantially below that of KS-Stream (6.8%) and ADWIN-ER (5.4%), demonstrating that the persistence gate absorbs the transient noise spikes that cause single-threshold detectors to fire spuriously.

Accuracy Trajectory Under Drift

In the unmonitored condition, each injected drift event produces a sharp accuracy depression; the two abrupt concept-rotation scenarios (C and D) cause the most severe degradation, with the model's AUPRC collapsing by up to 41.2 percentage points before the scenario window closes. Under DriftSentinel governance, each onset is detected within 4.7 s, retraining is dispatched, and near-baseline accuracy is recovered within the 91-second EERT budget. The residual 6.6-pp PDAD represents the accuracy exposure during the detection-to-promotion interval—a window spanning at most 35,000 transactions at the experimental arrival rate.

Ablation Study: Component Contributions

Disabling the Isolation Forest worker while retaining PSI and KS raised FPR from 2.3% to 6.9% with no material change in DDL on abrupt drift scenarios, confirming that the multivariate anomaly rate primarily contributes alarm precision rather than recall. Removing the PSI worker extended median DDL from 4.7 s to 7.3 s specifically on the gradual covariate migration scenarios (A and B); the binned-divergence mechanism of PSI is more sensitive to slow distributional migrations than the snapshot ECDF comparison of the KS test. Removing the KS worker increased FPR to 4.0% with negligible DDL impact. No two-component subset matched the full three-component system on all four evaluation metrics simultaneously.

Persistence-Gate Sensitivity Analysis

Varying n_p from 1 to 6 reveals a clear sensitivity–precision frontier. At $n_p = 1$ the system achieves DDL = 1.8 s but sustains FPR = 13.2%, producing a mean of 4.7 unnecessary retraining cycles per hour of drift-free operation. At $n_p = 5$ the false-positive rate falls to 0.7% but DDL extends to 8.8 s. The selected value $n_p = 3$ occupies the knee of this frontier, balancing detection speed against operational cost for financial transaction workloads.

Financial Application: Mixed-Mode Scenario

Scenario E—the mixed covariate-plus-concept shift modelling an intraday microstructure transition—yielded the most practically significant findings. In the unmonitored condition, the model's fraud-detection precision declined from 0.94 to 0.63 over a 14-minute observation window, producing an estimated 340 undetected fraudulent events per 10,000 processed. DriftSentinel registered the onset at 4.2 s, completed autonomous retraining in 88 s, and passed all three shadow-validation gates on the first attempt. The total exposure window was reduced to 1 minute 32 seconds—an estimated 96% reduction in fraud-exposure events relative to the unmonitored baseline.

V. CONCLUSION

This paper has reported the design, implementation, and empirical validation of DriftSentinel, a Kafka-native MLOps monitoring subsystem whose primary aim is closing the latency gap between the onset of distributional shift in a live data stream and the restoration of full model performance through autonomous retraining. By fusing three complementary detection signals—PSI, Kolmogorov–Smirnov distance, and Isolation Forest anomaly rate—under a calibrated evidence-fusion policy and protecting the trigger against transient noise through a persistence-gate circuit,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

the system achieves a median detection latency of 4.7 s, an autonomous response time of 91 s, and a false-positive retraining rate of 2.3%.

Three design lessons emerge from the experimental findings. First, heterogeneous signal fusion is non-negotiable: each individual detection component has a distinct failure mode, and the three-component ensemble is the only configuration that dominates all baseline alternatives on every reported metric. Second, the persistence gate disproportionately improves precision over recall—a worthwhile trade-off given that unnecessary retraining incurs real computational cost. Third, shadow-validated promotion is operationally essential: 7% of retraining runs failed at least one gate in our evaluation, preventing automatic serving degradation that a naive replace-on-completion policy would have introduced.

Future Research Directions

Several avenues warrant exploration beyond the scope of this paper. Online parameter adaptation—replacing periodic full retraining with incremental update rules from Hoeffding-tree or online-gradient-descent families—would eliminate the EERT latency floor at the cost of weaker convergence guarantees for non-convex hypothesis classes. Privacy-preserving federated drift monitoring, in which individual edge nodes contribute aggregate distributional statistics rather than raw observations under secure aggregation protocols , would extend DriftSentinel to regulatory environments where data-centralisation is prohibited. Replacing the static fusion weights with a meta-learning controller that updates (w_1, w_2, w_3) based on observed retraining outcomes would eliminate the manual grid-search calibration step and improve adaptability to novel drift regimes. Finally, developing calibrated proxy metrics for concept drift in the label-delayed setting—where ground-truth outcomes are unavailable for days after the corresponding feature observations—remains an open problem with substantial practical importance.

REFERENCES

[1]. D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, pp. 2503–2511, 2015.

[2]. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31–43, 2023.

[3]. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," *ACM Computing Surveys*, vol. 55, no. 6, Art. 114, 2022.

[4]. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, Art. 44, 2014.

[5]. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[6]. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

[7]. G. I. Webb, L. K. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

[8]. E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[9]. H. Mouss, D. Mouss, N. Mouss, and L. Sefouhi, "Test of Page–Hinkley, an approach for fault detection in an agro-alimentary production system," in *Proc. 5th Asian Control Conference*, 2004, vol. 2, pp. 815–818.

[10]. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM International Conference on Data Mining (SDM)*, 2007, pp. 443–448.

[11]. S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 20, April 2026
Impact Factor: 8.2

[12]. B. Yurdakul, "Statistical properties of Population Stability Index," *Risk & Decision Analysis*, vol. 7, no. 1–2, pp. 29–42, 2018.

[13]. H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," in *Proc. ICLR Workshop on Deep Generative Models*, 2018.

[14]. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422.

[15]. J. Klaise, A. Van Looveren, G. Vacanti, and A. Coca, "Alibi detect: Algorithms for outlier, adversarial and concept drift detection," *Journal of Machine Learning Research*, vol. 22, no. 147, pp. 1–7, 2021.

[16]. G. Symeonidis, E. Nerantzis, A. Kazakis, and G. A. Papakostas, "MLflow: A platform for the machine learning lifecycle," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 6, pp. 3854–3868, 2022.

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

[18]. E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in *Proc. 28th European Modelling and Simulation Symposium (EMSS)*, 2016, pp. 249–255.

[19]. P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 71–80.

[20]. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.