ARTIFICIAL INTELLIGENCE

RCDI: A Calibrated, Importance-Weighted Multi-Signal Framework for Drift Monitoring of Production Machine Learning Models

Kanishka Singhal

Department of Artificial Intelligence and Data Science, Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi, India

Published November 2026

Abstract

Models in production seldom fail with an error message. Their inputs drift, and accuracy declines quietly. Fixed-threshold drift monitors are the usual remedy, but their error rates depend on the window size and on the number of features tested. We describe the Risk-Aware Calibrated Drift Index (RCDI). For every feature it computes four distances (Kolmogorov–Smirnov, Population Stability Index, Jensen–Shannon divergence and Wasserstein-1), turns each into a calibrated p-value using a held-out reference null with a tail correction, and merges them with a Cauchy combination. Features are then weighted by model importance, and the alert threshold is set from a second, independent null sample; effect-size gating and temporal persistence are optional. In a 30-seed simulation (3 000 windows per cell), fixed PSI>0.10 raised false alarms on 92% of no-drift windows at n=100 but on none at n≥500, and uncorrected KS alarmed on 26–29% of windows at every size. Calibrated RCDI stayed at or below the nominal 1%, usually well below it. Importance weighting improved power for small shifts in relevant features (0.63 against 0.55 for Bonferroni-corrected KS at 0.25 SD, n=250) and almost completely suppressed alerts for irrelevant features (0.01–0.02 against 1.00). The same property is a weakness when drift and concept change occur together: weighted RCDI then predicted accuracy loss worse (AUROC 0.64) than unweighted RCDI (0.75) or max-PSI (0.79). On the Elec2 benchmark, weighted RCDI separated degraded from stable weeks with AUROC 0.75 for logistic regression, against 0.52 for max-PSI. We also report several design mistakes made along the way and provide code to regenerate every number.

Keywords: model drift; concept drift; distribution shift; MLOps; calibration; Cauchy combination test; production monitoring

1 Introduction

A deployed classifier keeps producing outputs whether or not the world still resembles its training data. User behaviour changes, sensors age, prices move and upstream pipelines are rewritten, and accuracy slips without any exception being raised [1]–[3]. Looking at the input distribution is the cheapest early warning, since labels often arrive late or never; this is also why label-free performance estimation and monitoring under limited labels are active topics [4]–[6].

In practice, monitors are set up with fixed thresholds, for example “alert if PSI exceeds 0.1” or “alert if the KS p-value is below 0.05”. This causes four problems (the false-alarm behaviour of drift detectors in continuous monitoring is studied in [7]). (P1) Window-size dependence: the sampling distribution of each distance statistic narrows as n grows, so a single threshold cannot suit every window size. (P2) Multiplicity: testing d features at level α gives about $1 - (1 - \alpha)^d$ false alarms per window. (P3) Relevance: a shift in a feature the model hardly uses is closer to a data-quality note than to an incident. (P4) Transience: isolated single-window blips lead to alert fatigue.

We study RCDI, a framework that tackles P1–P4 with standard statistical components. We have tried to evaluate it more critically than monitoring proposals usually are: against calibrated baselines and not only raw thresholds, with ablations, and with scenarios chosen to show where the method breaks. The contributions are the following.

1. A fully specified calibrated drift score: four distances, per-statistic empirical nulls with an exponential tail extrapolation, Cauchy combination, importance weights, and an alert threshold taken from a second, independent null sample, so that the nominal level is an input and not something tuned afterwards.
2. An evaluation over window sizes, shift types, non-Gaussian features, two drift-to-degradation regimes, persistence rules and the real Elec2 stream, using 30 seeds and confidence intervals.
3. Findings that qualify the method. The unweighted score is underpowered; importance weighting trades blindness to irrelevant features for blindness to features that become relevant; a score-level effect-size gate degenerates at large windows; and a calibrated min-p rule is a strong baseline that RCDI beats only in some settings.

2 Related work

Concept-drift detection. Recent surveys and systematic reviews of concept-drift detection [8], [9] and benchmarks of fully unsupervised detectors on real streams [10]–[12] cover a wide range of detectors. Drift is commonly categorised as abrupt, gradual, incremental or recurring. Error-based detectors need labels; RCDI instead targets label-free, feature-level monitoring.

Applied drift monitoring. Open-source drift-detection tools have been compared on practical use cases [13]; explainable profile-based drift detection [14] and joint-distribution testing at scale [15] address feature-level and multivariate drift in applied settings, and conformal-martingale methods monitor streams with sequential guarantees [16], [17]. Our building blocks are classical univariate statistics: the Kolmogorov–Smirnov statistic, the Population Stability Index, Jensen–Shannon divergence and the first-order Wasserstein distance, combined through the calibration described in the section 'The RCDI framework'.

Combining evidence. Heavy-tailed combination tests, the Cauchy combination test being the best known, aggregate dependent p-values without modelling their dependence. Recent work studies their validity and power under dependence [18], truncated and positive variants [19], [20], and behaviour for discrete data [21]. Four distances computed on the same feature are strongly dependent, which motivates this choice. Small p-values from finite resampling call for tail extrapolation, and alert thresholds can be set from held-out calibration scores in the split-conformal manner [16], [17].

Operational monitoring. Reviews of monitoring practice and MLOps tooling [1], [2], [22]–[24] and practitioner studies [3] document how monitoring is deployed and where gaps remain.

3 Problem formulation

Let $R = \{x_i\}_{i=1}^{N} \subset \mathbb{R}^d$ be reference data from a period believed stable, and $W_t = \{x_i^{(t)}\}_{i=1}^{n}$ the production window at time t. The task is to decide whether W_t departs from the reference distribution in a way that is operationally relevant. We separate three monitoring targets: input drift ($P(X)$ changes), prediction drift ($P(\hat{Y})$ changes) and performance drift (accuracy changes once labels arrive). Input drift is neither necessary nor sufficient for performance drift: a model may be robust to a shift, and concept

drift can occur with a stable P(X) [4], [8]. RCDI therefore detects input drift and is evaluated as an early-warning signal, not as a failure detector.

Success is defined by (a) a per-window false-alarm rate (FAR) close to a chosen nominal α regardless of n, (b) power for shifts that matter, and (c) insensitivity to shifts that do not.

4 The RCDI framework

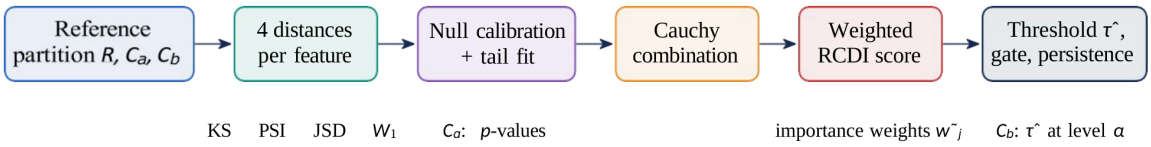

Figure 1. RCDI pipeline. The reference is split into a statistics reference R and two held-out stable partitions: Ca calibrates per-statistic p-values and Cb, independent of Ca, sets the alert threshold.

4.1 Reference partition

The reference is split into R (the distribution against which windows are compared) and two disjoint held-out stable partitions C_a, C_b. Null windows of size n are drawn with replacement from C_a or C_b (or, for autocorrelated data, as contiguous blocks). The reference is frozen during monitoring; adapting it automatically would absorb persistent drift.

4.2 Distance statistics

For feature j the window is compared with R through four statistics.

$$D_{\mathrm{KS}} = \sup_x |F_R(x) - F_W(x)| \tag{1}$$

$$\mathrm{PSI} = \sum_{i=1}^{K'} \left(p_i - q_i\right) \ln \frac{p_i}{q_i}, \; q_i = \frac{c_i + 0.5}{n + 0.5K'} \tag{2}$$

where bins are the (de-duplicated) deciles of R ($K' \le 10$), p_i the reference proportions and c_i the window counts. Jensen–Shannon divergence (base 2, hence in [0, 1]) uses the same bins:

$$\mathrm{JSD} = \frac{1}{2} D_{\mathrm{KL}}\left(P \middle\| M\right) + \frac{1}{2} D_{\mathrm{KL}}\left(Q \middle\| M\right), \; M = \frac{P+Q}{2} \tag{3}$$

In one dimension the first-order Wasserstein distance has the closed form $W_1 = \int_0^1 |Q_R(u) - Q_W(u)| \, du$, which we evaluate on a 200-point quantile grid.

4.3 Calibration with tail correction

Raw statistics live on different scales and depend on n. For each (feature, statistic) pair we draw B null windows of size n from C_a and record statistics $s_1, \ldots, s_B$. For an observed value s below the null's 90th percentile u the empirical p-value is

$$p = \frac{1 + \sum_{b=1}^{B} I\left(s_b \ge s\right)}{B + 1} \tag{4}$$

With B=1000 this cannot fall below 10^{-3}, which caps the evidence any single statistic can contribute. For $s > u$ we therefore extrapolate the tail with an exponential peaks-over-threshold model (the generalised Pareto family with shape fixed to zero):

$$p = \frac{k+1}{B+1} \exp\left(-\frac{s-u}{\sigma}\right) \tag{5}$$

where $k = \lfloor 0.1B \rfloor$ and σ is the mean excess of the k largest null values over u. Fixing the shape at zero is conservative for statistics with sub-exponential tails. Leaving it free proved unstable (see the Discussion). Implementation details: u is the (B − k)-th smallest null value; the k exceedances are the k largest null values, so the threshold observation itself is excluded from the mean excess; a statistic exactly equal to u uses the empirical branch (only strictly larger values use the tail); σ is floored at 10^{-12} in case all exceedances are tied (which happens for strongly discretised features); and every p-value is clipped to $[10^{-12}, 1]$. The procedure is run separately for each feature, statistic and window size. It is an approximate tail extrapolation, not a transformation with a proven validity guarantee.

4.4 Multi-signal aggregation

For feature j let $p_{j,m}$, $m = 1, \ldots, M$ ($M = 4$) be the calibrated p-values. The Cauchy combination

$$T_j = \frac{1}{M}\sum_{m=1}^{M} \tan\left[\pi\left(\frac{1}{2} - p_{j,m}\right)\right], \quad p_j = \frac{1}{2} - \frac{\arctan T_j}{\pi} \tag{6}$$

yields one score per feature. The validity of Cauchy-type combinations under dependence [18], [20] is established under conditions and for exact p-values; our inputs are estimated, partly extrapolated p-values, so we treat p_j as an approximate combined p-value and rely on the empirical threshold below, not on the analytic Cauchy tail, for error control. Features are then aggregated with importance weights:

$$\tilde{w}_j = \frac{w_j + \epsilon}{\sum_{k=1}^{d}\left(w_k + \epsilon\right)}, \quad \mathrm{RCDI} = \sum_{j=1}^{d} \tilde{w}_j \left(-\log_{10} p_j\right)$$

(7)

Here $w_j \geq 0$ is the importance of feature j, first normalised to sum to one, and $\epsilon = 0.02$ keeps every feature in the score.

Weights are computed once from the reference training set only, separately for each model: the absolute logistic-regression coefficient (multiplied by the feature standard deviation on Elec2, where features are not standardised) or, for gradient boosting, impurity importance. These are different estimators and the weighting is therefore model-dependent; our synthetic detection experiments use the LR weights, the GB weights appear only in the drift-to-degradation analyses (synthetic E4 and Elec2), and we do not claim that one weighting works for all architectures. Setting $\tilde{w}_j = 1/d$ gives the unweighted variant RCDI-U; the weighted variant is RCDI-W. Because RCDI is a weighted mean of log-p values rather than a p-value, its null distribution is not known analytically; this is why the threshold is calibrated empirically.

4.5 Alert threshold

A second null sample of B_2 windows from C_b, independent of the first, is scored with the already-fixed p-value calibration. The threshold is its $(1 - \alpha)$ empirical quantile $\hat{\tau}$ and a window alerts if RCDI > $\hat{\tau}$.

Remark (guarantee versus implementation). Let $\hat{\tau}$ be the $\lceil (1 - \alpha)(B_2 + 1) \rceil$-th smallest null score and alert only if the window score is strictly larger. If the window score under no drift is exchangeable with the B_2 null scores, the standard conformal rank argument (cf. [16], [17]) gives $P(\text{RCDI} > \hat{\tau}) \leq \alpha$ in finite samples. Our implementation instead uses a linearly interpolated empirical quantile (NumPy default) and null windows resampled with replacement from C_b, so that argument does not apply exactly; the false-alarm control we report is empirical, not a finite-sample guarantee, and it can fail if the reference is non-representative or autocorrelated.

4.6 Effect-size gate

Statistical significance is not operational significance. The effect size is the standardised mean difference $\Delta_j = |\mu_{W,j} - \mu_{R,j}|/(\sigma_{R,j} + \epsilon)$ with moments of R, and $G_j = I(\Delta_j > \delta)$, $\delta = 0.15$. Gates are applied after p-value calibration. The score gate (SG) replaces the score by $\text{RCDI}^G = \Sigma_j \, \tilde{w}_j \, G_j \, (-\log_{10} p_j)$; its threshold $\hat{\tau}^G$ is the $(1 - \alpha)$ quantile of this gated score over the same B_2 null windows from C_b. The AND gate (AG) uses the ungated threshold and alerts only if RCDI > $\hat{\tau}$ and $\max_j \Delta_j > \delta$, so its false-alarm rate cannot exceed that of ungated RCDI-W.

4.7 Temporal persistence

A (k, w) rule alerts at time t if at least k of the last w windows exceed $\hat{\tau}$. This suppresses isolated blips at the cost of delay.

4.8 Cost

Calibration is done once per reference and window size and costs $(B + B_2)\, d\, O(n \log n)$. Scoring a window costs $d\, O(n \log n + M \log B)$: on one CPU core (d=6) calibration takes 1.2–2.4 s per window size (B=1000, B2=2000) and scoring takes 0.4–1.0 ms per window for n from 100 to 1000 in our unoptimised NumPy implementation.

5 Experimental protocol

Synthetic data. $d = 6$ independent N(0, 1) features, with $y = I(2x_0 + 1.5x_1 + x_2 + 0.3x_0x_1 > 0)$ and 5% label noise. Features x_3, x_4, x_5 are irrelevant by construction. Logistic regression (LR) and gradient boosting (GB, 100 trees, depth 3) from scikit-learn are trained on $N = 4000$ reference points; importance weights are |LR coefficients| or GB impurity importance. The held-out partitions C_a, C_b have 2000 points each. Windows have $n \in \{100, 250, 500, 1000\}$; $B = 1000$, $B_2 = 2000$, $\alpha = 0.01$. Every cell below pools 30 seeds × 100 windows (3 000 windows; the 95% Wilson interval for a rate of 0.01 is about ±0.4 points). Each seed re-draws the reference, models and calibration.

Scenarios. E1: no drift (also with Student-t_3 and lognormal features). E2: mean shift or scale change of the relevant feature x_0. E3: shift of the irrelevant feature x_5. E4: drift versus degradation (below). E5: simulated streams.

Baselines. PSI>0.10 and PSI>0.25 on any feature; KS at $\alpha = 0.05$ on any feature (uncorrected, as often deployed); KS with Bonferroni correction at α/d using the asymptotic distribution; KS-cal., the same Bonferroni rule with the empirical null; and Cauchy-min, which Cauchy-combines the four calibrated statistics per feature and alerts if $\min_j p_j < \alpha/d$. The last two are the fair comparators for RCDI because they share its calibration. Detectors alert if any feature breaches.

E4 (drift vs. degradation). Windows of $n = 500$ are labelled degraded if accuracy falls more than 5 points below the model's reference accuracy. Whether input drift can predict degradation depends on how the world changes, so we simulate two regimes. In independent drift, covariate shift (1–2 random features, 0.25–2 SD) and concept drift (random perturbation of the label coefficients) occur independently. In coupled drift, covariate shift in 1–2 features is, in half of the windows, accompanied by a coefficient change on the same features proportional to the shift size, including on irrelevant features. Scores are compared by AUROC with 1000-sample bootstrap intervals over 9 000 windows per regime.

E5 (persistence). Streams of 200 windows ($n = 250$) with a 0.25 SD shift of x_0 beginning at window 100: sudden, gradual (a growing fraction of shifted samples over 50 windows), incremental (a growing mean shift over 50 windows) and recurring (20 windows on, 20 off). 600 streams per type.

Real data (Elec2). The NSW electricity benchmark (Elec2; file elec.csv from the public repository github.com/scikit-multiflow/streaming-datasets, originally compiled by M. Harries, 1999) (45 312 half-hour records, six features, label: price up/down relative to a moving average), a widely used stream with real drift whose timing is not documented. Rows 1–3000 form the reference and training set; rows 3001–6000 form Ca, Cb (1500 rows each); null windows are contiguous blocks of 336 rows (one week). The remaining stream is scored in 117 consecutive weekly windows. Because drift times are unknown, we report alert rates and how well each score tracks per-window accuracy loss (Spearman ρ; AUROC for a 5-point loss), with moving-block bootstrap intervals (block length 8) to respect autocorrelation.

Reproducibility. All results are produced by run_synth.py, run_real.py and aggregate.py with fixed seeds; the tables and plot coordinates in this paper are generated from the raw outputs.

6 Results

6.1 False alarms

Table 1 and Fig. 2 address RQ1. The PSI>0.10 rule depends strongly on window size: it fires on 92% of no-drift windows at n=100 and on none at n≥500. Uncorrected KS at α=0.05 fires on 26–29% of windows at every size, which matches the $1 - 0.95^6 = 0.26$ expected for six features. The Bonferroni-corrected and calibrated detectors do not exceed the nominal level: KS-Bonf. sits at 1.1–1.2%, and the calibrated detectors stay between 0.0 and 1.1%. RCDI is conservative at larger windows (0.001–0.003 for RCDI-W and 0.000–0.002 for RCDI-U at n ≥ 250), and the picture is the same for t_3 and lognormal features. This conservatism costs power (see 'Power for relevant-feature drift'). A zero in the table is not a zero true rate: with 3 000 windows per cell the 95% Wilson upper bound for no alarms is 0.13%, and for rates near 1% the interval is about ±0.4 points. The standard deviation of the RCDI-W false-alarm rate across seeds is 0.3– 0.7 percentage points, which reflects the noise in a threshold estimated from 2000 null windows.

Table 1. No-drift false-alarm rate (nominal α = 0.01). Top: Gaussian features. Bottom: heavy-tailed (t3) and skewed (lognormal) features.

n	PSI>0.10	PSI>0.25	KS (α=.05)	KS-Bonf.	KS-cal.	Cauchy-min	RCDI-U	RCDI-W
100	0.918	0.024	0.267	0.011	0.006	0.007	0.010	0.007
250	0.028	0.000	0.286	0.011	0.005	0.005	0.002	0.003
500	0.000	0.000	0.268	0.011	0.005	0.003	0.000	0.003
1000	0.000	0.000	0.258	0.012	0.004	0.004	0.000	0.001
t3, 100	0.922	0.026	0.261	0.008	0.006	0.011	0.005	0.009
t3, 500	0.000	0.000	0.256	0.010	0.002	0.004	0.000	0.001
logn., 100	0.922	0.023	0.276	0.012	0.006	0.008	0.008	0.009
logn., 500	0.000	0.000	0.258	0.009	0.004	0.005	0.000	0.003

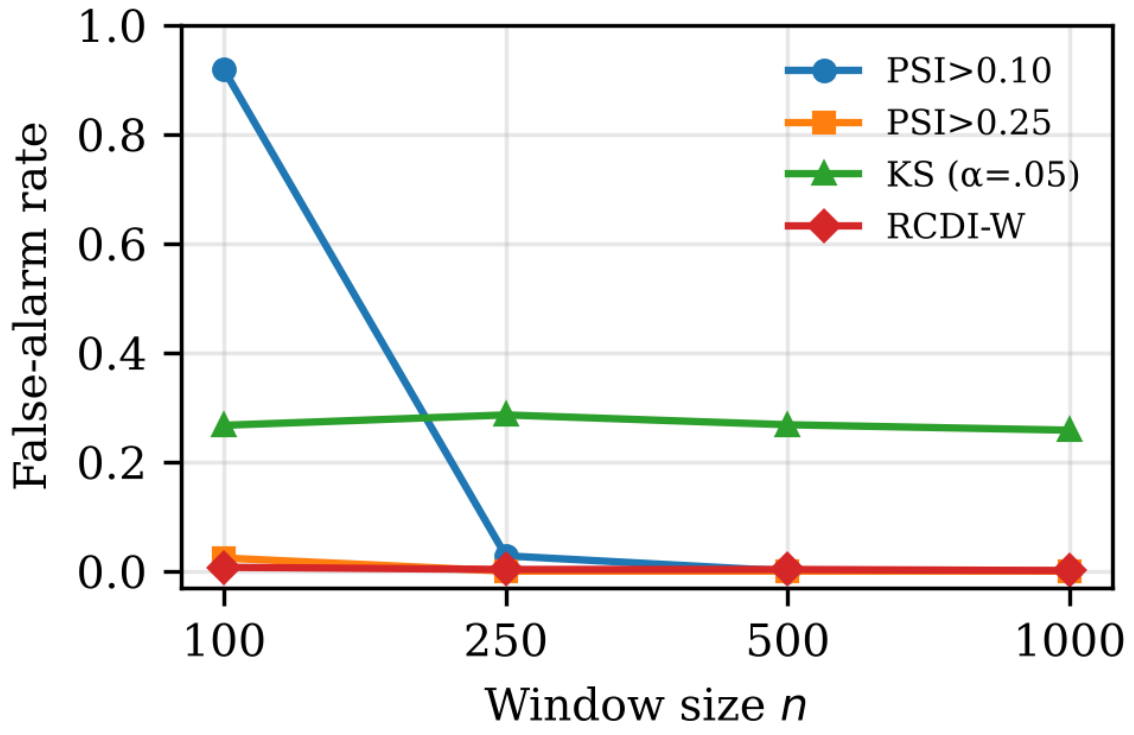

Figure 2. No-drift false-alarm rate versus window size (nominal 0.01).

6.2 Power for relevant-feature drift

At n=250 (Table 2) the calibrated detectors are indistinguishable for shifts of 0.5 SD and larger. Differences appear for small shifts and for changes of shape. For a 0.25 SD shift RCDI-W detects 0.63 of windows, against 0.55 for Bonferroni KS and 0.53 for Cauchy-min. The advantage comes from the importance weights, since RCDI-U, which differs only in using uniform weights, detects just 0.21. We read this as dilution: with d=6 and uniform weights, one drifted feature contributes at most 1/6 of the score and competes with the noise of the other five. For a scale change of ×1.25 the four-statistic combination matters: KS detects 0.07, Cauchy-min 0.26 and RCDI-W 0.38. JSD and W1 react to changes in spread, to which the sup-norm KS statistic is relatively insensitive. Fig. 3 plots these rates. At n=1000 every calibrated detector except RCDI-U reaches 1.00 for shifts of 0.25 SD or more, so the advantage of weighting disappears. For a 0.10 SD shift, Bonferroni KS detects 0.23 of windows, Cauchy-min 0.16, RCDI-W 0.15 and RCDI-U 0.01. For the ×1.25 scale change, Cauchy-min and RCDI-W detect 0.99 against 0.65 for KS-Bonf. The conclusions we keep are therefore modest. Weighting helps for small shifts in small windows, using several statistics helps for scale drift, and a calibrated min-p rule comes within 10–12 points of RCDI-W in the hardest n=250 rows and matches it at n=1000.

Table 2. Alert rate at n = 250. Rows 1–6: drift in the relevant feature x0 (higher is better). Rows 7–8: drift in the irrelevant feature x5.

Condition	PSI>0.25	KS-Bonf.	KS-cal.	Cauchy-min	RCDI-U	RCDI-W	RCDI-W+AG
x0 shift 0.10 SD	0.00	0.04	0.02	0.02	0.01	0.04	0.03
x0 shift 0.25 SD	0.00	0.55	0.42	0.53	0.21	0.63	0.63
x0 shift 0.50 SD	0.59	1.00	1.00	1.00	0.99	1.00	1.00

Condition	PSI>0.25	KS-Bonf.	KS-cal.	Cauchy-min	RCDI-U	RCDI-W	RCDI-W+AG
x0 shift 1.00 SD	1.00	1.00	1.00	1.00	1.00	1.00	1.00
x0 scale ×1.25	0.00	0.07	0.03	0.26	0.09	0.38	0.09
x0 scale ×1.5	0.34	0.64	0.45	0.99	0.89	1.00	0.21
x5 shift 0.50 SD	0.64	1.00	1.00	1.00	0.99	0.01	0.01
x5 shift 2.00 SD	1.00	1.00	1.00	1.00	1.00	0.02	0.02

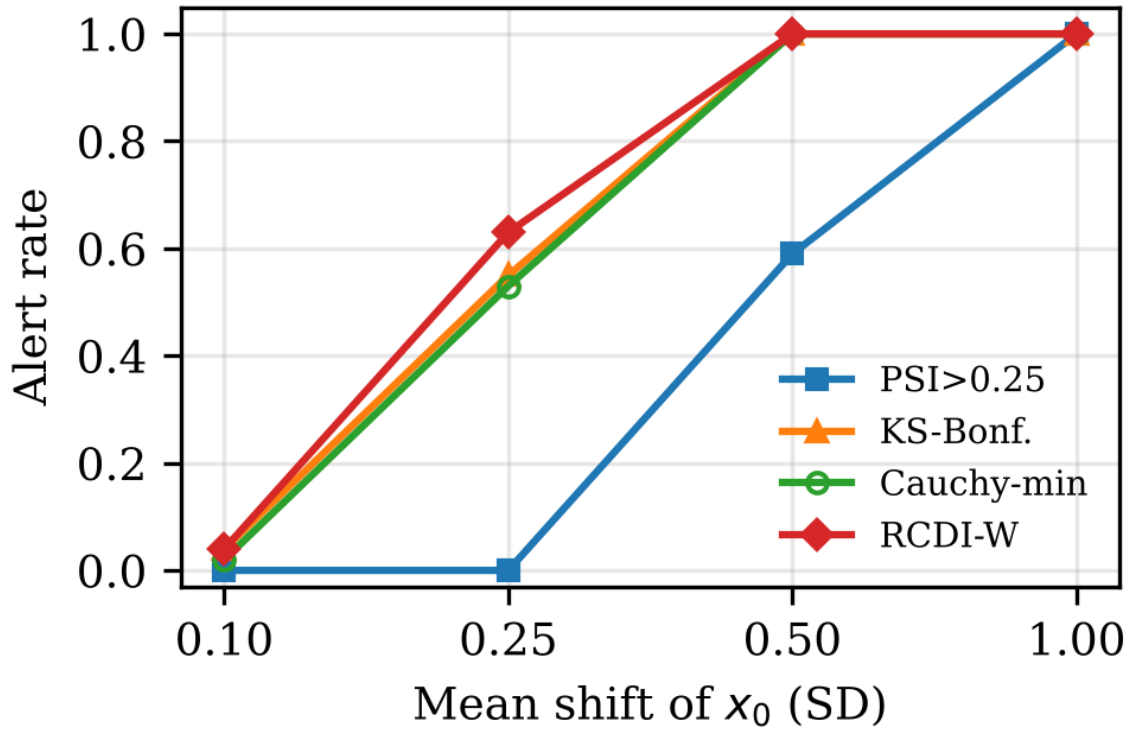

Figure 3. Alert rate versus mean shift of the relevant feature (n = 250).

6.3 Irrelevant-feature drift and ablation

Rows 7–8 of Table 2 address RQ3 on the suppression side. Every unweighted detector alerts on essentially all windows when the irrelevant feature x5 drifts, including the calibrated KS (1.00). RCDI-W alerts on only 1–2% of windows, because x5 has only ϵ-level weight. This is the intended risk-aware behaviour, but it is a design choice with a cost: a drift that does not matter to the model also goes unseen when it points to a broken upstream pipeline.

Table 3 isolates the components. The weights account for the jump from RCDI-U to RCDI-W (0.21 → 0.63 at 0.25 SD, n=250). The score gate (SG) raises power for the 0.25 SD shift at n=250 (0.80), but it degenerates at n=1000: the gated score is exactly zero for almost every null window, the calibrated threshold collapses to zero, and any feature that passes the gate triggers an alert, including the irrelevant x5 (0.97). The AND gate (AG) behaves better. It leaves detection of mean shifts unchanged and removes statistically significant but small shifts (0.10 SD at n=1000: 0.15 → 0.05). Its price is deliberate blindness to scale changes (power for ×1.25 drops from 0.38 to 0.09 at n=250 and from

0.99 to 0.00 at n=1000), so δ and the choice of effect-size measure should follow operational requirements.

Table 3. Ablation. SG: score gate; AG: AND gate (δ = 0.15).

n	Method	FAR	x0 0.25	x0 ×1.25	x5 0.50
250	KS-cal.	0.005	0.42	0.03	1.00
	Cauchy-min	0.005	0.53	0.26	1.00
	RCDI-U	0.002	0.21	0.09	0.99
	RCDI-W	0.003	0.63	0.38	0.01
	RCDI-W+SG	0.005	0.80	0.05	0.01
	RCDI-W+AG	0.001	0.63	0.09	0.01
1000	KS-cal.	0.004	0.99	0.34	1.00
	Cauchy-min	0.004	1.00	0.99	1.00
	RCDI-U	0.000	0.79	0.59	1.00
	RCDI-W	0.001	1.00	0.99	0.00
	RCDI-W+SG	0.000	1.00	0.00	0.97
	RCDI-W+AG	0.000	1.00	0.00	0.00

6.4 Drift versus performance degradation

Table 4 addresses RQ4, and the answer depends on the regime. When covariate and concept drift occur independently, no score carries information about degradation (AUROC 0.48–0.49, at or marginally below chance). Degradation there comes from concept drift that leaves P (X) untouched, which an input monitor cannot see, and prediction drift is blind to it as well. When drift is coupled to concept change, input scores become informative. Max-PSI does best (0.79 for LR), unweighted RCDI is next (0.75), and weighted RCDI is clearly worse (0.64). The reason is the same property that helped on irrelevant features: coupled drift in the three irrelevant features makes them relevant, and the LR-derived weights deliberately ignore them. Prediction drift is also weak (AUROC 0.58), because many damaging changes (coefficient changes on shifted irrelevant features) alter predictions little. Combining weighted RCDI with prediction drift by rank averaging does not recover the loss.

Table 4. AUROC (95% bootstrap interval) of drift scores for predicting a >5-point accuracy loss (n = 500 windows).

Score	Independent drift: LR	Independent drift: GB	Coupled drift: LR	Coupled drift: GB
Unweighted RCDI	0.49 (0.47–0.50)	0.49 (0.48–0.50)	0.75 (0.74–0.76)	0.75 (0.74–0.76)
Weighted RCDI	0.48 (0.47–0.49)	0.48 (0.47–0.49)	0.64 (0.63–0.65)	0.63 (0.62–0.64)
Max feature – log10 p	0.48 (0.47–0.50)	0.49 (0.48–0.50)	0.76 (0.76–0.77)	0.76 (0.76–0.77)
Max PSI	0.48 (0.47–0.49)	0.49 (0.48–0.50)	0.79 (0.78–0.80)	0.81 (0.80–0.82)

Score	Independent drift: LR	Independent drift: GB	Coupled drift: LR	Coupled drift: GB
Weighted W1 effect size	0.48 (0.46–0.49)	0.48 (0.47–0.50)	0.68 (0.67–0.69)	0.67 (0.66–0.68)
Prediction drift (KS)	0.48 (0.47–0.49)	0.48 (0.47–0.50)	0.58 (0.57–0.60)	0.60 (0.58–0.61)
Weighted RCDI + prediction drift	0.48 (0.47–0.49)	0.48 (0.47–0.50)	0.62 (0.61–0.63)	0.62 (0.61–0.63)
Degraded windows	44%	43%	35%	34%

6.5 Temporal persistence

Persistence rules trade delay for fewer false alerts (Table 5). The single-window rule fires on 185–218 of 57 600 pre-onset windows (0.32–0.38%), the 2-of-3 rule on 1–5, and the 3-of-5 rule on none. Zero observed alerts is not a zero rate: by the rule of three the 95% upper bound is about 0.005% per window, and only approximately so because consecutive windows share rule memory. Delay grows with the rule; for sudden drift it is 0.6, 2.4 and 4.3 windows. For gradual and incremental drift the delay is dominated by the ramp (25, 38 and 42 windows). Recall does not fall with persistence, contrary to the usual expectation, because the single-window power here is 0.63, above one half, so a k-of-w vote amplifies it (0.63 → 0.68 → 0.70 for sudden drift); at a power below 0.5 it would reduce recall. For recurring drift the episodes are short (20 windows) and longer rules lose ground (0.63, 0.63, 0.61).

Table 5. Persistence rules on simulated streams of 200 windows (n = 250, 0.25 SD shift of x0; 600 streams per type).

Drift	Rule	FAR (%)	Detected	Delay	Recall
Sudden	Single	0.35	1.00	0.6	0.63
	2-of-3	0.00	1.00	2.4	0.68
	3-of-5	0.00	1.00	4.3	0.70
Gradual	Single	0.32	1.00	25.2	0.63
	2-of-3	0.00	1.00	37.6	0.68
	3-of-5	0.00	1.00	42.3	0.71
Incremental	Single	0.38	1.00	25.3	0.62
	2-of-3	0.01	1.00	37.7	0.68
	3-of-5	0.00	1.00	42.5	0.71
Recurring	Single	0.37	1.00	0.6	0.63
	2-of-3	0.01	1.00	2.5	0.63
	3-of-5	0.00	1.00	4.6	0.61

FAR: pre-onset false alerts per window (57 600 windows per row). Detected: share of streams with any alert after onset. Delay: windows from onset to first alert (detected streams). Recall: share of fully drifted windows flagged.

6.6 Real data: Elec2

Elec2 is strongly non-stationary from the start (Table 6): PSI and Bonferroni KS alert on every window, including the first ten, and 74% of weeks lower the accuracy of the LR model by more than 5

points. The alert rates therefore cannot be read as false-alarm rates, since there is no ground truth. What we can assess is whether a score ranks windows by the damage they cause (Table 7). Weighted RCDI separates degraded from stable weeks with AUROC 0.75 for LR and 0.80 for GB, and its Spearman correlation with accuracy loss is 0.36 (LR) and 0.69 (GB). Max-PSI and the weighted Wasserstein effect size are near chance for LR (AUROC 0.52 and 0.47). The intervals are wide (117 autocorrelated windows, one dataset), and RCDI-U cannot be distinguished from RCDI-W here. The data therefore support the claim that calibrated multi-statistic scores track degradation better than raw PSI on real drift, but not that importance weighting helps. The short reference (about two months) also cannot represent the seasonal variation present later in the stream.

Table 6. Elec2: fraction of the 117 weekly windows on which each detector alerts (all windows; the first ten after the reference).

Detector	All windows	First 10
PSI>0.10	1.00	1.00
PSI>0.25	1.00	1.00
KS-Bonf.	1.00	1.00
KS-cal.	0.28	0.00
Cauchy-min	0.63	0.80
RCDI-U	0.85	1.00
RCDI-W	0.80	0.90
RCDI-W+AG	0.80	0.90

Table 7. Elec2: association of each score with per-window accuracy loss (moving-block bootstrap 95% intervals).

Score	Spearman ρ: LR	Spearman ρ: GB	AUROC: LR	AUROC: GB
Unweighted RCDI	0.38 (0.22–0.57)	0.56 (0.36–0.74)	0.76 (0.65–0.89)	0.72 (0.59–0.88)
Weighted RCDI	0.36 (0.20–0.54)	0.69 (0.52–0.79)	0.75 (0.65–0.87)	0.80 (0.69–0.91)
Max PSI	-0.01 (-0.19–0.25)	0.32 (-0.03–0.58)	0.52 (0.42–0.67)	0.60 (0.40–0.79)
Weighted W1 effect size	-0.11 (-0.33–0.16)	0.33 (0.02–0.58)	0.47 (0.33–0.66)	0.59 (0.38–0.78)
Prediction drift (KS)	0.27 (0.10–0.47)	0.62 (0.39–0.78)	0.71 (0.60–0.84)	0.76 (0.59–0.87)

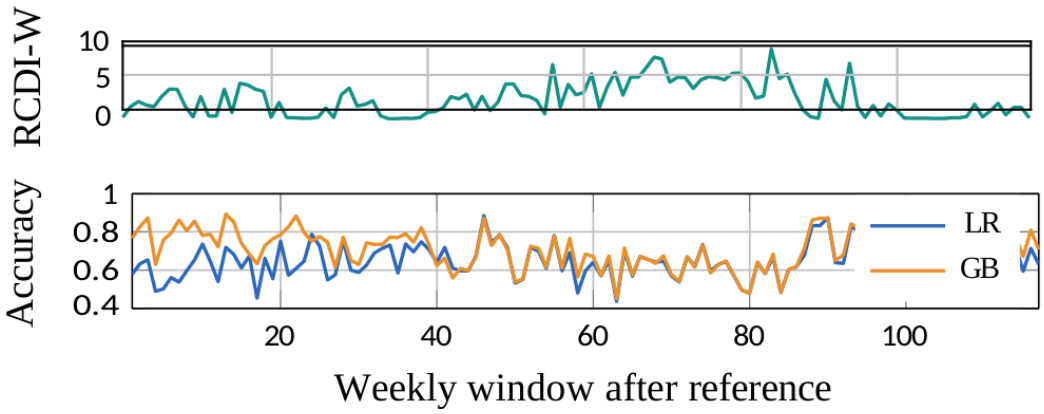

Figure 4. Elec2: weighted RCDI (top) and model accuracy (bottom) per weekly window.

7 Discussion

What the evidence supports. First, fixed thresholds on any-feature statistics are unreliable across window sizes and feature counts, and being standard does not make them safe: PSI>0.10 and uncorrected KS fail in opposite directions. Second, any calibrated detector fixes this. Calibration, and not the particular multi-signal score, is the main source of false-alarm control, and a calibrated min-p rule is close to RCDI-W in power throughout and equal at large windows. Third, combining four statistics helps for drift that is not a location shift. Fourth, importance weighting is a policy decision that moves power towards the features the model uses.

Design mistakes. We list four so that others can avoid them.

(a) A p-value floor of 1/(B+1) caps the − log10 p of each feature at about 3, and a mean over features then cannot fire on a single drifted feature, so tail extrapolation is necessary and not cosmetic. (b) In a preliminary run with a free-shape GPD tail fit, the calibrated threshold of the unweighted score varied between about 1.2 and 2.6 across seeds, and power for a 1 SD shift at n=1000 ranged from 4% to 100% depending on the seed. We attribute this to poorly determined shape estimates from about 100 exceedances (we did not isolate the mechanism further); fixing the shape at zero removed the instability. (c) Gating the score (SG) calibrates a threshold on a score that is zero for almost every null window, so the threshold collapses to zero at large n (see Table 3). (d) Gates based on the mean are blind to scale changes by construction.

Practical guidance. Calibrate thresholds, choose α per window explicitly, and report the per-window rate. Use importance weighting only when the features that drive risk are known and stable, and keep an unweighted or min-p channel for data-quality alerts. Treat input drift as an early warning and confirm it with delayed labels. An alert should name the feature, metric, magnitude, window size, model version and reference identifier.

8 Limitations

Our synthetic experiments use independent Gaussian features and a known generator, and the E4 result depends on the assumed coupling, which we vary but do not estimate from real systems. We use a single real dataset with 117 autocorrelated windows and no ground-truth drift dates; benchmark collections [11], [25] would allow a broader evaluation. Marginal per-feature tests cannot

see changes in dependence structure, for which joint-distribution tests [15] are complementary. Calibration assumes a stable, representative and exchangeable reference; the false-alarm guarantee is approximate and is not established for autocorrelated streams or contaminated references. A weighted mean of log-p values has no analytic null and depends on d; we have not studied $d \gg 10$, where max or top-k aggregation is probably preferable. Importance weights from correlated predictors are unstable, persistence rules were evaluated at one shift magnitude, and we did not compare against recent unsupervised stream detectors [10], [12] or conformal monitoring methods [16], [17].

9 Conclusion

RCDI turns four distribution distances into a threshold-calibrated and optionally risk-weighted drift score. Calibration gives what fixed thresholds cannot: an empirical false-alarm rate at or below the chosen level at every window size, including for non-Gaussian features. Importance weighting cuts both ways. It improves power for relevant-feature drift and suppresses irrelevant alerts, yet it predicts accuracy loss worse when new features become relevant. The evidence from one real drift stream is encouraging but limited. Future work should use further real streams with labelled drift events, multivariate statistics, top-k aggregation and online recalibration.

References

[1] H. Naveed, S. Barnett, C. Arora, J. Grundy, H. Khalajzadeh, and O. Haggag, "Monitoring machine learning systems: A multivocal literature review," *arXiv:2509.14294*, 2025.

[2] J. Leest, I. Gerostathopoulos, P. Lago, and C. Raibulet, "Monitoring and observability of machine learning systems: Current practices and gaps," *arXiv:2510.24142*, 2025.

[3] H. Naveed, J. Grundy, C. Arora, H. Khalajzadeh, and O. Haggag, "Understanding practitioners' *perspectives on monitoring machine learning systems*," arXiv:2509.25195, 2025.

[4] J. Białek, W. Kuberski, N. Perrakis, and A. Bifet, "Estimating model performance under covariate shift without labels," *arXiv:2401.08348*, 2024.

[5] T. Flühmann, A. Bissoto, T.-D. Hoang, and L. M. Koch, "Label-free estimation of clinically relevant performance metrics under distribution shifts," *arXiv:2507.22776*, 2025.

[6] A. Koebler, T. Decker, I. Thon, V. Tresp, and F. Buettner, "Incremental uncertainty-aware performance monitoring with active labeling intervention," in *Proc*. 28th Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR vol. 258, 2025.

[7] R. S. Singh, "When drift detectors cry wolf: False alarm rates in continuous ML monitoring," *arXiv:2607.17336*, 2026.

[8] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: Detecting concept drift," *Front.* Artif. Intell., vol. 7, art. 1330257, 2024, doi: 10.3389/frai.2024.1330257.

[9] G. Hovakimyan and J. M. Bravo, "Evolving strategies in machine learning: A systematic review of concept drift detection," *Information*, vol. 15, no. 12, art. 786, 2024, doi: 10.3390/info15120786.

[10] D. Lukats, O. Zielinski, A. Hahn, and F. T. Stahl, "A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams," *Int.* J. Data Sci. Anal., vol. 18, no. 3, 2024, doi: 10.1007/s41060-024-00620-y.

[11] V. Cerqueira, H. M. Gomes, M. Heyden, B. Pfahringer, and A. Bifet, "A framework for evaluating and benchmarking concept drift detection methods," in *Proc*. 32nd ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD), Jeju Island, South Korea, Aug. 2026.

[12] M. M. U. H. Khan and S. Sadaoui, "Learner-based concept drift detection: Analysis and evaluation," *arXiv:2606.20216*, 2026.

[13] R. Müller, M. Abdelaal, and D. Stjelja, "Open-source drift detection tools in action: Insights from two use cases," *arXiv:2404.18673*, 2024.

[14] U. Dara and M. Cavus, "From XAI to MLOps: Explainable concept drift detection with profile drift detection," *arXiv:2412.11308*, 2024.

[15] Ç. Pullu, M. E. Arslan, B. Balkaç, A. Ö. Balta, C. C. Palaçı, F. C. Yılmaz, and A. Cakir, "Beyond marginal monitoring: Distributed joint-distribution testing for data concept drift in large scale e-commerce operations," *arXiv:2610.08132*, 2026.

[16] D. Prinster, X. Han, A. Liu, and S. Saria, "WATCH: Adaptive monitoring for AI deployments via weighted-conformal martingales," in *Proc*. 42nd Int. Conf. Machine Learning (ICML), PMLR vol. 267, 2025.

[17] T. Liu, D. Prinster, and F. Orabona, "Monitoring for shifts at unknown times: Parameter-free conditional conformal martingales," *arXiv:2610.17343*, 2026.

[18] L. Gui, T. Mao, J. Wang, and R. Wang, "Validity and power of heavy-tailed combination tests under asymptotic dependence," *arXiv:2508.05818*, 2025.

[19] B. Chen, W. Xu, and X. Gao, "Truncated Cauchy combination test: A robust and powerful p-value combination method with arbitrary correlations," *arXiv:2506.12489*, 2025.

[20] Y. Ouyang, X. Liu, L. Zhu, and W. Xu, "Effective positive Cauchy combination test," *arXiv:2410.10345*, 2024.

[21] H. Alsulami and S. Liverani, "Evaluating the Cauchy combination test for count data," *arXiv:2504.14552*, 2025.

[22] B. Eken, S. Pallewatta, N. K. Tran, A. Tosun, and M. A. Babar, "A multivocal review of MLOps practices, challenges and open issues," *arXiv:2406.09737*, 2024.

[23] L. Berberi, V. Kozlov, G. Nguyen et al., "Machine learning operations landscape: Platforms and tools," *Artif*. Intell. Rev., vol. 58, art. 167, 2025, doi: 10.1007/s10462-025-11164-3.

[24] Y. J. Hu, J. Rombouts, and I. Wilms, "MLOps monitoring at scale for digital platforms," *arXiv:2504.16789*, 2025.

[25] D. Soukup, R. Plny, D. Vašata, and T. Čejka, "Drift-based dataset stability benchmark," *arXiv:2512.23762*, 2025.