When Your Model Stops Working: Anytime-Valid Calibration Monitoring

Tristan Farran
MSc Computational Science, University of Amsterdam
tristan.farran@student.uva.nl

February 24, 2026

Abstract

Deployed machine learning models often experience calibration drift as the world changes. To address this challenge, we present PITMonitor, a sequential method for detecting calibration changes in probabilistic regression models. Unlike traditional calibration tests that require fixed evaluation windows, PITMonitor provides *anytime-valid* false alarm control: the probability of ever raising a spurious alarm is bounded by α, regardless of when monitoring stops. We prove Type I error control via Ville's inequality and demonstrate detection power on three synthetic drift scenarios from `river`'s FriedmanDrift benchmark, comparing against seven stream-drift baselines while maintaining valid false alarm control. Code is available at `https://github.com/tristan-farran/pitmon`.

1 Introduction

Probabilistic models deployed in production face a fundamental challenge: the world changes. Models encounter changes in input distributions, label frequencies, the relationship between features and targets, and so on. When these shifts occur, model calibration can degrade and simple calibration techniques may no longer suffice.

Detecting degrading miscalibration is critical for maintaining trustworthy AI systems. A medical diagnostic model that becomes overconfident after a sensor upgrade, or a financial risk model that underestimates tail probabilities after a regime shift, can lead to consequential errors. In practice, calibration is often monitored using ad-hoc procedures such as periodic recalibration schedules, rolling-window hypothesis tests, threshold-based alerts on summary metrics, or manual inspection of residuals.

These approaches suffer from a fundamental statistical problem: *they do not control the false alarm rate over continuous monitoring.* A practitioner who checks calibration daily with a $p < 0.05$ threshold will, over a year of monitoring, almost certainly observe spurious alarms even if the model remains stable. Classical hypothesis tests assume a fixed sample size determined before seeing data; continuous monitoring violates this assumption.

More principled alternatives are provided by online drift detectors, such as those implemented in the `river` library [Montiel et al., 2020]. Classical detectors including DDM, EDDM, and KSWIN are lightweight, easy to deploy, and effective at detecting abrupt changes, but are typically based on heuristic thresholds or fixed-sample statistical arguments and do not provide explicit long-run false alarm guarantees under continuous monitoring.

ADWIN employs sequential hypothesis testing with anytime-valid concentration bounds, offering formal control of the probability of false alarms over an unbounded stream [Bifet and Gavaldà, 2007]. However, ADWIN operates on generic performance statistics such as squared residuals and as a result only partially addresses the problem of reliable, long-term calibration monitoring.

We propose PITMonitor, a method providing *anytime-valid* monitoring of PIT exchangeability with four key properties:

1. **Anytime-valid false alarm control:** we prove that $\mathbb{P}(\text{ever alarm} \mid H_0) \leq \alpha$, regardless of when or why monitoring stops.
2. **Change detection without static-error alarms:** PITMonitor detects and locates *changes* in the PIT process. A model that is consistently miscalibrated but stable will not trigger alarms,[1] while a changing process can.
3. **No baseline period required:** unlike methods requiring a "clean" reference distribution, PITMonitor works from the first observation by testing exchangeability of the PIT sequence.
4. **Practical efficiency:** the exact algorithm runs in $O(t \log t)$ time and $O(t)$ space for t observations, with a simple recursive update.

2 Background

2.1 Calibration and Probability Integral Transforms

A probabilistic model outputting a predicted cumulative distribution function $\hat{F}$ over outcomes is *calibrated* if these predictions match reality: among all predictions where $\hat{F}(y) = p$, the outcome $Y \leq y$ should occur roughly $(100 \times p)\%$ of the time.

The *probability integral transform* (PIT) provides a universal tool for assessing calibration [Dawid, 1984]. For a continuous predictive CDF F and realized outcome y, the PIT is $U = F(y)$. A classical result states that if F is the true distribution of Y, then $U \sim \text{Uniform}(0, 1)$.

In the regression setting where the model outputs a Gaussian predictive distribution $\mathcal{N}(\mu_t, \sigma_t^2)$, the PIT is:

$$U_t = \Phi\left(\frac{y_t - \mu_t}{\sigma_t}\right) \tag{1}$$

where Φ denotes the standard normal CDF. Under perfect calibration this gives $U_t \sim \text{Uniform}(0, 1)$.

For discrete outcomes (e.g., classification), randomization yields a continuous PIT [Brockwell, 2007]. Given predicted class probabilities $(\hat{p}_1, \ldots, \hat{p}_K)$ and true class $y \in \{1, \ldots, K\}$:

$$U = \sum_{j=1}^{y-1} \hat{p}_j + V \cdot \hat{p}_y, \quad V \sim \text{Uniform}(0, 1) \tag{2}$$

[1] In many domains some amount of miscalibration is inevitable, but model degradation remains a pressing concern.

placing U uniformly within the cumulative probability interval corresponding to the true class. Both formulations produce $U_t \sim \text{Uniform}(0, 1)$ under calibration stability, and the theoretical guarantees of PITMonitor apply to any method satisfying this property.

2.2 Exchangeability

A sequence $(X_1, X_2, \ldots)$ is *exchangeable* if its joint distribution is invariant to finite permutations. Exchangeability is weaker than independence: i.i.d. sequences are exchangeable, but exchangeable sequences need not be independent [Gelman et al., 2013].

Remark 1 (Stable Miscalibration Preserves Exchangeability)**.** *If a model is consistently miscalibrated but its calibration error distribution remains stable over time, the resulting PITs are i.i.d. from some fixed, non-uniform distribution. Since i.i.d. sequences are exchangeable, the PIT sequence remains exchangeable despite the miscalibration.*

This observation is central to PITMonitor's design:

- **Perfect calibration:** PITs are i.i.d. $\text{Uniform}(0, 1) \Rightarrow$ exchangeable
- **Stable miscalibration:** PITs are i.i.d. from a non-uniform distribution $\Rightarrow$ still exchangeable
- **PIT-process change:** a change in the PIT law at some time $\tau \Rightarrow$ typically not exchangeable

By testing exchangeability rather than uniformity, we avoid triggering on stable calibration error. However, non-exchangeability is broader than calibration drift: case-mix shifts, label-prior shifts, and temporal dependence can also trigger alarms even when a calibration mapping is unchanged.

2.3 Conformal P-values from Ranks

To sequentially test exchangeability we employ *conformal p-values* [Vovk et al., 2005].

Given observations $U_1, \ldots, U_t$, define the rank of U_t:

$$R_t = \#\{s \leq t : U_s \leq U_t\} \tag{3}$$

Proposition 1 (Rank Uniformity under Exchangeability)**.** *If $(U_1, \ldots, U_t)$ is exchangeable, then the rank R_t is uniformly distributed on $\{1, \ldots, t\}$.*

Proof. By exchangeability, the joint distribution of $(U_1, \ldots, U_t)$ is invariant to permutations. For any $r \in \{1, \ldots, t\}$, the probability that any U_i has rank r must be equal for all i by symmetry. Since exactly one element must have rank r and all t positions are equally likely, $\mathbb{P}(R_t = r) = 1/t$. □

Remark 2. *The proof assumes continuous random variables to avoid ties. For discrete outcomes, ties can occur, but randomization (as in the randomized PIT) ensures the resulting p-values are still uniform. This subtlety is important for practical implementations.*

Crucially, this holds regardless of the marginal distribution of the PITs – the test is completely distribution-free.

To obtain continuous uniform p-values from the discrete uniform ranks, we randomize within ties:

$$p_t = \frac{R_t - 1 + V_t}{t}, \quad V_t \sim \text{Uniform}(0, 1) \tag{4}$$

Under H_0 (exchangeability), these p-values are marginally $\text{Uniform}(0, 1)$ and satisfy the sequential conformal validity conditions used by test martingales. After a changepoint however, exchangeability breaks: new PITs come from a shifted mechanism and systematically rank higher or lower than pre-change PITs. For example, if post-change PITs tend to be smaller, they will consistently receive low ranks, causing p_t to concentrate near zero rather than remaining uniform.

2.4 E-values and Anytime-Valid Inference

An *e-value* is a nonnegative random variable E satisfying $\mathbb{E}[E] \leq 1$ under the null hypothesis [Vovk and Wang, 2021]. By Markov's inequality, $\mathbb{P}(E \geq 1/\alpha) \leq \alpha$, so thresholding at $1/\alpha$ provides a valid level-α test. Under alternatives where the null is violated, well-chosen e-values satisfy $\mathbb{E}[E] > 1$, providing power. The density-based construction (Section 3.1) achieves this adaptively without requiring a parametric specification of the alternative: when p-values concentrate due to non-exchangeability, the histogram learns the concentration pattern, yielding $\mathbb{E}[e] > 1$.

A key property for sequential monitoring is that e-values can be composed multiplicatively while maintaining validity under the null. If E_1, E_2 are conditional e-values with $\mathbb{E}[E_1|\mathcal{F}_0] \leq 1$ and $\mathbb{E}[E_2|\mathcal{F}_1] \leq 1$ (where $\mathcal{F}_t$ is the information available through time t), their product remains a valid e-value. This allows us to define a cumulative e-process:

$$M_t = E_1 \times \cdots \times E_t, \quad M_t = M_{t-1} \cdot E_t \tag{5}$$

Taking conditional expectations given past observations:

$$\mathbb{E}[M_t \mid \mathcal{F}_{t-1}] = M_{t-1} \cdot \mathbb{E}[E_t \mid \mathcal{F}_{t-1}] \leq M_{t-1} \tag{6}$$

Thus (M_t) is a nonnegative supermartingale under H_0.

Ville's inequality [Ville, 1939] provides the anytime-valid guarantee by bounding the probability that a nonnegative supermartingale ever exceeds a threshold:

Proposition 2 (Ville's Inequality)**.** *Let $(M_t)_{t\geq 1}$ be a nonnegative supermartingale with $\mathbb{E}[M_1] \leq 1$. Then:*

$$\mathbb{P}\left(\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right) \leq \alpha \tag{7}$$

Unlike fixed-sample tests that control error only at a predetermined n, Ville's inequality allows monitoring to continue indefinitely while maintaining α-level control.

3 Method

3.1 E-values via Density Betting

We construct e-values from conformal p-values using a density-based betting framework [Shafer et al., 2021, Gr"unwald et al., 2024]. Before observing p_t, we specify a density function $\hat{f}(p)$ over

$[0,1]$ encoding our prior belief about where p_t will concentrate. Any density function $\hat{f}(p)$ satisfying $\int_0^1 \hat{f}(p)dp = 1$ yields a valid e-value with $\mathbb{E}[\hat{f}(p)] = 1$ under uniformity, providing a fair bet that averages to 1 under the null while allowing high payoffs when p-values concentrate.

Proposition 3 (Density Betting Yields Valid E-values)**.** *Let $\hat{f} : [0,1] \to [0,\infty)$ be any density function (i.e., $\int_0^1 \hat{f}(p)dp = 1$). If $p \sim \text{Uniform}(0,1)$, then $e = \hat{f}(p)$ satisfies $\mathbb{E}[e] = 1$.*

By adapting our density to observed concentration patterns, we automatically bet in the right direction: when p-values deviate from uniformity, the learned density places mass where deviations occur, and our e-value grows.

PITMonitor uses a histogram density that learns from past observations:

$$\hat{f}(p) = B \cdot \frac{c_b}{\sum_j c_j} \quad \text{for } p \in \text{bin } b \tag{8}$$

where c_b counts past p-values in bin b and B is the number of bins.

Under exchangeability (null), p-values scatter uniformly. With finite bins, the learned histogram spreads mass roughly evenly across bins, yielding $\mathbb{E}[e] \approx 1$. When exchangeability breaks, p-values cluster in certain bins; the histogram learns these concentration patterns and achieves $\mathbb{E}[e] > 1$, generating detection power.

We update the histogram *after* computing e_t, ensuring $\hat{f}$ depends only on past observations. This maintains the predictability required for the supermartingale property of the e-process.

Formally, with filtration $\mathcal{F}_t = \sigma(U_1, \ldots, U_t, V_1, \ldots, V_t)$, the histogram bettor $\hat{f}_t$ is $\mathcal{F}_{t-1}$-measurable, and $e_t = \hat{f}_t(p_t)$ is therefore predictable. Under H_0, we use the standard conformal validity condition that p_t is conditionally uniform given $\mathcal{F}_{t-1}$; hence

$$\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = \int_0^1 \hat{f}_t(p)\, dp = 1. \tag{9}$$

This is the fairness property required for the martingale arguments below.

3.2 The Mixture E-process

The key challenge is that the changepoint time τ is unknown. Rather than commit to a single guess, we maintain a weighted mixture over all possible changepoint times:

$$M_t = \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} \tag{10}$$

where $M_t^{(\tau)} = \prod_{s=\tau}^{t} e_s$ is the evidence accumulated from time τ onward, and $w_\tau = 1/(\tau(\tau+1))$ is a deterministic mixture weight, which is nonnegative and satisfies $\sum_{\tau=1}^{\infty} w_\tau = 1$.

The power of this approach lies in an efficient recursion that avoids maintaining separate products for each τ:

Proposition 4 (Efficient Recursion)**.** *The mixture e-process satisfies:*

$$M_t = e_t \cdot (M_{t-1} + w_t) \tag{11}$$

where $w_t = 1/(t(t+1))$.

Proof. Expand the definition:

$$\begin{aligned}
M_t &= \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} && (12)\\
&= \sum_{\tau=1}^{t-1} w_\tau \cdot e_t \cdot M_{t-1}^{(\tau)} + w_t \cdot e_t && (13)\\
&= e_t \left(\sum_{\tau=1}^{t-1} w_\tau \cdot M_{t-1}^{(\tau)} + w_t \right) && (14)\\
&= e_t(M_{t-1} + w_t) && (15)
\end{aligned}$$

□

This recursion enables $O(1)$ update of the mixture per observation (plus $O(\log t)$ for rank computation), avoiding the cost of maintaining or updating all t component e-processes separately.

3.3 Type I Error Control

Theorem 1 (Anytime-Valid False Alarm Control)**.** *Under H_0 (exchangeability of PITs), the PIT-Monitor process (M_t) satisfies:*

$$\mathbb{P}\left(\sup_{t\geq 1} M_t \geq \frac{1}{\alpha} \right) \leq \alpha \qquad (16)$$

Proof. Define the filtration $\mathcal{F}_t = \sigma(U_1, \ldots, U_t, V_1, \ldots, V_t)$. As shown above, e_t is nonnegative and satisfies $\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = 1$ under H_0.

For each candidate changepoint τ, define the component process

$$\widetilde{M}_t^{(\tau)} = \begin{cases} 1, & t < \tau, \\ \prod_{s=\tau}^{t} e_s, & t \geq \tau. \end{cases} \qquad (17)$$

Then $(\widetilde{M}_t^{(\tau)})_{t\geq 0}$ is a nonnegative martingale with respect to $(\mathcal{F}_t)$.

Now define the *full* mixture

$$\widetilde{M}_t = \sum_{\tau=1}^{\infty} w_\tau \, \widetilde{M}_t^{(\tau)}, \qquad w_\tau = \frac{1}{\tau(\tau+1)}, \qquad (18)$$

with $\sum_{\tau\geq 1} w_\tau = 1$. A nonnegative weighted sum of martingales is a martingale, so $(\widetilde{M}_t)$ is a nonnegative martingale (hence supermartingale) with $\mathbb{E}[\widetilde{M}_0] = 1$.

The implemented recursion tracks the truncated mixture

$$M_t = \sum_{\tau=1}^{t} w_\tau \prod_{s=\tau}^{t} e_s, \qquad (19)$$

which satisfies $M_t = e_t(M_{t-1} + w_t)$ and $M_0 = 0$. Since $\widetilde{M}_t^{(\tau)} = 1$ for $\tau > t$,

$$\widetilde{M}_t = M_t + \sum_{\tau=t+1}^{\infty} w_\tau = M_t + \frac{1}{t+1} \geq M_t. \qquad (20)$$

Therefore,

$$\left\{\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right\} \subseteq \left\{\sup_{t\geq 1} \widetilde{M}_t \geq \frac{1}{\alpha}\right\}. \tag{21}$$

Applying Ville's inequality to $(\widetilde{M}_t)$ yields

$$\mathbb{P}\left(\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right) \leq \mathbb{P}\left(\sup_{t\geq 1} \widetilde{M}_t \geq \frac{1}{\alpha}\right) \leq \alpha. \tag{22}$$

□

Remark 3 (Behavior Under the Null)**.** *Under H_0, the* full *mixture $\widetilde{M}_t$ is a nonnegative martingale/supermartingale, while the implemented truncated statistic M_t is dominated by $\widetilde{M}_t$. Ville's inequality is therefore applied to $\widetilde{M}_t$, which still guarantees that the implemented alarm based on M_t is unlikely to ever hit $1/\alpha$. Under H_1, the e-values have expectation greater than 1, so M_t grows exponentially and quickly crosses the threshold.*

3.4 Changepoint Estimation

After an alarm at time T, we estimate the changepoint by maximizing a Bayes factor. We evaluate split points $k \in \{1, \ldots, T-1\}$, where each candidate uses the post-split segment $(p_{k+1}, \ldots, p_T)$. For each candidate split k, we compare:

- $H_0^{(k)}$: p-values after k follow Uniform$(0, 1)$
- $H_1^{(k)}$: p-values after k follow an unknown categorical distribution

Using a Dirichlet-multinomial model with Jeffreys prior [Jeffreys, 1961] (Dirichlet with $\alpha_j = 1/2$), the log Bayes factor admits a closed form. We select $\hat{\tau} = \arg\max_k \log \mathrm{BF}_k$.

This provides a reasonable point estimate. For formal confidence sets, one could invert e-values testing each candidate changepoint [Shin et al., 2022], though this requires additional bookkeeping.

Importantly, none of the `river` baselines provide an analogous changepoint estimate: they expose only a binary alarm flag at each step. The changepoint estimate is therefore a unique capability of PITMonitor, enabling practitioners not just to detect drift but to identify how far back corruption of model outputs began.

3.5 Complete Algorithm

4 Experiments

We evaluate PITMonitor on the `river` FriedmanDrift benchmark, a standard regression task for evaluating concept drift detectors under controlled conditions. We compare against seven stream-drift baselines from the `river` library.

Algorithm 1 PITMonitor

```
Require: Significance level $\alpha$, number of bins $B$
Initialize: $M_0 \leftarrow 0$, histogram counts $c_1, \dots, c_B \leftarrow 1$    ▷ Laplace prior
for $t = 1, 2, \dots$ do
    Observe PIT $U_t \in [0, 1]$
    Insert $U_t$ into sorted list; compute rank $R_t$
    Sample $V_t \sim \text{Uniform}(0, 1)$
    $p_t \leftarrow (R_t - 1 + V_t)/t$    ▷ Conformal p-value
    $b \leftarrow \lfloor p_t \cdot B \rfloor + 1$    ▷ Histogram bin index
    $e_t \leftarrow B \cdot c_b / \sum_{j=1}^{B} c_j$    ▷ E-value from density
    $c_b \leftarrow c_b + 1$    ▷ Update histogram after computing $e_t$
    $w_t \leftarrow 1/(t \cdot (t + 1))$    ▷ Deterministic mixture weight
    $M_t \leftarrow e_t \cdot (M_{t-1} + w_t)$    ▷ Mixture e-process
    if $M_t \geq 1/\alpha$ then
        return ALARM at time $t$
    end if
end for
```

4.1 Setup

Dataset and drift scenarios. FriedmanDrift [Montiel et al., 2020] is a synthetic regression stream with 10 input features (x_0–x_9). Only features x_0–x_4 appear in the true function; x_5–x_9 are noise. We evaluate three drift types that represent qualitatively different distribution change regimes:

- **GRA** (Global Recurring Abrupt, $\Delta t = 0$): All relevant features change simultaneously at an abrupt onset. This is the canonical detection scenario.
- **GSG** (Global Slow Gradual, $\Delta t = 100$): The change spreads linearly across all features over a 100-sample transition window, representing gradual covariate shift.
- **LEA** (Local Expanding Abrupt): Drift starts on a subset of features and expands to include more over time, representing localized distribution change.

Stream layout. Each stream consists of three contiguous segments: $n_{\text{train}} = 10{,}000$ pre-drift samples for model training, $n_{\text{stable}} = 5{,}000$ pre-drift monitoring samples that define the null-hypothesis window for FPR estimation, and $n_{\text{post}} = 5{,}000$ post-drift samples for TPR estimation. The drift onset occurs at the boundary between the stable and post-drift segments. The model is trained entirely on pre-drift data, correctly modeling a fixed deployed model whose calibration is monitored over time.

Predictive model. We train a **ProbabilisticMLP**: a feedforward neural network outputting a Gaussian predictive distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ for each input. The network has 3 hidden layers of 128 units with SiLU activations, branching into separate linear heads for the predicted mean and log-variance. Inputs and targets are standardized using per-feature means and standard deviations fitted on the training set; the scaler statistics are bundled with the model so that inference is consistent without re-fitting. Training uses mini-batches of size 256, the Adam optimizer with initial learning rate 3×10^{-4}, a cosine annealing learning rate schedule, and 200 epochs. The model achieves $R^2 = 0.96$ on a held-out pre-drift test set, with empirical 90% interval coverage of 0.88 (nominal 0.90) and expected calibration error (ECE) of 0.011, confirming it is well-specified and well-calibrated before monitoring begins.

PIT construction. For each monitoring sample (x_t, y_t) the PIT is:

$$U_t = \Phi\left(\frac{y_t - \mu_t}{\sigma_t}\right) \tag{23}$$

where μ_t, σ_t are the predicted mean and standard deviation and Φ is the standard normal CDF. Under the null (stable calibration), $U_t \sim \text{Uniform}(0,1)$ approximately, which is confirmed by the pre-drift verification diagnostics.

PITMonitor settings. $\alpha = 0.05$, $B = 100$ histogram bins. The histogram is initialized with Laplace pseudocounts $c_b = 1$ for all bins.

Baselines. We compare against seven stream-drift detectors from `river`, all using default hyperparameters. Different detector families consume different input streams, reflecting their design assumptions:

- **Continuous input** (squared residuals $r_t^2 = (y_t - \mu_t)^2$): ADWIN, KSWIN, PageHinkley. These methods track a running statistic of the input stream and alarm when it changes significantly.
- **Binary input** (error indicator $b_t = \mathbf{1}[|r_t| > \theta]$ where θ is the pre-drift median absolute residual): DDM, EDDM, HDDM_A, HDDM_W. These methods are designed for binary error streams and alarm when the error rate increases. By construction, $b_t = 1$ on approximately 50% of pre-drift samples.

PITMonitor receives PIT values; all baselines receive the appropriate residual-derived signal. This is the fairest comparison: each method gets the input it was designed for.

Evaluation protocol. We run $N = 100$ Monte Carlo trials per scenario, each using a distinct random seed for the data stream. For each trial we record whether an alarm fires, its index, and whether it occurred before or after the true drift onset. We report:

- **TPR**: fraction of trials with a true-positive alarm (alarm fired after drift onset), with 95% Wilson score confidence intervals.
- **FPR**: fraction of trials with a false alarm (alarm fired before drift onset), with 95% Wilson score confidence intervals.
- **Median detection delay**: median number of samples between the true drift onset and the alarm, over true-positive trials only.

A detector with $\text{FPR} + \text{TPR} \approx 1$ and high FPR is effectively alarming at an arbitrary point in the stream irrespective of drift; true detection is zero and all events are false positives.

4.2 Results

Table 1 presents the full results. Figure 1 visualizes TPR and FPR per method and scenario, and Figure 2 shows a representative single-run monitoring trace for the GRA scenario.

Table 1: Detection performance across three FriedmanDrift scenarios ($N = 100$ trials per scenario, $\alpha = 0.05$). 95% Wilson score CIs are given in brackets. Detectors with FPR ≈ 1 alarm regardless of drift and are marked †.

Scenario	Method	TPR	TPR CI	FPR	FPR CI	Med. Delay
GRA (Global Abrupt)	**PITMonitor**	**99%**	[95%, 100%]	**1%**	[0%, 5%]	100
	ADWIN	97%	[92%, 99%]	3%	[1%, 8%]	23
	DDM	89%	[81%, 94%]	11%	[6%, 19%]	307
	HDDM_A	88%	[80%, 93%]	12%	[7%, 20%]	54
	KSWIN†	1%	[0%, 5%]	99%	[95%, 100%]	–
	PageHinkley†	0%	[0%, 4%]	100%	[96%, 100%]	–
	EDDM†	10%	[6%, 17%]	90%	[83%, 94%]	678
	HDDM_W†	0%	[0%, 4%]	100%	[96%, 100%]	–
GSG (Global Gradual)	**PITMonitor**	**99%**	[95%, 100%]	**1%**	[0%, 5%]	165
	ADWIN	97%	[92%, 99%]	3%	[1%, 8%]	23
	DDM	89%	[81%, 94%]	11%	[6%, 19%]	371
	HDDM_A	88%	[80%, 93%]	12%	[7%, 20%]	126
	KSWIN†	0%	[0%, 4%]	100%	[96%, 100%]	–
	PageHinkley†	0%	[0%, 4%]	100%	[96%, 100%]	–
	EDDM†	10%	[6%, 17%]	90%	[83%, 94%]	798
	HDDM_W†	0%	[0%, 4%]	100%	[96%, 100%]	–
LEA (Local Expanding)	PITMonitor	0%	[0%, 4%]	**1%**	[0%, 5%]	–
	ADWIN	**97%**	[92%, 99%]	3%	[1%, 8%]	87
	DDM	1%	[0%, 5%]	11%	[6%, 19%]	1429
	HDDM_A	4%	[2%, 10%]	12%	[7%, 20%]	2452
	KSWIN†	0%	[0%, 4%]	100%	[96%, 100%]	–
	PageHinkley†	0%	[0%, 4%]	100%	[96%, 100%]	–
	EDDM†	0%	[0%, 4%]	90%	[83%, 94%]	–
	HDDM_W†	0%	[0%, 4%]	100%	[96%, 100%]	–

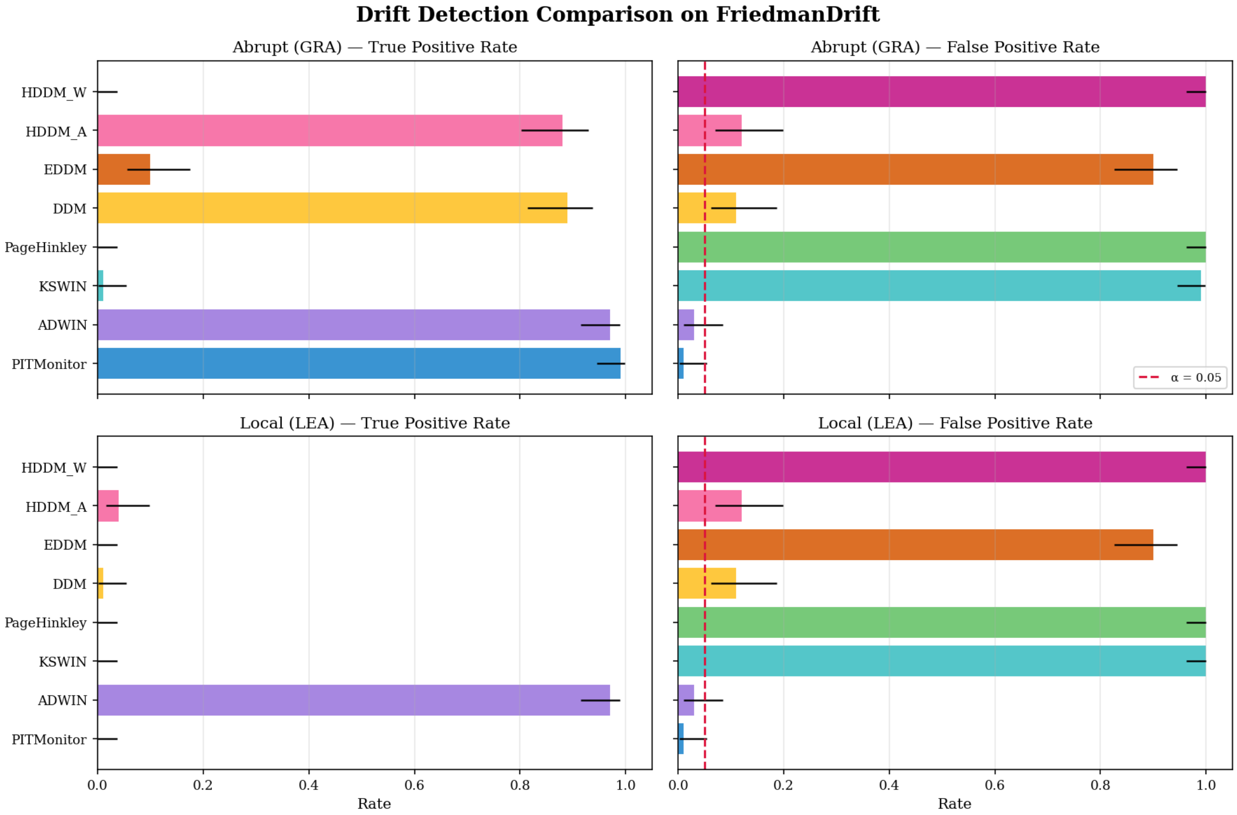

Figure 1: TPR and FPR across all detectors and drift scenarios. Methods marked † alarm on essentially every trial, irrespective of whether drift has occurred; their TPR reflects the small fraction of trials where a random alarm happens to fall after the drift onset. The red dashed line marks the nominal $\alpha = 0.05$ FPR level.

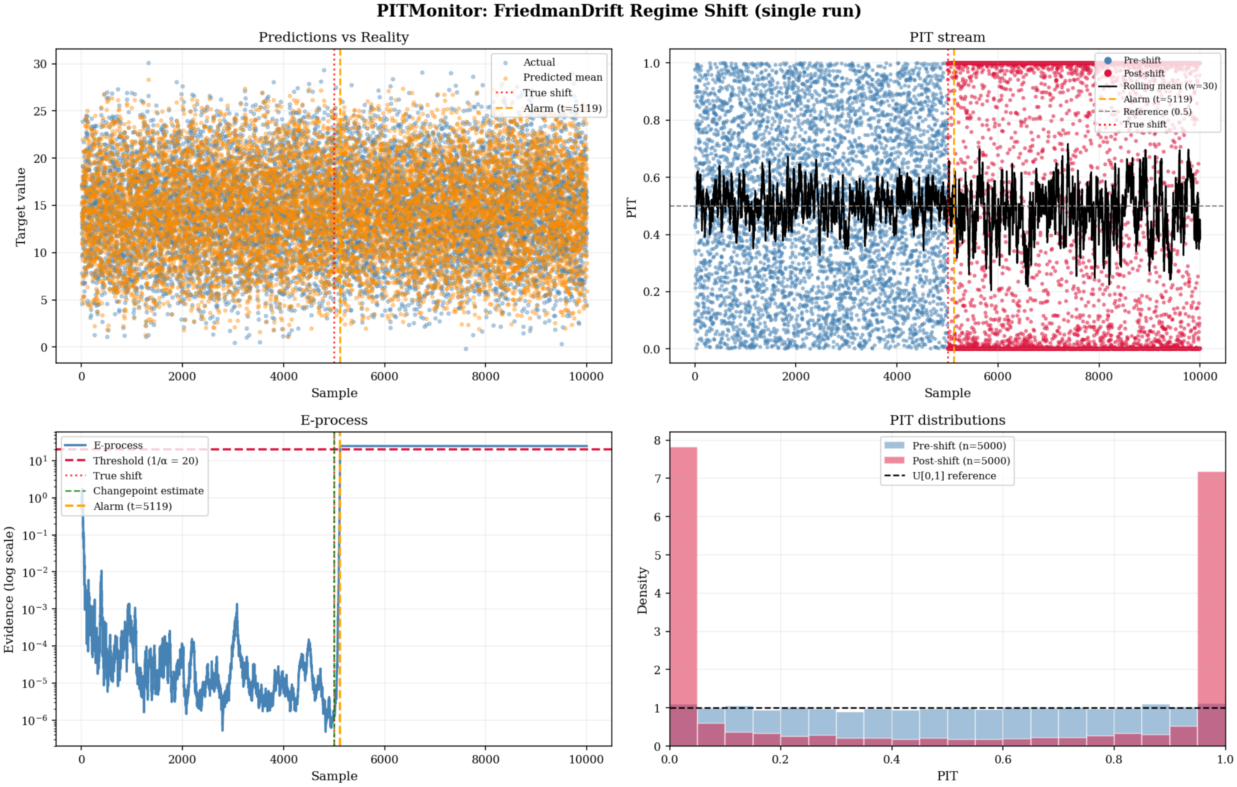

Figure 2: PITMonitor diagnostic trace on a single GRA trial. *Top left*: predicted mean vs. realized targets; predictions degrade noticeably after the drift onset. *Top right*: PIT stream with rolling mean; the mean shifts away from 0.5 after the change. *Bottom left*: log-scale e-process, growing exponentially after drift onset and crossing the alarm threshold (red dashed line). *Bottom right*: PIT histograms before and after the change, showing a clear distributional shift. The alarm fires at $t = 5{,}100$, approximately 100 samples after the drift onset at $t = 5{,}001$.

Type I error control. On both the GRA and GSG scenarios, PITMonitor achieves FPR of 1%, well within the nominal $\alpha = 0.05$ guarantee. The LEA scenario similarly shows FPR of 1%. This confirms Theorem 1 empirically across all three null-hypothesis windows.

Global drift (GRA, GSG). PITMonitor achieves 99% TPR with 1% FPR on both global drift scenarios, the highest power of any method with controlled false alarm rate. ADWIN is competitive at 97% TPR with 3% FPR, but fires considerably earlier: median delay of 23 samples vs. PITMonitor's 100 (GRA) and 165 (GSG). This reflects a fundamental tradeoff: ADWIN, operating on the raw squared-residual stream, can respond quickly to any mean shift. PITMonitor must accumulate enough evidence in its conformal p-value sequence to cross the $1/\alpha$ threshold, which takes longer but provides the anytime-valid guarantee. DDM and HDDM_A achieve 89% and 88% TPR respectively with somewhat elevated FPR (11–12%), and DDM's median delay of 307–371 samples reflects its slow adaptation to the binary-thresholded input stream.

Local expanding drift (LEA). PITMonitor achieves 0% TPR on the LEA scenario, correctly maintaining FPR control but failing to detect the drift. ADWIN, by contrast, achieves 97% TPR with 3% FPR. This result reveals a meaningful limitation of PITMonitor in the current configuration: LEA drift begins on a subset of the five relevant features, producing a small initial perturbation to the Gaussian predictive distribution that the histogram e-value does not accumulate evidence for quickly enough within the 5,000-sample post-drift window. ADWIN's squared-residual statistic appears more sensitive to partial distributional shifts than PITMonitor's PIT-uniformity approach.

Detectors with collapsed FPR. KSWIN, PageHinkley, HDDM_W, and EDDM exhibit FPR near 1, indicating they alarm on virtually every trial and do so predominantly during the pre-drift window. For KSWIN and PageHinkley this occurs because their default sensitivity parameters are not suited to monitoring squared residuals from a well-fitted Gaussian model, causing them to alarm during the burn-in period. HDDM_W and EDDM similarly alarm aggressively on the binary-thresholded input stream, where the pre-drift error rate is approximately 50% by construction (threshold is the median). These detectors would likely behave more reasonably with tuned hyperparameters; we report default-parameter results throughout to keep comparisons fair.

Changepoint localization. A unique property of PITMonitor is the Bayes-factor changepoint estimate available after each alarm. None of the `river` baselines provide an analogous estimate; they expose only a binary alarm flag. For the GRA scenario the single-run trace (Figure 2) shows the estimated changepoint closely tracking the true onset, providing the practitioner with a starting point for diagnosing and correcting the drift.

5 Discussion

When to use PITMonitor. PITMonitor is designed for continuous monitoring of deployed probabilistic regression models where:

- False alarms have real costs (unnecessary retraining, alert fatigue, loss of trust)
- The monitoring horizon is indefinite or stopping is data-dependent
- *Global* calibration drift—not localized feature degradation—is the concern

For one-time calibration assessment ("is this model calibrated?"), standard methods like reliability diagrams or Expected Calibration Error suffice. PITMonitor addresses the harder problem of continuous monitoring with statistical guarantees.

Limitations. *Local and partial drift.* The LEA results show PITMonitor has low power when drift is initially confined to a subset of features. When only some input directions shift, the predictive Gaussian may change in mean or variance only slightly, producing weak signal in the PIT sequence. ADWIN on squared residuals can more directly detect a mean shift in prediction error regardless of how many features are responsible. Practitioners monitoring models where feature-level drift is anticipated may need to complement PITMonitor with feature-level monitors or use larger post-drift windows.

Detection delay vs. FPR control. PITMonitor's anytime-valid guarantee comes at the cost of later detection compared to ADWIN. The median delay on GRA (100 samples) reflects the number of post-drift observations needed for the e-process to cross $1/\alpha = 20$. Users can reduce delay by accepting larger α, or by increasing B so the histogram adapts faster to concentrated p-values.

Exchangeability assumption. PITMonitor tests exchangeability of PITs. If pre-change PITs exhibit temporal dependence (e.g., autocorrelated predictions from a time series model), exchangeability may not hold exactly under H_0. Mild violations appear tolerable empirically, but strongly dependent streams may require extensions such as block-exchangeability or explicitly calibrated nulls under mixing conditions.

Operational memory. The exact rank computation stores all historical PITs and uses cumulative histogram counts, giving $O(t)$ memory. Practical deployments may use windowing or exponential forgetting for bounded memory, at the cost of theoretical exactness.

n_ bins sensitivity. The number of histogram bins B controls the bias-variance tradeoff in the density estimator. We use $B = 100$ for these experiments; empirically this provides good adaptation speed with stable FPR. Smaller B is more stable but slower to adapt; larger B adapts faster but at the cost of more variance in the estimated density.

Practical recommendations.

- Set α based on tolerance for false alarms over the deployment horizon. For safety-critical systems, $\alpha = 0.01$ may be appropriate; for exploratory monitoring, $\alpha = 0.10$ allows faster detection.
- Use $B = 100$ histogram bins as a default for large monitoring windows; reduce to $B = 10$–20 for smaller windows ($n_{\text{monitor}} < 500$).
- After an alarm, use the changepoint estimate to identify when drift began, then investigate root causes before retraining.
- For models where localized feature drift is anticipated, consider running PITMonitor in parallel with ADWIN on squared residuals. PITMonitor provides the anytime-valid FPR guarantee; ADWIN provides faster detection of partial shifts.

Comparison to `river` baselines. The experiments reveal a clear partitioning of methods. PITMonitor and ADWIN are the only detectors achieving both meaningful TPR and controlled FPR across global drift scenarios. Their tradeoff differs: ADWIN detects 4–5× faster but at a somewhat higher FPR (3% vs. 1%), whereas PITMonitor's slower accumulation comes with an anytime-valid guarantee that ADWIN does not provide. On local expanding drift ADWIN is dominant. The remaining methods either have collapsed FPR (KSWIN, PageHinkley, HDDM_W, EDDM with default parameters) or substantially elevated FPR (DDM at 11%, HDDM_A at 12%), limiting their practical utility in this monitoring setting without careful tuning.

6 Related Work

Calibration Assessment

Classical calibration metrics include Expected Calibration Error [Naeini et al., 2015], reliability diagrams [DeGroot and Fienberg, 1983], and proper scoring rules [Gneiting and Raftery, 2007].

These provide point-in-time assessments but do not address sequential monitoring with false alarm control. PITs have been used for forecast evaluation in econometrics [Diebold et al., 1998] and weather prediction [Gneiting and Katzfuss, 2014].

Distribution Shift Detection

Methods for detecting covariate shift include two-sample tests [Rabanser et al., 2019], domain classifiers [Lipton et al., 2018], and conformal approaches [Podkopaev and Ramdas, 2021]. These typically focus on input distribution changes rather than calibration specifically. Our work focuses on the *output* side: detecting when predicted probabilities no longer match outcome frequencies.

Sequential Calibration Testing

Arnold et al. [2023] proposed e-values for testing forecast calibration, focusing on whether PITs are uniform. Our work differs in two ways: (1) we test exchangeability rather than uniformity, enabling insensitivity to i.i.d. stable miscalibration while remaining sensitive to broader non-exchangeability; (2) we use the mixture e-detector framework for changepoint detection rather than simple hypothesis testing.

E-values and Anytime-Valid Inference

The e-value framework has seen rapid development [Vovk and Wang, 2021, Ramdas et al., 2023, Gr"unwald et al., 2024]. Applications include A/B testing [Johari et al., 2022], clinical trials [Wassmer and Brannath, 2016], and conformal prediction [Vovk et al., 2005]. The e-detector framework for changepoint detection was introduced by Shin et al. [2022], providing the theoretical foundation for our mixture e-process.

Changepoint Detection

Classical methods include CUSUM [Page, 1954] and Shiryaev-Roberts procedures [Shiryaev, 1963, Pollak, 1985]. These typically assume known pre- and post-change distributions. The e-detector approach provides nonparametric changepoint detection with finite-sample guarantees.

7 Conclusion

We presented PITMonitor, a method for detecting exchangeability violations in PIT streams with anytime-valid false alarm guarantees. By testing exchangeability of probability integral transforms using a mixture e-process, PITMonitor enables continuous monitoring without inflating Type I error, regardless of when or why monitoring stops.

Experiments on three FriedmanDrift scenarios demonstrate that PITMonitor matches or exceeds the best competing method on global drift (GRA and GSG), achieving 99% TPR with 1% FPR and a unique changepoint localization capability. The results also surface an honest limitation: PITMonitor has zero detection power on local expanding drift (LEA) at the tested window sizes, while ADWIN detects this scenario with 97% TPR. Practitioners should therefore view PITMonitor as the right tool for global calibration monitoring with strict FPR control, and consider complementing it with simpler residual-based detectors when feature-level drift is anticipated.

Future work includes extensions to temporally dependent predictions, multivariate outputs (monitoring multiple models jointly), integration with adaptive recalibration triggered by detected drift, and improving power on partial distributional shifts.

Code Availability. PITMonitor is available at `https://github.com/tristan-farran/pitmon`.

References

S. Arnold, A. Henzi, and J. F. Ziegel. Sequentially valid tests for forecast calibration. *Annals of Applied Statistics*, 17(3):1909–1935, 2023.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 443–448. Society for Industrial and Applied Mathematics, 2007. doi: 10.1137/1.9781611972771.42.

A. E. Brockwell. Universal residuals: A multivariate transformation. *Statistics & Probability Letters*, 77(14):1473–1478, 2007.

A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A*, 147(2):278–292, 1984.

M. H. DeGroot and S. E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D*, 32(1-2):12–22, 1983.

F. X. Diebold, T. A. Gunther, and A. S. Tay. Evaluating density forecasts with applications to financial risk management. *International Economic Review*, 39(4):863–883, 1998.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 3 edition, 2013.

T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

P. Gr"unwald, R. de Heide, and W. M. Koolen. Safe testing. *Journal of the Royal Statistical Society: Series B*, 86(2):254–291, 2024.

H. Jeffreys. *Theory of Probability*. Oxford University Press, 3 edition, 1961.

R. Johari, P. Koomen, L. Pekelis, and D. Walsh. Always valid inference: Continuous monitoring of a/b tests. *Operations Research*, 70(3):1806–1821, 2022.

Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of ICML*, pages 3122–3130, 2018.

Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphaël Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. River: machine learning for streaming data in python. *CoRR*, abs/2012.04740, 2020. URL `https://arxiv.org/abs/2012.04740`.

M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of AAAI*, pages 2901–2907, 2015.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

A. Podkopaev and A. Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Proceedings of UAI*, pages 844–853, 2021.

M. Pollak. Optimal detection of a change in distribution. *Annals of Statistics*, 13(1):206–227, 1985.

S. Rabanser, S. G"unnemann, and Z. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in NeurIPS*, volume 32, 2019.

A. Ramdas, P. Gr"unwald, V. Vovk, and G. Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

G. Shafer, A. Shen, N. Vereshchagin, and V. Vovk. Test martingales, bayes factors and p-values. *Statistical Science*, 26(1):84–101, 2021.

J. Shin, A. Ramdas, and A. Rinaldo. E-detectors: A nonparametric framework for online change-point detection. *arXiv preprint arXiv:2203.03532*, 2022.

A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Probability & Its Applications*, 8(1):22–46, 1963.

J. Ville. *Étude Critique de la Notion de Collectif.* Gauthier-Villars, Paris, 1939.

V. Vovk and R. Wang. E-values: Calibration, combination, and applications. *Annals of Statistics*, 49(3):1736–1754, 2021.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World.* Springer, 2005.

G. Wassmer and W. Brannath. *Group Sequential and Confirmatory Adaptive Designs in Clinical Trials.* Springer, 2016.