When Your Model Stops Working: Anytime-Valid Calibration Monitoring

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

February 25, 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 models. Unlike many traditional calibration tests, PITMonitor provides *anytime-valid* false alarm control: the probability of ever raising a spurious alarm is bounded by α for all time, without requiring a fixed horizon or stopping rule. We prove Type I error control via Ville's inequality and demonstrate detection power on three scenarios from `river`'s FriedmanDrift dataset, comparing performance against the seven included stream-drift detectors. Code is available at `https://github.com/tristan-farran/pitmon`.

1 Introduction

Probabilistic models deployed in production face a fundamental challenge: the world changes. Across many essential domains from medicine to finance, models may encounter non-stationary processes, regime shifts, and concept drift. When these shifts occur, model calibration can degrade drastically, leading to consequential issues downstream. 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., 2021]. 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 or changepoint localisation.

ADWIN employs adaptive windowing with statistical change detection via Hoeffding bounds [Bifet and Gavaldà, 2007]. Its parameter δ bounds the false alarm probability *per comparison at each time step* (with a Bonferroni-style correction $\delta' = \delta/n$ for window length n), not the probability of ever raising a false alarm over an unbounded stream. Over a long monitoring window the probability of at least one spurious alarm can substantially exceed δ, as our experiments confirm. Furthermore, ADWIN operates on generic performance statistics such as squared residuals and does not provide changepoint estimates, only partially addressing the problem of reliable, informative, long-term calibration monitoring.

We propose PITMonitor, an anytime-valid calibration monitoring method with four key properties:

1. **Anytime-valid false alarm control:** we prove that $\mathbb{P}(\text{ever alarm} \mid H_0) \leq \alpha$ for all time, without requiring a pre-specified monitoring horizon or stopping rule.
2. **Change detection and localisation 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, while a changing process can.[1]
3. **No baseline period required:** unlike methods requiring a "clean" reference distribution, PITMonitor works from the first observation by testing the PIT sequence's exchangeability.
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 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 $y \in \{1, \ldots, K\}$:

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

placing U uniformly within the cumulative probability interval corresponding to the true class.

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

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 [de Finetti, 1937].

Remark 1 (Stable Miscalibration Preserves Exchangeability). *If a model is consistently miscalibrated, with its calibration error distribution remaining 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. Uniform$(0, 1)$ ⇒ exchangeable
- **Stable miscalibration:** PITs are i.i.d. from a non-uniform distribution ⇒ still exchangeable
- **PIT-process change:** the PIT distribution changes at some time τ ⇒ not exchangeable

By testing exchangeability rather than uniformity, we avoid triggering on stable calibration error.

It should be noted, however, that non-exchangeability can also result from temporal dependence: autocorrelated PITs can trigger alarms even when the calibration distribution is unchanged. This can occur in time series models, models with lagged features, or whenever predictions are not independent across time. Practitioners should check for autocorrelation in the PIT sequence and be cautious when interpreting alarms in settings where temporal dependence is expected.

2.3 Conformal P-values

To sequentially test exchangeability we employ *conformal p-values from ranks* [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, $(U_1, \ldots, U_t)$ is equally likely to be in any of the $t!$ orderings. For any fixed rank $r \in \{1, \ldots, t\}$, exactly $(t-1)!$ of these orderings place U_t in position r. Therefore $\mathbb{P}(R_t = r) = (t-1)!/t! = 1/t$, giving uniform distribution on $\{1, \ldots, t\}$. □

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 Uniform$(0, 1)$. After a changepoint, 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

An *e-value* is a non-negative 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 $1/\alpha$ yields a valid level-α test without needing to know the distribution of E under the null. This is in contrast to p-values, which require knowing the null distribution explicitly to calibrate thresholds.

Under alternatives where the null is violated, an e-value has power if $\mathbb{E}[E] > 1$. The density-based construction in Section 3.1 achieves this adaptively: when conformal p-values concentrate in certain bins due to non-exchangeability, the histogram places more mass there, yielding $\mathbb{E}[e] > 1$ without requiring a parametric specification of the alternative.

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 $\mathbb{E}[E_1 \mid \mathcal{F}_0] \leq 1$ and $\mathbb{E}[E_2 \mid \mathcal{F}_1] \leq 1$ (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 non-negative supermartingale under H_0.

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., 2011, Grünwald 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: under uniformity, $\mathbb{E}[\hat{f}(p)] = \int_0^1 \hat{f}(p)\,dp = 1$, providing a fair bet that averages to 1 under the null while allowing high payoffs when p-values concentrate.

Proposition 2 (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:

$$\hat{f}(p) = B \cdot \frac{c_b}{\sum_j c_j} \quad \text{for } p \in \text{bin } b \tag{7}$$

where c_b counts past p-values in bin b and B is the number of bins. The histogram is initialized with Laplace pseudocounts $c_b = 1$ for all b, which ensures $\hat{f}$ is a valid density from the first observation and prevents zero-count bins from generating infinite or zero e-values during early monitoring.

Under exchangeability, p-values scatter uniformly and the learned histogram spreads mass roughly evenly across bins, yielding $\mathbb{E}[e] \approx 1$. If 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, guaranteeing that it is predictable, as required for the supermartingale property of the e-process.

3.2 The Mixture E-process

The key challenge is that the changepoint time τ is unknown. An e-process starting at τ would be sensitive to drift beginning at τ but would miss earlier changes, while one starting too early accumulates noise that dilutes its power. Rather than commit to a single guess, we maintain a weighted mixture over all possible changepoint times:

$$M_t = \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} \tag{8}$$

where $M_t^{(\tau)} = \prod_{s=\tau}^{t} e_s$ denotes the evidence accumulated from time τ onward (defined for $\tau \leq t$), and $w_\tau = 1/(\tau(\tau+1))$ is a deterministic weight satisfying $\sum_{\tau=1}^{\infty} w_\tau = 1$. Since each component $M_t^{(\tau)}$ forms a valid e-process sensitive to drift beginning at τ, the mixture is simultaneously sensitive to changepoints at any time, while remaining a valid e-process under the null by linearity of expectation.

This allows us to use an efficient recursion that avoids maintaining separate products for each τ:

Proposition 3 (Efficient Recursion)**.** *The mixture e-process satisfies:*

$$M_t = e_t \cdot (M_{t-1} + w_t) \tag{9}$$

Proof. Expand the definition:

$$\begin{aligned}
M_t &= \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} && (10)\\
&= \sum_{\tau=1}^{t-1} w_\tau \cdot e_t \cdot M_{t-1}^{(\tau)} + w_t \cdot e_t && (11)\\
&= e_t \left(\sum_{\tau=1}^{t-1} w_\tau \cdot M_{t-1}^{(\tau)} + w_t \right) && (12)\\
&= e_t(M_{t-1} + w_t) && (13)
\end{aligned}$$

□

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

3.3 Type I Error Control

Ville's inequality [Ville, 1939] provides the anytime-valid guarantee by bounding the probability that a non-negative supermartingale *ever* exceeds a threshold, regardless of the monitoring horizon or stopping rule:

Theorem 1 (Anytime-Valid False Alarm Control)**.** *Under H_0, PITMonitor satisfies:*

$$\mathbb{P}\left(\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right) \leq \alpha \tag{14}$$

Proof. As shown in subsection 2.3, each e-value satisfies $\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = 1$ under H_0.

The mixture $M_t = \sum_{\tau=1}^{t} w_\tau M_t^{(\tau)}$ is defined with $M_t^{(\tau)} = \prod_{s=\tau}^{t} e_s$ only for $\tau \leq t$. To apply Ville's inequality we work with an extended process defined for all $t \geq 0$. For each $\tau \geq 1$ define:

$$\widetilde{M}_t^{(\tau)} = \begin{cases} 1 & t < \tau \\ \prod_{s=\tau}^{t} e_s & t \geq \tau \end{cases} \tag{15}$$

Since $e_t \geq 0$ and $\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = 1$, each $(\widetilde{M}_t^{(\tau)})_{t\geq 0}$ is a non-negative martingale with $\widetilde{M}_0^{(\tau)} = 1$.

Define the full mixture over all $\tau \geq 1$:

$$\widetilde{M}_t = \sum_{\tau=1}^{\infty} w_\tau \, \widetilde{M}_t^{(\tau)} \tag{16}$$

A countable non-negative weighted combination of martingales with summable weights is itself a martingale, so $(\widetilde{M}_t)_{t\geq 0}$ is a non-negative martingale with $\widetilde{M}_0 = \sum_{\tau=1}^{\infty} w_\tau = 1$.

For $\tau > t$, $\widetilde{M}_t^{(\tau)} = 1$ since no e-values have been incorporated yet. Therefore:

$$\widetilde{M}_t = \sum_{\tau=1}^{t} w_\tau \prod_{s=\tau}^{t} e_s + \sum_{\tau=t+1}^{\infty} w_\tau \cdot 1 \tag{17}$$

$$= M_t + \sum_{\tau=t+1}^{\infty} \frac{1}{\tau(\tau+1)} \tag{18}$$

The tail sum telescopes: $\sum_{\tau=t+1}^{\infty} \frac{1}{\tau(\tau+1)} = \sum_{\tau=t+1}^{\infty} \left(\frac{1}{\tau} - \frac{1}{\tau+1}\right) = \frac{1}{t+1}$. Hence:

$$\widetilde{M}_t = M_t + \frac{1}{t+1} \geq M_t \tag{19}$$

$$\therefore \left\{\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right\} \subseteq \left\{\sup_{t\geq 1} \widetilde{M}_t \geq \frac{1}{\alpha}\right\} \tag{20}$$

Since $(\widetilde{M}_t)$ is a non-negative martingale, Ville's inequality gives $\mathbb{P}(\sup_{t\geq 0} \widetilde{M}_t \geq 1/\alpha) \leq \alpha$, thus:

$$\mathbb{P}\left(\sup_{t\geq 1} M_t \geq \frac{1}{\alpha}\right) \leq \mathbb{P}\left(\sup_{t\geq 1} \widetilde{M}_t \geq \frac{1}{\alpha}\right) \leq \alpha \tag{21}$$

□

3.4 Changepoint Estimation

After an alarm at time T, we estimate the changepoint location by selecting the split that best explains the post-split p-values as non-uniform. For each candidate $k \in \{1, \ldots, T-1\}$, we evaluate the segment $(p_{k+1}, \ldots, p_T)$ under two hypotheses:

- $\mathbf{H_0^{(k)}}$: p-values are Uniform$(0, 1)$, so each of B bins receives probability $1/B$.
- $\mathbf{H_1^{(k)}}$: bin probabilities are unknown, with a symmetric Dirichlet prior Dir$(\alpha, \ldots, \alpha)$.

We use a Bayes factor rather than a likelihood ratio because the flexible model H_1 would otherwise trivially outperform H_0 by overfitting - H_1 contains H_0 as a special case, so its MLE fit is always at least as good. Averaging over the Dirichlet prior instead of optimizing penalizes H_1 for the prior mass it wastes on configurations the data does not support. We set $\alpha = 1/2$ (Jeffreys prior), the standard non-informative choice for categorical data, which is invariant to the reparametrization of bin boundaries [Jeffreys, 1961].

Let $\mathbf{n} = (n_1, \ldots, n_B)$ denote the histogram of p-values in the post-split segment $(p_{k+1}, \ldots, p_T)$, where each n_b counts how many p-values fell into the b-th equal-width bin over $[0, 1)$, and $N = \sum_b n_b$ is the segment length. Under both hypotheses, the data likelihood is multinomial. Under H_0 every bin has probability $1/B$, so the multinomial reduces to:

$$\log p(\mathbf{n} \mid H_0) = \log \frac{N!}{\prod_b n_b!} - N \log B \tag{22}$$

Under H_1, the bin probabilities θ are unknown so we integrate them out over the Dirichlet prior:

$$\begin{aligned}\log p(\mathbf{n} \mid H_1) = \log \frac{N!}{\prod_b n_b!} &+ \log\Gamma(B\alpha) - \log\Gamma(N + B\alpha) \\ &+ \sum_{b=1}^{B} \bigl[\log\Gamma(n_b + \alpha) - \log\Gamma(\alpha)\bigr]\end{aligned} \tag{23}$$

Since the combinatorial factor appears in both likelihoods, it cancels in the log Bayes factor:

$$\begin{aligned}\log \mathrm{BF}_k &= \log p(\mathbf{n} \mid H_1) - \log p(\mathbf{n} \mid H_0) \\ &= \log\Gamma(B\alpha) - \log\Gamma(N + B\alpha) \\ &\quad + \sum_{b=1}^{B} \bigl[\log\Gamma(n_b + \alpha) - \log\Gamma(\alpha)\bigr] \\ &\quad + N \log B\end{aligned} \tag{24}$$

To identify our changepoint, we simply select $\hat{\tau} = \arg\max_k \log \mathrm{BF}_k$.

This changepoint estimate is a capability absent from all `river` baselines, which expose only a binary alarm flag. PITMonitor thus enables practitioners not just to detect that drift has occurred, but to identify how far back model outputs were already corrupted - directly actionable for deciding how much historical inference to distrust or recompute.

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, comparing against the seven included stream-drift detectors from the `river` library.

Algorithm 1 PITMonitor

```
Require: Significance level $\alpha$, number of bins $B$
Initialize: $M_0 \leftarrow 0$, histogram counts $c_1, \ldots, c_B \leftarrow 1$        ▷ Laplace prior
for $t = 1, 2, \ldots$ 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., 2021] 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 changes:

- **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, $\Delta t = 500$): The change spreads linearly across all features over a 500-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, $n_{\text{stable}} = 2{,}500$ pre-drift monitoring samples that define the null-hypothesis window for FPR estimation, and $n_{\text{post}} = 2{,}500$ post-drift samples for TPR estimation. The drift onset occurs at the boundary between the stable and post-drift segments.

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. Inputs and targets are standardized using per-feature means and standard deviations fitted on the training set. Training uses mini-batches of size 256, the Adam optimizer with initial learning rate 3×10^{-4}, a cosine annealing learning rate schedule, and 1,000 epochs. The model achieves $R^2 = 0.96$ on a held-out pre-drift test set confirming it is well-specified before monitoring begins.

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

$$U_t = \Phi\left(\frac{y_t - \mu_t}{\sigma_t}\right) \tag{25}$$

where μ_t, σ_t are the predicted mean and standard deviation and Φ is the standard normal CDF.

Detector settings.

- **PITMonitor**: false alarm bound $\alpha = 0.05$, bin number $B = 100$.
- **ADWIN** significance parameter $\delta = 0.05$, otherwise `river` defaults. Note that ADWIN's δ is not directly comparable to PITMonitor's α: δ bounds the per-step false alarm probability (subject to a Bonferroni correction over the window length), while α bounds the probability of *ever* raising a false alarm.
- **Other**: Default `river` parameters.

Baselines. We compare against seven stream-drift detectors from `river`. 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 median absolute residual on the training data): DDM, EDDM, HDDM_A, HDDM_W. These methods are designed for binary error streams and alarm when the error rate increases.

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 $N = 50$ 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, as well as distance from the true changepoint for PITMonitor only. We report:

- **TPR**: fraction of trials with a true-positive alarm (alarm fired after drift onset).
- **FPR**: fraction of trials with a false alarm (alarm fired before drift onset).
- **Mean detection delay**: mean number of samples between the true drift onset and the alarm, over true-positive trials only.

4.2 Results

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

Table 1: Drift detection results on FriedmanDrift (50 trials, $\alpha = 0.05$). Mean delay is in samples. Best TPR per scenario is **bolded**; FPR exceeding α is underlined.

	GRA			GSG			LEA		
Method	TPR	FPR	Delay	TPR	FPR	Delay	TPR	FPR	Delay
PITMonitor	**96.0%**	4.0%	80	**96.0%**	4.0%	205	0.0%	4.0%	—
ADWIN	76.0%	<u>24.0%</u>	27	76.0%	<u>24.0%</u>	27	**76.0%**	<u>24.0%</u>	121
KSWIN	2.0%	<u>98.0%</u>	14	10.0%	<u>90.0%</u>	131	2.0%	<u>98.0%</u>	801
PageHinkley	2.0%	<u>98.0%</u>	0	2.0%	<u>98.0%</u>	0	2.0%	<u>98.0%</u>	5
DDM	86.0%	<u>14.0%</u>	452	86.0%	<u>14.0%</u>	734	2.0%	<u>14.0%</u>	340
EDDM	8.0%	<u>92.0%</u>	438	8.0%	<u>92.0%</u>	639	0.0%	<u>92.0%</u>	—
HDDM_A	**96.0%**	4.0%	59	**96.0%**	4.0%	165	2.0%	4.0%	2290
HDDM_W	8.0%	<u>92.0%</u>	12	8.0%	<u>92.0%</u>	20	8.0%	<u>92.0%</u>	507

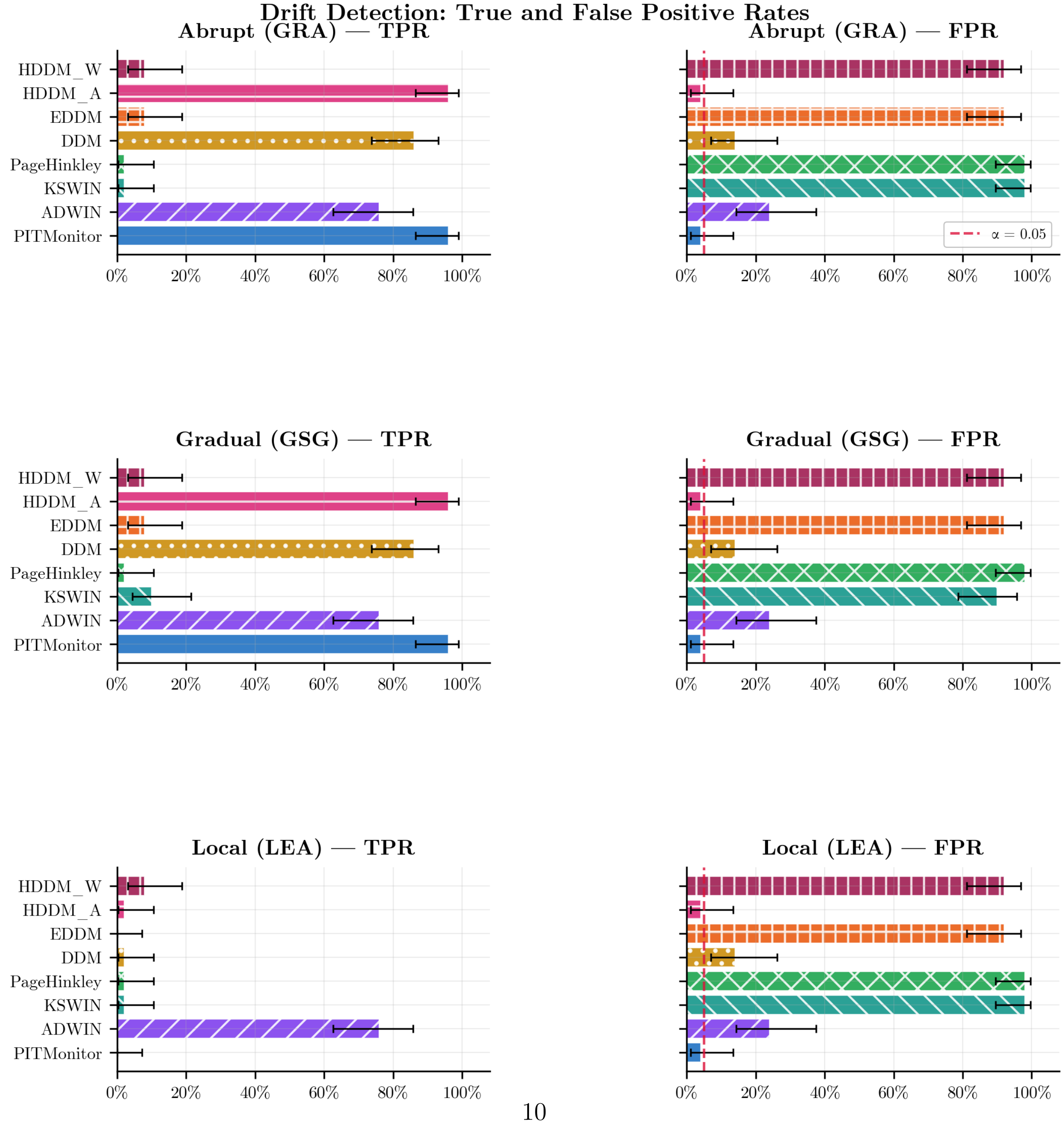

Figure 1: TPR and FPR across all detectors and drift scenarios. The red dashed line marks PITMonitor's false alarm guarantee $\alpha = 0.05$.

Type I error control. PITMonitor achieves FPR of 4.0%, within the nominal $\alpha = 0.05$ guarantee. This is consistent across all three scenarios, as expected since the pre-drift monitoring window is identical. ADWIN shows a consistent FPR of 24.0%, well above its $\delta = 0.05$ parameter; this is expected because ADWIN's δ provides a per-comparison bound (Theorem 3.1 of Bifet and Gavaldà [2007]), not an anytime-valid guarantee on the probability of ever alarming. Over a 2,500-sample stable window with many subwindow comparisons, the cumulative false alarm probability substantially exceeds the per-step bound.

Global drift (GRA, GSG). PITMonitor achieves 96.0% TPR with 4.0% FPR on both global drift scenarios, the highest power of any method with controlled false alarm rate. ADWIN achieves 76.0% TPR with 24.0% FPR; its consistently elevated FPR is an artifact of ADWIN's per-step guarantee, not a code error (see Type I error discussion above). PITMonitor's detection delay is 80 samples on GRA and 205 on GSG, compared to ADWIN's 27 on both. DDM and HDDM_A achieve 86.0% and 96.0% TPR respectively but with elevated FPR (14.0% and 4.0%), while DDM's mean delay of 452 samples (GRA) reflects its slower adaptation on the binary-thresholded input stream.

Local expanding drift (LEA). PITMonitor achieves 0.0% TPR on the LEA scenario, correctly maintaining FPR control but failing to detect the drift. ADWIN achieves 76.0% TPR (its identical detection rate across all scenarios reflects the fact that ADWIN fires on every trial; what varies is only whether the alarm occurs before or after drift onset). 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 2,500-sample post-drift window. ADWIN's squared-residual statistic appears more sensitive to partial distributional shifts than PITMonitor's PIT-based 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. Figure 2 shows the distribution of absolute changepoint error $|\hat{\tau} - \tau|$ across true-positive trials. PITMonitor's mean absolute changepoint error is 1.6 samples on GRA and 7.5 on GSG, demonstrating that the estimated changepoint closely tracks the true onset. This provides the practitioner with a starting point for diagnosing and correcting the drift.

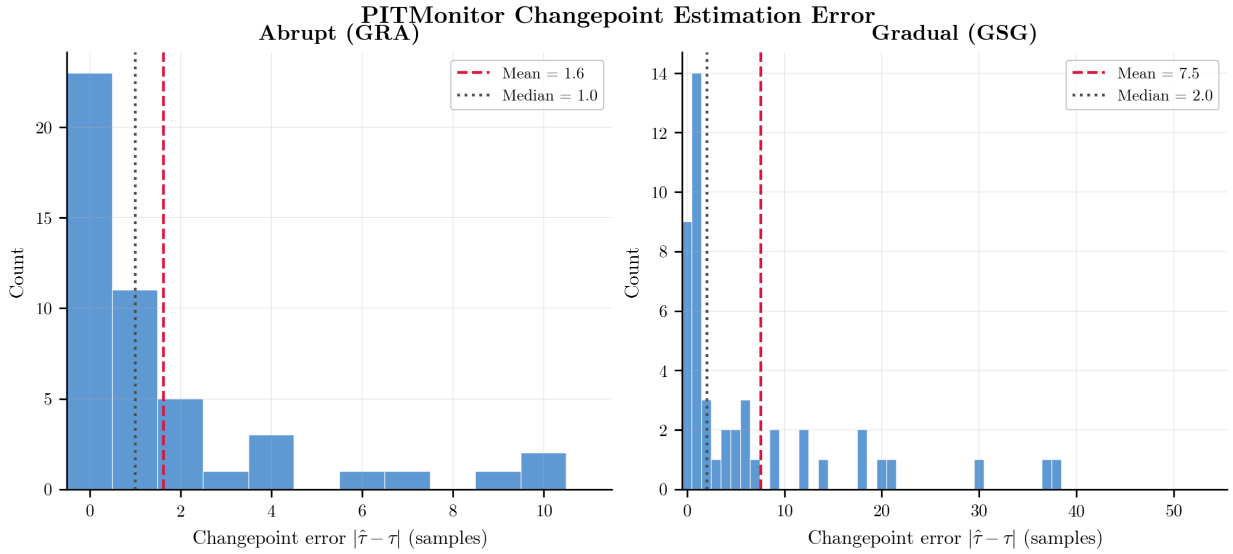

Figure 2: Distribution of PITMonitor changepoint estimation error $|\hat{\tau} - \tau|$ across true-positive trials. The abrupt (GRA) scenario yields highly concentrated estimates; the gradual (GSG) scenario shows slightly larger errors due to the extended transition window.

5 Discussion

When to use PITMonitor. PITMonitor is designed for continuous monitoring of deployed probabilistic 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
- Accurate changepoint localisation is valuable

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.

Detection delay vs. FPR control. PITMonitor's anytime-valid guarantee comes at the cost of later detection compared to ADWIN. The mean delay on GRA (76 samples) reflects the number of post-drift observations needed for the e-process to cross $1/\alpha = 20$. The most direct way to reduce delay is to accept a larger α: setting $\alpha = 0.10$ lowers the threshold to 10, requiring less accumulated evidence before an alarm fires. Users should calibrate α against their tolerance for false alarms over the full expected deployment horizon rather than treating it as a fixed convention.

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.

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 changepoint localisation and anytime-valid FPR control; ADWIN provides faster detection of partial shifts, accepting a higher false alarm rate.

Comparison to `river` baselines. The experiments reveal a clear partitioning of methods. PITMonitor and ADWIN are the only detectors achieving meaningful TPR across global drift scenarios. Their tradeoff is qualitatively different: ADWIN detects roughly 3× faster but with substantially higher FPR (24.0% vs. 4.0%). Crucially, ADWIN's elevated FPR is expected given its per-step guarantee semantics: $\delta = 0.05$ bounds the false alarm probability per comparison, not over the full stream, whereas PITMonitor's $\alpha = 0.05$ bounds the probability of *ever* alarming. On local expanding drift ADWIN is dominant. The remaining methods either have collapsed FPR (KSWIN, Page-Hinkley, HDDM_W, EDDM with default parameters) or elevated FPR (DDM at 14.0%, HDDM_A at 4.0%), 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, Grünwald 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. [2024], 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 96.0% TPR with 4.0% 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 76.0% TPR (though at 24.0% FPR). 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, and improving power on partial distributional shifts.

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

References

Sebastian Arnold, Alexander Henzi, and Johanna 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.

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

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

Bruno de Finetti. La prévision : ses lois logiques, ses sources subjectives. *Annales de l'institut Henri Poincaré*, 7(1):1–68, 1937. URL `http://eudml.org/doc/79004`.

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

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

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

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

Peter Grünwald, Rianne de Heide, and Wouter M. Koolen. Safe testing. *Journal of the Royal Statistical Society: Series B*, 86(5):1091–1128, 2024.

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

Ramesh Johari, Pete Koomen, Leonid Pekelis, and David Walsh. Always valid inference: Continuous monitoring of a/b tests. *Operations Research*, 70(3):1806–1821, 2022.

Zachary Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3122–3130, 2018.

Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. River: Machine learning for streaming data in Python. *Journal of Machine Learning Research*, 22(110):1–8, 2021.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2901–2907, 2015.

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

Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 844–853, 2021.

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

Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Glenn Shafer, Alexander Shen, Nikolai Vereshchagin, and Vladimir Vovk. Test martingales, Bayes factors and p-values. *Statistical Science*, 26(1):84–101, 2011.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. E-detectors: A nonparametric framework for sequential change detection. *The New England Journal of Statistics in Data Science*, 2(2): 229–260, 2024. doi: 10.51387/23-NEJSDS51.

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

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

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

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

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