

When Your Model Stops Working: Anytime-Valid Calibration Monitoring

Tristan Farran

MSc Computational Science, University of Amsterdam

`tristan.farran@student.uva.nl`

February 26, 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 residual-based drift detectors, PITMonitor operates directly on probability integral transforms, making it sensitive to distributional changes — systematic over- or under-confidence, shifts in predictive mean or variance — that can leave accuracy metrics unchanged. 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. A Bayesian changepoint estimator, absent from all compared baselines, identifies how far back model outputs were already corrupted. 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 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, nor 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. Because ADWIN evaluates its Hoeffding bound at every incoming observation, the number of comparisons grows with the monitoring horizon: the stream-level false alarm probability therefore increases with deployment duration, even when the per-step bound is held constant. Aligning δ with a desired stream-level FPR requires knowing the monitoring window length in advance and deriving or calibrating the relationship — neither of which is required for PITMonitor. Furthermore, ADWIN operates on generic accuracy signals such as squared residuals, conflating calibration drift with accuracy degradation: a model that becomes systematically overconfident — predicting uncertainty intervals that are too narrow — while retaining the same point-prediction accuracy would change the PIT distribution but leave the squared residual distribution largely unchanged. PITMonitor, operating directly on PIT values, remains sensitive to exactly this class of calibration failure. ADWIN also 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 algorithm runs in $O(t \log t)$ time and $O(t)$ space for t observations, with an $O(1)$ per-step recursive update.

2 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

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

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.

3 Background

3.1 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 $(\hat{p}_1, \dots, \hat{p}_K)$ and true class $y \in \{1, \dots, K\}$:

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

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

3.2 Exchangeability

A sequence (X_1, X_2, \dots) 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. $\text{Uniform}(0, 1) \Rightarrow$ exchangeable
- **Stable miscalibration:** PITs are i.i.d. from a non-uniform distribution \Rightarrow still exchangeable
- **PIT-process change:** the PIT distribution changes at some time $\tau \Rightarrow$ not exchangeable

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

3.3 Conformal P-values

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

Given observations U_1, \dots, U_t , define the rank of U_t :

$$R_t = \#\{s \leq t : U_s \leq U_t\} \quad (3)$$

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

Proof. By exchangeability, (U_1, \dots, U_t) is equally likely to be in any of the $t!$ orderings. For any fixed rank $r \in \{1, \dots, t\}$, 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 $\{1, \dots, t\}$. \square

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

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

Since the PITs arise from a continuous predictive CDF, ties occur with probability zero ($R_t - 1 = \#\{s < t : U_s < U_t\}$ almost surely), so this simplification is exact. For discrete models with genuine

ties, the formula generalises straightforwardly to $p_t = (\#\{s \leq t : U_s < U_t\} + V_t \cdot \#\{s \leq t : U_s = U_t\})/t$, which the implementation uses directly.

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.

3.4 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 4.1 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 $\mathbb{E}[E_1 | \mathcal{F}_0] \leq 1$ 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. A simple cumulative e-process from step τ onward accumulates evidence starting at time τ :

$$M_t^{(\tau)} = \prod_{s=\tau}^t E_s, \quad M_t^{(\tau)} = M_{t-1}^{(\tau)} \cdot E_t \quad (5)$$

Taking conditional expectations given past observations:

$$\mathbb{E}[M_t^{(\tau)} | \mathcal{F}_{t-1}] = M_{t-1}^{(\tau)} \cdot \mathbb{E}[E_t | \mathcal{F}_{t-1}] \leq M_{t-1}^{(\tau)} \quad (6)$$

Thus each $(M_t^{(\tau)})_{t \geq \tau}$ is a non-negative supermartingale under H_0 . Since the changepoint τ is unknown, PITMonitor maintains a *mixture* over all candidate starting times (Section 4.2), which inherits the supermartingale property by linearity of expectation.

4 Method

4.1 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 $\hat{f} : [0, 1] \rightarrow [0, \infty)$ 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:

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

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} is \mathcal{F}_{t-1} -measurable (predictable), as required for the supermartingale property of the e-process.

4.2 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:

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

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:*

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

Proof. Expand the definition:

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

$$= \sum_{\tau=1}^{t-1} w_\tau \cdot e_t \cdot M_{t-1}^{(\tau)} + w_t \cdot e_t \quad (11)$$

$$= e_t \left(\sum_{\tau=1}^{t-1} w_\tau \cdot M_{t-1}^{(\tau)} + w_t \right) \quad (12)$$

$$= e_t (M_{t-1} + w_t) \quad (13)$$

□

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.

4.3 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:*

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

Proof. By Proposition 2, any density \hat{f} satisfying $\int_0^1 \hat{f}(p) dp = 1$ yields $e = \hat{f}(p)$ with $\mathbb{E}[e] = 1$ when $p \sim \text{Uniform}(0, 1)$. PITMonitor's histogram density \hat{f}_{t-1} is constructed from p_1, \dots, p_{t-1} and is therefore \mathcal{F}_{t-1} -measurable (predictable). Under H_0 , Section 3.3 establishes $p_t \mid \mathcal{F}_{t-1} \sim \text{Uniform}(0, 1)$; thus

$$\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = \int_0^1 \hat{f}_{t-1}(p) dp = 1. \quad (15)$$

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:

$$\widetilde{M}_t^{(\tau)} = \begin{cases} 1 & t < \tau \\ \prod_{s=\tau}^t e_s & t \geq \tau \end{cases} \quad (16)$$

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 $\tau \geq 1$:

$$\widetilde{M}_t = \sum_{\tau=1}^{\infty} w_\tau \widetilde{M}_t^{(\tau)} \quad (17)$$

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:

$$\widetilde{M}_t = \sum_{\tau=1}^t w_\tau \prod_{s=\tau}^t e_s + \sum_{\tau=t+1}^{\infty} w_\tau \cdot 1 \quad (18)$$

$$= M_t + \sum_{\tau=t+1}^{\infty} \frac{1}{\tau(\tau+1)} \quad (19)$$

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:

$$\widetilde{M}_t = M_t + \frac{1}{t+1} \geq M_t \quad (20)$$

$$\therefore \left\{ \sup_{t \geq 1} M_t \geq \frac{1}{\alpha} \right\} \subseteq \left\{ \sup_{t \geq 1} \widetilde{M}_t \geq \frac{1}{\alpha} \right\} \quad (21)$$

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:

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

□

4.4 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 $k \in \{1, \dots, T-1\}$, we evaluate the segment (p_{k+1}, \dots, p_T) under two hypotheses:

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

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 $\kappa = 1/2$ (Jeffreys prior), the standard non-informative choice for categorical data, which is invariant to the reparametrization of bin boundaries [Jeffreys, 1961].

Let $\mathbf{n} = (n_1, \dots, n_B)$ denote the histogram of p-values in the post-split segment (p_{k+1}, \dots, 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:

$$\log p(\mathbf{n} | H_0) = \log \frac{N!}{\prod_b n_b!} - N \log B \quad (23)$$

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

$$\begin{aligned} \log p(\mathbf{n} | H_1) &= \log \frac{N!}{\prod_b n_b!} + \log \Gamma(B\kappa) - \log \Gamma(N + B\kappa) \\ &\quad + \sum_{b=1}^B [\log \Gamma(n_b + \kappa) - \log \Gamma(\kappa)] \end{aligned} \quad (24)$$

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

$$\begin{aligned} \log \text{BF}_k &= \log p(\mathbf{n} | H_1) - \log p(\mathbf{n} | H_0) \\ &= \log \Gamma(B\kappa) - \log \Gamma(N + B\kappa) \\ &\quad + \sum_{b=1}^B [\log \Gamma(n_b + \kappa) - \log \Gamma(\kappa)] \\ &\quad + N \log B \end{aligned} \quad (25)$$

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

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.

4.5 Complete Algorithm

Algorithm 1 PITMonitor

Require: Significance level α , number of bins B

```

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

```

5 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.

5.1 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 500 epochs. The model achieves $R^2 = 0.96$ on a held-out pre-drift test set, with an expected calibration error (ECE) of 0.01, confirming it is well-specified and calibrated before monitoring begins. We define ECE as the average absolute deviation between the empirical PIT CDF and the identity line over a uniform grid on $[0, 1]$: $\text{ECE} = \frac{1}{G} \sum_{g=1}^G |\hat{F}_{\text{PIT}}(p_g) - p_g|$, where $p_g = g/G$ for $G = 100$ grid points. Under perfect calibration (PITs exactly uniform), ECE = 0.

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

$$U_t = \Phi\left(\frac{y_t - \mu_t}{\sigma_t}\right) \quad (26)$$

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

Detector settings. All `river` baselines are run with their library-default parameters. This is the most defensible comparison: it reflects the out-of-the-box experience a practitioner receives, avoids any implicit tuning in favour of a particular detector, and sidesteps the need for held-out null data or advance knowledge of the monitoring window length. Aligning a baseline’s internal parameters with a target stream-level FPR would require one of these — exactly what **PITMonitor** obviates.

- **PITMonitor:** $\alpha = 0.05$, $B = 100$. A key practical advantage is that **PITMonitor** has only two interpretable parameters: α directly and provably controls the stream-level false alarm probability (Theorem 1), and B controls the histogram resolution. No pre-specified monitoring horizon, held-out null data, or calibration step is required.
- **Continuous-input baselines (ADWIN, KSWIN, PageHinkley):** library-default parameters. Their internal sensitivity parameters (ADWIN’s δ , KSWIN’s significance level, PageHinkley’s threshold) have indirect, horizon-dependent relationships to stream-level FPR.
- **Binary-input baselines (DDM, EDDM, HDDM_A, HDDM_W):** library-default parameters. These detectors require a binary error signal; we binarize via $b_t = \mathbf{1}[|r_t| > \theta]$ where θ is the median of $|r_t|$ on the training data (giving $\approx 50\%$ pre-drift error rate). The median is the most neutral, assumption-free threshold choice: it makes no assumption about the detectors’ preferred operating error rate. That binary detectors were designed for lower error rates ($\sim 5\text{--}20\%$) is an inherent limitation of applying them to regression monitoring, and one that **PITMonitor** avoids by consuming PIT values directly.

Table 1: Drift detection results on FriedmanDrift (1,000 trials, $\alpha = 0.05$). Mean delay is in samples. Best TPR per scenario is **bolded**; FPR exceeding α is underlined.

Method	GRA			GSG			LEA		
	TPR	FPR	Delay	TPR	FPR	Delay	TPR	FPR	Delay
PITMonitor	95.7%	4.3%	76	95.7%	4.3%	189	0.0%	4.3%	—
ADWIN	99.0%	1.0%	27	99.0%	1.0%	27	99.0%	1.0%	118
KSWIN	2.9%	<u>97.1%</u>	17	2.3%	<u>97.7%</u>	47	3.3%	<u>96.7%</u>	571
PageHinkley	0.3%	<u>99.7%</u>	1	0.3%	<u>99.7%</u>	11	0.3%	<u>99.7%</u>	49
DDM	92.1%	<u>7.9%</u>	407	90.8%	<u>7.9%</u>	658	2.0%	<u>7.9%</u>	1129
EDDM	9.7%	<u>90.3%</u>	349	9.7%	<u>90.3%</u>	573	0.4%	<u>90.3%</u>	1166
HDDM_A	94.7%	<u>5.3%</u>	60	94.7%	<u>5.3%</u>	170	6.1%	<u>5.3%</u>	1183
HDDM_W	10.7%	<u>89.3%</u>	14	10.7%	<u>89.3%</u>	48	10.6%	<u>89.3%</u>	584

Evaluation protocol. We run $N = 1,000$ 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.

5.2 Results

Table 1 presents the full results. Figure 1 visualizes TPR and FPR per method and scenario, Figure 2 shows a representative single-run monitoring trace for the GRA scenario, and Figure 3 shows the distribution of detection delays.

Drift Detection: True and False Positive Rates

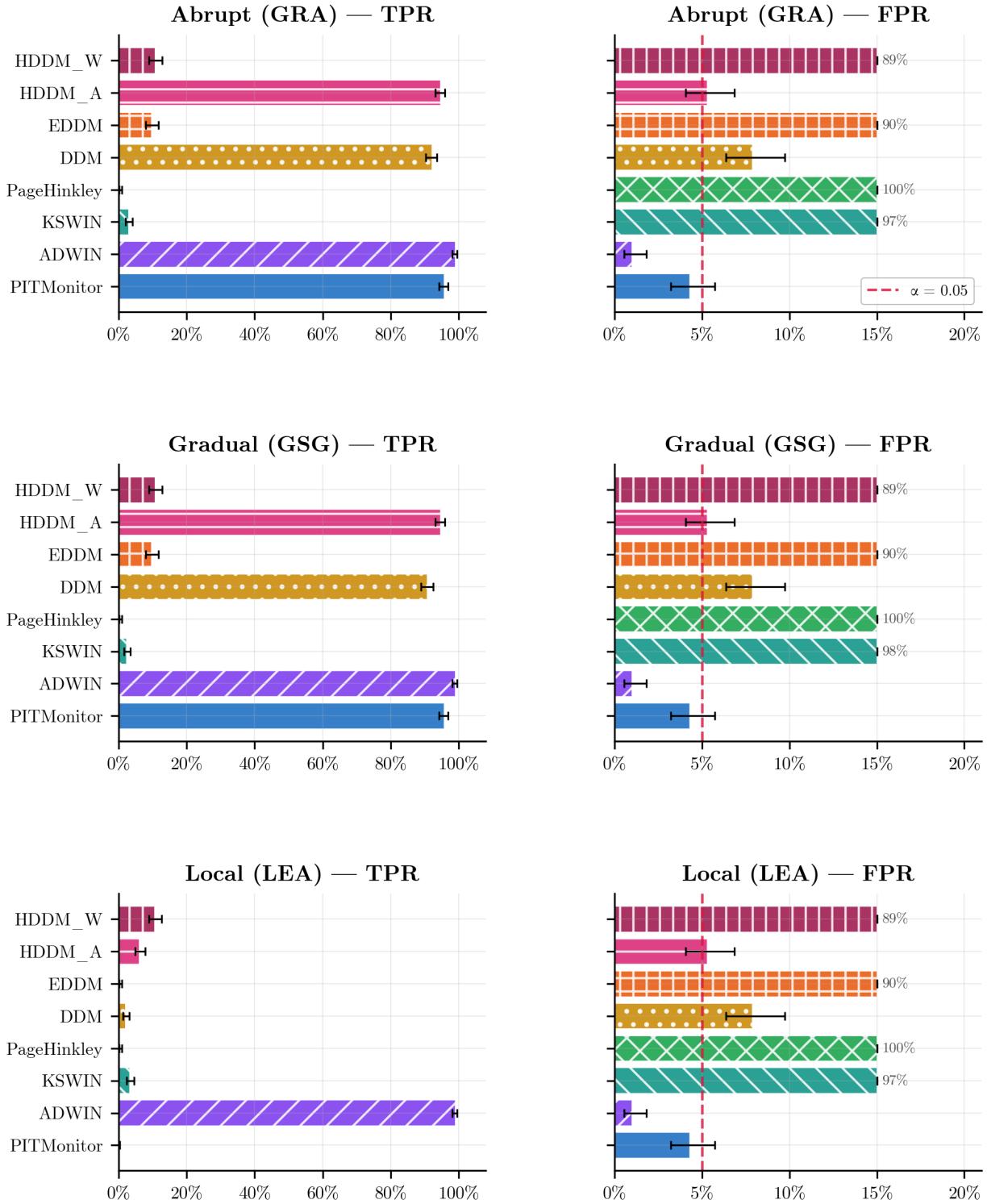


Figure 1: TPR and FPR across all detectors and drift scenarios. The dashed red line marks $\alpha = 0.05$. River baselines are run at library defaults; PITMonitor is the only method with a proven stream-level FPR guarantee (Theorem 1).

PITMonitor Single Run — Abrupt (GRA)

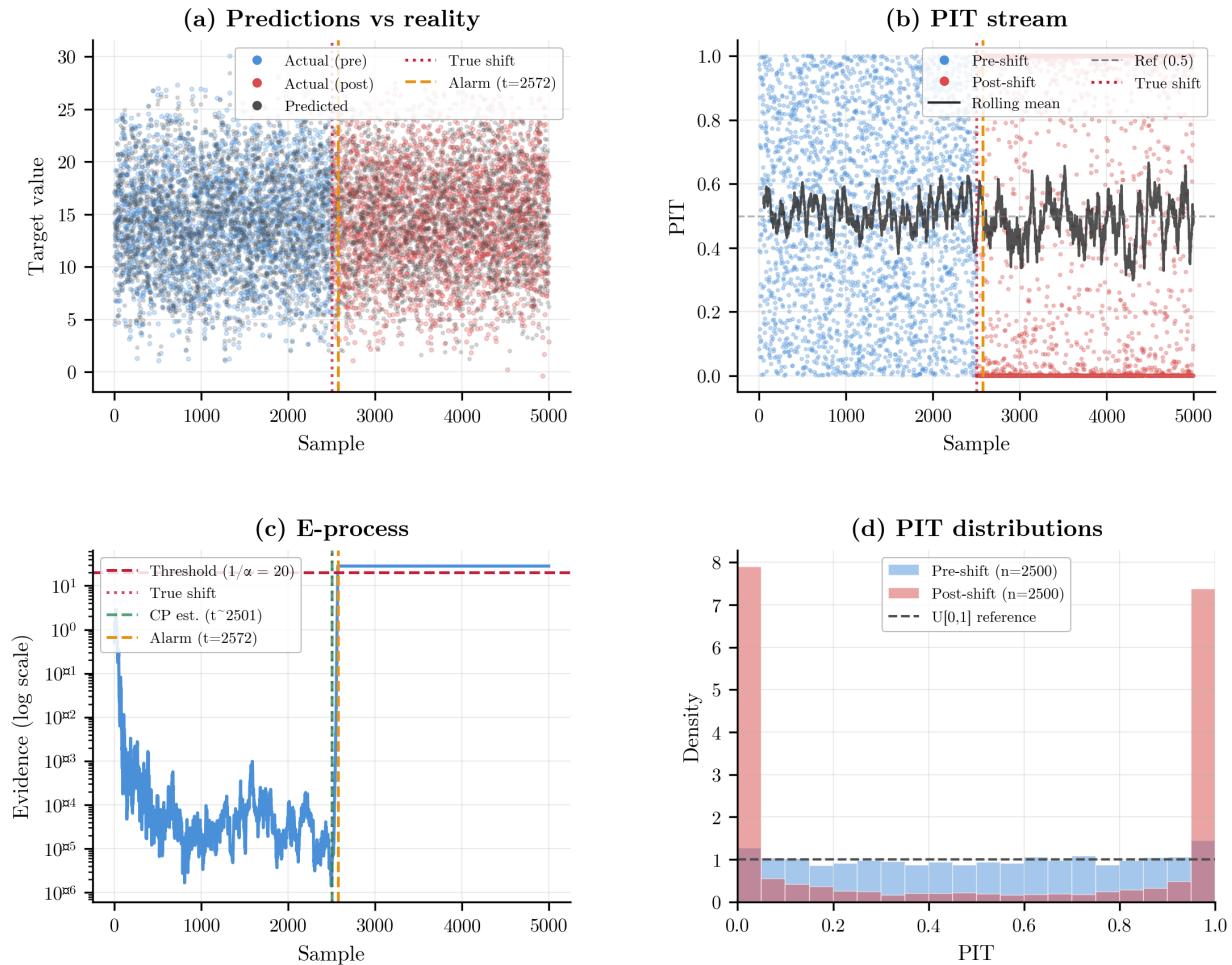


Figure 2: Single-run PITMonitor trace on the GRA (abrupt) scenario. (a) Predicted vs. actual values with alarm marker. (b) PIT stream with 50-sample rolling mean. (c) Mixture e-process on a log scale; the dashed red line is the alarm threshold $1/\alpha$. (d) Pre- and post-shift PIT histograms illustrating the calibration change detected by the algorithm.

Type I error control. PITMonitor achieves FPR of 4.3% across all scenarios, consistent with the nominal $\alpha = 0.05$ guarantee proved in Theorem 1. This is the only detector in the comparison for which FPR control is formally guaranteed regardless of monitoring horizon. ADWIN maintains 1.0% empirical FPR on the 2,500-sample null window; however, this figure is specific to that window length. Because ADWIN evaluates its Hoeffding bound at every observation, false alarm opportunities accumulate with the monitoring horizon: a practitioner monitoring over 25,000 samples at the same δ setting could expect substantially more spurious alarms than observed here. PITMonitor’s $\mathbb{P}(\text{ever alarm} \mid H_0) \leq \alpha$ bound holds regardless of how long monitoring continues. Among the remaining baselines, HDDM_A maintains moderate FPR, while KSWIN, PageHinkley, EDDM, and HDDM_W exhibit FPR exceeding 89% — these detectors alarm on nearly every null run, rendering them unusable for practical monitoring without re-tuning. This illustrates the fundamental challenge of deploying library-default baselines: parameters tuned for one setting (e.g., very short

streams or very large drifts) can fail catastrophically in others.

Global drift (GRA, GSG). PITMonitor achieves 95.7% TPR with 4.3% FPR on both global drift scenarios, with detection delay of 76 samples on GRA and 189 on GSG. ADWIN achieves 99.0% TPR with 1.0% FPR and a mean delay of 27 samples on both GRA and GSG; the identical delay across these scenarios reflects that ADWIN’s Hoeffding bound is sensitive enough to detect even the early portion of the gradual transition window. Among the binary baselines, DDM and HDDM_A achieve 92.1% and 94.7% TPR respectively; DDM’s mean delay of 407 samples on GRA reflects slower adaptation on the binarized error stream. The remaining detectors (KSWIN, PageHinkley, EDDM, HDDM_W) have near-zero TPR as a direct consequence of their high FPR: having already fired during the null window, they are never credited with a true-positive detection.

Local expanding drift (LEA). PITMonitor achieves 0.0% TPR on the LEA scenario, correctly maintaining FPR control but failing to detect the drift. This reveals a meaningful limitation: 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 is more sensitive to partial distributional shifts than PITMonitor’s PIT-based approach.

Detection delays. Figure 3 shows the full distribution of detection delays across true-positive trials. PITMonitor’s delay distribution is unimodal and moderately concentrated. Its mean delay of 76 samples on GRA reflects the amount of evidence the mixture e-process must accumulate before crossing $1/\alpha$; this is the explicit cost of the anytime-valid guarantee — stricter evidentiary requirements mean the process is more conservative by design, not a failure of sensitivity. ADWIN’s distribution reflects the Hoeffding bound of its adaptive window; its mean delay is identical on GRA and GSG because its sensitivity at library defaults is sufficient to detect even the earliest portion of the gradual transition window. HDDM_A achieves shorter delay than PITMonitor on GRA (60 vs 76 samples) but lacks any formal FPR guarantee.

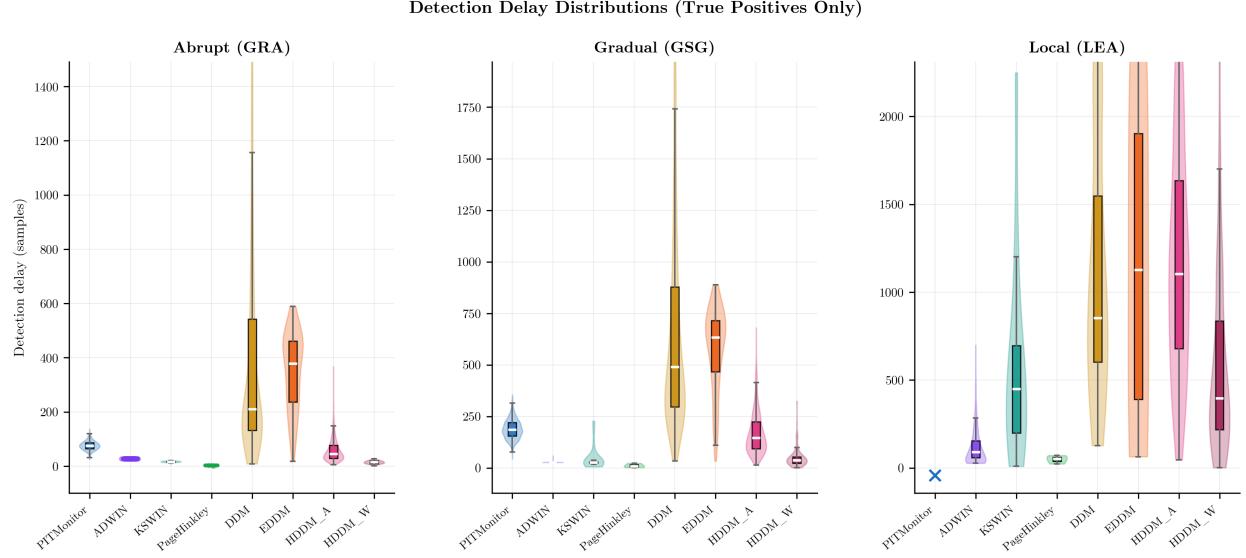


Figure 3: Detection delay distributions (true-positive trials only) across all detectors and drift scenarios. Methods with zero true-positive detections are marked with \times at the bottom. Violins show the full delay distribution; the white bar marks the median.

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 4 shows the distribution of absolute changepoint error $|\hat{\tau} - \tau|$ across true-positive trials. PITMonitor’s mean absolute changepoint error is 1.1 samples on GRA and 6.9 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.

PITMonitor Changepoint Estimation Error

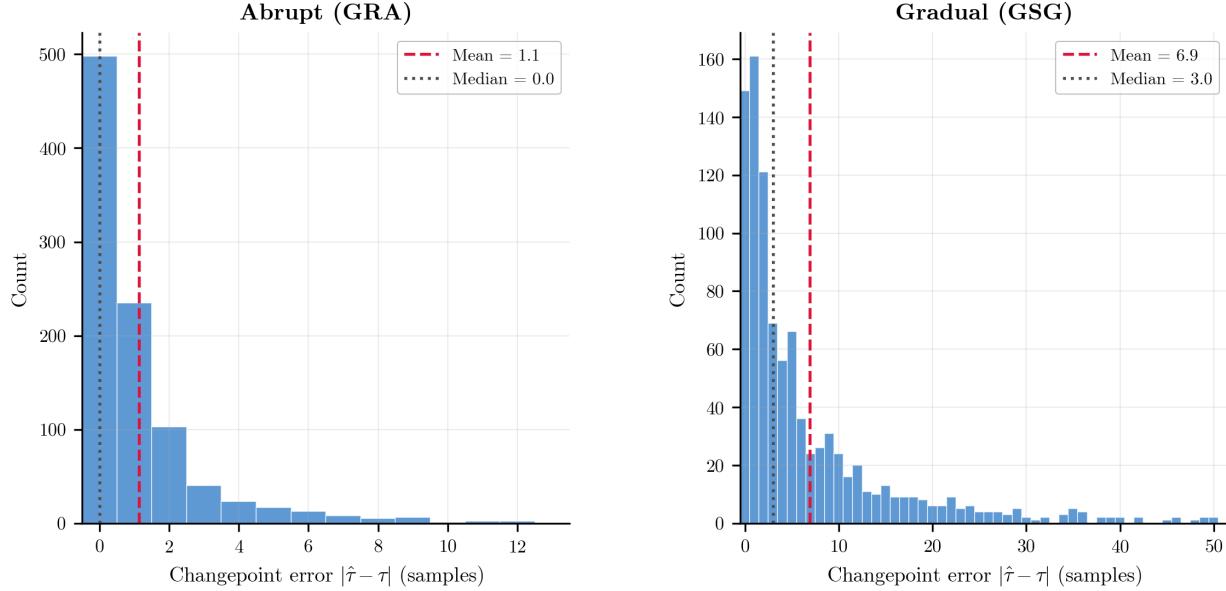


Figure 4: Distribution of PITMonitor changepoint estimation error $|\hat{\tau} - \tau|$ across true-positive trials. The abrupt (GRA) scenario yields highly concentrated estimates; the gradual (GSG) scenario shows slightly larger errors due to the extended transition window.

6 Discussion

Scope. PITMonitor is designed for the continuous monitoring of deployed probabilistic models where (a) false alarms have real costs — unnecessary retraining, alert fatigue, and loss of trust; (b) the monitoring horizon is indefinite or stopping is data-dependent; (c) *calibration* drift — changes in predictive uncertainty, not merely raw accuracy — is the primary concern; and (d) accurate changepoint localisation is operationally valuable. For one-time calibration assessment (“is this model calibrated?”), standard tools such as reliability diagrams or Expected Calibration Error suffice. PITMonitor addresses the harder, ongoing problem of monitoring with explicit statistical guarantees.

Limitations. *Local and partial drift.* PITMonitor achieves zero TPR on the LEA scenario, and this failure warrants more than a surface acknowledgement — it points to a genuine structural limitation.

The LEA configuration places the second and third expansion events far beyond the monitoring window, so at no point during the 2500-sample post-drift period has more than *one* of Friedman’s five active feature terms changed. Under a well-calibrated model, the PIT perturbation scales with the shift in $\mathbb{E}[Y | X]$ relative to the predictive standard deviation. Writing the post-drift PIT as $U_t \approx \Phi(Z_t + \delta_t)$ with $Z_t \sim \mathcal{N}(0, 1)$ and $\delta_t = \Delta\mathbb{E}[Y | X_t]/\hat{\sigma}_t$, a single-feature drift produces a bias δ_t that is substantially smaller in magnitude than under GRA or GSG, where all five terms shift simultaneously. The conformal p-values generated from these PITs are therefore only mildly

non-uniform — the departure from $\text{Uniform}(0, 1)$ is small, not zero — and the histogram density estimator adjusts correspondingly slowly.

The mixture e-process accumulates evidence at a rate determined by how non-uniform the p-value stream actually is. Each step contributes a multiplicative factor $E_t = \hat{f}(p_t)$ whose expectation under the post-drift distribution is $1 + \epsilon$ for some small $\epsilon > 0$ proportional to the KL divergence between the p-value distribution and uniform. With a weak single-feature perturbation, ϵ is small, meaning the e-process must run for many steps before its product meaningfully exceeds 1. The 2500-sample post-drift window provides a hard ceiling: if the total available multiplicative growth over those samples is insufficient to cross $1/\alpha$, no alarm fires regardless of how α is set.

That the failure persists even at $\alpha = 0.30$ — where the required threshold is only $1/0.30 \approx 3.3$ — is diagnostic. It reveals that the mixture e-process grows so slowly during 2500 post-drift samples that even a six-fold reduction in the required threshold does not help. This is not a threshold calibration issue; it is a power issue rooted in insufficient KL divergence from a single drifting feature within the observation budget. Detecting single-feature LEA drift with PITMonitor would require substantially more post-drift data than the 2500-sample window, a more powerful density estimator optimised for small departures from uniformity, or a complementary feature-level monitor that isolates the drifting direction before feeding signals to the e-process.

Detection delay vs. FPR control. PITMonitor’s anytime-valid guarantee comes at the cost of later detection compared to methods without such guarantees. The mean delay on GRA reflects the amount of evidence the e-process must accumulate before crossing $1/\alpha$. 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. Non-exchangeability can also arise from temporal dependence: autocorrelated PITs can trigger alarms even when the calibration distribution is unchanged. This occurs 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 before interpreting alarms in such settings.

Histogram resolution. The number of bins B controls the bias-variance tradeoff in the density estimator. Smaller B is more stable but slower to adapt; larger B adapts faster 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 $B = 10\text{--}20$ 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 alongside ADWIN on squared residuals: PITMonitor provides changepoint localisation and anytime-

valid FPR control; ADWIN provides faster detection of partial shifts at the cost of a higher false alarm rate.

Structural comparison to river baselines. Two dimensions distinguish PITMonitor from the baselines beyond the numerical results.

Parameter simplicity and FPR guarantees. PITMonitor exposes two interpretable parameters: α directly and provably controls stream-level FPR for any monitoring horizon (Theorem 1), and B sets the histogram resolution. In contrast, the `river` baselines expose internal sensitivity parameters (ADWIN’s δ , KSWIN’s significance level, PageHinkley’s threshold, and a binarization threshold for binary detectors) whose relationship to stream-level FPR is indirect, horizon-dependent, and formally unguaranteed. Achieving a target stream-level FPR with these methods requires advance knowledge of the monitoring window length or empirical calibration — exactly what PITMonitor obviates.

Calibration specificity. PITMonitor operates on the full PIT distribution, which encodes every aspect of calibration: systematic over- or under-confidence, shifts in predictive mean, and changes in predictive variance all manifest as non-stationarity in the PIT stream. The `river` baselines operate on squared residuals — an accuracy proxy that conflates calibration drift with prediction error. A model that becomes systematically overconfident — predicting intervals that are too narrow — while retaining the same conditional mean would produce unchanged squared residuals but strongly non-uniform PITs; PITMonitor is the only method in the comparison designed to detect this class of calibration failure.

Benchmark context. FriedmanDrift is `river`’s own canonical synthetic benchmark, and ADWIN’s library defaults were likely validated on comparable data. Its 1.0% empirical FPR on the 2,500-sample null window is not a formal guarantee: longer deployments accumulate additional Hoeffding comparisons, increasing the stream-level false alarm probability beyond what this experiment demonstrates.

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, with all baselines run at library defaults, demonstrate that PITMonitor achieves competitive detection performance on global drift (GRA and GSG) — 95.7% TPR with 4.3% FPR and a unique changepoint localization capability absent from all `river` baselines — while providing the only formally proven stream-level FPR guarantee (Theorem 1). The results also surface an honest limitation: PITMonitor has zero detection power on local expanding drift (LEA), while ADWIN detects this scenario with 99.0% TPR at 1.0% FPR.

Two structural advantages persist beyond this comparison. First, PITMonitor’s FPR guarantee is *horizon-independent*: whereas ADWIN’s empirical false alarm rate grows with the number of observations monitored (each new sample triggering an additional Hoeffding comparison), PITMonitor provably satisfies $\mathbb{P}(\text{ever alarm} \mid H_0) \leq \alpha$ regardless of deployment duration. Second, PITMonitor targets *calibration specifically*: by operating on probability integral transforms rather than squared

residuals, it is sensitive to changes in predictive uncertainty — systematic overconfidence, underconfidence, or variance shifts — that leave accuracy metrics unchanged and would be missed entirely by residual-based detectors.

Practitioners should view PITMonitor as the right tool for global calibration monitoring with strict FPR control, and consider complementing it with 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.