When Your Model Stops Working: Anytime-Valid Calibration Monitoring

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

March 5, 2026

Abstract

Practitioners monitoring deployed probabilistic models face a fundamental trap: any fixed-sample test applied repeatedly over an unbounded stream will eventually raise a false alarm, even when the model is perfectly stable. We present PITMonitor, an *anytime-valid* calibration monitor built on the e-detector framework. By testing the exchangeability of Probability Integral Transform (PIT) values, PITMonitor guarantees $\mathbb{P}(\text{ever alarm} \mid H_0) \leq \alpha$ for all time with no pre-specified monitoring horizon. We prove Type I error control via Ville's inequality and evaluate PITMonitor on three drift scenarios from `river`'s FriedmanDrift dataset, comparing against the seven included detectors. PITMonitor achieves detection rates competitive with the best-performing baseline on global drift scenarios, with substantially longer detection delays on local expanding drift, while being the only calibration-specific monitor in the comparison to provide both an anytime-valid false alarm guarantee and Bayesian changepoint localization. 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 encounter non-stationary processes, regime shifts, and concept drift, which can cause calibration to degrade drastically with consequential effects downstream. Practitioners must therefore monitor their models continuously, seeking to answer two questions: *has* the world changed, and if so, *when*? In practice, this monitoring often relies on 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 localization.

ADWIN [Bifet and Gavaldà, 2007] improves on fixed-window methods by adapting its window size to detected changes, bounding the false alarm probability *within a single window* via a Hoeffding inequality. However, this per-window guarantee does not translate to stream-level control: because ADWIN evaluates its bound at every incoming observation, the number of tests grows unboundedly with the monitoring horizon, and therefore so does the probability of ever raising a false alarm. ADWIN also operates on generic accuracy signals such as squared residuals, conflating calibration drift with accuracy degradation; a model that becomes systematically overconfident would alter the PIT distribution while leaving residuals largely unchanged. Finally, ADWIN does not provide changepoint estimates, limiting its utility for reliable, informative long-term calibration monitoring.

We propose PITMonitor, an anytime-valid calibration monitoring method with five 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 localization 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 drift that breaks exchangeability of the PIT sequence will.[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.
5. **Distribution-free:** PITMonitor specifies neither the pre- nor post-change distributions. The histogram estimator adapts to whatever form drift takes without requiring a fixed alternative.

2 Related Work

Calibration Assessment

Classical calibration metrics include Expected Calibration Error [Guo et al., 2017], 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 static forecast evaluation in econometrics [Diebold et al., 1998] and weather prediction [Gneiting and Katzfuss, 2014], though these applications are batch-oriented and do not address sequential monitoring with false alarm control - the gap which PITMonitor aims to fill.

Distribution Shift Detection

Methods for detecting covariate shift include two-sample tests [Rabanser et al., 2019], domain classifiers [Lipton et al., 2018], and conformal methods for prediction under label shift [Podkopaev and Ramdas, 2021]. These address changes in the input or label distribution, while our work focuses on the *output* side: identifying when predicted probabilities no longer match outcome frequencies.

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

Sequential Calibration Testing

Arnold et al. [2023] proposed e-values for testing forecast calibration, focusing on whether PITs are Uniform(0,1). PITMonitor differs in three ways. First, we test exchangeability rather than uniformity and thus PITMonitor alarms only on *changes* in calibration - the more natural null for monitoring where some miscalibration is often irreducible. Second, we formulate the problem as changepoint detection rather than goodness-of-fit testing, enabling localization of *when* calibration changed, not just *whether* it deviates from uniformity. Third, PITMonitor uses a nonparametric plug-in betting strategy (in the spirit of Fedorova et al. [2012]) rather than a parametric alternative.

E-values

The e-value framework has seen rapid development [Vovk and Wang, 2021, Ramdas et al., 2023, Grünwald et al., 2024]. The e-detector framework for changepoint detection was introduced by Shin et al. [2024]. PITMonitor is an instantiation of their e-SR procedure; our contribution is the conformal e-value construction that bridges exchangeability testing with their framework, enabling the anytime-valid false alarm guarantee (Theorem 1) as well as Bayesian changepoint localization.

Changepoint Detection

Classical methods include CUSUM [Page, 1954] and Shiryaev-Roberts procedures [Shiryaev, 1963, Pollak, 1985]. These assume known pre-change and post-change distributions, while the e-detector approach provides nonparametric changepoint detection with anytime-valid false alarm 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 approximately $(100 \times p)\%$ of the time. The *probability integral transform* $U = \hat{F}(y)$, provides a universal tool for assessing calibration, as it satisfies $U \sim \text{Uniform}(0, 1)$ whenever $\hat{F}$ is the true distribution of Y [Dawid, 1984]. Therefore, non-uniformity of the PIT distribution indicates miscalibration, and changes in the PIT distribution indicate changes in calibration.

3.2 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, 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], a fully non-parametric construction that is valid under exchangeability without further assumptions.

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

$$R_t = \#\{s \leq t : U_s \leq U_t\} \tag{1}$$

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. Since PITs arise from a continuous predictive CDF, ties occur with probability zero, so the rank R_t is almost surely well-defined. 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 $\{1, \ldots, t\}$. □

While Proposition 1 establishes that R_t is discrete uniform on $\{1, \ldots, t\}$, the density-based e-value construction requires $p_t \sim \text{Uniform}(0,1)$ exactly: without this, $\mathbb{E}[e_t \mid \mathcal{F}_{t-1}] = 1$ holds only approximately due to discretisation. Adding a uniform jitter converts the discrete rank to an exactly continuous uniform p-value:[2]

$$p_t = \frac{R_t - 1 + V_t}{t}, \quad V_t \sim \text{Uniform}(0,1) \tag{2}$$

Under H_0 (exchangeability), these p-values are exactly $\text{Uniform}(0,1)$. 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] It should be noted that the jitter V_t introduces an additional source of randomness beyond the data, and thus PITMonitor's output is stochastic given a fixed stream, though the additional variability is negligible as it is $O(1/t)$.

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. Under alternatives, 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, typically 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 [Shafer et al., 2011, Ramdas et al., 2023]. If e_1, e_2 are 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 filtration through time t), their product remains a valid e-value. A simple cumulative e-process accumulates evidence from time τ onwards:

$$M_t^{(\tau)} = \prod_{s=\tau}^{t} e_s, \quad M_t^{(\tau)} = M_{t-1}^{(\tau)} \cdot e_t \tag{3}$$

Taking conditional expectations given past observations:

$$\mathbb{E}[M_t^{(\tau)} \mid \mathcal{F}_{t-1}] = M_{t-1}^{(\tau)} \cdot \mathbb{E}[e_t \mid \mathcal{F}_{t-1}] \leq M_{t-1}^{(\tau)} \tag{4}$$

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 *weighted mixture* over all candidate starting times (Section 4.2).

4 Method

4.1 E-values via Density Betting

We construct e-values from conformal p-values using the betting framework of Shafer et al. [2011], treating each p-value as a wager against the null. Before observing p_t, we specify a density function $\hat{f}(p)$ satisfying $\int_0^1 \hat{f}(p)\, dp = 1$, thereby encoding our prior belief about where p_t will concentrate.

Proposition 2 (Density Betting Yields Valid E-values)**.** *Let $\hat{f} : [0,1] \to [0,\infty)$ be any density function satisfying $\int_0^1 \hat{f}(p)\, dp = 1$. If $p \sim \mathrm{Uniform}(0,1)$, then $e = \hat{f}(p)$ satisfies $\mathbb{E}[e] = 1$.*

PITMonitor uses a histogram plug-in density estimator for betting on conformal p-values (for the broader plug-in martingale lineage, see Fedorova et al. [2012]), which places mass proportional to where past p-values concentrated and requires no specification of a fixed parametric alternative:

$$\hat{f}(p) = B \cdot \frac{c_b}{\sum_j c_j} \quad \text{for } p \in \text{bin } b \tag{5}$$

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.

By adapting our density to observed concentration patterns, we bet in the right direction. Under exchangeability, p-values scatter uniformly, and $\mathbb{E}[e_t] = 1$. If exchangeability breaks, p-values cluster, the histogram learns these concentration patterns, and we achieve $\mathbb{E}[e] > 1$ whenever deviations are sufficiently persistent, generating detection power. We update the histogram *after* computing e_t, ensuring $\hat{f}$ is 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 would accumulate excess noise that dilutes its power. Following Shin et al. [2024]'s e-SR construction, we address this by maintaining a weighted mixture over all possible changepoint times:

$$M_t = \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} \tag{6}$$

where $M_t^{(\tau)} = \prod_{s=\tau}^{t} e_s$ denotes the evidence accumulated from time τ onward (defined for $\tau \leq t$). Since each component $M_t^{(\tau)}$ forms a valid e-process sensitive to drift beginning at τ, the mixture is sensitive to changepoints at all times while remaining a valid e-process by linearity of expectation.

We use $w_\tau = 1/(\tau(\tau+1))$ as this sequence satisfies $\sum_{\tau=1}^{\infty} w_\tau = 1$ exactly, as required for the e-SR, and enables an efficient recursion that avoids maintaining separate products for each τ:

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

$$M_t = e_t \cdot (M_{t-1} + w_t) \tag{7}$$

Proof. Expand the definition:

$$\begin{aligned} M_t &= \sum_{\tau=1}^{t} w_\tau \cdot M_t^{(\tau)} && (8) \\ &= \sum_{\tau=1}^{t-1} w_\tau \cdot e_t \cdot M_{t-1}^{(\tau)} + w_t \cdot e_t && (9) \\ &= e_t \left(\sum_{\tau=1}^{t-1} w_\tau \cdot M_{t-1}^{(\tau)} + w_t \right) && (10) \\ &= e_t(M_{t-1} + w_t) && (11) \end{aligned}$$

□

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 e-processes separately.

4.3 Type I Error Control

The following instantiates the e-SR result of Shin et al. [2024] for PITMonitor's specific construction, with Ville's inequality [Ville, 1939] supplying the anytime-valid guarantee.

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

$$\mathbb{P}\left(\sup_{t \geq 1} M_t \geq \frac{1}{\alpha} \right) \leq \alpha \tag{12}$$

Proof. 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:

$$\widetilde{M}_t^{(\tau)} = \begin{cases} 1, & t < \tau, \\ \prod_{s=\tau}^{t} e_s, & t \geq \tau. \end{cases} \tag{13}$$

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

$$\widetilde{M}_t = \sum_{\tau=1}^{\infty} w_\tau \, \widetilde{M}_t^{(\tau)}. \tag{14}$$

Since $\widetilde{M}_t^{(\tau)} = 1$ for $\tau > t$, we can expand:

$$\begin{aligned} \widetilde{M}_t &= \sum_{\tau=1}^{t} w_\tau \prod_{s=\tau}^{t} e_s + \sum_{\tau=t+1}^{\infty} w_\tau & (15) \\ &= M_t + \sum_{\tau=t+1}^{\infty} \frac{1}{\tau(\tau+1)}. & (16) \end{aligned}$$

With $w_\tau = \frac{1}{\tau(\tau+1)}$, the tail 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

$$M_t \leq M_t + \frac{1}{t+1} = \widetilde{M}_t, \tag{17}$$

$$\therefore \left\{ \sup_{t \geq 1} M_t \geq \frac{1}{\alpha} \right\} \subseteq \left\{ \sup_{t \geq 1} \widetilde{M}_t \geq \frac{1}{\alpha} \right\}. \tag{18}$$

Since each $(\widetilde{M}_t^{(\tau)})_{t \geq 0}$ is a non-negative supermartingale with $\widetilde{M}_0^{(\tau)} = 1$. By Tonelli's theorem,

$$\begin{aligned} \mathbb{E}[\widetilde{M}_t \mid \mathcal{F}_{t-1}] &= \sum_{\tau=1}^{\infty} w_\tau \, \mathbb{E}[\widetilde{M}_t^{(\tau)} \mid \mathcal{F}_{t-1}] & (19) \\ &\leq \sum_{\tau=1}^{\infty} w_\tau \, \widetilde{M}_{t-1}^{(\tau)} = \widetilde{M}_{t-1}, & (20) \end{aligned}$$

so $(\widetilde{M}_t)$ is a non-negative supermartingale with $\widetilde{M}_0 = \sum_{\tau=1}^{\infty} w_\tau = 1$. Applying Ville's inequality,

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

□

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 $k \in \{1, \ldots, T-1\}$, 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$(\kappa, \ldots, \kappa)$.[3]

We use a Bayes factor rather than a likelihood ratio because H_1 would otherwise trivially outperform H_0 by overfitting, since it contains H_0 as a special case. Averaging over the Dirichlet prior penalizes H_1 for the probability mass it wastes on configurations the data does not support. We set $\kappa = 1/2$, since it 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, where each n_b counts how many p-values fell into the b-th equal-width bin, and $N = \sum_b n_b$ is the segment length.

Under H_0 every bin has probability $1/B$, so the multinomial log-likelihood 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:

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

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

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

We then identify the changepoint simply as $\hat{\tau} = \arg\max_k \log \mathrm{BF}_k$.[4]

This localization capability is 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 approximately how far back model outputs may have been affected by the shift - directly actionable for deciding how much historical inference to distrust or recompute.

[3] It should be noted that this is a heuristic approximation since the p_t sequence carries dependence across time.

[4] Note that this is a batch computation run after an alarm, not an online update.

4.5 Complete Algorithm

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
    $b \leftarrow \min(\lfloor p_t \cdot B \rfloor + 1,\, B)$    ▷ Bin count adjustment
    $e_t \leftarrow B \cdot c_b / \sum_{j=1}^{B} c_j$    ▷ E-value from density
    $c_b \leftarrow c_b + 1$    ▷ Histogram update
    $w_t \leftarrow 1/(t \cdot (t + 1))$    ▷ Deterministic 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
```

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): All features change simultaneously at an abrupt onset, representing a rapid regime shift.
- **GSG** (Global Slow Gradual): 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. A training stream consisting of $n_{\text{train}} = 10{,}000$ pre-drift samples is generated, along with 10,000 independent test streams containing $n_{\text{stable}} = 2{,}500$ 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 pre- and post-drift segments.

Predictive model. We train 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. The model is trained once on the training stream and subsequently held fixed across all 10,000 Monte Carlo test trials.

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. 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, as would be required to align a baseline's parameters with a target stream-level FPR.

- **PITMonitor**: Significance level $\alpha = 0.05$, number of bins $B = 100$. 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.
- **Continuous-input baselines (ADWIN, KSWIN, PageHinkley)**: library-default parameters. Their internal sensitivity parameters (ADWIN's δ, KSWIN's significance level, PageHinkley's threshold) have opaque, 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. Since these methods were designed for classification, no guidance exists for thresholding, and the median is the most assumption-free choice.

Evaluation protocol. We run $N = 10{,}000$ 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.

Table 1: Drift detection results on FriedmanDrift (10,000 trials, $\alpha = 0.05$).

	GRA			GSG			LEA		
Method	TPR	FPR	Delay	TPR	FPR	Delay	TPR	FPR	Delay
PITMonitor	96.2%	3.8%	77	96.2%	3.8%	189	96.2%	3.8%	1919
ADWIN	99.1%	0.9%	27	99.1%	0.9%	27	99.1%	0.9%	115
KSWIN	2.9%	97.1%	16	2.8%	97.2%	172	2.8%	97.2%	656
PageHinkley	0.4%	99.6%	1	0.4%	99.6%	6	0.4%	99.6%	70
DDM	90.8%	9.2%	405	90.0%	9.2%	666	22.0%	9.2%	2050
EDDM	8.7%	91.3%	344	8.7%	91.3%	562	1.6%	91.3%	1759
HDDM_A	94.0%	6.0%	60	94.0%	6.0%	168	52.5%	6.0%	2032
HDDM_W	9.4%	90.5%	15	9.4%	90.5%	46	9.4%	90.5%	586

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 GSG scenario, and Figure 3 shows the distribution of detection delays. The null window is drawn from the same pre-drift distribution for all three scenarios, so each detector's FPR is near-identical across scenarios by construction.

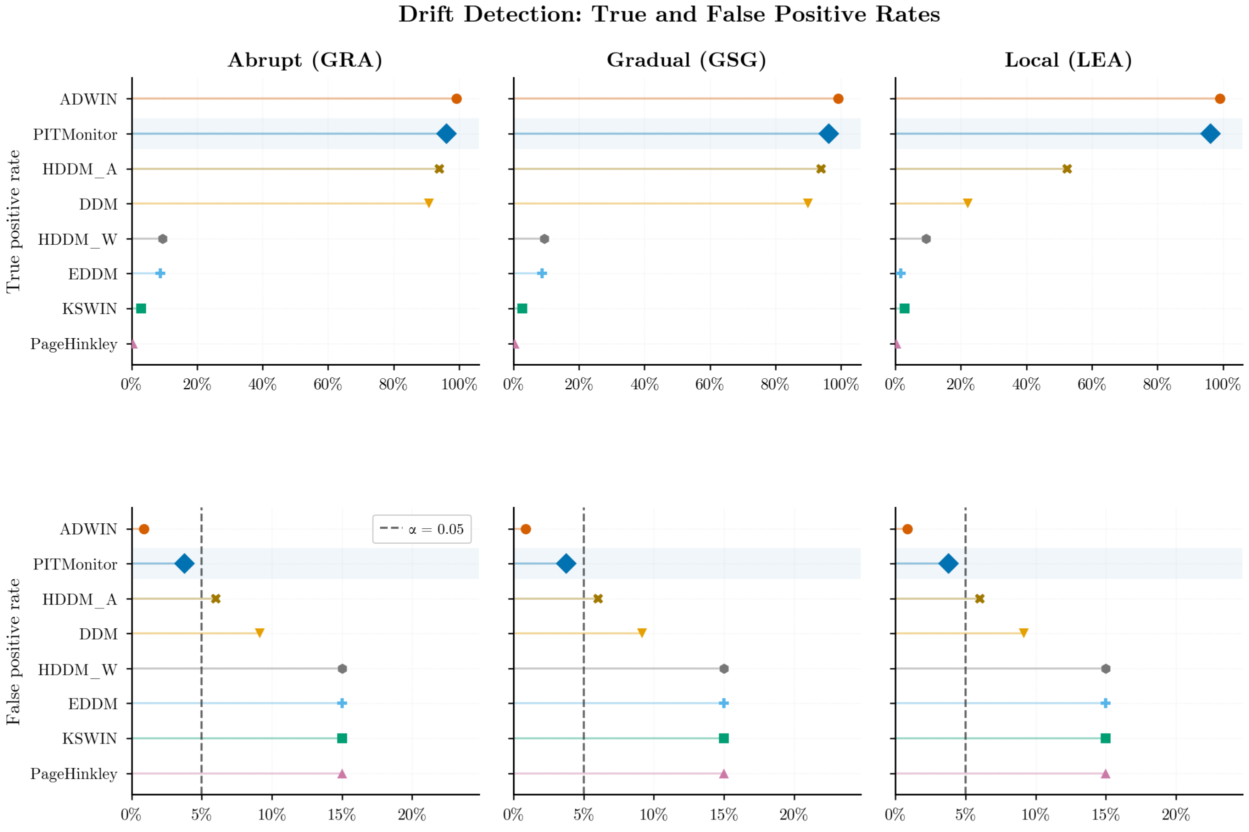

Figure 1: TPR and FPR across all detectors and drift scenarios. The dashed line marks $\alpha = 0.05$.

Type I error control. PITMonitor achieves FPR of 3.8% across all scenarios, consistent with the nominal $\alpha = 0.05$ guarantee in Theorem 1. In this benchmark. ADWIN yields 0.9% empirical FPR on the 2,500-sample null window, but this number is specific to the chosen horizon and dataset. Among the other baselines, DDM and HDDM_A achieve 9.2% and 6.0% FPR, while KSWIN, PageHinkley, EDDM, and HDDM_W exceed 89% FPR with default settings.

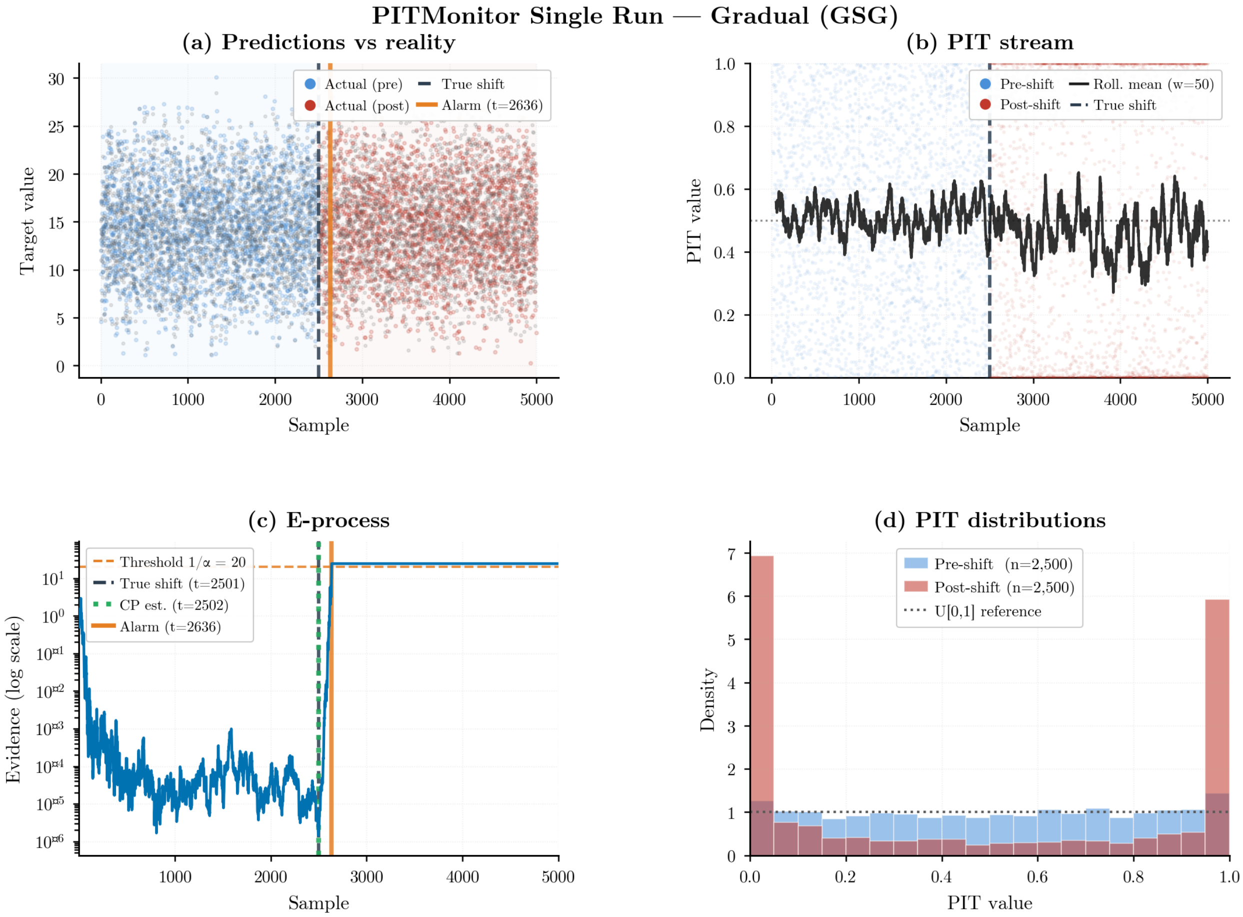

Figure 2: Single-run PITMonitor trace. (a) Predicted vs. actual values. (b) PIT stream. (c) Mixture e-process on a log scale; dashed line at the threshold $1/\alpha$. (d) Pre- and post-shift PIT histograms.

Global drift (GRA, GSG). PITMonitor achieves 96.2% TPR with 3.8% FPR on both global scenarios, with mean delays of 77 (GRA) and 189 (GSG) samples. ADWIN attains higher TPR (99.1%) and faster delay (27), but its false-alarm behavior here remains an empirical estimate tied to this finite monitoring window. DDM and HDDM_A reach 90.8% and 94.0% TPR respectively.

Local expanding drift (LEA). On LEA, PITMonitor maintains 96.2% TPR with 3.8% FPR, but mean delay rises to 1919 samples. This pattern is consistent with the expanding-drift structure: early phases induce weaker PIT distortion, so evidence accumulates slowly until later expansion stages. ADWIN detects earlier in this scenario (115 mean delay, 99.1% TPR), indicating a speed-guarantee tradeoff in this benchmark.

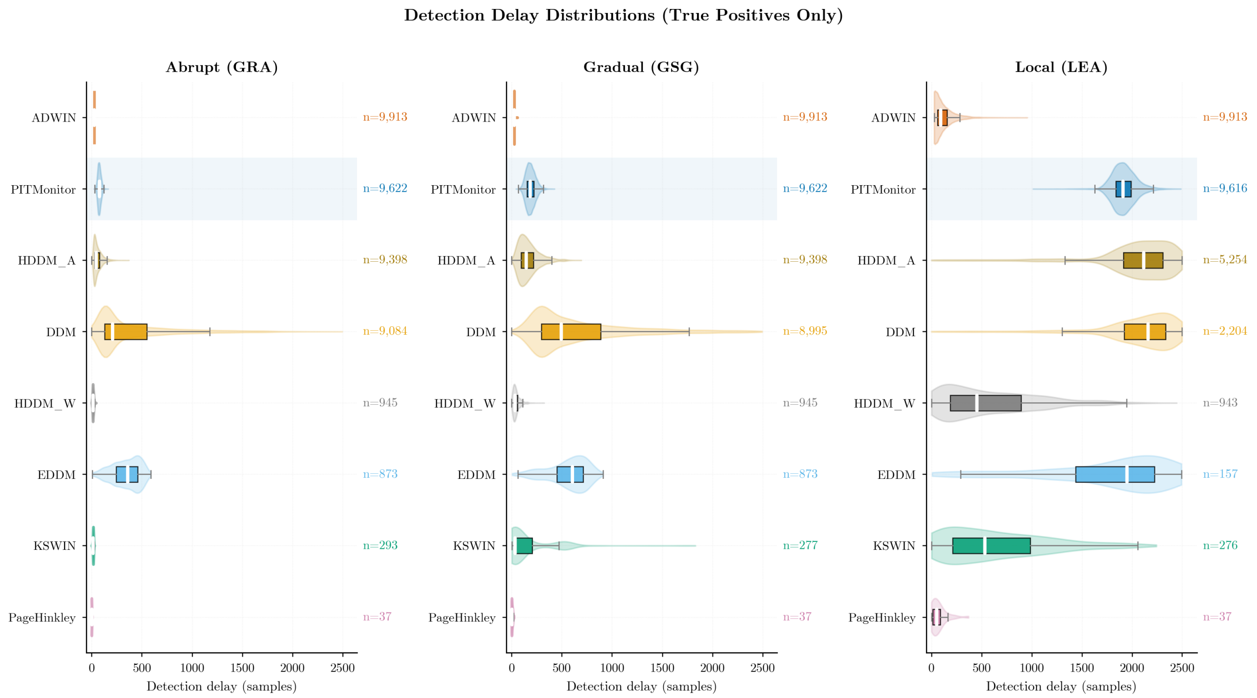

Figure 3: Detection delay distributions (true-positive trials only) across all detectors and drift scenarios. Violins show the full delay distribution, while the white bar marks the median.

Detection delays. Figure 3 reports delay distributions over true-positive trials only. Accordingly, it should be interpreted jointly with TPR: short delay at low TPR reflects performance on few runs.

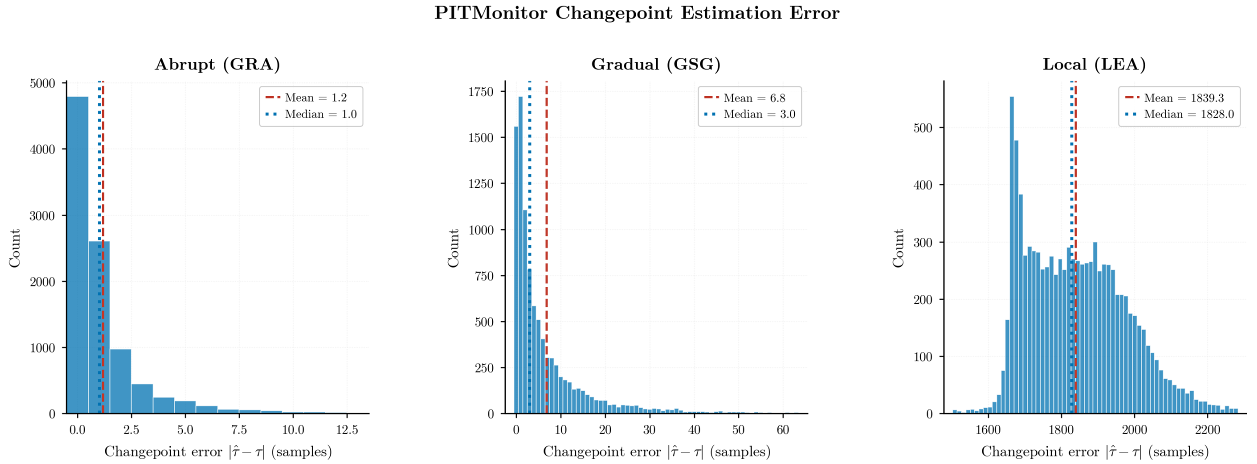

Figure 4: Distribution of PITMonitor changepoint estimation error $|\hat{\tau} - \tau|$ across true-positive trials.

Changepoint localization. On GRA and GSG, mean absolute error is 1.2 and 6.8 samples, indicating accurate localization under global drift. On LEA, mean error increases to 1839.3 because the estimator selects the split with strongest post-split non-uniformity, which typically occurs in later expansion phases (near offset 1,666) rather than at initial onset.

6 Discussion

Scope. PITMonitor is intended for continuous monitoring of deployed probabilistic models when false alarms are costly, monitoring duration is open-ended, and calibration drift (not only accuracy drift) is operationally important. For one-time calibration checks or accuracy-only monitoring without need of false-alarm control or changepoint localization, standard diagnostics remain appropriate.

Limitations.

Expanding drift. PITMonitor detects LEA drift with 96.2% TPR but with 1919 mean delay, reflecting slower evidence accumulation when only a subset of features has shifted.

Changepoint localization on expanding drift. localization is less reliable under sequential expansion: the estimator targets the phase of strongest distributional change, which can lag the true onset.

Detection delay vs. FPR control. Anytime-valid control can increase delay relative to methods without stream-level guarantees. In practice, α should be set against long-run false-alarm tolerance rather than treated 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, for example using the Ljung-Box test or inspecting the sample autocorrelation function. A practical mitigation is to monitor a thinned PIT sequence (using a lag larger than the estimated dependence horizon), which reduces serial dependence at the cost of slower detection.

Persistence requirement. PITMonitor reacts to accumulated evidence, not single-step anomalies. Under non-exchangeability, the adaptive histogram attempts to track the current concentration pattern of p-values; when tracking is accurate, expected log-growth is positive (i.e., $\mathbb{E}[\log e_t] > 0$) and cumulative evidence increases until crossing the alarm threshold. If deviations are brief, rapidly reversing, or drift faster than the estimator can track, growth is reduced or intermittent, so alarms may be delayed or absent.

Practical recommendations. *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 and should be scaled with the expected number of monitoring samples. Additionally, during the early monitoring phase (roughly the first B observations), the histogram is dominated by the Laplace prior and e-values are nearly 1 regardless of the data, yielding low sensitivity. Practitioners expecting drift immediately after deployment may wish to use a smaller B or include a warm-start period before monitoring begins.

Structural comparison to `river` baselines.

Parameter simplicity and FPR guarantees. In contrast to PITMonitor's two parameters (α bounding stream-level FPR, B setting histogram resolution), the `river` baselines expose many internal sensitivity parameters 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 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; the strong performance of library-default parameters on this benchmark should be interpreted in that context.

7 Conclusion

We presented PITMonitor, an anytime-valid method for monitoring calibration of deployed probabilistic models. 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 baselines at library defaults, show that PITMonitor attains competitive TPR on global drift (96.2% vs. ADWIN's 99.1%) while keeping empirical FPR under the target level (3.8% at $\alpha = 0.05$). On local expanding drift, PITMonitor retains comparable TPR but with substantially larger delay, reflecting the slower evidence accumulation when only a subset of features has shifted.

Three structural advantages persist beyond the numerical comparison. First, PITMonitor's FPR guarantee is horizon-independent: 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 that leave accuracy metrics unchanged and would be missed by residual-based detectors. Third, it provides actionable changepoint estimates that identify how far back model outputs were corrupted, directly informing decisions about how much historical inference to distrust or recompute.

In practice, PITMonitor is most suitable when formal long-run error control, changepoint localization, and calibration-specific monitoring are priorities. When gradually expanding drift is anticipated, combining PITMonitor with a faster residual-based detector can improve early warning at the cost of weaker long-run false-alarm guarantees.

Future work has three directions. First, for temporally dependent predictions, we aim to develop a dependence-robust construction (beyond pragmatic thinning, which reduces power). Second, for multivariate outputs, a practical baseline is to reduce each prediction-outcome pair to a scalar PIT and reuse the same monitoring pipeline; developing fully multivariate transforms (e.g., Rosenblatt transforms) that better preserve joint dependence remains open. Third, we aim to improve detection power and changepoint localization under 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. doi: 10.1214/23-AOAS1768.

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. 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.

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.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

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

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.

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. doi: 10.1214/10-STS347.

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.