Breakpoint based online anomaly detection

Etienne Krönert, Dalila Hattab, Alain Celisse

Breakpoint based online anomaly detection

Etienne Krönert[a,b,c,*], Dalila Hattab[a], Alain Celisse[b,d]

[a] *FSLab, Worldline, 59000, Seclin, France*

[b] *Modal Team, INRIA, 59000, Lille, France*

[c] *University of Lille, CNRS, UMR 8524 , Laboratoire Paul Painlevé, 59000, Lille, France*

[d] *Laboratoire SAMM, Université Paris 1 - Panthéon Sorbonne, 90 rue de Tolbiac, Paris, 75013, France*

Abstract

The goal of anomaly detection is to identify observations that are generated by a distribution that differs from the reference distribution that qualifies normal behavior. When examining a time series, the reference distribution may evolve over time. The anomaly detector must therefore be able to adapt to such changes. In the online context, it is particularly difficult to adapt to abrupt and unpredictable changes. Our solution to this problem is based on the detection of breakpoints in order to adapt in real time to the new reference behavior of the series and to increase the accuracy of the anomaly detection. This solution also provides a control of the False Discovery Rate by extending methods developed for stationary series.

Keywords: Anomaly detection, Time series, Breakpoint detection, FDR

1. Introduction

Anomaly detection has historically, and often still today, been achieved by manually defining fixed thresholds for each monitored parameter. The limitation of this method is that the notion of anomaly is relative to the context: as an example, on a monitored card transaction flow of a bank, it is not expected to have the same number of transactions at night as during the day. The thresholds must therefore be adapted and redefined for each metric, on a regular basis. This task is time consuming and requires expert knowledge. The use of Machine Learning in anomaly detection aims at automating this step [1, 2, 3, 4]. Unsupervised models [5] learn reference behavior from historical data, then they are used to compare reference behavior from seen data and raise alarm when the difference is high. The diversity of anomaly detectors [1, 6] is useful to adapt to different patterns in time series data that can be learned such as trend, seasonality and autocorrelation. Anomalies are detected by comparing the observed metric to its expectation derived from the reference distribution. Using this method, the anomaly detector is able to adapt to changes reference distribution experienced in the past. There are many methods

*Corresponding author

Email addresses: etienne.kronert@worldline.com (Etienne Krönert), dalila.hattab@worldline.com (Dalila Hattab), alain.celisse@univ-paris1.fr (Alain Celisse)

in the literature, from the simple EWMA [7, 8] or ARMA to the more complex LSTM [9, 10], which perform learning of the reference distribution from historical data. But these strategies are unable to adapt to unexpected and unpredictable new normal changes. Such a change is called a breakpoint in the literature, it is detected using breakpoint detectors [11, 12].

Therefore, this article proposes a new anomaly detector based on breakpoint detection to adapt to the new reference distribution. The article [13] gives a good review of existing methods for breakpoint detection. The Kernel Changepoint (KCP) introduced in [14] is used as a breakpoint detector. Thanks to the use of reproducing kernels and the possibility to estimate the number of breakpoints as described in [15], any kind of change, such as in the mean or in the variance, can be detected.

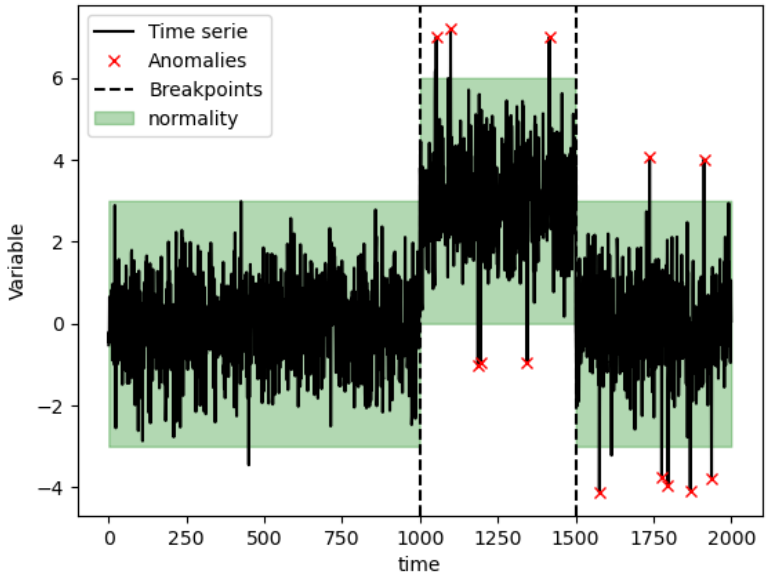


Figure 1: Anomaly detection based on breakpoints.

As illustrated in Figure 1, the novelty of the anomaly detector presented in this paper is to partition the time series into homogeneous segments in order to identify anomalies within each segment. Unlike the proposals cited in [16, 17], the breakpoints do not correspond to anomalies, but to changes in the reference distribution.

However, if poorly calibrated an anomaly detector can lead to alarm fatigue. An overwhelming number of alarms desensitizes the people tasked responding to them, leading to missed or ignored alarms or delayed responses [18, 19]. One of the reasons for alarm fatigue is the high number of false positives which take time to be managed [20, 21]. To reduce the number of false positives, the aim is to control the False Discovery Rate (FDR) [22], while minimizing the False Negative Rate (FNR). The classical method to control the FDR is the Benjamini-Hochberg (BH) procedure [22, 23], which also guarantees a low false negative rate. In the article [24, 25], the FDR is controlled in the case of offline anomaly detection using BH. In a recent work of ours [26], a new strategy has been designed in the online framework for controlling the FDR for stationary series using a modified version of BH applied to subseries. In the present paper, we extend this work to the nonstationary case.

The main contributions of this paper are summarized as follows:

- a versatile online anomaly detector based on breakpoint detection is built to adapt to changes in the reference behavior of the time series. Each component of the detector is studied in depth to provide the best possible parameters and improve the performance of the anomaly detector.
- the notions of active set and calibration set are introduced to deal with the difficulties of the online nature of the anomaly detector.
- the anomaly detector is empirically evaluated in numerous scenarios to determine its capabilities and limitations.

In Section 2, the anomaly detector is introduced and some challenges are raised related to non-stationarity and uncertainty of the estimation of the breakpoint positions in the online context. In Section 3, the breakpoint detector is described. While detecting breakpoints, a good scoring function is necessary to filter the anomalies. This question is discussed widely and illustrated with experiments in Section 4. In addition, the online nature of anomaly detection makes the decision of an abnormal state much more difficult. Solutions are elaborated in Section 5 on how to tackle the uncertainty of abnormal status. Thanks to results published in [26] on how to better control the FDR, Section 6 integrates these results to have an optimal p-value and threshold selection used in the anomaly detector. Finally, multiple experiments and numerical results are elaborated in Section 8.

2. Anomaly detection based on breakpoint detection

This section introduces the problem of anomaly detection in time series containing breakpoints, it explains why it differs from the i.i.d anomaly detection problem and why it cannot be solved with an anomaly detector that does not consider the breakpoints. Finally, a high level view of the proposed main ideas is given.

2.1. Modeling of the problem

Let $(\mathcal{X}, \Omega, \mathbb{P})$ be a probability space and assume a realization of the independent random variables $(X_t)_{t\geq 1}$, with X_t taking values in $\mathcal{X}$ for all t, is observed at equal time steps. $T \in \mathbb{N} \cup \{\infty\}$ is the length of the time series. Normality is a concept that is dependent on a context that changes over time. The instants at which the reference distribution changes are called breakpoints. Supposing there are D breakpoints where $D \in \mathbb{N} \cup \infty$, the position of the breakpoints is noted $(\tau_i)_1^D \in [1, T]^D$. To model these different reference behaviors, several reference probability distributions are introduced and noted $\mathcal{P}_{0,i}$. For each segment i in $[\![1, D]\!]$, for each point t in this segment $[\![\tau_i, \tau_{i+1} - 1]\!]$, the observation X_t is called "normal" if $X_t \sim \mathcal{P}_{0,i}$. Otherwise X_t is an "anomaly". Between two consecutive breakpoints, all "normal" observations are generated by the same law defining a homogeneous segment. The time series (X_t) is piecewise stationary.

As illustrated in Figure 2, an observation x_t is an anomaly if it is not generated from the reference distribution corresponding to the current segment. Figure 2 shows two anomalies detected in the second segment between breakpoints τ_2 and τ_3. Four anomalies have been detected in the last segment 3.

The aim of an online anomaly detector is to find all anomalies among the new observations along the time series $(X_t)_{t\geq 1}$: for each instant $t > 1$, a decision is taken about the status of X_t based on past observations: $(X_s)_{1\leq s\leq t}$. Let $\mathcal{H}^0$ be the set of all normal data of the time series,

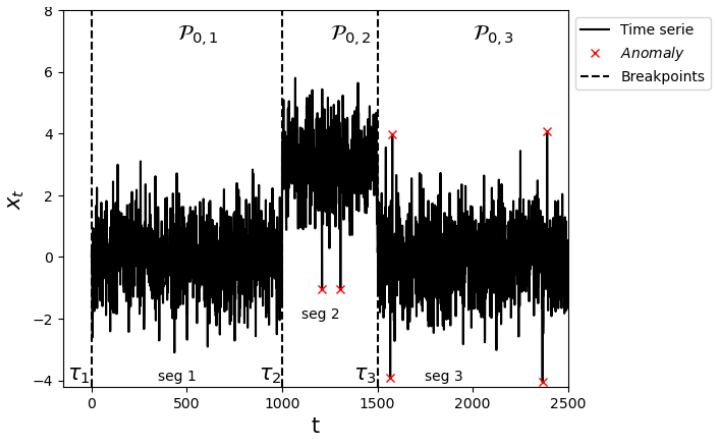


Figure 2: Illustration of piecewise stationary time series.

$\mathcal{H}^1$ be the set of all abnormal data and $\mathcal{R}$ be the set of all data points detected by the anomaly detector. The FDR (resp. FNR) introduced in Section 1 can be expressed as the expectation of the False Discovery Proportion (FDP) (resp. False Negative Proportion (FNP)):

$$FDR_1^T = \mathbb{E}[FDP_1^T] = \mathbb{E}\left[\frac{|\mathcal{H}^0 \cap \mathcal{R}|}{|\mathcal{R}| \vee 1}\right]$$
$$FNR_1^T = \mathbb{E}[FNP_1^T] = \mathbb{E}\left[\frac{|\mathcal{H}^1 \cap \mathcal{R}|}{|\mathcal{H}^1| \vee 1}\right]$$

The control of the FDR at a targeted level α can be expressed by $FDR \leq \alpha$. In the following, the construction of an anomaly detector that controls the FDR at a desired level while minimizing the FNR is studied, in the case of piecewise stationary time series.

2.2. Online anomaly detection in piecewise stationary time series

The goal of this section is to develop a suitable anomaly detector for the nonstationary series described in Section 2.1. First, a generic anomaly detector tailored to the stationary case is described. Then, it is modified to be adapted to the presence of breakpoints in the time series.

Starting point: anomaly detection in stationary time series. Usually, to retrieve anomalies, a unique probability distribution $\mathcal{P}_0$ is considered as the reference distribution assuming no breakpoint in the time series data. Anomalies are defined by observations not generated under the reference distribution: $X_t \nsim \mathcal{P}_0$. In the paper [26], the following general online anomaly detector description was suggested. It uses multiple testing ideas from [24] and the online context from [27]. This online detector relies on the following notions:

- An atypicity score a to compare the observation X_t from a *training set* $\mathcal{X}^{train} = \{X_1, ..., X_q\}$ generated by $\mathcal{P}_0$. The more X_t deviates from the points in the training set, the more the abnormality score $s_t = a(\mathcal{X}^{train}, X_t)$ is high.
- A p-value estimator $\hat{p}$, based on a *calibration set* of scores $\mathcal{S}^{cal} = \{s_{t-m-n}, ..., s_{t-m}\}$ containing scores of data points generated from $\mathcal{P}_0$, to estimate the p-value $\hat{p}_t = p\text{-value}(s_t, \mathcal{S}^{cal})$. In the online context, the calibration set can change over time.

- The value of the threshold ε can be chosen either as a fixed value for all p-values or to be data driven for subseries of p-values, called the test set. Data driven threshold allows better control of the number of false positives through the False Discovery Rate (FDR). $\hat{\varepsilon}_t = \hat{\varepsilon}(\{p_{t-m+1}, \ldots, p_t\})$

Usually, the training set and calibration set are either chosen from the start of the time series labeled with anomalies or evolve over time using sliding windows. When the training set cannot be labeled, a robust atypicity score is required. An example of a training set, calibration set and test set, in the context of online anomaly detection is shown in the following:

$$\underbrace{X_1, \ldots, X_q}_{\text{Training set}}, \ldots \underbrace{X_{t-n-m}, \ldots, X_{t-n}}_{\text{Calibration set}}, \underbrace{X_{t-m}, \ldots, X_t}_{\text{Test set}}$$

Algorithm 1 Generic Online Anomaly Detector for stationary time series

Require: $T > 0$, $(X_t)_1^T$ time series, a an abnormality score, ε a threshold
  for $1 \leq t \leq T$ **do**
    $S_t \leftarrow a(X^{train}, X_t)$
    $p_t \leftarrow p\text{-value}(S_t, S^{cal})$
    if $p_t < \hat{\varepsilon}(\hat{p}_{t-m}, \ldots, \hat{p}_t)$ **then**
      $d_t = 1$
    else
      $d_t = 0$
    end if
  end for
  Output: $(d_t)_1^T$ detected anomalies boolean

As described in Algorithm 1, for each new observation X_t, the function a is used to get the atypicity score. The value of the score cannot be interpreted directly because the distribution of the scores under $\mathcal{H}_0$ is unknown. So its p-value is estimated using the calibration set. The more the data point is atypical, the closer the p-value is to 0. The final step of Algorithm 1 is to compare the p-value with a threshold. The observation is considered abnormal if the p-values is less than the threshold.

The next section discusses the reason why this anomaly detector cannot be applied in case of time series containing breakpoints. Indeed, the definitions of training and calibration used in Algorithm 1 have to be reconsidered.

What are training, calibration and test sets in the case piecewise stationarity?
. Suppose the strategy used for stationary data is applied to a time series where a shift in the mean of the reference distribution occurs. Before the first shift, there are no differences with the stationary case. After the shift, all data points appear as anomalies when using the scoring function trained on the initial training set based on data before the shift. To adapt to the shift, the training and the calibration sets have to be rebuilt on the new segment of data in order to reapply the anomaly detector.

$$\underbrace{X_1, \ldots, X_{\tau_1}}_{\text{Segment 1}}, \underbrace{\underbrace{X_{\tau_1+1}, \ldots, X_{\tau_1+q}}_{\text{train}}, \underbrace{X_{\tau_1+q+1}, \ldots, X_{\tau_1+q+n}}_{\text{calibration}}, \underbrace{X_{\tau_1+q+n+1}, \ldots, X_t}_{\text{test}}}_{\text{Segment 2}}$$

However, it would take a lot of time to gather enough data for the training and calibration sets. This is the reason why two improvements are suggested. The first improvement in the case where the score is stationary across different segments, data for the calibration set can be taken from previous segments. For example, suppose the shift occurs in the mean and the score is the z-score: $(x - \mu)/\sigma$.

$$\underbrace{\underbrace{X_1, \ldots, X_q}_{\text{train}}, \underbrace{X_{q+1}, \ldots, X_{q+m}}_{\text{calibration}}}_{\text{Segment 1}}, \ldots, \underbrace{\underbrace{x_{\tau_1+1}, \ldots, X_{\tau_1+q}}_{\text{train}}, \underbrace{X_{\tau_1+q+1}, \ldots, X_t}_{\text{test}}}_{\text{Segment 2}}$$

If the scoring function is robust to the presence of anomalies inside the training set, the training can have anomalies. The whole segment can be used as training set. The test set can be part of the training set, using a leave-one-out strategy. The segment length required for anomaly detection can thus be further reduced, this constitute the second improvement.

$$\underbrace{\underbrace{X_1, ..., X_n, \ldots}_{\text{calibration}}}_{\text{Segment 1 and train}} \quad , \quad \underbrace{\ldots, \underbrace{X_{t-m} ..... X_t}_{\text{test}}}_{\text{Segment 2 and train}}$$

2.3. The uncertainty of estimations

The setup of the training and calibration sets described in the previous section relies on the knowledge of the breakpoint positions. In practice, neither the number of segments D, nor the positions of the breakpoints τ_i nor the laws of the segments $\mathcal{P}_{0,i}$ are known. All these quantities must be learned using the breakpoint detector and the scoring function to perform anomaly detection.

Moreover, in an online context, the lack of knowledge of the whole series influences a good estimation of these quantities and has a negative impact on the quality of the detection. With each new observation, different situations may occur: the position of a previous breakpoint may be adjusted or removed, or a new breakpoint may appear. These new observations influence the composition of each segment and therefore modify the score value and status assigned to each point. Consequently, the values of quantities associated with a data point X_u change over the time t. To reflect this evolution, a subscript t is added. For example, $\hat{p}_{u,t}$ is the p-value estimated for X_u at time t. Similarly $d_{u,t}$ is the status of the point X_u at time t. The concept of the active set is introduced to collect the last points observed in an Online context and whose "abnormal" or "normal" status is uncertain since it may evolve due to the introduction of new data points.

2.4. High level description for Breakpoint Based Anomaly Detection

Using the different concepts introduced in Sections 2.1, 2.2 and 2.3, the anomaly detection algorithm based on breakpoint detection is proposed in Algorithm 2, using the following notions.

1. **breakpointDetection**: is a breakpoint detector that estimates the number, $\hat{D}_t$, and the positions, noted $\hat{\tau}(t)_1, \ldots, \hat{\tau}(t)_{\hat{D}_t}$, of breakpoints in the current time series X_1^t.
2. **activSelection**: returns the active set. This is the set of observations whose normal versus abnormal status needs to be reevaluated because it is too uncertain.

3. **calibrationSelection**: constructs a calibration set representative of the current reference distribution. It can extract a subsequence of data from the current segment, or from previous segments that have a statistical law distribution very similar to the current segment.

4. **atypicityScore**: A score $a : \mathcal{X} \to \mathbb{R}$ is a function reflecting the atypicity of an observation X_t. The atypicity score is defined as a non conformity measure, note $\overline{a}$, to the segment.

$$s_u = a(X_u) = \overline{a}(Seg(u), X_u)$$

where $Seg(t)$ is the unique homogeneous segment that contains X_t.

5. **pvalueEstimator**: Estimates the probability of observing a normal data point with an atypicity score $a(X)$ greater than $a(X_t)$. This is done using a p-value estimator and the calibration set.

$$p\text{-value}(s_t, \mathcal{S}_t^{cal}) = \frac{1}{|\mathcal{S}_t^{cal}|} \sum_{s \in \mathcal{S}_t^{cal}} \mathbb{1}[s > s_t]$$

6. **thresholdChoice**: A multiple testing procedure is applied to determine a detection threshold on the active set, which plays the role of the test set.

Algorithm 2 Breakpoints based anomaly detection

Require: Let $T > 0$ be the time series length, $(X_t)_1^T$ be the time series, *breakpointDetection* implements breakpoint detector, *activSelection* retrieves the active set, *activSelection* retrieves the calibration set, *pvalueEstimator* implements the p-value estimator and *thresholdChoice* selects the best threshold to be applied.

for $t = 1$ to T **do**
  $\hat{\tau}(t) \leftarrow$ breakpointDetection(X_1^t)
  $\mathcal{I}^{active} \leftarrow$ activSelection$(X_1^t, \hat{\tau}(t))$
  $\mathcal{I}^{cal} \leftarrow$ calibrationSelection$(X_1^t, \hat{\tau}(t))$
  $\mathcal{S}^{cal} \leftarrow \{\overline{a}(Seg(u), X_u), u \in \mathcal{I}^{cal}\}$
  for u in $\mathcal{I}^{active}$ **do**
    $s_{u,t} \leftarrow \overline{a}(Seg(u), X_u)$
  end for
  for u in $\mathcal{I}^{active}$ **do**
    $\hat{p}_{u,t} =$ pvalueEstimator$(s_u, \mathcal{S}^{cal})$
  end for
  $\hat{\varepsilon}_t =$ thresholdChoice$(\{\hat{p}_{u,t}, u \in \mathcal{I}^{active}\})$
  for u in $\mathcal{I}^{active}$ **do**
    if $\hat{p}_{u,t} < \hat{\varepsilon}_t$ **then**
      $d_{u,t} = 1$
    else
      $d_{u,t} = 0$
    end if
  end for
end for
Output: $(d_{t,T})_{t=1}^T$ boolean list that represent the detected anomalies.

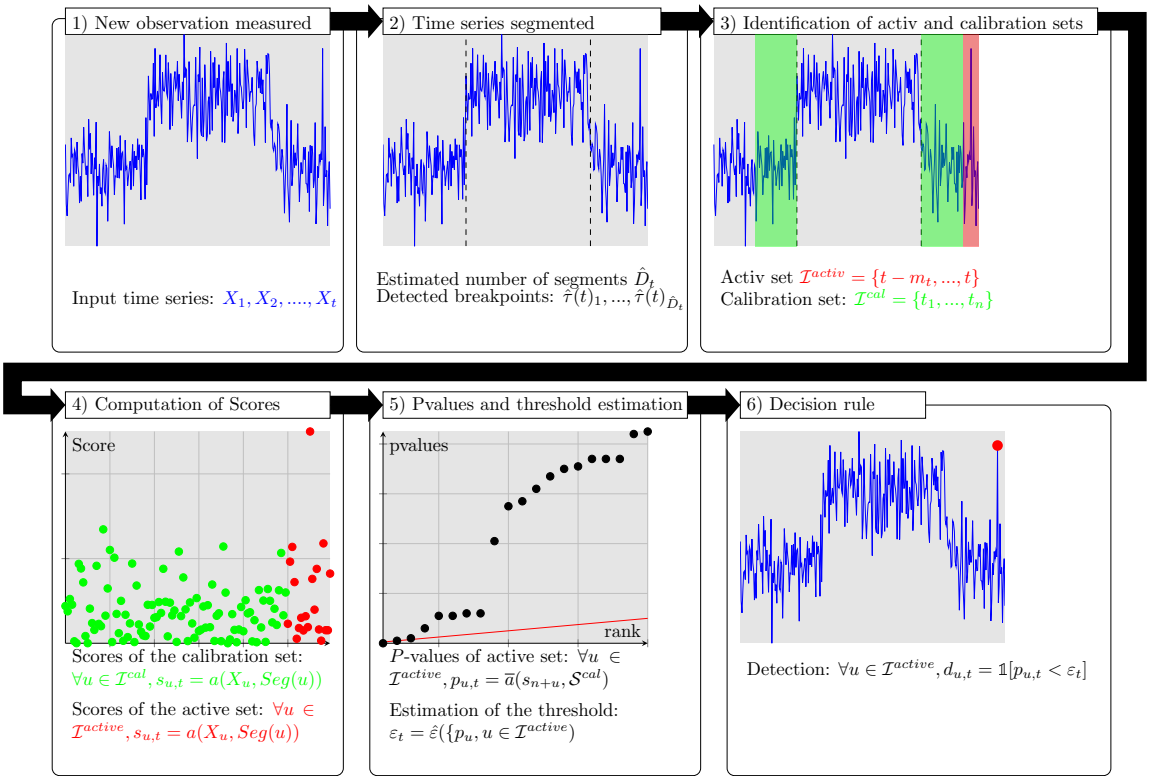


Figure 3: Description flow of Algorithm 2.

The description of the flow illustrated in Figure 3 is given as the following:

Step 1 : a new observation point is given.

Step 2 : the segmentation of the time series is updated. Each segment is homogeneous.

Step 3 : the points coming from the Online streaming data with a status that are unsure are identified. An active set and the calibration set are built.

Step 4 : The calibration set and active set data points are scored.

Step 5 : The p-values of the active set are estimated using the calibration set. The multiple testing procedure is applied to the active set, to get the data driven threshold.

Step 6 : A decision is applied to give the abnormal status to the data point above the threshold

The modularity of our method allows a better adaptation to the diversity of time series. In the following Sections 3, 4, 5, 6 and 7 the different components of the algorithm are discussed and described in more detail.

3. How are breakpoints estimated?

As described at Section 2.1 the time series $(X_t)_{1 \leq t \leq T}$ has D breakpoints denoted $\tau_1, ..., \tau_{D+1}$. These breakpoints are ordered such that $i < j$ implies $\tau_i < \tau_j$. The segment formed by the data between two consecutive breakpoints are iid. The segment $X_{\tau_i}^{\tau_{i+1}-1}$ is said homogeneous.

Informative features for anomaly detection, such as the mean or the variance, can be extracted if the breakpoints are correctly identified. A good breakpoint detector is important to increase the accuracy of anomaly detection. If a shift is not well detected, the analyzed segment will be heterogeneous and the estimation of the law under $\mathcal{H}_0$ will be biased. If too many breakpoints are detected in a segment while it is homogeneous, the analyzed segments will contain fewer points and the variance of the predictions will be too high. To maximize the performance of the anomaly detection, the number and the positions of breakpoints have to be accurately estimated. The article [13] is a review of existing offline breakpoint detectors. The authors show that a breakpoint detector can be described as an optimization problem, using three notions.

- **Cost function**. A cost function $c(\cdot)$ measures the homogeneity of a given subseries $X_{t_1}^{t_2}$. With a well chosen cost function, $c(X_{t_1}^{t_2})$ is high when there is at least one breakpoint between t_1 and t_2. The cost function is low when there is no breakpoint in this subseries.
- **Search method**: The search method enables to explore a set of possible segmentations, denoted $\mathcal{T}$, of the optimization problem. Each search method is a trade-off between accuracy and computational complexity [13].
- **Penalty function**: The penalty function is useful when the number of breakpoints is unknown. It avoids overestimating the number of breakpoints by penalizing segmentations with a large number of breakpoints. The penalty function $pen(\cdot)$ increases based on the number of breakpoints, noted D_τ.

The segmentation returned by the breakpoint detector is the one that minimizes the penalized cost function among the explored solutions:

$$\hat{\tau} \in \arg\min_{\tau \in \mathcal{T}} \sum_{i=1}^{D_\tau} c(X_{\tau_i}^{\tau_{i+1}-1}) + pen(D_\tau) \tag{1}$$

In this article, the Kernel Change Point (KCP) introduced in [15] is used for its advantages. The kernel-based cost function could be used for any kind of time series, univariate or multivariate, without changing the breakpoint detector. To handle the diversity of time series data, the kernel and its hyperparameters have to be carefully chosen to be able to detect any kind of change points in the time series. This accuracy is guaranteed by the oracle inequality given in [15]. For a given segmentation τ and a kernel K, the cost is given by:

$$\hat{R}(\tau) = \frac{1}{t}\sum_{u=1}^{t} K(X_u, X_u) - \frac{1}{t}\sum_{i=1}^{D_\tau} \frac{1}{\tau_{i+1}-\tau_i} \sum_{u,v=\tau_i}^{\tau_{i+1}-1} K(X_u, X_v) \tag{2}$$

First, the candidate segmentations that minimize the criterion are identified for each possible number of D segments. $\mathcal{T}^D$ is the space of all candidate segmentations with D segments, $\hat{\tau}_{D,t}$ is the best candidate segmentation with D segments and $L_{D,t}$ is the cost associated with this segmentation.

$$L_{D,t} = \min_{\tau \in \mathcal{T}^D} \hat{R}(\tau)$$
$$\hat{\tau}_{D,t} = \arg\min_{\tau \in \mathcal{T}^D} \hat{R}(\tau)$$

To estimate the number of segments and thus the best segmentation, one searches for the segmentation $\hat{\tau}_{D,t}$ that minimizes the penalized criterion described in Eq. 1. The penalty function

is given by:

$$pen(\tau) = c_1 D_\tau + c_2 \log \binom{t-1}{D_\tau - 1} \tag{3}$$

where the coefficients c_1 and c_2 are estimated by fitting the penalty function on the estimated cost for over-segmented segmentations [28].

KCP is designed as an offline breakpoint estimator. By using Dynamic Programming, the segmentation costs can be estimated without performing the same computation between time t and $t+1$ as described in [14]. This feature is necessary to be applied in an online anomaly detection. The data driven penalty function enables good accuracy in estimating the number of breakpoints. The breakpoints are detected by solving the optimization problem with the algorithm:

Algorithm 3 Dynamic Programming for breakpoint detection.

Require: $T > 0$, (X_t) time series, C Kernel based cost function, D_{max} maximum breakpoint number explored and $SlopeHeuristic$ implement the slope heuristic.
  for $t \in [\![1, T]\!]$ **do**
    for $D \in [\![1, D_{max}]\!]$ **do**
      $L_{D,t} \leftarrow \min_{t' \leq t} L_{D-1,t'} + C_{t',t}$
      $\hat{\tau}_{D,t} \leftarrow \arg\min_{t' \leq t} L_{D-1,t'} + C_{t',t}$
    end for
    $c_1, c_2 \leftarrow SlopeHeuristic(L)$
    $\hat{D} \in \arg\min_D L_{D,t} + c_1 D + c_2 \log \binom{t-1}{D-1}$
    $\hat{\tau}_t \leftarrow \hat{\tau}_{\hat{D},t}$
  end for
  Output: $\forall t \in [\![1, T]\!], (\tau(t))$ estimated segmentation at each time step.

The main degree of freedom in KCP is the choice of the kernel. Characteristic kernels [29], like the Gaussian kernel, are able to detect any kind of change: shift in the mean, shift in the variance, shift in the third moment,...

$$K(x, y) = \exp(-\frac{||x - y||^2}{2h^2}) \tag{4}$$

However, due to the fact that the number of points within a segment is finite, the performance of a characteristic kernel depends on the choice of hyperparameters. In the case of the Gaussian kernel, the only parameter is the bandwidth h. For changes that occur in the mean, the *median heuristic*, shown in Eq. 5, gives good results [30]. Defining a method to select the most relevant kernel for any kind of breakpoint is still an open question.

$$h = \text{median}_{i \neq j}(||X_i - X_j||) \tag{5}$$

Breakpoint detection is used to define homogeneous segments. In the next section, the characterization of atypicity in each segment is studied.

4. How to choose a good abnormality score in the context of breakpoints ?

In this paper, an anomaly is a data point that does not follow the reference distribution of the segment to which it belongs. To construct an atypicity score that is higher for abnormal points,

a point must be compared to the rest of the segment.The Nonconformity Measure (NCM) from [31] is introduced. The Nonconformity Measure $\tilde{a}$, is a real valued function $\tilde{a}(B, z)$ that measure how different z is from the set B. A nonconformity measure can be used to compare a data point with the rest of the segment. If all points within a segment are generated by the reference distribution, then the Nonconformity Measure provides an atypicity score for this segment.

$$\forall i \in [\![1, D]\!], \quad \forall [\![\tau_i, \tau_{i+1}[\![, \quad a(X_t) = \tilde{a}(\{X_{\tau_i}, \ldots, X_{\tau_{i+1}-1}\} \backslash \{X_t\}, X_t) \tag{6}$$

The following properties are required to enable the usage of the nonconformity measure to build a good atypicity score:

- anomalies should have higher atypicity score than normal data points.
- the NCM must be robust [32, 33, 34] to the presence of anomalies. The anomalies present in the segment do not affect the value of the returned measure.
- the values returned between different segments must be comparable, so that a p-value can be estimated, with a calibration set containing values from different segments. The uniformity property of scoring is introduced:

  Definition 1 (Stationarity and piecewise dependency of the score)**.** Let T be the length of the time series and $\tau_1 < ... < \tau_{D+1}$ the breakpoints defining D segments. And let $\mathcal{P}_{0,1} \ldots, \mathcal{P}_{0,D}$ be probability distributions Let $(X_t)_{1 \leq t \leq T}$ be a segmented time series. Let $\tilde{a}$ be a nonconformity measure. $\tilde{a}$ is stationary on $(X_t)_{1 \leq t \leq T}$ if s_t, given in Eq. 6, is stationary:

  $$\forall i \in [\![1, D]\!], \quad \forall [\![\tau_i, \tau_{i+1}[\![, \quad s_t = \tilde{a}(\{X_{\tau_i}, \ldots, X_{\tau_{i+1}-1}\} \backslash \{X_t\}, X_t) \tag{7}$$

  Furthermore, s_t is said piecewise dependent if score values between two different segments are independent.

The property of score stationarity depends on the time series. For example, the z-score with true known mean and standard deviation satisfies the stationarity property only if the changes generated by the breakpoints are shifts in the mean or in the standard deviation. If the change occurs in higher moments, the property is not satisfied. Furthermore, the property is not satisfied for the z-score if the mean and standard deviation need to be estimated. Since the stationarity of the score is difficult to obtain, it is approached with the following strategies:

- Ensure that the segment contains enough points to ensure the convergence of the nonconformity measures. For example, since the mean and variance must be estimated, the z-score is not stationary. However, if these parameters converge to the true mean and standard deviation, then the z-score can be considered stationary once the segments have enough points. The faster convergence is achieved, the easier it is to ensure the stationarity property. An NCM is said to be efficient when convergence is achieved for a low number of points.
- Instead of using the entire segment, the training set can be built by resampling a specified number of points. It can be used on NCM that are highly dependent on the training set cardinality, like kNN.
- Rather than trying to ensure that the law of non-conformity measures is identical in each segment, identify the segments with the most similar laws.

Many NCMs depend on segment parameters to be estimated, e.g. the z-score requires knowledge of the mean and variance. To satisfy the properties of a good atypicity function, the estimators need to satisfy the following requirements:

1. the estimator should be robust to anomalies in the training set: the estimation should not be affected by the presence of anomalies in the training set.
2. The estimator should be efficient [35]. High precision estimation of the parameter should be obtained with a minimal number of samples.

4.1. Experiments

It has been seen that to build a good score function, the estimators used must verify the robustness and efficiency properties. To assess the robustness and the efficiency of the atypicity score, synthetic data are used for experimentation and analysis. The robustness of an estimator is its ability to be unbiased in the presence of anomalies. An estimator is said efficient when it is close to the parameter value with a limited number of data points. In this analysis, three categories of estimators are tested: one "efficient and not robust", a second "not efficient and robust" and a third "robust and efficient". These three estimators are analyzed considering the absence or presence of anomalies. The assessment is based on the parameter estimation error and on the anomaly detection performances using FDR and FNR.

4.1.1. Description

In this experiment, the focus is on the z-score. The atypicity of a data point x is calculated from the mean μ and standard deviation σ as follows $a_z(x, \mu, \sigma) = (x - \mu)/\sigma$. In an anomaly detection context, the mean and standard deviation are unknown and need to be estimated. There are many estimators of the mean and standard deviation. These estimators have different properties in terms of robustness and efficiency. In order to study the relationship between these properties and the performance of the anomaly detector, three estimators are chosen for each of these two values.
For the mean value the three estimators are defined as the following:

- Maximum Likelihood Estimator: $\mu_{mle} = \frac{1}{n}\sum_i x_i$. This estimator is efficient but not robust against anomaly contamination.
- Median: $\mu_r = \text{median}(x_1, ..., x_n)$. This estimator is robust but less efficient than the MLE estimator.
- Biweight location, introduced in [36]. This estimator is robust and efficient.

$$\mu_{bw} = \frac{\sum_{i=1}^{\ell}(1-u_i^2)x_i\mathbb{1}[|u_i|<1]}{\sum_{i=1}^{\ell}(1-u_i^2)}$$
$$u_i = \frac{x_i - \overline{x}}{9MAD}$$

  Where $\overline{x}$ is median of the x_i and MAD is the median absolute deviation.

For the standard deviation, the three estimators are defined as the following:

- Maximum Likelihood Estimator: $\sigma_{mle} = \sqrt{\frac{1}{n}\sum_i (x_i - \mu)^2}$. This estimator is efficient but not robust against anomaly contamination.
- Median: $\sigma_{mad} = \text{median}(||x_i - \mu|)$. This estimator is robust but less efficient than the MLE estimator.

- Biweight Midvariance estimator: introduced in [37]. This estimator is robust and efficient.

$$\sigma_{bw}^2 = \frac{\ell \sum_{i=1}^{\ell}(x_i - \overline{x})^2(1-u_i^2)^4 \mathbb{1}[|u_i|<1]}{(\sum_{i=1}^{\ell}(1-u_i^2)(1-5u_i^2)\mathbb{1}[|u_i|<1])^2}$$
$$u_i = \frac{x_i - \overline{x}}{9MAD}$$

  Where $\overline{x}$ is the median of the x_i and MAD is the median absolute deviation.

All the six estimators are evaluated according to two measures:

1. First, the precision and the robustness of the estimator is evaluated using the Mean Squared Error (MSE), applying the following procedure: Let θ be either the mean or the standard deviation parameter, and $\hat{\theta}$ be an one estimator of the parameter θ. Let ℓ be the cardinality of the segment used to estimate θ. Let B be the number of repetitions for the experiments.
   (a) Generate the segment data: For b in $[1, B]$ and for i in $[1, \ell]$, $X_{b,i} \sim \mathcal{N}(0,1)$, if the segment contains only normal data. For b in $[1, B]$, for i in $[1, \ell_0]$, $X_{b,i} = 4$ and for i in $[\ell_1, \ell]$, $X_{b,i} \sim \mathcal{N}(0,1)$, if the segment is contaminated by anomalies
   (b) Estimate the parameter using the estimator: For b in $[1, B]$, $\hat{\theta}_b = \hat{\theta}(X_{b,1}, ..., X_{b,\ell})$.
   (c) Compute the MSE, $MSE = \frac{1}{k}\sum_{b=1}^{B}(\hat{\theta}_b - \theta)^2$

   Different values of the segment length ℓ are tested, from 10 to 1000. For each value of ℓ, two values of ℓ_1 are tested. One with $\ell_1 = 0$, for the case where there are no anomaly in the training set. The other with $\ell_1 = \lfloor 0.02\ell \rfloor$ for the case of contamination with anomalies. For each set of parameter values, the experiment is repeated $B = 1000$ times.

2. Then, the Anomaly Detection capacity is evaluated using the FDR and FNR criteria. This is done by simulating multiple detections inside a segment applying the following procedure: using n the calibration set cardinality, ℓ the length of the segment, ℓ_1 the number of anomalies in the training set, m the test set cardinality and m_1 the number of anomalies in the test set:
   (a) Generate training segment data with ℓ_1 anomalies.

$$\forall i \in [\![1, \ell_1]\!], \quad X_i \sim \mathcal{N}(4, 0.1), \text{ and } \forall i \in [\![\ell_1, \ell]\!], \quad X_i \sim \mathcal{N}(0,1)$$

   And estimate the segment mean and standard deviation

$$\hat{\mu} = \hat{\mu}(X_1^{\ell}), \quad \hat{\sigma} = \hat{\sigma}(X_1^{\ell})$$

   (b) Generate the calibration set

$$\forall j \in [\![1, n]\!], \quad Y_j \sim \mathcal{N}(0,1)$$

   (c) Generate the test segment data

$$\forall i \in [\![1, m_1]\!], \quad Z_i \sim \mathcal{N}(4, 0.1), \text{ and } \forall i \in [\![m_1, m]\!], \quad Z_i \sim \mathcal{N}(0,1)$$

   (d) Compute the p-values of the test set, using calibration set and affected by the parameter estimations

$$\forall i \in [\![1, m]\!], \quad \hat{p}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}[X_j > (Z_i - \hat{\mu})/\hat{\sigma}]$$

(e) Anomalies are detected using the Benjamini-Hochberg procedure on the p-values. The threshold of the BH procedure is noted $\hat{\varepsilon}_{BH_\alpha}$ as defined in our previous work [26]:

$$\hat{\varepsilon} = \hat{\varepsilon}_{BH_\alpha}(\hat{p}_1, \ldots, \hat{p}_m)$$

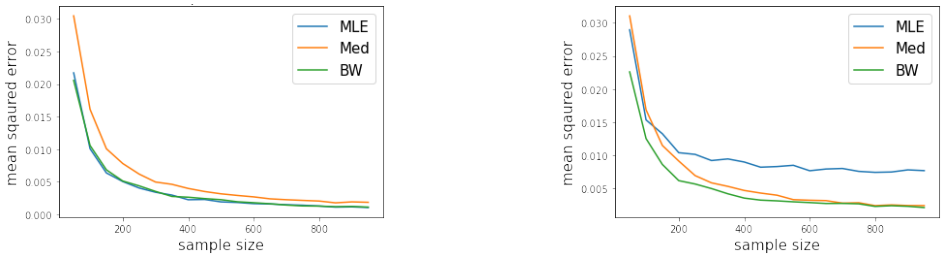
(f) Compute FDP and FNP. Remembering that anomalies are generated in the first m_1 values of the test set:

$$FDP = \frac{\sum_{j=m_1}^{m} \mathbb{1}[\hat{p}_j \leq \hat{\varepsilon}]}{\sum_{j=1}^{m} \mathbb{1}[\hat{p}_j \leq \hat{\varepsilon}]}$$
$$FNP = \frac{\sum_{j=1}^{m_1} \mathbb{1}[\hat{p}_j > \hat{\varepsilon}]}{m_1}$$

Different values of segment length ℓ are tested, from 10 to 500. For each value of ℓ, two values of ℓ_1 are tested. One with $\ell_1 = 0$, for the case where there are no anomaly in the training set. The other with $\ell_1 = \lfloor 0.02\ell \rfloor$ for the case of contamination with anomalies. For each set of parameter values, the experiment is repeated $B = 10^4$ times.

4.1.2. Results

Figures 4 and 5 illustrate the estimators performances of the mean estimators. Figure 4 compares different mean estimators according to the segment length. The MSE decreases rapidly with the sample size for all estimators in Figure 4a. However the MLE and BW estimators have very close and slightly better performances compared to the median estimator. This illustrates the efficiency of the MLE and BW estimators. But in the presence of anomalies, the performance of the MLE estimator is severely degraded as shown in Figure 4b compared to the median and BW estimators showing more robustness in the presence of outliers.

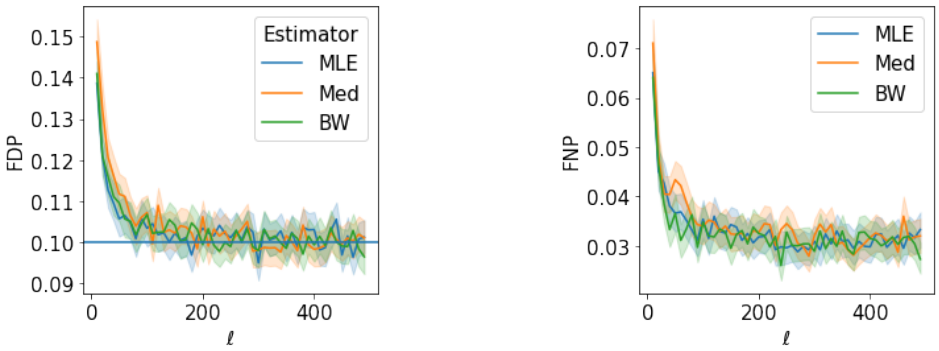


(a) Without anomaly contamination.

(b) With anomaly contamination.

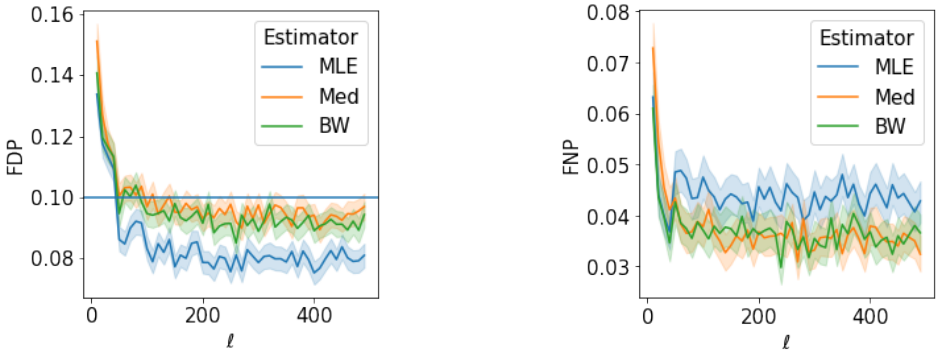
Figure 4: Estimation error of the mean as a function of segment length and mean estimator used.

Figure 5 illustrates the FDR and FNR of the anomaly detector according to the mean estimator used. As shown in Figure 5a and 5b, in the case of non contamination by anomalies, the FDR and FNR results are very close to the target for all the estimators. However, in presence of anomalies, the MLE performance is degraded. In Figure 5c, the FDR is below the targeted level and in Figure 5d, the FNR is higher than other estimators. Either Med or BW can be used to do anomaly detection.



(a) FDR, without anomaly

(b) FNR, without anomaly

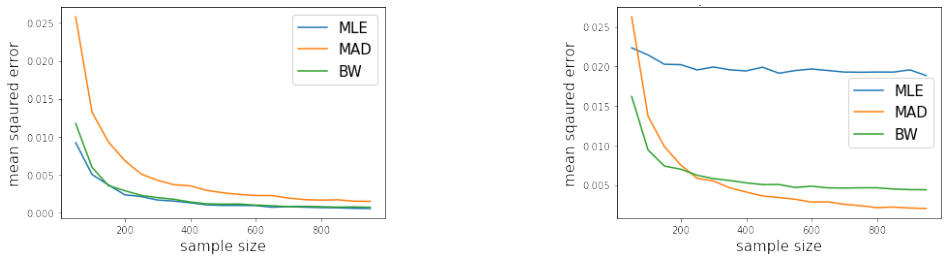


(c) FDR, with anomalies

(d) FNR, with anomalies

Figure 5: Anomaly detector performances as a function of the segment length and the mean estimator used.

Figures 6 and 7 illustrate the performances of the standard deviation estimators. Figure 6 compares the precision using the MSE of the different standard deviation estimators according to the segment length. The MSE decreases rapidly with the sample size for all estimators in Figure 6a. However the MLE and BW estimators have very close and better performances when compared with the MAD estimator. This illustrates the efficiency of the MLE and BW estimators. But in the presence of anomalies, the performance of the MLE estimator is severely degraded as shown in Figure 6b. On the contrary, the MAD and BW estimators are less degraded and show more robustness in presence of outliers.

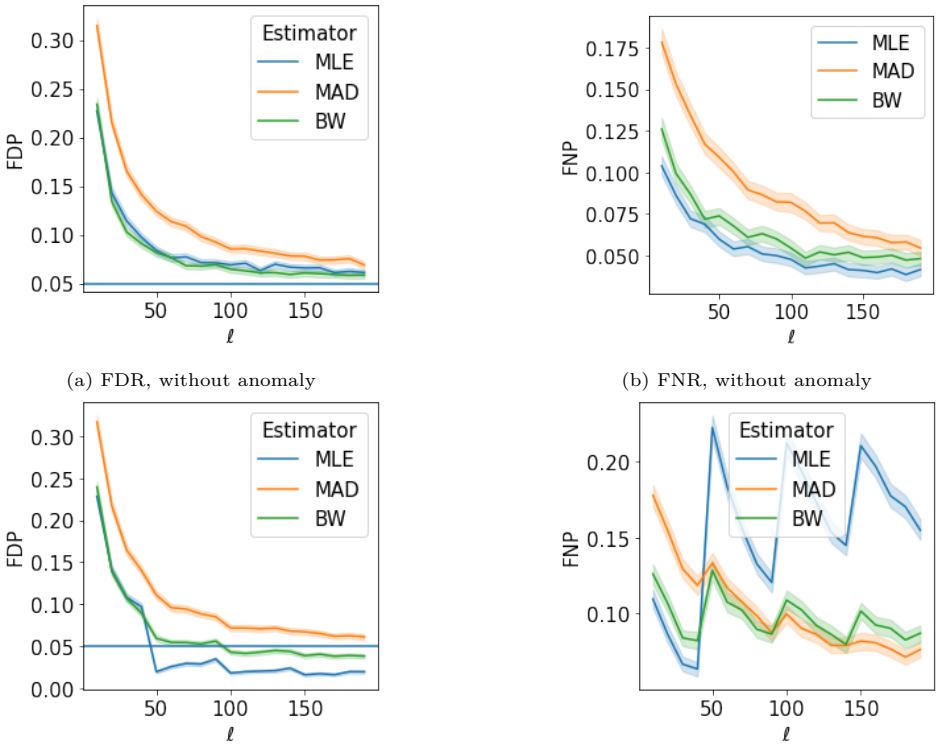


(a) Without anomaly contamination.

(b) With anomaly contamination.

Figure 6: Estimation error of standard deviation as a function of the segment length and the standard deviation estimator used.

Figure 7 shows the performances measured by FDR and FNR once the anomaly detection is applied. As illustrated in Figures 7a and 7b, FDR and FNR for MAD are higher compared to MLE and BW. But in presence of anomalies, the MLE performance is degraded. The FDR is below the targeted level, as shown Figure 7c, and the FNR is higher than other estimators, as shown in Figure 7d. The strange behavior of the MLE curve in Figure 7d with spikes in the FNR is due to the number of anomalies increasing with every 50 data points because $\ell_1 = \lceil 0.02\ell \rceil$. The best estimator for standard deviation in case of anomaly detection is BW.



(a) FDR, without anomaly

(b) FNR, without anomaly

(c) FDR, with anomalies

(d) FNR, with anomalies

Figure 7: Anomaly detector performances as a function of segment length and standard deviation estimator used.

4.1.3. Conclusion

The experiments show the importance of the robustness and efficiency to build a good atypicity score. High MSE implies lower performance in terms of FDR and FNR control. The classical standard deviation estimators, MLE and MAD, are underperforming. For the following sections of this paper, the BW (Biweight midvariance) estimator is used to implement the scoring function.

5. How to manage uncertainty of decisions?

The online nature of anomaly detection adds much more uncertainty. Indeed, breakpoint detection in an Online context induces two sources of error:

- a delay in detecting a breakpoint and other breakpoint detection updates.

- a difficulty in identifying parameters of a small length segment. This difficulty may lead to higher errors in the scoring function estimation.

These errors can be corrected by reevaluating the decision after new data points have been collected. However the status of each point in the whole time series should not reevaluated at each new observation. To ensure that the status of each data point should only be reevaluated when it is useful, two notions are introduced: the confidence score assigned to a data point and the setup of the active set.

5.1. Definition of confidence score, active set and important properties

A confidence score assigned to each data point in the time series reflects the confidence in its abnormal or normal status. If the confidence score is high, the status of the point should not change when acquiring new data points.

Definition 2 (Confidence Score)**.** The confidence score $c_{u,t}$ assigns to the decision made for the data point X_u, at time t, the probability that it remains unchanged until the end of the analysis.

$$c_{u,t} = \mathbb{P}\left[\forall t' > t, \quad d_{u,t} = d_{u,t'}\right]$$

The active set is the set of points belonging to the current segment whose "abnormal" or "normal" status has to be reevaluate at time t because the confidence in the decision made at time $t-1$ is too low. The active set, is the points from the current segment having a confidence score lower than a threshold $1-\eta$. Figure 8 illustrates in red the active set capturing the incoming streamed data points. With each new data point, the multiple test procedure is applied to the active set, in order to reevaluate the status of its points. The active set is also the set set defined in Section 2.3.

Definition 3 (Active Set)**.** Let $\eta > 0$ be the confidence threshold. The active set is the set of points with a confidence score lower than $1-\eta$.

$$\mathcal{A}_{\eta,t} = \{u, c_{u,t} < 1-\eta\} \tag{8}$$

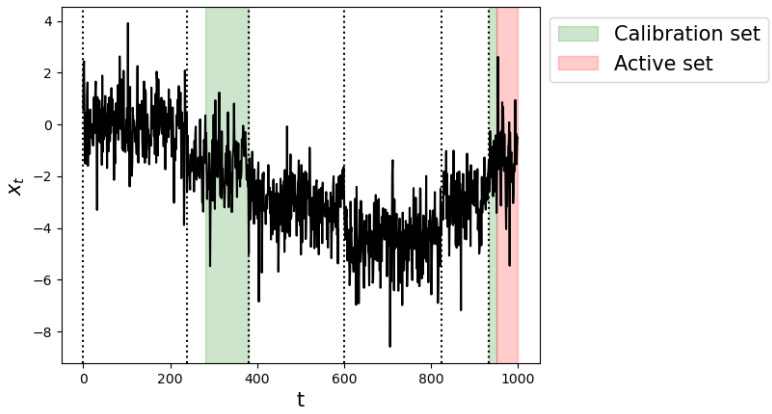


Figure 8: Illustration of calibration set and active set.

To be able to estimate the confidence score of a given point and to build the active set, the reasons that cause the decision change are investigated. The process for determining the status of a given data point described in greater detail.

After applying the breakpoint detector to the whole time series $X_1, \ldots, X_t$ at time t, the set of estimated breakpoints is given by $\hat{\tau}_t = (\hat{\tau}_{t,1}, \ldots, \hat{\tau}_{t,\hat{D}-1}, \hat{\tau}_{t,\hat{D}+1} = t)$. The last breakpoint $\hat{\tau}_{t,\hat{D}}$ is noted $\hat{b}_t$. The segment of data $X_{\hat{b}_t}, \ldots, X_t$ is called the current segment, and the length of the current segment is noted ℓ_t, which is calculated as $\ell_t = t - \hat{b}_t$. For each data point position u in the current segment, the distance from a data point position u to the last observed data point is noted $\lambda_{u,t}$, which is calculated as $\lambda_{u,t} = t - u$. X_u is the data point at position u, it has the status $d_{u,t-1}$ at time $t-1$. This status could change or remain the same at time t. Let us on elaborate the cases where a status could change:

- if at time t, a new breakpoint is detected between the position u and the last breakpoint at time $t-1$, the point X_u is assigned to a new segment. The detection of this new breakpoint has an influence on the training and calibration sets, which needs to be adjusted.
- Otherwise, if no new breakpoint is detected, an additional point is added to the current segment. If the cardinality of the current segment is too small, this new point may affect the atypicity score such that the status changes.

To formalize this analysis and compute the confidence score, two events are introduced:

- "The status of data point X_u at step t changes over time"

$$\mathrm{CSt}_{u,t} = \{\exists t' > t, d_{u,t} \neq d_{u,t}\}$$

- "The segment to which the data point X_u is assigned at time t changes over time"

$$\mathrm{CASg}_{u,t} = \left\{\exists t' > t, \tau(t') \cap [\![\hat{b}_t, u]\!] \neq \emptyset\right\} \tag{9}$$

Then, it is possible to express the confidence score according to the probabilities of the two events.

$$\begin{aligned} 1 - c_{u,t} = \mathbb{P}\left[\mathrm{CSt}_{u,t}\right] &= \mathbb{P}\left[\mathrm{CSt}_{u,t} | \mathrm{CASg}_{u,t}\right] \mathbb{P}\left[\mathrm{CASg}_{u,t}\right] + \mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}\right] \mathbb{P}\left[\overline{\mathrm{CASg}_{u,t}}\right] \\ &\leq \mathbb{P}\left[\mathrm{CASg}_{u,t}\right] + \mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}\right] \end{aligned}$$

If it can be ensured that $\mathbb{P}\left[\mathrm{CASg}_{u,t}\right] \leq \eta/2$ and $\mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}\right] \leq \eta/2$ then it gives $c_{u,t} > 1 - \eta$. By Definition 3, this point does not belong to the active set. Otherwise, if $\mathbb{P}\left[\mathrm{CASg}_{u,t}\right] > \eta$ or $\mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}\right] > \eta$ then X_u belongs to the active set.

In order to define the probabilities $\mathbb{P}\left[\mathrm{CASg}_{u,t}\right]$ or $\mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}\right]$ and to build the active set some assumptions are made:

Proposition 1 (Stationarity)**.** Let $\eta > 0$.

- Assuming $f_\tau : \lambda \mapsto \mathbb{P}\left[\mathrm{CASg}_{t-\lambda,t}\right]$ is decreasing to 0 and does not depend on t.

  Then, there exists λ_η such that:

$$\forall t \in [\![1, T]\!], \forall u \in [\![1, t]\!], \quad |u - t| \geq \lambda_\eta, \quad \mathbb{P}\left[\mathrm{CASg}_{u,t}\right] \leq \eta/2. \tag{10}$$

  The smallest value respecting this property is noted $\lambda_\eta^\star$.

- Assuming $f_d : \ell \mapsto \mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}, \ell_t = \ell\right]$ is decreasing to 0 and does not depend on t. Then, there exist a segment length ℓ_η such that:

$$\forall t \in [\![1, T]\!], \forall u \in [\![1, t]\!], \quad \ell \geq \ell_\eta, \quad \mathbb{P}\left[\mathrm{CSt}_{u,t} | \overline{\mathrm{CASg}_{u,t}}, \ell_t = \ell\right] \leq \eta/2. \tag{11}$$

  The smallest value of ℓ_η is noted $\ell_\eta^\star$.

The conclusions of Proposition 1 follow directly from the definition of convergence to 0.

Proof of Proposition 1. By assumption the probability $\mathbb{P}\left[\text{CASg}_{t-\lambda,t}\right]$ does not depend on t and is noted $f_\tau(\lambda)$. According to the second assumption, $f_\tau(\lambda)$ converges to 0 when λ tends to $+\infty$. Therefore, by definition of convergence:

$$\forall \eta > 0, \exists \lambda_\eta > 0 \quad \lambda \geq, \forall \lambda_\eta, \quad f_\tau(\lambda) \leq \eta/2.$$

Moreover, by definition $\lambda = t - u$, it follows that:

$$\forall \eta > 0, \quad \exists \lambda > 0, \quad \forall t \in [\![1, T]\!], \forall u \in [\![1, t]\!], \quad |u - t| \geq \lambda_\eta, \quad \mathbb{P}\left[\text{CASg}_{u,t}\right] \leq \eta/2.$$

The second result is proven using similar arguments. □

The function $f_\tau : \lambda \mapsto \mathbb{P}\left[\text{CASg}_{t-\lambda,t}\right]$ gives the probability that the segment assigned to $X_{t-\lambda}$ changes as a function of the distance λ from the last observation. It is assumed to be decreasing because the probability of missing a breakpoint decreases with the number of points. The function $f_d : \ell \mapsto \mathbb{P}\left[\text{CSt}_{u,t} | \overline{\text{CASg}_{u,t}}, \ell_t = \ell\right]$ gives the probability of changing the status of a point conditional on the assigned segment remaining unchanged, as a function of the length ℓ of the segment. It is assumed to decrease with the number of points inside the segment.

Assuming that the probabilities $\mathbb{P}\left[\text{CASg}_{t-\lambda,t}\right]$ and $\mathbb{P}\left[\text{CSt}_{u,t} | \overline{\text{CASg}_{u,t}}, \ell_t = \ell\right]$ do not depend on t, it is possible to use the same model for the entire series. Thus, there is no need to recalculate these probabilities for each observation time. Since the probability of detecting a breakpoint depends on the position of the actual breakpoint, this assumption can only be verified by assuming that the position of the breakpoints is determined by a stationary process. Furthermore, the probability of detecting a breakpoint depends on the of the shift that occurs in the time series. Therefore, another necessary condition is to assume that at each breakpoint the segment law changes according to transition rules that are the same throughout the series.

5.2. How to build the active set?

As mentioned earlier, if the cardinality of the active set is too small, this can lead to an uncertain decision. A proposal on how to build an active set is given in the following. Let η be a confidence threshold. According to Proposition 1, there are $\lambda_\eta^\star$ such that $f_\tau(\lambda_\eta^\star) < \eta$ and $\ell_\eta^\star$ such that $f_d(\ell_\eta^\star) < \eta$. To compute $\lambda_\eta^\star$, f_τ is estimated on historical data, this estimation is noted $\hat{f}_\tau$. To compute $\ell_\eta^\star$, f_d is estimated on historical data, this estimation is noted $\hat{f}_d$.

$$\hat{\ell}_\eta = \arg\min \left\{ \ell, \hat{f}_\tau(\ell) < \eta \right\}, \qquad \hat{\lambda}_\eta = \arg\min \left\{ \lambda, \hat{f}_d(\lambda) < \eta \right\}$$

As shown in Algorithm 4, the procedure starts by comparing the length ℓ_t of the current segment with the threshold length $\hat{\ell}_\eta$. If the length ℓ_t is lower than this threshold, the whole segment is considered as the active set since the segment does not contain enough points to estimate the atypicity score with high precision. Otherwise, the segment contains enough points and the source of the status change is segment reassignment. Considering the data points whose distance to the end of the time series is less than $\hat{\lambda}_\eta$, the risk of being reassigned to another segment is high. Consequently, the active set will contain all points that are after the position $t - \hat{\lambda}_\eta$. In the case $\hat{\lambda}_\eta$ is larger than the length of the current segment, the calibration set will include the current segment. Given m_t the active set cardinality, the active set is equal to:

$$\mathcal{I}^{active} = \{t - m_t + 1, \ldots, t\}$$

Algorithm 4 Computation of active set cardinality.

if $\ell_t < \hat{\ell}_\eta$ **then**
    $m_t \leftarrow \ell_t$
else
    $m_t \leftarrow min(\hat{\lambda}_\eta, \ell_t)$
end if
return m_t

The estimation of the probability $\hat{f}_\tau(\lambda)$ is described in Section 5.3. The estimation of the probability $\hat{f}_d(\ell)$ is described in Section 5.4.

5.3. How estimate the probability of segment assignment change?

As introduced in Section 5.1, $f_\tau(\lambda)$ is the probability that the segment assignment changes when a data point is at distance λ from the end of the time series. This probability $f_\tau(\lambda)$ is needed to build the active set containing data points with uncertain status, as described in Section 5.2. In the following, a procedure is proposed to estimate $f_\tau(\lambda)$.

5.3.1. Description of the method

As a reminder, the existence of $f_\tau(\lambda)$ is ensured by the stationarity assumption described in Proposition 1. However, stationarity is not sufficient to calculate these probabilities directly from historical data in the same time series and thus to estimate $\hat{f}_\tau$. It must also be assumed that the series $\mathbb{1}\left[\text{CASg}_{t-\lambda,t}\right]$ is ergodic.

Proposition 2 (Ergodicity)**.** Assume $\mathbb{1}\left[\text{CASg}_{t-\lambda,t}\right]$ is stationary and ergodic. Then

$$\mathbb{P}[\text{CASg}_{t-\lambda,t}] = \lim_{T\to\infty} \frac{1}{T} \sum_{\tilde{t}=1}^{T} \mathbb{1}[\text{CASg}_{\tilde{t}-\lambda,\tilde{t}}] \tag{12}$$

The conclusion of the Proposition 2 follows directly from the definition of ergodic process [38]. A sufficient condition to verify the ergodicity is the weak dependence or mixing [39]. There are several ways to characterize this property. The general idea is that the dependence between two points $\mathbb{1}[\text{CASg}_{t_1-\lambda,t_1}]$ and $\mathbb{1}[\text{CASg}_{t_2-\lambda,t_2}]$ must go to 0 as $t_1 - t_2$ goes to infinity. This property is assumed to be verified when these criteria are satisfied:

- the positions of the breakpoints are iid,
- the transitions between two segment distributions are iid.
- the breakpoint detector is based on KCP,

The breakpoint position iid property assumes the existence of random variables that generate the positions of these breakpoints. These random variables are assumed to be iid and uniform over $[\![1, T]\!]$. The iid property for the transition of the reference distribution implies the existence of random variables that generate the reference law of segment i from that of segment $i-1$. These random variables must be iid to guarantee uniformity in the difficulty of detecting breakpoints. As an example the time series generated in Section 8.1 satisfy these criteria: the segment length follows an exponential law, the positions of breakpoints follow a stationary Poisson process. The transition law is described as a homogeneous Markov chain on the parameters of the reference distribution.

The time series is split into two parts: historical and recent data. The historical data set is built using the first $\tilde{T}$ data points of the time series. The estimation of f_τ is based on the previous segment assignment changes that occurred while detecting breakpoints on historical data. To estimate this probability, the list of all previous segmentations $\mathcal{D} = (\hat{\tau}_1, \dots, \hat{\tau}_{\tilde{T}})$ is used. Assuming stationarity and ergodicity of $\mathbb{1}\left[\text{CASg}_{t-\lambda,t}\right]$, where the event $\text{CASg}_{t-\lambda,t}$ is described in Eq. 9, these historical data are used to estimate $f_\tau(\lambda)$ using Eq. 12.

$$\mathbb{P}\left[\text{CASg}_{t-\lambda,t}\right] \approx \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \mathbb{1}\left[\text{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}\right] = \hat{f}_\tau(\lambda) \tag{13}$$

where $\text{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda} = \left\{\exists t' \in [\![\tilde{t}, \tilde{T}]\!], \quad \hat{\tau}_{t'} \cap [\![\hat{b}_{\tilde{t}}, \tilde{t} - \lambda]\!] \neq \emptyset\right\}$.

However, to improve computation time, the following expression of $\text{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}$ is preferred:

$$\text{CASg}^{\tilde{T}}_{\tilde{t}-\lambda,\tilde{t}} = \left\{ \left(\bigcup_{\tilde{T} \geq t' > \tilde{t}} \hat{\tau}_{t'} \right) \cap [\![\hat{b}_{\tilde{t}}, \tilde{t} - \lambda]\!] \neq \emptyset \right\} \tag{14}$$

With this formulation, each breakpoint is checked only once to see if it belongs to $[\![\hat{b}_{\tilde{t}}, \tilde{t} - \lambda]\!]$. Indeed, many breakpoints remain at the same position from one step to the next step while applying the breakpoint detection procedure.

Algorithm 5 implements Eq. 14 to give an estimation of f_τ. Where $I_{\tilde{t},u} = \mathbb{1}[\text{CASg}^{\tilde{T}}_{\tilde{t},u}]$ and $S_\lambda = \sum_{\tilde{t}=\lambda}^{\tilde{T}} I_{\tilde{t},\tilde{t}-\lambda}$ so $\hat{f}_\tau(\lambda) = \frac{S_\lambda}{\tilde{T}-\lambda}$.

Algorithm 5 Exact computation of probability of segment assignment change.

Require: $(\tau(\tilde{t}))_1^{\tilde{T}}$ list of successive segmentations
  $I, S \leftarrow 0$
  $\tau_{global} \leftarrow \emptyset$
  for $\tilde{t} \in [\tilde{T}, 1]$ **do**
    $\tau_{global} \leftarrow \tau_{global} \cup \hat{\tau}(\tilde{t})$
    for $u \in [\hat{b}_{\tilde{t}}, \tilde{t}]$ **do**
      for $b' \in \tau_{global}$ **do** ▷ b' is a breakpoint
        if $\hat{b}_{\tilde{t}} < b' \leq u$ **then**
          $I_{\tilde{t},u} \leftarrow 1$
        end if
      end for
    end for
  end for
  for $\lambda \in [0, \tilde{T}]$ **do**
    for $\tilde{t} \in [\lambda, \tilde{T}]$ **do**
      $S_\lambda \leftarrow S_\lambda + I_{\tilde{t},\tilde{t}-\lambda}$
    end for
    $\hat{f}_{\tau,\lambda} \leftarrow S_\lambda/(\tilde{T} - \lambda)$
  end for
  Output: $\hat{f}_{\tau,\lambda}$ list of $\hat{f}_\tau(\lambda)$ values for different λ
  return $\hat{f}_{\tau,\lambda}$

The complexity of the exact computation of $\hat{f}_\tau$, described in Algorithm 5, is quadratic in time and space, which is a drawback regarding any practical use. Another version of the implementation of $\hat{f}_\tau$ is given in Algorithm 7 which is more convenient for an online context since it is linear in time and space.

By observing the evolution of the breakpoints over time (not shown in this paper), it appears that the position of the last breakpoint is the most likely to change, while that of the other breakpoints are generally stable. This leads us to modify the characterization of the "assigned segment change" event by considering only the change caused by the last breakpoint instead of the entire segmentation.

$$\forall t, \lambda \in [\![1, T]\!]^2 \qquad \mathbb{1}[\mathrm{CASg}^T_{t,t-\lambda}] = \mathbb{1}\left[\exists t' \in [\![t, T]\!], \quad \hat{b}_t < \hat{b}_{t'} \leq t - \lambda\right] \tag{\textbf{Last}}$$

Under this assumption, the computation of $\hat{f}_\tau(\lambda)$ can be simplified using Proposition 3.

Proposition 3. Let $(X_t)_{1\leq t\leq T}$ be a time series of length T. Let $(\tau(t))_{1\leq \tilde{t}\leq \tilde{T}}$ be the sequence of successive segmentations of the time series. Let $(\mathbb{1}[C^T_{t,u}])_{1\leq t\leq T, 1\leq u\leq T}$ the family of "assigned segment change" events. Assume that the assumption (**Last**) is verified. Then the estimator $\hat{f}_\tau$ described in Eq. 14 is computed as

$$\hat{f}_\tau(\lambda) = \frac{1}{\tilde{T}} \sum_{\tilde{t}=1}^{\tilde{T}} \mathbb{1}\left[r_{\tilde{t}} > \lambda\right] \tag{15}$$

where $r_{\tilde{t}}$ is the maximum distance from the the end of the time series with X_t having segment reassigned. It is computed as:

$$r_{\tilde{t}} = \max_{t'>\tilde{t}, \hat{b}_{t'}>\hat{b}_{\tilde{t}}, \hat{b}_{t'}<\tilde{t}} \tilde{t} - \hat{b}_{t'} \tag{16}$$

Notice that $r_{\tilde{t}}$ does not depend on λ. It is sufficient to calculate $r_{\tilde{t}}$ once for all λ. Therefore, it's easy to deduce the value of $f_\tau(\lambda)$ for all λ. The most demanding part is the computation of $r_{\tilde{t}}$. Two implementations of $r_{\tilde{t}}$ computation are proposed. Algorithm 6 gives the most naive version, each $r_{\tilde{t}}$ is calculated one after the other. The problem is that the calculation of $r_{\tilde{t}}$ is itself linear in the length of the series. Therefore, the time complexity is quadratic. Algorithm 7 improves the computation by swapping the two loops. This limits the total number of comparisons performed. In the second loop, $\tilde{t}$ takes on the values between b'_t and t'. The number of values taken by $\tilde{t}$ is the length of a segment, not a the length of the time series. Algorithmic complexity is therefore linear.

Proof of Proposition 3. Based on Eq. 14, the estimator $\hat{f}_\tau$ is given by:

$$\hat{f}_\tau(\lambda) = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \mathbb{1}\left[\mathrm{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}\right]$$

With assumption (**Last**) it gives:

$$\mathbb{1}\left[\mathrm{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}\right] = \mathbb{1}\left[\exists t' \in [\![\tilde{t}, \tilde{T}]\!], \quad \hat{b}_{\tilde{t}} < \hat{b}_{t'} \leq \tilde{t} - \lambda\right]$$

The inequality $\hat{b}_{\tilde{t}} < \hat{b}_{t'} \leq \tilde{t} - \lambda$ is equivalent to $\hat{b}_{\tilde{t}} < \hat{b}_{t'}$ and $\tilde{t} - \hat{b}_{t'} > \lambda$ which gives

$$\mathbb{1}\left[\mathrm{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}\right] = \mathbb{1}\left[\bigcup_{t'>\tilde{t},\hat{b}_{t'}>\hat{b}_{\tilde{t}},\hat{b}_{t'}<\tilde{t}} \tilde{t} - \hat{b}_{t'} > \lambda\right]$$

Since, a set contains a number greater than λ, if and only if its maximum is greater than λ, it gives:

$$\mathbb{1}\left[\mathrm{CASg}^{\tilde{T}}_{\tilde{t},\tilde{t}-\lambda}\right] = \mathbb{1}\left[\left(\max_{t'>\tilde{t},\hat{b}_{t'}>\hat{b}_{\tilde{t}}} \tilde{t} - \hat{b}_{t'}\right) > \lambda\right]$$

Since $\lambda > 0$, when $\hat{b}_{\tilde{t}} < \hat{b}_{t'}$ and $\tilde{t} - \hat{b}_{t'} > \lambda$ it also implies that $\tilde{t} \geq \hat{b}_{t'}$ so $\mathbb{1}\left[\left(\max_{t'>\tilde{t},\hat{b}_{t'}>\hat{b}_{\tilde{t}}} \tilde{t} - \hat{b}_{t'}\right) > \lambda\right] = \mathbb{1}\left[\left(\max_{t'>\tilde{t},\hat{b}_{t'}>\hat{b}_{\tilde{t}},\hat{b}_{t'}<\tilde{t}} \tilde{t} - \hat{b}_{t'}\right) > \lambda\right]$. The number $r_{\tilde{t}}$ is introduced as equal to $\max_{t'>\tilde{t},\hat{b}_{t'}>\hat{b}_{\tilde{t}},\hat{b}_{t'}<\tilde{t}} \tilde{t} - \hat{b}_{t'}$. The $\hat{f}_\tau$ estimator can be written as follows

$$\hat{f}_\tau(\lambda) = \frac{1}{\tilde{T}} \sum_{\tilde{t}=1}^{\tilde{T}} \mathbb{1}\left[r_{\tilde{t}} > \lambda\right]$$

□

Algorithm 6 Naive computation of $r_{\tilde{t}}$.

```
for t̃ in [[1, T̃]] do
    for t' in [[t̃, T̃]] do
        if b̂_{t'} < t̃ and b̂_{t̃} > b̂_{t'} then
            r_{t̃} = max(r_{t̃}, t̃ − b̂_{t'})
        end if
    end for
end for
```

Algorithm 7 Efficient computation of $r_{\tilde{t}}$.

```
for t' in [[1, T̃]] do
    for t̃ in [[b̂_{t'}, t']] do
        if b̂_{t̃} > b̂_{t'} then
            r_{t̃} = max(r_{t̃}, t̃ − b̂_{t'})
        end if
    end for
end for
```

5.3.2. Application on simulated data

In the previous Section 5.3.1, two algorithms for estimating the probability of a segment assignment change were described: the exact estimation using Algorithm 5 and an efficient estimation using Algorithm 7. In this section, these different methods are assessed by experiments on simulated data.

Description of the experiment. The following notations are used: Let T be the length of the time series, θ the average segment length, Δ the size jumps to generate a breakpoint and σ the standard deviation of the data point within a segment.

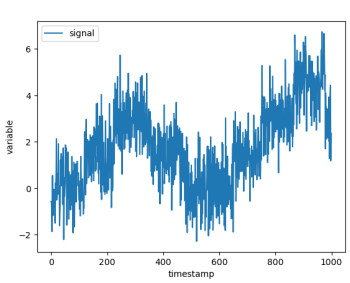
Time series are generated according to the following rules:

- The number of breakpoints follows the exponential distribution $D \sim Exp(T/\theta)$.
- Each breakpoint position is generated according to uniform distribution $\forall i \in [1, D], \tau_i \sim U(1, T)$

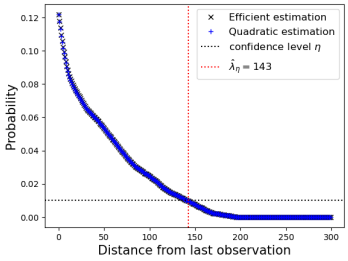
- The mean of the time series μ_i is piecewise constant with respect to the segmentation τ_i, with $\mu_{\tau_i} - \mu_{\tau_i+1} = \xi\Delta\sigma$
- The time series is generated according to the following rule $X_t \sim \mathcal{N}(\mu_t, \sigma)$

Then $\hat{f}_\tau(\lambda)$ is estimated using the two different methods: Algorithm 5 and Algorithm 7.

Results and analysis:. An example of generated time series is illustrated in Figure 9a. Figure 9b gives the estimated probability of segment assignment change according to the two estimation Algorithms 5 and 7. The two algorithms give results that are almost the same, as shown in Figure 9b. The selected $\lambda_\eta^\star$ is equal to 143, in the two cases. This supports assumption that (**Last**) is verified. In practice, we recommend to use the Algorithm 7 method since it is more computationally efficient. To compute the probability $\hat{f}_\tau(\lambda)$ on a PC (4 CPU, 16G), the Algorithm 7 gives results within 30 seconds compared to the exact computation which gives the results within 15mn, for a time series of length 10^4.



(a) Generated time series

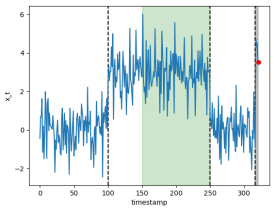


(b) Probability of assignment change as a function of distance to time series end.

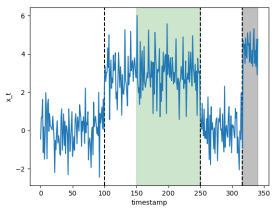
Figure 9: Results for estimation of the "segment assignment change" probability

5.4. How to estimate the probability of a status change under a stable breakpoint?

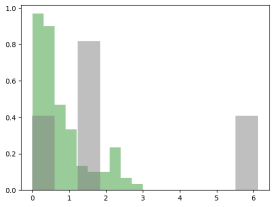
As introduced in Section 5.1, $f_d(\ell)$ is the probability that the status of a point changes under the conditions the last breakpoint remains unchanged and the segment cardinality is equal to ℓ. This probability $f_d(\ell)$ is necessary to build the active set containing data points with uncertain status, as described in Section 5.2. In this section, a procedure to estimate $f_d(\ell)$ is proposed.



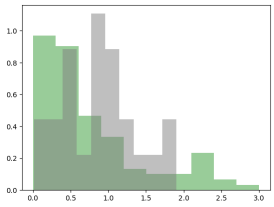
(a) Starting new segment with small length.



(b) Data points added to the current segment becoming larger.



(c) Z-score for small segment



(d) Z-score for larger segment

Figure 10: Atypicity score estimation according to the length of the current segment.

Figure 10 illustrates how the the length of the current segment has an influence on the accuracy of the atypicity score estimation and consequently on the uncertainty of a data point status. Indeed, Figure 10a shows a time series with a newly detected current segment highlighted in gray color, and a calibration set in green color inside the previous segment. The atypicity scores, z-scores based on the mean and the standard deviation, are shown in Figure 10c computed for the current segment in gray and the calibration set in green. Since the current segment has few points, its z-score estimation shows a high discrepancy compared to the score distributions of the calibration set, despite the fact that there are no anomalies. As shown in Figure 10d, when new data points are added, the estimation of the abnormality score of the current segment is more accurate.

This example highlights that the status of a data point can change even if the breakpoints remain unchanged, whereas Section 5.3 deals only with the case where the change in status is due to a change in the detected breakpoints. Uncertainty also comes from having too few points in the current segment, leading to score estimation errors. Estimating the probability $\hat{f}_d(\ell)$ is useful to build the active set that takes this into account. The following section suggests a procedure to estimate the probability $f_d(\ell)$.

5.4.1. Description of the method

The method is based on the learning phase using a set $\mathcal{D}$ of historical detected segments having a low probability to change. This training set D is defined by,

$$\mathcal{D} = \{(X_1, \ldots, X_{\hat{\tau}_{1,T}}), (X_{\hat{\tau}_{1,T}+1}, \ldots, X_{\hat{\tau}_{2,T}}), \ldots, (X_{\hat{\tau}_{D-1,T}+1}, \ldots, X_{\hat{\tau}_{D,T}})\} \tag{17}$$

In the following, the training procedure is based on six different steps needed to estimate the $\hat{f}_d(\ell)$ probability. Let $\overline{a}$ be the NCM (Non Conformity measure), used to define the atypicity score, as described in Section 4. As a reminder, $\overline{a}(S, x)$, measures the "nonconformity" between the set S and the point x.

Training procedure:. The principle of the training phase is to simulate, using resampling, numerous examples where the current segment changes from a length ℓ to a larger length. At each case,

anomaly detection is applied to the test set of cardinality m and the proportion of statuses that have changed by modifying the length of the current segment is measured. Since the breakpoints are assumed to be stable, the simulation is inspired by the description of the detector given in Section 2.2, without the parts concerning breakpoint detection. These steps are repeated B times. Let $b \in [\![1, B]\!]$:

Step 1: Figure 11a illustrates that two segments are resampled from the historical data set $\mathcal{D}$. $\mathcal{S}_{1,b}$ is considered as the calibration set and $\mathcal{S}_{2,b}$ as a current segment in the simulation:

$$\mathcal{S}_{1,b}, \mathcal{S}_{2,b} \sim U(\mathcal{D})$$

Step 2: The current segment $\mathcal{S}_{2,b}$ is sub-sampled into a smaller segment of length ℓ and noted $\tilde{S}_{2,\ell,b}$, as shown in Figure 11b. $\tilde{S}_{2,\ell,b}$ is considered as the same segment than $\mathcal{S}_{2,b}$ at a previous step, having only ℓ points.

$$\tilde{\mathcal{S}}_{2,\ell,b} \sim U(\mathcal{S}_{2,b})$$

.

Step 3: The current segment $\mathcal{S}_{2,b}$ is sub-sampled into an other segments of size m and noted $\overline{S}_{2,m,b}$. $\overline{S}_{2,m,b}$ is considered as the test set.

$$\overline{\mathcal{S}}_{2,m,b} \sim U(\mathcal{S}_{2,b})$$

Step 4: As illustrates in Figure 11c and 11d, the scores of the three segments are computed:

- The score of $\mathcal{S}_{1,b}$:

$$\forall i \in [\![1, n]\!], X_i \in \mathcal{S}_{1,b}, \quad c_{i,b} = \hat{a}(\mathcal{S}_{1,b} \backslash \{X_i\}, X_i)$$

- The score of the test set using $\mathcal{S}_{2,b}$ as training set:

$$\forall i \in [\![1, m]\!], Y_i \in \overline{S}_{2,m,b}, \quad s_{i,b} = \hat{a}(S_{2,b} \backslash \{Y_i\}, Y_i)$$

- The score of the test set using $\tilde{S}_{2,\ell,b}$ as a training set:

$$\forall i \in [\![1m]\!], Y_i \in \overline{S}_{2,m,b}, \quad \tilde{s}_{i,\ell,b} = \hat{a}(\tilde{S}_{2,\ell,b} \backslash \{Y_i\}, Y_i)$$

Step 5: Figure 11e illustrates that the empirical p-values are computed for the two scores obtained from the test set using the same calibration set:

- p-values of the test set when using the complete current segment as training set: $\forall i \in [\![1, m]\!], p_{i,b} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}[s_{i,b} < c_{j,b}]$
- p-values of the test set when using the length ℓ sub-sample of the current segment as training set: $\forall i \in [\![1, m]\!], \tilde{p}_{i,\ell,b} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}[\tilde{s}_{i,\ell,b} < c_{j,b}]$

Step 6: Detect the anomalies in the two cases, by applying the Benjamini-Hochberg procedure $\hat{\varepsilon}_{BH_\alpha}$ on the estimated p-values, as shown in Figure 11f:

- In case the training set is the entire current segment:

$$\forall i \in [\![1, m]\!], \quad d_{i,b} = \mathbb{1}[p_{i,b} < \hat{\varepsilon}_{BH_\alpha}(p_{1,b}, ..., p_{m,b})]$$

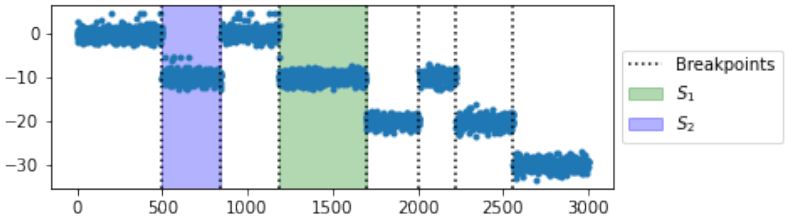
- In case the training set is the size ℓ sub-sample:

$$\forall i \in [\![1, m]\!], \quad \tilde{d}_{i,\ell,b} = \mathbb{1}[\tilde{p}_{i,\ell,b} < \hat{\varepsilon}_{BH_\alpha}(p_{1,\ell,b}, ..., p_{m,\ell,b})]$$

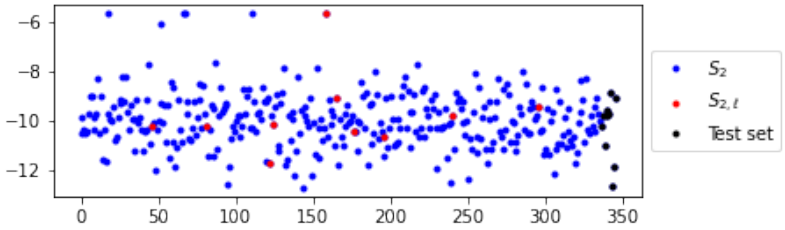
Step 7: The number of decisions that differ between the two cases, $\mathcal{S}_{2,b}$ or $\tilde{\mathcal{S}}_{2,\ell,b}$ used as the training set, is computed:

$$n_d = \sum_{i=1}^{m} \mathbb{1}[d_{i,b} \neq \tilde{d}_{i,\ell,b}]$$

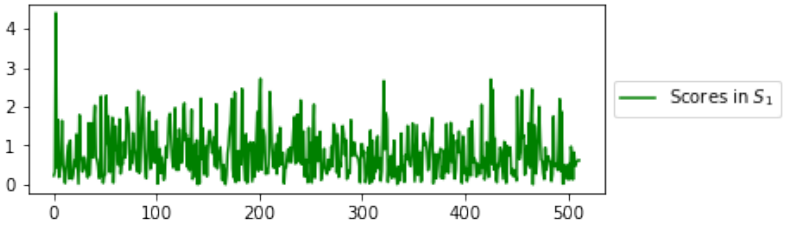
The training procedure simulates the behavior of the online anomaly detector: $\mathcal{S}_1$ plays the role of the calibration set. $\mathcal{S}_2$ plays the role of current segment. $\mathcal{S}_{2,\ell}$ plays the role of the current segment at the beginning of a new segment. The first m elements $Y_1, \ldots, Y_m$ from $\mathcal{S}_2$ constitute the test set.



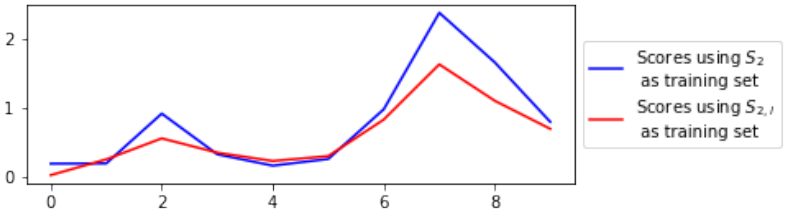
(a) Step 1: Segments resampling



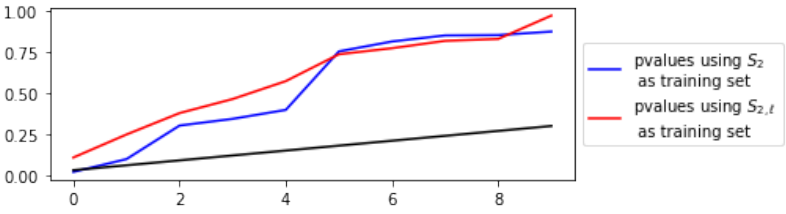
(b) Step 2 and 3: Sub-sampling



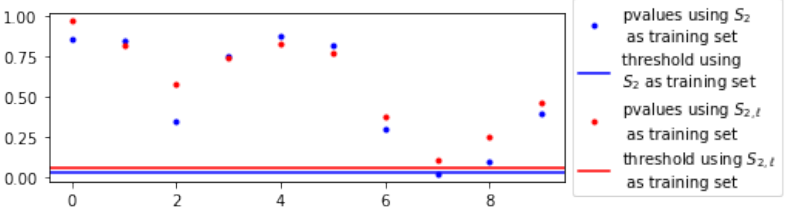
(c) Step 4: Calibration set scoring



(d) Step 5: Test set scoring



(e) Step 6: p-value estimation



(f) Step 7: Anomaly detection

Figure 11: Illustration of the different steps of the training procedure to estimate the status change probability under stable breakpoints.

Assuming stationarity and piecewise dependence, as stated in Definition 1, by repeating this resampling process many times, as the length of the time series converges to infinity, the proportion of status changes converges to the expectation, according to the law of large numbers [40]:

$$\lim_{T,B\to\infty} \frac{1}{mB} \sum_{b=1}^{B} \sum_{j=1}^{m} \mathbb{1}[d_{j,b} \neq \tilde{d}_{j,\ell,b}] = \mathbb{E}_{\mathcal{S}_1,\mathcal{S}_2\sim U(\mathcal{D})} \mathbb{E}_{\mathcal{S}_{2,\ell}\sim U(\mathcal{S}_2)} \sum_{i=1}^{m} \mathbb{1}[d_i \neq \tilde{d}_i]$$

Under the assumptions of score stationarity stated in Definition 1, the limit is equal to $f_d(\ell)$. Indeed, under score stationarity, the calibration set can be built from any segment of the time series. This implies that it is possible to use described training procedure as an estimator of $\hat{f}_d(\ell)$.

$$\hat{f}_d(\ell) = \frac{1}{mB} \sum_{b=1}^{B} \sum_{j=1}^{m} \mathbb{1}[d_{j,b} \neq \tilde{d}_{j,\ell,b}] \approx f_d(\ell)$$

5.4.2. Application on simulated data

The training procedure in Section 5.4.1 is applied for different scoring functions adapted to different types of time series considered in Section 4. : The goal is to check if the estimation approach of $\hat{f}_d(\ell)$ can be applied to different scoring functions.

Description of the experiment. Different series that require different scoring functions are considered: Gaussian and Mixture of Gaussian.

- Figure 12a shows a Gaussian white noise with anomalies in distribution tail.

$$\begin{aligned} \forall t \in [\![1, T]\!], \quad & A_t \sim Ber(\pi), \\ & \text{if } A_t = 0, X_t \sim \mathcal{N}(0, 1) \\ & \text{else } X_t = \Delta \end{aligned}$$

  The z-score applied on X_t to detect anomalies that are in the tail of the distribution, is computed by,

$$\hat{a}(S, X_t) = |X_t - \hat{\mu}_S|/\hat{\sigma}_S \tag{18}$$

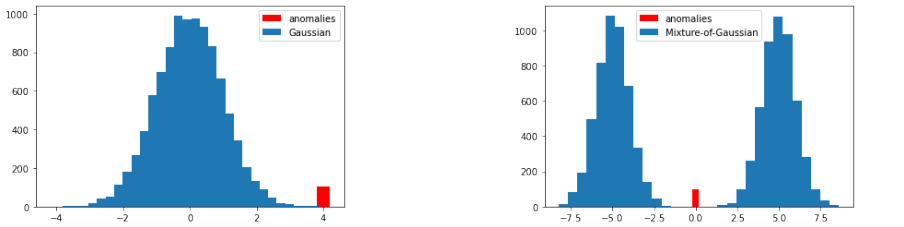
  where S is a segment of data, $\hat{\mu}_S$ the mean estimator on S and , $\hat{\mu}_S$ the standard deviation on S

- Figure 12b shows a Mixture of Gaussians with anomalies between the distribution modes.

$$\begin{aligned} \forall t \in [\![1, T]\!], \quad & A_t \sim Ber(\pi), \\ & \text{if } A_t = 0, X_t \sim 0.5\mathcal{N}(\Delta, 1) + 0.5\mathcal{N}(-\Delta, 1) \\ & \text{else } X_t = 0 \end{aligned}$$

  The kernel based score, inspired from other works on kernel based anomaly detection [41, 42], applied to detect anomalies having large distance from the normal data, is computed by,

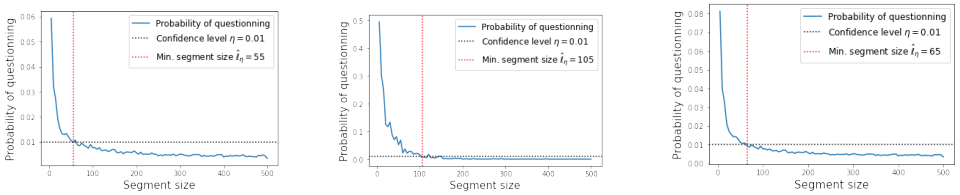
$$\hat{a}(S, X_t) = \frac{1}{|S|^2} \sum_{s,s'\in S^2} K(s, s') - \frac{2}{|S|} \sum_{s\in S} K(X_t, s) + K(X_t, X_t) \tag{19}$$



(a) Gaussian with anomalies in tail (b) Gaussian mixture with anomalies in center

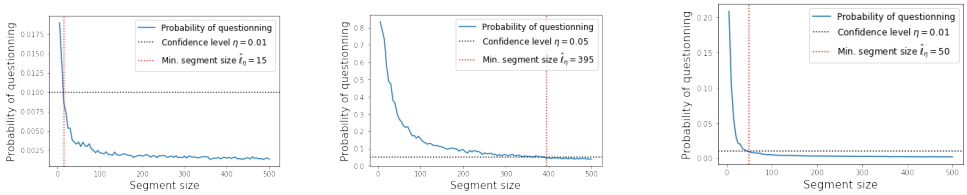
Figure 12: Different time series distributions and anomalies.

Results and analysis. As stated previously, two types of time series are considered in the experiments: results of Gaussian data shown in Figure 13 and results of Gaussian mixture data shown in Figure 14. For both, three line charts representing the probability of status change as a function of the current segment length in relation to the initial status: (a) the status is normal, (b) the status is abnormal and (c) unknown status.



(a) normal status (b) abnormal status (c) unknown status

Figure 13: Probability that status changes under stable breakpoints as a function of segment length, for Gaussian data.



(a) normal status (b) abnormal status (c) unknown status

Figure 14: Probability that status change under stable breakpoint as a function of segment length, for Gaussian mixture data.

For Gaussian data and in the unknown status, Figure 13c shows clearly that the probability of status change decreases with the length of the current segment. This probability is higher when the status is abnormal, as shown in Figure 13b. Nevertheless, with a segment length of 100, the probability is less than 1%. For Gaussian mixture data and in the abnormal status scenario shown in Figure 14b, the length of the current segment needs to be at least equal to 500 to get a probability of changing status around 5%. For the normal status scenario in Figure 14a, the probability of changing quickly decreases to 0. The results are also promising in the unknown status scenario in Figure 14c, where the change probability is low.

Conclusion. A solution to compute the probability of status change under "stable" breakpoints has been built. Empirical results show that the choice of an optimal $\hat{\ell}_\eta$ which reduces the uncertainty of a data point status depends on the type of data and the scoring function that is used. The method can help to select an atypicity score. A good atypicity score, satisfying requirements discussed in Section 4 (been robust and efficient) should have low $\hat{\ell}_\eta$ value.

6. How to build the calibration set?

Section 2.4 introduces the notion of calibration set by giving a high level description of the Breakpoint Based Anomaly Detector. It is a collection of data points representing the reference behavior, inspired by Conformal Anomaly Detection[27, 43]. It is built using data from the current segment, or from another segment in the history with a similar distribution probability compared to the current segment. The cardinality of the calibration set follows two constraints:

- it should be large enough to ensure that the p-values are estimated with sufficient precision to generate a low false positive and false negative rate.
- it should not be too large to maximize the homogeneity of the data and to limit computation time.

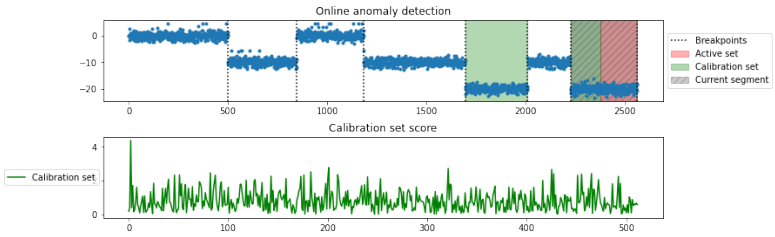


Figure 15: Illustration of the current segment, the active set and the calibration set.

As shown in Figure 15, while data are collected online, the length of the current segment after the new breakpoint is too small to build the whole calibration set. By identifying similar segments and merging them to build the calibration set, the current segment can be completed with enough data points to estimate the p-values accurately. Similar segments are found using a similarity function, like the Bhattacharyya distance proposed in [44]. This similarity function is defined between two segments S_1, S_2 with means μ_1 and μ_2 and standard deviations σ_1 and σ_2 by:

$$d(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{8\sigma^2}(\mu_1 - \mu_2)^2 + \frac{1}{2}\ln\frac{\sigma}{\sqrt{\sigma_1\sigma_2}} \tag{20}$$

The similarity function allows to sort all historical segments according to their similarity to the current segment. To build a calibration set of cardinality n, it is initialized using the scores of data points of the current segment that are not assigned to the active set. The data points scores with a "normal" status from the previous segments are added to the calibration set in descending order of similarity until n scores are reached. After having described how a calibration set of a given cardinality n is built, Section 7 describes how the optimal cardinality n is chosen.

7. p-value, threshold and optimal calibration set cardinality

After having defined the active set and the calibration set, the empirical p-values of each data point of the active set are computed using the calibration set. The threshold is chosen using the

p-values of the active set to ensure the control of the FDR at a given level α. Finally, the status of each data point of the active set is reevaluated comparing its p-value to the threshold.

In [26] we detail a new strategy for controlling the FDR of an anomaly detector in the online framework. This goal is achieved by efficiently controlling the modified FDR criterion (mFDR) of subseries so that the FDR value of the full time series is controlled at the prescribed level α. To be lore specific, [26] designs a modified version of the Benjamini-Hochberg procedure. Instead of applying BH to the active set with a slope α, it is applied with a slope $\alpha' = \frac{\alpha}{1+\frac{1-\alpha}{m\pi}}$, where m denotes the length of the active set, α is the desired global FDR level, and π refers to the proportion of anomalies.

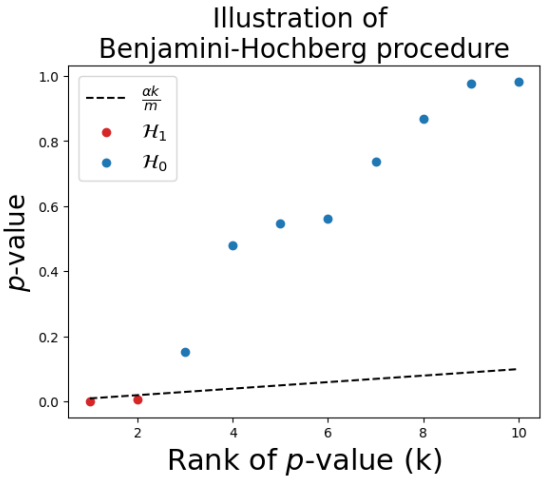


Figure 16: Example of Benjamini-Hochberg procedure.

The calibration set is used to compute the p-values. The FDR and the FNR of the modified BH procedure is very sensitive to the cardinality of the calibration set used to estimate the p-value. In [26], we study under which conditions the cardinality of the calibration set ensures a control of the FDR. Given m the cardinality of the active set and α' the modified slope for BH, the calibration set cardinality has to be chosen among:

$$n \in \{k\frac{m}{\alpha'} - 1, \quad k \in \mathbb{N}^*\} \tag{21}$$

As explained more deeply in [26], the number of false negatives decreases with higher k. But a larger k also increases the computation time, which can make any real-time decision difficult. We recommend to try different values of k, to monitor the decision time and to choose the largest k which allows real time decisions.

In order to verify that the results of article [26] apply to the present case, two conditions need to be satisfied. First, it must be verified that the mFDR can be controlled locally by the mBH. For this purpose, it must be verified that the p-values verify superuniformity and positive regression dependency (PRDS) properties. A sufficient condition is to assume that the score is iid, a property stronger than the property of stationarity with piecewise dependence described in Definition 1. Moreover, since in our configuration the calibration set is the same for all values in the active set, the local control of the mFDR on the active set is strict. Second, to obtain FDR

control over the whole series, the status series $d_{u,t}$ needs to be weakly dependent. It needs to be show that there exists a number e, such that $d_{u_1,t}$ and $d_{u_2,t}$ are independent if $|u_1 - u_2| > e$. Ones assumed that true breakpoints are detected and that the score is iid, then the status verify this property, with $e = n + m$ in the case the calibration set is chosen just after the active set. In practice, such properties are difficult to verify, but the choice of a good breakpoint detector, a good atypicity score and an active set allow to come closer.

8. Empirical study

An anomaly detector based on breakpoint detection has been proposed in Section 2.4. The core components have been separately elaborated and evaluated in Sections 3, 4, 5, 6 and 7 . In this section, the performance of the whole anomaly detector is assessed. The experiments are conducted in several steps. First, the anomaly detector is applied to several synthetic time series. The flexibility of the detector is evaluated and the roles played by the kernel and the atypicity score are highlighted. Second, the anomaly detector is applied by choosing different hyperparameters involved in the core components, not necessarily the same as those proposed in the previous analyses. The relevance of the different components and their associated analyses are evaluated. Third, the anomaly detector is applied by replacing some estimators with true knowledge in order to explore more deeply the reasons for the errors made by the anomaly detector. Finally, the anomaly detector is evaluated against alternative anomaly detectors.

An experimental framework is designed to conduct the experiments and to evaluate different aspects of the anomaly detector. The framework described in Section 8.1 is adapted for different time series and anomaly detector parameters.

8.1. Experimental framework

Let's consider a time series generation process and an anomaly detector. The following steps are repeated on different samples of the time series:

1. Generate the time series, according to the the first reference distribution $\mathcal{P}_{0,1}$, the proportion of anomalies π, the alternative distribution $\mathcal{P}_{1,1}$ and the transition rule describing how the parameters of the reference distribution will change between two segments.
   (a) The number D of breakpoints is generated by $Exp(T/\theta)$, where θ is the average distance between two breakpoints.
   (b) The position of the D breakpoints follows $U([1, T])$. In addition to the previous step, this implies that the process of breakpoint positions is a Poisson process.
   (c) The rule is applied iteratively to get the reference and alternative distributions for each segment. Two types of rules are considered:
      - Breakpoint in the mean with a jump size of Δ. For each i in $[\![1, D-1]\!]$, let μ_i be the mean of the reference distribution in the ith segment. The mean of a segment is equal to the mean of the previous one shifted with Δ.

$$\forall i \in [\![1, D-1]\!], \quad \mu_{i+1} = \mu_i + \zeta_i \Delta \tag{22}$$

        With ζ_i, a random variable following the Rademacher distribution and defining the sign of the jump.

- Breakpoint in the variance with a jump scale size of Δ. For each i in $[\![1, D-1]\!]$, let σ_i be the standard deviation of the reference distribution in the ith segment. The standard deviation of a segment is equal to the standard deviation of the previous segment multiplied or divided by Δ.

$$\forall i \in [\![1, D-1]\!], \quad \sigma_{i+1} = \exp(\zeta_i \ln \Delta/2) * \sigma_i \tag{23}$$

With ζ_i, a random variable following the Rademacher distribution and defining if the standard deviation is multiplied or divided by Δ.

(d) The position of anomalies are generated by a Bernoulli distribution: $A_t \sim Ber(\pi_1)$

(e) All the values of the time series are computed as follows:

$$\forall i \in [\![1, D]\!], \quad \forall t \in [\![\tau_i, \tau_{i+1}[\![, \quad \begin{cases} X_t \sim \mathcal{P}_{0,i}, & \text{if } A_t = 0 \\ X_t \sim \mathcal{P}_{1,i}, & \text{otherwise} \end{cases}$$

2. Apply the anomaly detector on the generated time series. Three core components need to be defined:
   (a) the appropriate kernel to identify the breakpoints using KCP,
   (b) the scoring function a
   (c) and parameters n for the length of the calibration set and λ and ℓ to define the active set.
3. Compare the detections with true anomalies and calculate the proportion of false discoveries and of false negatives.

The two criteria FDR and FNR are estimated as the average of the FDP and of the FNP over all repetitions.

The experimental framework is used in different scenarios: At Section 8.2, different synthetic time series are tested and analyzed. At Section 8.3, the effect of hyperparameter choice on performance is evaluated. At Section 8.4 the causes of underperformances of the anomaly detector are studied. Finally, in Section 8.5, the proposed anomaly detector is compared to alternative anomaly detectors using various public data collections.

8.2. Application on synthetic data

The goal of this section is to check if the breakpoint based anomaly detector is able to detect anomalies with a controlled FDR considering different scenarios of time series. For the first scenario, Gaussian time series are considered with breakpoints in the mean and anomalies in the tail of the distribution. This simplest scenario is used as a reference before evaluating a more complex one. The second scenario considers Gaussian mixture time series with breakpoints in the mean and anomalies in the center of the distribution between the two Gaussian modes. In this case, the detector is checked for anomalies that are not present in the tail of the distribution. For the third scenario, 2D Gaussian time series with breakpoints in the covariance are used to evaluate the detector on multidimensional data. Indeed, the breakpoint in the covariance ensures that breakpoints and anomalies cannot be detected by applying the anomaly detector to each dimension. The third scenario evaluates the detector on heteroscedastic time series, considering Gaussian time series with breakpoints in the mean and in the variance. For the last scenario, Gaussian data with breakpoints in the variance are used to evaluate how the anomaly detector can be applied with changes in the variance, which is a more difficult case study.

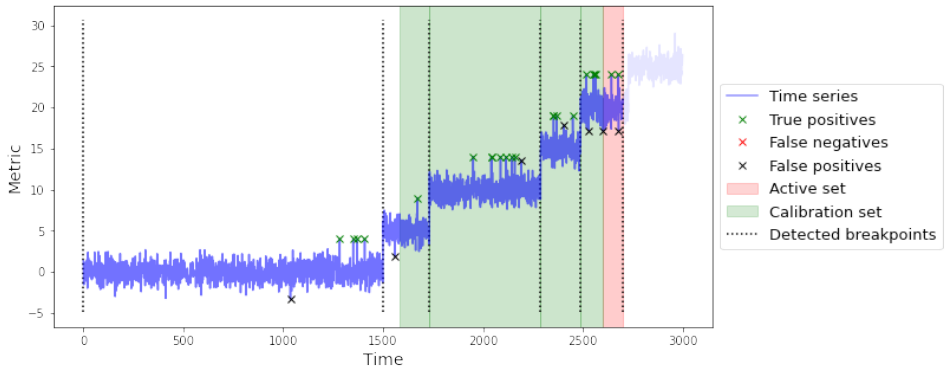
8.2.1. Gaussian time series with breakpoints in the mean

This scenario considers Gaussian data with breakpoints in the mean. The z-score is used to capture anomalies. The ability of the detector to control the FDR with a low FNR on different difficulties is assessed by varying the desired level of FDR control α and the size of the shift between two segments Δ.

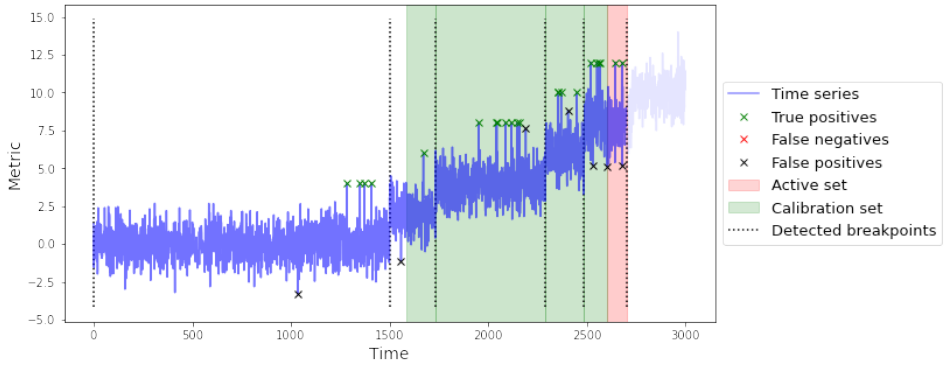
Description of the experiment. By applying the framework of Section 8.1, multiple choices have been made:

- The Gaussian distribution is considered as the reference $\mathcal{P}_{0,1}$ and the proportion of anomalies is equal to $\pi = 0.01$. These anomalies are generated in the tail of the reference distribution and follow $\Delta'\zeta$, where ζ is the Rademacher distribution and $\Delta' = 4$ is the spike size of the anomalies.
- The transition rule between two breakpoints is a jump in the mean of size Δ taking values in $\{2, 3, 5\}$.
- For the breakpoint detector, the Gaussian kernel with bandwidth estimated using the median heuristic is considered, as presented in Section 3. The z-score is used as the scoring function with the mean estimated using the median estimator and the standard deviation estimated using the biweight midvariance estimator, as defined in Section 4.1.1.
- According to preliminary experiments in Section 5, the active set is built using $\hat{\lambda} = \hat{\ell} = 100$. Based on the rules defined in Section 7, Benjamini-Hochberg is applied on the active set with the modified parameter $\alpha' = \frac{\alpha}{1+\frac{1-\alpha}{m\pi}}$. The calibration set is built according to the rules of Section 7, where the value n is chosen equal to $m/\alpha' - 1$. Two cases are considered $\alpha = 0.2$ and $\alpha = 0.1$. In the case $\alpha = 0.2$, then the following values are chosen $\alpha' = 0.1$ and $n = 999$. In the case $\alpha = 0.1$, then $\alpha' = 0.05$ and $n = 1999$.

Results and analysis. Figure 17 shows an example of anomaly detection for one time series. The x-axis is the timestamp and the y-axis the value of the generated time series, shown in blue. The light blue data points are those that are not observed at the time the results are presented. The vertical black lines are the detected breakpoints, the red band is the subseries defined as the active set, the green band is the subseries used to build the calibration set. Detected anomalies are the green crosses, false positives are the black crosses and red crosses are the false negatives. As shown in Figure 17a, there are no false negative and the false positives seem to be a small fraction of the true detected anomalies. As expected, the breakpoints are positioned exactly where the means of the series change. The active set contains the most recent observations. And the calibration set gathers data from several segments since the current segment does not contain enough data.



(a) $\Delta = 5$



(b) $\Delta = 2$

Figure 17: Application of our anomaly detector on Gaussian time series having breakpoints in the mean, for different shift size values Δ.

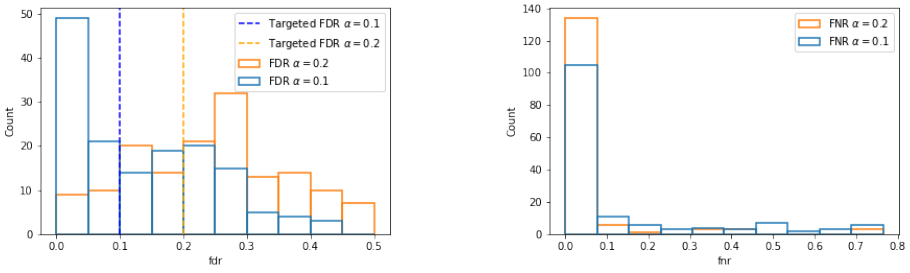
Table 1 gives the FDR and the FNR after having applied the anomaly detector to a collection of $B = 50$ Gaussian time series with breakpoint in the mean for different shift sizes Δ. The FNR is always close to 0. This is necessary to ensure the FDR control with the modified BH procedure. For all the cases, the FDR remains close to the desired α level. The FDR is well influenced by the choice of α level but less by the value of Δ. However, it is always slightly higher than alpha. Indeed, for $\Delta = 5$, it is equal to 0.23 instead of $\alpha = 0.20$, as shown in Table 1.

α	Δ	FDR	FNR
0.10	2	0.133	0.123
	3	0.134	0.111
	5	0.129	0.106
0.20	2	0.242	0.039
	3	0.242	0.042
	5	0.236	0.037

Table 1: FDR and FNR for anomaly detection on Gaussian time series having breakpoints in the mean according to the α level and the shift size Δ.

The histogram in Figure 18 shows more detailed results applied to the collection of time series

for different values of α parameter. Figure 18a shows the distribution of the FDR values compared to the target FDR in vertical lines. Figure 18b shows the distribution of the FNR values. As shown in Figure 18a,the performance of the anomaly detector is poor for some time series since the FDR values are higher and far from the target FDR. This explains why the measured FDR is slightly higher than the targeted FDR in Table 1. The diagnosis of this inefficiency will be examined in Section 8.4. In the next Sections 8.2.2, 8.2.3, 8.2.3, 8.2.4 and 8.2.5 the anomaly detector is applied and checked to more complex time series.



(a) FDR

(b) FNR

Figure 18: Histograms of the FDR and FNR for different targeted FDR α levels.

8.2.2. Gaussian mixture time series with breakpoints in the mean

In this section, the aim is to show how to handle anomalies that occur between two modes of a Gaussian mixture. These anomalies, which do not occur in the tail of a distribution, cannot be detected by z-scores because they are close to the mean. Therefore, it is necessary to adapt to this new situation by using another atypicity score, such as the kNN score introduced in [43]. Indeed, in this case, anomalies can be characterized by their distance from other segment data.

Description of the experiment. The anomaly detector applied to Gaussian mixture data considers the reference distribution $\mathcal{P}_{0,1} = 0.5\mathcal{N}(\Delta', 1) + 0.5\mathcal{N}(-\Delta', 1)$, with an anomaly spike size of $\Delta' = 6$. The anomalies are chosen to be equal to 0 to ensure they lie in the middle between the two Gaussian distributions.

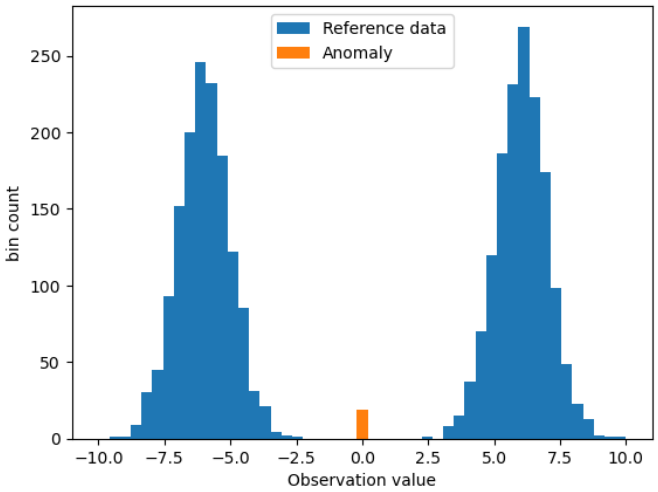


Figure 19: Histogram that represent the Gaussian mixture reference distribution with anomalies in the center.

As explained previously, to adapt to this new difficulty of time series data with Gaussian mixture, the atypicity score needs to be chosen accordingly. The kNN score introduced in [43] is applied. To ensure that the distribution of the score is the same between two segments and not affected by segment cardinality, the kNN distance is computed after having resampled $B_s = 100$ points from the segment. To obtain a robust score, the number k of nearest neighbors should be chosen carefully because the kNN distance should not be affected by the presence of anomalies in the segment. In particular, the k nearest neighbors of an anomaly should not be an anomaly, otherwise the distance will be close to 0, which leads to a false positive. By choosing $k = 10$ and ensuring $k/B = 0.1 >> 0.01 = \pi$, this issue is avoided with high probability. Experimental parameters not specified in this section have the same values as in Section 8.2.1.

Results and analysis. The result in Figure 20 clearly shows that for this example, the anomaly detector is able to detect the breakpoints, in the dashed black lines, and the anomalies, represented by the green crosses, with few false positives.

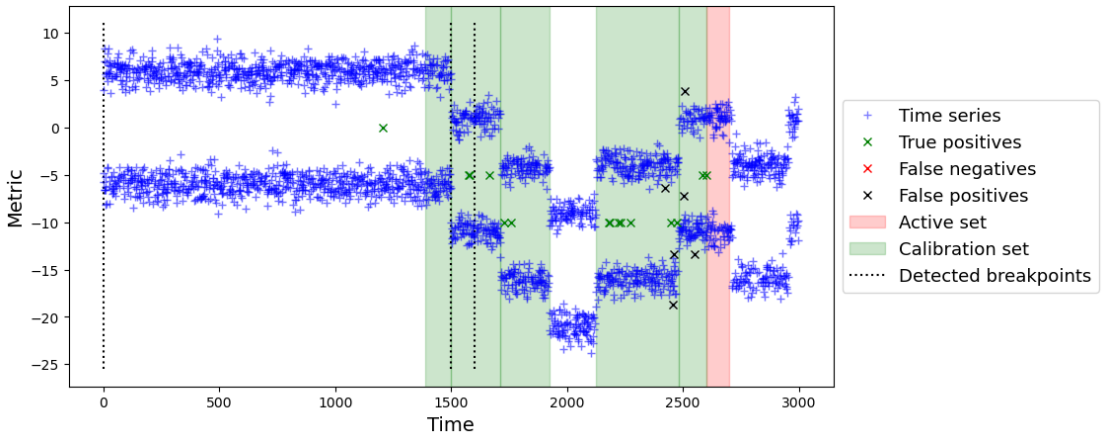


Figure 20: Application of our anomaly detector on Gaussian mixture time series having breakpoints in the mean.

The anomaly detector has been applied to 50 time series and the results are summarized in Table 2. The FDR is controlled at the desired level of 0.1 or 0.2 while the FNR is slightly higher compared to the Gaussian case in Table 1. This is probably due to the kNN score, which is less efficient than the z-score.

α	FDR	FNR
0.1	0.118	0.246
0.2	0.202	0.137

Table 2: FDR and FNR for anomaly detection on Gaussian mixture time series with breakpoints in the mean according to α level.

8.2.3. 2D Gaussian time series with breakpoint in the covariance

In this section, the aim is to show how to handle anomalies that occur on multidimensional data. Previously, the kernel method in KCP demonstrated high accuracy in detecting breakpoints for univariate time series data. Hopefully, the paper [45] shows that this kernel method is also applicable to multivariate time series. Once the time series is segmented, a scalar atypicity score is computed for the multidimensional data points. An alternative would be to apply univariate anomaly detection to each univariate time series. However, some breakpoints, such as those occurring in the covariance, cannot be detected by this alternative method.

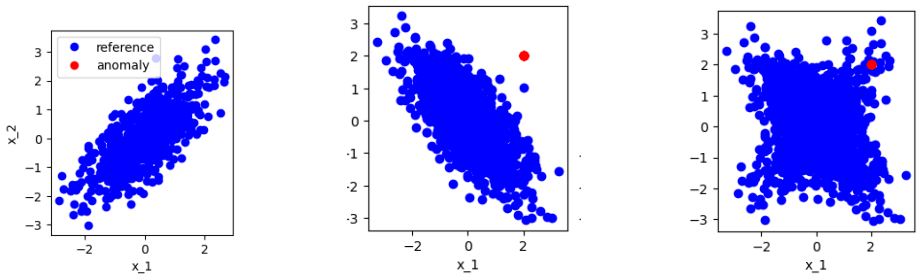
Description of the experiment. Data are generated according the following rule:

$$\forall t \in [\![1, T]\!], \quad X_t = \begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} \sim \mathcal{N}(0, \Sigma_t) \tag{24}$$

With the covariant matrix equal to:

$$\Sigma_t = \begin{cases} \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} & \text{if } t \leq \tau_1 \\ \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} & \text{else} \end{cases} \tag{25}$$

The reference distribution generates two-dimensional Gaussian data X_t. For each component the mean is 0 and the standard deviation is 1. To simplify the generation, one breakpoint τ_1 is considered linked to the change of the covariance from 0.7 to -0.7. Figure 21a shows that the covariance is positive before the breakpoint and negative after the breakpoint in Figure 21a. Anomalies are considered in the second segment, and are set to the value $(1, 1)$. This value has interesting properties to evaluate the capacity of the anomaly detector. First, "1" appears as a typical value at each one dimensional component of the time series. This implies that the anomalies cannot be detected by working on each component independently. Second, the value $(1, 1)$ is fairly typical before the breakpoint τ_1, as shown Figure 21a. Consequently, the breakpoint detector enables the detection of anomalies while they are hidden in the data mixture as shown in Figure 21c.



(a) Before the breakpoint. (b) After the breakpoint. (c) Data mixture.

Figure 21: 2D Gaussian data with breakpoint in covariance matrix

For this scenario, the Gaussian kernel is used to detect the breakpoint in the covariance. As a characteristic kernel, it should detect the change in the covariance, which is the change at the second moment order. The median heuristic is used to select the bandwidth. Since each component cannot be treated independently to detect anomalies, the Mahalanobis distance [46] is preferred over the Euclidean distance. The Mahalanobis distance is defined as the following, where $\hat{\mu}$ is the estimated mean vector and $\hat{\Sigma}$ is the estimated covariance matrix.

$$s_t = \sqrt{(X_t - \hat{\mu})^T \hat{\Sigma}^{-1} (X_t - \hat{\mu})}$$

To ensure a good atypicity score, the estimator of the covariance has to be robust and efficient, as shown in Section 4. Inspired by the results of Section 5.3.2, the biweight-midcovariance [47] is used to estimate each coefficient of the matrix $\hat{\Sigma}$.

Results and analysis. The result is represented for one example in Figure 22. The multidimensional time series is represented using one plot for each dimension. The anomaly detector successfully detects the breakpoints in the dashed black lines, and the anomalies that are represented by green dots with few false positives.

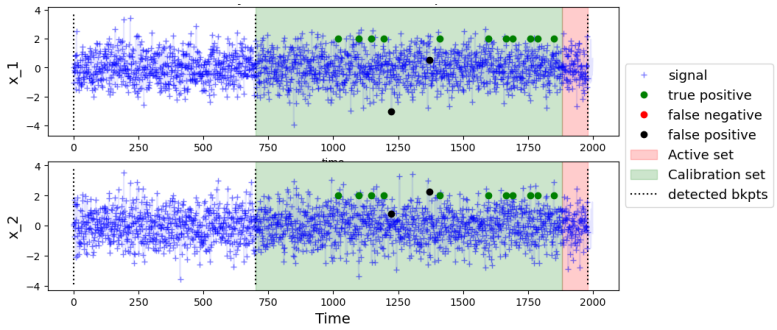


Figure 22: Application of our anomaly detector on 2D Gaussian time series having breakpoints in the covariance.

The anomaly detector has been applied to 50 time series and the results are summarized in Table 3. The FNR is close to 0 and the FDR is smaller than expected, 0.12 instead of 0.2. This confirms that the detector can be applied to multidimensional data with minor adaptation.

α	FDR	FNR
0.2	0.126	0.054

Table 3: FNR and FDR for anomaly detection on 2D Gaussian time series with breakpoint in the covariance.

8.2.4. Gaussian data with breakpoints in the mean and in the variance

In the experiments conducted so far, all scenarios considered homoscedastic time series where the change is in the mean while the variance is constant between two segments. In this section, the case of heteroscedasticity in time series is studied where the variance changes between two segments. Therefore, time series will have parts where the variance is very low and parts where it is very high. The struggle is that a kernel may be good at detecting breakpoints in a low variance context, but have difficulty when the variance is high, and vice versa. Therefore, several kernels are tested by varying the bandwidth size.

Experiment description.. Let's consider a time series generation process and an anomaly detector described in Section 8.1. To adapt to the heteroscedasticity hypothesis, the transition rule is modified so that at each breakpoint the variance changes as follows, where Δ_σ is the variance shift size equal to 2:

$$\sigma_{i+1} = \exp(\zeta_{\sigma,i} \ln \Delta_\sigma) * \sigma_i$$

To ensure that the variance covers a wide range of values, the variable ζ_i is chosen asymmetric. In the case of this experiment, ζ_i has a probability of 0.9 of being $+1$. Thus, the variance is more likely to increase than to decrease at each breakpoint. To ensure the visibility of the breakpoint in the mean to any variance, the size of the shift in the mean needs to be proportional to the maximum of the variance of the segment before and after the breakpoint, as described in the following:, where Δ_μ is the mean shift size equal to 2:

$$\mu_{i+1} = \zeta_{\mu,i+1} \Delta_\mu \max(\sigma_i, \sigma_{i+1}) + \mu_i$$

According to the median heuristic, breakpoints are easily detected by a Gaussian kernel when the standard deviation of the data is of the same order as the bandwidth h. Several kernels are tested:

- Gaussian kernel with bandwidth $h = 1$. This kernel with a small bandwidth is relevant to detect breakpoints when the variance of the time series is small, but may fail when the variance is high.
- Gaussian kernel with bandwidth $h = 100$. In this situation, the kernel is more relevant to detect breakpoints when the variance is high.
- To consider both scenarios, where breakpoints appear in some parts of the series with high variance and in parts of the series with low variance, a linear combination of the two Gaussian kernels may be a good response. This kernel is characteristic as a sum of two characteristic kernels and is defined by:

$$k(x, y) = 0.5k_{h_1}(x, y) + 0.5k_{h_2}(x, y) \tag{26}$$

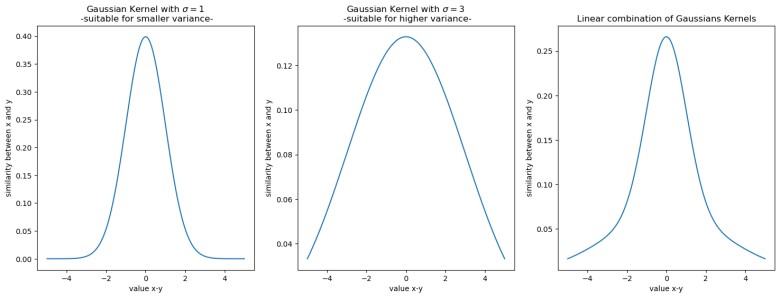


Figure 23: Illustration of the different kernels

Result analysis. The anomaly detector is applied three times to the same time series, changing only the kernel used in Figures 24, 25 and 26:

- Figure 24 illustrates the result using the Gaussian kernel with small bandwidth, $h = 1$. The breakpoint was not detected at ①, which leads to a false negative ② and a large number of false positives at ③.
- Figure 25 illustrates the result using the Gaussian kernel with large bandwidth, $h = 100$. At the position ①, the breakpoint with low variance is not detected. It leads to false positives at ② because data with different variances belong to the same calibration set.
- Figure 26 illustrates the result when using the linear combination of the two Gaussian kernels. All breakpoints are detected, reducing the number of false positives and false negatives.

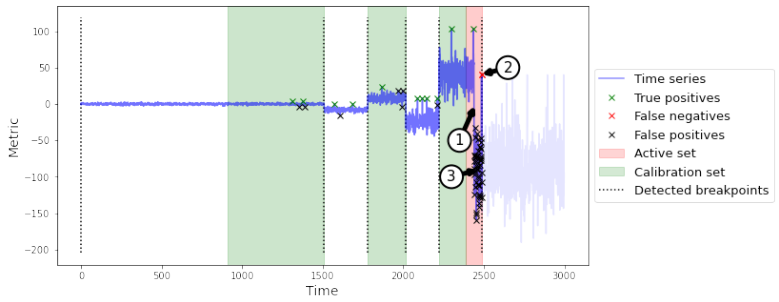


Figure 24: Application of our anomaly detector on Gaussian time series having breakpoints in the mean and in the variance, using a Gaussian kernel having a small bandwidth.

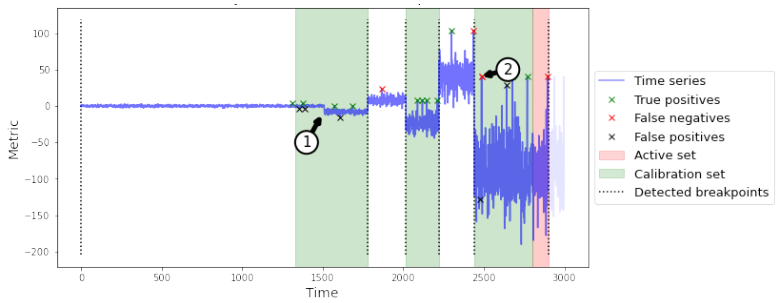


Figure 25: Application of our anomaly detector on Gaussian time series having breakpoints in the mean and in the variance, using a Gaussian kernel having a large bandwidth.

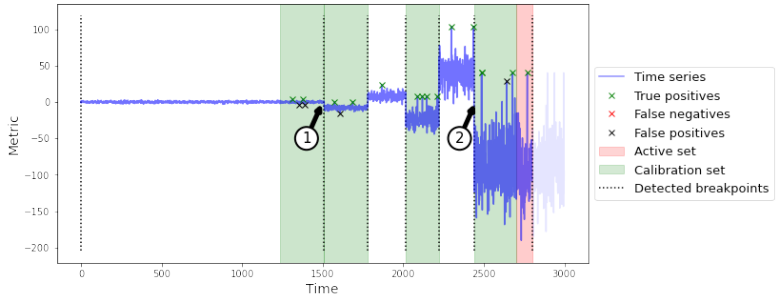


Figure 26: Application of our anomaly detector on Gaussian time series having breakpoints in the mean and in the variance, using a linear combination of two Gaussian kernels.

The anomaly detector has been applied to 50 time series and the FDR and FNR results are summarized in Table 4. Different kernels, bandwidth h, are considered in combination with α levels in $\{0.1, 0.2\}$:

- Gaussian kernel (labeled Gaussianh) with bandwidth h in $\{1, 10, 100\}$
- Linear combination of two Gaussians (labeled CombG1G100)

The performances are strongly influenced by the kernel bandwidth: The FNR is lower when using the Gaussian kernel with bandwidth $h = 1$ or using the combination of Gaussian kernels while it is high when using kernels with larger bandwidth. The FDR is slightly higher than expected α for all tested kernels. However, the FDR is smaller when using the combination of Gaussians compared to the Gaussian kernel with $h = 1$. Thus, anomaly detection remains possible when the variance of the time series changes under heteroscedasticity. However, there is no general way to build a dedicated kernel that responds to this scenario, but combining specialized kernels to adapt to the different regimes of the time series seems to be a promising approach.

α	Kernel	FDR	FNR
0.10	Gaussian1	0.188	0.054
	Gaussian10	0.127	0.136
	Gaussian100	0.148	0.456
	CombG1G100	0.134	0.057
0.20	Gaussian1	0.323	0.017
	Gaussian10	0.232	0.102
	Gaussian100	0.232	0.397
	CombG1G100	0.253	0.018

Table 4: FDR and FNR for anomaly detection on Gaussian time series with breakpoints in the mean and in the variance according to the α level and the chosen kernel

8.2.5. Gaussian data with breakpoints in the variance

In this section, the more challenging scenario of time series with changes in variance without a shift in mean is addressed.

Description of the experiment. To generate the data, a Gaussian distribution is used as the reference one. The breakpoints in the variance are generated according to the rule described in Eq. 23. Since the variance of the time series changes along the time series, it may be difficult to detect all the breakpoints with the same kernel. To evaluate the detector in this scenario, it is based on the same kernels defined in Section 8.2.4 and on the z-score atypicity function.

Results and analysis. Figures 27 and 28 show two examples of anomaly detection. In Figure 27, all the breakpoints are successfully detected, allowing correct anomaly detection with few false positives. In Figure 28, the procedure fails and no breakpoint is detected in ①. After the change with higher variance, all data are considered as anomalies. It is an evidence that the efficiency of the anomaly detector is strongly influenced by its ability to detect the true breakpoints.

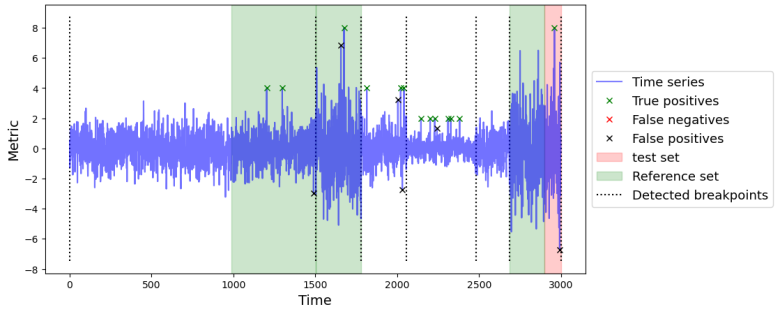


Figure 27: Example of successful anomaly detection on time series with breakpoints in the variance.

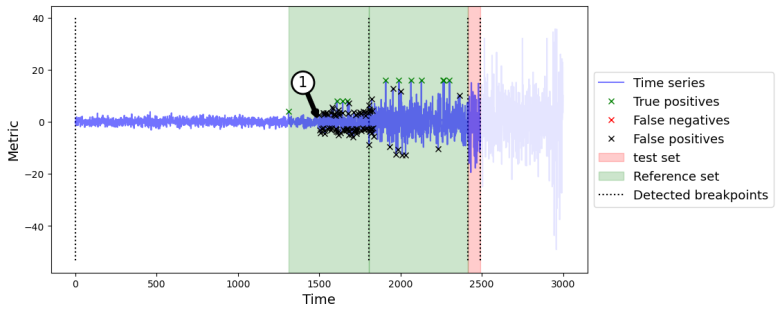


Figure 28: Examples of failure of anomaly detection on time series with change in the variance.

Table 5 summarizes the FDR and FNR results obtained for 50 time series using the same kernels and α levels as in Table 4. In all cases, the anomaly detection shows a poor accuracy, since on one side there is a lack of control of the FDR with respect to the target value alpha, and on the other side the FNR is very high. However, the best FNR and FDR values are obtained for the combination of Gaussian kernels, which allows better detection of breakpoints in the variance.

α	Kernel	FDR	FNR
0.10	Gaussian1	0.272	0.321
	Gaussian10	0.806	0.712
	Gaussian100	0.835	0.599
	CombG1G100	0.229	0.298
0.20	Gaussian1	0.313	0.241
	Gaussian10	0.649	0.511
	Gaussian100	0.685	0.396
	CombG1G100	0.282	0.225

Table 5: FDR and FNR for anomaly detection on Gaussian time series with breakpoint in the variance according α level and chosen kernel.

These results show how challenging the case of time series with breakpoints in the variance is. Indeed, the change in the variance is much harder to detect than the shift in the mean presented in Section 8.2.1. One approach is to carefully tune the kernel by choosing the right combination of kernels to enable the detection of specific types of breakpoints.

8.3. How hyperparameter choices affect the the anomaly detector performances?

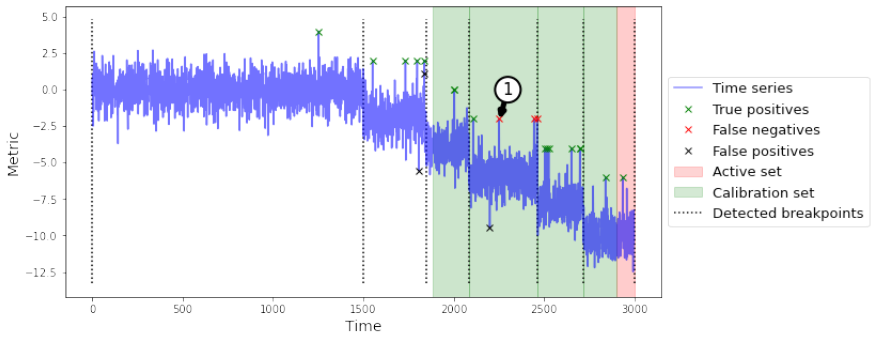
The goal of this section is to show how incorrect hyperparameter values of the anomaly detector lead to a degradation of the anomaly detector's performances. This evaluation is done for three core components: the variance estimator, the cardinality of the calibration set, and the cardinality of the active set. The hyperparameters of these components are intentionally set very far from the recommendations given in Sections 4 and 5 and the consequences are observed and discussed to confirm the recommendations, the rules and the analyses stated in the paper.

8.3.1. Bad choice of variance of segments estimator

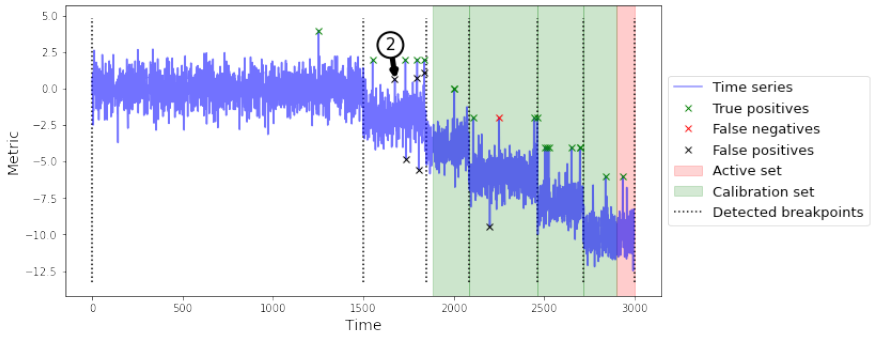
In Section 4, it was established that a good atypicity score should respect two properties: robustness to the presence of anomalies in the training set and efficiency. In this scenario, a bad

choice is made for an atypicity score that does not respect the requirements of being robust and efficient. To simulate this case and evaluate the effects, the experiment with Gaussian time series having breakpoint in the mean introduced in Section 8.2.1 is reused by replacing the biweight estimator of the variance in the z-score function by the MLE estimator or the MAD estimator. Indeed, MLE estimator is efficient but not robust, and the MAD estimator is robust but not efficient while the biweight midvariance is robust and efficient.

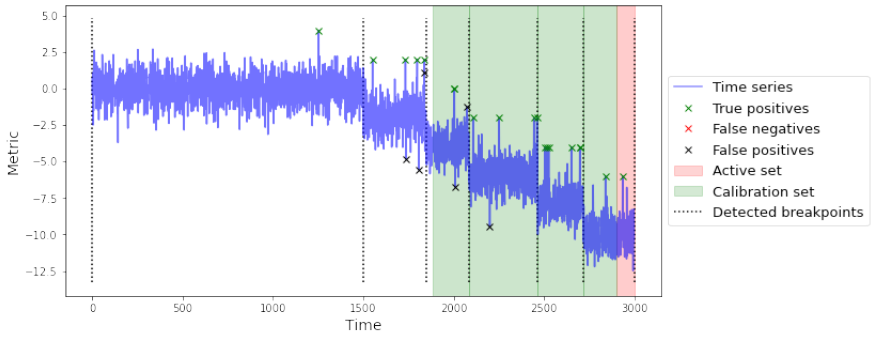
Result and analysis. To analyze and compare the effect of the different estimators, the same example is considered in Figure 29, for different variance estimators. Since the MLE estimator is not robust, Figure 29a at ① shows false negatives due to variance overestimation caused by the presence of anomalies in the current segment. The choice of the robust MAD estimator reduces the false negatives while it generates a higher number of false positives as shown in Figure 29b at ②. The variance is underestimated due to the lack of data points and the lower efficiency of MAD. The Biweight estimator is advantageous as it is both robust and efficient and is able to reduce false positives and false negatives, as shown in Figure 4.1.



(a) MLE



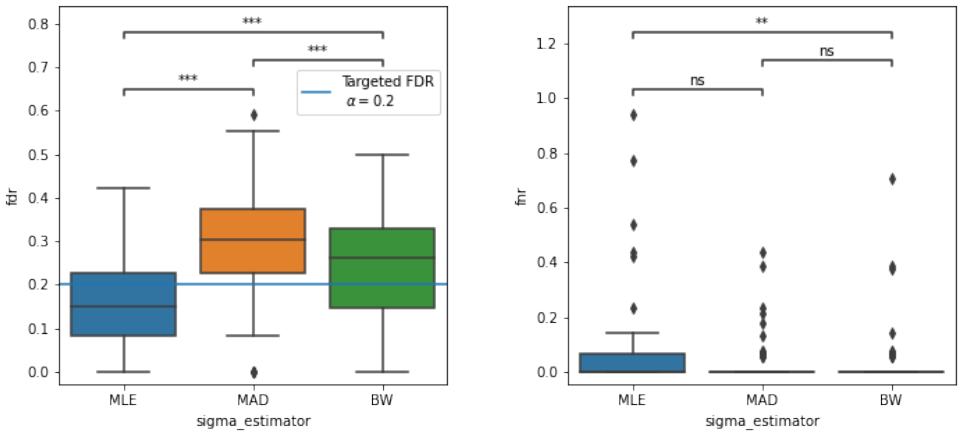
(b) MAD



(c) BW

Figure 29: Application of our anomaly detector on Gaussian time series having breakpoints in the mean, using different variance estimators.

In Figure 30, the boxplots represent the distribution of FNR and the FDR over a set of 50 time series based on the standard deviation estimator (MLE, MAD or BW). Paired permutation tests [48] are used to compare the performances of two estimators. For each pair of estimators, the hypothesis tested is: "The mean FDR (or FNR) is the same using these two variance estimators". The results are represented by adding a symbol ("ns" the difference is not significant, "*" significance at 5%, "**" significance at 1%, "***" significance at 0.1%) between the two tested estimators.



(a) FDR

(b) FNR

Figure 30: Boxplots of the FNR and FDR according to the choice of the variance estimator.

The FDR and FNR results are summarized in Table 6. The FNR is significantly higher when the MLE variance estimator is used compared to the more robust MAD and biweight midvariance estimators, which have close performances. However, the FDR is significantly higher when the MAD is used compared to the biweight midvariance estimator, for which the FDR is better controlled.

sigma_estimator	FDR	FNR
MLE	0.16	0.08
MAD	0.29	0.04
BW	0.24	0.04

Table 6: FNR and FDR according to the choice of the variance estimator.

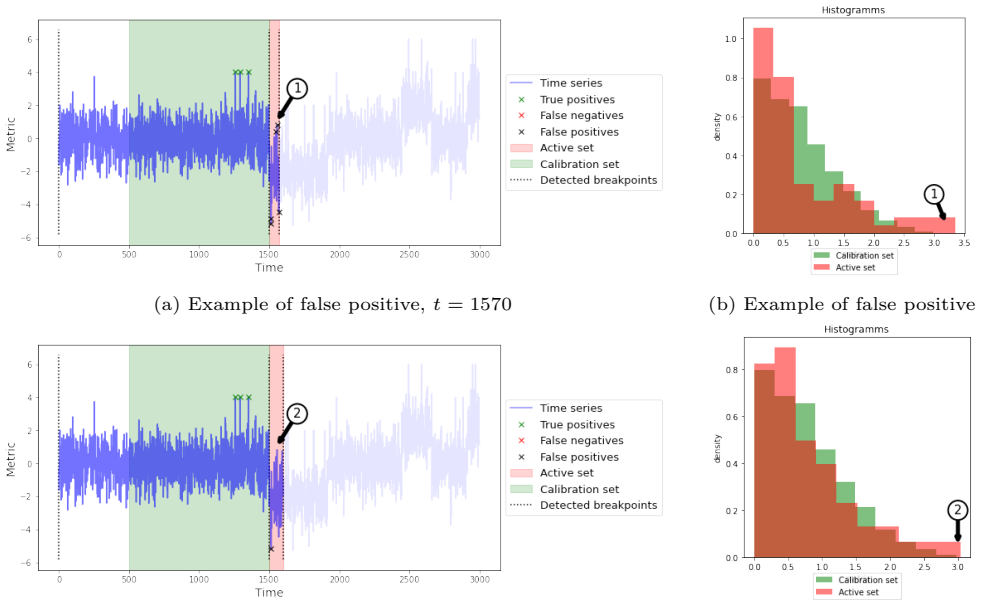
8.3.2. Bad choice for active set cardinality

Section 5 introduced the notion of an active set to deal with the uncertainty of status. It also provides rules to compute the cardinality of the calibration test. In this section the relevance of this rule is evaluated.

Description of the experiment. To evaluate the performance degradation due to a bad choice of the active set cardinality, the experiment framework introduced in Section 8.2.1 is reused. According to the results of the experiments in Section 5.3.2 and Section 5.4.2, status can be ensured with strong confidence with an active set cardinality equal to $m = 100$. For each time series generated, two anomaly detectors are applied, one with an active set cardinality equal to 100 and the second with an active set cardinality equal to 10.

Results and analysis. In order to understand how the active set improves the anomaly detector, the results are observed at two different instants: at time $t = 1570$ in Figures 31a and 31b, and at time $t = 1600$ in Figures 31c and 31d. The histograms of the z-scores of the calibration

set in green and the active set in red are shown in Figures 31b and 31d. At time $t = 1570$, the new current segment contains few points, resulting in a variance estimation error and an overestimation of the z-score of the active set in ① Figure 31b and false positives in ① Figure 31a. At time $t = 1600$, the segment has acquired new data points, the variance estimate is improved and the z-score is not overestimated in ② Figure 31d. The number of false positives is reduced in ② Figure 31c. The status of the data point at $t = 1570$ is corrected at time $t = 1600$ because the active set is large enough, otherwise its status would be fixed to the wrong one.



(a) Example of false positive, $t = 1570$

(b) Example of false positive

(c) Example of accurate detection, $t = 1600$

(d) Example of false negative

Figure 31: Abnormality status update after new data points acquisition in the current segment.

Figure 32 illustrates the boxplots of the FDR distribution according to the active set cardinality. The results, summarized in Table 7, show that the FDR is significantly higher when the active set has a cardinality of $m = 10$. On the contrary, using a cardinality of $m = 100$ allows to control the FDR at the desired level $\alpha = 0.2$. This experiment illustrates the benefits of following the recommendations in Section 5 to improve anomaly detection performances.

α	m	FDR	FNR
0.2	10	0.529	0.006
	100	0.186	0.053

Table 7: FDR and FNR mean according to the active set cardinality.

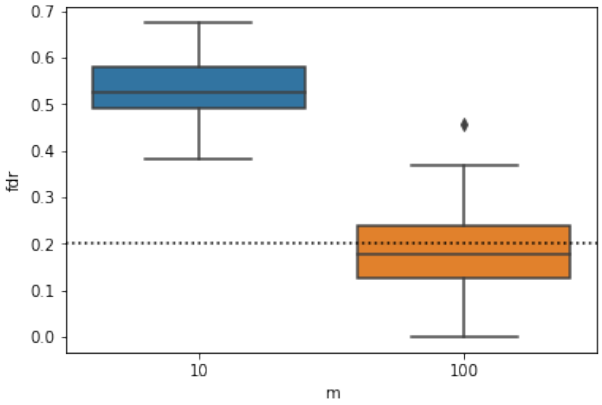


Figure 32: FDR boxplots according to the active set cardinality

8.3.3. Bad choice of cardinality for the calibration set

It was established in Section 7 that the FDR can only be controlled if the cardinality of the calibration set takes specific values. This section verifies this claim in the case of breakpoint based anomaly detection.

Description of the experiment. To evaluate the degradation of the FDR control due to a bad choice of the calibration set cardinality, time series are generated according to the framework design introduced in Section 8.2.1. For each generated time series, with the target FDR $\alpha = 0.2$ (resp. $\alpha = 0.1$) and the active set cardinality $m = 100$, two anomaly detectors are applied: one with a calibration set cardinality equal to 999 (resp. 1999) respecting the recommendation. The second with a calibration set cardinality equal to 1000 (resp. 2000), not respecting the recommendation. Indeed, since the proportion of anomalies is equal to $\pi = 0.01$, the goal of a FDR equal to $\alpha = 0.2$ (resp. $\alpha = 0.1$) can be achieved using Benjamini-Hochberg with $\alpha' = 0.1$ (resp. 0.05) according to Section 7. The calibration set cardinality should then be equal to 999 (resp. 1999) according to Eq. 21.

Results and analysis. The results in Table 8 show that the FDR is controlled at the desired level for $n = 999$ and $n = 1999$, while the FNR is higher. This confirms that the FDR can only be controlled by selecting the parameter n using the rule in Section 7. To reduce the FNR while maintaining control of the FDR, the values n must be chosen among the values $\{1999, 2999, 3999, \ldots\}$ as discussed in the paper [26].

α	n	FDR	FNR
0.2	999	0.21	0.030
	1000	0.30	0.0
0.1	1999	0.1	0.1
	2000	0.16	0.03

Table 8: FDR and FNR according to the calibration set cardinality.

8.4. Diagnose the causes of underperformance

Our Breakpoint based anomaly detector has been tested on different time series data in Section 8.2, it shows good performances to ensure low FNR with an FDR almost controlled in different cases. However, the FDR is never completely under control, and is always slightly higher than expected. This section examines why this lack of complete control of the FDR occurs by replacing some estimators with knowledge of the true values and evaluate the effect on the FDR.

8.4.1. Description of the experiment

The BKAD is applied to the synthetic time series, where some estimators are replaced by true knowledge, called oracle version. Three estimators are chosen to be replaced by their oracle versions:

- The breakpoint estimator: can be replaced by the true breakpoint position,
- The mean and standard deviation estimators: can be replaced by their true values,
- The anomaly removed: As described in Section 6 when building the calibration set, estimated anomalies are removed to avoid biasing the estimation of the p-values. The oracle version of this is to remove the true anomalies.

Using the framework from Section 8.1, five anomaly detectors are applied to each time series. Multiple combinations of the true knowledge (marked "O") versus estimated values (marked "E") are used to produce different versions of anomaly detectors described in Table 9. As an example, for detector 3, the breakpoints and anomalies in the calibration set are detected using their true values, but the segment mean and variance parameters are estimated.

Detector	Breakpoint	Mean and variance	Anomaly Removing
Detector 1	O	O	O
Detector 2	O	O	E
Detector 3	O	E	O
Detector 4	E	E	O
Detector 5	E	E	E

Table 9: Description of the different detectors.

The significance of the results is checked using permutation tests. It is possible that the cause of this underperformance depends on the data distribution or on breakpoint types. Different probability distributions are tested with different kinds of shifts.

8.4.2. Results and analysis

The complete empirical results can be found in Appendix A. The performances of the different detectors are evaluated on a different laws generating the time series (Student, Gaussian, Mixture of Gaussians noted MoG). The FDR and FNR distributions are represented by a boxplot with the significance differentiating two detectors ("ns" the difference is not significant, "*" significance at 5%, "**" significance at 1%, "***" significance at 0.1%).

In the following paragraphs, the effects of the various core components are studied: breakpoint detector, mean and variance segment estimator and anomalies removed from the calibration set.

Breakpoint Estimation. Table 10 shows the performance of anomaly detectors 3 and 4 (see Table 9) for different types of data and shifts. The only difference between the two detectors is that Detector 3 uses a breakpoint detector while Detector 4 has knowledge of true breakpoints. The bold values highlight the cases where the difference between the two estimators is significant. Table 10 illustrates that the breakpoint estimation does not strongly affect the FDR performance except in the case where breakpoints occur in the variance. This is expected since breakpoints in the variance are more difficult to to detect, as discussed earlier in Section 8.2.5. FNR increases in few cases where the breakpoint positions are estimated.

Type of shift	law	α	Breakpoints	FDR	FNR
Mean	Gaussian	0.10	E	0.104	0.105
			O	0.100	0.091
		0.20	E	0.182	0.054
			O	0.176	0.054
	Student	0.10	E	0.119	0.066
			O	0.117	0.065
		0.20	E	0.199	0.033
			O	0.198	0.032
	MoG	0.10	E	0.113	0.131
			O	0.108	0.124
		0.20	E	0.186	0.072
			O	0.190	0.071
Mean and var.	Gaussian	0.10	E	**0.114**	0.090
			O	0.099	0.078
		0.20	E	**0.188**	0.051
			O	0.167	0.040
Variance	Gaussian	0.10	E	**0.200**	**0.214**
			O	0.110	0.109
		0.20	E	**0.257**	**0.128**
			O	0.174	0.062

Table 10: Anomaly detector performances with and without knowledge of true breakpoint positions, according different time series.

Segment mean and variance parameters. Table 11 shows the performance of anomaly detectors 1 and 3 (see Table 9) for different types of data and shifts. The only difference between the two detectors is that Detector 3 estimates the mean and the variance parameters of the segments while Detector 1 has knowledge of the true parameters. According to Table 11, the estimators do not strongly affect the FDR of the anomaly detector. There are few significant differences, displayed in bold, which are smaller than in Table 10.

type of shift	law	α	Mean and variance	FDR	FNR
Mean	Gaussian	0.10	E	0.082	0.043
			O	0.105	0.000
		0.20	E	0.167	0.000
			O	0.188	0.000
	Student	0.10	E	0.117	0.065
			O	0.113	0.056
		0.20	E	0.198	0.032
			O	0.196	0.026
Mean and var.	Gaussian	0.10	E	**0.091**	0.000
			O	0.107	0.000
		0.20	E	**0.167**	0.000
			O	0.200	0.000

Table 11: Anomaly detector performances with knowledge of the true segment mean and standard deviation values and with estimation of these parameters, according different time series.

Anomalies Removing. Table 12 represents the results for different detectors considering different laws, alpha levels and kind of shift. The four detectors are chosen to identify the effect of removing detected anomalies from the calibration set instead of removing the true anomalies, in case other components are estimators and in case other components are oracles. Note that for Gaussian Mixture (MoG), the "Mean and Variance" component is marked with a "X", since the kNN atypicity score does not use mean and variance parameters. It is clear that the control of the FDR is worse when the calibration set is built based on detected anomalies. Indeed, the false positives and false negatives detected at time t will badly affect the detection at time $t+1$. Despite the fact that a robust score is chosen, these observations lead to a conclusion that the p-value estimator is sensitive to:

- False negatives: If there is a missed anomaly in the calibration set, the p-values of all data points in the active set will be underestimated. This situation leads to generate more false negatives, which will confound the calibration sets of subsequent instants.
- False positives: The p-value estimator is also sensitive to false positives due to the way the calibration set is constructed. As a reminder, detected anomalies are replaced by a random points belonging to a segment similar to the current segment. The problem arises when an anomaly is falsely detected. Generally speaking a false positive is a point with a high score. When a false positive is replaced with a random point, its score will be statistically lower. Thus, removing the false positives from the calibration set reduces the average score in the calibration set and consequently reduces the p-values of the data points in the active set. This leads to more false positives, which will affect the construction of calibration sets at later times.

Type of shift	Law	α	Breakpoint	Mean and variance	**Anomaly removing**	FDR	FNR
Mean	Gaussian	0.1	E	E	E	**0.134**	0.123
			E	E	O	0.104	0.105
			O	O	E	**0.165**	0.041
			O	O	O	0.121	0.048
		0.2	E	E	E	**0.242**	0.039
			E	E	O	0.182	0.054
			O	O	E	**0.301**	0.018
			O	O	O	0.197	0.018
	Student	0.1	E	E	E	**0.158**	0.059
			E	E	O	0.119	0.066
			O	O	E	**0.154**	0.035
			O	O	O	0.113	0.056
		0.2	E	E	E	**0.289**	0.026
			E	E	O	0.199	0.033
			O	O	E	**0.301**	0.013
			O	O	O	0.196	0.026
	MoG	0.1	E	X	E	0.118	0.246
			E	X	O	0.113	0.131
			O	X	E	0.103	0.294
			O	X	O	0.108	0.124
		0.2	E	X	E	0.202	0.137
			E	X	O	0.186	0.072
			O	X	E	0.221	0.111
			O	X	O	0.190	0.071
Mean and var.	Gaussian	0.1	E	E	E	0.134	0.057
			E	E	O	0.114	0.090
			O	O	E	**0.955**	0.022
			O	O	O	0.119	0.054
		0.2	E	E	E	**0.253**	0.018
			E	E	O	0.188	0.051
			O	O	E	**0.961**	0.021
			O	O	O	0.205	0.029

Table 12: Anomaly detector performances with and without knowledge of true anomalies for removing anomalies, according different time series.

Conclusion. The conclusion of this analysis is that most of the underperformance relative to the ideal case, such as higher than expected FDR, is explained by the non-robustness of the empirical p-value estimator and the contamination of the calibration set by false negatives and false positives.

8.5. Evaluation against competitors

After studying the conditions that must be met to ensure high detection performance and control of the FDR in the previous sections, the breakpoint detection based anomaly detector (BKAD) proposed in this paper is compared to alternative anomaly detectors from the literature on different data collections. The goal is to determine if and under which conditions the new anomaly detector can improve the state of the art.

8.5.1. Methods

BKAD is evaluated against state-of-the-art anomaly detectors presented in the review [49]. The most representative unsupervised anomaly detectors for univariate time series data are selected. The implementation of [49] is used, with default hyperparameters. The detectors selected are those that theoretically capable of detecting anomalies in piecewise iid data. These algorithms fall into two categories: the one that build a context such as a segment, a sliding window or a cluster, and on the other that use subseries instead of single points. On the other hand, predictive or regression models are of little interest on piecewise iid data.

Median [50]. A sliding windows is used to estimate the median and dispersion parameter of last observations. The atypicity score used is the z-score. The main difference with the BKAD approach is the use of sliding windows instead of using a breakpoint detector to define the segments.

CBLOF [51]. Cluster based local outlier factor identifies the cluster to which individual points belong, then it computes the local outlier factor associated with that cluster. The use of clusters is similar to that of breakpoints in that it attempts to group similar points together, but has no temporal notion.

Sub. IF [52]. The method divides the time series in subsequences and uses Isolation Forest on the subsequences set.

DWT [53]. Method based on wavelet to remove noise. Atypicity score is computed using the Gaussian distribution on the Discrete Wavelet Transform, with Haar wavelet. Anomalies can be detected as abnormal Haar coefficients.

Sub. LOF [54]. The method divides the time series in subsequences and uses Local Outlier Factor on the subsequences set.

FFT [55]. Method based on Fast Fourier Transform. It uses Local outlier factor on the Fast Fourier Transform of the subsequences. Anomalies can be detected as abnormal frequency coefficients.

8.5.2. Threshold

After applying these different methods, an atypicity score is obtained. This score is sufficient to compute the AUC metric, but does not allow detection and calculation of the FDR and FNR without thresholds. To calculate these thresholds, the method introduced in [26] is used, which guarantees FDR control at a fixed α level in case the time series of scores is iid. The threshold of BKAD is chosen as described in Section 7. Here α is set to 0.2 for all detectors and time series.

8.5.3. Data

To ensure a comprehensive analysis, different kind of time series data are considered:

- Time series with breakpoints
- Time series with seasonality
- Residual from time-series with seasonality
- Real data time series

Time series with breakpoints. The time series with breakpoints are generated according to the experimental design presented in Section 8.1 with the following hyperparameters: the reference distribution is Gaussian $\mathcal{P}_{0,1} = \mathcal{N}(0, 1)$ and all anomalies follow the law of 4ζ, where zeta follows the Rademacher distribution. Anomalies are generated with a proportion of $\pi = 0.01$. The breakpoint positions are generated according to the Poisson process with an average segment length of 125. To avoid having too few segments, breakpoints are removed if a segment has less than 100 points. For the benchmark *breakpoint-mean*, breakpoints occur in the mean with a $\Delta = 2$. And, for the benchmark *breakpoint-var*, breakpoints occur in the variance with $\Delta = 1.5$.

Time series with seasonality. To study how the anomaly detector behaves on time series not following the statistical model introduced in Section 2.1, time series with seasonality and trend are considered.

Let the following components be given:

1. $R_t \sim \mathcal{N}(0, \sigma)$, the residual, $\sigma = 1$
2. $A_t \sim B(\pi)$ the abnormality variable, $\pi = 0.01$
3. $S_{1,t} = A_1 \sin(2\pi f_1 t)$ the seasonality with long period, where the amplitude A_1 and the frequency f_1 are random variables, $A_1 \in \{1, 3, 5\}$ and $f_1 \in \{5, 10, 20\}$
4. $S_{2,t} = a_{21} A_1 \sin(2\pi w_{21} f_1 t)$ the seasonality with short period, where the frequency multiple w_{21} and the amplitude attenuation are random variable, $a_{21} \in \{0.5, 0.3, 0.1\}$ and $w_{21} \in \{2, 3, 5\}$
5. $\sigma_t = \sin(t) + 1.5$ the seasonal variance
6. $T_t = Bt$ the linear trend

The following collections are generated:

1. *simple-seasonality*: $X_t = S_{1,t} + (1 - A_t)R_t + A_t\zeta_t\Delta'$
2. *complex-seasonality*: $X_t = S_{1,t} + S_{2,t} + (1 - A_t)R_t + A_t\zeta_t\Delta'$
3. *variance-seasonality*: $X_t = ((1 - A_t)R_t + A_t\zeta_t\Delta')\sigma_t$
4. *trend-seasonality*: $X_t = T_t + S_t + (1 - A_t)R_t + A_t\zeta_t\Delta'$

Residual from time series with seasonality. In practical applications, to simplify the detection of anomalies, seasonality and other predictable patterns are removed during a preprocessing step. To evaluate how the anomaly detector performances are affected by this preprocessing step, a new benchmark is built from the residuals extracted for each time series in the "Time series with seasonality" benchmark. In this experiment, the residual is extracted by removing the trend and the seasonality using the "seasonal_decompose" function from the Python library *statsmodels*.

Time series from real data:. The anomaly detectors are evaluated on various time series datasets coming from different sources. The Numenta Anomaly Benchmark (NAB) from [56], the dodger dataset from the UCI at [57] and Mars Science Laboratory (MSL) and Soil Moisture Active Passive (SMAP) provided by NASA in [58] are used to build the complete benchmark.

8.5.4. Metrics

To measure the performances of different anomaly detectors, two metrics are reported: The Area Under Curve (AUC)[59, 60] in Table 13 and the FDR/FNR in Table 14. The advantage of the Area Under the Curve (AUC) is to be able to evaluate the anomaly detector without evaluating the threshold selection method. However, this can also be a limitation for real use, since a threshold is needed for practical applications. To determine the ability of anomaly detectors to control the false positive rate to a desired level while keeping the false negative rate low, the FDR and FNR metrics are reported. The disadvantage of these metrics is that it can be difficult to compare two detectors if one performs better on FDR and the other on FNR. Furthermore, they only take into account values for a single threshold, which have to be precised for detectors that return only an atypicity score. The threshold policy used for this experiment is the one implemented in [26], as stated in Section 8.5.2.

8.5.5. Results and analysis

The results are summarized in two tables. Table 13 represents the AUC metric according to benchmarks and anomaly detectors and Table 14 represents the FDR and FNR metrics.

Benchmark	BKAD(Ours)	Median	Sub. IF	DWT	Sub. LOF	LOF	VALMOD	CBLOF	FFT
Breakpoint in mean	**1.00**	0.95	0.64	0.61	0.65	0.70	0.42	0.80	0.83
Breakpoint in variance	**0.98**	0.89	0.52	0.54	0.60	0.56	0.36	0.78	0.16
Simple seasonality	0.88	**0.98**	0.56	0.57	0.71	0.68	0.47	0.73	0.72
Complex seasonality	0.94	**0.98**	0.57	0.55	0.72	0.79	0.45	0.85	0.64
Seasonality in variance	**1.00**	0.99	0.54	0.57	0.56	0.87	0.43	0.92	0.71
Seasonality and trend	0.88	**0.98**	0.53	0.57	0.71	0.63	0.47	0.67	0.72
Res. simple seasonality	**0.99**	0.98	0.62	0.57	0.69	0.92	0.47	0.99	0.89
Res. complex seasonality	**1.00**	0.99	0.63	0.56	0.71	0.94	0.45	1.00	0.91
Res. seasonality and trend	**0.99**	0.98	0.61	0.56	0.69	0.93	0.47	0.99	0.89
DODGER	0.56	0.30	**0.67**	0.65	0.54	0.51	0.41	0.48	0.30
NAB	0.57	0.45	0.66	**0.73**	0.67	0.48	0.47	0.54	0.20
NASA-MSL	0.57	0.56	**0.84**	0.81	0.61	0.56	0.48	0.68	0.56
NASA-SMAP	0.60	0.39	0.83	**0.90**	0.69	0.51	0.61	0.61	0.47

Table 13: AUC metric according to the anomaly detectors on benchmarks.

	BKAD(Ours)		Median		LOF		CBLOF		Sub. LOF		Sub. IF		DWT		FFT	
Benchmark	FDR	FNR	FDR	FNR	FDR	FNR	FDR	FNR	FDR	FNR	FDR	FNR	FDR	FNR	FDR	FNR
Breakpoint in mean	0.25	0.04	0.07	0.48	0.52	0.70	0.83	0.52	0.99	0.81	0.99	0.44	0.99	0.01	0.93	0.46
Breakpoint in variance	0.36	0.17	0.19	0.36	0.72	0.71	0.72	0.37	0.95	0.54	0.91	0.58	0.93	0.28	0.89	0.27
Simple seasonality	0.34	0.47	0.21	0.45	0.48	0.76	0.62	0.72	0.99	0.73	0.99	0.96	0.97	0.30	0.92	0.58
Complex seasonality	0.27	0.44	0.19	0.44	0.37	0.64	0.41	0.64	0.99	0.68	0.99	0.95	0.95	0.43	0.95	0.65
Seasonality and trend	0.50	0.46	0.22	0.45	0.58	0.86	0.77	0.78	0.99	0.77	0.79	0.89	0.97	0.11	0.92	0.57
Seasonality in variance	0.34	0.35	0.10	0.23	0.33	0.50	0.32	0.48	0.99	0.86	0.95	0.94	0.95	0.20	0.93	0.63
Res. simple seasonality	0.26	0.42	0.19	0.56	0.39	0.70	0.25	0.41	0.99	0.77	0.99	0.94	0.95	0.35	0.93	0.61
Res. complex seasonality	0.17	0.15	0.12	0.31	0.58	0.80	0.21	0.14	0.99	0.72	0.99	0.92	0.97	0.32	0.93	0.45
Res. seasonality and trend	0.26	0.43	0.20	0.57	0.44	0.73	0.29	0.41	0.99	0.79	0.97	0.94	0.95	0.36	0.92	0.64
dodger	0.41	0.66	0.79	0.99	0.89	0.09	0.97	1.00	0.71	0.92	0.39	0.91	0.70	0.55	0.89	0.09
NAB	0.61	0.91	0.62	0.85	0.67	0.58	0.50	0.82	0.64	0.74	0.55	0.62	0.77	0.27	0.87	0.25
NASA-MSL	0.78	0.91	0.62	0.84	0.71	0.72	0.45	0.72	0.62	0.82	0.49	0.70	0.65	0.42	0.68	0.49
NASA-SMAP	0.69	0.92	0.76	0.63	0.80	0.27	0.70	0.52	0.75	0.64	0.65	0.44	0.83	0.05	0.80	0.33

Table 14: FDR and FDR metrics according to the anomaly detectors on benchmarks.

The BKAD detector gets the highest AUC scores on series with breakpoints ("Breakpoint in mean" and "Breakpoint in variance"), as shown in Table 13. It can also be seen that this detector remains efficient even when the time series contain seasonality ("simple seasonality", "complex seasonality",...). This shows the benefits of splitting the time series into simpler segments based on breakpoints, even if it does not follow the model introduced in Section 2.1. The results show the importance of preprocessing the data. Indeed, the performance of the detector increases when

it is applied to the residuals of the seasonal series instead of the original seasonal time series, as shown for "Res. simple seasonality" or "Res. complex seasonality". Nevertheless, Table 14 shows that it can be difficult to obtain control of the FDR and FNR even for the best AUC score. This illustrates that FDR control relies heavily on the (piecewise) iid hypothesis. Finally, BKAD is not very efficient on tested real data such as ("DODGER", "NAB", ...) containing anomalies which do not follow the formalism introduced in Section 2.1. The most efficient methods: "Sub. IF" and "DWT", define an atypicity score on subseries instead of data points. An interesting approach for the future might be to find a better preprocessing to apply it to real data and improve the anomaly detection.

9. Conclusion

In this paper, an online anomaly detector has been developed that detects anomalies and controls the FDR at a given level α on piecewise stationary time series. The research was conducted to address three challenges:

- Changes in the reference distribution: the changes are detected using a breakpoint detector. Anomalies are retrieved in each homogeneous segment by defining an atypicity score and a calibration set.
- Uncertainty: Due to the online nature of the detection, the abnormality status of the data points is uncertain. The notion of an active set is introduced to collect the data points that need to be re-evaluated since there status is too uncertain.
- and control of the FDR: modified Benjamini-Hochberg procedure is applied to the active set to control the FDR on the entire time series.

The result of our research is a modular anomaly detector where all core components have been studied through theoretical or empirical analysis to optimize their performance. The detector has been evaluated on a variety of scenarios to understand its strengths and limitations. It demonstrates state-of-the-art capabilities to detect anomalies on time series presenting a distribution shift. The main drawback of our method is that it relies on non-robust estimation of p-values. Also, the piecewise stationary hypothesis is often not respected in practice. Further work concerns the integration of a robust p-value estimator and the development of a preprocessing step to apply the anomaly detector to time series that are not piecewise stationary.

10. Acknowledgments

The authors would like to thank Cristian Preda, director of the MODAL team at Inria, for valuable discussions.

11. Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Worldline Company; Inria.

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 1–58.

[2] P. Notaro, J. Cardoso, M. Gerndt, A survey of aiops methods for failure management, ACM Transactions on Intelligent Systems and Technology (TIST) 12 (6) (2021) 1–45.

[3] G. Pang, J. Li, A. van den Hengel, L. Cao, T. G. Dietterich, Andea: anomaly and novelty detection, explanation, and accommodation, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4892–4893.

[4] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks, in: International conference on artificial neural networks, Springer, 2019, pp. 703–716.

[5] M. Goldstein, S. Uchida, A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, PloS one 11 (4) (2016) e0152173.

[6] A. Blázquez-García, A. Conde, U. Mori, J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Computing Surveys (CSUR) 54 (3) (2021) 1–33.

[7] S. Roberts, Control chart tests based on geometric moving averages, Technometrics 42 (1) (2000) 97–101.

[8] K. M. Carter, W. W. Streilein, Probabilistic reasoning for streaming anomaly detection, in: 2012 IEEE Statistical Signal Processing Workshop (SSP), IEEE, 2012, pp. 377–380.

[9] T. S. Buda, B. Caglayan, H. Assem, Deepad: A generic framework based on deep learning for time series anomaly detection, in: Pacific-Asia conference on knowledge discovery and data mining, Springer, 2018, pp. 577–588.

[10] M. Munir, S. A. Siddiqui, A. Dengel, S. Ahmed, Deepant: A deep learning approach for unsupervised anomaly detection in time series, Ieee Access 7 (2018) 1991–2005.

[11] B. Darkhovsky, Non-Parametric Methods in Change-Point Problems, Springer, 1993.

[12] A. Tartakovsky, I. Nikiforov, M. Basseville, Sequential analysis: Hypothesis testing and changepoint detection, CRC press, 2014.

[13] C. Truong, L. Oudre, N. Vayatis, Selective review of offline change point detection methods, Signal Processing 167 (2020) 107299.

[14] A. Celisse, G. Marot, M. Pierre-Jean, G. Rigaill, New efficient algorithms for multiple change-point detection with reproducing kernels, Computational Statistics & Data Analysis 128 (2018) 200–220.

[15] S. Arlot, A. Celisse, Z. Harchaoui, A kernel multiple change-point algorithm via model selection, Journal of machine learning research 20 (162) (2019).

[16] M. Alenezi, M. J. Reed, Denial of service detection through tcp congestion window analysis, in: World Congress on Internet Security (WorldCIS-2013), IEEE, 2013, pp. 145–150.

[17] A. T. Fisch, L. Bardwell, I. A. Eckley, Real time anomaly detection and categorisation, Statistics and Computing 32 (4) (2022) 55.

[18] M. Cvach, Monitor alarm fatigue: an integrative review, Biomedical instrumentation & technology 46 (4) (2012) 268–277.

[19] J. M. Blum, K. K. Tremper, Alarms in the intensive care unit: too much of a good thing is dangerous: is it time to add some intelligence to alarms?, Critical care medicine 38 (2) (2010) 702–703.

[20] J. M. Solet, P. R. Barach, Managing alarm fatigue in cardiac care, Progress in Pediatric Cardiology 33 (1) (2012) 85–90.

[21] K. Lewandowska, W. Mędrzycka-Dąbrowska, L. Tomaszek, M. Wujtewicz, Determining factors of alarm fatigue among nurses in intensive care units—a polish pilot study, Journal of Clinical Medicine 12 (9) (2023) 3120.

[22] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, Journal of the Royal statistical society: series B (Methodological) 57 (1) (1995) 289–300.

[23] Y. Benjamini, D. Yekutieli, The control of the false discovery rate in multiple testing under dependency, Annals of statistics (2001) 1165–1188.

[24] D. Mary, E. Roquain, Semi-supervised multiple testing, Electronic Journal of Statistics 16 (2) (2022) 4926–4981.

[25] A. Marandon, L. Lei, D. Mary, E. Roquain, Machine learning meets false discovery rate, arXiv preprint arXiv:2208.06685 (2022).

[26] E. Krönert, A. Célisse, D. Hattab, Fdr control for online anomaly detection, arXiv preprint arXiv:2312.01969 (2023).

[27] R. Laxhammar, Conformal anomaly detection: Detecting abnormal trajectories in surveillance applications, Ph.D. thesis, University of Skövde (2014).

[28] J.-P. Baudry, C. Maugis, B. Michel, Slope heuristics: overview and implementation, Statistics and Computing 22 (2012) 455–470.

[29] K. Fukumizu, A. Gretton, G. Lanckriet, B. Schölkopf, B. K. Sriperumbudur, Kernel choice and classifiability for rkhs embeddings of probability distributions, Advances in neural information processing systems 22 (2009).

[30] D. Garreau, W. Jitkrittum, M. Kanagawa, Large sample analysis of the median heuristic, arXiv preprint arXiv:1707.07269 (2017).

[31] G. Shafer, V. Vovk, A tutorial on conformal prediction., Journal of Machine Learning Research 9 (3) (2008).

[32] R. G. Staudte, S. J. Sheather, Robust estimation and testing, John Wiley & Sons, 2011.

[33] P. J. Rousseeuw, M. Hubert, Anomaly detection by robust statistics, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2) (2018) e1236.

[34] C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 665–674.

[35] N. Koyuncu, C. Kadilar, Efficient estimators for the population mean, Hacettepe Journal of Mathematics and Statistics 38 (2) (2009) 217–225.

[36] A. Gross, J. W. Tukey, The estimators of the Princeton robustness study, Department of Statistics, Univ., 1973.

[37] L. H. Shoemaker, T. P. Hettmansperger, Robust estimates and tests for the one-and two-sample scale models, Biometrika 69 (1) (1982) 47–53.

[38] R. M. Gray, R. Gray, Probability, random processes, and ergodic properties, Vol. 1, Springer, 2009.

[39] B. Bobbia, P. Doukhan, X. Fan, A review on some weak dependence conditions, HAL, preprint (2022).

[40] S. Csörgö, On the law of large numbers for the bootstrap mean, Statistics & probability letters 14 (1) (1992) 1–7.

[41] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, The Journal of Machine Learning Research 13 (1) (2012) 723–773.

[42] A. M. Safin, E. Burnaev, Conformal kernel expected similarity for anomaly detection in time-series data, Advances in Systems Science and Applications 17 (3) (2017) 22–33.

[43] V. Ishimtsev, A. Bernstein, E. Burnaev, I. Nazarov, Conformal k-nn anomaly detector for univariate data streams, in: Conformal and Probabilistic Prediction and Applications, PMLR, 2017, pp. 213–227.

[44] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distribution, Bulletin of the Calcutta Mathematical Society 35 (1943) 99–110.

[45] D. S. Matteson, N. A. James, A nonparametric approach for multiple change point analysis of multivariate data, Journal of the American Statistical Association 109 (505) (2014) 334–345.

[46] P. C. Mahalanobis, On the generalized distance in statistics, Sankhyā: The Indian Journal of Statistics, Series A (2008-) 80 (2018) S1–S7.

[47] F. Mosteller, J. W. Tukey, Data analysis and regression. a second course in statistics, Addison-Wesley series in behavioral science: quantitative methods (1977).

[48] R. A. Fisher, et al., The design of experiments., no. 7th Ed, Oliver and Boyd. London and Edinburgh, 1960.

[49] S. Schmidl, P. Wenig, T. Papenbrock, Anomaly detection in time series: a comprehensive evaluation, Proceedings of the VLDB Endowment 15 (9) (2022) 1779–1797.

[50] S. Basu, M. Meckesheimer, Automatic outlier detection for time series: an application to sensor data, Knowledge and Information Systems 11 (2007) 137–154.

[51] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern recognition letters 24 (9-10) (2003) 1641–1650.

[52] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[53] M. Thill, W. Konen, T. Bäck, Time series anomaly detection with discrete wavelet transforms and maximum likelihood estimation, in: Intern. Conference on Time Series (ITISE), Vol. 2, 2017, pp. 11–23.

[54] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[55] F. Rasheed, P. Peng, R. Alhajj, J. Rokne, Fourier transform based spatial outlier mining, in: Intelligent Data Engineering and Automated Learning-IDEAL 2009: 10th International Conference, Burgos, Spain, September 23-26, 2009. Proceedings 10, Springer, 2009, pp. 317–324.

[56] A. Lavin, S. Ahmad, Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark, in: 2015 IEEE 14th international conference on machine learning and applications (ICMLA), IEEE, 2015, pp. 38–44.

[57] A. Ihler, J. Hutchins, P. Smyth, Adaptive event detection with time-varying poisson processes, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 207–216.

[58] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[59] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (7) (1997) 1145–1159.

[60] J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (roc) curve., Radiology 143 (1) (1982) 29–36.

Appendix A. Figures related to experiment of Section 8.4

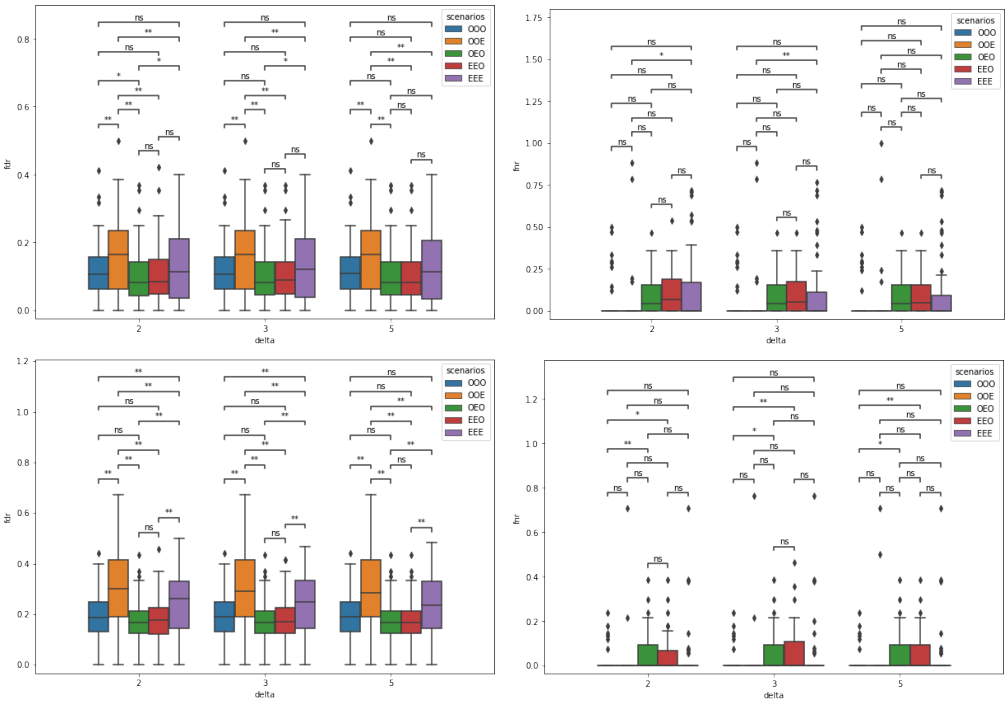


Figure A.33: Boxplots of FDR and FNR for anomaly detection on Gaussian time series having breakpoint in the mean according to the different Detectors described in Table 9 and shift size Δ. Top-left: FDR while $\alpha = 0.1$, Top-right: FNR while $\alpha = 0.1$, Bottom-left: FDR while $\alpha = 0.2$, Top-right: FNR while $\alpha = 0.2$.

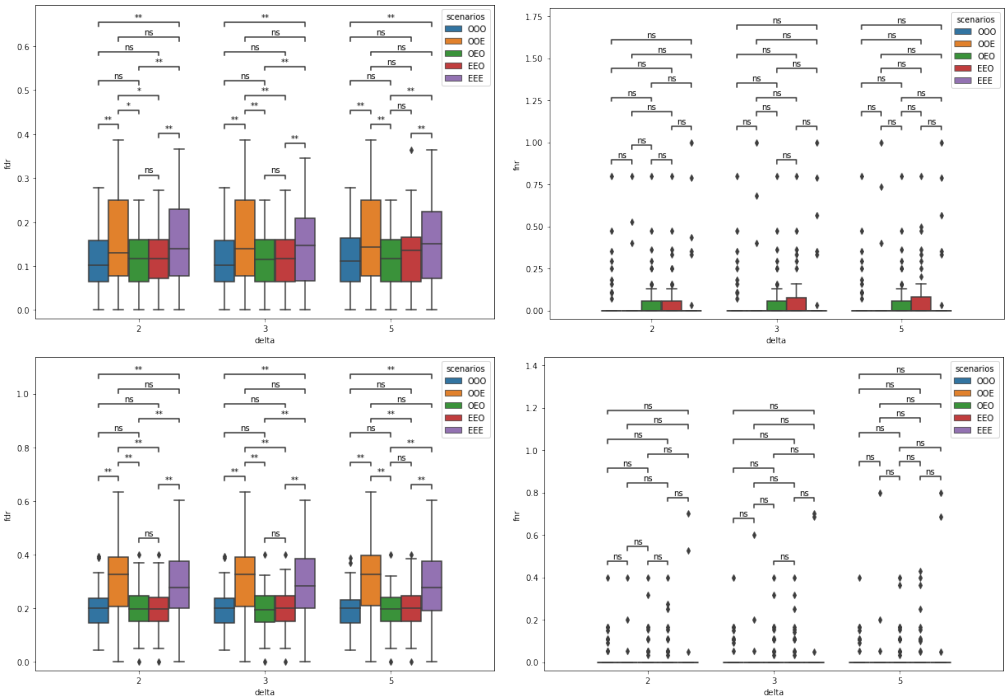


Figure A.34: Boxplots of FDR and FNR for anomaly detection on Student time series having breakpoint in the mean according to the different Detectors described in Table 9 and shift size Δ. Top-left: FDR while $\alpha = 0.1$, Top-right: FNR while $\alpha = 0.1$, Bottom-left: FDR while $\alpha = 0.2$, Top-right: FNR while $\alpha = 0.2$

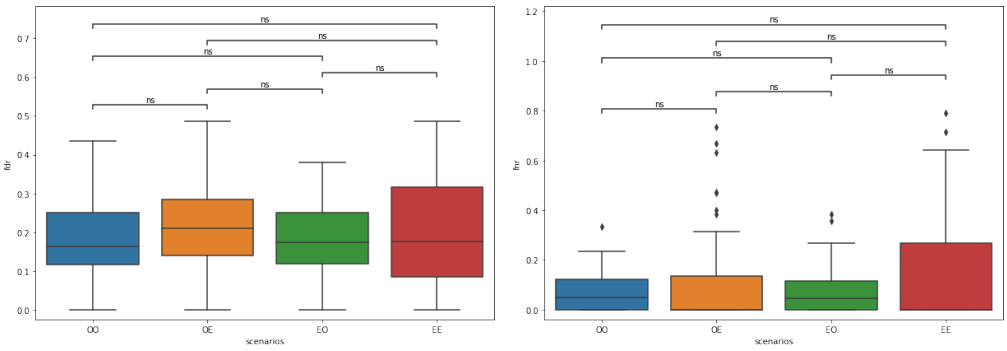


Figure A.35: Boxplots of FDR and FNR for anomaly detection on Gaussian Mixture time series having breakpoint in the mean according to the different Detectors described in Table 9. Left: FDR while $\alpha = 0.2$, right: FNR while $\alpha = 0.2$.

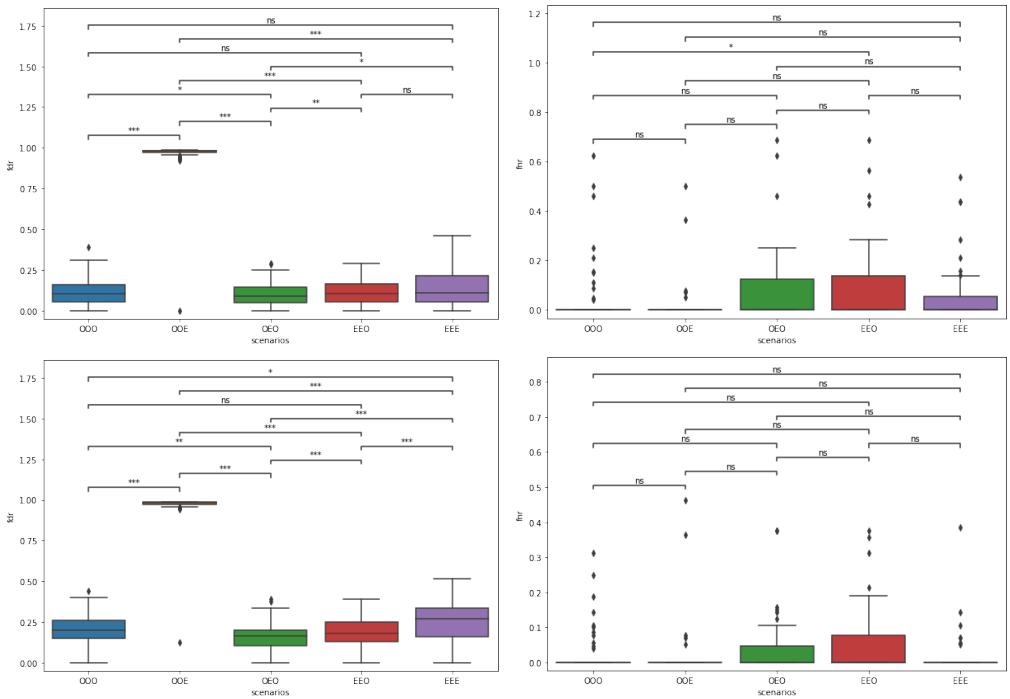


Figure A.36: Boxplots of FDR and FNR for anomaly detection on Gaussian time series having breakpoint in the mean and variance according to the different Detectors described in Table 9. Top-left: FDR while $\alpha = 0.1$, Top-right: FNR while $\alpha = 0.1$, Bottom-left: FDR while $\alpha = 0.2$, Top-right: FNR while $\alpha = 0.2$

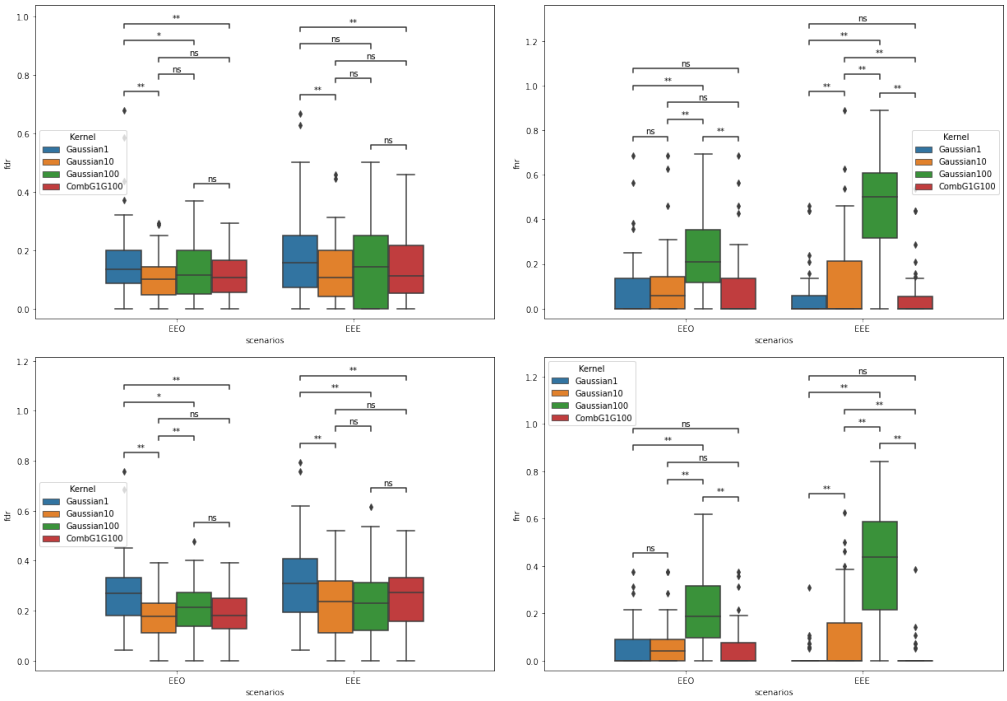


Figure A.37: Boxplots of FDR and FNR for anomaly detection on Gaussian time series hawing breakpoint in the mean and and in the variance according to the chosen Kernel. Top-left: FDR while $\alpha = 0.1$, Top-right: FNR while $\alpha = 0.1$, Bottom-left: FDR while $\alpha = 0.2$, Top-right: FNR while $\alpha = 0.2$

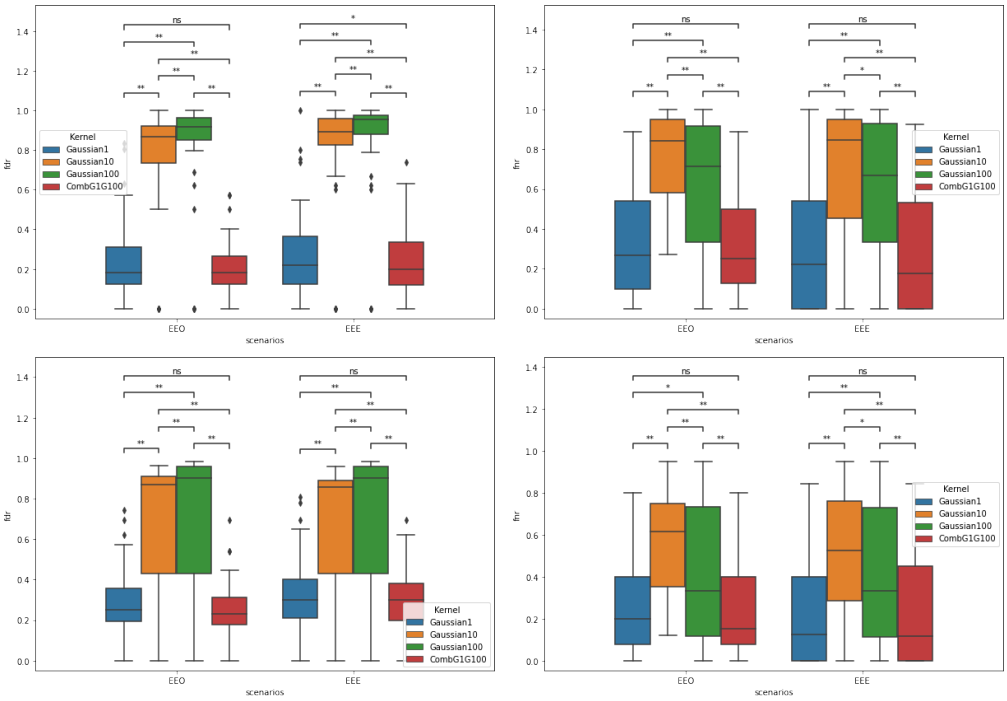
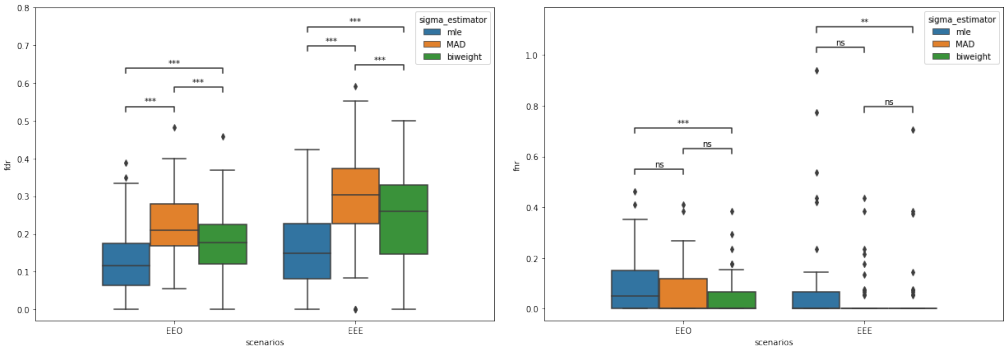


Figure A.38: Boxplots of FDR and FNR for anomaly detection on Gaussian time series having breakpoints in the variance according to different the chosen Kernel. Top-left: FDR while $\alpha = 0.1$, Top-right: FNR while $\alpha = 0.1$, Bottom-left: FDR while $\alpha = 0.2$, Top-right: FNR while $\alpha = 0.2$



(a) FDR

(b) FNR

Figure A.39: Boxplots of the FNR and FDR according to the chosen variance estimator.