



# A communication-efficient, online changepoint detection method for monitoring distributed sensor networks

Ziyang Yang<sup>1</sup> · Idris A. Eckley<sup>2</sup> · Paul Fearnhead<sup>2</sup>

Received: 20 October 2023 / Accepted: 25 March 2024  
© The Author(s) 2024

## Abstract

We consider the challenge of efficiently detecting changes within a network of sensors, where we also need to minimise communication between sensors and the cloud. We propose an online, communication-efficient method to detect such changes. The procedure works by performing likelihood ratio tests at each time point, and two thresholds are chosen to filter unimportant test statistics and make decisions based on the aggregated test statistics respectively. We provide asymptotic theory concerning consistency and the asymptotic distribution if there are no changes. Simulation results suggest that our method can achieve similar performance to the idealised setting, where we have no constraints on communication between sensors, but substantially reduce the transmission costs.

**Keywords** Changepoints · MOSUM · Online · Real-time analysis · Internet of Things · Distributed computing

## 1 Introduction

During the last decade, there has been a significant focus on the important challenge of efficient and accurate detection of changes in both univariate and multivariate data sequences (Cho and Fryzlewicz 2014; Fisch et al. 2022; Kovács et al. 2023; Truong et al. 2020; Tveten et al. 2022; Wang and Samworth 2018). More recently, focus has turned to translating the efficiency of such approaches to the online setting, typically motivated by an applied challenge such as how to deal with limited computational power (e.g. Ward et al. 2024). Recent major contributions to the online setting include Adams and MacKay (2007), Tartakovsky et al. (2014), Yu et al. (2023), Chen et al. (2022), and Romano et al. (2023). In this paper we consider a less studied scenario, monitoring edge-behaviour within distributed sensor networks, which are common architectures within the Internet of Things framework (IoT). The importance of efficiently detecting changes at the edge efficiently, whilst minimising communication between sensors and the cloud is perhaps best

appreciated by considering two key applications: detecting cyber-attacks on smart cities (Alrashdi et al. 2019) and optimising the performance of base stations (Wu et al. 2019).

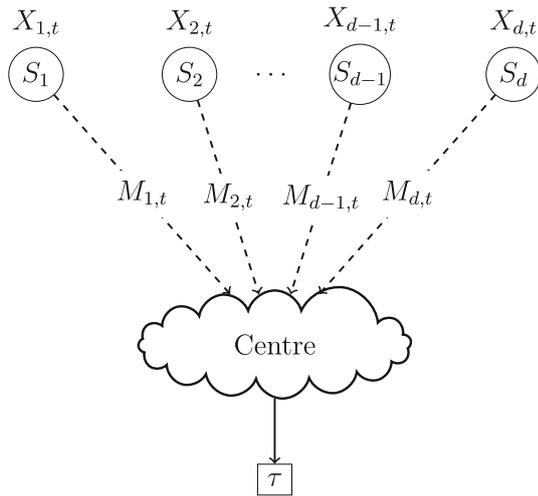
Consider, by way of example, Fig. 1 which shows a schematic representation of real-time monitoring within a distributed network. Here we assume that  $d$  data streams are monitored, each by its own sensor. Communication between the sensors and the centre is possible as shown by the dashed lines. An unusual event happens at time  $\tau$ , and we want to detect this event as quickly as possible. However, in modern sensor networks that deploy IoT devices the computational resources of the sensors can be substantial. Moreover, communication between the sensors and the cloud can be problematic due to the heavy energy usage involved with transmitting data (Varghese et al. 2016; Pinto and Castor 2017). As such, we need algorithms that can identify the time when it is important for information to be shared with the cloud. More specifically, in this article, we seek to develop a new method to detect changes within such a network in real time with high statistical power and as little communication and computation as possible.

Changepoint methods which can be applied in the fully centralised problem, when the data from the sensors is processed and transmitted to the centre (cloud) at every time step, are well studied. Approaches typically seek to calculate the maximum or the sum of all the test statistics (see, e.g., Mei 2010; Xie and Siegmund 2013; Chan 2017; Chen

✉ Idris A. Eckley  
i.eckley@lancaster.ac.uk

<sup>1</sup> STOR-i Centre for Doctoral Training, Lancaster University, Lancaster, UK

<sup>2</sup> Department of Mathematics and Statistics, Lancaster University, Lancaster, UK



**Fig. 1** Schematic representation of a sensor network made up of  $d$  sensors, where  $S_i$  is the index for sensor  $i$ ,  $X_{i,t}$  is the data observed at sensor  $i$ , and  $M_{i,t}$  is the message transmitted from sensor  $i$  to centre at time  $t$

et al. 2022; Gösmann et al. 2022). The rationale behind these methods is to set thresholds and raise the alarm if the aggregated test statistics from multiple streams exceed pre-defined thresholds. Numerical experiments Mei (2010) indicate that taking the maximum is the optimal method when there are only a few affected data streams—what we will term a sparse change. Conversely, taking the sum is optimal when most data streams are affected, also known as a dense change.

Recent contributions to this distributed problem include (Rago et al. 1996; Veeravalli 2001; Appadwedula et al. 2005; Mei 2005, 2011; Tartakovsky and Kim 2006; Banerjee and Veeravalli 2015). Among them, two recent papers of particular interest develop *communication efficient* schemes for monitoring a large number of data streams (Zhang and Mei 2018; Liu et al. 2019). The key idea is that each sensor computes a local monitoring statistic and then employs a thresholding step, only sending the statistic to the centre if there is some evidence of a change. The information from multiple sensors is then combined at the centre. This approach reduces unnecessary transmission by ignoring streams with little evidence for a change, while only focusing on data streams that show signs of change.

Although computationally feasible, existing works assume that the pre- and post-change mean are known. In practice, the pre-change mean can be estimated based on historical data. However assuming a known post-change mean is typically unrealistic in practice, with an incorrect value potentially leading to a failure to detect, or poor detection power. Liu et al. (2019) approximate the post-change mean recursively but, as a consequence, somewhat sacrifice statistical power of the algorithm.

Our approach builds on recent work developing the moving sum (MOSUM) as a window-based changepoint methods (see, e.g., Aue et al. 2012; Kirch and Kamgaing 2015; Kirch and Weber 2018). Specifically, we propose an online communication-efficient changepoint detection algorithm (distributed MOSUM) to detect changes in real-time within the distributed network setting. A local threshold is chosen to filter out unimportant information and only transmit the statistically important test statistic to the centre. The change will be alarmed when the aggregated test statistic exceeds the pre-defined global threshold in the central cloud. The low time complexity and communication efficient scheme of our proposed method makes it suitable for online monitoring. We also establish that the proposed method can achieve similar statistical power as the idealistic setting, where there is no communication constraint, at detecting large changes whilst substantially reducing the transmission cost. Moreover, we also show how to make the detection performance of distributed MOSUM close to that of the idealised setting by increasing the window size, which will only sacrifice the storage cost and a little transmission cost.

The key differences between our work and previous distributed changepoint detection contributions (e.g., Liu et al. 2019) are: Firstly, a moving window-based test statistic MOSUM is chosen to avoid the requirement of knowledge of the post-change mean. Secondly, earlier works have been based on the framework that controls the average run length (ARL)—the average amount of time until incorrectly detect a change. However, such a metric gives a somewhat limited amount of information since the distribution of run length is usually unknown. For instance, if multiple procedures end quickly while a few replications stop significantly longer, the ARL would be the same if all the replications terminated around the same time. Conversely, in this work, we present methods in terms of controlling the error rate under the null at a specific level, and with asymptotic power 1 under alternatives. Furthermore, our ideas generalise trivially to methods controlling the average run length.

The structure of this paper is as follows. In Sect. 2, the problem setting is outlined, before introducing the distributed MOSUM methodology in Sect. 3. Several theoretical results for this new approach are given in Sect. 4. Simulation studies are carried out in Sect. 5, before ending with some concluding remarks (Sect. 6).

## 2 Problem setting

We begin by assuming that we have  $d$  sensors, each of which is observed as follows:  $\mathbf{X}_t = (X_{1,t}, X_{2,t}, X_{3,t}, \dots, X_{d,t})$  at every time point  $t \in \mathbb{N}$ . Here  $\mathbf{X}_t$  could be raw data or the residuals after pre-processing the data. These observations are assumed to be identically distributed and independent

across series. Such assumptions are common in the problem of detecting changes within a distributed system setting (e.g., Tartakovsky and Veeravalli 2002; Mei 2010; Xie and Siegmund 2013; Liu et al. 2019). We do not strictly assume time independence here, but our method is optimal when this assumption holds. Moreover, the impact of time dependence will be numerically studied in Sect. 5.3.

We begin by assuming that at some unknown time,  $\tau$ , the distribution of some unknown subsets of  $d$  sensors will change. For simplicity, we only consider change in mean, but the ideas below are easily extended to other changepoint settings. Therefore, in this illustrative change in mean setting, the model for the data is expressed as follows:

$$X_{i,t} = \mu_i + \delta_i \mathbb{1}_{\{t > \tau\}} + \epsilon_{i,t}, \quad t \in \mathbb{N}, 1 \leq i \leq d, \quad (2.1)$$

where  $\mu_i$  is the known pre-change mean,  $\delta_i$  is the mean shift, and  $\{\epsilon_{i,t} : t \in \mathbb{N}\}$  are strictly stationary error sequences. After time  $\tau$ , the mean of the  $i$ -th data stream changes immediately from  $\mu_i$  to  $\mu_i + \delta_i$ . Here it is useful to note that our setting also permits some  $\delta_i = 0$ , which means that only a subset of data streams are affected by the change. Without loss of generality, we assume  $\mu_i = 0$ . Under the null hypothesis, the model for the data can be rewritten as

$$X_{i,t} = \epsilon_{i,t}, \quad t \in \mathbb{N}, 1 \leq i \leq d. \quad (2.2)$$

Moreover under the alternative hypothesis, the model is  $X_{i,t} = \delta_i + \epsilon_{i,t}, t \in \mathbb{N}, 1 \leq i \leq d$ . Our aim is to monitor such a system and raise the alarm as soon as possible following the event at time  $\tau$ . One way of achieving this is to perform hypothesis testing sequentially, i.e., evaluate the null hypothesis of no change in mean at each time point  $t \in \mathbb{N}$ . The algorithm will stop and declare a change when we can reject the null hypothesis.

In the classical sequential changepoint detection problem, we evaluate the performance of an algorithm subject to a constraint on its false alarm rate. First, consider an open-ended stopping rule where the algorithm never we have an infinite time-window of measurements and the algorithm never halts until it detects a change. The false alarm rate can be evaluated in two ways. Assume there is no change, and let  $\hat{\tau}$  be the time at which we detect a change, with the convention that  $\hat{\tau} = \infty$  if we detect no change. One approach is to control the average run length,  $E^\infty(\hat{\tau})$ , the expected time of to a false alarm. This makes sense for procedures with a constant threshold for detection, for which we are certain to detect a change under the Null if we monitor for an infinite time period. Alternatively, one can control the false alarm rate,  $P^\infty(\hat{\tau} < \infty)$ , the probability of a false alarm. To control this over an infinite time horizon requires increasing the threshold for detecting a change over time. Equivalently, this can be achieved by multiplying the test statistic with a weight

function  $w(\cdot) < 1$ . See Leisch et al. (2000); Zeileis et al. (2005); Horváth et al. (2008); Aue et al. (2012); Kirch and Kamgaing (2015); Weber (2017); Yau et al. (2017); Kirch and Weber (2018); Kengne and Ngongo (2022) for examples of how to choose an appropriate weight function.

In our paper, we focus on controlling the false alarm rate. However Aue et al. (2012) states that ‘‘applying open-ended procedures built from the asymptotic critical values have a tendency to be too conservative infinite samples’’. Therefore, our paper considers a close-ended stopping rule. In this approach, the algorithm will stop either upon detecting a change or upon reaching the predefined monitoring time  $T$ . We thus control the false alarm rate over a time window of length  $T$ . However, the ideas we present can easily be adapted to the open-ended setting, and also to methods which control the average run length.

Under the context of distributed changepoint detection problem, we additionally evaluate the index—the average transmission cost  $\bar{\Delta}$ . This is the average number of transmissions at each time step for  $d$  sensors, and should be smaller than the pre-specified transmission cost  $\Delta$ .

Before introducing our proposed method, we first review relevant work. At time  $t$ , the local monitoring statistic,  $\mathcal{T}_i$  is calculated for the  $i$ th stream. Then all the local statistics  $\mathcal{T}_i$  can be combined into a global monitoring statistic  $\mathcal{T}$  at the fusion centre. There are two common choices of message combinations for monitoring changes within the distributed system. One of these two types, the SUM scheme (Mei 2010), declares a change when the sum of all the local monitoring statistics exceeds a pre-defined threshold, that is:

$$\begin{aligned} \hat{\tau}_{\text{sum}}(c_{\text{Global}}) &= \inf \{t \geq 1 : \mathcal{T} \geq c_{\text{Global}}\} \\ &= \inf \left\{ t \geq 1 : \sum_{i=1}^d \mathcal{T}_i \geq c_{\text{Global}} \right\}, \end{aligned}$$

where  $c_{\text{Global}}$  is global threshold. This way of combining statistics across streams is known to be good if the series are independent and the changes are dense. However, implementing this method on the distributed system requires sending every  $\mathcal{T}_i$  to the fusion centre, which is expensive. A sum-shrinkage method (Liu et al. 2019) is proposed to reduce the communication cost by thresholding the test statistics before summing them:

$$\begin{aligned} \hat{\tau}_{\text{sum}}(c_{\text{Local}}, c_{\text{Global}}) &= \inf \{t \geq 1 : \mathcal{T} \geq c_{\text{Global}}\} \\ &= \inf \left\{ t \geq 1 : \sum_{i=1}^d \mathcal{T}_i \mathbb{I}(\mathcal{T}_i \geq c_{\text{Local}}) \geq c_{\text{Global}} \right\}. \end{aligned}$$

Empirically the sum-shrinkage method could achieve similar performance as the SUM scheme in the dense case and surprisingly performs better in the sparse case.

When the change is sparse, it has been shown both theoretically and empirically (Mei 2010; Liu et al. 2019; Chen et al. 2022) that monitoring the maximum of the test-statistics across series is best. In such a setting, the MAX procedure (Tartakovsky and Veeravalli 2002) monitors the maximum of test statistics and raises the alarm when the maximum of the local test statistics exceeds the thresholds, that is:

$$\hat{t}_{\max}(c_{\text{Global}}) = \inf \{t \geq 1 : \mathcal{T} \geq c_{\text{Global}}\} \\ = \inf \left\{ t \geq 1 : \max_{1 \leq i \leq d} \mathcal{T}_i \geq c_{\text{Global}} \right\}.$$

The best choice of different schemes depends on the sparsity of changes which is based on the number of affected data streams  $p$ . This can be made precise if we consider an asymptotic setting where  $p \rightarrow \infty$  (Enikeeva and Harchaoui 2019), and define a change to be sparse if the number of affected streams is  $p = o(\sqrt{d})$ , and it to be a dense change otherwise. A recent paper (Chen et al. 2022) combines both SUM procedure and MAX procedure to achieve good performance regardless of the sparsity. In the context of distributed monitoring, the MAX procedure is trivially implemented without any communication. Specifically, each sensor has the threshold for the max-statistic and flags a change if their local statistic is above this threshold. Therefore, within this paper, we only focus on developing a communication-efficient version of the SUM scheme. Our aim is a method that performs well for dense changes, but limits the communication cost. We will use the SUM scheme as the ideal method to compare against since it has no restrictions on communication.

### 3 Distributed change point detection method

Our proposed methodology is summarized in Algorithm 1, and described in detail below. The method essentially comprises of three steps. The first step involves the parallel local monitoring of each data stream by the sensors. As the monitoring unfolds, messages are occasionally sent from the sensors to the centre to indicate the presence of a potential change. Finally, at the centre these messages are aggregated to find changes that occur across a number of data streams.

#### 3.1 Local monitoring

##### 3.1.1 Estimating the baseline parameters

Our sequential testing approach requires a historic data set of length  $m$  to estimate the baseline parameters. Theoretical results are obtained later in the paper when  $m \rightarrow \infty$ . The parameters of interest are the mean of each data stream  $\mu_i$

---

#### Algorithm 1: Centralized and distributed MOSUM

---

```

input : historic data  $x_{i,t}$  for  $i = 1, 2, \dots, d$ , and  $1 \leq t \leq m$ 
Estimating the baseline parameters // can be done offline
for  $i = 1$  to  $d$  do
    | estimate  $\hat{\mu}_i$  and  $\hat{\sigma}_i$ 
end
Data:  $x_{i,t}$  for  $i = 1, 2, \dots, d$  at time  $t$ 
while change is detected or reached the maximum monitoring time  $T$  do
    | Local monitoring // parallel computing
    for  $i = 1$  to  $d$  do
        |  $\mathcal{T}_i(m, k, h) = \frac{1}{\hat{\sigma}_i} \left| \sum_{t=m+k-h+1}^{m+k} (X_{i,t} - \hat{\mu}_i) \right|$ .
    end
    | Message passing
    if  $w(k, h)\mathcal{T}_i(m, k, h) > c_{\text{Local}}$  then
        |  $M_{i,t} = \mathcal{T}_i(m, k, h)$  // centralized scheme: set
        |  $c_{\text{Local}} = 0$ 
        | else  $M_{i,t} = 0$ 
    end
    | Global monitoring
    if  $w(k, h)\sqrt{\sum_i^d M_{i,t}} > c_{\text{Global}}$  then
        | stop the algorithm
        | output:  $\hat{t} = t$ 
    end
    |  $t \leftarrow t + 1$ 
end

```

---

and the variance of the errors  $\sigma_i^2$ . For the  $i$ th data stream these estimates are,

$$\hat{\mu}_i = \frac{1}{m} \sum_{t=1}^m X_{i,t}, \tag{3.1}$$

$$\hat{\sigma}_i^2 = \frac{1}{m} \sum_{t=1}^m (X_{i,t} - \hat{\mu}_i)^2.$$

If the errors cannot be assumed to be independent we can estimate the long run variance. This requires specifying a kernel function  $K(\cdot)$ :

$$\hat{\sigma}_i^2 = \frac{1}{m} \sum_{t=1}^m (X_{i,t} - \hat{\mu}_i)^2 + 2 \sum_{j=1}^{m-1} K\left(\frac{j}{l}\right) \hat{\gamma}_j^{(i)}, \tag{3.2}$$

where  $\hat{\gamma}_j^{(i)} = \frac{1}{m-j} \sum_{t=1}^{m-j} (X_{i,t} - \hat{\mu}_i)(X_{i,t+j} - \hat{\mu}_i)$ .  $\tag{3.3}$

In this setting, the Kernel function can be seen as a weighting function for sample covariance  $\hat{\gamma}_j^{(i)}$ . The kernel function must be symmetric and such that  $K(0) = 1$ . Various kernel functions are proposed. Standard kernel functions include Truncated (White and Domowitz 1984), Bartlett (Newey and West 1986) and Parzen (Gallant 2009) amongst others. Among them, the Bartlett kernel is frequently used in Econo-

metrics. This kernel takes the form:

$$K_{\text{Bartlett}}\left(\frac{j}{l}\right) = \begin{cases} 1 - \frac{j}{l}, & \text{for } 0 \leq j \leq l - 1, \\ 0, & \text{otherwise.} \end{cases}$$

For more details, see Horváth and Hušková (2012); Kiefer and Vogelsang (2002a); Kiefer and Vogelsang (2002b).

### 3.1.2 Starting local monitoring

Once the baseline parameters have been estimated, beginning at time  $m + 1$  data  $X_{i,m+1}, X_{i,m+2}, \dots$  are observed sequentially and monitored for a change. This is achieved using a MOSUM statistic which at monitoring time,  $k$ , takes a window containing the most recent  $h$  observations:

$$\mathcal{T}_i(m, k, h) = \frac{1}{\hat{\sigma}_i} \left| \sum_{t=m+k-h+1}^{m+k} (X_{i,t} - \hat{\mu}_i) \right|. \tag{3.4}$$

Following Aue et al. (2012), the MOSUM statistic will declare a change at time  $k$  when the weighted local MOSUM statistic  $w(k, h)\mathcal{T}_i(m, k, h)$  exceeds a pre-defined threshold. A weight function  $w(\cdot, \cdot)$  is introduced to control the asymptotic size of the detection procedure. Typically  $w(\cdot, \cdot)$  depends on the monitoring time  $k$ , and the window size  $h$ ,

$$w(k, h) = \frac{1}{\sqrt{h}} \rho\left(\frac{k}{h}\right), \tag{3.5}$$

for some appropriate  $\rho(\cdot)$ . The choice of the weight function controls the sensitivity of the test. A wide range of weight functions can be used as long as they are continuous functions that satisfy  $\inf_{0 \leq t \leq T} \rho(t) > 0$ . In this paper, we use the weight function proposed in Leisch et al. (2000) and Zeileis et al. (2005):

$$\rho(t) = \max(1, \log(1 + t))^{-1/2}.$$

Intuitively, if there is no change the weighted MOSUM will remain small, but it will be large if there is a change. Figure 2 gives the behavior of weighted MOSUM statistic under the null and the alternative assumptions for one data stream.

### 3.2 Message passing

The local monitoring described in the previous section is applied to each sensor independently. In order to make global decisions about the state of the system, messages from the sensors must be passed to the central hub (see Fig. 1). However, since there are constraints on communication in the system, the message passing process must be carefully designed.

At time  $t = m + k$ , where  $m$  is the historic period of length  $m$ , and  $k$  is the monitoring time, each sensor makes a decision as to whether or not to transmit a message to the centre. This message vector is denoted as  $\mathbf{M}_t = (M_{1,t}, M_{2,t}, \dots, M_{d,t})$ . We consider two different messaging regimes:

- Centralized messaging regime:  $\mathbf{M}_t = \mathcal{T}_i(m, k, h)$ .
- Distributed messaging regime:

$$M_{i,t} = \begin{cases} \mathcal{T}_i(m, k, h) & \text{if } w(k, h)\mathcal{T}_i(m, k, h) > c_{\text{Local}}, \\ NULL & \text{otherwise.} \end{cases} \tag{3.6}$$

The centralized messaging regime is one where there is no constraint on the communication between the sensors and the centre, so all sensors send a message to the centre at each time instant. This is similar to the ‘‘SUM’’ scheme change-point detection method proposed by Mei (2010). However, when communication is expensive, a ‘‘distributed’’ messaging regime can be used where each of the sensors only send local monitoring statistics that exceed a chosen threshold. The *NULL* means no message is sent. The threshold  $c_{\text{Local}}$  can be chosen to control the fraction of transmitting sensors when there is no change. It is worth noting that when  $c_{\text{Local}} = 0$ , the ‘‘distributed’’ messaging regime is equivalent to ‘‘centralized’’ messaging regime.

### 3.3 Global monitoring

In our paper, we assume that there is no communication delay between sensors and the central hub, so the message could be immediately received by the centre at time  $t$ . Based on the messages received, the centre will make the decision as to whether or not to flag a change.

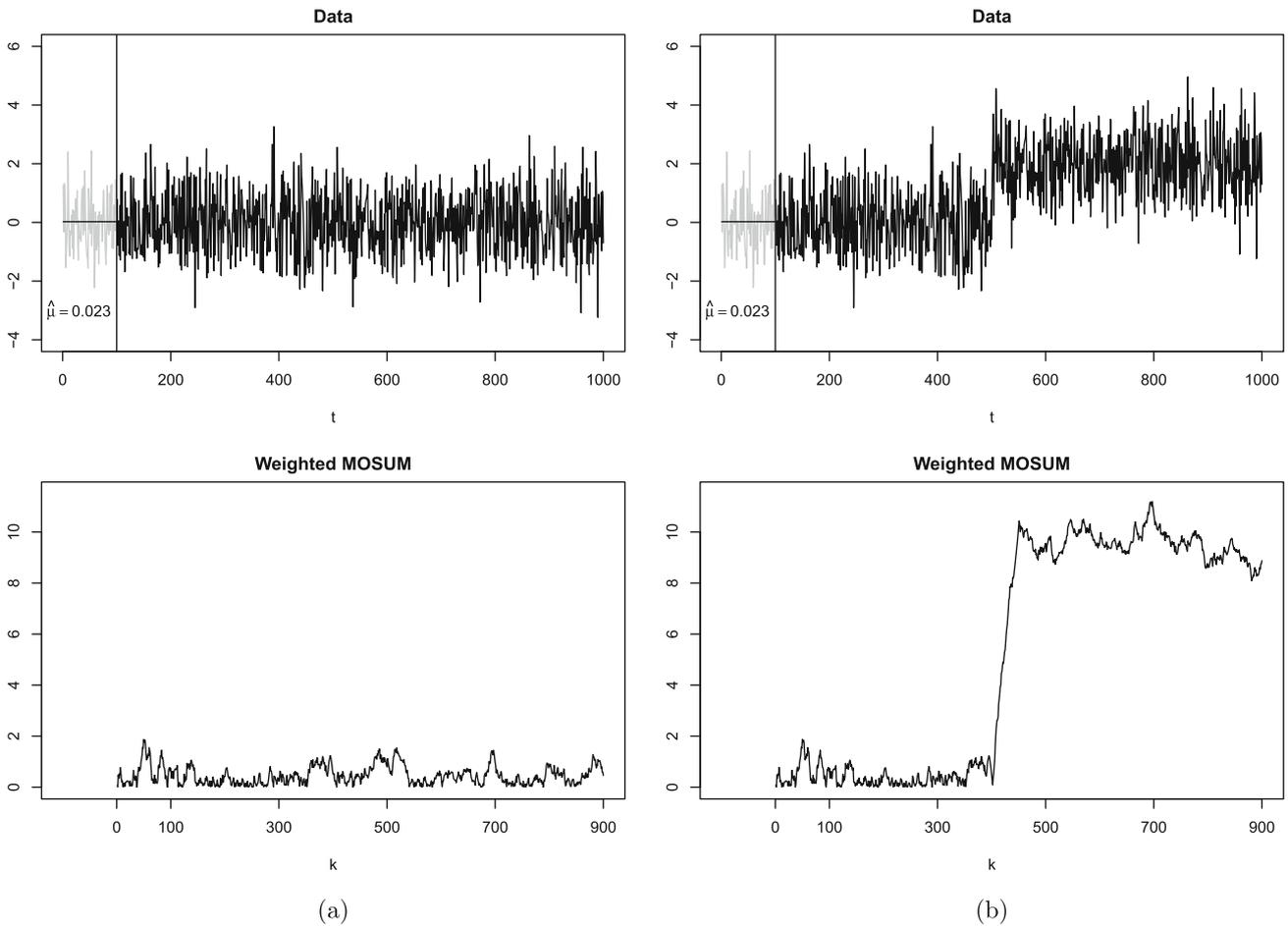
#### 3.3.1 Combining messages

Depending on different messaging regimes, the global MOSUM statistics are constructed as follows:

- Centralized global MOSUM statistic:

$$\mathcal{T}(m, k, h) = \sqrt{\sum_{i=1}^d M_{i,t}^2}, \tag{3.7}$$

This is similar to the SUM scheme mentioned in Sect. 2. By using such a scheme, Formula 3.7 is the idealistic scheme under dense change.



**Fig. 2** Example time series with no change (a) and a single change (b) in the top row. The bottom row shows the weighted MOSUM statistic with a historic period of length  $m = 100$  and a window size of  $h = 50$

- Distributed global MOSUM statistic:

$$T(m, k, h) = \sqrt{\sum_{i=1}^d M_{i,t}^2 \mathbb{1}_{\mathcal{T}_i(m,k,h) > c_{Local}}}, \quad (3.8)$$

where  $NULL$  values in Formula 3.6 are taken to be zeros in the sum. The form of Eq. (3.8) is taken from the multivariate MOSUM (Kirch and Kamgaing 2015; Weber 2017; Kirch and Weber 2018).

### 3.3.2 Declaring the change

Similar to the local monitoring procedure, a change is declared as soon as the weighted global MOSUM exceeds a threshold. A closed-end stopping rule can be used when the aim is to monitor changes within a fixed time. This can be formalised as

$$\tau_{m, \tilde{T}} = \min \left\{ 1 \leq k \leq \lfloor m\tilde{T} \rfloor : w(k, h)T(m, k, h) > c_{Global} \right\}, \quad (3.9)$$

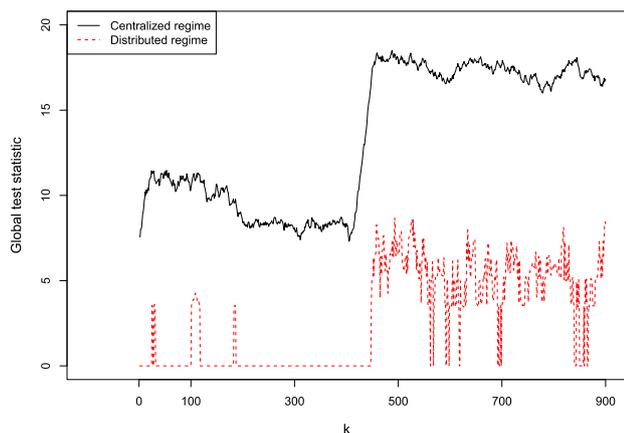
where  $\min\{\emptyset\} = \infty$  and the total length of the data  $T = m\tilde{T}$ . If no change is detected by this stopping rule prior to  $\lfloor m\tilde{T} \rfloor$ , the monitoring procedure is terminated. The parameter  $\tilde{T} > 0$  governing the length of the monitoring period is chosen in advance (Horváth et al. 2008; Aue et al. 2012).

Figure 3 shows the weighted global MOSUM statistic for the distributed and centralized messaging regimes on the same dataset. Whenever the weighted global MOSUM of distributed regime hits zero, there is no communication between the edges and the centre at that time.

In the next section, we will show the theoretical properties of our proposed method under  $H_0$  and  $H_A$ .

## 4 Theoretical properties for distributed MOSUM

This section considers the theoretical properties of the closed-end stopping rule,  $\tau_{m, \tilde{T}}$  defined in Eq. (3.9) as  $m \rightarrow \infty$ . Firstly, in Sect. 4.1 we find the limiting distribution under the



**Fig. 3** Example of the weighted global MOSUM statistic for the distributed (red dashed line) and centralized (black line) regime. The result is obtained with  $T = 1000$ ,  $d = 100$ ,  $m = 100$ ,  $h = 50$ ,  $\delta = 0.5$  and the number of affected sensors  $p = 50$ . A value of  $c_{Local} = 3.44$  was used in the distributed regime. (Color figure online)

null hypothesis for the different procedures. Then, appropriate choices for the thresholds,  $c_{Local}$  and  $c_{Global}$  are given in Sect. 4.2 using these results. Finally, in Sect. 4.3 we prove that the detection procedures we have studied are consistent under alternatives.

Three key assumptions are made in order to derive asymptotic results, which are the same in Horváth et al. (2008), Aue et al. (2012), and Weber (2017):

**Assumption 1** (Clean historic data)  $h \rightarrow \infty$  as  $m \rightarrow \infty$  and the location of the changepoint  $\tau > m$  for  $1 \leq i \leq d$ .

This assumption is to guarantee we can get good estimators based on the training dataset, and it can be easily achieved in real applications.

**Assumption 2** (Asymptotic regime)  $h \rightarrow \infty$  as  $m \rightarrow \infty$  and

$$\lim_{m \rightarrow \infty} \frac{h}{m} \rightarrow \beta \in (0, 1].$$

This assumption quantifies the long run connection between the length of the historical period  $m$  and the window size  $h := h(m)$ .

**Assumption 3** (FCLT on errors)

$$\lim_{m \rightarrow \infty} \frac{1}{\sqrt{m}} S_i(mt) \xrightarrow{\mathcal{D}} \sigma_i W_i(t)$$

where  $\sigma_i > 0$ ,  $\{W_i(t), 0 \leq t < \infty\}$  is a standard Brownian motion when  $h \rightarrow \infty$ , and  $S_i(x) = \sum_{t=1}^{\lfloor x \rfloor} \epsilon_{i,t}$ .  $\sigma_i$  can be estimated by  $\hat{\sigma}_i$ . Furthermore,  $\hat{\sigma}_i$  satisfying  $\hat{\sigma}_i \xrightarrow{\mathcal{P}} \sigma_i$  as  $m \rightarrow \infty$ .

This assumption is a functional central limit theorem on the errors,  $\epsilon$ , in the model for the data (2.1).

### 4.1 Asymptotics under the null

In this part, the asymptotic theories of our proposed method will be given, which can help guide the choice of thresholds.

The local monitoring process of our proposed method within each sensor is the same as univariate MOSUM detection process. Thus, Theorem 1 and Corollary 1 of the local MOSUM can be directly cited from Horváth et al. (2008), Aue et al. (2012) and Weber (2017). For simplicity, we denote

$$Z_i(t) = \left| W_i \left( \frac{1}{\beta} + t \right) - W_i \left( \frac{1}{\beta} + t - 1 \right) - \beta W_i \left( \frac{1}{\beta} \right) \right|, \quad 1 \leq i \leq d \tag{4.1}$$

where  $\{W_i(t), 0 \leq t < \infty\}$  are independent standard Brownian motions.

**Theorem 1** (Local MOSUM) *If Assumption 1–3, and model 2.2 holds, then under  $H_0$ , let  $k = ht$  for any  $t > 0$*

$$\lim_{m \rightarrow \infty} w(k, h) \mathcal{T}_i(m, k, h) \xrightarrow{\mathcal{D}} \rho(t) Z_i(t).$$

**Corollary 1** (Local MOSUM - asymptotic type-I error) *Under  $H_0$ , for any  $\tilde{T} > 0$  and  $i$ th data stream,*

$$\lim_{m \rightarrow \infty} P \left( \tau_{m, \tilde{T}}^{(i)} < \infty \right) = P \left( \sup_{0 \leq t \leq \tilde{T}/\beta} \rho(t) Z_i(t) > c_{Local} \right).$$

Thus, the false alarm rate for one data stream is asymptotically equal to a pre-specified type-I-error  $\in (0, 1)$ .

Following the results of local MOSUM, similar results for global MOSUM follow readily. These can be used to choose thresholds given the pre-defined Type-I-error. Below we obtain two limiting distributions, for the centralized and distributed regime settings of Sect. 3 respectively.

**Theorem 2** (Global MOSUM) *Let  $k = ht$  for any  $t > 0$ , then under  $H_0$ ,*

$$\lim_{m \rightarrow \infty} w(k, h) \mathcal{T}(m, k, h) \xrightarrow{\mathcal{D}} \rho(t) \begin{cases} \sqrt{\sum_{i=1}^d Z_i(t)^2} & \text{centralized case,} \\ \sqrt{\sum_{i=1}^d Z_i(t)^2} \mathbb{1}_{\rho(t) Z_i(t) > c_{Local}} & \text{distributed case.} \end{cases}$$

**Proof** See Appendix A.1. □

Thus, their limiting distribution will be a function of Gaussian process. Using the Theorem 2, the following may be obtained:

**Corollary 2** (Global MOSUM—asymptotic type-I error) *Under  $H_0$ , for any  $\tilde{T} > 0$ ,*

**Table 1** Critical values for the centralized procedures, results averaged over five thousand replications

$\alpha$	$c_{Local}$	$c_{Global}^{Centralized}(\alpha)$
0.10	0	14.1
0.05	0	14.4
0.01	0	15.0

**Table 2** Critical values for the distributed procedure with different values for  $c_{Local}$ , results averaged over five thousand replications

$\alpha$	$c_{Local}$	$c_{Global}^{Distributed}(\alpha)$
0.10	3.15	7.48
	3.44	6.70
	4.05	5.20
0.05	3.15	7.89
	3.44	7.16
	4.05	6.02
0.01	3.15	8.74
	3.44	8.01
	4.05	6.59

### 4.2.1 The choice of local threshold $c_{Local}$

The values for  $c_{Local}$  used in Table 2 are somewhat arbitrary. The main influence of the value of local threshold is that it controls the proportion of messages that the system can pass (on average) per iteration. For  $d$  streams, the number of sensors passing message at each time step is:

**Corollary 3** (Transmission cost) *For any  $t > 0$  and  $k = ht$ , the expected fraction of transmitting sensors at each time step is*

$$\bar{\Delta}_t = d P(\rho(t)|Z| > c_{Local}).$$

where  $Z$  is the standard normal distribution.

Therefore, the local threshold can be chosen based on the restriction of the transmission cost. Combined with pre-defined type-I-error, the global threshold will be given based on Theorem 2.

$$\lim_{m \rightarrow \infty} P(\tau_{m, \tilde{T}} < \infty) = \begin{cases} P\left(\sup_{0 \leq t \leq \tilde{T}/\beta} \rho(t) \sqrt{\sum_{i=1}^d Z_i(t)^2} > c_{Global}\right) & \text{centralized case,} \\ P\left(\sup_{0 \leq t \leq \tilde{T}/\beta} \rho(t) \sqrt{\sum_{i=1}^d Z_i(t)^2} \mathbb{1}_{\rho(t)Z_i(t) > c_{Local}} > c_{Global}\right) & \text{distributed case.} \end{cases}$$

This result can lead us to find the local and global thresholds which can obtain the pre-defined type-I-error.

### 4.2 Obtaining critical values

Using the results of the previous section, appropriate critical values can be found such that the asymptotic type-I error is controlled for the different procedures. To achieve this the stochastic processes  $\{Z_i(t), 0 \leq t \leq \tilde{T}/\beta, 1 \leq i \leq d\}$  need to be approximated on a fine grid. This is done in the same way as Aue et al. (2012), simulating the component standard Brownian motions using ten thousand i.i.d. standard normal random variables. The parameters used were  $\beta = 1/2$  and  $\tilde{T} = 10$ . Tables 1 and 2 give critical values for  $\alpha \in \{0.10, 0.05, 0.01\}$ .

Since the critical values obtained above are valid asymptotically (in  $m$ ), an important question to consider is how they perform in finite samples. Numerical results of empirical size in the finite sample are shown in Table 3. These indicate that the implementation in the finite sample setting can be conservative, as per Aue et al. (2012). However, approximately, the type-I error is controlled at the correct level for both of the global procedures in finite samples.

### 4.3 Asymptotics under the alternative

Under the alternative it is assumed that there is a changepoint at monitoring time  $k^*$  and a subset  $S$  of the data streams have an altered mean

$$H_A : \tau = m + k^* \quad \& \quad \exists S \subset \{1, 2, \dots, d\} : \delta_i \neq 0 \quad \text{for } i \in S.$$

Deriving sharp asymptotic results on the detection delay of the proposed method is challenging, and thus we focus only on giving consistency results. A procedure is consistent if it stops in finite time with probability approaching one as  $m \rightarrow \infty$ . In other words, the test statistic should tend to infinity as  $m \rightarrow \infty$ . In the asymptotic regime of interest, we additionally assume that the changepoint  $k^*$  grows at the same order as  $h$ , that is  $\frac{k^*}{h} \rightarrow \gamma \geq 0$ , and the size of change  $\delta_{i,t}$  satisfies  $\sqrt{h}|\delta_{i,t}| \rightarrow \infty$  as  $m \rightarrow \infty$  and  $h \rightarrow \infty$ . These assumptions are the same in Aue et al. (2012).

**Theorem 3** (Global MOSUM: Consistency) *If the assumption above holds, under  $H_A$ ,*

- (i) *the changepoint  $k^* \leq \lfloor hv \rfloor$  for some  $0 < \nu < \tilde{T} \frac{m}{h}$ ,*

**Table 3** Empirical size, results averaged over one thousand replications with  $\alpha = 0.05, \tilde{T} = 10$ , and  $\beta = 1/2$

Method	$c_{\text{Local}}$	$c_{\text{Global}}$	Proportion of false alarms		
			$m = 200$ (%)	$m = 400$ (%)	$m = 500$ (%)
Distributed	3.15	7.89	5.14	5.18	5.5
	3.44	7.16	5.3	5.38	5.12
	4.05	6.02	5.5	5.64	5.16
Centralized	–	14.4	5.92	5.28	5.3

(ii) there exists a constant  $c > 0$  such that  $\rho(x + 1) \geq c$  for all  $x \in (v, \tilde{T} \frac{m}{h} - 1)$ .

Then, as  $m \rightarrow \infty$  and  $h \rightarrow \infty$

$$\max_{1 \leq k \leq \lfloor m\tilde{T} \rfloor} w(k, h)\mathcal{T}(m, k, h) \xrightarrow{\mathcal{P}} \infty.$$

**Proof** See Appendix A.2. □

Thus, our proposed method is consistent.

### 5 Simulations

In this section, we will present the numeric performance of our algorithm. Since the SUM procedure that is optimal when the change is dense, we will evaluate the performance in the dense case, specifically when the affected data streams  $p = d$ . Firstly, the different practical choices of thresholds at fixed type-I-error will be investigated. Here the performance of our proposed method was also compared against the idealistic setting. Finally, the effect of parameters and the violation of the independence assumption are investigated.

The set-up of the simulations is as follows. For simplicity, the data generating process under the null is that  $X_{i,t} \sim N(0, 1)$  for  $1 \leq i \leq d$  and  $1 \leq t \leq T$ . To compare fairly, the type-I-error of all procedures is controlled to be 0.05 under the null.

The family of alternatives considered is that

$$\begin{aligned} X_{i,t} &\sim N(0, 1) \quad \text{for } 1 \leq i \leq d, 1 \leq t < \tau \quad \text{and} \\ X_{i,t} &\sim N(\delta_i, 1) \quad \text{for } 1 \leq i \leq d, \tau \leq t \leq T. \end{aligned}$$

We assume the change will affect all the sensors instantaneously. But the size of the change is unknown. We consider two scenarios of mean shift: 1) Same size:  $\delta_i = \delta =$  some constant values for  $1 \leq i \leq d$ ; 2) Random size:  $\delta_i = \eta N(0, 1)$ , where  $\eta$  is the scale factor controlling the magnitude of size. The average detection delay (ADD)  $\bar{D}$  and average communication cost  $\bar{\Delta}$  are then measured:

$$\bar{D} = E(\hat{\tau} - \tau | \hat{\tau} > \tau)$$

$$\bar{\Delta} = \sum_{t=m+1}^{\hat{\tau}} \frac{\sum_{i=1}^d \mathbb{1}(w(k, h)\mathcal{T}_i(m, k, h) > c_{\text{Local}})}{\hat{\tau} - m - 1},$$

#### 5.1 The numerical dependency on local thresholds

Our proposed method requires specifying two thresholds. Usually,  $c_{\text{Global}}$  can be given based on the Theorem 2 once  $\alpha$  and  $c_{\text{Local}}$  are confirmed. Therefore, it is crucial to pick an appropriate local threshold. This section gives numeric results with different values of local thresholds, which may provide some guidance in choosing the local threshold.

Figure 4 gives the average detection delay and transmission cost for different values of local thresholds. There is a trade-off between communication savings and detection performance when choosing the local threshold. Larger local thresholds can reduce the transmission cost but will also lead to longer delays, especially when the change is small. However, with the increase in the mean shift, the detecting power of larger thresholds will close to that of small thresholds.

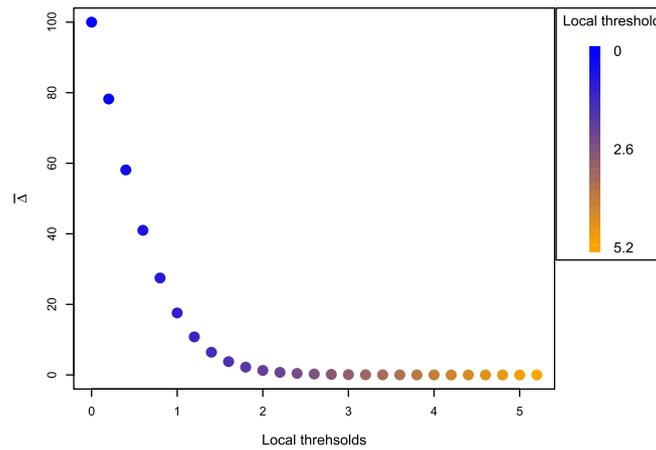
A centralized framework can be seen as an idealistic setting, which is equivalent to distributed setting when  $c_{\text{local}} = 0$ . Compared with the idealistic setting, the distributed MOSUM can achieve similar performance when the size of the change is not small but also reduces massive transmission costs. But we will lose power in detecting small changes. We show the result below that distributed MOSUM can approximate the performance of idealistic setting overall by increasing the window size.

#### 5.2 The numerical dependency on parameters

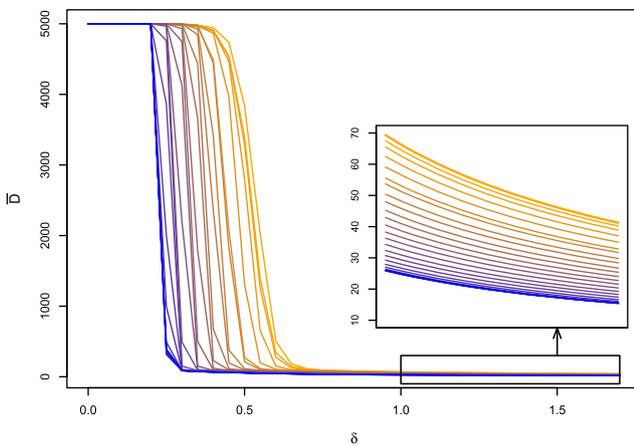
One advantage of using MOSUM statistics is that we do not need to specify the post-change mean. Instead, our proposed method requires specifying the window size  $h$  and the training size  $m$ . In this section, we will investigate the impact of bandwidth and training size.

##### 5.2.1 The impact of bandwidth

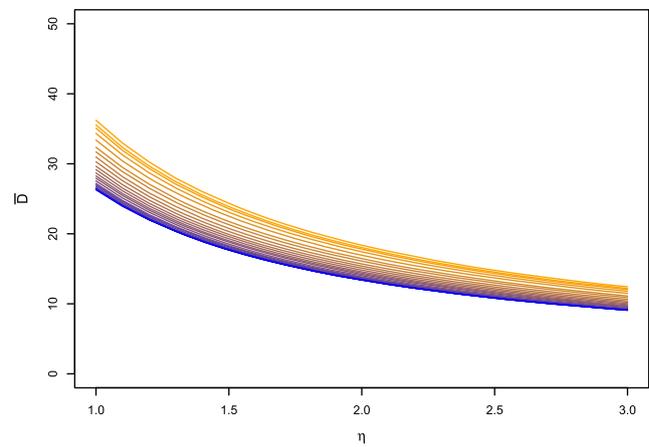
As shown in Fig. 5a, increasing the window size can increase the power of detecting small changes while leading to a slight delay in detecting large changes. Although increasing window size will increase the storage cost, it will not significantly



(a) Transmission cost for  $d = 100$  data streams.



(b) average detection delay versus fixed  $\delta$ .



(c) average detection delay versus  $\eta$ .

**Fig. 4** The average number of messages transmitted to the centre (top) and average detection delay across varying mean shifts (bottom). Results are obtained when  $m = 200, h = 100, T = 10,000, \tau = 5000, \alpha = 0.05$ . Each line corresponds to a different local threshold,

which is labelled on the top right. The colour changes from orange to blue as the local thresholds increase from 0 to 5.2. When the local threshold is 5.2, the global threshold will be 0. So all possible combinations of thresholds are covered. (Color figure online)

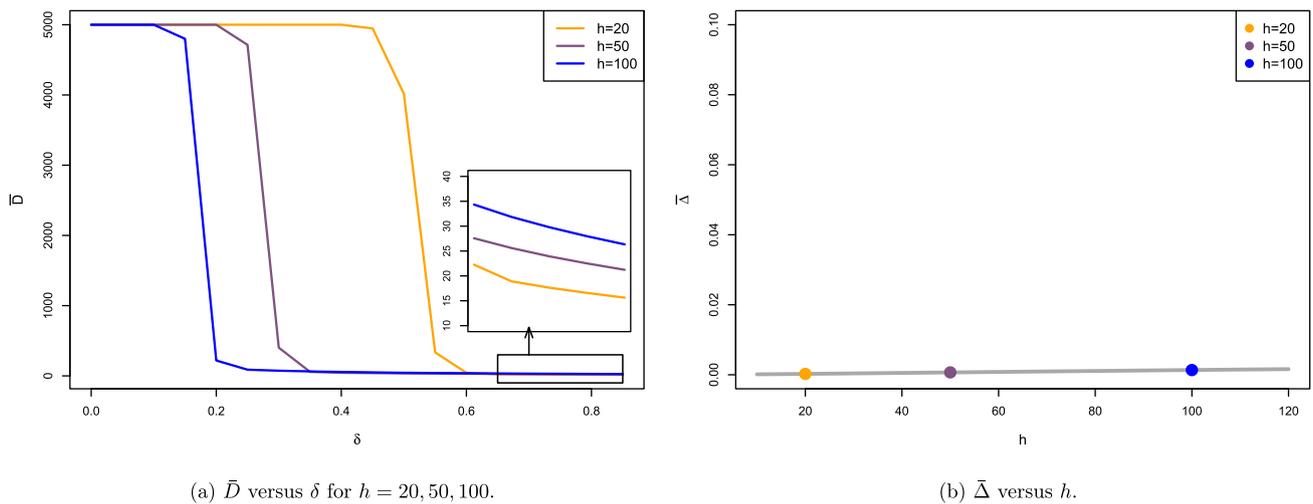
increase the transmission cost as shown in Fig. 5b. This drive us to think about whether we can improve the ability of distributed MOSUM with a large threshold to detect small changes by increasing the window size. Ideally, we would expect distributed MOSUM with increased window size can achieve similar performance as the idealistic setting.

*Recovering detectability*

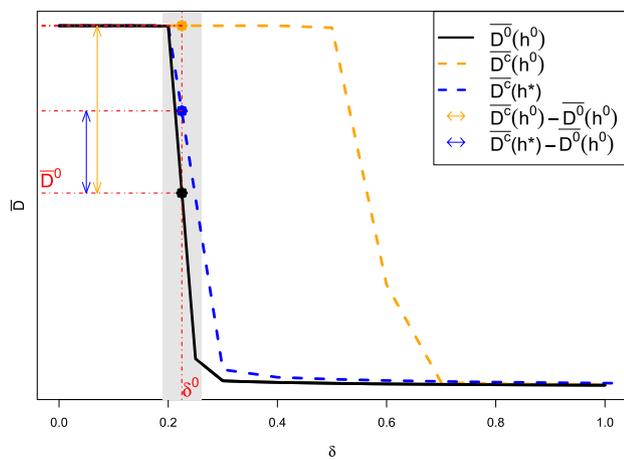
For simplicity, we denote that the default window size for centralized MOSUM is  $h^0$ , and  $h^*$  is the smallest window size that would allow distributed MOSUM to have similar performance as the idealistic setting. It is difficult to develop a neat theoretical formula between  $h^*$  and  $h^0$ . But we can approximately find  $h^*$  under alternatives by simulation. Our

idea can be summarized as follows, and Fig. 6 is the graphic explanation:

- The behaviour of  $\bar{D}$  will decrease dramatically when the mean shift is within a certain range (gray area). Therefore, we can find the median or mean  $\delta$  of this certain range, denoted by  $\delta^0$ . Also, the corresponding ADD  $\bar{D}^0$  can be calculated.
- Fix  $\delta^0$ , calculate  $\bar{D}^{cLocal}(h)$  iteratively for distributed MOSUM, where  $h \in [h^0, m]$ .
- The optimal window size  $h^* = \arg \min \{ \bar{D}^{cLocal}(h) - \bar{D}^0(h^0) \}$ . See blue arrow ( $h^*$ ) is shorter than yellow arrow (h).



**Fig. 5** The influence of window size. Results are obtained over 1000 replications and take  $m = 200, d = 100, T = 10,000, \tau = 5000, \alpha = 0.05, c_{Local} = 3.44$



**Fig. 6** An graphic explanation of our proposed idea. Black line is the ADD for centralized MOSUM with window size  $h$ . Yellow dashed line is the ADD for distributed MOSUM with window size  $h$ ; while blue dashed is the ADD for distributed MOSUM with window size  $h^*$ . (Color figure online)

Figure 7 displays the simulation results that, for distributed regime, we can recover the same detectability of the centralized statistic by inflating  $h$ .

### 5.2.2 The impact of the training dataset

Fix the bandwidth  $h$ , the impact of the size of the training dataset can be investigated. Table 4 gives the thresholds, empirical size, and mean square errors (MSE) of estimated baseline parameters in our simulation. As we expected, the larger the training size is, the more accurate estimators are Fig. 8 indicates that overall the detection powers of four different sizes of training datasets are similar. A larger training size could slightly increase the detection power when

detecting small changes, which is attributed to more accurate estimators. Thus, in the real application, it is beneficial to choose a large-size training dataset because it is not expensive that can be done offline.

### 5.3 The violation of the independence assumption

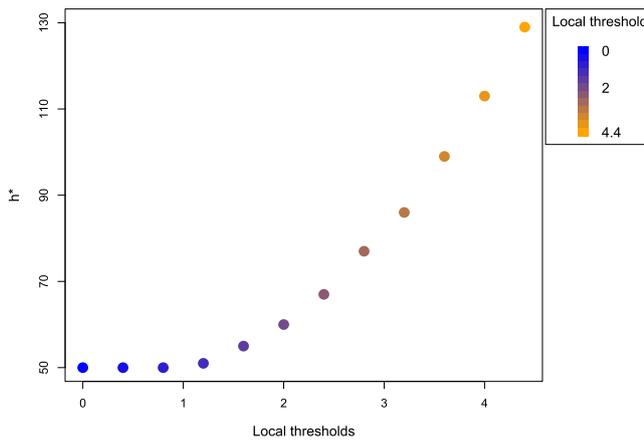
Before, we assume that there is temporal independence among observations. However, it may not always hold in the real application. This section will investigate the performance when this assumption is violated. Here we measure our algorithm under AR(1) noise process, that is

$$X_{i,t} = \delta_{i,t} \mathbb{1} \{t > \tau\} + \epsilon_{i,t},$$

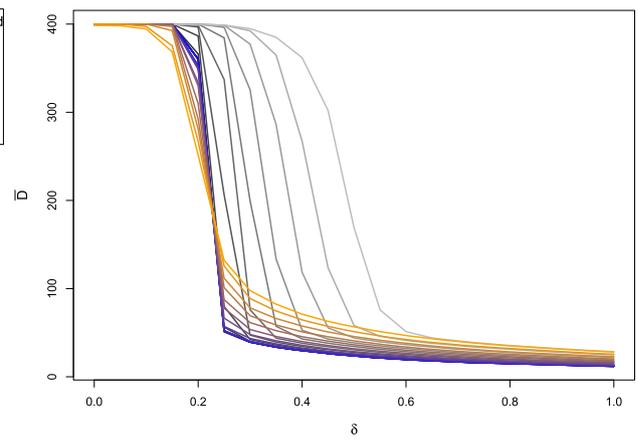
where  $\epsilon_{i,t} = \phi \epsilon_{i,t-1} + v_t$  with  $v_t \sim N(0, 1)$ .  $|\phi| < 1$  is used to measure the strength of the auto-correlation.

Auto-correlation will inflate the variance of data. There are two possible ways to handle this problem. The first one is to estimate the long-run variance as shown in Sect. 3.2. And one can also inflate the thresholds. We measure the false positives, average detection delay and the number of transmitted messages with fixed type-I-error of these two solutions under different scenarios. For better comparison, we also show the result of MOSUM without any adjustment. This will give us hints that to what extent our method fails to detect the change if we ignore the auto-correlation.

As Table 5 shows, our proposed method without adjustment can lose the ability to detect changes when introducing auto-correlation, that it fails to detect the change and always alarms. The performances of MOSUM with inflating thresholds are generally better than MOSUM with LRV since the former can detect the change in most scenarios. However, for those scenarios that the MOSUM with LRV can detect



(a)  $h^*$  found by our idea



(b) average detection delay for distributed MOSUM (fixed  $h = 50$ , gray line) and distributed MOSUM ( $h = h^*$ , colored line). Different lines corresponds to different  $c_{Local}$

**Fig. 7** An simple example showing that distributed MOSUM could approximate the detection power of centralized MOSUM by inflating window size. Results are obtained over 500 replications and take  $m =$

200,  $d = 100$ ,  $T = 1000$ ,  $\tau = 600$ ,  $\alpha = 0.05$ , and  $c_{Local} \in [0, 4.4]$ . When  $c_{Local} = 4.4$ ,  $c_{Global} = 0$ . So all possible local thresholds are covered. For centralized setting, window size  $h^0 = 50$

**Table 4** Empirical size, and MSE for estimated mean and standard deviation, results averaged over one thousand replications with  $c_{Local} = 3.44$ ,  $h = 50$ ,  $T = 6000$  and  $\alpha = 0.05$

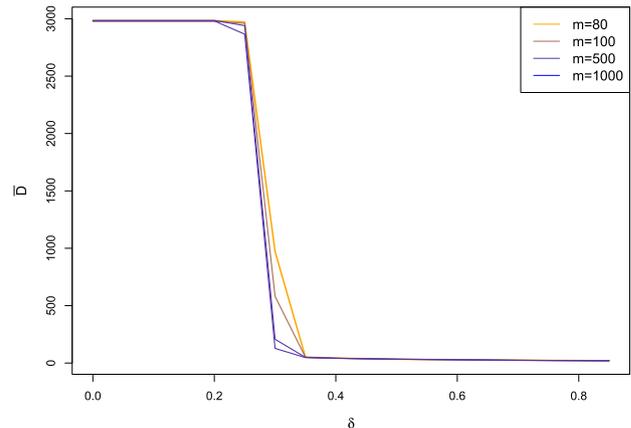
	$m = 80$	$m = 100$	$m = 500$	$m = 1000$
$c_{Global}$	9.039	8.159	6.014	5.708
Empirical size	0.007	0.007	0.009	0.006
MSE for mean	0.0125	0.01	0.002	0.001
MSE for SD	0.006	0.005	0.0001	0.0001

(usually  $\delta$  is not small and  $\phi$  is not large), it always has the lowest transmission cost and reasonable detection power. For example, when  $p = 100$ ,  $\delta = 1$ , and  $\phi = 0.25$ , both solutions have similar false positive rates and average detection delay, while MOSUM with LRV has lower transmission cost. It is surprising that estimating LRV has the lowest false positive rates and average detection delay when  $\phi = 0$  and  $p = 100/50$ . This may be because it underestimates the variance.

However, when the auto-correlation is serious, it is not appropriate to apply our method to the raw data. Instead, it is more reasonable to apply our method after pre-processing the data, such as the residuals of AR models.

### 6 Conclusion

Within this paper, we proposed an online communication-efficient distributed changepoint detection method, and it can achieve similar performance as an idealistic setting but save many transmission costs. Numerically, we show that the local



**Fig. 8**  $\bar{D}$  versus  $\delta$  when varying the size of training dataset. Result averaged over 500 replications with  $\alpha = 0.05$ ,  $c_{Local} = 3.44$ ,  $T = 6000$ ,  $\tau = 3000$  and  $h = 50$ . The corresponding global thresholds are shown in Table 4

threshold and window size have an impact on the performance of our algorithm, and there is a trade-off in choosing a local threshold and window size. In application, we recommend choosing a large local threshold in general cases. But when the change is extremely small, the choice of the local threshold depends on the communication and storage budgets. If the communication budget is much more limited, choosing a large threshold with a large window size is sensible. If the storage cost is much more expensive, choosing a small threshold with small window size will approximately achieve the idealistic performance.

The violation of independent assumptions will negatively affect the power of our proposed method. We tried

**Table 5** Results are obtained over 1000 replications with  $T = 10,000, m = 200, h = 100, \tau = 5000, d = 100, c_{Local} = 3.44,$  and  $\alpha = 0.05$  for all three methods

$\phi$	Methods	$p = 100$						$p = 50$						$p = 10$											
		$\delta = 0.5$		$\delta = 1$		$\delta = 0.5$		$\delta = 1$		$\delta = 0.5$		$\delta = 1$		$\delta = 0.5$		$\delta = 1$		$\delta = 0.5$		$\delta = 1$					
		FP	D	Trans	D	FP	Trans	FP	Trans	FP	D	Trans	D	FP	Trans	FP	D	Trans	D	FP	Trans	FP	D	Trans	D
0	Constant	<b>0.30</b>	<b>90.67</b>	<b>0.03</b>	<b>45.75</b>	<b>0.02</b>	<b>0.29</b>	<b>116.68</b>	<b>0.03</b>	<b>50.07</b>	<b>0.03</b>	<b>0.23</b>	<b>3767.69</b>	<b>0.18</b>	<b>63.09</b>	<b>0.03</b>	<b>63.09</b>	<b>0.03</b>	<b>0.03</b>	<b>0.03</b>	<b>0.23</b>	<b>3767.69</b>	<b>0.18</b>	<b>63.09</b>	<b>0.03</b>
	Inflating	<b>0.30</b>	<b>90.67</b>	<b>0.03</b>	<b>45.75</b>	<b>0.02</b>	<b>0.29</b>	<b>116.68</b>	<b>0.03</b>	<b>50.07</b>	<b>0.03</b>	<b>0.23</b>	<b>3767.69</b>	<b>0.18</b>	<b>63.09</b>	<b>0.03</b>	<b>63.09</b>	<b>0.03</b>	<b>0.03</b>	<b>0.03</b>	<b>0.23</b>	<b>3767.69</b>	<b>0.18</b>	<b>63.09</b>	<b>0.03</b>
	LRV	<b>0.19</b>	<b>85.85</b>	<b>0.05</b>	<b>43.41</b>	<b>0.05</b>	<b>0.28</b>	<b>105.74</b>	<b>0.06</b>	<b>48.42</b>	<b>0.06</b>	<b>0.32</b>	<b>4198.59</b>	<b>0.34</b>	<b>64.32</b>	<b>0.05</b>	<b>64.32</b>	<b>0.05</b>	<b>0.05</b>	<b>0.05</b>	<b>0.32</b>	<b>4198.59</b>	<b>0.34</b>	<b>64.32</b>	<b>0.05</b>
0.25	Constant	63.49	79.82	0.19	40.36	0.18	62.04	92.62	45.90	0.19	61.95	2059.06	0.20	62.85	0.18	62.85	0.18	62.85	0.18	0.18	61.95	2059.06	0.20	62.85	0.18
	Inflating	<b>0.33</b>	<b>104.78</b>	<b>0.22</b>	<b>49.04</b>	<b>0.20</b>	<b>0.21</b>	<b>1413.09</b>	<b>55.98</b>	<b>0.76</b>	<b>0.40</b>	<b>5000.00</b>	<b>0.48</b>	<b>81.90</b>	<b>0.21</b>	<b>81.90</b>	<b>0.21</b>	<b>81.90</b>	<b>0.21</b>	<b>0.20</b>	<b>0.40</b>	<b>5000.00</b>	<b>0.48</b>	<b>81.90</b>	<b>0.21</b>
	LRV	<b>0.30</b>	1294.64	0.21	<b>57.21</b>	<b>0.05</b>	<b>0.21</b>	4228.80	<b>63.95</b>	<b>0.30</b>	<b>0.22</b>	4995.26	<b>0.09</b>	<b>85.23</b>	<b>0.05</b>	<b>85.23</b>	<b>0.05</b>	<b>85.23</b>	<b>0.05</b>	<b>0.05</b>	<b>0.22</b>	4995.26	<b>0.09</b>	<b>85.23</b>	<b>0.05</b>
0.5	Constant	499.42	64.17	1.20	33.07	1.20	507.02	78.94	40.47	1.21	503.93	764.67	1.14	64.74	1.20	64.74	1.20	64.74	1.20	1.21	503.93	764.67	1.14	64.74	1.20
	Inflating	<b>0.57</b>	4663.87	5.15	<b>60.51</b>	<b>1.26</b>	<b>0.34</b>	5000.00	<b>73.66</b>	<b>3.06</b>	<b>0.59</b>	5000.00	<b>1.10</b>	5000.00	4.28	5000.00	4.28	5000.00	4.28	<b>1.28</b>	<b>0.59</b>	5000.00	<b>1.10</b>	5000.00	4.28
	LRV	<b>0.43</b>	4996.68	0.09	<b>83.57</b>	<b>0.05</b>	<b>0.38</b>	5000.00	5000.00	<b>97.18</b>	<b>0.06</b>	<b>0.59</b>	5000.00	<b>0.03</b>	3878.04	<b>0.35</b>	3878.04	<b>0.35</b>	3878.04	<b>0.06</b>	<b>0.59</b>	5000.00	<b>0.03</b>	3878.04	<b>0.35</b>
0.75	Constant	4219.42	5.74	8.35	3.99	8.35	4225.67	7.56	5.35	8.35	4228.56	8.25	6.94	8.37	8.37	6.94	8.37	8.37	8.35	4228.56	8.25	6.94	8.37	8.37	8.37
	Inflating	<b>0.76</b>	5000.00	10.19	<b>498.83</b>	<b>10.43</b>	0.85	5000.00	5000.00	5000.00	7.84	5000.00	6.02	5000.00	7.59	5000.00	7.59	5000.00	7.59	15.63	0.69	5000.00	6.02	5000.00	7.59
	LRV	0.64	5000.00	0.05	4980.28	0.25	0.59	5000.00	5000.00	5000.00	0.04	0.78	5000.00	0.04	5000.00	0.06	5000.00	0.06	5000.00	0.14	0.78	5000.00	0.04	5000.00	0.06

The bold are labelled when both the false positive rates and average detection delay are small

to solve this problem by inflating thresholds or estimating the long-run variance. Both ways can, to some extent, improve our algorithm when the auto-correlation problem is not severe. However, both approaches fail to detect changes in highly auto-correlated data. Therefore, one of the future research directions is how to detect change within highly auto-correlated data in real-time.

**Acknowledgements** Yang gratefully acknowledges the financial support of the EPSRC via the STOR-i Centre for Doctoral Training (EP/S022252/1). This research was also supported by EPSRC grant EP/R004935/1 (Eckley) together with financial support from BT Research (Eckley, Fearnhead, Yang). The authors are also grateful to Lawrence Bardwell who played a key role in inspiring this work, and Dave Yearling (BT) for several helpful conversations that helped shape this research.

**Author Contributions** All authors contributed to the main ideas and the writing of the paper. Ziyang Yang developed all the code and performed the simulation study.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## A Proofs

### A.1 Proof of Theorem 2

Squaring and expanding the weighted global MOSUM statistic in Eq. (3.8) for the two different cases gives

$$\begin{aligned} & (w(k, h)\mathcal{T}(m, k, h))^2 \\ &= \begin{cases} \sum_{i=1}^d (w(k, h)\mathcal{T}_i(m, k, h))^2 & \text{dense case,} \\ \sum_{i=1}^d (w(k, h)\mathcal{T}_i(m, k, h) \mathbb{1}_{\{w(k, h)\mathcal{T}_i(m, k, h) > c_{\text{Local}}\}})^2 & \text{sparse case.} \end{cases} \end{aligned} \tag{A.1}$$

From Theorem 1 with  $k = ht$ , the weighted local MOSUM and its hard-thresholded counterpart have limit

$$\begin{aligned} & \lim_{m \rightarrow \infty} (w(k, h)\mathcal{T}_i(m, k, h))^2 \\ & \xrightarrow{\mathcal{D}} \rho(t)^2 Z_i(t)^2, \\ & \lim_{m \rightarrow \infty} (w(k, h)\mathcal{T}_i(m, k, h) \mathbb{1}_{\{w(k, h)\mathcal{T}_i(m, k, h) > c_{\text{Local}}\}})^2 \\ & \xrightarrow{\mathcal{D}} \rho(t)^2 Z_i(t)^2 \mathbb{1}_{\{\rho(t)Z_i(t) > c_{\text{Local}}\}}, \end{aligned} \tag{A.2}$$

where  $Z_i(t)$  is defined in Eq. (4.1). Taking the limit of (A.1) as  $m \rightarrow \infty$  gives the result.

### A.2 Proof of Theorem 3

It is enough to show that

$$(w(\tilde{k}, h)\mathcal{T}(m, \tilde{k}, h))^2 \xrightarrow{\mathcal{P}} \infty,$$

for a time  $\tilde{k}$  later than the change point  $k^*$  but before the end of the monitoring time  $\lfloor m\tilde{T} \rfloor$ .

Since  $k^* \leq \lfloor hv \rfloor$ , we can choose  $\tilde{k} = \lfloor x_0 h \rfloor + h$  where  $v < x_0 < \tilde{T} \frac{m}{h} - 1$  so that  $k^* \leq \tilde{k} - h$ .

If data stream  $i$  is affected by the change, so that  $i \in \mathcal{S}$  and  $\delta_i \neq 0$  then

$$\begin{aligned} \frac{1}{h} \mathcal{T}_i(m, \tilde{k}, h) &= \frac{1}{h\hat{\sigma}_i} \left| \sum_{t=m+\lfloor x_0 h \rfloor+1}^{m+\lfloor x_0 h \rfloor+h} (X_{i,t} - \hat{\mu}_i) \right| \\ &= \frac{1}{h\hat{\sigma}_i} \left| \sum_{t=m+\lfloor x_0 h \rfloor+1}^{m+\lfloor x_0 h \rfloor+h} (\mu_i + \delta_i + \epsilon_{i,t} - \hat{\mu}_i) \right| \\ &= \frac{1}{h\hat{\sigma}_i} \left| h(\mu_i - \hat{\mu}_i) + h\delta_i + \sum_{t=m+\lfloor x_0 h \rfloor+1}^{m+\lfloor x_0 h \rfloor+h} \epsilon_{i,t} \right| \\ &= \frac{1}{\hat{\sigma}_i} \left| \mu_i - \hat{\mu}_i + \delta_i + \frac{1}{h} \sum_{t=m+\lfloor x_0 h \rfloor+1}^{m+\lfloor x_0 h \rfloor+h} \epsilon_{i,t} \right| \\ &= \frac{1}{\hat{\sigma}_i} |\delta_i + o_P(1)|. \end{aligned}$$

On the other hand if  $i \notin \mathcal{S}$ ,

$$\frac{1}{h} \mathcal{T}_i(m, \tilde{k}, h) = \frac{1}{\hat{\sigma}_i} |o_P(1)|.$$

For the global dense procedure

$$\begin{aligned} (w(\tilde{k}, h)\mathcal{T}(m, \tilde{k}, h))^2 &= w(\tilde{k}, h)^2 \sum_{i=1}^d \mathcal{T}_i(m, \tilde{k}, h)^2 \\ &= \left( \frac{1}{\sqrt{h}} \rho \left( \frac{\tilde{k}}{h} \right) \right)^2 \times h^2 \times \sum_{i=1}^d \left( \frac{1}{h} \mathcal{T}_i(m, \tilde{k}, h) \right)^2 \\ &= h\rho \left( \frac{\tilde{k}}{h} \right)^2 \sum_{i=1}^d \left( \frac{1}{h} \mathcal{T}_i(m, \tilde{k}, h) \right)^2 \\ &= h\rho(x_0 + 1 + o(1))^2 \sum_{i \in \mathcal{S}} \left( \frac{\delta_i}{\hat{\sigma}_i} \right)^2 + o_P(1) \xrightarrow{\mathcal{P}} \infty, \end{aligned}$$

as  $m, h \rightarrow \infty$ .

For the global sparse procedure the local MOSUM's  $w(\tilde{k}, h)T(m, \tilde{k}, h)$  are hard thresholded. Since these diverge to infinity individually then the same argument used for the dense procedure applies.

## References

- Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742* (2007)
- Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., Ming, H.: Ad-iot: anomaly detection of iot cyberattacks in smart city using machine learning. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0305–0310 (2019)
- Appadwedula, S., Veeravalli, V.V., Jones, D.L.: Energy-efficient detection in sensor networks. *IEEE J. Sel. Areas Commun.* **23**(4), 693–702 (2005)
- Aue, A., Horváth, L., Kühn, M., Steinebach, J.: On the reaction time of moving sum detectors. *J. Stat. Plan. Inference* **142**(8), 2271–2288 (2012)
- Banerjee, T., Veeravalli, V.V.: Data-efficient quickest change detection in sensor networks. *IEEE Trans. Signal Process.* **63**(14), 3727–3735 (2015)
- Chan, H.P.: Optimal sequential detection in multi-stream data. *Ann. Stat.* **45**(6), 2736–2763 (2017)
- Chen, Y., Wang, T., Samworth, R.J.: High-dimensional, multiscale online changepoint detection. *J. R. Stat. Soc. Ser. B Stat Methodol.* **84**(1), 234–266 (2022)
- Cho, H., Fryzlewicz, P.: Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *J. R. Stat. Soc. Ser. B Stat Methodol.* **77**, 475–507 (2014)
- Enikeeva, F., Harchaoui, Z.: High-dimensional change-point detection under sparse alternatives. *Ann. Stat.* **47**(4), 2051–2079 (2019)
- Fisch, A.T.M., Eckley, I.A., Fearnhead, P.: A linear time method for the detection of collective and point anomalies. *Stat. Anal. Data Min.: ASA Data Sci. J.* **15**(4), 494–508 (2022)
- Gallant, A.R.: *Nonlinear Statistical Models*. Wiley, London (2009)
- Gösmann, J., Stoehr, C., Heiny, J., Dette, H.: Sequential change point detection in high dimensional time series. *Electron. J. Stat.* **16**(1), 3608–3671 (2022)
- Horváth, L., Hušková, M.: Change-point detection in panel data. *J. Time Ser. Anal.* **33**(4), 631–648 (2012)
- Horváth, L., Kühn, M., Steinebach, J.: On the performance of the fluctuation test for structural change. *Seq. Anal.* **27**(2), 126–140 (2008)
- Kengne, W., Ngongo, I.S.: Inference for nonstationary time series of counts with application to change-point problems. *Ann. Inst. Stat. Math.* **74**(4), 801–835 (2022)
- Kiefer, N.M., Vogelsang, T.J.: Heteroskedasticity-autocorrelation robust standard errors using the Bartlett kernel without truncation. *Econometrica* **70**(5), 2093–2095 (2002)
- Kiefer, N.M., Vogelsang, T.J.: Heteroskedasticity-autocorrelation robust testing using bandwidth equal to sample size. *Economet. Theor.* **18**(6), 1350–1366 (2002)
- Kirch, C., Kamgaing, J.T.: On the use of estimating functions in monitoring time series for change points. *J. Stat. Plan. Inference* **161**, 25–49 (2015)
- Kirch, C., Weber, S.: Modified sequential change point procedures based on estimating functions. *Electron. J. Stat.* **12**(1), 1579–1613 (2018)
- Kovács, S., Bühlmann, P., Li, H., Munk, A.: Seeded binary segmentation: a general methodology for fast and optimal changepoint detection. *Biometrika* **110**(1), 249–256 (2023)
- Leisch, F., Hornik, K., Kuan, C.-M.: Monitoring structural changes with the generalized fluctuation test. *Economet. Theor.* **16**(6), 835–854 (2000)
- Liu, K., Zhang, R., Mei, Y.: Scalable SUM-shrinkage schemes for distributed monitoring large-scale data streams. *Stat. Sin.* **29**, 1–22 (2019)
- Mei, Y.: Information bounds and quickest change detection in decentralized decision systems. *IEEE Trans. Inf. Theory* **51**(7), 2669–2681 (2005)
- Mei, Y.: Efficient scalable schemes for monitoring a large number of data streams. *Biometrika* **97**(2), 419–433 (2010)
- Mei, Y.: Quickest detection in censoring sensor networks. In: 2011 IEEE International Symposium on Information Theory Proceedings, pp. 2148–2152 (2011)
- Newey, W.K., West, K.D.: *A simple, positive semi-definite, heteroskedasticity and autocorrelation-consistent covariance matrix*. National Bureau of Economic Research Cambridge, Mass., USA (1986)
- Pinto, G., Castor, F.: Energy efficiency: a new concern for application software developers. *Commun. ACM* **60**(12), 68–75 (2017)
- Rago, C., Willett, P., Bar-Shalom, Y.: Censoring sensors: a low-communication-rate scheme for distributed detection. *IEEE Trans. Aerosp. Electron. Syst.* **32**(2), 554–568 (1996)
- Romano, G., Eckley, I.A., Fearnhead, P., Rigai, G.: Fast online change-point detection via functional pruning cusum statistics. *J. Mach. Learn. Res.* **24**, 1–36 (2023)
- Tartakovsky, A., Nikiforov, I., Basseville, M.: *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. CRC Press (2014)
- Tartakovsky, A.G., Kim, H.: Performance of certain decentralized distributed change detection procedures. In: 2006 9th International Conference on Information Fusion, pp. 1–8. IEEE (2006)
- Tartakovsky, A.G., Veeravalli, V.V.: An efficient sequential procedure for detecting changes in multichannel and distributed systems. In: Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997), vol. 1, pp. 41–48 (2002)
- Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. *Signal Process.* **167**, 107299 (2020)
- Tveten, M., Eckley, I.A., Fearnhead, P.: Scalable change-point and anomaly detection in cross-correlated data with an application to condition monitoring. *Ann. Appl. Stat.* **16**(2), 721–743 (2022)
- Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., Nikolopoulos, D.S.: Challenges and opportunities in edge computing. In: 2016 IEEE International Conference on Smart Cloud (SmartCloud), pp. 20–26 (2016)
- Veeravalli, V.V.: Decentralized quickest change detection. *IEEE Trans. Inf. Theory* **47**(4), 1657–1665 (2001)
- Wang, T., Samworth, R.J.: High dimensional change point estimation via sparse projection. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **80**(1), 57–83 (2018)
- Ward, K., Dilillo, G., Eckley, I.A., Fearnhead, P.: Poisson-FOCuS: an efficient online method for detecting count bursts with application to gamma ray burst detection. *J. Am. Stat. Assoc.*, page (to appear) (2024)
- Weber, S.M.: *Change-Point Procedures for Multivariate Dependent Data*. Ph.D. thesis, Karlsruhe Institut für Technologie (KIT) (2017)
- White, H., Domowitz, I.: Nonlinear regression with dependent observations. *Econom.: J. Econom. Soc.*, pp. 143–161 (1984)
- Wu, H., Hu, J., Sun, J., Sun, D.: Edge computing in an IoT base station system: reprogramming and real-time tasks. *Complexity*, 2019:4027638:1–4027638:10 (2019)
- Xie, Y., Siegmund, D.: Sequential multi-sensor change-point detection. *Ann. Stat.* **41**(2), 670–692 (2013)

- Yau, C.Y., Sze Him Isaac, L., Ng, W.L.: Sequential change-point detection in time series models based on pairwise likelihood. *Stat. Sinica*, **27** (2017)
- Yu, Y., Madrid Padilla, O.H., Wang, D., Rinaldo, A.: A note on online change point detection. *Seq. Anal.* **42**(4), 438–471 (2023)
- Zeileis, A., Leisch, F., Kleiber, C., Hornik, K.: Monitoring structural change in dynamic econometric models. *J. Appl. Economet.* **20**(1), 99–121 (2005)
- Zhang, R., Mei, Y.: Asymptotic statistical properties of communication-efficient quickest detection schemes in sensor networks. *Seq. Anal.* **37**(3), 375–396 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.