CHANGE-POINT DETECTION IN TIME SERIES: A STUDY OF ONLINE METHODS APPLIED TO COMPUTER NETWORK MEASUREMENTS

Cleiton Moya de Almeida

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Rio de Janeiro
Julho de 2024

CHANGE-POINT DETECTION IN TIME SERIES: A STUDY OF ONLINE METHODS APPLIED TO COMPUTER NETWORK MEASUREMENTS

Cleiton Moya de Almeida

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Aprovada por: Prof. Edmundo Albuquerque de Souza e Silva
Prof. Rosa Maria Meri Leão
Prof. Antonio Augusto de Aragão Rocha
Prof. Magnos Martinello

RIO DE JANEIRO, RJ – BRASIL
JULHO DE 2024

Moya de Almeida, Cleiton

Change-point detection in time series: a study of online methods applied to computer network measurements/Cleiton Moya de Almeida. – Rio de Janeiro: UFRJ/COPPE, 2024.

XI, 66 p.: il.; 29, 7cm.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2024.

Referências Bibliográficas: p. 60 – 66.

1. Change-point detection. 2. Anomaly detection. 3. Time series. 4. Online machine learning. 5. Computer network measurements. I. Albuquerque de Souza e Silva, Edmundo *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Acknowledgments

I ask for a license to express gratitude in my native language.

Primeiramente, agradeço à minha família todo o suporte e incentivo recebidos.

Agradeço aos professores Edmundo e Rosa os ensinamentos, suporte e ótima orientação. Agradeço também aos demais professores da UFRJ os conhecimentos obtidos nas disciplinas que tive a oportunidade de cursar.

Agradeço aos colegas do LAND as discussões e colaborações. Em especial, agradeço ao Daniel Atkinson a parceria e o desenvolvimento da plataforma de coleta de dados que utilizamos em nossos trabalhos. À Isabela Siqueira, agradeço as revisões e sugestões no texto.

Agradeço à Gigalink a parceria que possibilitou a obtenção de dados reais. Também agradeço à Rede Nacional de Ensino e Pesquisa (RNP) e ao projeto Measurement Lab (M-Lab) pelo suporte, infraestrutura e acordo de cooperação M-Lab/RNP que possibilitou ampliar a coleta dos dados.

Por fim, agradeço ao Conselho Nacional de Pesquisas (CNPq) e ao Google Research o suporte financeiro para o desenvolvimento deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CHANGE-POINT DETECTION IN TIME SERIES: A STUDY OF ONLINE METHODS APPLIED TO COMPUTER NETWORK MEASUREMENTS

Cleiton Moya de Almeida

Julho/2024

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

Nesta dissertação, estudamos métodos clássicos e do estado da arte para a identificação de pontos de mudanças, de modo online, em séries temporais de latência e vazão de rede. Examinamos os modelos clássicos Shewhart, EWMA e CUSUM e mostramos que implementações básicas podem não ser adequadas para detectar tais pontos. Propomos, então, estratégias simples para contornar este problema. Estudamos também os métodos BOCD e RRCF (este originalmente proposto para detecção de anomalias) e, de forma análoga aos métodos clássicos, propomos modificações simples que aumentaram o desempenho nos conjuntos de dados analisados. Avaliamos ainda um novo método para detecção de mudanças, o VWCD, que oferece flexibilidade, interpretabilidade e permitiu a detecção de pontos de mudança com maior precisão.

Aplicamos os métodos estudados a um conjunto de dados não rotulados de latência e vazão de rede, construído por nós usando a ferramenta M-Lab NDT, e apresentamos uma aplicação simples que pode ser usada para o monitoramento da qualidade de serviço de rede. Além disso, avaliamos o desempenho dos métodos utilizando um conjunto de medições de latência com pontos de mudanças rotulados. Os algoritmos propostos, incluindo o VWCD, apresentaram desempenho competitivo com um algoritmo offline do estado da arte, o Pelt não-paramétrico.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CHANGE-POINT DETECTION IN TIME SERIES: A STUDY OF ONLINE METHODS APPLIED TO COMPUTER NETWORK MEASUREMENTS

Cleiton Moya de Almeida

July/2024

Advisors: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Department: Systems Engineering and Computer Science

In this work, we study classical and state-of-the-art methods to identify change points in time series of network latency and throughput, in an online setting. First, we study the classic methods of Shewhart, EWMA and CUSUM, concluding that their straightforward implementations may not be suitable for detecting such points. We then present simple strategies to mitigate this problem. We also study the methods BOCD, RRCF (this was initially proposed for anomaly detection) and, similarly to the classical methods, we propose simple strategies that improve their performance in the studied datasets. We also introduce a novel change-point detection method, the VWCD, that offers flexibility and interpretability and increases the precision of the change-point detection.

We applied the methods to a non-labeled dataset of latency and throughput, built by us using the M-Lab NDT tool, and showed a simple application that can be used to access the network quality of service. Furthermore, we assessed the methods' performance using a latency dataset with labeled change points. The proposed algorithms, including the VWCD, showed competitive performance with a state-of-the-art offline algorithm, the non-parametric Pelt.

Contents

List of Figures

Abbreviations and Acronyms

2S-CUSUM	two-sided CUSUM
ADF	Augmented Dickey-Fuller
ARL	average run lenght
BOCD	Bayesian Online Changepoint Detection
CUSUM	Cumulative Sum
EWMA	Exponential Weighted Moving Average
FPR	false positive rate
GLR	generalized likelihood ratio
HMM	Hidden Markov Model
i.i.d.	independent and identically distributed
ISP	internet service provider
LLR	log-likelihood ratio
M-Lab	Measurement Lab
MAP	maximum a posteriori
MBIC	Modified Bayesian Information Criteria
MLE	maximum likelihood estimate
NDT	Network Diagnostic Tool
NP	Neyman-Pearson
Pelt	Pruned Exact Linear Time
Pelt-NP	Non-parametric Pelt
QoS	quality of service
r.v.	random variable
RNP	Rede Nacional de Ensino e Pesquisa
RRCF	Robust Random Cut Forest
RTT	round trip time
SPC	Statistical Process Control
TPR	true positive rate
VWCD	Voting Windows Changpoint Detection
WL-CUSUM	Window-Limited CUSUM

Chapter 1

Introduction

Change-point detection refers to locating points in time or position in a sequence where some data property, such as location, scale, or distribution, changes (FEARNHEAD and RIGAILL, 2019). This problem has been researched for decades and has applications in many areas, among them: quality control, target detection and tracking, navigation systems integrity monitoring, segmentation of signals and images, seismic data processing, mechanical systems integrity monitoring, finance and economics, computer network surveillance and security, smart grid monitoring, cyber-physical systems, sensor networks, social networks, epidemic detection, genomic signal processing, astronomy, neurophysiology and climatology (CHO and KIRCH, 2021; TARTAKOVSKY *et al.*, 2015; XIE *et al.*, 2021).

The research on change-point detection can be grouped in two main directions: online and offline settings. In the online setting, the data is acquired and processed in real-time or near real-time (a batch of fixed size is accumulated before the processing stage). The goal is to decide on a change-point with minimum delay and false alarm rate. On the other hand, in the offline setting, the data is fully available before applying the method. In this work, we focus on online algorithms despite using an offline method as a base for performance comparison.

Change-point detection methods typically do not require training labels. While there are some supervised approaches using traditional machine learning methods (AMINIKHANGHAHI and COOK, 2017), they are uncommon. Change-point detection is usually performed using unsupervised learning methods (OLTEANU *et al.*, 2022), which is the focus of this work.

These methods can be further classified as univariate or multivariate. Classical methods Shewhart (SHEWHART, 1929), Exponential Weighted Moving Average (EWMA) (ROBERTS, 1959) and Cumulative Sum (CUSUM) (PAGE, 1954) are original univariate, although there are also multivariate versions available (LOWRY and MONTGOMERY, 1995). On the other hand, the Bayesian Online Change-point Detection (BOCD) (ADAMS and MACKAY, 2007) and the Robust Random

Cut Forest (RRCF) (GUHA *et al.*, 2016) are naturally suitable for multivariate data. Similarly, the new proposed method Voting Windows Changpoint Detection (VWCD) is immediately applicable to the multivariate case.

A related issue to change-point detection is anomaly detection. While they are treated interchangeable in some research, they have distinct concepts and goals. This work explores these differences and shows that the classic selected methods and the more recent BOCD cannot distinguish change-points from point anomalies. To address this, we propose straightforward strategies to enhance the performance of these algorithms. Furthermore, we study a recent tree-based method initially designed for anomaly detection, the RRCF, and propose a simple framework that allows the use of this algorithm also in the change-point detection task.

In addition to the algorithms above, we present and evaluate a new change-point detection method proposed by our group, the VWCD. This method is based on the generalized likelihood ratio (GLR) test, on the ensemble concept, and has the advantage of flexibility and easy interpretation of its hyperparameters and results.

1.1 Objectives

This work focus on online unsupervised methods for change-point detection in time series. Although many studies use synthetic data to evaluate change-point methods, our objective is restricted to real data time series. Specifically, using datasets of network measurements, we are interested in investigating the following questions:

- How do the classical statistical methods Shewhart, EWMA and CUSUM perform on network measurements (real data)? This same question applies to the most recent and highly cited BOCD method.
- Can the RRCF - a popular tree-based algorithm - be used to change-point detection? How does it perform on network measurement data?
- How does the new method proposed by our group, the VWCD, perform on network measurement data?
- How is the above methods' performance compared to offline methods?

1.2 Motivation

As the Introduction mentions, the change-point detection problem has applications in many areas. For this work, we are specifically interested in studying change points algorithms suitable for network measurements and quality of service (QoS) monitoring. In this scenario, the use of online and unsupervised algorithms is a

natural choice given the high amount of data generated and collected from networks and since labeling that data is complex and costly.

The choice of the classical methods Shewhart, EWMA and CUSUM is motivated by several reasons. First, following the Occam's Razor learning principle[1], simpler models are preferred whenever possible. Furthermore, these methods are widely used in various domains, are simple to understand and implement, have a solid theoretical ground and are computationally "cheap". Moreover, they're still being researched, with many recent updates and extensions in the literature.

On the other hand, the BOCD method is highly cited, has recent extensions, *e.g.* (ALTAMIRANO *et al.*, 2023) and was reported in a recent evaluation work with several datasets (BURG and WILLIAMS, 2020) as having the best performance among other competitors, even offline methods, when allowed to optimize its hyperparameters.

The choice to study the RRCF, in turn, is motivated by the fact that tree algorithms are popular and competitive across various tasks and domains. In addition, there are good implementation options in Python, *e.g.* (BARTOS *et al.*, 2019), Matlab, and it is also available in the Amazon Web Services, having been used with success in real-world applications, *e.g.*, (KRISHNAN, 2020). Finally, GUHA *et al.* (2016) suggest that, although the method was initially designed for anomaly detection, it could also be adapted to change-point detection.

One known characteristic of the traditional methods found in the literature is that they usually produce many spurious alarms STREIT *et al.* (2023); TARTAKOVSKY *et al.* (2013). Although this can be tolerated in certain applications or mitigated using complementary strategies (see, *e.g.*, TARTAKOVSKY *et al.* (2015)), in the task of QoS monitoring usually a high precision is desirable or even mandatory. For example, consider an internet service provider (ISP) that uses a system to detect QoS issues in their network. The system triggers alarms for the operators; in this case, a high number of false alarms could overload the operators and decrease their confidence in the system.

1.3 Methodology and dissertation outline

To investigate the questions raised in Section 1.1, we first perform a literature review to study the main concepts and related work (Chapter 2), followed by a study of the selected methods (Chapter 3). After that, in Chapter 4, we propose enhancements in the literature methods to improve their performance. Additionally, we also propose a framework for change-point detection using the RRCF method. In Chapter 5,

[1] "The simplest model that fits the data is also the most plausible." (ABU-MOSTAFA *et al.*, 2012, p. 167)

we introduce the new method proposed by our group and describe new strategies investigated.

To evaluate the performance of the selected methods (Chapter 6), we use two different datasets of real-data network measurements:

- NDT dataset: This dataset comprises 296 time-series of round trip time (RTT) and throughput (for both download and upload tests) that were collected by us using the M-Lab Network Diagnostic Tool (NDT) tool (GILL *et al.*, 2022) (real data), with a total of 45687 measurements. For this dataset, we do not have labels for the change points.

- Shao dataset: a labeled dataset of RTT time series provided in (SHAO *et al.*, 2017). The dataset comprises 50 time series with 408087 measurements and 1047 labeled change-points.

With the NDT dataset, we study the algorithms' performance by comparing the number of detected change points and verifying it visually to gain insights into the number of false alarms. We also compare the elapsed time of each method. Furthermore, we discuss some examples visually. In the experiment with the Shao dataset, we tune the hyperparameters using grid search and evaluate the performance using the Precision, Recall and F1 metrics. The results are compared to recent works and discussed in Chapter 6. Finally, the conclusions are stated in Chapter 7.

1.4 Contributions

The main contributions of this work are:

- We built a dataset with controlled M-Lab NDT tests executed in residential networks. The time series data and code used in the work are available in a public repository.

- We proposed extensions of the classical change-point detection, improving their performance for the studied series. Furthermore, we evaluated a recently proposed variant of CUSUM, the Window-Limited CUSUM (WL-CUSUM). Similarly, we proposed simple modifications to the BOCD and RRCF methods in order to make them more competitive with the state-of-the-art Non-parametric Pelt (Pelt-NP). To the best of our knowledge, this is the first work proposing a framework to use the RRCF method in the task of change-point detection. Also, to the best of our knowledge, this is the first work evaluating the WL-CUSUM with real data.

- We performed a detailed study of a new change-point detection method proposed by our group, firstly introduced in STREIT *et al.* (2023), the VWCD; we showed that it outperformed the Pelt-NP in terms of precision and false positive rate. We investigated different ways to aggregate the votes in a window and evaluated VWCD using real data from two distinct network measurements datasets.

Chapter 2

Background and related work

Several terms and research topics related to changes and anomaly identification are found in the literature, each with distinct or fuzzy nuances. In this section, we formulate the change-point detection problem and discuss the following related concepts: anomaly detection, outlier detection, noisy removal and signal segmentation. The goal is not to exhaust the themes or propose new concepts but to clarify the terminology and discuss related work.

2.1 The change-point detection problem

The change-point detection problem is usually formulated differently in the online and offline settings.

2.1.1 Offline setting

In the offline setting, the problem can be stated as an optimization problem for the time series (signal) segmentation. Consider a time series represented by a random process $\mathbf{X} = X_1, X_2, \ldots, X_t, \ldots X_T$, $X_t \in \mathbb{R}$. This time series is assumed to be piece-wise stationary, *i.e.*, some characteristic of the process change abruptly at some unknown instants $\{\tau_1, \tau_2 \ldots, \tau_k, \ldots . \tau_K\}$, $K \in [1, T]$. These change-points split the time series in $K + 1$ segments, and we define the k-th segment as $\mathbf{x}_{(\tau_{k-1}+1):\tau_k}$[1]. Then, the change-point detection problem falls into two categories: if the number of change-points is known a priori, the problem is only to estimate the indexes τ_k. If we don't know in advance the number of change-points (a more realistic scenario), we also have to estimate K (TRUONG *et al.*, 2020).

Let's consider the second case (unknown number of change-points) and let $\mathcal{C}(\cdot)$ be a cost function for the segments. Then, change-point detection can be stated as a penalized minimization problem (HAYNES, 2017):

[1] it's also usual in the literature to index the segment as $\mathbf{x}_{\tau_k:(\tau_{k+1}-1)}$.

$$Q(\mathbf{x},\beta) = \min_{\boldsymbol{\tau}_{1:K}} \left\{ \sum_{i=1}^{K+1} \left[\mathcal{C}\left(\mathbf{x}_{(\tau_{i-1}+1):\tau_i}\right) + \beta \right] \right\}, \tag{2.1}$$

where β is a penalty added to avoid over-fitting. It's also usual to restrict the minimum size of a segment (hyperparameter `min_seg` in Table A.2).

This formulation using cost functions is very general and can be employed with several approaches; see (TRUONG *et al.*, 2020) for a survey. In the Section 3.4, we introduce the Pruned Exact Linear Time (Pelt) algorithm (specifically, a non-parametric extension) that we use to compare the performance of our selected online methods.

2.1.2 Online (sequential) setting

In this section, we introduce the formulation used in the Sequential Analysis research area (TARTAKOVSKY *et al.*, 2015; WALD, 1947; XIE *et al.*, 2021), in which the classical models are derived. In this setting, the problem is also known as *quick change detection* (XIE *et al.*, 2021). In the subsequent chapters, we will also discuss other approaches when introducing the BOCD, RRCF and VWCD methods.

In the sequential setting, observations are made one at a time, and we are interested in detecting a change as soon as possible. So, at every new observation, we must decide to let the process continue or stop and raise an alarm of change. Formally, the problem can be stated in the following manner: given a series of observations $x_1, x_2, \ldots$ lets consider that a change occurs at $t = \tau$. Then, $x_1, x_2, \ldots, x_\tau$ follows one probability distribution and $x_{\tau+1}, x_{\tau+2}, \ldots$ follows another distribution[2]. The most simple and prevalent assumption is to consider that observations are i.i.d. and the post-change distribution does not depend on τ. In this case, the time series model becomes

$$p_\tau(\mathbf{x}) = \prod_{i=1}^{\tau} p_0\left(x_i\right) \times \prod_{i=\tau+1}^{T} p_1\left(x_i\right). \tag{2.2}$$

where p_τ is a joint distribution and $p_0(\cdot)$, $p_1(\cdot)$ are the marginal distributions.

The change-point variable can be modeled as an unknown deterministic number or a random variable (r.v.). So, two classes of sequential procedures are derived: bayesian and non-bayesian. Furthermore, the probability distributions can modeled using parametric or non-parametric approaches, thus deriving another two sub-classes of methods.

Also note that, in the sequential setting, the monitoring procedure can produce two possible outputs: a false alarm or a true change-point detected with a potential

[2]It is also usual to define τ as the first instant after the change

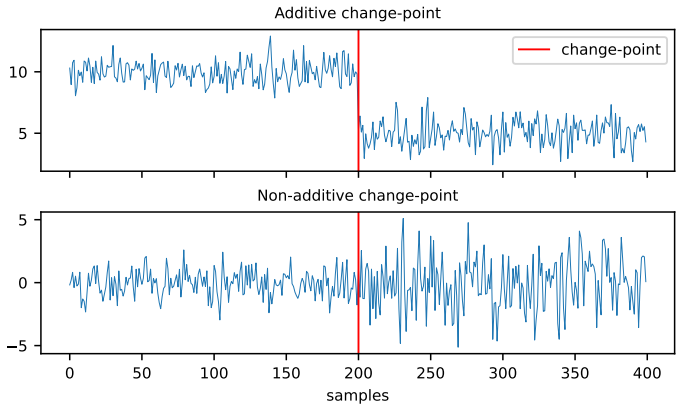


Figure 2.1: Example of change-point types (synthetic data). In the top subplot, the process mean changes from 10 to 5, characterizing and additive type change-point; In the bottom, there is an increase of the process variance (non-additive change-point).

delay. This leads to a natural trade-off problem: we want to detect the change-point with a low false alarm rate while keeping a low average detection delay. So, the main question tackled by the Sequential Change-point research area is how to formulate and solve this optimization problem.

2.1.3 Change-point types

TARTAKOVSKY *et al.* (2015) classify the change points in two types:

- Additive change: a change in the mean value of the sequence;
- Non-additive change: the change can occur in the signal or system's variance, correlations, spectral characteristics or dynamics. The non-additive change-points are typically more challenging to detect, even for humans.

The Fig. 2.1 illustrates these two types of change-point.

2.2 Anomaly, noise and outlier detection

Anomalies and *outliers* are frequently used interchangeably in the machine learning and data mining literature (AGGARWAL, 2017; OLTEANU *et al.*, 2022). For CHANDOLA *et al.* (2009), "*anomalies are patterns in data that do not conform to a well-defined notion of normal behavior*". On the other hand, HAWKINS (1980) defines an outlier as an "*observation which deviates so much from other observations as to arouse suspicions that a different mechanism generated it*". Another related concept, *noisy removal*, deals with data that is not interesting to the analyst (AGGARWAL, 2017; CHANDOLA *et al.*, 2009).

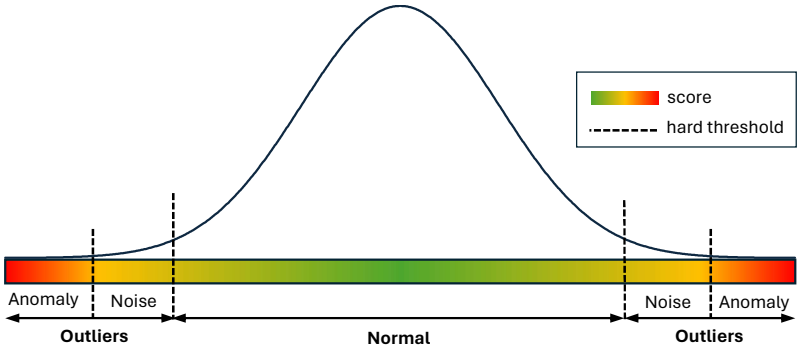


Figure 2.2: Relation of outliers, noise and anomaly. We use the normal distribution as the underlying model and the probabity density function as the score in this example. However, in general, the model and score may not be probabilistic. Elaborated by the author based on the Fig. 1.2 of (AGGARWAL, 2017)

AGGARWAL (2017) propose to relate the three concepts using the "outlierness" score, used by most of the methods. For the author, anomalies typically have a higher score than noise. Still, this separation may not be clear in some applications, and ultimately, it is determined by the analyst in an *ad hoc* manner. In the Fig. 2.2, we illustrate these concepts using a normal distribution as the underlying model.

Anomalies are usually classified in three categories (AGGARWAL, 2017; CHANDOLA *et al.*, 2009; OLTEANU *et al.*, 2022):

- **Point anomalies**: a single observation is classified as an anomaly. This is the focus of most anomaly detection techniques.
- **Contextual or conditional anomalies**: the abnormal condition of the point is determined by a context that, in turn, is induced by contextual attributes (see, for example, Fig. 2.3).
- **Collective anomalies**: an individual data instance may not be anomalous, but their occurrence together with other instances characterizes an anomaly (Fig. 2.4).

2.3 Relation of change-points and anomalies

For OLTEANU *et al.* (2022), the main difference between the change-point and anomaly is that, in the former, the data is considered "normal" on both sides of the change-point (each side follows a model). In contrast, in anomaly detection, it is usually assumed a model only for the normal data. However, in the literature of Sequential Change-point Detection, this definition is not appropriate. TARTAKOVSKY *et al.* (2015), for example, do not distinct the two concepts and employ change-point methods to detect network anomalies. In the same direction, TRUONG *et al.* (2020)

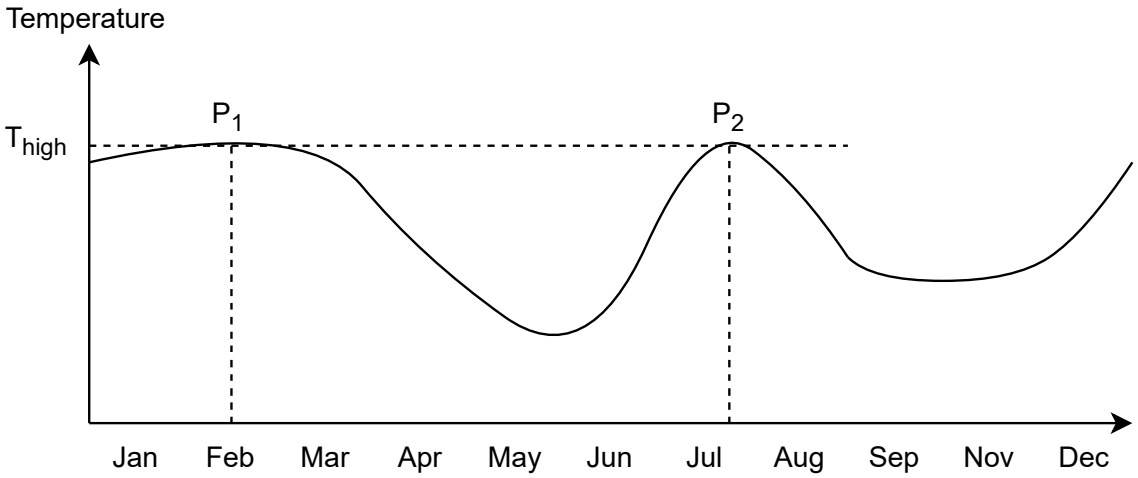


Figure 2.3: Contextual anomaly example. The temperature at point P_1 is normal for the context (summer in Brazil). But the same temperature at time P_2 (winter) is abnormal. Elaborated by the author based on Fig. 3 of CHANDOLA *et al.* (2009).

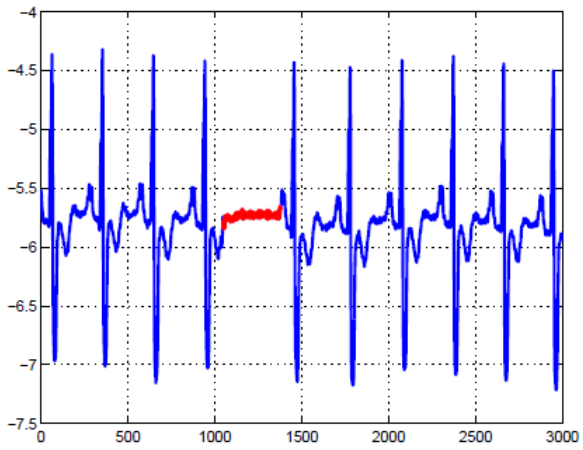


Figure 2.4: Collective anomaly. Atrial premature contraction in a human electrocardiogram. Note that this type of anomaly can also be interpreted as a change-point. Reproduced from CHANDOLA *et al.* (2009).

states that, in the online setting, change-point detection is often referred as event or anomaly detection. As discussed in Chapter 4, the sequential methods typically cannot distinguish change-points from point anomalies.

However, distinguishing change-points from anomalies is desirable in many real-world applications (ALTAMIRANO *et al.*, 2023; FEARNHEAD and RIGAILL, 2019; LIU *et al.*, 2023; XIMENES *et al.*, 2018). To illustrate, consider the examples of Fig. 2.5. It seems to be consensual that in case (a), there is no change-point, and in case (f), there is only one change-point ($t = 10$). In case (b), people usually label the point $t = 10$ as an outlier or anomaly (see *e.g.* Fig. 6.9) and not a change-point. However, the cases (c), (d) and (e) are more subjective. These difficulties were also commented on and addressed by XIMENES *et al.* (2018).

Based on the example of Fig. 2.5 and corroborating with XIMENES *et al.* (2018),

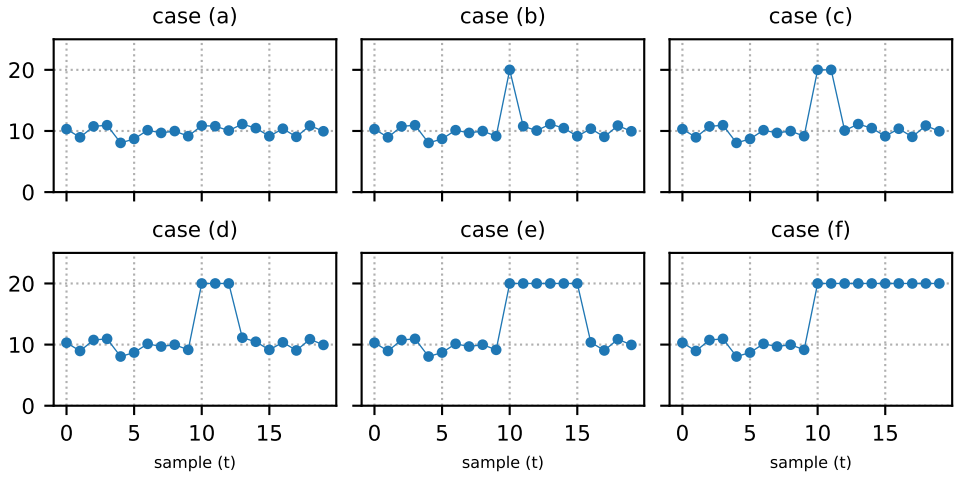


Figure 2.5: Difficult in labeling change-point and anomalies (example with generated data). The cases (a), (b) and (f) seem to be obvious, whereas (c), (d) and (e) are more subjective.

and argue that the main characteristic that differs change-point from anomaly is the duration of the system in the new state after the change. We classify an instant as a change-point only if the system persists in the new state for a certain period. The difficulty, of course, is to define this minimum period.

Using the above understanding, collective anomalies can also be viewed as the occurrence of two change-points, a first that changes the process to an abnormal state and a second that returns the process to their normal state. That could be the cases (c), (d) and (e) of Fig. 2.5. This idea - interpreting collective anomalies as change-points - is known in the literature as *epidemic change-point* (FISCH *et al.*, 2022; JUODAKIS and MARSLAND, 2023).

Another clear difference between anomaly and change-point methods is that the former can be applied to more general data types, such as objects and graphs. In contrast, change-point usually deals only with numeric data. Lastly, change-point methods are mostly unsupervised, while anomaly detection can be supervised, semi-supervised or unsupervised (CHANDOLA *et al.*, 2009).

2.4 Methods surveys

For the task of anomaly detection, there are an extensive literature review, surveys and benchmarks; see *e.g.* BRAEI and WAGNER (2020), SCHMIDL *et al.* (2022) and HAN *et al.* (2022). Interestingly, despite the deep-learning progress, these works concluded that the classical statistical methods still perform better for most of the datasets evaluated. However, many proposed methods use deep learning; see PANG *et al.* (2021) for a review.

For the task of change-point detection, the number of works (methods, review

and survey papers) seems to be significantly smaller than anomaly detection, especially when considering the machine learning approach. TRUONG *et al.* (2020) perform a selective review of offline methods. AMINIKHANGHAHI and COOK (2017) presents a survey using online and offline methods and evaluates some of them. However, they did not use the same dataset and metrics for all the algorithms, so comparing performance is difficult.

The lack of a labeled time series partly justifies the few surveys and evaluation works that use various (real-data) datasets in the change-point task. Recognizing this, BURG and WILLIAMS (2020) proposed a framework to label the datasets considering possibly different labels given by a group of persons. Then, using five different annotators, they provided a benchmark evaluating 14 algorithms in a 37 time series set from different domain areas. As a result, with hyperparameter tuning, the BOCD outperformed all other methods, including those offline.

Regarding neural networks, HUSHCHYN *et al.* (2020) proposed two online models and compared them with offline methods, including Pelt. They used 11 datasets: three synthetics (mean, variance and co-variance jump), two with human activities sensor data, two with astronomy measurements, two with high energy physics, and one using sequence handwritten digits of the MNIST dataset. They evaluated the F1 score and similarity-based index and showed that their models outperformed the others in various cases. One criticism is that they used a vast margin size ($\delta = 50$), comparable to the signal length and dynamics, to compute the true positive rate. This margin δ will be defined in Section 2.5. Another recent work is (ZHOU *et al.*, 2024), where they proposed a model for mean shift detection with theoretical guarantees. Still, the model is offline, and they evaluated it using only one time series of stock prices. Similarly, LI *et al.* (2022) proposed a deep-learning model and reported to achieve comparable performance with CUSUM, but the proposed model is *offline*, and they tested using only one dataset.

Interestingly, despite the intrinsic connections between change points and anomalies, they are not usually explored together in the literature. Exceptions are TAKEUCHI and YAMANISHI (2006), SU *et al.* (2013), FEARNHEAD and RIGAILL (2019), OLTEANU *et al.* (2022) and (LIU *et al.*, 2023).

2.5 Evaluation metrics

Let the set of true change points be denoted by $\mathcal{T} = \{\tau_1, \ldots, \tau_M\}$ and the set of detected change-points by the algorithms $\hat{\mathcal{T}} = \{\hat{\tau}_1, \ldots, \hat{\tau}_K\}$. Note that the cardinalities of theses sets, $|\mathcal{T}| = M$ and $|\hat{\mathcal{T}}| = K$ may be different. It's common to allow a certain margin of error δ to identify each change-point (see the example of Fig. 2.6). In this case, we define the true positives as (TRUONG *et al.*, 2020)

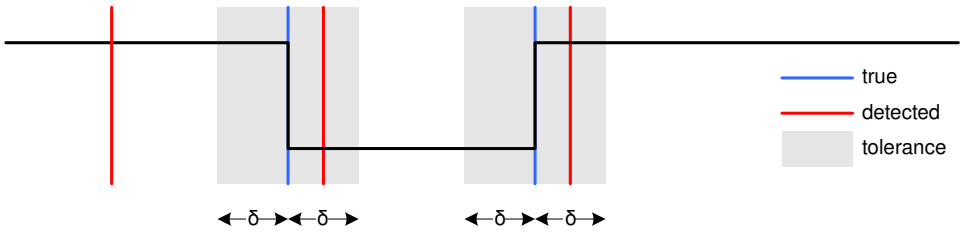


Figure 2.6: Change-point metrics. In this example, $|\,\mathrm{TP}\,| = 2$, $P = 2/3$ and $R = 2/2$. Based on fig. 5 of TRUONG *et al.* (2020).

$$\mathrm{TP}(\mathcal{T}, \hat{\mathcal{T}}) := \left\{ \tau \in \mathcal{T} \mid \exists \hat{\tau} \in \hat{\mathcal{T}} \text{ such that } |\tau - \hat{\tau}| < \delta \right\} \tag{2.3}$$

with the restriction that the association of τ and $\hat{\tau}$ must be one-to-one or, strictly, 0..1 to 0..1.

Similarly, we can define the false positives as

$$\mathrm{FP}(\mathcal{T}, \hat{\mathcal{T}}) := \left\{ \hat{\tau} \in \hat{\mathcal{T}} \mid \nexists \tau \in \mathcal{T} \text{ such that } |\tau - \hat{\tau}| < \delta \right\}. \tag{2.4}$$

.

Now, let $|\mathcal{X}|$ be the total number of observations. Then, the total negatives, *i.e.*, the number of time instants that are not change points, is $|\mathcal{N}| = |\mathcal{X}| - |\mathcal{T}|$. With these definitions, the true positive rate (TPR) and false positive rate (FPR) can be expressed as

$$\mathrm{TPR} = \frac{|\,\mathrm{TP}(\mathcal{T}, \hat{\mathcal{T}})|}{|\mathcal{T}|} \qquad \mathrm{FPR} = \frac{|\,\mathrm{FP}(\mathcal{T}, \hat{\mathcal{T}})|}{|\mathcal{N}|}, \tag{2.5}$$

In the same way, the Precision (P) and Recall (R) can be expressed as

$$P = \frac{|\,\mathrm{TP}(\mathcal{T}, \hat{\mathcal{T}})|}{|\hat{\mathcal{T}}|} \qquad R = \frac{|\,\mathrm{TP}(\mathcal{T}, \hat{\mathcal{T}})|}{|\mathcal{T}|} \tag{2.6}$$

Note that the number of detected change points is given by the sum of the true positives and false positives, *i.e.*, $\hat{\mathcal{T}} = \mathrm{FP} + \mathrm{TP}$. Then, metric Precision considers the false positives: the more false positives, the lower the Precision. On the other hand, the Recall considers the false negatives (undetected change points): the more false negatives, the lower the recall. Usually, there is a trade-off between Precision and Recall: if one wants to have a high precision (low FPR), then some true change points can be undetected, leading to a low recall (high false negative rate). Conversely, a high recall usually implies more false positives (low precision). Aiming to combine both metrics, the F1-score is defined as the harmonic mean of Precision

and Recall.

In the Fig. 2.6, denote the two true change-points (in blue) by τ_1 and τ_2 and suppose that the value of δ would be sufficiently larger such that the two tolerance region (in gray) overlaps. In this case, note that a detected change-point $\hat{\tau}$ between the two regions ($\tau_1 - \delta \leq \hat{\tau} \leq \tau_2 + \delta$) could be assigned for both labels, *i.e.*, $\hat{\tau} = \tau_1$ or $\hat{\tau} = \tau_2$. In other words, computing the true positives using a tolerance margin is not always straightforward since multiple solutions can be admitted. To deal with this problem, SHAO *et al.* (2017) proposed an elegant solution: first, they defined that an optimal mapping occurs when the cardinality of TP is maximized while the sum of distances $|\tau - \hat{\tau}|$ is minimized. With this formulation, they show the problem can be translated to the well-known problem of finding the *minimum cost maximum-cardinality* of a bipartite graph, which can be solved by the Hungarian algorithm (KONIG, 1931). See (SHAO, 2017) for a detailed explanation. In this work, we use this solution to compute the true positives.

Beyond these metrics familiar in the supervised classification, it is also possible to use other metrics commonly employed for clustering. See (TRUONG *et al.*, 2020) and (BURG and WILLIAMS, 2020) for a more-in-depth discussion.

Another critical issue is that the number of positive labels (change points) is usually infrequent compared to the negative labels (all other points). Probably because of this, differently from works in traditional machine learning, even the most recent works in change-point detection (*e.g.,* (BURG and WILLIAMS, 2020)) usually do not divide the datasets into training and test sets. Neither this problem is mentioned in the surveys (TRUONG *et al.*, 2020) and AMINIKHANGHAHI and COOK (2017).

2.6 Applications with network measurements

MATIAS *et al.* (2011) compared the performance of Shewhart, EWMA and CUSUM methods to monitor the quality of network traffic forecasts based on the residuals of an autoregressive moving average (ARMA) model. They conclude Shewhart performed better than CUSUM but worse than EWMA, but they restrict the analysis for traffic data, not considering QoS metrics.

Using the M-Lab NDT dataset, FARKAS (2016) applied the two-sided CUSUM to time series of download throughput and package re-transmissions using a sliding window to detect anomalous segments and points in the window. The author first proposed searching for a segment of the window that complies with the Shapiro-Wilker normality test. Then, this segment is used to estimate the pre-change distribution parameters and to tune the hyperparameters. Unlike our work, the author did not restart the surveillance process after a deviation in the CUSUM statistic.

Thus, the algorithm's output is anomalous points that are not necessarily change points. Nonetheless, we were inspired by this work to propose a framework for the classical methods (Section 4.2).

In our group, XIMENES *et al.* (2018) used an offline change-point detection method and a spatial-temporal correlation to detect network regions with similar performance. SANTOS *et al.* (2019) extended that work employing Hidden Markov Model (HMM) to detect changes in package loss time series, including a proposed online framework. The focus of this last work was to detect changes in the quality of services and to correlate with user calls. Regarding the HMM model, they considered a binomial (discrete-time) distribution for the observations and 4 (hidden) states, each of them with a distribution of package loss. Then, they interpreted these distributions by relating them to network quality states: *good*, *intermediate*, *bad* and *network unavailable*. The main difference to our work is that we study other methods and focus on latency and throughput time series instead of package loss. Another difference is that we do not restrict the number of states. Yet regarding research in our group, beyond the change-point problem, STREIT *et al.* (2021) tackled the issue of anomaly detection using tensor decomposition.

Investigating the matching of RTT and path (routes) changes, SHAO *et al.* (2017) employed change-point methods to RTT time series and also provided a labeled dataset, which we use in this work. Furthermore, they provided a methodology to compute the true positives, introduced in Section 2.5, and evaluated the dataset using the Pelt method (Section 3.4) with different cost functions (normal, exponential, Poisson and non-parametric) and different penalties. The Gaussian model with Modified Bayesian Information Criteria (MBIC) resulted in the best recall but worst precision, whereas the performance of the other models was relatively closed. Based on this work, we selected the Pelt-NP method as the offline reference method to compare the performance of our selected online algorithms.

Also using the SHAO *et al.* (2017) dataset, MOUCHET *et al.* (2020) studied a non-parametric bayesian offline HMM model, HDP-HMM (FOX *et al.*, 2011). The proposed method performed at least well as the Pelt methods, with an apparent advantage in terms of precision and recall (MOUCHET, 2020), but no statistical confidence interval or test was provided to support the results; only the median values. Furthermore, the implementation code was not available.

Chapter 3

Selected methods

In this chapter, we present the selected online methods used in this work: the classical Shewhart, EWMA and CUSUM, including the recently proposed variant WL-CUSUM; and the more recent proposals BOCD and RRCF. We also describe the offline method Pelt-NP, used as a reference for performance comparison.

3.1 Classical methods

3.1.1 Shewhart

The Shewhart control charts (SHEWHART, 1929) were proposed and are still widely used in the context of Statistical Process Control (SPC). Formally, the Shewhart method can be stated as a sequence of Neyman-Pearson (NP) hypothesis tests. Let m be the sample size (in our work, $m = 1$), $\lambda(t)$ the log-likelihood ratio (LLR) for the sample t, h the decision threshold and d_t a random variable that indicates if there is a change ($d_t = 1$) or not. Then, the test can be written as (TARTAKOVSKY *et al.*, 2015):

$$d_t = \begin{cases} 0 & \text{if} \quad \lambda(n) < h \\ 1 & \text{if} \quad \lambda(n) \geq h \end{cases}, \qquad \lambda(t) = \sum_{i=(n-1)m+1}^{nm} \log \frac{p_1(x_i)}{p_0(x_i)}. \tag{3.1}$$

For the particular case of a change in the mean of a Gaussian sequence $X_t \sim \mathcal{N}(\mu, \sigma_0)$ from $\mu = \mu_0$ (hypothesis $\mathcal{H}_0$) to $\mu = \mu_1 \neq \mu_0$), and assuming that σ_0^2 is known, it can be shown that the likelihood ratio test is equivalent to the standard Z-test for large random samples (RAMACHANDRAN and TSOKOS, 2020, p. 268). Furthermore, in the case of a unitary sample, the hypothesis test reduces to

$$|x_t - \mu_0| \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \kappa\sigma_0, \tag{3.2}$$

where κ is the number of acceptable standard deviations, equivalent to the Z-test's score.

It is worth mentioning that, in the SPC literature, the Shewhart method for individual measurements (known as X-chart or I-MR chart) is usually built using the moving range statistic instead of the standard deviation. According to NELSON (1982), the aim is to minimize inflationary effects on the variability caused by trends and oscillations. However, from the example provided in that work, one can verify that the difference is not too significant. So, for simplicity, we choose to use the sample standard deviation as an estimator for σ_0. Furthermore, a bias correction factor should be applied because the sample standard deviation is a biased estimator. However, this factor can be neglected for sample sizes above 10 (the minimum window size used in our experiments). Refers to(MONTGOMERY, 2013) for details.

3.1.2 Exponential Weighted Moving Average

The Exponential Weighted Moving Average (EWMA) (ROBERTS, 1959), also called geometric moving average, is defined as

$$z_t = \lambda x_t + (1 - \lambda) z_{t-1}, \tag{3.3}$$

where $0 < \lambda < 1$ is a hyperparameter that weights the past observations: the higher the λ, the more importance is given to the recent observations and less to past ones.

Developing recursively the Eq. (3.3), for an arbitrary t:

$$z_t = \lambda \sum_{j=0}^{t-1} (1 - \lambda)^j x_{t-j} + (1 - \lambda)^t x_0 \tag{3.4}$$

where we can see the weights $\lambda(1 - \lambda)^j$ decrease geometrically with t; it can be shown that they always sums to 1. Furthermore, a distinct characteristic compared to the Shewhart method is that at each time step t, the EWMA statistic considers the entire history of observations. In contrast, Shewhart statistic considers only information from the last sample.

It is interesting to note that, in its classical formulation (ROBERTS, 1959), the EWMA is non-parametric (no distribution is assumed to the data). However, it can also be formulated using the LLR. In this way, the classical formulation can be derived as a special case, considering a change in the mean of a Gaussian sequence. (TARTAKOVSKY *et al.*, 2015). Nonetheless, the EWMA is known to be robust against non-normality (BORROR *et al.*, 1999). For (MONTGOMERY, 2013, p. 439), "*It is almost a perfectly non-parametric procedure*".

For $X_t \sim \mathcal{N}(\mu_0, \sigma^2)$, the variance of Z_t is (MONTGOMERY, 2013)

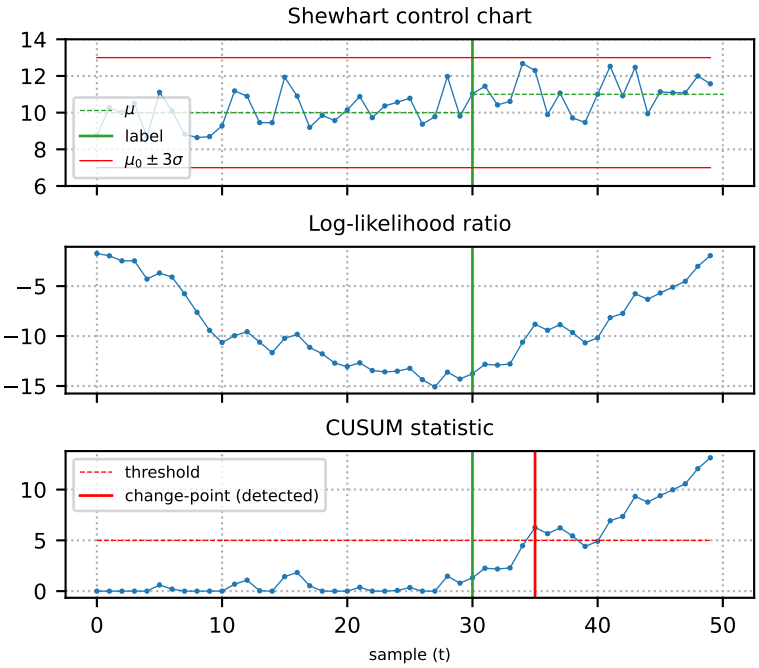


Figure 3.1: Shewhart and CUSUM example with $X_t \sim \mathcal{N}(\mu_0 = 10, \sigma^2 = 1)$. The mean changed to $\mu_1 = 11$ at $t = 30$. The Shewhart chart could not detect the shift, but CUSUM identified the change-point at $t = 35$.

$$\sigma^2_{Z_t} = \sigma^2 \left(\frac{\lambda}{2 - \lambda} \right) \left[1 - (1 - \lambda)^{2t} \right], \tag{3.5}$$

with $\sigma^2_{Z_t}$ being used to set the control limits $\pm \kappa_d \sigma_{Z_t}$. As t increases, the term $(1-\lambda)^{2t}$ rapidly approaches 0, and, because of this, in our implementation, we neglected it.

3.1.3 Cumulative Sum

Perhaps the most popular sequential change-point detection method is the Cumulative Sum (CUSUM) (PAGE, 1954). The procedure analyzed as a repeated Sequential Probability Ratio Test using the framework of Sequential Analysis. Furthermore, the CUSUM can also be formulated as GLR test (TARTAKOVSKY *et al.*, 2015). Here, we present an intuitive interpretation of the CUSUM provided in TARTAKOVSKY *et al.* (2015).

Consider the observation model of Eq. (2.2). The key idea is to note that the LLR shows a negative drift before the change and a positive drift after the change, as is illustrated in the example of Fig. 3.1.

Let λ_t be the LLR and Z_t a r.v. defined as

$$\lambda_t = \sum_{i=1}^{t} Z_i = \sum_{i=1}^{t} \log \frac{p_1(x_i)}{p_0(x_i)}. \tag{3.6}$$

The relevant information (concerned to the change) is the difference between the LLR $\lambda_n = \sum_{i=1}^{t} Z_i$ (the *cumulative sum*) and its current minimum value:

$$g_t = \lambda_t - \min_{0 \leq j \leq t} \lambda_j \tag{3.7a}$$

$$= \sum_{i=1}^{t} Z_i - \min_{0 \leq j \leq t} \sum_{i=1}^{j} Z_i \tag{3.7b}$$

$$= \max_{1 \leq j \leq t+1} \sum_{i=j}^{t} Z_i \tag{3.7c}$$

$$= \max \left\{ 0, \max_{1 \leq j \leq t} \sum_{i=j}^{t} Z_i \right\} \tag{3.7d}$$

where $\sum_{i=t+1}^{n} Z_i = 0$. Using h as a threshold value for the statistics, we decide on a change-point when $g_t \geq h$.

Trough Eq. (3.7a), the detection rule can viewed as a comparison of the cumulative sum (LLR) λ_t with an adaptive threshold. In the words of (TARTAKOVSKY *et al.*, 2015, p. 377): "*this threshold is not only modified online but keeps complete memory of the entire useful information contained in the past observations*". This ability to track past information is the key feature that differs CUSUM from the Shewhart.

Last but not least important, the CUSUM statistic, Eq. (3.7d), can also be written in a recursive form:

$$\begin{aligned} g_t &= [g_{t-1} + Z_t]^+ \\ &= \left[g_{t-1} + \log \frac{p_1(x_t)}{p_0(x_t)} \right]^+, \ t \geq 1, \ g_0 = 0. \end{aligned} \tag{3.8}$$

Many variants of the CUSUM are proposed in the literature. XIE *et al.* (2021) review the main results and point to new research directions. In this work, we focus on the most known formulation of the algorithm, the two-sided CUSUM (2S-CUSUM), and in a recent variation proposal, the WL-CUSUM.

Two-sided CUSUM

Let's consider the particular case of an additive change in a Gaussian i.i.d. sequence of r.v. $X_t \sim \mathcal{N}(\mu, \sigma^2)$, from $\mu = \mu_0$ to $\mu = \mu_1$, keeping constant the variance σ^2. With straightforward algebraic manipulation, the r.v. Z_t in the CUSUM statistic (Eq. (3.8)) simplifies to

$$Z_t = \frac{(\mu_1 - \mu_0)}{\sigma^2}\left[X_t - \frac{(\mu_0 + \mu_1)}{2}\right]. \tag{3.9}$$

Now, lets consider the two-sided case: an *upper change* in the mean, $\mu_1 = \mu_0 + \delta\sigma$, or a *lower change* in mean: $\mu_1 = \mu_0 - \delta\sigma$, where $\pm\delta\sigma$ is the change magnitude. From Eq. (3.9), it is clear that if want to re-write the CUSUM statistic in therms of δ, it leads to two equations:

$$\begin{aligned} g_t^u &= \left[g_{t-1}^u + x_t - \mu_0 - \frac{\delta\sigma}{2}\right]^+, \; g_0^u = 0 \\ g_t^\ell &= \left[g_{t-1}^\ell - x_t + \mu_0 - \frac{\delta\sigma}{2}\right]^+, \; g_0^\ell = 0. \end{aligned} \tag{3.10}$$

where g_t^u monitors the upper change and g_t^ℓ the lower one. In the above equations, we have dropped the constant scale therm δ/σ that appears in the derivation since it can be incorporated into the decision threshold, as is usual in the literature.

In the SPC literature, the Gaussian two-sided CUSUM formulation for monitoring a shift in the mean is also known as *tabular* CUSUM.

The Window-Limited CUSUM

The main problem of the CUSUM algorithm is that it assumes that both pre-change $p_0(\cdot)$ and post-change $p_1(\cdot)$ distributions are known. The first premise usually is not a problem since $p_0(\cdot)$ can be estimated from past data. However, not knowing the $p_1(\cdot)$ and selecting arbitrary δ in Eq. (3.10) may lead to performance degradation.

Recently, XIE *et al.* (2023) propose to estimate $p_1(\cdot)$ at each new data sample using a sliding window scheme. They called this scheme Window-Limited CUSUM (WL-CUSUM).

In the WL-CUSUM scheme, the parameter θ_1 of $p_1(\cdot)$ is substituted by a consistent estimate:

$$g_t = g_{t-1}^+ + \log\frac{p_1(x_t, \hat{\theta}_{1,t-1})}{p_0(x_t)}, \; g_w = 0, \; n = w_1 + 1, w_1 + 2, \ldots \tag{3.11}$$

where w_1 is the sliding window. Then, θ_1 can can be estimated through maximum likelihood estimate (MLE):

$$\hat{\theta}_{1,t} = \arg\max_{\theta\in\Theta} \sum_{i=0}^{w_1-1} \log p_1(x_{t-i}, \theta). \tag{3.12}$$

With a sufficiently large window, the authors showed that WL-CUSUM statistic behaves like the original CUSUM while maintaining optimality properties. Similar

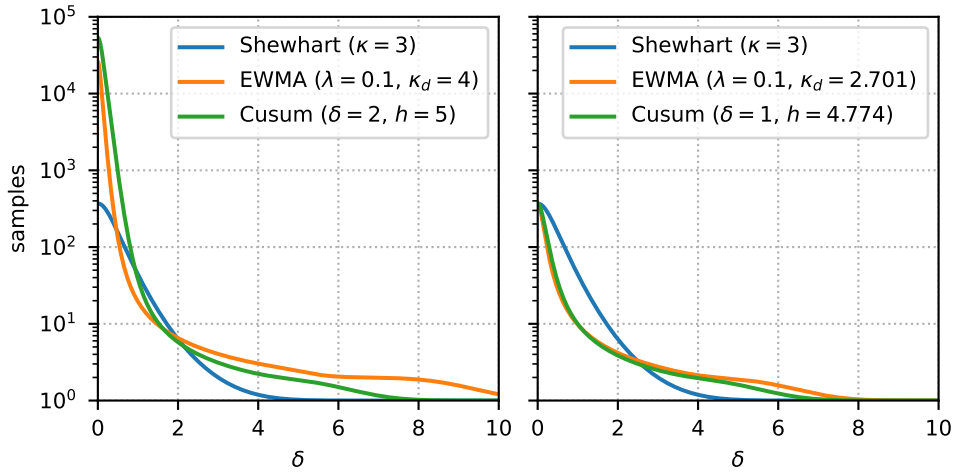


Figure 3.2: ARL comparison for Shewhart, EWMA and CUSUM. In the left, hyperparameters are tuned with common values; in the right, we adjust them to get a ARL to false alarm equal 370. Computed with the R package `SPC` (KNOTH, 2022)

.

to other methods that use sliding windows, one issue is the choice of the window size.

The authors evaluated the proposed algorithm using only simulated data. To our knowledge, this is the first work evaluating this proposal with real data.

3.1.4 Theoretical comparison

The sequential change-point algorithms are usually characterized and evaluated through the average run lenght (ARL) function, which relates the expected number of samples to detect a change in terms of some parameter of interest, for example, the change magnitude in the mean. This function provides the false alarm rate and the expected delay to detection. For the Shewhart method, the ARL function can be computed exactly using the Z-test's type II error. For the EWMA and CUSUM methods, the derivation of the ARL function is more complex but well studied (see *e.g.* (TARTAKOVSKY *et al.*, 2015)).

In Fig. 3.2, we plot the ARL function of Shewhart (X chart), 2S-CUSUM and EWMA to detect a shift of δ standard deviations in the mean of a Gaussian process. The graphic suggests that Shewhart performs better for shifts more significant than $\pm 3\sigma$, whereas EWMA and CUSUM have small ARL for small shifts.

According to HUNTER (1986), as stated in POLUNCHENKO *et al.* (2013), the EWMA can be viewed as a compromise between the Shewhart and the CUSUM. Also, LUCAS and SACCUCCI (1990) showed that EWMA can be as powerful as CUSUM to detect a shift in the mean of Gaussian noise.

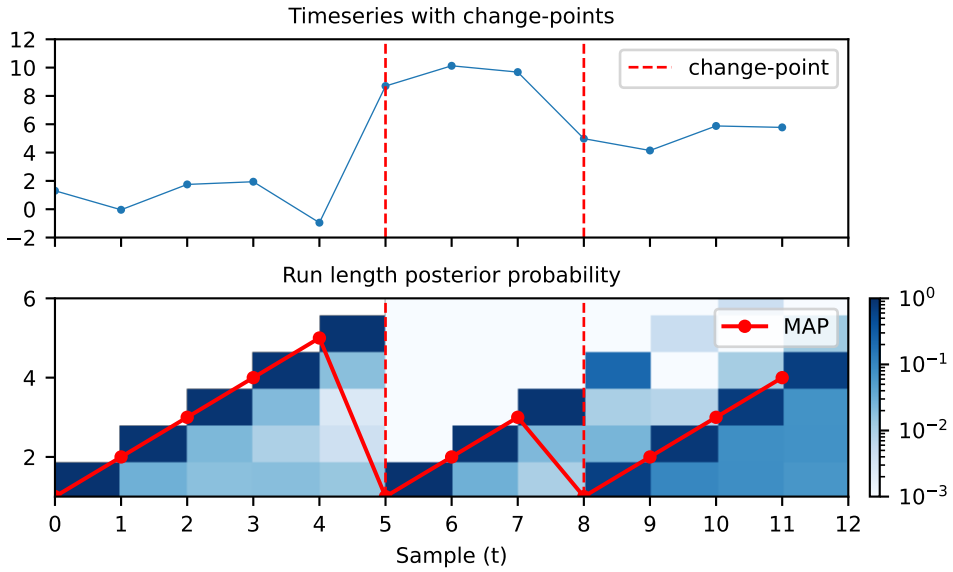


Figure 3.3: BOCD example (synthetic data) showing the relation of the change-points (top) and the run length variable (bottom).

3.2 Bayesian Online Changepoint Detection

The Bayesian Online Changepoint Detection (BOCD) was introduced independently by ADAMS and MACKAY (2007) and FEARNHEAD and LIU (2007) (MURPHY, 2023). A key idea of the BOCD is, instead of modeling the change-point itself as a variable, to use the *run lenght* discrete variable r_t, defined as the amount of time spent since the last change-point:

$$r_t = \begin{cases} 0, & \text{if change-point at time } t \\ r_{t-1} + 1, & \text{else.} \end{cases} \tag{3.13}$$

This variable is illustrated in the Fig. 3.3, where we can see that at each time step, r_t is increased by one or drops to zero when the change-point occurs.

In the BOCD model, the run length is a latent random variable, and its posterior probability can be computed from the joint distribution:

$$p(r_t \mid \mathbf{x}_{1:t}) = \frac{p(r_t, \mathbf{x}_{1:t})}{p(\mathbf{x}_{1:t})} \tag{3.14}$$

Then, another key idea is to compute the joint distribution in a recursive way [see (GUNDERSEN, 2019) for a detailed derivation]:

$$p(r_t, \mathbf{x}_{1:t}) = \sum_{r_{t-1}} \overbrace{p(r_t \mid r_{t-1})}^{\text{Change-point prior}} \overbrace{p(x_t \mid r_{t-1}, \mathbf{x}_t^{(r)})}^{\text{UPM predictive}} \overbrace{p(r_{t-1}, \mathbf{x}_{1:t-1})}^{\text{Message}}. \tag{3.15}$$

where UPM is the underlying probabilistic model. Note that this term must be computed for each run length r_t. When using a model from the exponential family, the

posterior predictive can be computed in closed form using only sufficient statistics because of its prior-conjugacy property. In this work, we used the Normal-Inverse-Gamma distribution, which is the conjugate prior of the normal distribution with unknown mean and variance.

A known problem of the Bayesian inference affecting the performance of the BOCD is the lack of robustness under outliers. To address this problem, recent works propose robust versions of BOCD using the Generalized Bayesian (GB) inference framework. The GB inference aims to estimate the posterior replacing the Kullback-Leibler divergence (used in the classical Bayes posterior) with a general divergence metric and the negative log-likelihood (also the standard in the Bayes posterior) with a general risk function. (ALTAMIRANO *et al.*, 2023; MURPHY, 2023).

Our study tested a recent proposal, the DSM-BOCD, (ALTAMIRANO *et al.*, 2023), that claims to deliver provable robustness without sacrificing scalability. However, the robust posterior proposed (based on the Fischer divergence metric) requires computing gradient matrices and is more prone to numerical instability issues. To contour this, the authors employed a standard-scaling pre-processing of the time series in their implementations. Of course, this is not suitable for real-world online applications.

In the Chapter 4, we propose a simple modification in the BOCD method to increase the robustness to outliers.

3.3 Robust Random Cut Forest

The Robust Random Cut Forest (RRCF) GUHA *et al.* (2016) is a non-parametric anomaly detection model based on the idea of Isolation Trees introduced in (LIU *et al.*, 2008, 2012), but designed to handle streamed data, including time series.

The idea of isolation-based anomaly detection is illustrated in the Fig. 3.4. At each iteration, the method randomly chooses a dimension (feature) and then divides, or "cut" (also randomly) this dimension in two subsets. This process is repeated until the desired point is isolated or the maximum number of allowed steps is achieved. On the left side of the figure, an anomaly point took 5 cuts to be isolated, whereas on the right side, a normal point took 20 steps.

This process can be efficiently handled with binary trees. LIU *et al.* (2008) proposed a data structure (iTrees) where the number of cuts is translated to the path length from the root to the leaf containing the point. Then, with a forest of iTrees, the authors show that average path length converges.

Motivated to allow online learning, GUHA *et al.* (2016) introduced a new model of isolation tree, the Robust Random Cut Tree (RRCT), and a new anomaly statistic. RRCT differs from iTree in selecting the dimension to cut: RRCT sets the

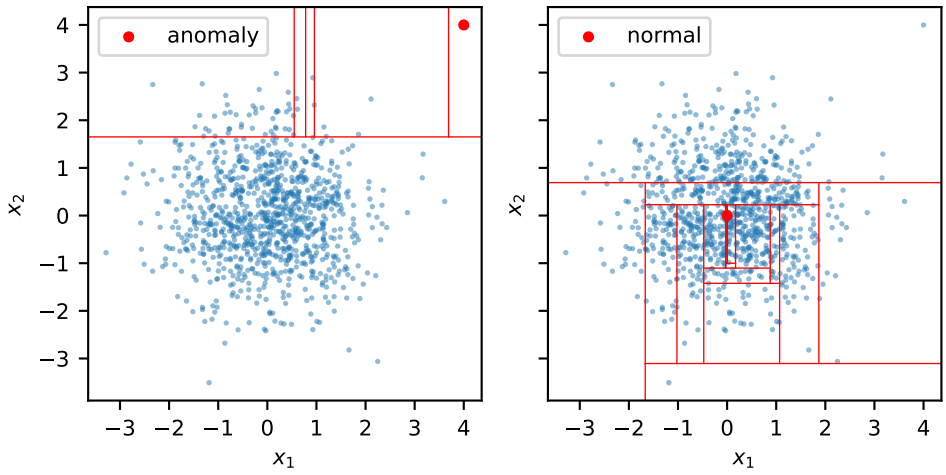


Figure 3.4: Isolation-based anomaly detection (synthetic data). The number of random cuts in the axes required to isolate an anomaly point (left side) is less than that required for a normal point (right side). Based on (LIU *et al.*, 2012).

probability of choosing a dimension proportional to its "relevance", instead of picking it uniformly. Furthermore, the authors show that the statistic used by iForest (the average path length) is not always helpful in characterizing anomalies, so they introduce a new statistic, the "collusive displacement", related to increased model complexity when a new instance is inserted in the forest.

3.3.1 Anomaly detection example

The Robust Random Cut Forest (RRCF) was initially introduced for anomaly detection. To gain insight into the method behavior, we reproduce in the Fig. 3.5 the same synthetic example (univariate time series) from (GUHA *et al.*, 2016), but introducing additional point anomalies at $t = 105$, $t = 305$ and $t = 309$.

When using time series, GUHA *et al.* (2016) proposed to use a sliding window (hyperparameter `shingle_size`) with the current sample and a certain number of past samples. In the example of Fig. 3.5, a window of size $w = 4$ is used, so, when processing the sample t, the point $\mathbf{x} = \{x_t, x_{t-1}, \ldots x_{t-w}\}$ is inserted at each tree of the forest. They call these sliding windows "shingles".

The motivation for using this shingle scheme is to consider the auto-correlation of the time series, improving the detection of anomalies in some instances. However, a side effect of this scheme is the lag introduced in the statistics when processing a point anomaly. This is illustrated in the example of Fig. 3.5: after a point anomaly at $t = 105$, the RRCF statistic persists at a high level for more 3 samples. If we use a threshold value to detect the anomaly point, the point $t = 105$ is classified as an anomaly (and $t = 106$, 107, 108). So, practical implementations should be aware of this characteristic. In our implementation, we use $w = 2$ to minimize this problem.

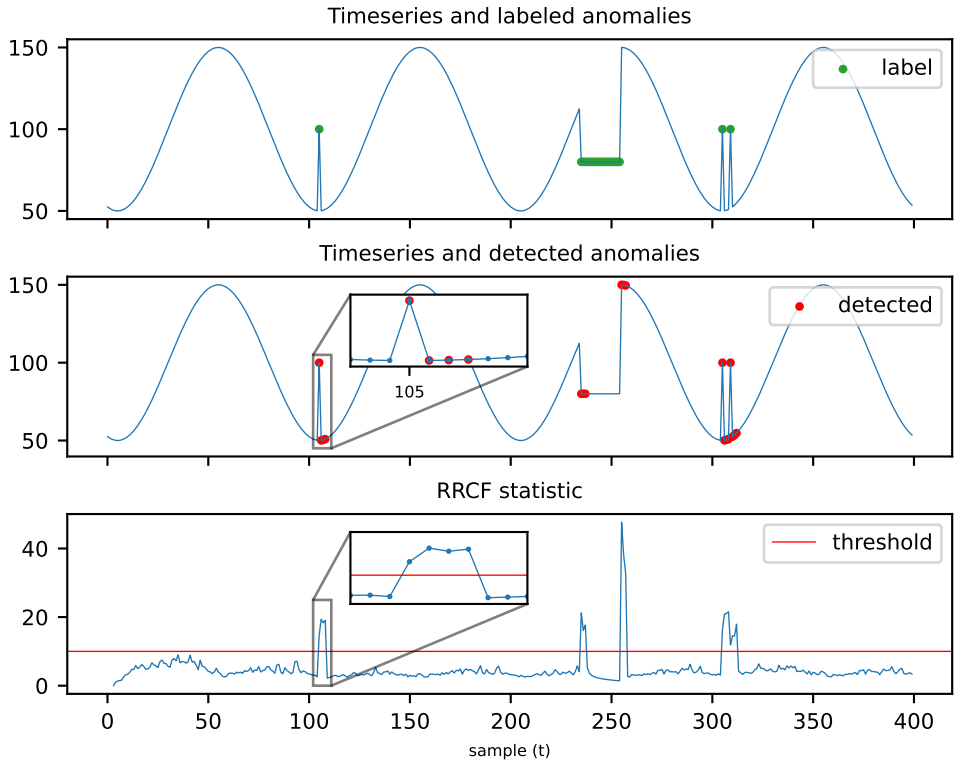


Figure 3.5: RRCF original example (synthetic data) of (GUHA *et al.*, 2016) extended with point anomalies at $t = \{105, 305, 309\}$ and a threshold for anomaly detection.

In the Fig. 3.5, it is also possible to note another characteristic of the original RRCF proposed scheme: because the forest is updated after each sample (increasing learning), the model quickly adapts itself in the presence of a collective anomaly. So, only the start and the end index of collective anomalies are detected. In the Section 4.4, we propose a simple framework to adapt the RRCF method to change-point detection.

3.4 Non-parametric Pelt

The Non-parametric Pelt (Pelt-NP) (HAYNES *et al.*, 2017) is an extension of the offline method Pelt (KILLICK *et al.*, 2012). We use it as a reference to compare the performance of the online algorithms because this was also used in the previous works (MOUCHET *et al.*, 2020; SHAO *et al.*, 2017) with RTT time series and also because Pelt is one of the start-of-the algorithms for offline segmentation. (CHO and KIRCH, 2021; HAYNES, 2017; TRUONG *et al.*, 2020).

In the offline formulation for the change-point problem (Section 2.1.1), the cost of segments are additive and the Bellman optimality principle holds, thus allowing the use of Dynamic Programming to write Eq. (2.1) in a recursive form. Using this scheme, JACKSON *et al.* (2005) proposed an exact method, the Optimal Par-

titioning (OP), but with a quadratic computational cost in the number of samples, $\mathcal{O}(t^2)$.

To reduce the cost of the OP algorithm, KILLICK *et al.* (2012) proposed the Pelt that, under mild conditions - most importantly, the number of change points increases linearly with t, the complexity reduces to linear. The main idea of Pelt is to sequentially check each sample regarding a pruning rule, discarding, when possible, the sample from the set of potential change points.

Usually, the negative log-likelihood is used as the cost function $\mathcal{C}(\cdot)$ (TRUONG *et al.*, 2020), but this requires the knowledge (or assumption) of the underlying distribution of generating data. To contour this, HAYNES *et al.* (2017) proposed an extension of Pelt that uses an empirical cumulative distribution function. This method, combined with the Modified Bayesian Information Criteria (mBIC) penalty (ZHANG and SIEGMUND, 2007), which takes into account the length of the segments, was reported by SHAO *et al.* (2017) to give the best recall and F2 score[1] when applied to the task of RTT time series segmentation.

[1]The F2 score gives more weight to recall than precision, whereas F1 score gives the same.

Chapter 4

Implementation and enhancements proposals

In this chapter, we present proposals for implementing the classical methods and modifications in the BOCD and RRCF methods to improve their performance.

4.1 Basic implementation of the classical methods

The sequential change-point methods assume that the parameters of pre-change distribution p_0 (Eq. (2.2), 7) are known. However, this is not the case for applications such as network quality monitoring. In our basic implementation of the sequential methods, we consider the following simple strategy: whenever the process monitoring begins (or after a change-point), we use the first samples, a fixed window of size w_0, to estimate the parameters using MLE. During this estimating phase, change-point detection is not performed. This procedure is illustrated in the flowchart of Fig. 4.1.

4.2 Proposed framework for the classical methods

In this section, we propose a series of simple strategies for improving the performance of the classical methods.

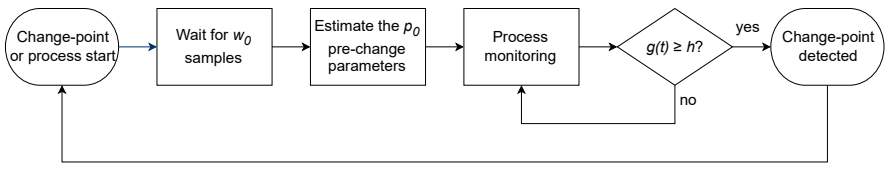


Figure 4.1: Sequential methods basic implementation flowchart. Here, p_0 is the pre-change distribution density function, $g(t)$ the method statistic and h the threshold.

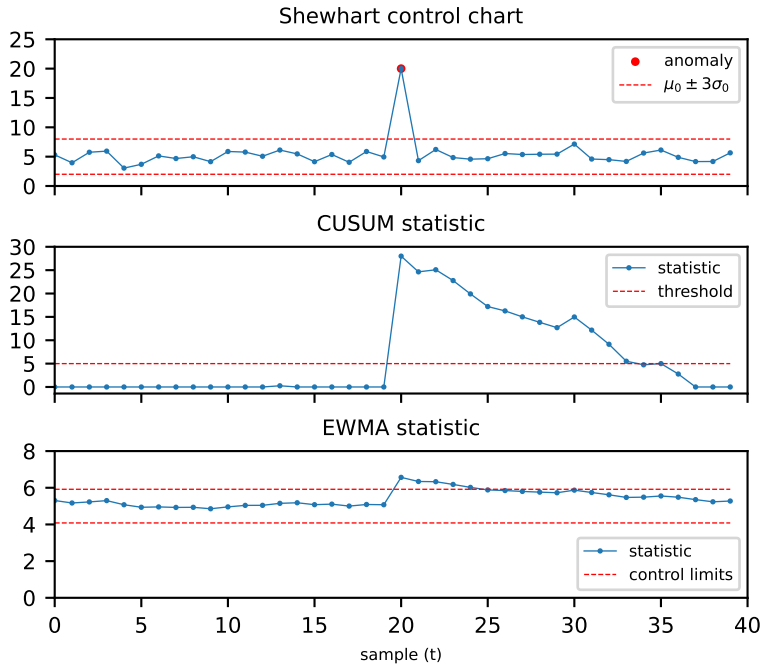


Figure 4.2: Influence of point anomalies on the classical methods. In this example (synthetic data), we used $\mu_0 = 5$, $\mu_1 = 7$ (for CUSUM algorithm), $\sigma_0 = 2$, and the hyperparameters $h = 5$ for CUSUM and $\lambda = 0.5$, $k_d = 4$ for EWMA.

4.2.1 Distinguishing point anomalies from change points

When a point anomaly occurs, abrupt changes in the statistical properties of the data impact the classical methods statistics so that, in general, they cannot distinguish a change-point from a point anomaly. This is illustrated on Fig. 4.2.

A usual solution to this problem is to include a filter (GUSTAFSSON, 2000). However, this solution can introduce an unacceptable lag or even prevent the detection of changes in the variance.

To contour this problem, we propose a simple alternative, illustrated in the flowchart of figure Fig. 4.3. Once the test statistic $g(t)$ deviates from the limit h, a change-point is confirmed only if c_{lim} observations lead, in sequence, to deviations. In the case of CUSUM and EWMA, their statistics have memory, *i.e.,* they considered the current and all the past samples. So, whenever a deviation occurs ($g(t) \geq h$), we reset the statistic to the last value before the deviation.

Once this proposal is straightforward, it has probably been employed in other works, but we have not been able to find references for this simple extension. A recent work employing LSTM (Long short-term memory) neural network ensembles mentions this same strategy to distinguish anomalies from changepoints (ATASH-GAHI *et al.*, 2021).

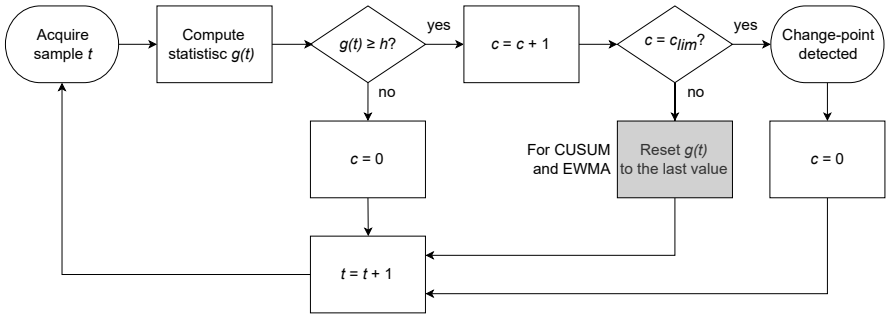


Figure 4.3: Classical methods - distinguishing point anomalies from change points.

4.2.2 Distinguishing noise from point anomalies

To distinguish noise (that is not of interest to the analyst) from anomalies, we use the z-score

$$z = \frac{|X_n - \mu_0|}{\sigma_0} \geq \kappa_a, \tag{4.1}$$

where κ_a is a threshold for anomalies (we use $\kappa_a = 5$). Furthermore, we classify the point anomalies in two classes: *upper anomaly*, when $x_t - \mu_0 > 0$ and *lower anomaly*, on the contrary.

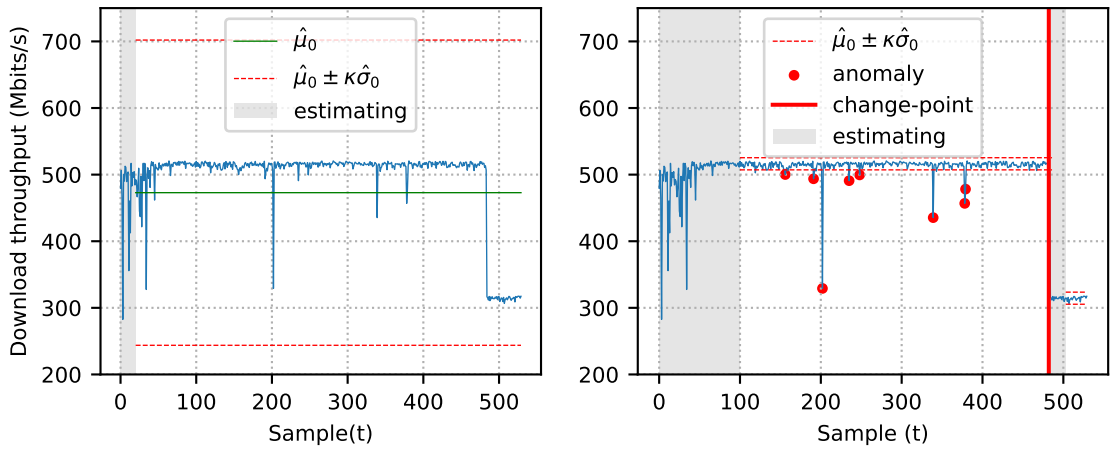
4.2.3 Improving the pre-change parameters estimation

In the basic implementation of the classical methods, the estimated parameters can lead to poor performance and missed change points if the process is not stabilized. To illustrate this, consider the Shewhart chart shown in Fig. 4.4a where the first 20 samples were used to estimate the pre-change parameters. In this example, note that the estimated is significant. This large variance leads to large control limits, not allowing the method to detect the change-point near the sample $t = 500$.

To improve the estimation of the parameters and ensure the Gaussian model's validity, FARKAS (2016) proposed to perform a Shapiro-Wilker normality hypothesis test in subsequences of sliding windows. In that work, the author searches for subsequences until the p-value meets the required specification or decreases the size of the subsequence until it reaches the limit size of 24 samples.

The Shapiro-Wilker tests the null hypothesis that the sample X_t is drawn from a normally distributed population. In (RAZALI and WAH, 2011), the authors employed Monte Carlo simulation and showed that the Shapiro-Wilker is more potent than the Kolomogorov-Smirnof, Lilliefors and Anderson-Darling tests.

Inspired in (FARKAS, 2016), we propose a similar procedure that employs the Shapiro-Wilker test but takes another step to check for variance increasing $\Delta\,\mathrm{Var}$ after a change-point. The rationale is that, during a process transient, the variance



(a) Basic estimation

(b) Proposed estimation

Figure 4.4: Pre-change parameters estimation.
NDT Dataset, Client 8, gig03, down. throughput.

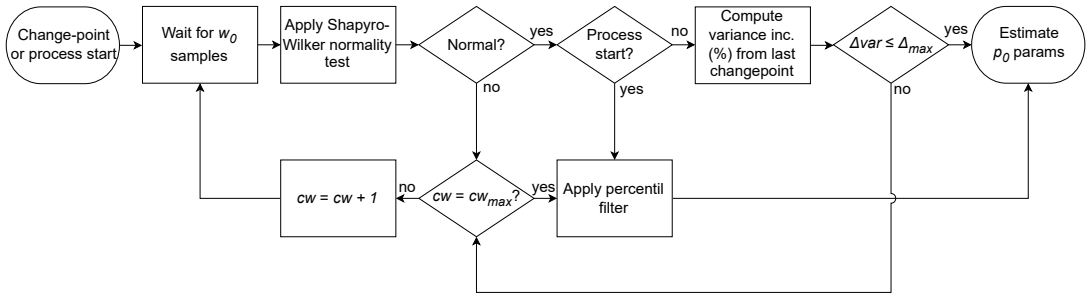


Figure 4.5: Parameters estimation proposed procedure. The key main idea is to apply an normality test and check if the variance is not too high before estimating the parameters of p_0.

can grow before the stabilization and, even with a high variance, the normality hypothesis may not be rejected. The proposed procedure is shown in the flowchart of Fig. 4.5.

In our proposal, we also define a maximum allowed number of consecutive estimating windows (hyperparameter cw_{max} in the flowchart); after that, the parameters are forced to be estimated, but applying first a percentile filter. This filter is also used when the process monitoring starts; in this case, we do not have a previous variance estimate.

The Fig. 4.4b shows an example of applying the proposed procedures. Starting with the first 20 samples, the Shapiro-Wilk test rejected the null hypotheses of normality. So, we wait for the following 20 samples. This repeats until $t = 100$ when the process gets stabilized. With the variance properly estimated, the Shewhart method identified the change-point near $t = 500$.

4.2.4 Checking for additional change after estimation

After the estimation procedure of Fig. 4.5, whenever the process takes more than one window to get stabilized, *i.e.*, $1 < cw \leq cw_{max}$, we also check for a possible change-point using the Augmented Dickey-Fuller (ADF) test (DICKEY and FULLER, 1979). If the current window used for parameters estimation pass in the ADF test, but the last did not, than we declare that a change-point occurred between these two windows.

The motivation to use the ADF test in addition to the normality test is the following fact: the Shapiro-Wilker normality test can reject the null hypothesis of normal distribution even if the process is stationary.

It's worth mentioning that, in the procedure of Fig. 4.5, we already tried to use the ADF test together with the Shapiro-Wilker test or as a substitute for it. However, these strategies do not reveal good results with our datasets.

4.3 BOCD enhancements

4.3.1 Change-point decision in the online setting

Despite its name, the BOCD is not genuinely full online. The posterior probability of the run length is computed online, but deciding on a change-point online is challenging, especially in the presence of outliers. To illustrate this, consider the example of Fig. 4.6, where we plotted the run length posterior probability and its maximum a posteriori (MAP) value evaluated online. Note that the MAP value oscillates from one state to another.

ADAMS and MACKAY (2007) provide no clue on deciding a change-point other than visual inspection. In the code repository of (ALTAMIRANO *et al.*, 2023), a convoluted algorithm is used, but in an offline setting, the algorithm uses the entire run length matrix.

To remedy this situation, we propose a simpler strategy: if the current run length probability drops below a threshold value (we use 0.05, hyperparameter `p_thr_rl` in Table A.2), we identify the most probable change-point and check if this point or its vicinity were not already previously identified.

Note that the need to check the list of previous change points is due to the oscillating behavior shown in the Fig. 4.6. Also, since we wait for the run length probability to drop below the threshold of a low value (instead of simply verifying the maximum a posterior run), in some instances, our solution leads to a lag in the detection. This lag and the false positive rate are a trade-off. It's worth mentioning that, while we did not take too much time investigating this problem, more efficient solutions are probably possible, but we did not find any in the literature.

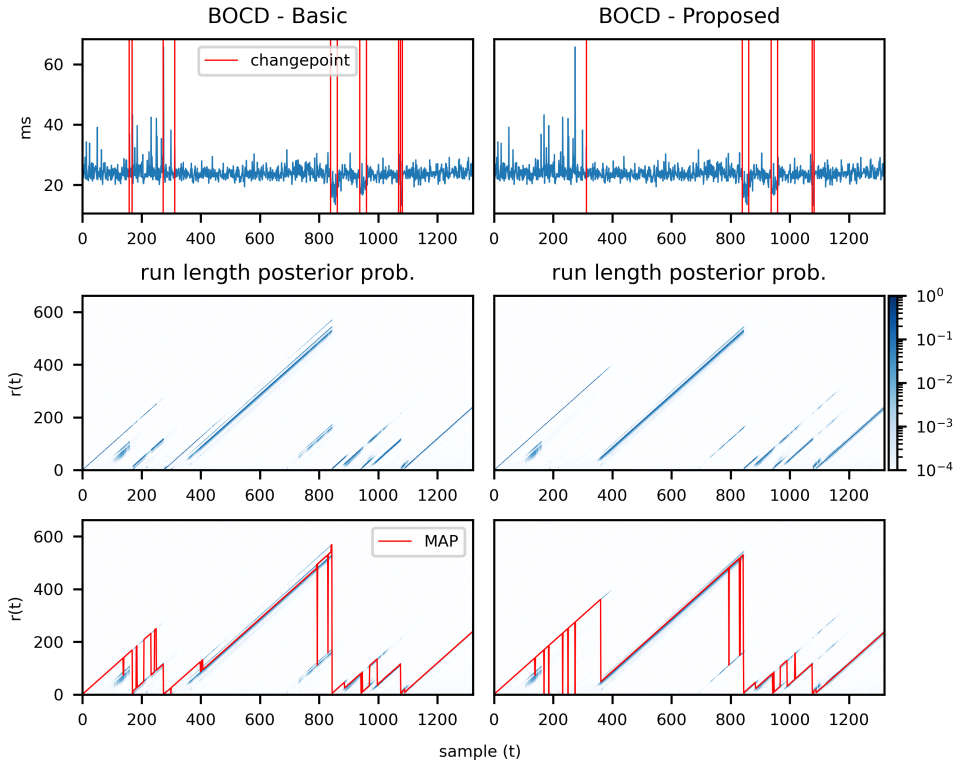


Figure 4.6: BOCD basic and proposed versions. The basic algorithm (left subplot) classified and outlier between $t = 200$ and 400 as a change-point, whereas the proposed version (right subplot) did not.
Example 1 - NDT Dataset, Client 4, rnp-rj, down. RTT.

Lastly, to limit the computational cost to a constant value per iteration, we follow ALTAMIRANO *et al.* (2023) and "prune" the run length posterior keeping only the $K = 50$ most probable run lengths.

4.3.2 Resilience to point anomalies

As discussed in Section 3.2, the BOCD lacks robustness to point anomalies. To remedy this, we propose a strategy similar to that proposed for the classical methods. The difference here is that the BOCD uses a matrix of posterior probabilities and not a simple statistic. The proposed procedure is described below and illustrated in Fig. 4.7:

- Suppose a deviation in the current run (possible change-point) is identified at $t = \tau$. We wait for 4 more additional points (hyperparameter `min_seg` in Appendix A) to verify if this change persists. We use the last value before the deviation $\mathbf{x}(\tau - 1)$ to update the run length posteriors during this stage.

- If the change-point is confirmed, we re-compute the run length posteriors using the actual values $\{\mathbf{x}(\tau - c + 1), \ldots, \mathbf{x}(\tau)\}$ instead of $\mathbf{x}(\tau - 1)$.

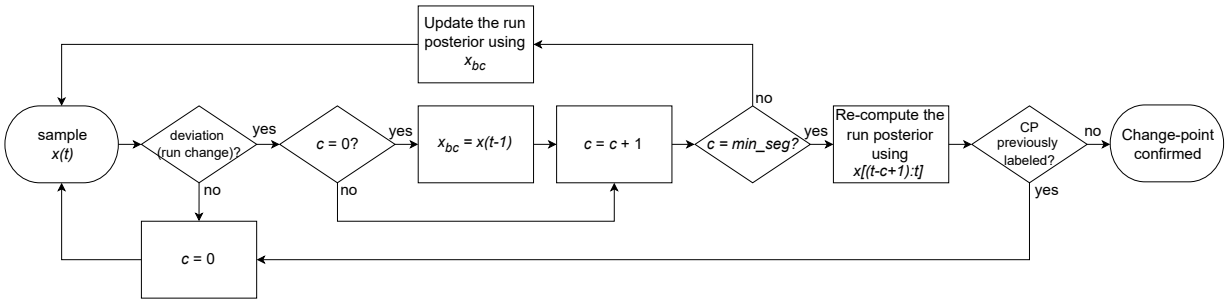


Figure 4.7: BOCD proposed scheme to increase robustness to point anomalies. The basic algorithm (left subplot) classified mny outlier as a change-point, whereas the proposed version (right subplot) did not.

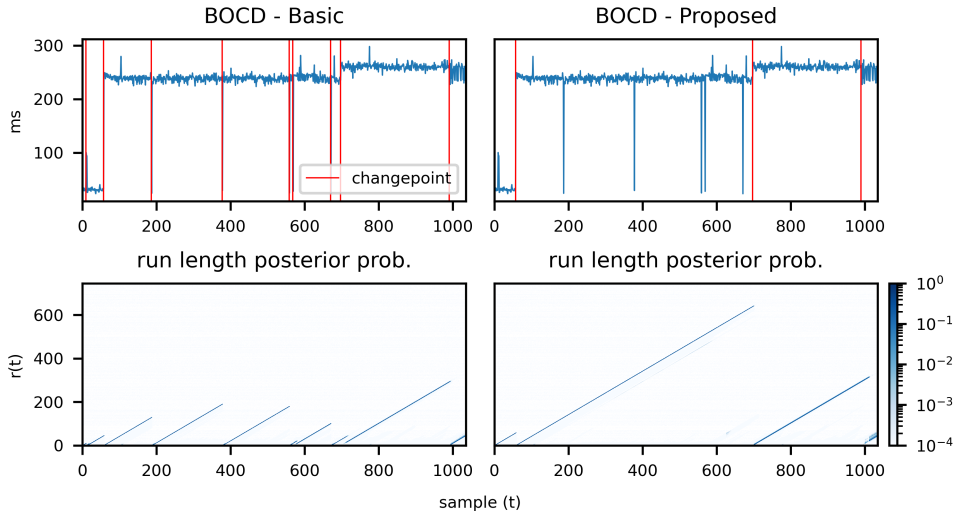


Figure 4.8: BOCD basic and proposed versions
Example 2 - NDT Dataset, Client 4, gig03, down. RTT.

The Fig. 4.8 shows another example of both versions of the BOCD. It can be noted that the basic algorithm is considerably more susceptible to point anomalies than our proposal.

4.4 RRCF framework for change points

As discussed in Section 3.3, the Robust Random Cut Forest (RRCF) original scheme (GUHA *et al.*, 2016) is capable of detecting only anomalies and not change points. To remedy this, we propose a simple framework similar to that proposed for the classical methods. This scheme is illustrated in Fig. 4.9 and described below:

- Whenever an anomaly is identified (the score $S(t)$ deviates from the threshold), check if the next c_{lim} samples also conduct to an anomaly. We consider $c_{lim} = 4$ consecutive anomalies as a change-point.

- Due to the incremental learning characteristic of the model (see Section 3.3),

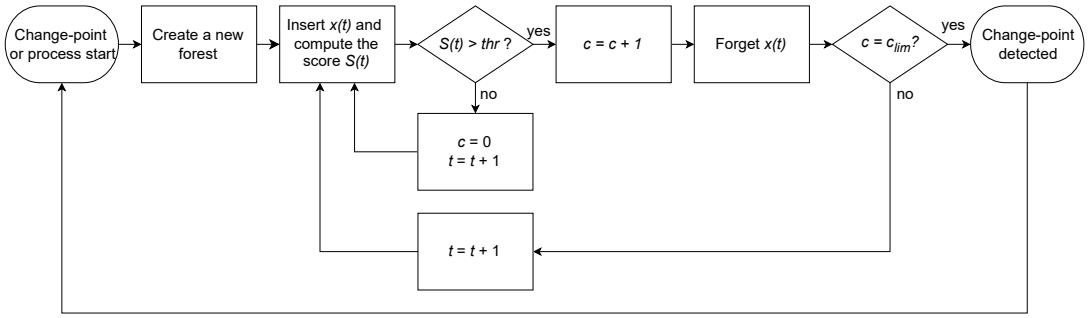


Figure 4.9: RRCF proposed framework for change-points. The main idea is to classify a change-point only after c_{lim} number of consecutive deviations in the test statistic.

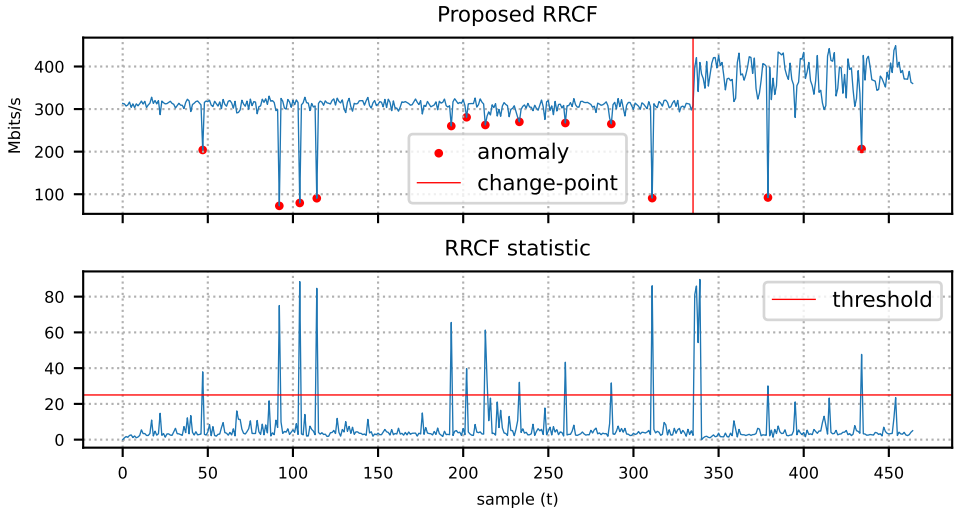


Figure 4.10: RRCF proposed framework - example. The proposed extension of the method identifies both anomalies and change-points - NDT Dataset, Client 1, gru05, down. throughput

whenever a deviation occurs, we need to forget the sample $x(t)$ before evaluating the next one.

- After a change-point, instead of allowing the model to adapt itself incrementally, we re-start the model with a new empty forest.

The Fig. 4.10 shows an example of the proposed framework applied to a download throughput time series. It is possible to note that the change-point between $t = 300$ and $t = 350$ was identified, as well as anomaly points.

Chapter 5

Voting Windows Change-point Detection

The VWCD, first presented in (STREIT *et al.*, 2023), is a new change-point detection method studied by our group. The method is suitable for the online setting and does not assume the previous knowledge of the parameters distributions before and after the change-point. A key feature is that its output is easy to interpret and adjust according to the studied problem.

The method follows the concept of the window-limited GLR (LAI and SHAN, 1999; WILLSKY and JONES, 1976), but is based on a Bayesian setting and ensemble concept. The main idea is illustrated in the conceptual diagram of Fig. 5.1. Each window chooses the time instant that most probably a change-point occurs and *votes* in this instant. As the window slides, each sample is visited for w windows; thus, it can receive up to w votes. These votes are probabilistic and used to decide on a change-point.

5.1 Formalization of the method

Let x_t be the current observation and $\mathcal{D} = \{x_{t-w+1}, \ldots, x_t\}$ a set with the last w observations. Suppose that a change-point occurs at $t = \tau$, $t - w + 1 \leq \tau < t$. Then, using the model of Eq. (2.2) and substituting the density parameters by their MLE estimate $\hat{\theta}_1$ and $\hat{\theta}_2$, the likelihood function is given by

$$p(\mathcal{D} \mid \tau, \hat{\theta}_1, \hat{\theta}_2) = \prod_{i=t-w+1}^{\tau} f_0(x_i \mid \hat{\theta}_1) \prod_{i=\tau+1}^{w} f_1(x_i \mid \hat{\theta}_2) \tag{5.1}$$

To decide on a change-point, usually a composite hypothesis test is performed (BASSEVILLE and NIKIFOROV, 1993, pg. 57). However, in the Bayesian setting, the hypothesis can be accessed using the computed MAP probability of the change-

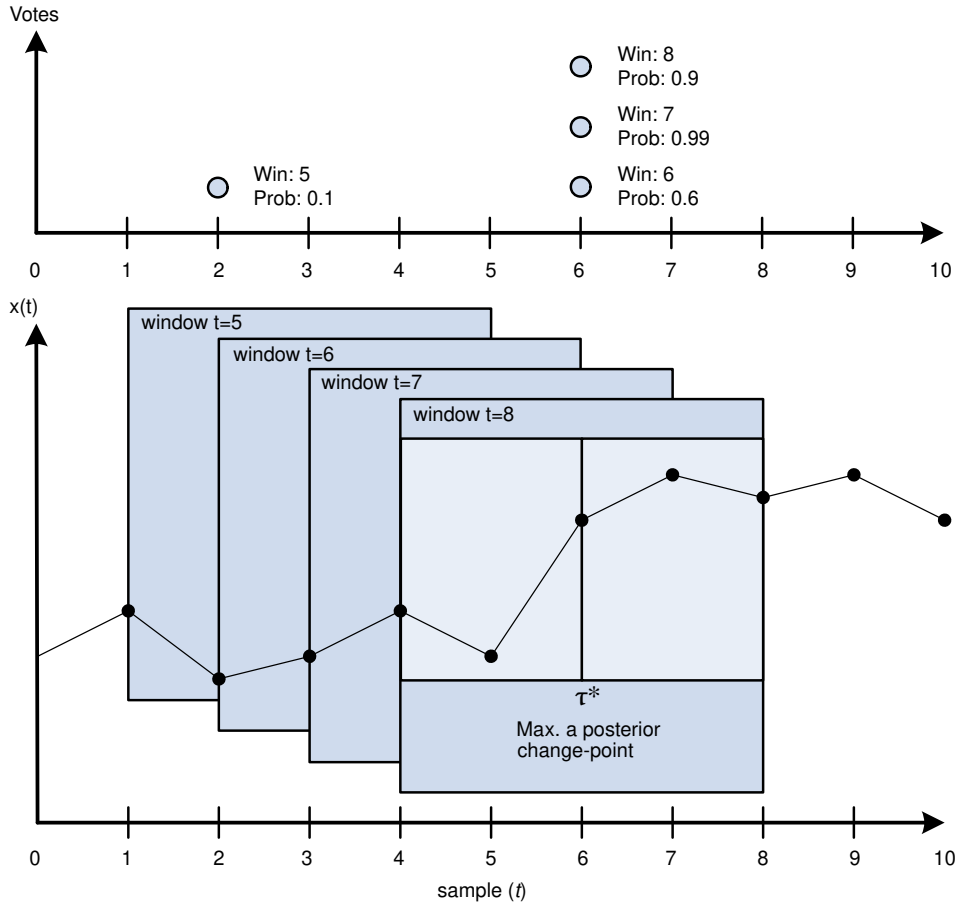


Figure 5.1: VWCD conceptual diagram. As it slides, the window identifies the timestamp with the higher probability of being a change-point, "voting" in this point with a certain probability.

point, assuming a prior $p(\tau)$. Trough the Bayes theorem, omitting $\hat{\theta}_1$ and $\hat{\theta}_2$:

$$p(\tau \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \tau)p(\tau)}{\sum_{\tau=t-w+1}^{w} p(\mathcal{D} \mid \tau)p(\tau)} \tag{5.2}$$

Now, we can use the MAP criteria to find the most probable change-point location:

$$\tau^* = \arg\max_{\tau} p\left(\tau \mid \mathcal{D}\right). \tag{5.3}$$

We store the vote $p(\tau^* \mid \mathcal{D})$, to aggregate it later. There are several ways to compute the vote for each window. In this work, we check if its above a threshold probability p_{thr} (hyperparameter):

$$p(\tau^* \mid \mathcal{D}) \geq p_{thr}. \tag{5.4}$$

Using a uniform distribution is a natural choice for the prior $p(\tau)$. Another possibility is to give less weight to points located near the extremes of the window

using a beta-binomial distribution (hyperparameters α, β), thus avoiding computing the MLE with few data points. We explored these two strategies in our experiments.

5.2 Votes aggregation

To decide on a change-point, we must evaluate the votes with different weights. In this work, we use a simple strategy: we take the mean of the votes and apply a threshold (pa_{thr}). Furthermore, to increase the confidence of the voting scheme, we aggregate the votes for a timestamp $t = \tau$ only after all possible windows have visited that point, *i.e.*, at $t = \tau + w - 1$ and if the number of stored votes for that timestamp is more significant than a threshold value (hyperparameter n_{thr}). Despite this simple scheme, voting schemes are used in several works, *e.g.* (NORDMANN and PHAM, 1999) and (LIU *et al.*, 2021); thus, other strategies for aggregation can be further investigated.

5.3 Hyperparameters tuning

One great advantage of the method is its intuitive hyperparameter adjustment. For example, the minimum number of votes received to classify a timestamp as a change-point (n_{thr}) can be based on the probability associated with the vote. For instance, if one wants to get a high precision, it makes sense to select only votes with a high probability, for example, $p_{thr} = 0.8$ and $pa_{thr} = 0.9$, as we used for the NDT dataset (Table A.2 of Appendix A).

The less intuitive hyperparameter to adjust is the window size (w). However, this difficult occurs not only for VWCD, but for all window-based method (see *e.g.*, (XIE *et al.*, 2023)).

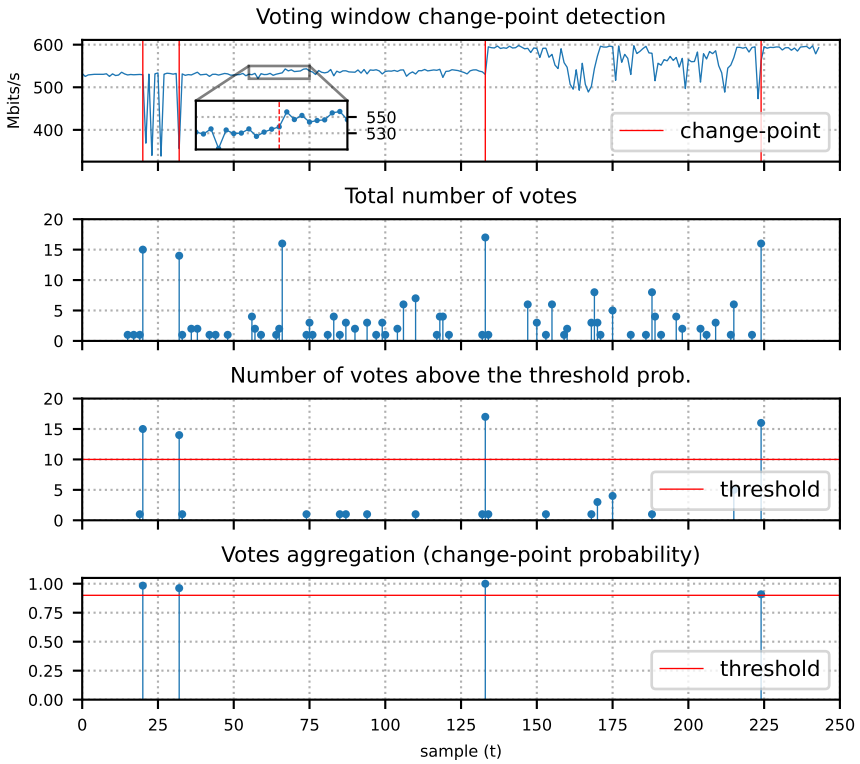
5.4 Time complexity

As stated in (WANG and XIE, 2023), it is reasonable to assume at least $\mathcal{O}(w)$ operations are required to find the MLE of a set of w points (note that it could be optimized using, for example, sufficient statistics). We can consider this same cost to the MAP estimate. Furthermore, similarly to the window-limited GLR, the VWCD procedure scans through all the potential change-point within the sliding window, so the time complexity is $\mathcal{O}(w^2)$.

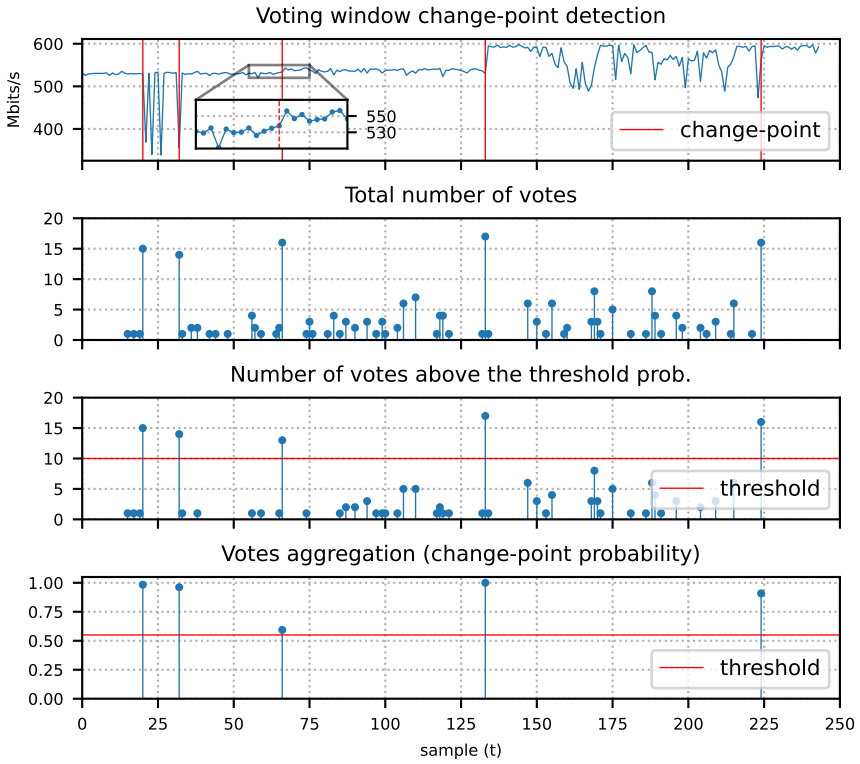
Example The Fig. 5.2 shows an example of the method application. In the subplot Fig. 5.2a, we considered the default hyperparameters used to the dataset (Table A.2), while in Fig. 5.2b is fine-tuned as explained below.

In the Fig. 5.2a, the VWCD method detected the jump in the mean between $t = 125$ and $t = 150$ and the variance and mean change near $t = 25$ and $t = 225$. One can note that there is also a change-point between $t = 50$ and $t = 75$, from 530 Mbit/s to 550 Mbits/s (less than 5% of change). This minor change-point was not detected in this example because the number of votes above the threshold p_{thr} (third row from the top to the bottom) did not achieve the value of 10 votes. However, it would be possible to detect it decreasing p_{thr} and pa_{thr} (the probability threshold for votes aggregation, depicted in the lower row). This is done Fig. 5.2b, illustrating the flexibility of the method.

We provide a pseudo-code for one implementation of the VWCD method in the Appendix A.2.



(a) Default tuning



(b) Fine-tuned with $p_{thr} = 0.4$ and $pa_{thr} = 0.55$

Figure 5.2: VWCD example - NDT dataset - Client 3, gig01, down. throughput

Chapter 6

Experiments

In this section we present the experiments realized with two datasets: the NDT Dataset (built by us using the M-Lab NDT tool), and the labeled Shao Dataset SHAO *et al.* (2017).

6.1 Experiment 1: NDT dataset

6.1.1 The M-Lab project and the NDT

The Measurement Lab (M-Lab) is an open-source project with the objective of measuring the internet, save the data, and make it universally accessible and useful. The project provides an open protocol specification - the Network Diagnostic Tool (NDT), reference implementations for the client and server software, officially supported servers in many countries and public access to the data. In November 2021, 2.9 million NDT tests were executed per day, on average, coming from 239 countries, with the majority of these tests (90%) triggered by the Google-Mlab integration. If one searches for "internet speed" in the Google, the site suggests the user to perform an NDT test. Furthermore, the USA (26%), India (18%) and Brazil (7%) were the top originators of the tests (CLARK and WEDEMAN, 2021; GILL *et al.*, 2022).

In addition to the existing M-Lab server in Brazil we also used the two RNP servers in Rio and São Paulo that implemented the NDT protocol following a Memorandum of Understanding (MOU) recently signed by RNP and M-Lab.

6.1.2 Dataset building

The Fig. 6.1 shows the architecture of the experiment. We omit the implementation details here because Data Engineering is not our focus. Still, it should be mentioned that this stage (design, implementation, and testing of the data collection and processing framework) consumed a significant portion of our work time.

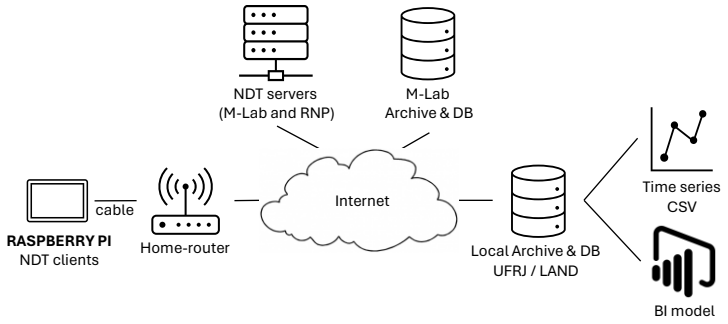


Figure 6.1: Architecture of the NDT data collection experiment

The idea was to use Raspberry PI[1] devices to trigger NDT tests from the user network. The Raspberry is connected directly to the home router using a twisted-pair cable to measure the network condition of the user, not considering wifi limitations. The interval between tests was set to follow an exponential distribution with a mean of 30 minutes, so the expected number of tests per device, at one day, is 48.

From each NDT test, we extract the following time series:

- Download mean throughput (Mbits/s);
- Download mean RTT (ms);
- Upload mean throughput (Mbits/s);
- Upload mean RTT (ms);

It is worth mentioning that the mean values of RTT are not computed and provided in the summary of the NDT test, only the min. value. To overcome this, we processed the JSON file returned by the NDT client and computed the mean value using the values reported along the test.

NDT clients and servers

We used nine Raspberry PI devices, seven installed in the home of student volunteers and two installed in the network of a partner Internet Service Provider. The device hardware specification is shown on Table A.1, and for the client software, we used the `ndt7-client v0.7.0`[2].

When an NDT client is called to target the M-Lab official servers, the application automatically chooses the destination server, considering the client-server physical distance and load balancing. As already stated before, in addition to the official M-Lab servers, we also used in this work two additional NDT servers of the RNP. The M-Lab servers are grouped in pods called *sites*.

[1] `https://www.raspberrypi.com/`
[2] `https://github.com/m-lab/ndt7-client-go/`

Period:	2023-05-01 to 2023-30-11
Number of clients:	9
Number of sites:	9
Metrics:	4
Number of time series:	296
Total number of observations (tests):	45687
Min. num. of observation per series:	102
Max. num. of observation per series:	1495
Mean num. of observations per series:	617

Table 6.1: NDT Dataset description

Time series selection

To build the dataset, we filtered the pairs client-site with more than 100 measures in the period from 2023-05-01 to 2023-30-11 (7 months). We considered the nine clients and the four metrics previously mentioned. So, each time series is specific to a determined client, site, and metric. Also, after this filtering, we got a total of nine sites: gig01...gig04, gru02....gru05, rnp-rj and rnp-sp.

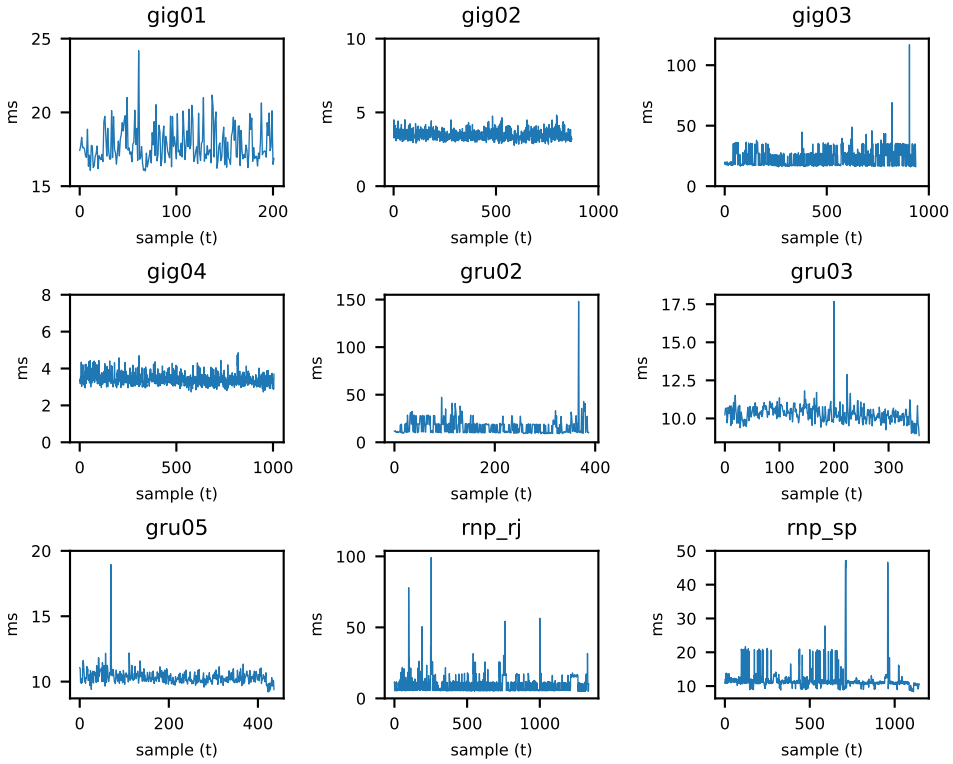
In principle, we would expect to have (9 clients) $\times$ (9 sites) $\times$ (4 metrics) = 324 time series. However, the clients were installed on different dates, and once some M-Lab sites were retired during the experiment, some pairs of client-sites did not achieve the minimum number of measures established (100). In this way, our final dataset turned out to have a total of 296 time series. Aside from other statistics about the dataset, these numbers are summarized in Table 6.1.

To illustrate the dataset, the Fig. 6.2 depicts the time series of download RTT and throughput for Client 8. This figure shows that changes in the throughput are not always correlated with the latency. This justifies monitoring the throughput variable and the RTT for QoS purposes.

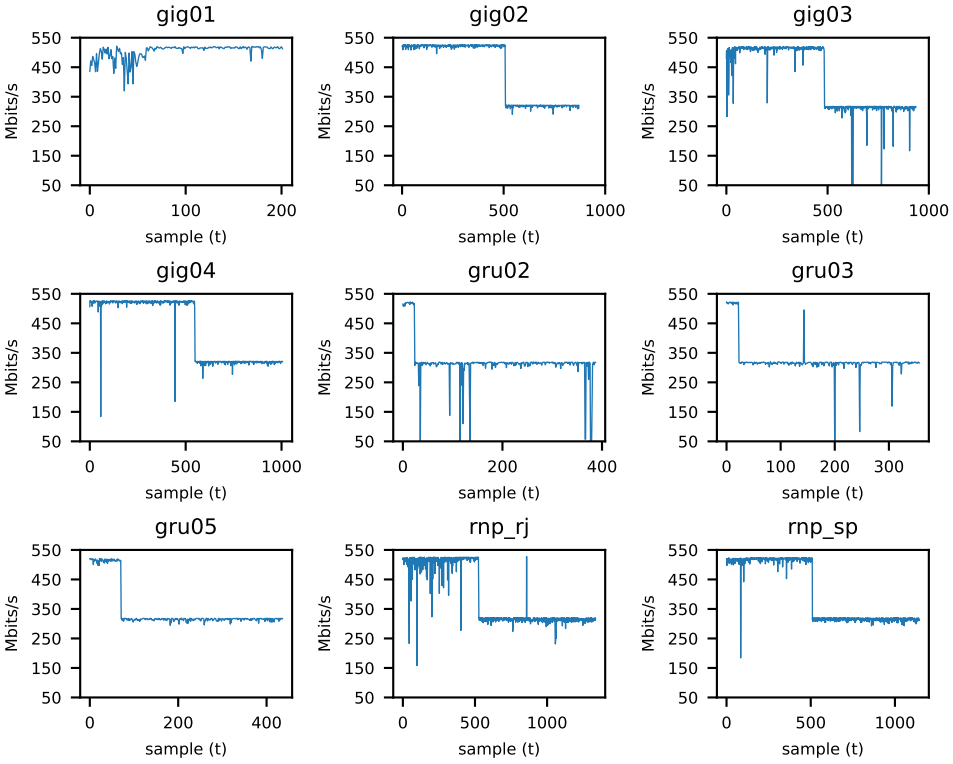
6.1.3 Normality assumption

For the parametric methods (Shewhart, 2S-CUSUM, WL-CUSUM, BOCD and VWCD), we use the Gaussian distribution as the underlying model. This was motivated both by theoretical and empirical reasons. First, our dataset is composed of mean measurements (mean throughput and mean RTT). By the Central Limit Theorem, the limit distribution of the mean of any random sample is Gaussian, even if the random variable itself is not.

By visually inspecting some of the time series, we also noted that, despite the outliers, the normal model seems to adhere to most cases. To illustrate this, in the Fig. 6.3, we plot some time series of the NDT dataset with a change-point detected by the Pelt-NP. In addition, we plot a probability plot (using the normal



(a) Download RTT



(b) Download throughput

Figure 6.2: NDT Dataset - time series of Client 8. This client was installed in the home of a student volunteer connected through the ISP TIM S/A, AS26615.

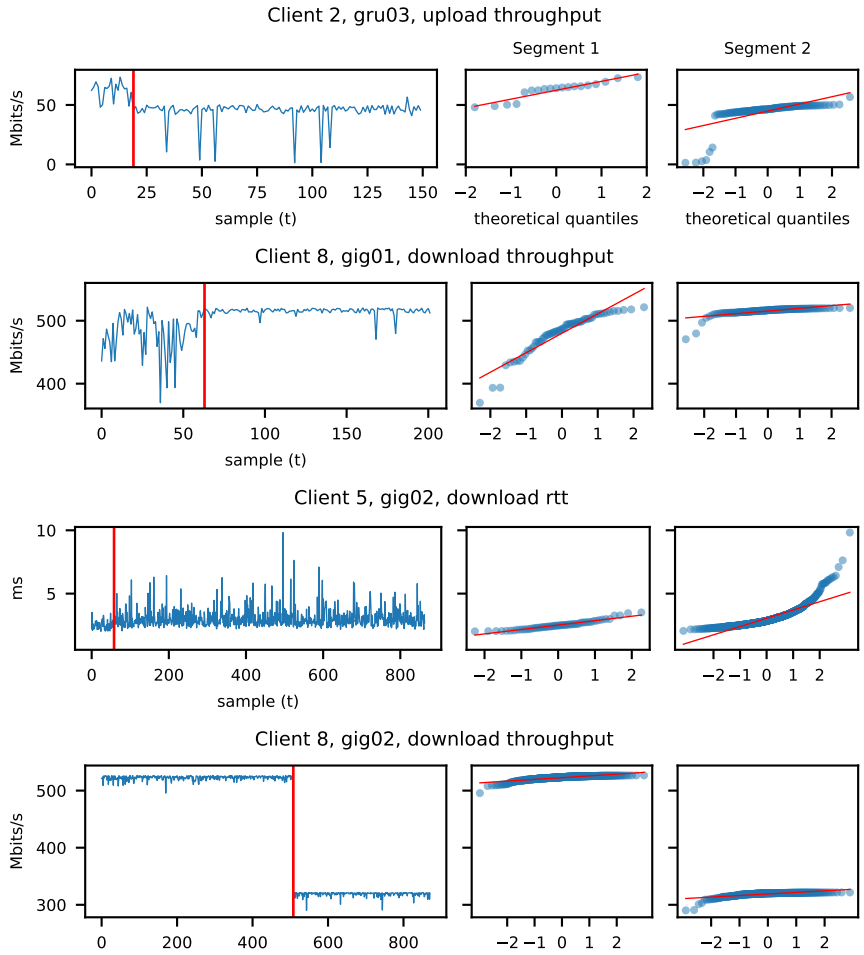


Figure 6.3: NDT Dataset - Normality assumption. We show four examples of time series with a single change-point (detected by the Pelt-NP). The change-point segments the series in two parts, showing a probability plot with the normal distribution as a reference.

distribution as the reference line, in red) for the two segments before and after the change-point. In this figure, it is possible to note that, despite the outliers, the normality hypothesis seems reasonable.

6.1.4 Results

For the classic methods, the hyperparameters were tuned according to recommended values in the literature (MONTGOMERY, 2013). For BOCD and RRCF, we used recommendations and examples provided by their authors (ADAMS and MACKAY, 2007; GUHA *et al.*, 2016). For the VWCD, we adjusted it according to the guideline discussed in Section 5.3.

Since we don't have labels for the NDT dataset, we analyzed the results by visually inspecting the change points identified in some series, the total number of changes identified and the processing time.

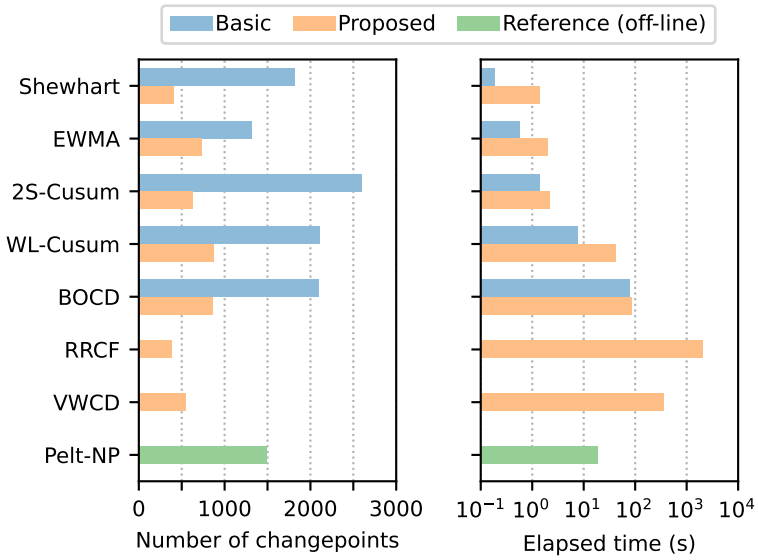


Figure 6.4: NDT Dataset - Number of detected change-points and elapsed time

The Fig. 6.4 shows the number of change points identified and the processing time required by each method to run the total dataset. Since the classical methods do not distinguish anomalies from change points, the number of change points identified using basic implementations was superior to those identified by the proposed implementations.

Regarding the required processing time, it can be observed that the classical methods, even with the proposed framework, are very light: the proposed Shewhart, EWMA and 2S-CUSUM took 2 s to process the entire dataset; WL-CUSUM, on the other hand, took one magnitude order above, since it uses a sliding window to estimate the post-change parameter. The same order of magnitude was observed for BOCD and Pelt-NP. Finally, the VWCD executed in one order of magnitude above BOCD and Pelt-NP (three orders above classical methods). The heavier method was RRCF, four orders above the classical techniques.

The figures 6.5 and 6.6 show two examples of the application of the methods. In both figures, it is possible to note that the classical methods' basic implementation performed very poorly, especially the WL-CUSUM. On the other hand, the proposed framework improved the performance.

In the Fig. 6.5, it is possible to note that all the proposed methods could identify the change-point between $t = 125$ and $t = 150$. The minor change-point between $t = 50$ and $t = 75$ was not detected by the RRCF and VWCD methods. However, adjusting the hyperparameters would be possible to detect it (see, *e.g.*, Fig. 5.2). On the other hand, the VWCD method marked as change-point the oscillation that occurred near $t = 25$, whereas the other methods did not.

In the Fig. 6.6, it is interesting to note that even the Pelt-NP failed to distinguish

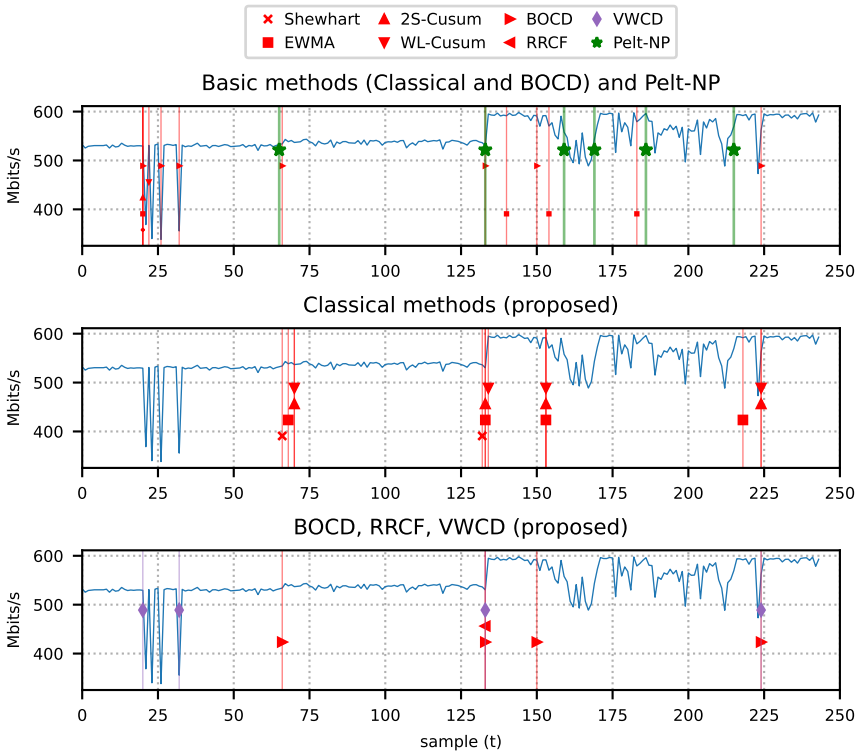


Figure 6.5: NDT Client 3, gig01, down. throughput - Detected change points

some point anomalies (near $t = 100$ and $t = 600$) from change-points. In the same manner, the basic BOCD marked as change points various point anomalies, *e.g.*, near $t = 200$, whereas the proposed BOCD did not. In turn, the VWCD exhibited better performance than the classical methods but also suffered from the influence of two outliers near $t = 550$. These could be mitigated by increasing the vote probability hyperparameter. In the Fig. 6.7, for example, we have set $p_{thr} = 0.99$.

In the case of parametric models, it is possible to identify the model parameters after a change and to verify, in real-time, if this change leads to a QoS worsening. The Fig. 6.8 illustrate a simple application of QoS monitoring. It shows in the y-axis the number of change-points that were responsible for a worsening (download throughput reduction), and in the x-axis the magnitude of the reduction (in megabits per second). The goal is to identify which clients showed a QoS degradation. To generate the graph, we used all the time series available for each client, each corresponding to a specific NDT site. It is possible to note that, for Client 8, all the methods detected at least eight change points with a magnitude greater than 150 Mbits/s, indicating a possible worsening in the network quality. In fact, in Fig. 6.2b, it is clear that Client 8 had a throughput decrease. Thus, this figure provides insights into how the online change-point methods can monitor the network quality in real time and without labels.

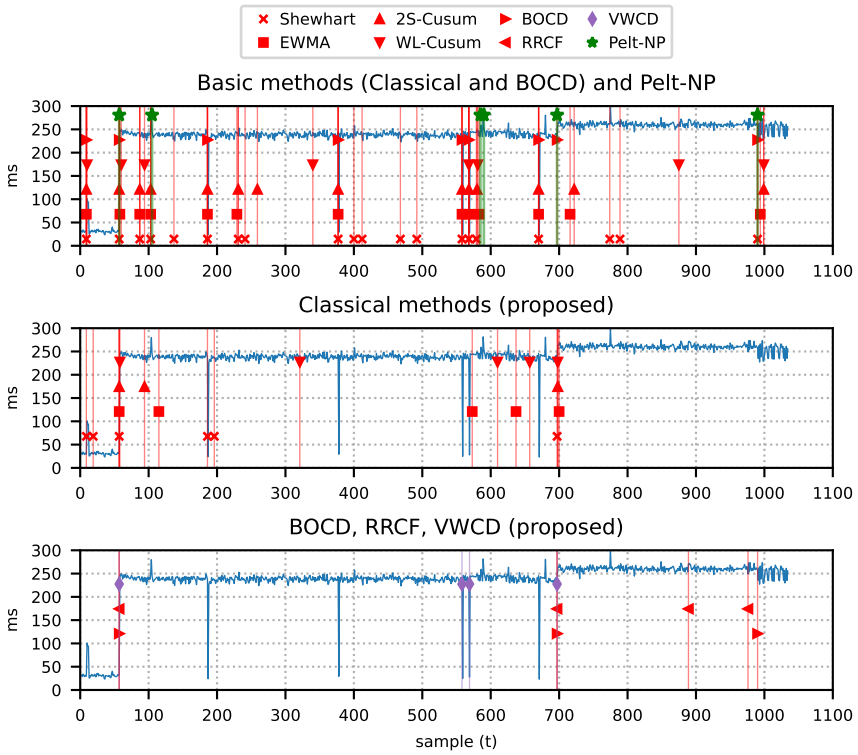


Figure 6.6: NDT Client 4, gig03, down. RTT - Detected change points

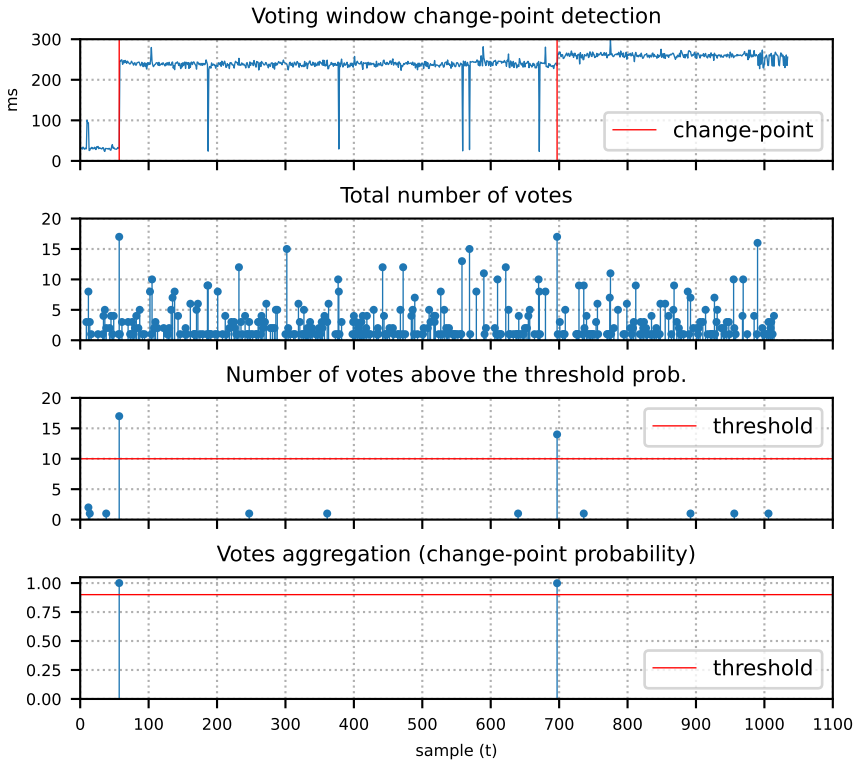


Figure 6.7: NDT Client 4, gig03, down. RTT - VWCD method fine-tuned with a higher voting probability threshold ($p_{thr} = 0.99$)

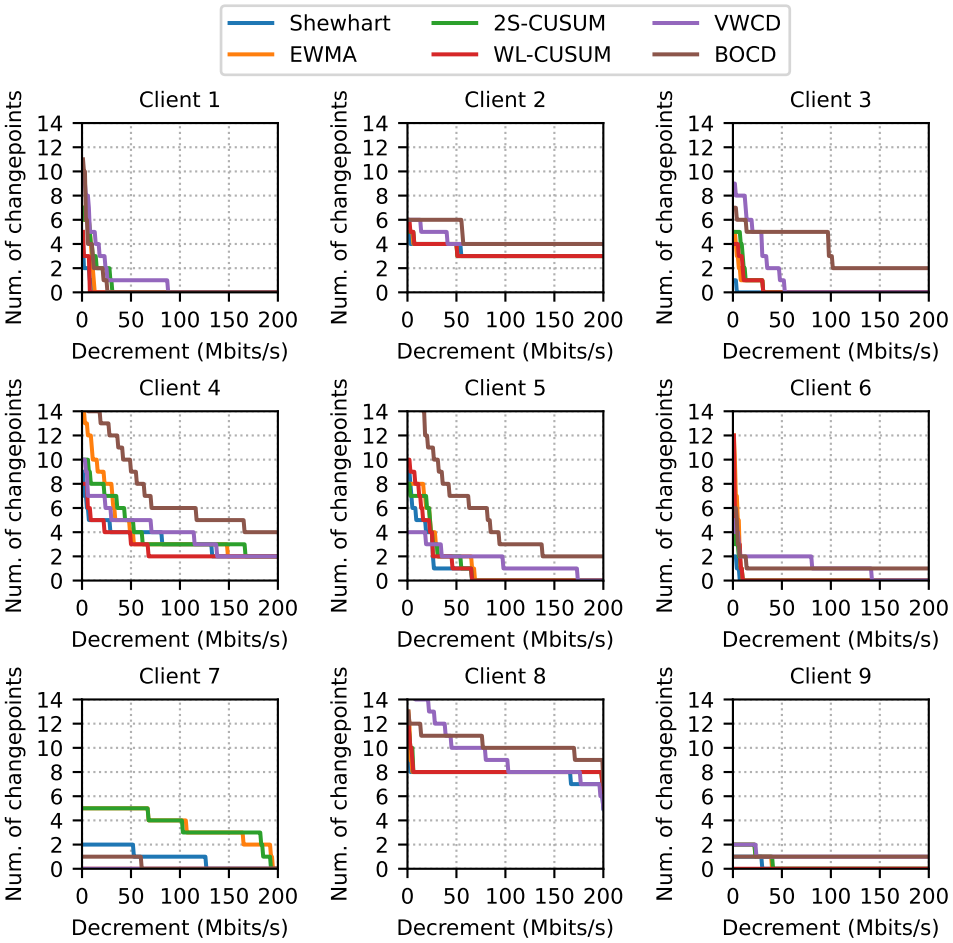


Figure 6.8: A simple unsupervised QoS monitoring application. The value in the y-axis is the number of changes detected so that the download throughput worsened up to the value in the x-axis. It is possible to note that, for Client 8, the BOCD detected 10 decrements with magnitude 150 Mbits/s, indicating worse network quality than the other clients.

6.2 Experiment 2: Shao dataset

6.2.1 Dataset description

In SHAO *et al.* (2017), a labeled dataset of RTT measurements was made publicly available. This dataset consists of a 50 time series from the RIPE Atlas built-in measurements (RIPE, 2024) and was manually labeled by the authors using a methodology capable of evaluating the quality of the labels.

Each time series consists of RTT (ping) measured from the probe to the target server at a regular interval of 4 minutes. The dataset contains 408087 RTT measurements, and the labelers identified 1047 change points. Each time series has (in the mean) about 8000 measures and 20 change points. The Fig. 6.9 depicts some examples of time series with their change-point labels. We plot only the first 1000 samples for better visualization.

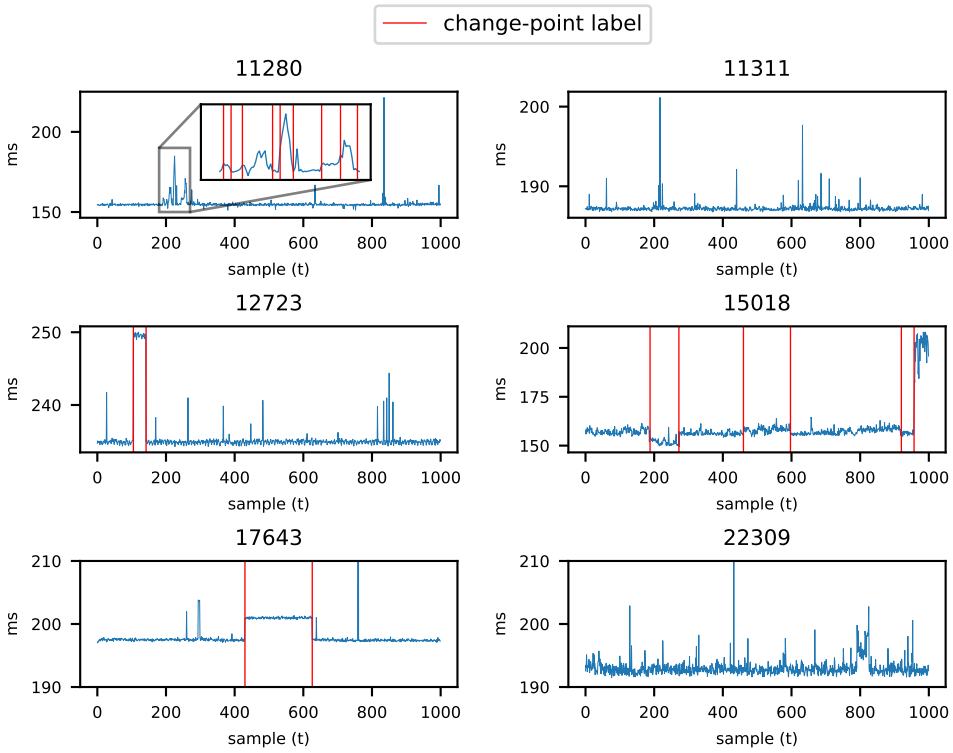


Figure 6.9: Some time series from Shao Dataset (first 1000 samples).

6.2.2 Results

We applied the methods and evaluated the metrics precision, recall and F1 score (Section 2.5). To tune the hyperparameters, we performed a grid search (see Table A.2, Appendix A). The range was selected based on the values used for the NDT Dataset (Section 6.1.4).

In the same way as in the previous experiment and discussion (Section 6.1.3), we use the Gaussian distribution as the underlying model for the parametric methods Shewhart, 2S-CUSUM, WL-CUSUM and VWCD.

We discuss the results by analyzing two examples depicted in Fig. 6.10 and Fig. 6.11. All the proposed methods correctly identified the labeled change points in the first example. In contrast, the performance of the basic methods (classical and BOCD) was not satisfactory: they classified many outliers as change points.

The second example, Fig. 6.11, is more challenging. In Fig. 6.13 we "zoomed" two segments: Segment 2 ($\tau_1 = 188$ to $\tau_2 = 272$); and Segment 5, ($\tau_4 = 597$ to $\tau_5 = 920$). Also, we plotted only the change points detected by the proposed methods and by Pelt-NP. For Segment 2, it is clear that after $t = 210$, the mean changes from 152 ms to 150 ms. This change was not labeled by the human annotator but detected by Pelt-NP, EWMA, 2S-CUSUM and WL-CUSUM. Similarly, in Segment 5, the detected change points "make sense", although they are relatively smaller in magnitude. Again, this example illustrates the difficulty of labeling changes in time

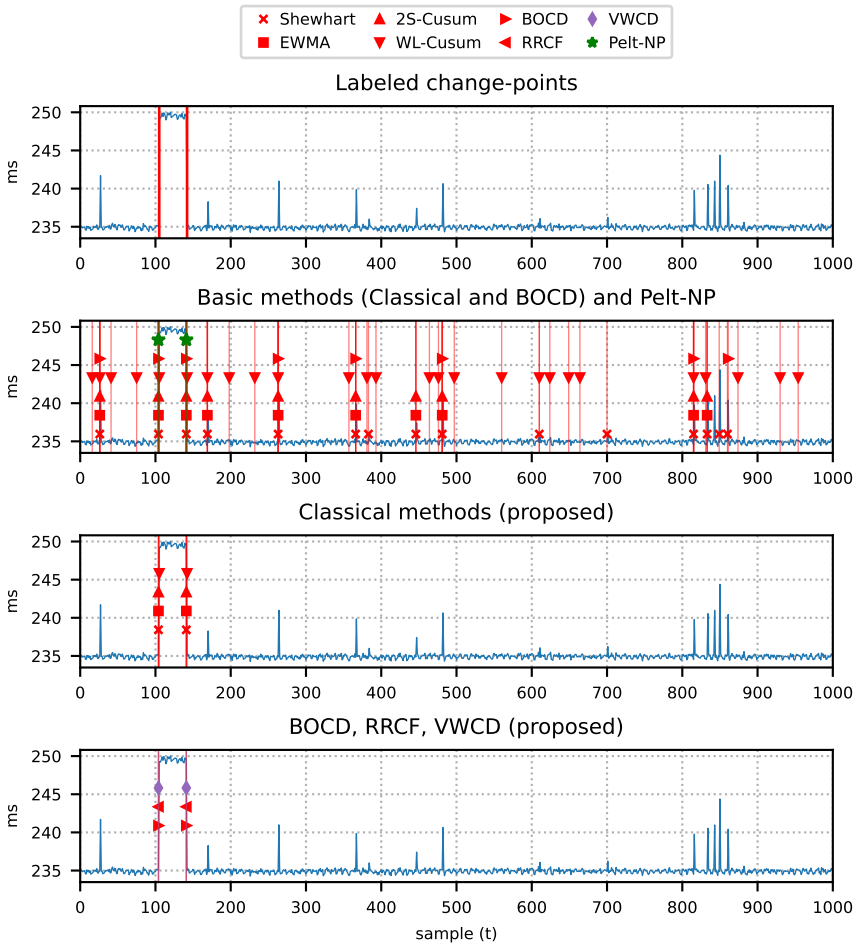


Figure 6.10: Shao Dataset - Results for the time series 12723 (first 1000 samples)

series through visual inspection and, consequently, the difficulty of evaluating the change-point methods. Yet regarding this series, the Fig. 6.12 shows the change points detected by the VWCD method with an alternative tuning increases the recall, once again illustrating the flexibility of the method.

The behavior of the methods in these two examples is helpful to understand the overall performance for the dataset, shown in Fig. 6.14. In this figure, we plot the boxplot of precision, recall and F1-score aggregating these metrics for all the 50 series. Furthermore, we highlighted in red the two best methods for each metric. From figure Fig. 6.14, we note that:

- Regarding the classical methods (Shewhart, EWMA, 2S-CUSUM and WL-CUSUM), the proposed framework significantly improved the performance in terms of precision, recall and F1 sore. The Shewhart, EWMA, 2S-CUSUM had similar performance, whereas the WL-CUSUM had the worst in terms of precision and F1 score.

- In terms of F1 score, excepting WL-CUSUM, the proposed methods had comparable performance with the Pelt-NP. The proposed BOCD had the better

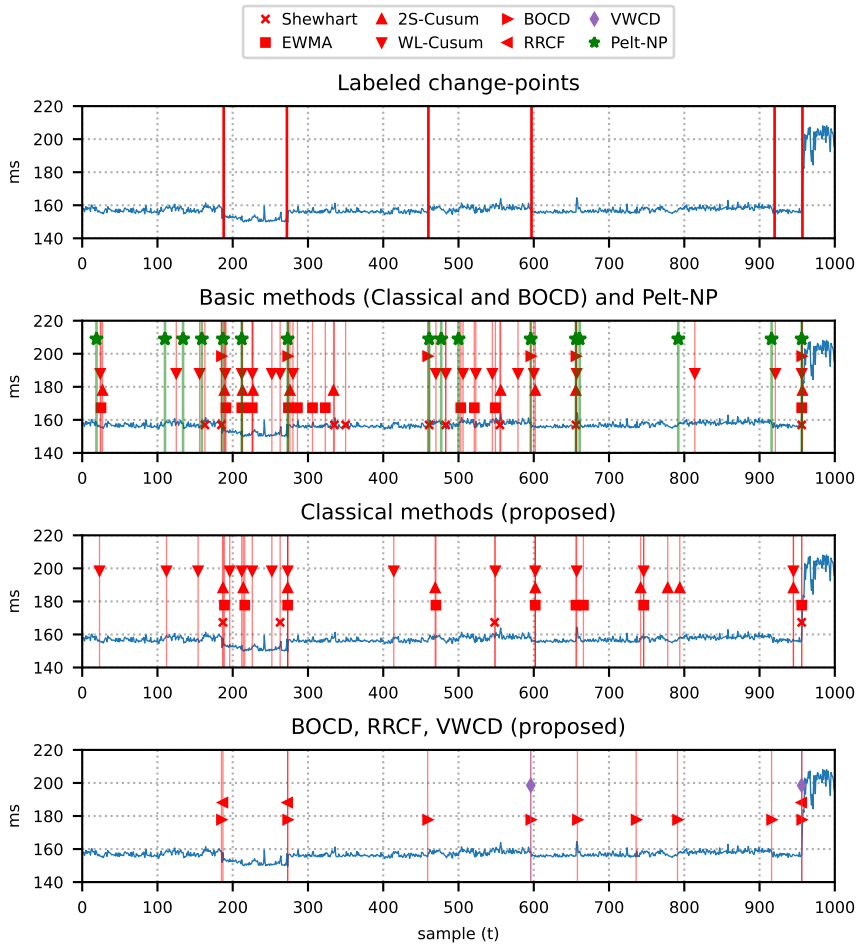


Figure 6.11: Shao Dataset - Results for the time series 15018 (first 1000 samples)

median F1 score (0.62). Still, using the estimated confidence interval (CI) - the notches of the boxplots - we cannot claim a statically different result since the CI of the two boxes overlaps.

- In terms of precision, the VWCD had the better performance, including when comparable with Pelt-NP. The other proposed methods showed similar performance.

- Regarding recall, the proposed BOCD performed similarly to the Pelt-NP.

We plot the confusion matrix for the BOCD (proposed), VWCD and Pelt-NP in Fig. 6.15. Different from Fig. 6.14, in the confusion matrix we considered the total of observations, and not the median value of the 50 series. Because of this, the metrics computed from the matrix can have different values. Furthermore, because the dataset is too imbalanced, the true negative rate (TNR) and the FPR become more difficult to compare.

In the Fig. 6.16, we plotted the number of detected change points and the processing time for each method. We note that:

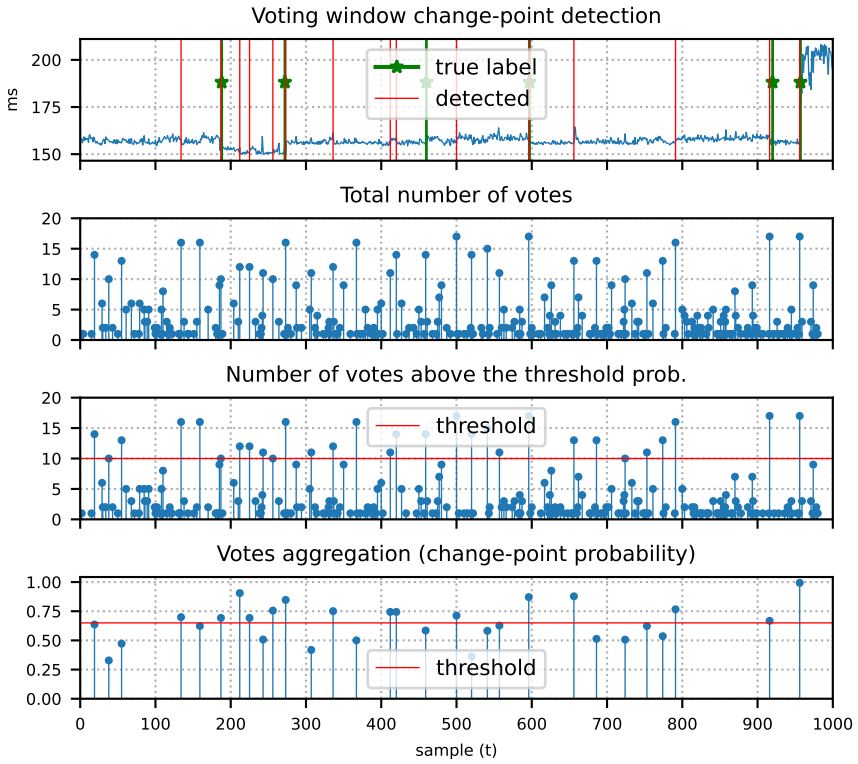


Figure 6.12: Shao series 15018 (first 1000 samples) - VWCD with an alternative tuning increasing the Recall (sensibility).

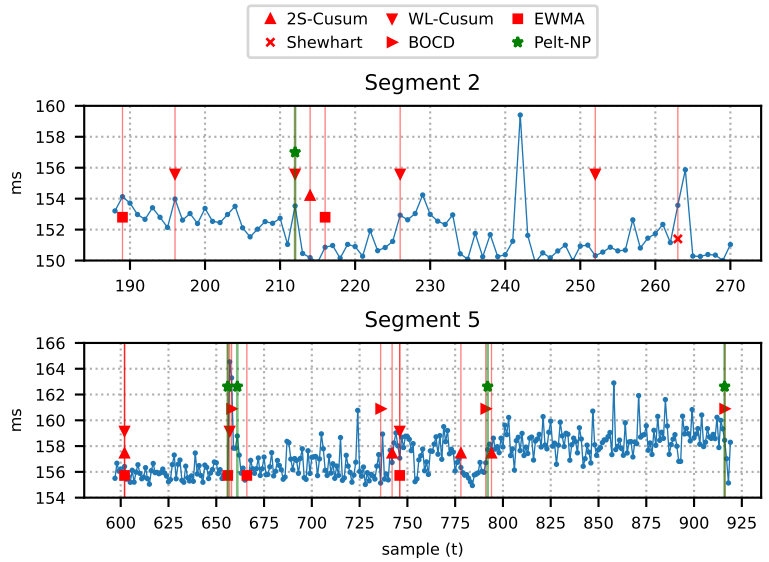


Figure 6.13: Segment 2 ($t = 188 \dots 272$) and Segment 5 ($t = 597 \dots 920$) from Shao time series 15018. Here, we plot the change points detected only by our proposed methods and by Pelt-NP.

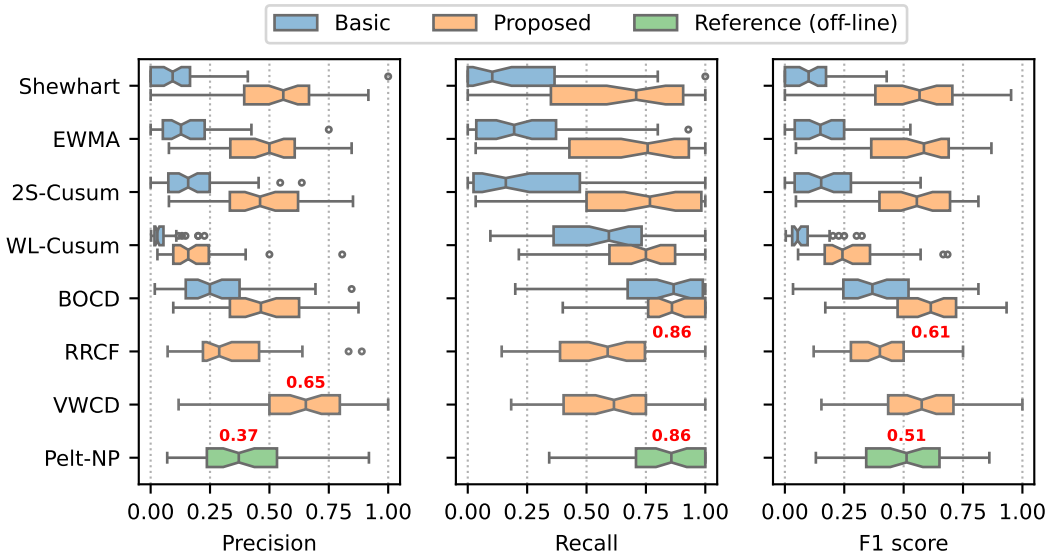
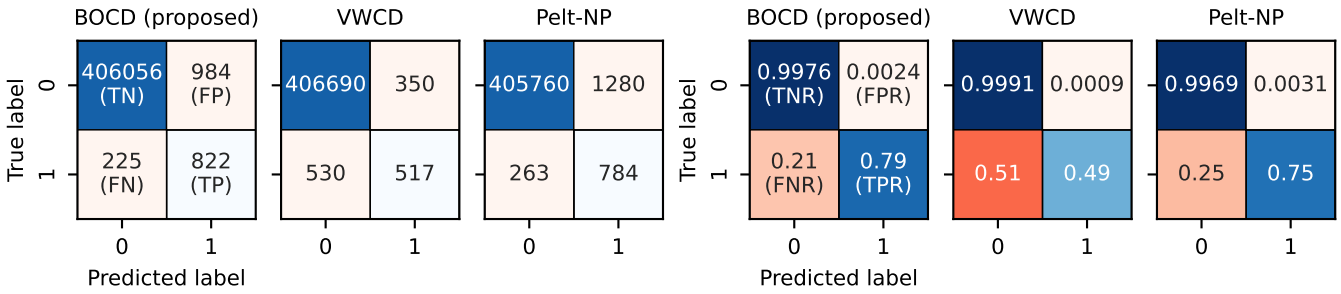


Figure 6.14: Shao Dataset - Boxplot of precision, recall and F1 score



(a) Absolute values

(b) Normalized by row

Figure 6.15: Shao Dataset - Confusion matrix of BOCD (proposed), VWCD and Pelt-NP

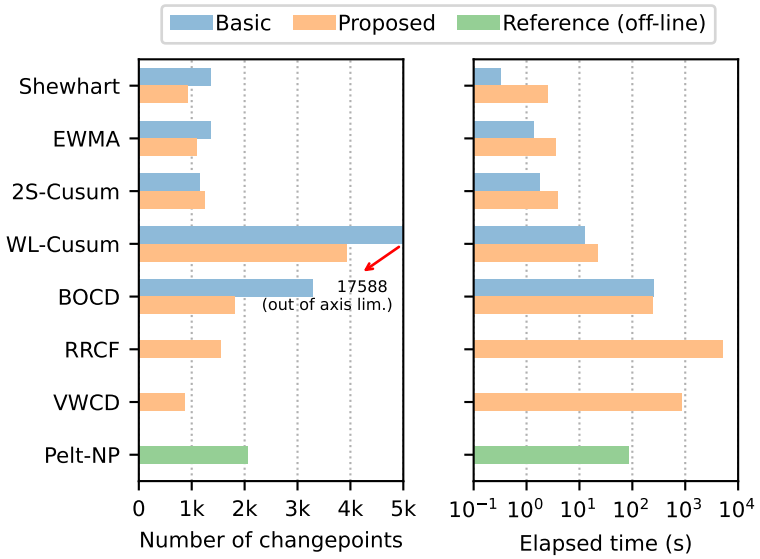


Figure 6.16: Shao Dataset - Number of detected change-points and elapsed time

- The WL-CUSUM proved to be very sensitive, detecting the number of change-points, but most of them are false positives, as reported by the low precision depicted in Fig. 6.14).
- The Pelt-NP also detected many change-points, approximately double the classical methods and VWCD. This yields a high recall in cost of low accuracy, also depicted in Fig. 6.14.
- In terms of execution time, the performance was similar to the NDT dataset discussed in the previous section.

Chapter 7

Conclusions

Using two datasets with real network measurements, first, we showed that the classical change-point methods - Shewhart, EWMA and CUSUM - present challenges to real-world implementation because the theoretical models assume the full knowledge of the distribution before and after the change-point. To overcome this, we proposed a simple framework that improves the parameter estimation and the resilience to outliers. As a result, the performance of the classical methods using our framework was significantly better than the basic implementation, being competitive to the state-of-the-art BOCD and Pelt-NP, this last offline. Furthermore, the classical methods are very light and suitable even for low-resource devices. On the other hand, one weakness is the relatively high variance in recall. This is possible because of the need to wait for the process to stabilize before estimating and start detecting change points.

Using similar ideas of the classical framework, we proposed a simple modification in the BOCD method to enhance its resilience to outliers and a simple scheme to decide on a change-point in the online setting. As a result, we do not have statistical confidence that the F1-score is better than the other methods (except for WL-CUSUM), but we can claim that the recall was the same as Pelt-NP and outperformed the other methods. On the other hand, the hyperparameters are more challenging to adjust.

In addition to the well-established methods and using the same ideas for outlier resilience, we adapted the RRCF, initially designed for anomaly detection, to change-point detection. Despite exhibiting a relatively poor recall and high computational cost, the RRCF outperformed the Pelt-NP in precision. Furthermore, an advantage (besides being online) is that RRCF is intrinsically multivariate. On the other hand, the main drawback of the method is the non-intuitive score threshold hyperparameter.

We also evaluated a new method proposed by our group, the VWCD. A significant advantage of this method, when comparable to the classical ones, is that it

does not assume the knowledge of distribution parameters; they are learned using the MLE approach. Furthermore, based on a voting scheme, the hyperparameters are more intuitive and easier to adjust, and we can easily increase confidence when deciding on a change point. This confidence increase is reflected in the precision: VWCD showed significantly better precision than Pelt-NP, but at the cost of a smaller recall. On the other hand, VWCD is significantly heavier than the classical methods and may require a larger window; furthermore, in its current implementation, the method operates with a delay of one window of samples. There are several ways to take advantage of the voting probabilities. We did not explore those in this work but the different possibilities will be explored by our group in the future.

As demonstrated in various examples throughout the work, labeling change points in real-data time series is not a trivial task for humans, being subjective. This affects the evaluation of the method. Some recent works (BURG and WILLIAMS, 2020; SHAO *et al.*, 2017) seek to minimize this issue by developing specialized tools and methodologies for annotation and metrics evaluation. However, a degree of uncertainty in the labels persists, and the question of evaluating the change-point methods in real data does not seem completely clear.

Finally, we conclude that no single method fits all applications and requirements. However, the online methods studied and proposed in this work showed to be competitive to the state-of-the-art Pelt-NP (offline) when applied to time series of network measurements, enabling their use for practical and real-time applications, such as network quality monitoring. Using domain-specific knowledge to adjust the trade-off between precision, recall and execution time, one can select the method(s) that best fits the requirements.

Appendix A

Reproducibility

A.1 Code, data and hyperparameters

The code and data used in this work are available at the Github repository https://github.com/cleitonmoya/msc_thesis together with instructions to reproduce the experiments.

The author implemented the classical methods and the VWCD using Python 3.9 with standard packages for scientific computing (details in the Github repository).

To compute the change-point metrics, we rely on the benchmark module of SHAO *et al.* (2017) (https://github.com/WenqinSHAO/rtt).

For the BOCD, we used the Python implementation of ALTAMIRANO *et al.* (2023) (https://github.com/maltamiranomontero/DSM-bocd).

For the RRCF, we used the Python package `rrcf` (BARTOS *et al.*, 2019).

For the Pelt-NP, we use the R package `changepoint.np` (https://cran.r-project.org/package=changepoint.npd) over the `rpy2` (https://github.com/rpy2/rpy2) bridge to Python.

The Table A.1 shows the hardware infrastructure used to run the experiments; the Table A.2 lists the hyperparameters values and the range used in the grid search (for the Shao dataset).

Workstation (change-points alghorithms)	
CPU:	Intel Core i7-6700 (4 cores, 8 threads, 3.40Ghz, 4x256kiB L2 cache, 8MiB L3 cache)
RAM:	16 GB
OS:	Windows 10 22H2
Raspberry PI - Data collection	
Model:	Raspberry PI 4 Model B Rev. 1.4
CPU:	Cortex-A72 (4 cores, ARMv8, 1.8GHz)
RAM:	8 GB
OS:	Raspberry PI OS 64 bits (2023-05-03)

Table A.1: Hardware infrastructure

.

A.2 VWCD pseudo-code

The Algorithm 1 presents the pseudo-code for the VWCD method. For vectors and lists, the index begins at 1. For legibility, we assumed that the log-likelihood can be computed even for only one sample, but not that it is not true for the Gaussian distribution (it requires two samples to have a non-null standard deviation).

Algorithm 1 Voting Windows Change-point Detection

function VWCD($\mathbf{x}$, w, α, β, p_{thr}, pa_{thr}, n_{thr}, y_0, y_w)
  ▷ *w: window size* ◁
  ▷ *p_{thr}: threshold prob. for a change at each window pos.* ◁
  ▷ *pa_{thr}: threshold prob. for a change after votes aggregation* ◁
  ▷ *n_{thr}: min. number of votes for a change-point* ◁
  ▷ *α, β: hyperp. for the beta-binom prior* ◁
  ▷ *y_0, y_1: hyperp. for the logistic prior* ◁
  $\boldsymbol{\pi}_w \leftarrow \text{betabinom}(\text{size} = w, \alpha, \beta)$ ▷ *prior prob. for a change at each window pos.*
  $\boldsymbol{\pi}_v \leftarrow \text{logistic}(\text{size} = w, y_0, y_w)$
  V $\leftarrow$ empty_dictionary() ▷ *dictionary with the list of votes for each n*
  $CP \leftarrow$ empty_list() ▷ *list of changepoints*
  $N \leftarrow$ number of elements of $\mathbf{x}$

  for $n = w \dots N$ **do**:
    $\mathbf{x}_w \leftarrow \mathbf{x}[(n - w + 1) : n]$
    LLR $\leftarrow \text{empty\_array}(w)$ ▷ *log-likelihood ratio for each possible cp. in w*
    for $\nu = 1 \dots w$ **do** ▷ *$\mathcal{H}_v$ composite hypothesis*
      $\mathbf{x}_1 \leftarrow \mathbf{x}_w[1 : \nu]$
      $\mathbf{x}_2 \leftarrow \mathbf{x}_w[(\nu + 1) : w]$
      $(\log \mathcal{L}_1, \hat{\boldsymbol{\theta}}_1) \leftarrow \text{mle}(\mathbf{x}_1)$ ▷ *max. likelihood estimation*
      $(\log \mathcal{L}_2, \hat{\boldsymbol{\theta}}_2) \leftarrow \text{mle}(\mathbf{x}_2)$
      $\mathbf{LLR}[\nu] \leftarrow \log \mathcal{L}_1 + \log \mathcal{L}_2$

    ▷ *Compute the vote of window and store it if meets the threshold prob.* ◁
    $(\nu_{map}, p_\nu) \leftarrow \text{map}(\mathbf{LLR}, \boldsymbol{\pi}_w)$
    if $p_\nu \geq p_{thr}$ **then**
      $V[n - w + \nu_{map}]$.append(p_ν)

    ▷ *If the num. of votes for $x[n - w - 1]$ is greater than n_{thr}* ◁
    ▷ *aggregate the votes and decide for a change-point* ◁
    votes $\leftarrow V[n - w + \nu_{map}]$
    $n_{votes} = \text{num\_elements}(\mathbf{votes})$
    if $n_{votes} \geq n_{thr}$ **then**
      agg_vote $\leftarrow$mean(**votes**)
      if agg_vote $\geq pa_{thr}$ **then**
        CP.append($n - w + 1$)
  return CP

Method	**Param.**	**NDT**	**Shao**	**Grid search**	**Ref.**
All	δ	–	5	–	Eq. (2.3)
Classical (all)	w_0	10	10	–	Section 4.1
Classical (all)	c_{lim}	4	4	–	Section 4.2.1
Classical (all)	κ_a	5	5	–	Eq. (4.1)
Classical (all)	α_{norm}	0.01	0.01	–	Section 4.2.3
Classical (all)	α_{stat}	0.01	0.01	–	Section 4.2.4
Classical (all)	cw_{max}	4	4	–	Section 4.2.3
Classical (all)	Δ_{max}	1.2	1.2	–	Section 4.2.3
Shewhart	κ	3	4	$[1, 2, 3, \mathbf{4}]$	Eq. (3.2)
EWMA	λ	0.1	0.5	$[0.1, 0.2, \mathbf{0.5}]$	Eq. (3.3)
EWMA	κ_d	4	4	$[3, \mathbf{4}, 5]$	Section 3.1.2
2S-CUSUM	h	5	6	$[4, 5, \mathbf{6}]$	Section 3.1.3
2S-CUSUM	δ	2	3	$[1, 2, \mathbf{3}]$	Eq. (3.10)
WL-CUSUM	h	5	6	$[4, 5, \mathbf{6}]$	Eq. (3.10)
WL-CUSUM	w_1	20	5	$[\mathbf{5}, 10]$	Eq. (3.11)
BOCD	λ	$1e4$	$1e10$	$[\mathbf{1e10}, 1e20]$	Section 3.2
BOCD	κ_0	0.01	0.5	$[0.01, 0.1, \mathbf{0.5}]$	Section 3.2
BOCD	α_0	0.01	0.01	$[\mathbf{0.01}, 0.05, 0.1]$	Section 3.2
BOCD	ω_0	0.1	1	$[0.1, 0.5, \mathbf{1.0}]$	Section 3.2
BOCD	K	50	50	–	Section 4.3.1
BOCD	`p_thr_rl`	0.05	0.05	–	Section 4.3.1
BOCD	`min_seg`	4	4	–	Section 4.3.2
RRCF	`num_trees`	40	40	–	Section 3.3
RRCF	`tree_size`	100	200	$[75, 100, \mathbf{200}, 256]$	Section 3.3
RRCF	`shingle_size`	2	2	–	Section 3.3.1
RRCF	thr	20	20	$[\mathbf{20}, 25, 30, 35, 40]$	Section 3.3
RRCF	c_{lim}	4	4	–	Section 4.4
VWCD	w	20	20	–	Eq. (5.1)
VWCD	α, β	1	1	$[\mathbf{1}, 5]$	Section 5.1
VWCD	p_{thr}	0.8	0.6	$[\mathbf{0.6}, 0.8]$	Eq. (5.4)
VWCD	pa_{thr}	0.9	0.9	–	Section 5.2
VWCD	n_{thr}	0.5	0.7	$[0.5, \mathbf{0.7}]$	Section 5.2
Pelt-NP	`min_seg`	4	4	–	Section 2.1.1
Pelt-NP	`custom_cost`	MBIC	MBIC	–	Section 3.4

Table A.2: Hyperparameters

References

ABU-MOSTAFA, Y. S., MAGDON-ISMAIL, M., LIN, H.-T., 2012, *Learning from data*. New York, AMLBook.

ADAMS, R. P., MACKAY, D. J., 2007, "Bayesian online changepoint detection", *arXiv preprint arXiv:0710.3742*.

AGGARWAL, C. C., 2017, *Outlier Analysis*. 2 ed. New York, Springer.

ALTAMIRANO, M., BRIOL, F.-X., KNOBLAUCH, J., 2023, "Robust and Scalable Bayesian Online Changepoint Detection", *arXiv preprint arXiv:2302.04759*.

AMINIKHANGHAHI, S., COOK, D. J., 2017, "A survey of methods for time series change point detection", *Knowledge and information systems*, v. 51, n. 2, pp. 339–367.

ATASHGAHI, Z., MOCANU, D. C., VELDHUIS, R. N., PECHENIZKIY, M., 2021, "Unsupervised online memory-free change-point detection using an ensemble of LSTM-autoencoder-based neural networks". In: *8th ACM celebration of women in computing womencourage*.

BARTOS, M., MULLAPUDI, A., TROUTMAN, S., 2019, "rrfc: Implementation of the Robust Random Cut Forest algorithm for anomaly detection on streams", *Journal of Open Source Software*, v. 4, n. 35, pp. 1336.

BASSEVILLE, M., NIKIFOROV, I. V., 1993, *Detection of abrupt changes: theory and application*. Rennes, France, Prentice Hall.

BORROR, C. M., MONTGOMERY, D. C., RUNGER, G. C., 1999, "Robustness of the EWMA control chart to non-normality", *Journal of quality technology*, v. 31, n. 3, pp. 309–316.

BRAEI, M., WAGNER, S., 2020, "Anomaly detection in univariate time-series: A survey on the state-of-the-art", *arXiv preprint arXiv:2004.00433*.

BURG, G., WILLIAMS, C., 2020, "An evaluation of change point detection algorithms", *arXiv preprint arXiv:2003.06222*.

CHANDOLA, V., BANERJEE, A., KUMAR, V., 2009, "Anomaly detection: A survey", *ACM computing surveys (CSUR)*, v. 41, n. 3, pp. 1–58.

CHO, H., KIRCH, C., 2021, "Data segmentation algorithms: Univariate mean change and beyond", *Econometrics and Statistics*.

CLARK, D. D., WEDEMAN, S., 2021, "Measurement, meaning and purpose: Exploring the M-Lab NDT dataset". In: *TPRC49: The 49th Research Conference on Communication, Information and Internet Policy*.

DICKEY, D. A., FULLER, W. A., 1979, "Distribution of the estimators for autoregressive time series with a unit root", *Journal of the American statistical association*, v. 74, n. 366a, pp. 427–431.

FARKAS, K., 2016. "CUSUM anomaly detection". Available at: <`https://www.measurementlab.net/publications/CUSUMAnomalyDetection.pdf`>. [Online; accessed 12-February-2024].

FEARNHEAD, P., LIU, Z., 2007, "On-line inference for multiple changepoint problems", *Journal of the Royal Statistical Society Series B: Statistical Methodology*, v. 69, n. 4, pp. 589–605.

FEARNHEAD, P., RIGAILL, G., 2019, "Changepoint detection in the presence of outliers", *Journal of the American Statistical Association*, v. 114, n. 525, pp. 169–183.

FISCH, A. T. M., ECKLEY, I. A., FEARNHEAD, P., 2022, "A linear time method for the detection of collective and point anomalies", *Statistical Analysis and Data Mining: The ASA Data Science Journal*, v. 15, n. 4, pp. 494–508.

FOX, E. B., SUDDERTH, E. B., JORDAN, M. I., WILLSKY, A. S., 2011, "A sticky HDP-HMM with application to speaker diarization", *The Annals of Applied Statistics*, pp. 1020–1056.

GILL, P., DIOT, C., OHLSEN, L. Y., MATHIS, M., SOLTESZ, S., 2022, "M-Lab: User initiated Internet data for the research community", *ACM SIGCOMM Computer Communication Review*, v. 52, n. 1, pp. 34–37.

GUHA, S., MISHRA, N., ROY, G., SCHRIJVERS, O., 2016, "Robust random cut forest based anomaly detection on streams". In: *International conference on machine learning*, pp. 2712–2721. PMLR.

GUNDERSEN, G., 2019. "Bayesian Online Changepoint Detection". Available at: <https://gregorygundersen.com/blog/2019/08/13/bocd/#barry1992product>. [Online; accessed 7-Frebuary-2024].

GUSTAFSSON, F., 2000, *Adaptive filtering and change detection*. Chichester, England, John Wiley & Sons.

HAN, S., HU, X., HUANG, H., JIANG, M., ZHAO, Y., 2022, "Adbench: Anomaly detection benchmark", *Advances in Neural Information Processing Systems*, v. 35, pp. 32142–32159.

HAWKINS, D. M., 1980. "Identification of outliers". .

HAYNES, K., 2017, *Detecting abrupt changes in big data*. Ph.D. thesis, Lancaster University, United Kingdom.

HAYNES, K., FEARNHEAD, P., ECKLEY, I. A., 2017, "A computationally efficient nonparametric approach for changepoint detection", *Statistics and computing*, v. 27, pp. 1293–1305.

HUNTER, J. S., 1986, "The exponentially weighted moving average", *Journal of quality technology*, v. 18, n. 4, pp. 203–210.

HUSHCHYN, M., ARZYMATOV, K., DERKACH, D., 2020, "Online neural networks for change-point detection", *arXiv preprint arXiv:2010.01388*.

JACKSON, B., SCARGLE, J. D., BARNES, D., ARABHI, S., ALT, A., GIOUMOUSIS, P., GWIN, E., SANGTRAKULCHAROEN, P., TAN, L., TSAI, T. T., 2005, "An algorithm for optimal partitioning of data on an interval", *IEEE Signal Processing Letters*, v. 12, n. 2, pp. 105–108.

JUODAKIS, J., MARSLAND, S., 2023, "Epidemic changepoint detection in the presence of nuisance changes", *Statistical Papers*, v. 64, n. 1, pp. 17–39.

KILLICK, R., FEARNHEAD, P., ECKLEY, I. A., 2012, "Optimal detection of changepoints with a linear computational cost", *Journal of the American Statistical Association*, v. 107, n. 500, pp. 1590–1598.

KNOTH, S., 2022. "spc: Statistical Process Control - Calculation of ARL and Other Control Chart Performance Measures". Available at: <https://cran.r-project.org/web/packages/spc/index.html>. [online; accessed 12-April-2024].

KONIG, D., 1931, "Graphok es matrixok (Hungarian)[Graphs and matrices]", *Matematikai és Fizikai Lapok*, v. 38, pp. 116–119.

KRISHNAN, H., 2020. "How NASA uses AWS to protect life and infrastructure on earth". Available at: <https://web.archive.org/web/20240405051037/https://www.amazon.science/how-nasa-uses-aws-to-protect-life-and-infrastructure-on-earth>. [online; accessed 05-April-2024].

LAI, T. L., SHAN, J. Z., 1999, "Efficient recursive algorithms for detection of abrupt changes in signals and control systems", *IEEE Transactions on Automatic Control*, v. 44, n. 5, pp. 952–966.

LI, J., FEARNHEAD, P., FRYZLEWICZ, P., WANG, T., 2022, "Automatic change-point detection in time series via deep learning", *arXiv preprint arXiv:2211.03860*.

LIU, F. T., TING, K. M., ZHOU, Z.-H., 2008, "Isolation forest". In: *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE.

LIU, F. T., TING, K. M., ZHOU, Z.-H., 2012, "Isolation-based anomaly detection", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, v. 6, n. 1, pp. 1–39.

LIU, J., YANG, D., ZHANG, K., GAO, H., LI, J., 2023, "Anomaly and change point detection for time series with concept drift", *World Wide Web*, v. 26, n. 5, pp. 3229–3252.

LIU, Z., ZHANG, Z., LIU, Y., 2021, "Power Grid Security Risk Assessment Method Based on Weighted Voting Ensemble Machine Learning Algorithm". In: *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, pp. 607–613. IEEE.

LOWRY, C. A., MONTGOMERY, D. C., 1995, "A review of multivariate control charts", *IIE transactions*, v. 27, n. 6, pp. 800–810.

LUCAS, J. M., SACCUCCI, M. S., 1990, "Exponentially weighted moving average control schemes: properties and enhancements", *Technometrics*, v. 32, n. 1, pp. 1–12.

MATIAS, R., CARVALHO, A. M., ARAUJO, L. B., MACIEL, P. R., 2011, "Comparison analysis of statistical control charts for quality monitoring of network traffic forecasts". In: *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 404–409. IEEE.

MONTGOMERY, D. C., 2013, *Introduction to Statistical Quality Control*. 7 ed. New York, Wiley.

MOUCHET, M., 2020, *Modélisation robuste du délai Internet et schémas de mesure intelligents pour l'automatisation des réseaux overlay*. Ph.D. Thesis, Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire.

MOUCHET, M., VATON, S., CHONAVEL, T., ABEN, E., DEN HERTOG, J., 2020, "Large-scale characterization and segmentation of Internet path delays with infinite HMMs", *IEEE Access*, v. 8, pp. 16771–16784.

MURPHY, K. P., 2023, *Probabilistic Machine Learning: Advanced Topics*. London, England, MIT press.

NELSON, L. S., 1982, "Control charts for individual measurements", *Journal of Quality Technology*, v. 14, n. 3, pp. 172–173.

NORDMANN, L., PHAM, H., 1999, "Weighted voting systems", *IEEE Transactions on Reliability*, v. 48, n. 1, pp. 42–49.

OLTEANU, M., ROSSI, F., YGER, F., 2022, "Challenges in anomaly and change point detection", *30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2022), Oct 2022, Bruges, Belgium*.

PAGE, E. S., 1954, "Continuous inspection schemes", *Biometrika*, v. 41, n. 1/2, pp. 100–115.

PANG, G., SHEN, C., CAO, L., HENGEL, A. V. D., 2021, "Deep learning for anomaly detection: A review", *ACM computing surveys (CSUR)*, v. 54, n. 2, pp. 1–38.

POLUNCHENKO, A. S., SOKOLOV, G., TARTAKOVSKY, A. G., 2013, "Optimal design and analysis of the exponentially weighted moving average chart for exponential data", *arXiv preprint arXiv:1307.7126*.

RAMACHANDRAN, K. M., TSOKOS, C. P., 2020, *Mathematical statistics with applications in R*. 3. ed. Oxford, UK, Academic Press.

RAZALI, N. M., WAH, Y. B., 2011, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests", *Journal of statistical modeling and analytics*, v. 2, n. 1, pp. 21–33.

RIPE, 2024. "RIPE Atlas Built-in measurements". Available at: <`https://atlas.ripe.net/docs/built-in-measurements/`>. [Online; accessed 17-Frebuary-2024].

ROBERTS, S., 1959, "Control Chart Tests Based on Geometric Moving Averages", *Technometrics*, v. 1, n. 3, pp. 239–250.

SANTOS, G. H., MENDONÇA, G., DE SOUZA, E., LEÃO, R. M., MENASCHÉ, D. S., 2019, "Análise nao supervisionada para inferência de qualidade de experência de usuários residenciais". In: *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 958–971. SBC.

SCHMIDL, S., WENIG, P., PAPENBROCK, T., 2022, "Anomaly detection in time series: a comprehensive evaluation", *Proceedings of the VLDB Endowment*, v. 15, n. 9, pp. 1779–1797.

SHAO, W., 2017, *Measurement-based inter-domain traffic engineering: scalability, data interpretation and network event visibility*. Ph.D. thesis, Telecom ParisTech, Paris, France.

SHAO, W., ROUGIER, J.-L., PARIS, A., DEVIENNE, F., VISTE, M., 2017, "One-to-one matching of RTT and path changes". In: *2017 29th International Teletraffic Congress (ITC 29)*, v. 1, pp. 196–204. IEEE.

SHEWHART, W. A., 1929, "Control of quality of manufactured product", .

STREIT, A., SANTOS, G. H., LEÃO, R. M., E SILVA, E. D. S., MENASCHÉ, D., TOWSLEY, D., 2021, "Network anomaly detection based on tensor decomposition", *Computer Networks*, v. 200, pp. 108503.

STREIT, A., RIBEIRO, M., LEÃO, R. M., DE SOUZA E SILVA, E., TOWSLEY, D., 2023, "Residential Traffic Profiles: What Could Be Learned from Lightweight Measures During the Pandemic?" *Available at SSRN 4646454*.

SU, W.-X., ZHU, Y.-L., LIU, F., HU, K.-Y., 2013, "On-line outlier and change point detection for time series", *Journal of Central South University*, v. 20, n. 1, pp. 114–122.

TAKEUCHI, J.-I., YAMANISHI, K., 2006, "A unifying framework for detecting outliers and change points from time series", *IEEE transactions on Knowledge and Data Engineering*, v. 18, n. 4, pp. 482–492.

TARTAKOVSKY, A., NIKIFOROV, I., BASSEVILLE, M., 2015, *Sequential analysis: Hypothesis testing and changepoint detection*. Boca Raton, Florida, CRC press.

TARTAKOVSKY, A. G., POLUNCHENKO, A. S., SOKOLOV, G., 2013, "Efficient Computer Network Anomaly Detection by Changepoint Detection Methods", *IEEE Journal of Selected Topics in Signal Processing*, v. 7, n. 1, pp. 4–11. doi: 10.1109/JSTSP.2012.2233713.

TRUONG, C., OUDRE, L., VAYATIS, N., 2020, "Selective review of offline change point detection methods", *Signal Processing*, v. 167, pp. 107299.

WALD, A., 1947, *Sequential analysis*. New York, Dover Pvublication, Inc.

WANG, H., XIE, Y., 2023, "Sequential change-point detection: Computation versus statistical performance", *Wiley Interdisciplinary Reviews: Computational Statistics*, p. e1628.

WILLSKY, A., JONES, H., 1976, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems", *IEEE Transactions on Automatic control*, v. 21, n. 1, pp. 108–112.

XIE, L., ZOU, F., XIE, Y., VEERAVALLI, V. V., 2021, "Sequential (Quickest) Change Detection: Classical Results and New Directions", *IEEE Journal on Selected Areas in Information Theory*, v. 2, n. 2, pp. 494–514. doi: 10.1109/JSAIT.2021.3072962.

XIE, L., MOUSTAKIDES, G. V., XIE, Y., 2023, "Window-limited CUSUM for sequential change detection", *IEEE Transactions on Information Theory*.

XIMENES, D., MENDONÇA, G., , G. H., DE SOUZA, E., LEÃO, R. M., MENASCHÉ, D. S., 2018, "O Problema de Detecção e Localização de Eventos em Séries Temporais Aplicado a Redes de Computadores". In: *Anais do XVII Workshop em Desempenho de Sistemas Computacionais e de Comunicação*. SBC.

ZHANG, N. R., SIEGMUND, D. O., 2007, "A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data", *Biometrics*, v. 63, n. 1, pp. 22–32.

ZHOU, H., ZHU, H., WANG, X., 2024, "Change point detection via feedforward neural networks with theoretical guarantees", *Computational Statistics & Data Analysis*, v. 193, pp. 107913.