Tampere University

Anton Kondratev

ONLINE CHANGE DETECTION FOR ION-MOBILITY SPECTROMETRY READINGS

Master of Science Thesis
Mechanical Engineering
Examiners: Philipp Müller, Risto Ritala, Antti Vehkaoja
January 2021

ABSTRACT

Anton Kondratev: Online Change Detection for Ion-mobility Spectrometry Readings
Master of Science Thesis
Tampere University
Robotics and Advanced Robotics
January 2021

Classification of scents using machine learning methods is of importance in many fields. Ability to classify scents can be used for safety, recreational purposes or quality control in the food industry[1]. Nowadays portable gas analyzers exist, whose functionality can be potentially extended. For example, Virtual Reality helmets can be equipped with scent generating appliance giving more immersion. This study is aimed to implement and analyze various change detection algorithms for readings from Electronic Nose (eNose). The task is to detect the point where readings become stable after presenting a scent source to the eNose. Knowledge about change point in the eNose readings is important for further classification tasks.

The samples in the form of time-series were taken using Ion Mobility Spectrometry based eNose device ChemPro100i. The device has 14 channels. The time-series data of each channel is assumed to have a transient and stable phase. The task of implemented algorithms is to find the end of the transient phase. The algorithms implemented in this work are based on log-likelihood ratio test, which follows shifts in mean values. The majority of the considered algorithms are various extensions of the Cumulative Sum (CUSUM) algorithm.

The readings do not always provide useful results because certain channels do not react to certain scents. The other problem is that the ground truth points were determined manually. The manually determined ground truth points are subjective, which affects accuracy of the algorithms. As a result, the change points were detected successfully for good data sets with clear and visually detectable change points. For the data sets, where change points are not visually detectable performance of the algorithms was worse.

From classification methods point of view, small error in detections will not play a big role because in case of two similarly bad or good data sets the results will be similar as well. This provides possibility to characterize different readings giving possibility to use it as features in classification techniques.

For the data sets classified as good, all the algorithms performed almost equally well. For the other classes of data sets the CUSUM, Multivariate Max CUSUM and Matrix Form CUSUM performed the best and the Bayesian Online Change Point detection algorithm performed worst.

Keywords: change detection, CUSUM, algorithms

PREFACE

Many thanks to my supervisors Philipp Müller and Antti Vehkaoja for their priceless help and endless patience in guiding this thesis. Finishing this thesis is a vital part of my life and the supervisors were playing a very important role. This thesis supported my interest in developing my knowledge in the field of machine learning. I would like to thank my colleagues for the collaboration, sharing their knowledge with me, and providing a friendly work environment that gave me confidence.

Tampere, 18th January 2021

Anton Kondratev

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Classifying scents from unknown sources can be potentially useful in preventing chemical threats [2], an inspection of food quality [1], and also for example for recreational purposes. In recreational meaning, it can add a new modality when using virtual reality devices [1]. One application we are interested in is smelling a scent at location A and transferring this scent to a location B [3]. For transferring scent from location A to B it needs to be accurately classified. Classification of scent really means a classification of Volatile Organic Compounds (VOCs) emitted by scent source and for simplicity they are called scents in this work. Different techniques may be applied for the scent classification. The one implemented technique utilizes the "K nearest neighbors "-algorithm [1]. A suitable technology for gathering samples of VOCs is ion mobility spectrometry (IMS). The IMS was chosen because it enables real-time monitoring, fast analysis and quick response time. Commercial IMS devices are relatively affordable and portable [1].

The ion mobility spectrometry based devices also called "eNose" are intended to detect scents. The ion mobility spectrometry is a study about ions moving in gases under an electric field. The principle of IMS devices is the separation of ions moving in inert gases (also called "buffer gas") under the impact of an electric field [4].

Readings from the IMS device used in this work often have a transient and stable phase when measured in a controlled setting. The controlled setting means that a scent is presented to the eNose device and readings are changing until stabilisation. The readings of the IMS-device require 20-30 seconds to stabilize. The time before stabilizing of the readings is called transient phase. The stable phase is the time after the transient phase. Different scents have different lengths of transient phases [3, p. 3], which means, that we have to wait long enough for the measurement to stabilize [1].

The classification algorithms mentioned in [1] work with data from both the transient and the stable phases. Nonetheless, including the transient phase into the classification usually increases the misclassification rate. The reason why samples only from the stable phase are often used for classification is that they show more robust results [3, p. 2]. However transient phase may contain valuable information for classification itself and might help accelerating classification algorithm. From the transient phase, features can be extracted, which are possibly useful for the classification process. Such features can be a variance of samples, length of the transient phase, etc. Thus we have several options: using samples only from the stable phase, using features from the transient phase or using a mixture of them. For distinguishing phases there is a need to know the time when the transient phase ends. The phases of a time-series can be distinguished manually or by means of the change detection algorithms. Often the change detection algorithms are borrowed from the statistical control theory [5][6]. These algorithms consider either data points one after another or construct log-likelihood ratio test of samples. This work is intended to answer the questions:

1. Is it possible to find change points in the IMS readings by means of the change detection algorithms?
2. What is the computational cost?
3. How reliable these algorithms are?
4. What is the most effective algorithm?

Chapter 2 briefly describes the theoretical aspects of the ion mobility spectrometry and the devices using this approach. Chapter 3 tells about the necessity of change point detection algorithms and methods applied in this thesis. All implemented algorithms are discussed in Chapter 4. Chapter 5 evaluates results and describes the data sets used in this work. Chapter 6 discuss the achieved results and tries to answer the main questions of this work.

2 ION MOBILITY SPECTROMETRY

In IMS devices ions move through the buffer gas impacted by an electric field with a certain velocity. Velocity correlates with the mobility of a specific ion. Mobility of an ion is calculated:

$$K = \frac{v}{E}$$

where v is the velocity and E is the electric field. The faster the ion the greater its mobility [4]. Eventually, ions are separated by their differences in mobilities in space or time. The parameters that can affect an IMS experiment are temperature, pressure, and humidity. All these parameters are usually taken into account, and they are provided when reporting the results of the measurements.

There are several types of IMS devices [4]. Some of them are:

- Drift Tube Ion Mobility Spectrometry
- Travelling Wave Ion Mobility Spectrometry
- Trapped Ion Mobility Spectrometry
- Differential Mobility Analyzer

Figure 2.1 summarises different types of IMS technologies.

Drift tube ion mobility spectrometry (DTIMS). This type of IMS is considered classic. The key concept of the DTIMS is a uniform electric field, which propagates through the drift region. The drift region is space with applied weak uniform electric field and gas, which does not have directional flow. Analytes move through the drift region caused by the electric field.

Travelling wave ion mobility spectrometry (TWIMS). The structure of the TWIMS technology is similar to DTIMS except that the electric field is not uniform. The electric field oscillates, producing electric waves, which push ions towards the mass analyzer.

Trapped ion mobility spectrometry (TIMS). TIMS devices consist of 3 main parts: entrance funnel, analyzer, and exit funnel. The ions entering the TIMS are positioned in an electrical field by the gas flow. Gas flow pushes the ions forward with the force proportional to their collision cross-section value [4], which is calculated from the mobility value. The electrical field holds ions at certain places in the drift region separating them. The ions can be released by manipulating the electrical field.

Differential Mobility Analyzers (DMA). Ions drift axially in a drift gas between two electrodes. When a voltage is applied particles are diverted from their straight paths. Classification is performed using the electrical mobility of the particles. The DMA devices are typically functioning under constant electric field and ambient pressure. DMA is able to perform measurements that are not possible with DTIMS. The DMA is usually utilized to detect large analytes.

Field Asymmetric Ion Mobility Spectrometry (FAIMS). FAIMS is also known as Differential Mobility Spectrometry (DMS) or Differential Ion Mobility Spectrometry (DIMS). FAIMS operates like a ion mobility filter. After entering the drift region, ions are impacted by high-voltage asymmetric waves with combination of static waveform. The ions move towards the upper plate impacted by high voltage and towards the bottom plate with lower voltage with different vertical displacement. Eventually, certain ions annihilate after collision with the plates and specific ions move into the detector. Thus FAIMS operates more like a filter for ions. FAIMS operates under ambient pressure. This technology is not able to provide collision cross-section information.

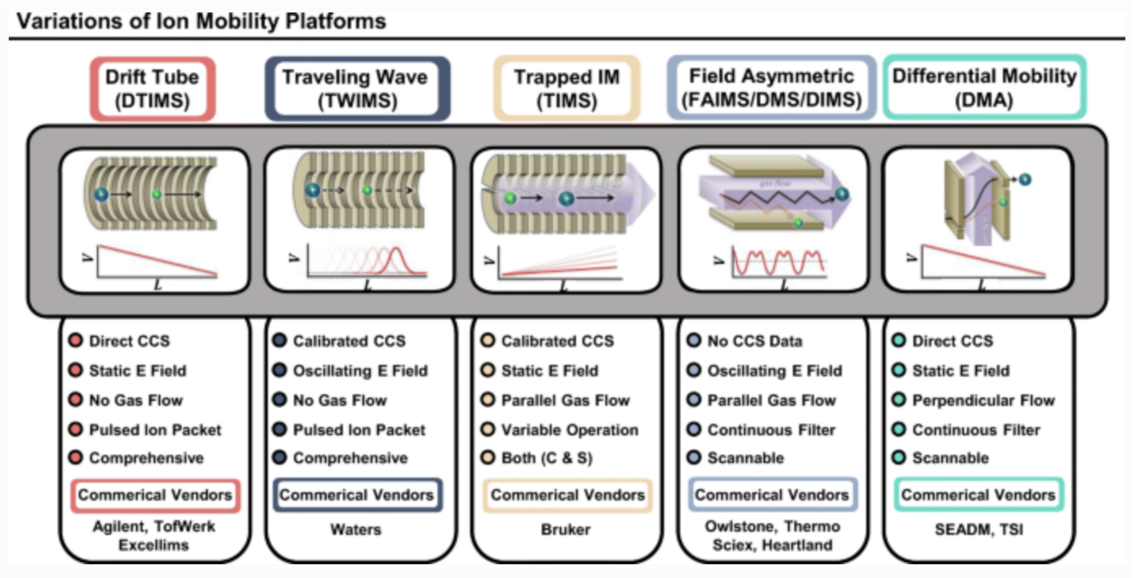


Figure 2.1. *Types of IMS. Source [4].*

In this work, we used the ChemPro100i eNose developed by Environics. The purpose of the ChemPro100i is to detect chemical agents in ambient air such as dangerous gases, warfare agents, etc. This device does not belong to any type of IMS devices mentioned above but it uses a proprietary technique developed by Environics. The ChemPro100i utilizes an "open-loop aspiration type IMS" configuration patented by Environics, which uses IMS with different types of semiconductor sensors. This configuration allows the detector to characterize gaseous chemical compounds. The open-loop configuration has advantages over other types of IMS because it does not have any consumable materials (dryers, desiccants, etc), except a dust filter that require regular maintenance.

The ChemPro100i contains several sensors:

- Miniaturized open-loop IMS sensor with 16 measuring channels
- Tin Oxide Semiconductor gas sensor (SCCell)
- Metal oxide semiconductor gas sensors (MOS)
- Humidity sensor
- Pressure sensor
- Mass flow sensor
- Temperature sensors

The Tin Oxide SCCell outputs changes in resistance, which is caused by the absorption of ions in the sample gas on the sensor surface. The MOS sensors output change in resistance as well. The resistance change on the surface of the MOS sensor is caused by absorption and chemical reactions. The MOS sensors are sensitive to different chemicals depending on the coating material used. When performing measurements the software related to the ChemPro100i gives access to the values of all of these sensors. The workflow of the ChemPro100i consists of several steps. The mobility of various ions differs due to the difference in their molecular weight, charge, and geometry between compounds [7]. The eNose measures the ions with seven separate electrode pairs. The electric fields in the IMCell are continuously switched between positive and negative polarities.

3 METHODS

In this work, we are mainly interested in the Miniaturized open-loop IMS sensor, which gives 16 measuring channels. There are two control channels among these 16. The control channels must always be zero. When performing measurements 14 channels often have readings with transient and stable phases.

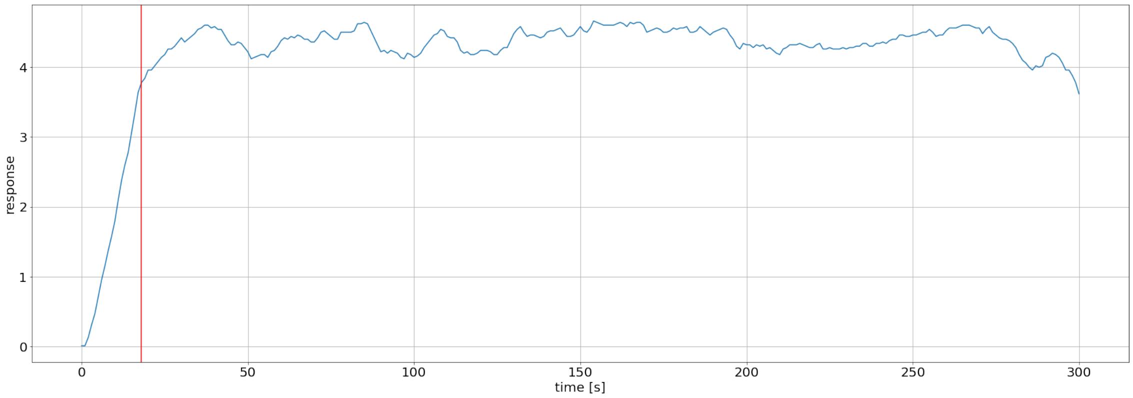


Figure 3.1. *Readings of one channel. Jasmine smelled in a flask.*

Figure 3.1 demonstrates example readings of one channel. The red line depicts the approximate point, where the phase changed from transient to stable. The readings are not always as clear as in Figure 3.1. Sometimes readings do not have any clear distinction between the phases. The existence of transient and stable phases usually depends on the scent being measured and the conditions of the measurement. Measurement performed from a bottle, where ambient air, temperature, and humidity do not affect the results usually has more clear phases. On the other hand, the measurement performed on a plate is more uncertain in terms of phases. Results often depend on the scent being measured. For example, the reaction caused by jasmine scent is usually more significant than the reaction caused by grape scent.

The task of this work is to detect points, where the phase changes from the transient to the stable. For this task change point detection algorithms are used. The analysis for the change points can be performed online and offline. The online algorithms could also be run on offline data. The offline algorithms can be applied only after the whole time-series data is available for the analysis. The online algorithms assume that the data points enter an algorithm sequentially as time goes on. Since ChemPro100i provides readings immediately and we want to classify scent sources in real time we are interested in the online change detection techniques.

Change detection algorithms usually decide whether the change point is present or not by following shifts in a mean or variance (standard deviation). Since the data shows the noticeable change in the standard deviation (Chapter 5.1) we decided to test changepoint algorithms, that follow the shift in standard deviation. For exploring the possibility of following the shift in standard deviation we performed Fisher's p-value test for running variance ratio with significance level $\alpha = 0.01$ and size of moving window 20. The Fisher's test can be used for comparing variances of two samples. Since initially calculated sample variance is sequentially compared to the variances of the moving window the test must be suitable to capture the difference. Fisher's test considers two hypotheses:

$$H_0 : \sigma_0^2 = \sigma_1^2$$
$$H_1 : \sigma_0^2 \neq \sigma_1^2$$

the decision about rejecting H_0 is calculated as follow

$$f = \frac{\bar{\sigma}_0^2}{\bar{\sigma}_1^2} \tag{3.1}$$

$$p = 2min[P(F \leq f), 1 - P(F \leq f)] \tag{3.2}$$

where F is F cumulative distribution function. If p-value is less than $\alpha = 0.01$, then we can reject the null hypothesis.

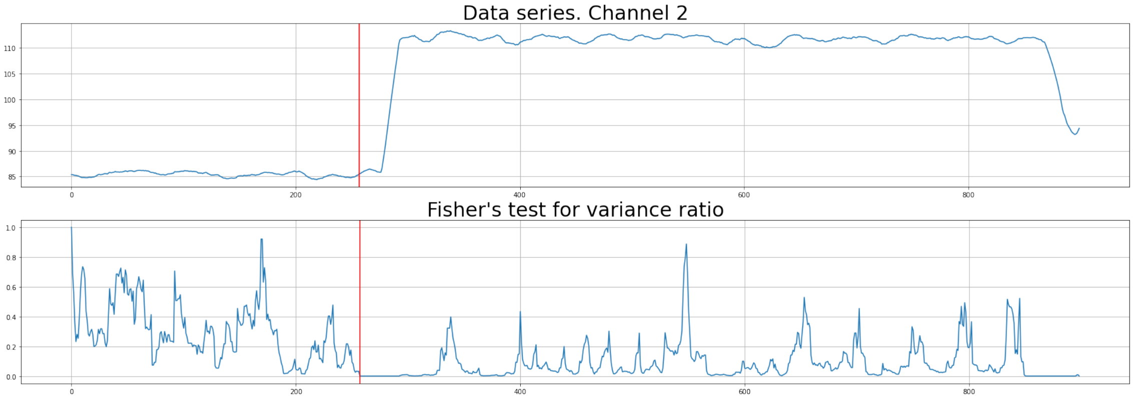


Figure 3.2. *Fisher's test for running variance ratio*

Figure 3.2 shows the results of Fisher's test for the jasmine scent. The red line on the plots shows the time step, where the null hypothesis was rejected. As we can see the variance changes before the transition phase starts. For ensuring this we performed a test run of the CUSUM algorithm adjusted to follow variance on one of the channels.

Figure 3.3 demonstrates failing of this approach. The top plot shows the decision function of the CUSUM-variance algorithm. The red line designates the point, where the transition phase starts. The second plot is an enlarged plot of the decision function. The enlarged area is marked on the top plot with blue. On the second plot, we can clearly see that the variance has changed although the transition phase has not started yet. The third plot shows the normalized version of the raw readings. The peak on the third plot shows the transition phase. The bottom plot shows raw readings from the ChemPro. Thus adjusting changepoint algorithms to follow a shift in variance does not suit our needs.

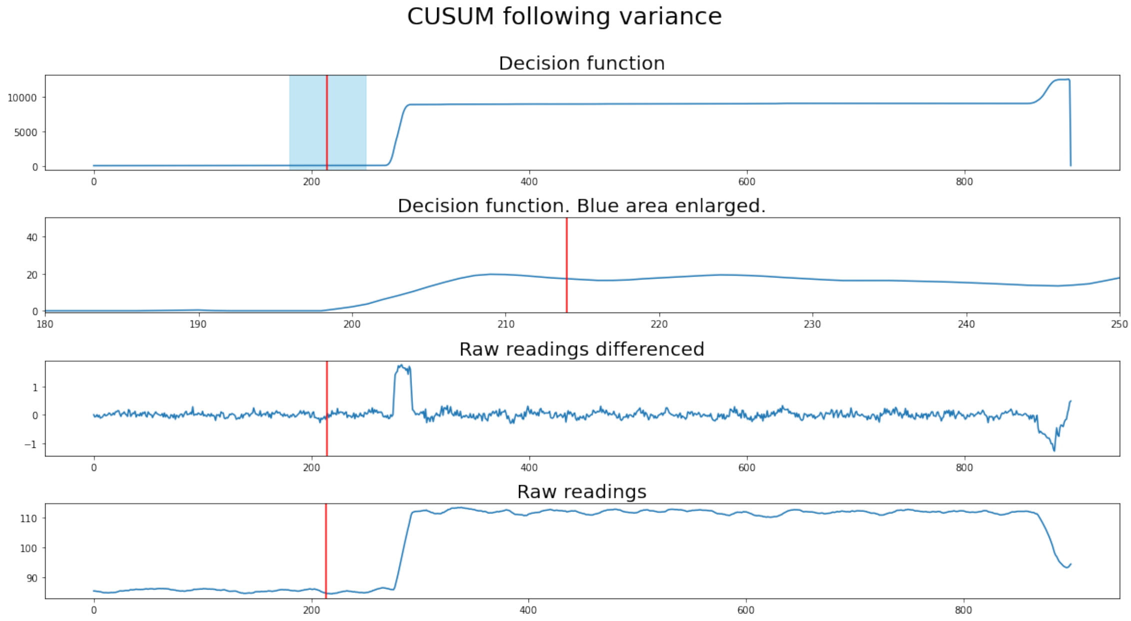


Figure 3.3. *CUSUM test for following variance*

Considering the possibility of using a shift in mean values in change point detection algorithms we performed a t-test on readings of one channel. Figure 3.4 demonstrates results of the t-test. The top and the middle plots demonstrate raw and normalized readings from the ChemPro100i. The bottom plot shows the p-values of the running t-test for a mean shift. The red line on the plots demonstrates the moment when the null hypothesis was rejected.

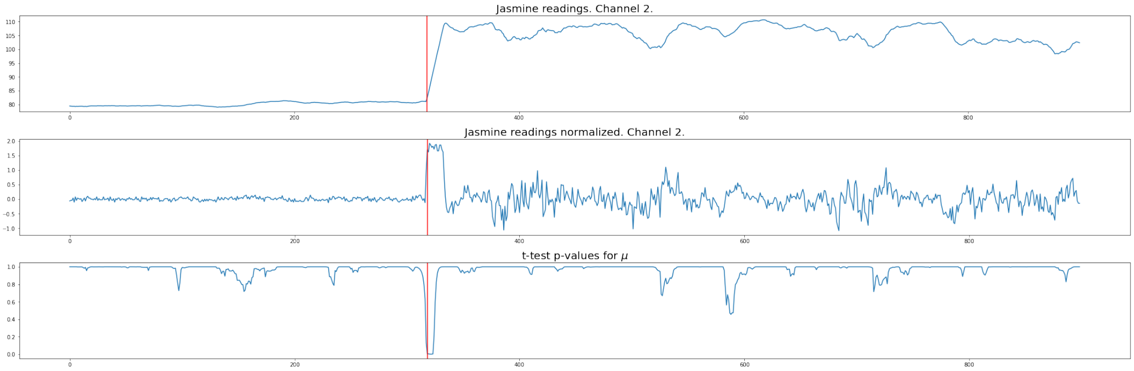


Figure 3.4. *Results of t-test for change in mean*

The test was performed with a significance level $\alpha = 0.01$. We tested two hypotheses:

$$H_0 : \mu_1 = \mu_0$$
$$H_1 : \mu_1 > \mu_0$$

The p-value for testing these hypotheses is calculated as follow

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{(}n)}$$
$$p = 1 - F(t)$$

where s is sample standard deviation, n is the length of the sample, and F is the Fisher's cumulative distribution function. Figure 3.4 shows, that the t-test works well and it potentially can be utilized in change detection. The t-test detects a shift in mean in the right moment so it was decided to adjust change detection algorithms to follow a shift in mean values.

4 ALGORITHMS

All approaches and techniques used in this work are mostly described in two books and a few articles. The studies of change-point detection are very often based on statistical control theory. In statistical control theory there is a process, which is developing in time. The task of the statistical control theory is to find a moment, where the process is not considered evolving normally.

The book "Statistical Quality Control" written by D.Montgomery [5] explains and reviews modern techniques of statistical control. The book focuses on algorithms that often treat incoming data points one after another. This thesis, in contrast, mostly focuses on algorithms using log-likelihood ratio. The log-likelihood ratio is a statistic that is used to decide which of the two distributions is more likely. The book has a detailed description of the Shewhart Charts algorithm, CUSUM, and their families. However, some of the algorithms are not explained in detail in this section. For example, the CUSUM v-mask technique has many hyperparameters, which are not explained. However, not all the algorithms described in this book are applicable for this work because Montgomery presents algorithms that are intended to capture anomalies having prior information about the process, while the task of this work is detecting change-points.

M. Basseville and I. Nikiforov wrote the book "Detection of Abrupt Changes: Theory and Application" [6]. This book is focusing entirely on change-point detection techniques. The algorithms are explained in chapter 2 of the book. Almost all presented algorithms are adjusted for using LLR-technique. The parts of this book are online and offline algorithms. In this work, we are interested in online approaches. As mentioned above, a big part of considered algorithms have their roots in Statistical Control theory. Such algorithms are Shewhart Control Charts and CUSUM. Generally, the book has very good explanations, examples and derivations. The plots often illustrate the behavior of decision functions clearly and make an understanding of the algorithms easier. However, not all approaches are described clearly and some algorithms are described without many details. An example is the Bayesian type algorithm described in chapter 2.3, which has a parameter that was not explained. This algorithm would be potentially applicable to this work, but the parameter is left unexplained, making it difficult to use the algorithm's full potential. Generally, the book gives comprehensive studies about change-point techniques. Many algorithms implemented in this work are of the CUSUM family. These algorithms use

LLR-technique as their base.

In this thesis, the focus is on the change point detection algorithms derived from control chart algorithms. Both the Shewhart Control Charts algorithm and CUSUM-type algorithms are originally used in control theory. This chapter contains algorithms for detecting change points both separately for each channel and for all channels simultaneously. The Bayesian online change point detection algorithm uses another approach that differs from all other presented algorithms and does not originate from control theory.

4.1 Shewhart Control Charts

There are many change-detection techniques. The simplest algorithms use the log-likelihood ratio (LLR) approach. The LLR-approach is a method for comparing two distributions. The log-likelihood is defined by

$$L(y) = ln\frac{p_{\theta_1}(y)}{p_{\theta_0}(y)}, \tag{4.1}$$

where $p_{\theta_1}(y)$ is a distribution with some parameters θ_1 and $p_{\theta_0}(y)$ is a distribution with parameters θ_0. The θ can represent any parameter or a set of parameters of a distribution. In this work we focus on a shift in mean values and theta, thus, represents the mean of the normal distribution. The natural logarithm is a non-linear function, which accepts only positive values and returns both positive and negative values. The natural logarithm applied to likelihood ratio simplifies calculations by turning multiplications into summations. In case of the Gaussian distribution the natural logarithm allows also to avoid computation of the exponential terms.

Figure 4.1 shows an example of two normal distributions with parameters θ_0 and θ_1 representing mean. The distributions have the same standard deviations. Sequentially calculating LLR will result in a negative value if the distribution with mean θ_0 is more likely and a positive value if the distribution with mean θ_1 is more likely to be observed. Thus a change-point is detected when the sign of the LLR changes, because as stated in [6, p. 25]:

$$E[L(y)] > 0, \text{ When } p_{\theta_1} \text{ is more likely to observe}$$
$$E[L(y)] < 0, \text{ When } p_{\theta_0} \text{ is more likely to observe}$$

This approach is used in the Shewhart Control Charts algorithm. This algorithm requires us to know the parameters of the distributions before and after a change point, as shown in the example in Figure 4.1.

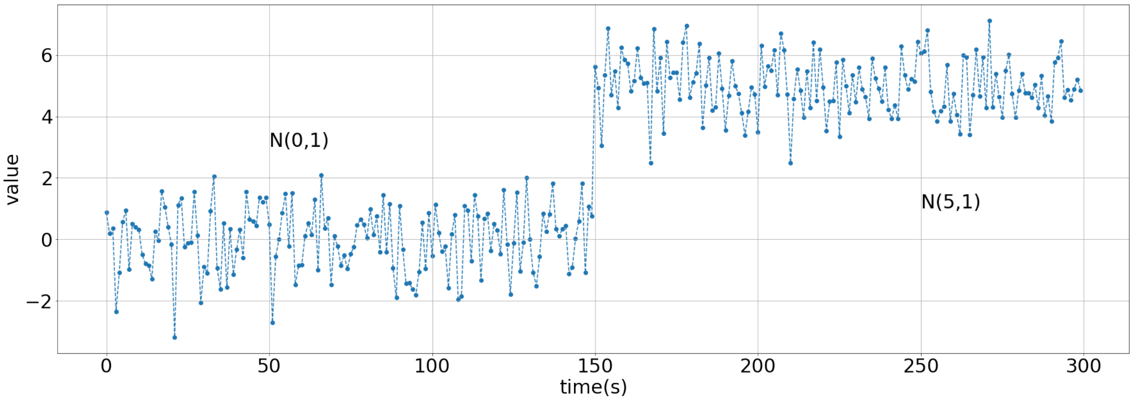


Figure 4.1. *Example with two mean-shifted normal distributions*

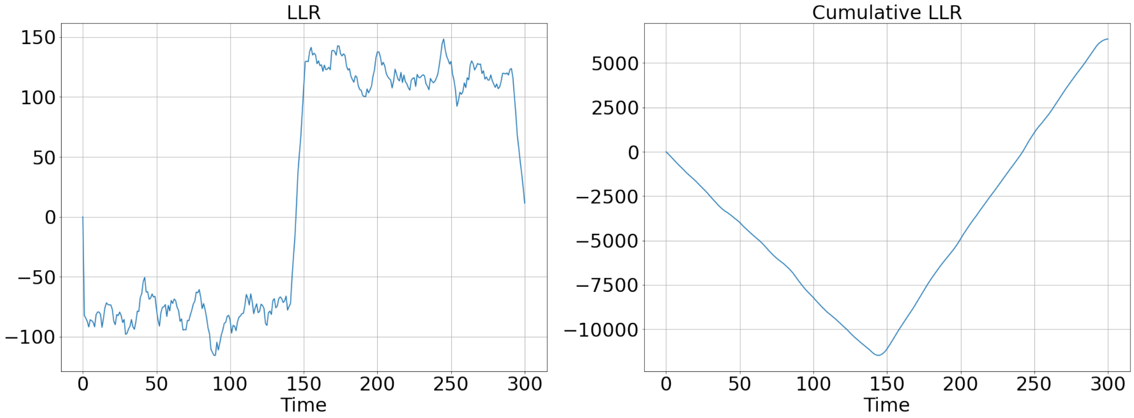


Figure 4.2. *Typical behavior of LLR-function*

The main idea of the Shewhart Control Charts algorithm is to test the following two hypotheses sequentially:

$$H_0 : \theta = \theta_0$$
$$H_1 : \theta = \theta_1$$

When a decision is made in favor of H_1 the algorithm stops running.

Let

$$L_j^k = \sum_{i=j}^{k} l_i \tag{4.2}$$
$$l_i = ln\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}$$

A decision is made when the sum of LLR crosses some threshold. Below this threshold, the decision is made in favor of H_0 otherwise the decision is made in favor of H_1.

$$d_k = \begin{cases} 0, L_j^k \leq h \\ 1, L_j^k > h \end{cases}$$

where h is the threshold, d_k is called *decision rule* and k is the sample number of size S (the size of the moving window). Consider an example from the book [6, p. 27], where the algorithm is derived for a known variance. All computations needed for defining LLR for two normal distributions are presented in the book. The changing parameter is here the mean and the standard deviation is known. The LLR is computed by

$$L_1^N = \frac{b}{\sigma}\sum_{i=1}^{N}(y_i - \mu_0 - \frac{v}{2}) \tag{4.3}$$

where

$$v = \mu_1 - \mu_0$$
$$b = \frac{v}{\sigma}$$

A sliding window of length S will be created and calculations will be performed on the samples of size S. A threshold value is also required. This can be done either manually, if data points have always approximately the same structure, or by using confidence intervals. Knowing distribution parameters before a change-point the confidence interval can be set:

$$|\bar{y}_i - \mu_0| \geq \kappa\frac{\sigma}{\sqrt{S}}$$

where κ is σ-distance from μ-value, which usually takes on values [1;5] and $\bar{y}_i$ is the sample mean of sub-time series. The confidence interval is related to the value of LLR as stated in [6, p. 28]. However, in this work the confidence interval method is not used because we use LLR values for decision making. The underlying distribution of LLR values is unknown. Although 4.3 assumes using cumulative LLR-values, in this work we found that it is more convenient to use its derivative or analyzing LLR-values separately.

The typical behavior (Figure 4.2) of non-cumulative LLR-function assumes that at some point its values become positive. After the change-point, the data is distributed approximately around 0. When a shift in mean happens, the LLR-function must become positive and cross 0.

In the case of the scent dataset, we assume that the data is distributed approximately normally especially after the change point. We usually don't know the parameters of distribution before a change point. However, after the change-point distribution of change between two consecutive measurements is expected to be around zero and variance small. Algorithm 1 shows the structure of the Shewhart Charts approach. The initial set of points of size S is taken and computed sample mean and sample standard deviation. It is assumed that the data before the change-point is distributed approximately with these parameters. A threshold h is a value, that can be chosen conveniently such that $h \geq 0 + m$. The m is an optional parameter, which can be set for example as half of the sliding window size. When the moving window rolls along the time series the change point is approaching from the right. The algorithm will probably capture the change point either in the end or in the middle of the moving window. This idea is behind adding the length or half of the length of the moving window.

Algorithm 1: Shewhart Charts

```
Input: s - size of sliding window, $x_i$ - initial sample of size s
Result: time step $i$ where distribution has changed
$\bar{\mu} = \frac{1}{s}\sum_{i=0}^{N} x_i$
$\bar{\sigma} = \frac{\sum(x_i-\bar{\mu})}{\sqrt{N-1}}$
$\mu_0$ = $\bar{\mu}$, $\sigma_0$ = $\bar{\sigma}$;
i = 0;  counter
do
    $x_i = [y_i : y_{i+s}]$; take sample of size s
    $v = 0 - \mu 0$;  $\mu_1$ is expected to be 0
    $b = \frac{v}{\sigma}$;
    $L_i = \frac{b}{\sigma}\sum_{i=1}^{N}(x_i - \mu_0 - \frac{v}{2})$;
    i = i + 1;
while $L_i$ < threshold;
```

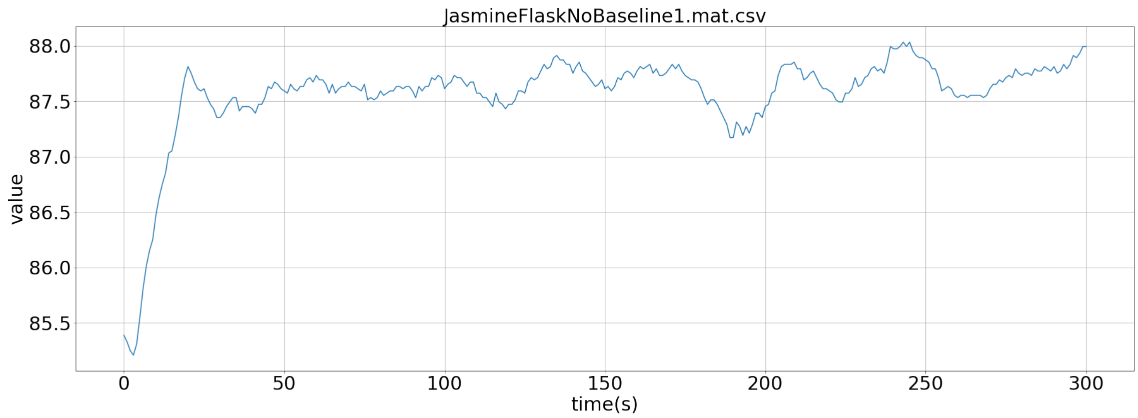


Figure 4.3. *Example readings from jasmine flask dataset (IMS_abs1)*

Figure 4.3 shows an example of readings of the ion-mobility spectrometer (IMS) sensor. This reading has a clear transient phase, which lasts for the first approximately 20 - 25 seconds and a stable phase afterwards. Before applying the Shewhart Control algorithm the data will be preprocessed by performing a difference operation. Difference operation normalizes data and removes trends and seasonality from a time-series. The difference operation is defined by

$$D(y_i, y_{i+1}) = y_{i+1} - y_i \tag{4.4}$$

Figure 4.4 shows the difference data of the time-series shown in Figure 4.3. From the figure it can be seen that there is a mean-shift and that the time-series is distributed approximately normally around zero after the change-point.

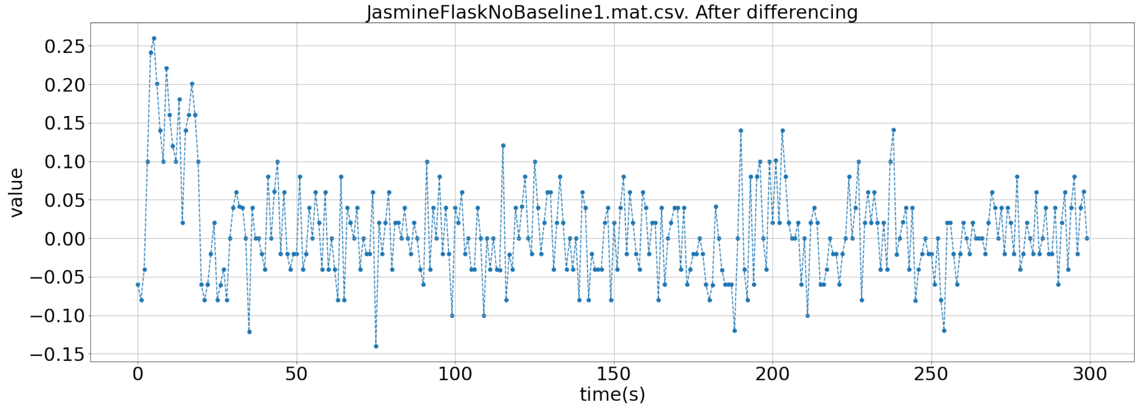


Figure 4.4. *Differencing the data in 4.3*

Now the Shewhart Charts algorithm can be applied to this data with parameters

$$\begin{aligned} S &= 10, \\ h &\geq 0 \end{aligned} \tag{4.5}$$

where S is the size of the sliding window and h is the threshold. Half of the window size was added to the found time step because it is assumed that the change occurred in the center of the window (see the explanation on page 14). The greater the size of the sliding window the smoother a decision function will be. Figure 4.5 demonstrates the results of applying the algorithm to the Jasmine time-series shown in Figure 4.3. The top plot shows readings of the time series and change point detected at 21 sec. The bottom

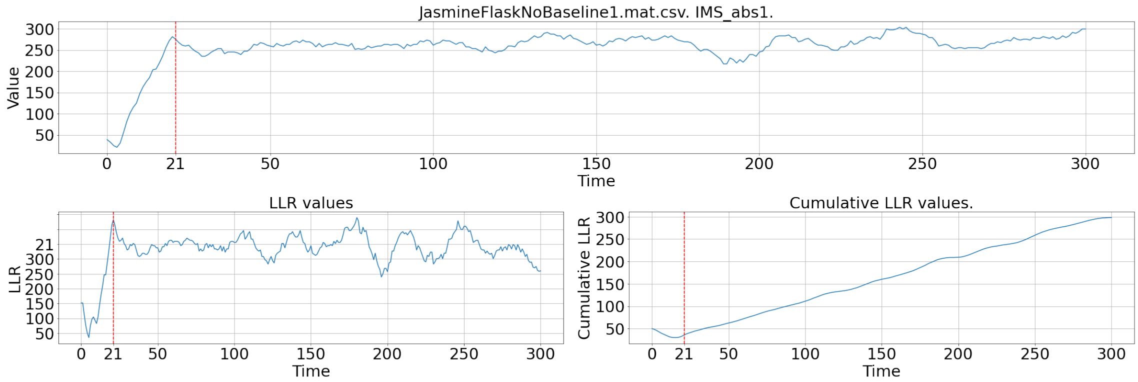


Figure 4.5. *Result of running Shewhart Charts-algorithm*

left plot shows the behavior of LLR values and the bottom right shows the behavior of cumulative LLR.

4.2 CUSUM

This algorithm was proposed by Page [8] in 1954. The CUSUM (CUmulative SUM) is the most popular and widely used algorithm for the detection of change points. It has many extensions: CUSUM V-mask, Self Starting CUSUM, Tabular CUSUM, etc. The classic CUSUM LLR-based algorithm will be described here.

The main idea of this algorithm uses the cumulative sum of LLR-values as in the previous section. We accumulate LLR values and compare them to the minimum over previous LLR values. The algorithm may be implemented with the preservation of the history of all previous values or only a minimum value before a particular step. We are only interested in the latter case, so storing full history is not necessary.

The LLR function was defined in 4.3. The CUSUM algorithm has an adaptive threshold, which is defined as

$$g_k = L_k - m_k \geq h, \tag{4.6}$$

where

$$m_k = \underset{1\leq j\leq k}{min} L_j.$$

It can be rewritten as

$$g_k = L_k \geq h + m_k, \tag{4.7}$$

where L_k is 4.3 of **kth** sample and h is conveniently chosen threshold [6, p. 27].

Thus the decision rule is defined as:

$$d_k = \begin{cases} 0, L_k < h + m_k \\ 1, L_k \geq h + m_k \end{cases} \tag{4.8}$$

Consider two groups of points generated with $N(0,1)$ and $N(5,1)$ Figure 4.6 demonstrates shift in mean from 0 to 5 at 150th second. The algorithm can be tested since

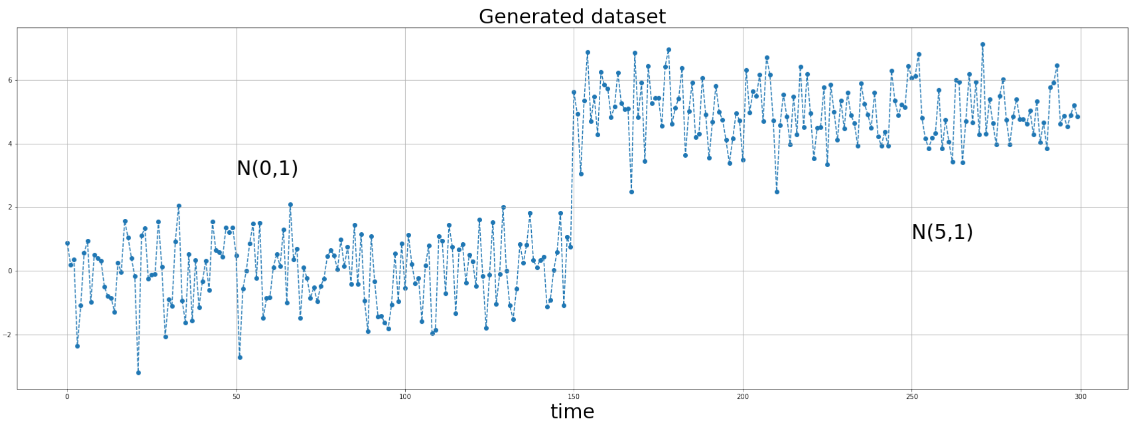


Figure 4.6. *Dataset generated with parameters N(0,1) and N(4,1)*

parameters of both distributions and the time of change are known. Let us set the threshold $h = 0$, $m_k = 0$, window size S = 5. The algorithm will be stopped when $d_k = 1$.

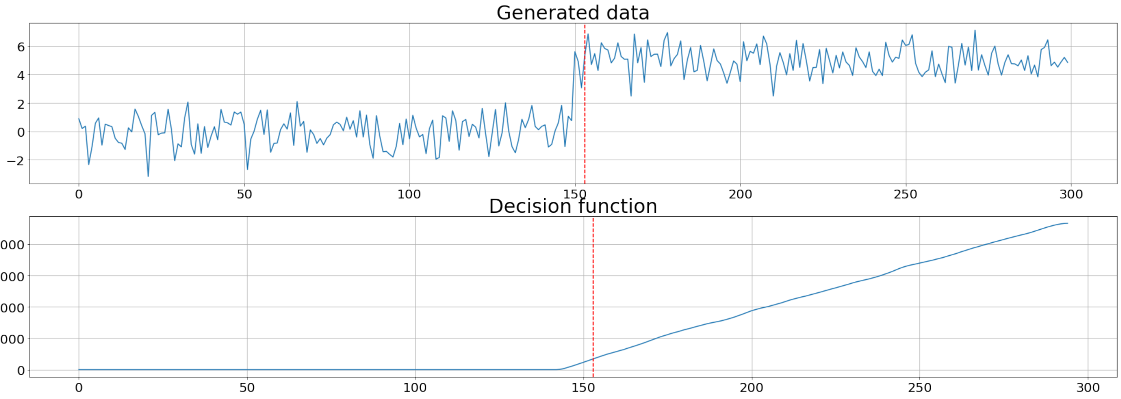


Figure 4.7. *CUSUM. Typical behavior of decision function*

Figure 4.7 demonstrates typical behavior of the decision function and the generated data set with detected change point. As can be seen, the decision function starts growing at 148th second. The CUSUM algorithm detected the change point precisely enough at 153th second. In order to improve the result, the size or half of the size of the moving window should be added to the found time step. Now let us consider the DIGITS dataset and test the CUSUM on it. Figure 4.8 demonstrates readings of ChemPro100i for the jasmine scent. The upper plot shows raw measurements and the bottom plot shows the same measurements differenced. The measurement has clear transient and stable phases. From these plots it can be seen that the changepoint is between 25 and 30 seconds. The differenced time series demonstrates more clearly where the mean has shifted and distribution has changed its parameters. The standard deviation for the first around 30 seconds is larger compared to the rest of the time series. Readings from the DIGITS dataset, which was analyzed in this thesis, often show that the standard deviation of the first 3 - 5 points in the time series is unstable. Therefore, it is a good idea to use

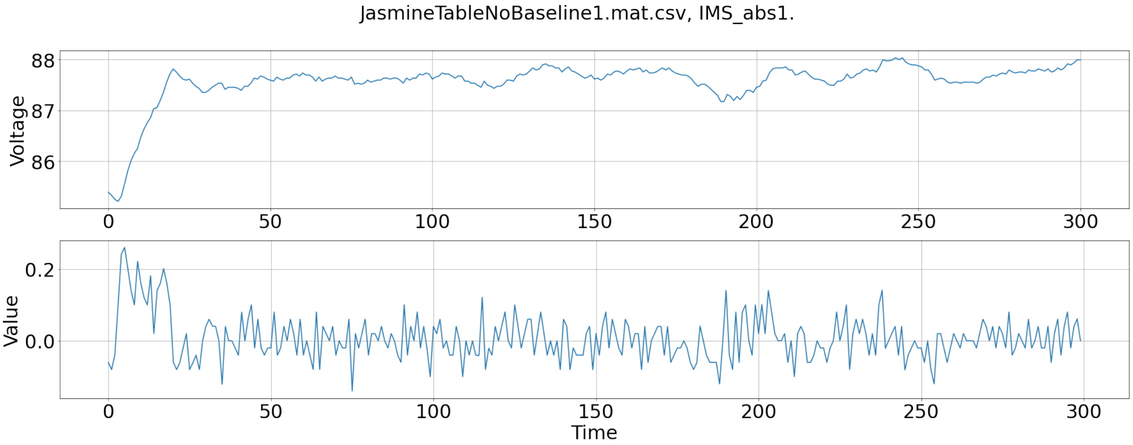


Figure 4.8. *Jasmine scent measured on the table. Absolute measurement 5.*

more than 3 points for calculating the sample mean and for the size of the sliding window. However, the transient phase is usually short, which forces us to use no more than 15 points.

```
**Algorithm 2:** CUSUM
**Input:** s - sliding window size, $x_i$ - sample of size $s$
**Result:** time step $i$ where distribution has changed
$L = 0$ # initialize cumulative sum variable
A = [] # initialize array for storing cumulative sums
$\bar{\mu}_0 = \frac{1}{s}\sum_{i=0}^{s} x_i$ # sample mean of the first s samples
$\bar{\sigma}_0 = \frac{\sum(x_i - \bar{\mu})}{\sqrt{s-1}}$ # sample standard deviation of the first S samples
detected = False
i = 0 # counter
while detected == False do
    i = i + 1;
    sample ← $[y_i : y_{i+s}]$; Get the next sample of data
    $v = 0 - \bar{\mu}_0$;
    $b = \frac{v}{\bar{\sigma}_0}$;
    $L = L + (1/\bar{\sigma}_0)(\sum sample - s \cdot \bar{\mu}_0 - \frac{s \cdot v}{2})$;
    $A[i] \leftarrow L$;
    if $(L - min(A)) > 0$ then
        $detected \leftarrow True$;
        $change\_point \leftarrow i$ ;
    end
end
```

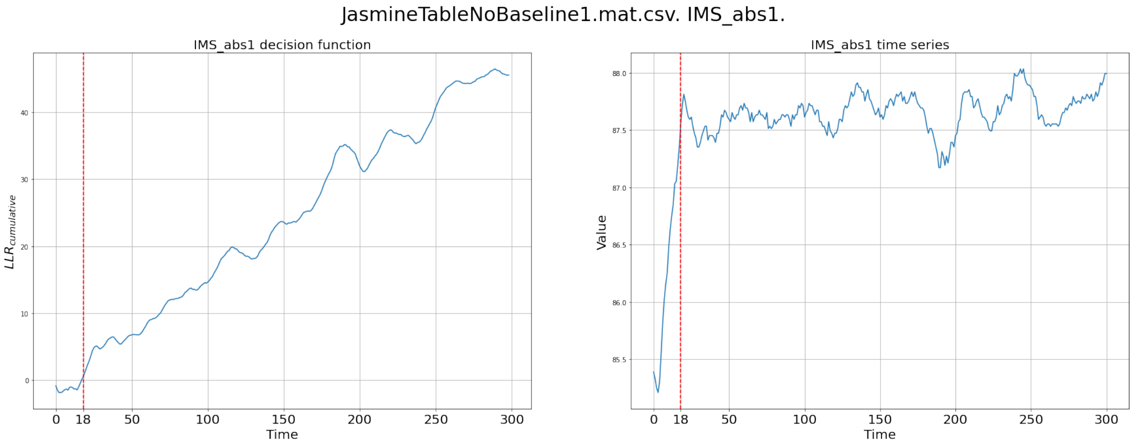


Figure 4.9. *Result of applying the CUSUM algorithm.*

Algorithm 2 shows the pseudo-code of the CUSUM approach. Figure 4.9 demonstrates the results of the algorithm. As we can see, the change point is located 18th second.

As shown in Algorithm 2, to make a decision it takes the minimum value from the whole preserved history of LLR. This makes the algorithm inefficient as it consumes memory. To overcome this problem the algorithm can be modified to store only the lowest LLR and update it at each iteration. Any value for h and m_k greater than or equal to zero can be chosen depending on empirical considerations. The limitation of the CUSUM is that it performs poorly when there are no clear transient and stable phases.

4.2.1 Tabular CUSUM

The CUSUM algorithm in control theory does not use LLR. The tabular CUSUM takes points one after another. This approach assumes that the shift in mean will be either positive or negative. Eventually, there is a need to apply the algorithm two times to the data for capturing shift or keep two variables for counting positive and negative cumulative values, because the sign of the shift has to be taken into account.

Montgomery proposes using each sample one after another [5, p. 404] and calculating cumulative sums as:

$$L_i^+ = max[0, x_i - (\mu_0 + K) + L_{i-1}^+]$$
$$L_i^- = max[0, (\mu_0 - K) - x_i + L_{i-1}^-]$$
$$K = \frac{|\mu_1 - \mu_0|}{2}$$

The x_i is the ith reading of the channel j, where $j \in [1; 14]$ for the IMS data analyzed in this thesis. The behavior of L_i^+ or L_i^- will be similar to Figure 4.7 depending on the trend of the time series. For example, if the time series is uptrending, then the L_i^+ will be similar to 4.7. Figure 4.10 shows readings of the jasmine scent from the first channel. The first two plots show raw and differenced readings. As can be seen on the differenced readings the mean has shifted positively. The bottom-left plot shows the decision function for the positive shift and the bottom-right plot shows the decision function for the negative shift. The bottom-left plot shows the typical behavior of the CUSUM algorithm and the bottom-right shows the peak near the change point. The decision about the change point is made with the expected behavior as on the bottom-left plot. Figure 4.11 show the same information for the readings from the third channel. Now the readings are uptrending and decision function for the negative shift demonstrates the behavior typical to the CUSUM's decision function (Figure 4.11, bottom-right plot) and the decision function for positive shift (bottom-left plot) does not contain any valuable information. Here the decision function for the negative shift has to be used.

Readings from ChemPro100i are often uptrending for electrodes measuring positive currents ("IMS_abs1" - "IMS_abs7") and downtrending for electrodes measuring negative currents ("IMS_abs9" - "IMS_abs15"). However, this is not always true. We do not know beforehand what trend the data will have.

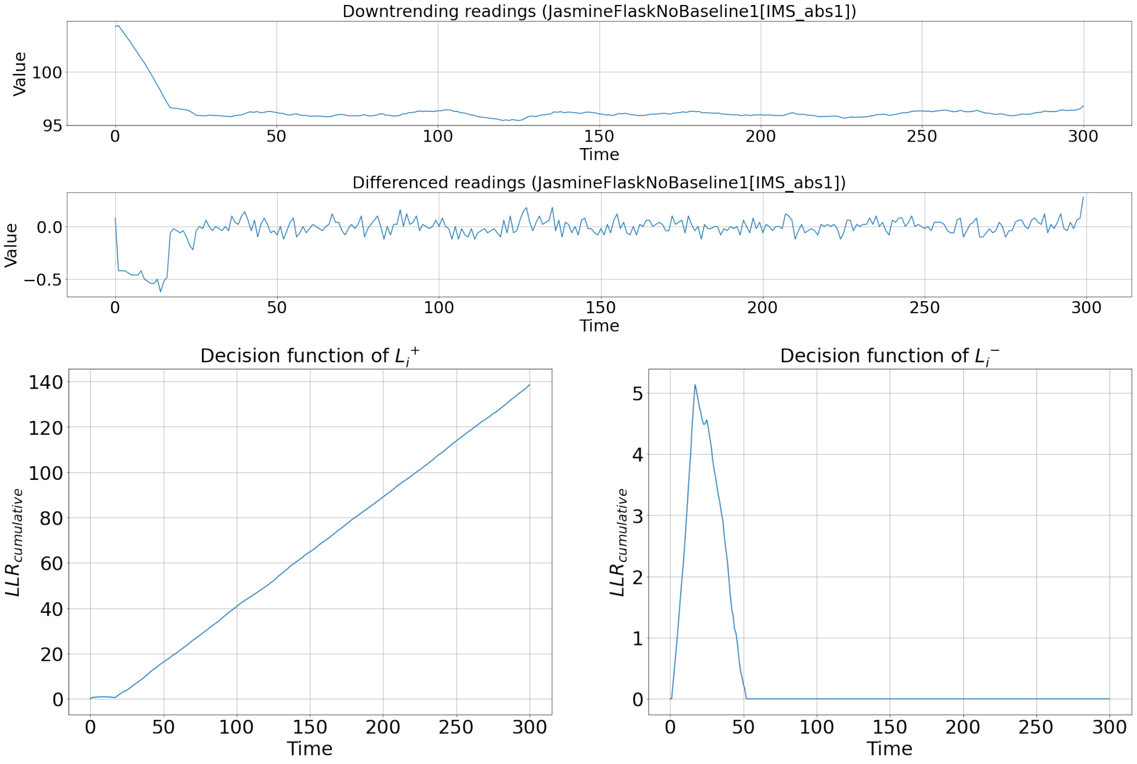


Figure 4.10. *Downtrending readings and decision functions. Jasmine IMS_abs1*

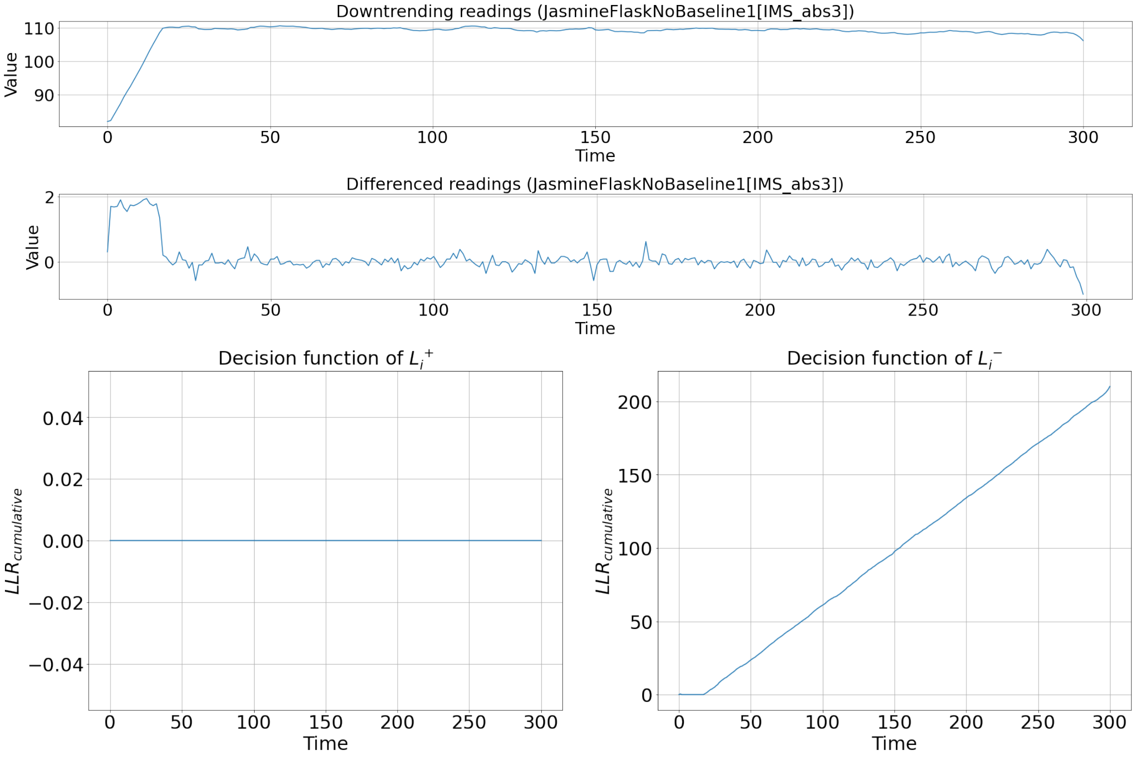


Figure 4.11. *Downtrending readings and decision functions. Jasmine IMS_abs3*

4.2.2 CUSUM V-mask

CUSUM V-mask uses both approaches: LLR and in turn samples. The idea of this algorithm is to apply the so-called "v-mask" as shown in Figure 4.12 The CUSUM V-mask

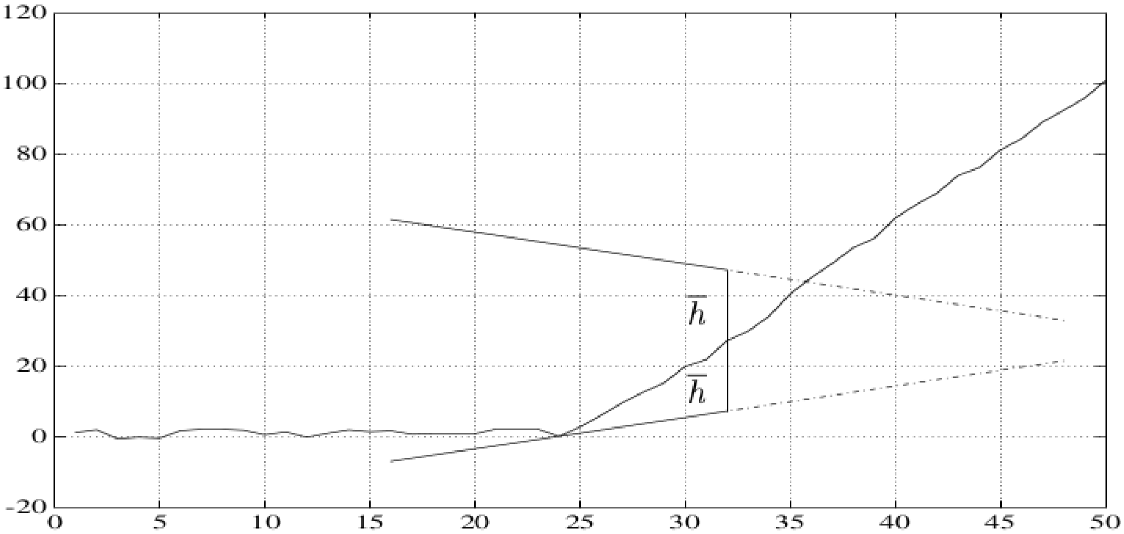


Figure 4.12. *CUSUM V-mask visualisation. Source [6, p. 43]*

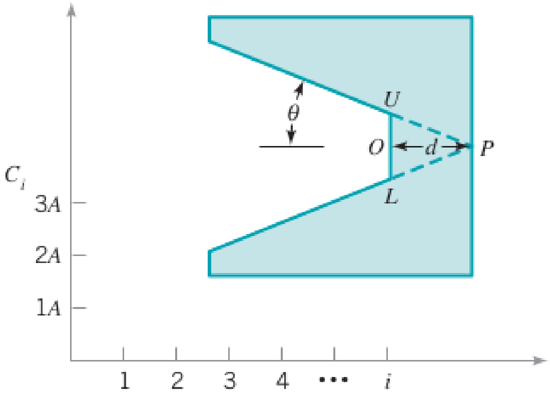


Figure 4.13. *V-mask structure. Source [5, p. 415]*

has many hyperparameters and there is no literature on how to set them for best performance. Instead, these parameters are chosen randomly. For example, Montgomery uses random values for calculating hyperparameters [5, p. 416].

The V-mask is set on the latest point O. The hyperparameters U and L define vertical distance below and above the point O. Parameter d defines the distance from the latest point O to the vertex of the v-mask and θ defines the angle of the opening backward arms. Basseville in his book [6, p. 41-43] does not explain how to choose these parameters. Montgomery in his book criticized this algorithm [5, p. 416]. He advises against using this approach because of the ambiguity of some parameters and difficulty of determining how far backward the so-called arms (Figure 4.13 the two slopes starting at the point P) should be extended.

4.2.3 Matrix Form CUSUM

For implementing the Matrix Form CUSUM the classic CUSUM was modified by the author of this thesis to process all channels simultaneously. The approach for calculating LLR and decision functions was kept unchanged. However, all the calculations were moved into vectors and matrices. This results in an approximately ten times faster computation of change detection points compared to the classic CUSUM algorithm and improved robustness to outliers.

Let

$$\mathbf{X}_i = \begin{bmatrix} x_{1,1} & ... & x_{1,S} \\ \vdots & \ddots & \vdots \\ x_{14,1} & ... & x_{14,S} \end{bmatrix} \in \mathbb{R}^{14\times S} \tag{4.9}$$

be a matrix containing a part of the 14-dimensional IMS time series data, where S (scalar) is the size of the sliding window and $x_{i,j}$ is the jth reading of the ith channel. Then the vector of sample means for each row will be:

$$\mathbf{m} = \begin{bmatrix} (1\times x_{1,1} + 1\times x_{1,2} + ... + 1\times x_{1,S})\frac{1}{S} \\ (1\times x_{2,1} + 1\times x_{2,2} + ... + 1\times x_{2,S})\frac{1}{S} \\ ... \\ (1\times x_{14,1} + 1\times x_{14,2} + ... + 1\times x_{14,S})\frac{1}{S} \end{bmatrix} = \mathbf{X_i} \times \begin{bmatrix} \frac{1}{S} \\ \vdots \\ \frac{1}{S} \end{bmatrix}_{S\times 1} = \begin{bmatrix} \bar{\mu}_1 \\ \bar{\mu}_2 \\ ... \\ \bar{\mu}_{14} \end{bmatrix} \tag{4.10}$$

The standard deviation for one dimension is calculated by:

$$\bar{\sigma}_i = \sqrt{\frac{\sum(x_i - \bar{\mu}_i)^2}{S-1}} \tag{4.11}$$

Let us now extend 4.11 to matrix form. Let

$$\mathbf{F}_\mu = \begin{bmatrix} \mathbf{m} & \mathbf{m} & ... & \mathbf{m} \end{bmatrix} \in \mathbb{R}^{14\times S}, \mathbf{1}_S = \begin{bmatrix} \frac{1}{S-1} \\ \vdots \\ \frac{1}{S-1} \end{bmatrix} \in \mathbb{R}^{14\times 1} \tag{4.12}$$

The value under the square root is calculated as follow

$$D = \begin{bmatrix} \frac{1}{S-1} \times (x_{1,1} - \bar{\mu}_1)^2 + \frac{1}{S-1} \times (x_{1,2} - \bar{\mu}_1)^2 + ... + \frac{1}{S-1} \times (x_{1,S} - \bar{\mu}_1)^2 \\ \vdots \\ \frac{1}{S-1} \times (x_{14,1} - \bar{\mu}_{14})^2 + \frac{1}{S-1} \times (x_{14,2} - \bar{\mu}_{14})^2 + ... + \frac{1}{S-1} \times (x_{14,S} - \bar{\mu}_{14})^2 \end{bmatrix} =$$

$$= \begin{bmatrix} (x_{1,1} - \bar{\mu}_1)^2 & (x_{1,2} - \bar{\mu}_1)^2 & ... & (x_{1,S} - \bar{\mu}_1)^2 \\ \vdots & \vdots & \vdots & \vdots \\ (x_{14,1} - \bar{\mu}_{14})^2 & (x_{14,2} - \bar{\mu}_{14})^2 & ... & (x_{14,S} - \bar{\mu}_{14})^2 \end{bmatrix} \begin{bmatrix} \frac{1}{S-1} \\ \vdots \\ \frac{1}{S-1} \end{bmatrix} \tag{4.13}$$

Then 4.13 can be rewritten in more compact notation as

$$(\mathbf{X_i} - \mathbf{F}_\mu)^2 \times \mathbf{1}_S \tag{4.14}$$

Note that square and square root in 4.15 are applied to each entry of the matrix. Now the vector of standard deviations for all rows of $\mathbf{X_i}$ is calculated by

$$\mathbf{STD} = \sqrt{(\mathbf{X_i} - \mathbf{F}_\mu)^2 \times \mathbf{1}_S} \tag{4.15}$$

and the vector of sample means for each row is

$$\mathbf{m} = \mathbf{X_i} \times \mathbf{1}_s. \tag{4.16}$$

The LLR 4.3 can be rewritten as

$$L_1^N = (\frac{b}{\sigma}) \sum_{i=1}^{S} (y_i - \mu_0 - \frac{v}{2}) = (\frac{b}{\sigma})(\sum_{i=1}^{S} y_i - S\mu_0 - \frac{Sv}{2}), \tag{4.17}$$

where

$$\sigma = \mathbf{STD}$$

$$\mathbf{v} = -1 \times \mathbf{m} \tag{4.18}$$

$$\mathbf{b} = \mathbf{v} \odot \frac{1}{\mathbf{STD}} \tag{4.19}$$

$$\sum_{i=1}^{S} y_i = \mathbf{X_i} \times \begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix}^T \tag{4.20}$$

$$S\mu_0 = S \times \mathbf{m} \tag{4.21}$$

Note that S is scalar and L_1^N is the LLR calculated for the first sample of size N. Symbol $\odot$ in 4.19 denotes element-wise multiplication.

The decision rule 4.7 is modified

$$g_k = average(L_k) \geq h + m_k, \tag{4.22}$$

where

$$L_k = \sum_{i=1}^{k-1} L_i^{(S)}, k \in [1 ... \infty), L_k \in \mathbb{R}^{14}. \tag{4.23}$$

The summation in 4.23 runs from 1 to infinity until a change point is detected. Sequential calculation of 4.17 yields LLR values for all channels simultaneously. In the one-dimensional version of the CUSUM, the history of calculated cumulative LLR for finding minimum was kept. If the history was kept in this algorithm as well a matrix need to be created, where each row contains the cumulative LLR history for one channel. The alternative way to implement it is to create a vector, which will keep only the minimum values for each channel. For example, in Algorithm 3 this vector is created (line 7) and updated (lines 14,15).

Algorithm 3: Matrix Form CUSUM

```
   Input: S - sliding window size, $X_i$ - sample of size $14 \times S$
   Result: time step $i$ where distributions have changed
 1 $\mathbf{L_i} \leftarrow \begin{bmatrix}0 & 0 & ... & 0\end{bmatrix}^T$ # initialize vector of cumulative LLR
 2 $\mathbf{M} = \mathbf{X_i} \times \mathbf{1}_S$ # calculate vector of means
 3 $\mathbf{F}_\mu \leftarrow \begin{bmatrix}\mathbf{M} & \mathbf{M} & ... & \mathbf{M}\end{bmatrix}$ # populate M into the matrix
 4 $\mathbf{STD} = \sqrt{(\mathbf{X_i} - \mathbf{F}_\mu)^2 \times \mathbf{1}_S}$ # calculate vector of standard deviations
 5 $\mathbf{v} = -1 \times \mathbf{M}$
 6 $\mathbf{b} = \mathbf{v} \odot \frac{1}{\mathbf{STD}}$
 7 $\mathbf{I} \leftarrow \begin{bmatrix}0 & 0 & ... & 0\end{bmatrix}^T$ # vector for keeping minimum values of LLR
 8 detected = False
 9 point = 0
10 i = 0 # loop counter
11 while detected == False do
12     $\mathbf{X_i} \leftarrow$ new chunk of data of size $14 \times S$
13     $\mathbf{L_i} = \mathbf{L_i} + (\frac{\mathbf{b}}{\mathbf{STD}})(\mathbf{X_i} \times \begin{bmatrix}1 & 1 & ... & 1\end{bmatrix}^T - S \times \mathbf{M} - \frac{S}{2} \times \mathbf{v})$
14     if $\mathbf{L_i} < \mathbf{I}$ then
15         $\mathbf{I} \leftarrow \mathbf{L_i}$
16     end
17     if $average(\mathbf{L_i} - \mathbf{I}) > 0$ then
18         point = i + $S$
19         $detected \leftarrow True$
20     end
21     i = i + 1
22 end
```

Lines 14 - 15 of Algorithm 3 show how each entry of two vectors is compared and only the entries satisfying the condition are updated. Adding the length of the sliding window improves the performance of this algorithm (Algorithm 3 line 18). This was tested on several scents of the DIGITS dataset.

Figure 4.14 shows readings of the ChemPro100i on the left and decision functions on the right. As can be seen from the figure readings "IMS_abs9" and "IMS_abs10" are either failing or they have a more smooth and longer transition phase. Regardless of these two readings, the algorithm detects the right change point that happened at the 18th second.

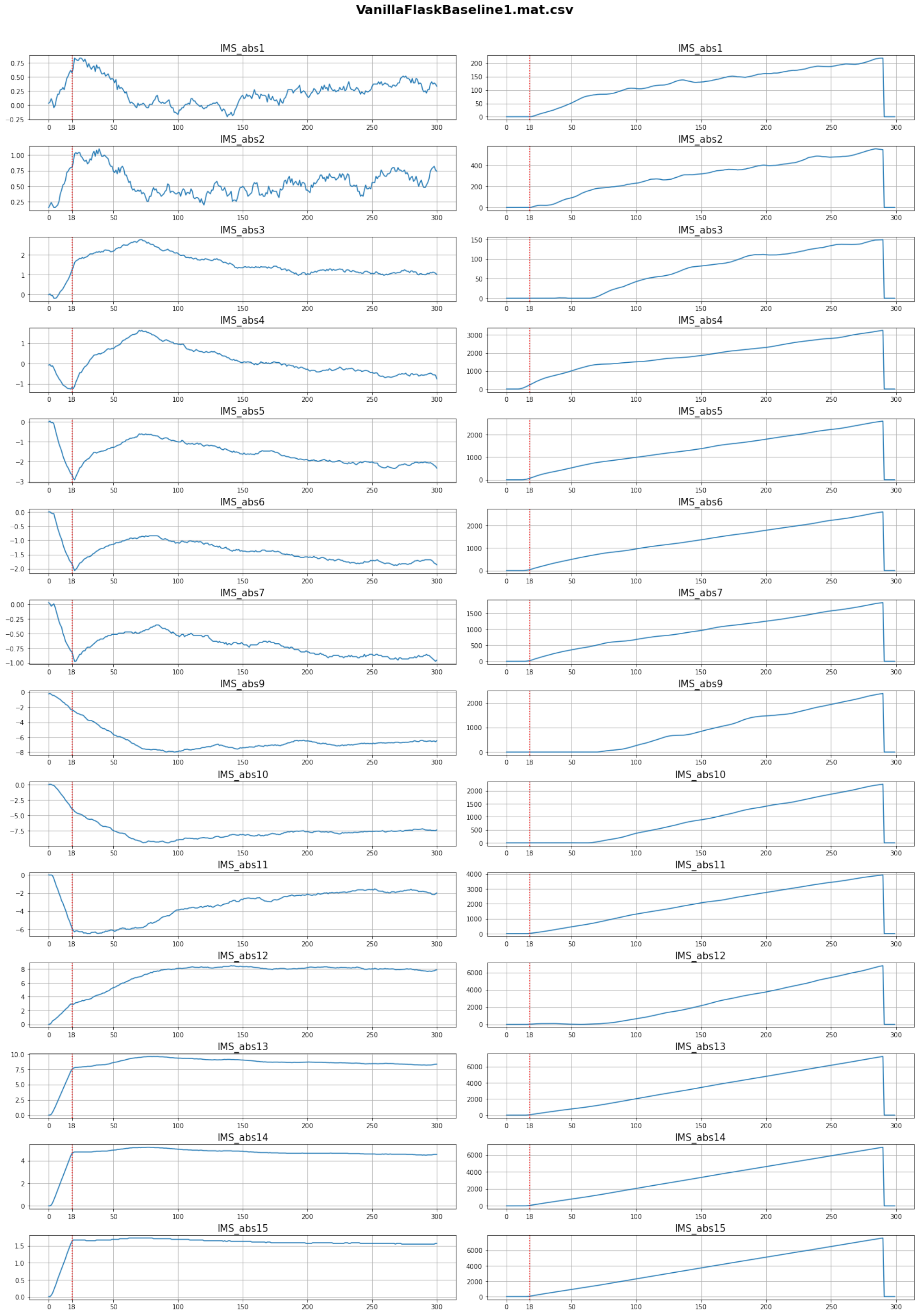


Figure 4.14. *Results of running Matrix Form CUSUM on the Vanilla Flask readings.*

4.2.4 Multivariate Max-CUSUM Chart

This algorithm was proposed in [9]. The Multivariate Max-CUSUM Chart (Max-MCUSUM) is a naturally multivariate algorithm, which uses multivariate Gaussian distribution. The idea of the Max-MCUSUM is to reduce a multivariate normal process to a univariate process.

In this thesis, the algorithm was modified by adding a smoothing operation. The algorithm assumes, that the data points come one after another. In this case, a sliding window of size S was created. Eventually, the data will come as a matrix of size $\mathbb{R}^{14 \times N}$. All derivations of the formulas below are presented in the original paper.

Let

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{14} \end{bmatrix} \tag{4.24}$$

be the vector of mean values of all channels and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,14} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,14} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{14,1} & \sigma_{14,2} & \cdots & \sigma_{14,14} \end{bmatrix} \tag{4.25}$$

be the covariance matrix calculated on the initial data sample. The mean-vector before the change point is denoted as $\boldsymbol{\mu_0}$ and the mean-vector after the change point is denoted $\boldsymbol{\mu_1}$. As in the case of the classic CUSUM algorithm, the $\boldsymbol{\mu_0}$ and the Σ are calculated from the initial sample of data points. It is assumed, that they are Gaussian distributed with these parameters. The distribution after the change point is assumed to be Gaussian with the mean approximately zero and covariance matrix approximately Σ.

The noncentrality parameter is defined as

$$z = \sqrt{(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{\mu_1} - \boldsymbol{\mu_0})} \tag{4.26}$$

and the CUSUM decision rule is defined as

$$L_i = max(0, L_{i-1} + a(\boldsymbol{x_i} - \boldsymbol{\mu_0}) - 0.5z) > h, \tag{4.27}$$

where

$$a = \frac{(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T \Sigma^{-1}}{\sqrt{(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T \Sigma^{-1} (\boldsymbol{\mu_1} - \boldsymbol{\mu_0})}} \tag{4.28}$$

and h is a conveniently chosen threshold value. In the original paper $\boldsymbol{x_i}$ in 4.27 denotes an m-dimensional vector of data points $\mathbb{R}^m$. In this implementation, a matrix $\mathbf{M} \in \mathbb{R}^{14\times N}$ is collected and the mean-value is calculated along each row (Algorithm 4, row 11).

Algorithm 4: Multivariate Max-CUSUM Chart

Input: S - sliding window size, X_i - sample of size $14 \times S$

Result: time step i where distributions have changed

```
1  $\mu_0 = \bar{\mu}$ # sample mean along each row
2  $\Sigma = \bar{\Sigma}$ # sample covariance matrix
3  $\Sigma_{i,j} = \Sigma_{i,j} + 1e - 10,$ where i = j # add a small value to the main diagonal
     to avoid singularity
4  a = $\frac{(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})^T\Sigma^{-1}}{\sqrt{(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})}}$
5  $L_i = 0$ # initial value for cumulative sum
6  detected = False
7  point = 0
8  i = 0 # counter
9  **while** *detected == False* **do**
10     $\boldsymbol{X_i}$ = *data points of size* $14 \times N$
11     $\boldsymbol{\mu_1}$ = mean($X_i$)
12     $\boldsymbol{D} = \sqrt{(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})^T\Sigma^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})}$
13     $L_i = max(0, L_{i-1} + a(\boldsymbol{X_i} - \boldsymbol{\mu_0}) - 0.5 \times D$
14     **if** $S_i > 0$ **then**
15         point = i + N
16         detected = True
17     **end**
18     i = i + 1
19 **end**
```

As can be seen, the Max-MCUSUM algorithm is very similar in its structure to the classic CUSUM algorithm. Calculation on line 13 (Algorithm 4) yields a one-dimensional value L_i. The covariance matrix can be singular if variances on the main diagonal are close to zero. This problem is avoided by adding a small value to the entries in the main diagonal.

Figure (4.15) shows the behavior of the decision function applied to the JasmineFlaskNoBaseline1 data set. As can be seen, the algorithm detects accurately the change point at the twentieth second. The algorithm detects the change point for all the channels simultaneously.

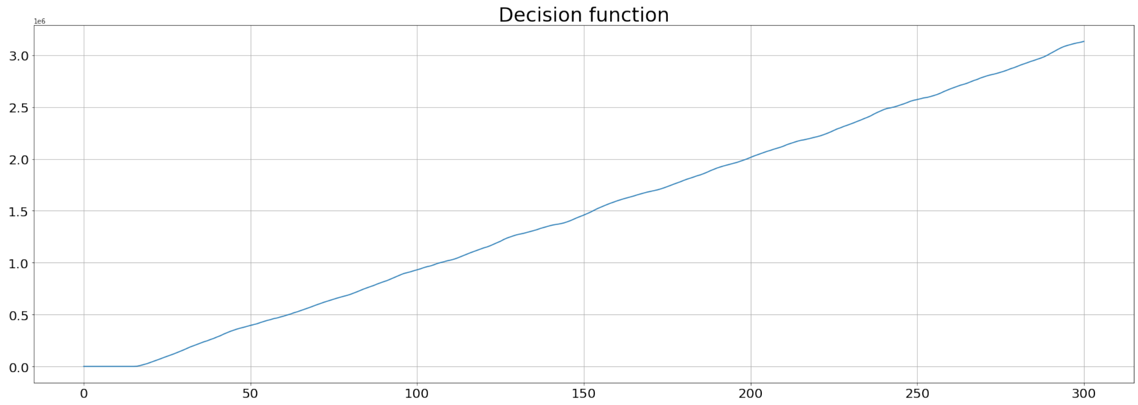


Figure 4.15. *Decision function for JasmineFlaskNoBaseline1 data set.*

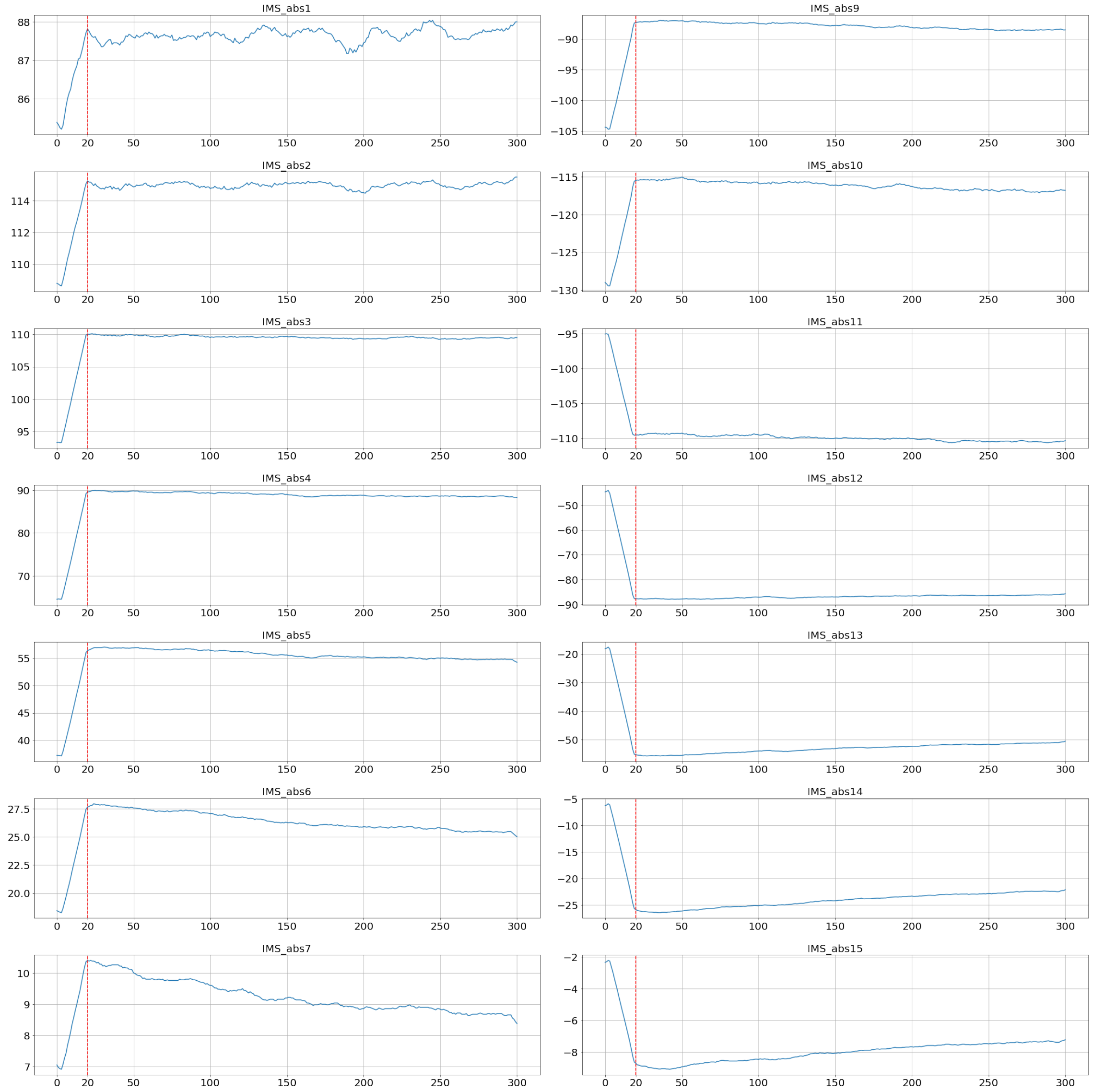


Figure 4.16. *JasmineFlaskNoBaseline1. Result of running the algorithm.*

4.3 Bayesian Online Change Point Detection

The Bayesian Online Change Point Detection algorithm was proposed by Adams and MacKay in 2007 [10]. It is not related to the log-likelihood ratio statistics. The change point is detected in terms of so-called run lengths.

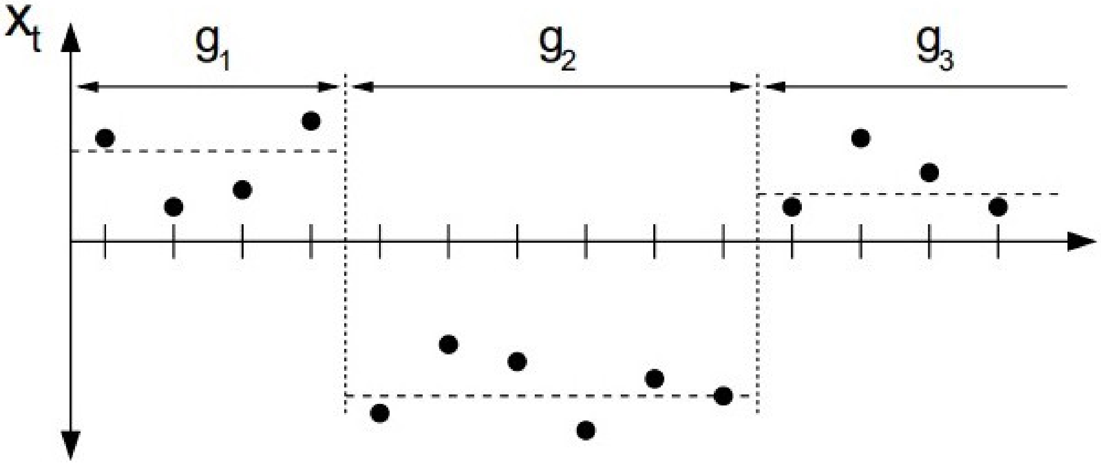


Figure 4.17. *Partitioning the data by run lengths. Source [10]*

Figure 4.17 represents the data arriving in time. As can be seen, the mean got changed after the 4th time step. The next mean got changed after the 10th time step. The g_i is a partition of the data, which share the same mean. These partitions are separated by change points.

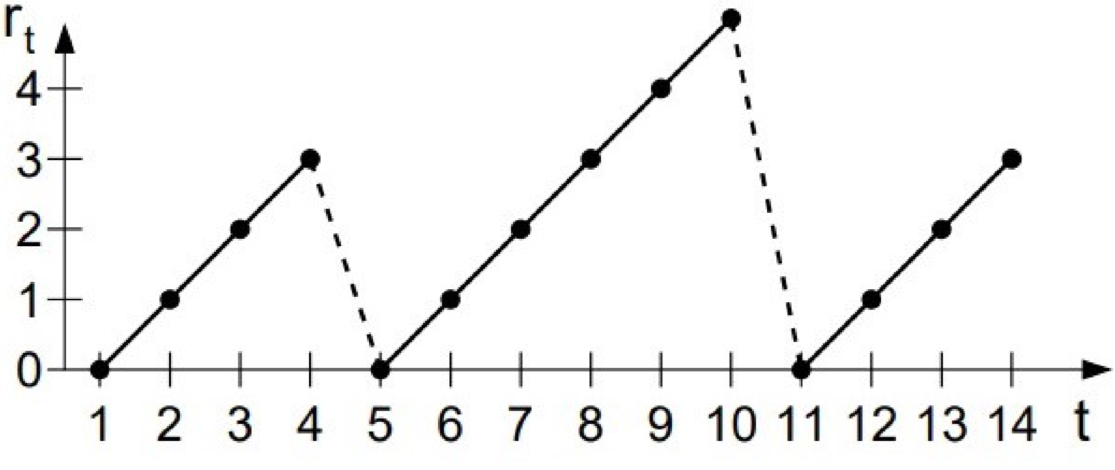


Figure 4.18. *Run lengths. Source [10].*

Figure 4.18 demonstrates a plot of run lengths. As we can see at time steps 5 and 11 run lengths drop to zero. Taking into account the previously observed data, the run length increases by one if a new datapoint probably belongs to the same partition. Otherwise, the run-length will drop to zero. These drops indicate a change in distribution.

The algorithm enables us to add a priori information into the system. For example, in the DIGITS dataset change points occur usually within the first 100 timesteps. We can enter this information using the so-called "Hazard function" or "hazard rate", which in our case may be a constant function. The hazard rate is a priori information about how often change points can occur. More information about the hazard function may be found in [11, p. 9-48]. Information about the underlying probability distribution function can be added as well. Here the Gaussian and the t-distribution will be used as underlying probability models.

Each node in 4.19 keeps information needed for further calculations: the predicted probability of the new point, updated parameters of the underlying probability model, hazard function, probability of growth. The original paper [10] describes 4.19 as follow: *"Figure shows the trellis on which the message-passing algorithm lives. Solid lines indicate that probability mass is being passed "upwards," causing the run length to grow at the next time step. Dotted lines indicate the possibility that the current run is truncated and the run length drops to zero.*"

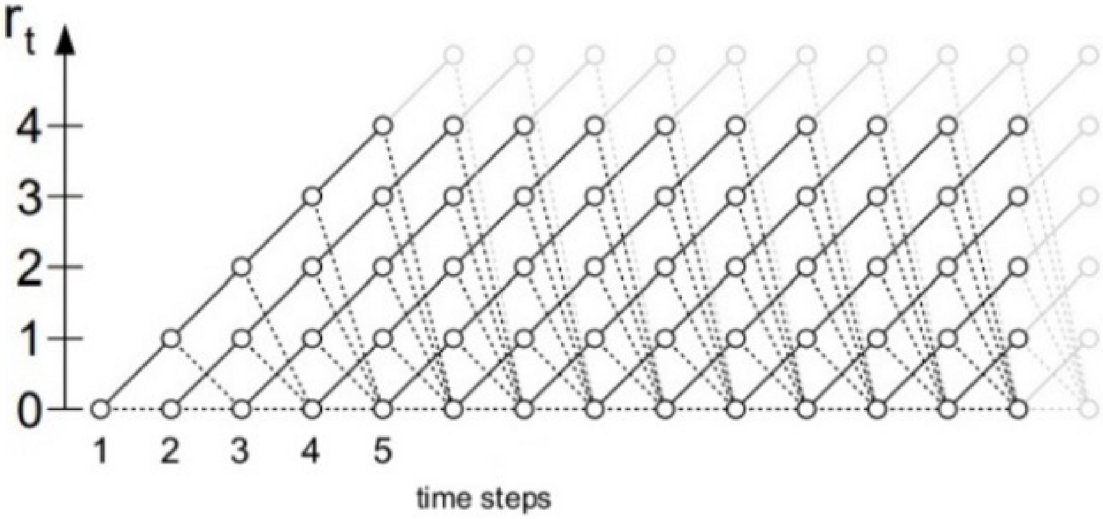


Figure 4.19. *Message passing trellis. Source [10].*

For example, at time step 4 there are 4 nodes along r_t axis. Each of these nodes contains the probability of the 4th point appearance at each run length. This probability is calculated using a posterior predictive distribution function with parameters from each node of the previous time step.

First, this algorithm will be considered using data generated from Gaussian distribution with known mean before and after the shift. In this case, the Gaussian distribution will be used as the underlying probability model. It is assumed that the variance is known beforehand.

Many authors (e.g. [12, p. 12]) define the hazard function as discrete Geometric distribution with parameter λ because it is computationally efficient and easy to interpret. The Hazard function is then defined as

$$H(t) = \frac{1}{\lambda}, \forall t \tag{4.29}$$

where t is time and λ is called *precision* and often set $\lambda = \frac{1}{\sigma^2}$.

4.3.1 Parameter μ is unknown and σ is known

Only the calculations will be presented here with short descriptions of derivations if needed. The more detailed derivations are explained in [10] and [12, p. 11-15]. In this algorithm, a shift in the mean is detected while the variance is constant.

The first step is to calculate the posterior predictive probability of the arrived point for each possible run length. The posterior predictive is calculated by

$$\pi_i = N(x_i|\mu_i, \sigma^2 + \frac{1}{\lambda}) \tag{4.30}$$

where x_i is a new point from a time series and

$$\mu_i = \frac{\frac{\mu_{prior}}{\lambda} + \frac{\sum x_i}{\lambda_{prior}}}{\frac{i}{\lambda_{prior}} + \frac{1}{\lambda}} \tag{4.31}$$

$$\lambda_i = \lambda_{prior} + i\lambda. \tag{4.32}$$

This step in the algorithm is realized by defining probability distribution functions for each μ and λ. In equations 4.31 and 4.32 λ_{prior} represents our prior knowledge about the precision and λ_i is the value of the λ updated on each step. The prior will be set to $\lambda_{prior} = 1$.

The second step is to calculate growth probabilities for each possible run-length $r > 0$ of the current time step given run lengths on the previous steps by

$$P(r_t = l, x_{1:t}) = P(r_{t-1}, x_{1:t-1})\pi^l_{t-1}(1 - H(r_{t-1})), \tag{4.33}$$

where $H(r_{t-1})$ is the hazard rate from previous steps, which represents our prior beliefs about change point, and l is the possible run length at the current time step.

The third step is to calculate change point probabilities as

$$P(r_t = 0, x_{1:t}) = \sum P(r_{t-1}, x_{1:t-1})\pi^l_{t-1}H(r_{t-1}) \tag{4.34}$$

Summation in 4.34 means summing up all probabilities at the previous steps.

The fourth step is normalizing probabilities calculated at two previous steps:

$$P(r=l) = \frac{P(r=l)}{\sum P(r=l)} \tag{4.35}$$

In the last step, the parameters are updated using information learned in this current step. Updates are calculated using 4.31 and 4.32.

Let us consider an example for which data was generated from two normal distributions with the change point at the 50th second (see Figure 4.20). The exact parameters of the two distributions are known and the algorithm can be initialized. The first S samples are taken for calculating sample mean $\bar{x}$. Let:

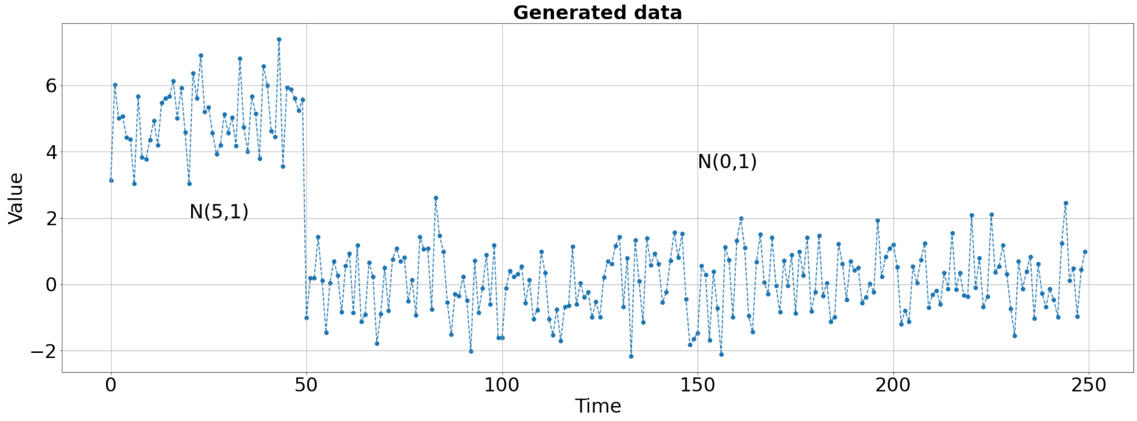


Figure 4.20. *Generated data for testing the algorithm.*

$$\mu_{prior} = \bar{\mu}_{1:S} \tag{4.36}$$

$$\sigma^2 = 1 \tag{4.37}$$

$$\lambda = \frac{1}{\sigma^2} = 1 \tag{4.38}$$

$$H(t) = \frac{1}{100} \tag{4.39}$$

$$R[0,0] = 1 \tag{4.40}$$

The hazard rate is set to $1/100$ to show, that the algorithm is insensitive to this value. The right value for the hazard rate must represent prior beliefs about how often change points happen.

Two vectors will be created, which contain all mean values and all λ for all run lengths at each time step. These two vectors are needed for further calculations.

A matrix defined by 4.21 will be created to represent structure 4.19. The first column of the matrix R represents probabilities of change point or that the run length drops to zero. All other columns represent growth probabilities. The matrix R exists here only for visualization purposes. It is not mandatory to keep all the joints in memory. Only the joint

$$R = \begin{array}{c|ccccc} & r_0 & r_1 & r_2 & \dots & r_n \\ \hline t_1 & P(r_t = 0) & 0 & 0 & \dots & 0 \\ t_2 & P(r_t = 0) & P(r_t = 1) & 0 & \dots & 0 \\ t_3 & P(r_t = 0) & P(r_t = 1) & P(r_t = 2) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ t_n & P(r_t = 0) & P(r_t = 1) & P(r_t = 2) & \dots & P(r_t = S) \end{array}$$

Figure 4.21. *Matrix R for storing probabilities of growth*

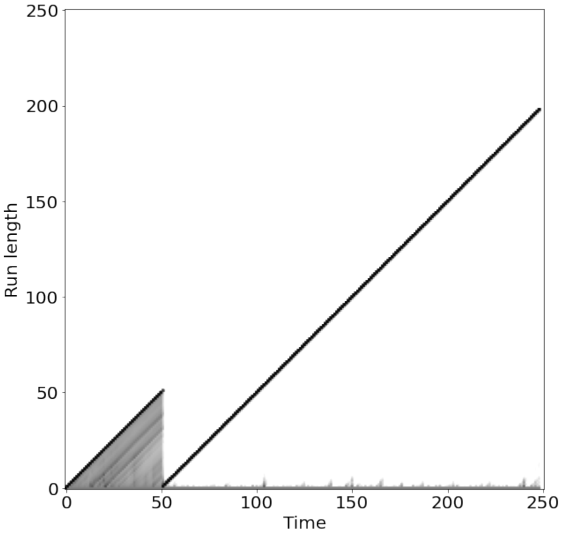


Figure 4.22. *Typical behavior of the decision function. BOCPD.*

from the previous iteration is needed.

The data will be fed into the algorithm one point after another. At each time step, a row will be added to R for the new time step.

Let us now observe one new point at $t = 2$ and calculate posterior predictive distribution π_t of observing this point at each possible run length with 4.30. The probabilities of growth are computed using posterior predictive results with

$$P(r = i) = R[t-1, :] * \pi_t * (1 - H(t)). \tag{4.41}$$

The probability of $t = 2$ being a change point is

$$P(r = 0) = \sum R[t-1, :] * \pi_t * H(t), \tag{4.42}$$

Vectors 4.41 and 4.42 are concatenated into one vector

$$V = [P(r = 0), P(r = i)]. \tag{4.43}$$

The probabilities in this vector are renormalized by

$$R[i,:] = \frac{V}{\sum v_i} \tag{4.44}$$

as a new row into the matrix R. The last step is updating μ and λ using 4.31 and 4.32.

Figure 4.22 shows the typical behavior of matrix R for the data in Figure 4.20 in grey shade and rotated by 90 degrees for readability. The darker the colour the more probability mass is concentrated at a point. In the figure it can be seen, that until the 50th second probability mass is mostly concentrated at the top designating probability of growth. At the 50th second probability of increasing run length drops to zero, which means that the change point has occured almost certainly.

This configuration of BOCPD does not perform on the scent data, because the exact parameters of probability distributions are unknown beforehand. To overcome this problem the BOCPD can be reconfigured to use the t-distribution as an underlying probability model.

4.3.2 Both μ and σ are unknown

The BOCPD framework has 4 main steps:

1. Define posterior predictive distribution
2. Calculate probability of growth
3. Calculate probabitliy of change point
4. Update parameters

In case μ and σ are unknown only steps 1 and 4 must be modified. Previously it was assumed that μ is unknown and σ is known, which required using Normal distribution as a conjugate prior. The first step is modified to use another conjugate prior for estimating μ and σ. The conjugate prior used in this case is a Normal-Gamma as proposed in [12, p. 17]. The posterior predictive distribution, in this case, will be a generalized t-distribution with the following parameters:

$$v = 2\alpha \textit{ degrees of freedom} \tag{4.45}$$

$$\mu = \bar{\mu} \textit{ sample mean} \tag{4.46}$$

$$\sigma^2 = \frac{\beta_n(\kappa_n + 1)}{\alpha_n \kappa_n} \tag{4.47}$$

The Normal-Gamma probability distribution function has four parameters: κ, μ, α and β.

The updating rules for these parameters are

$$\bar{\mu}_n = \frac{\kappa_0\mu_0 + n\bar{x}}{\kappa_0 + n} \tag{4.48}$$

$$\kappa_n = \kappa_0 + n \tag{4.49}$$

$$\alpha_n = \alpha_0 + \frac{n}{2} \tag{4.50}$$

$$\beta_n = \beta_0 + \frac{\kappa_i(x_i - \mu_i)^2}{2(\kappa_i + n)} \tag{4.51}$$

where $n = 1$ since the data arrives one point at a time.

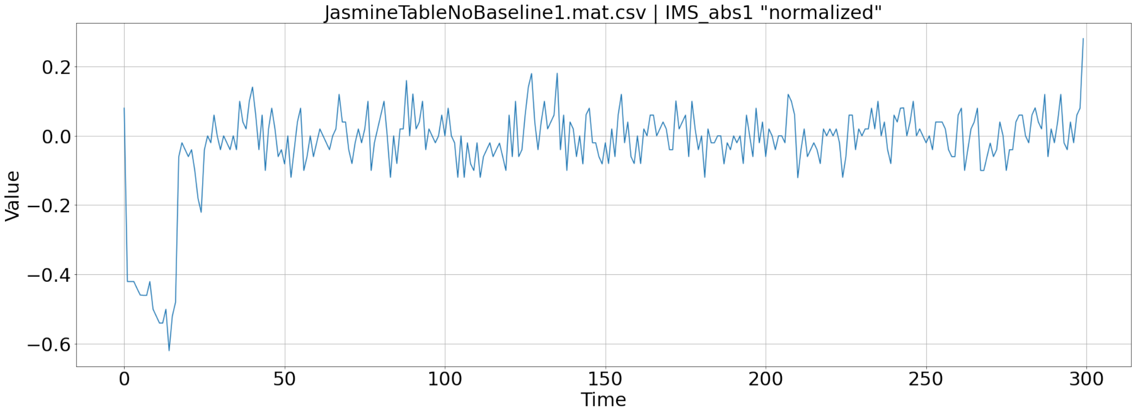
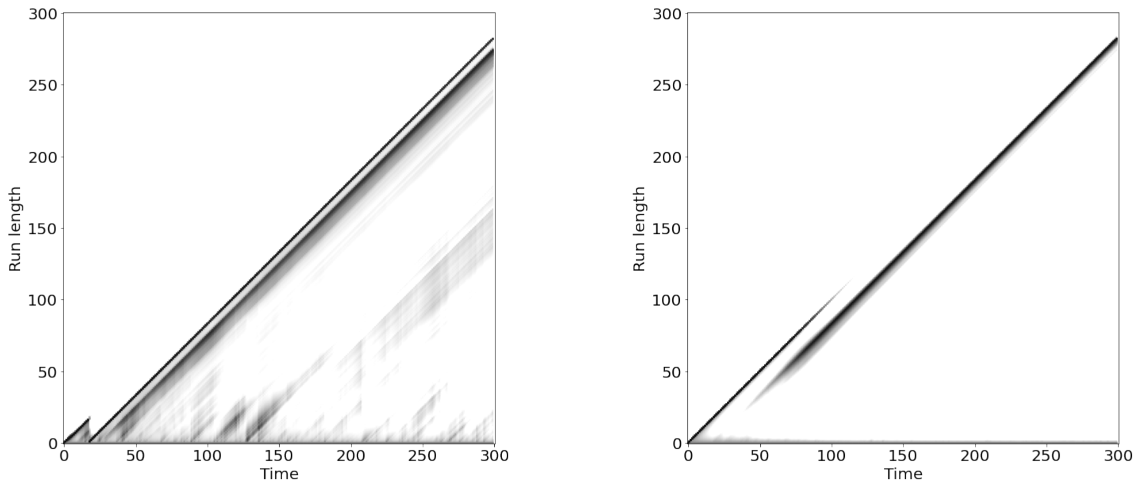


Figure 4.23. *Jasmine Table data normalized*

Before the first iteration κ_0, α_0, β_0 and μ_0 must be initialized, but how to initialize these parameters is an open question. In the literature, a commonly used initialization is to set all parameters to one. However, this setting does not work for our data-set. Figure 4.24b shows how the BOCPD performs with all parameters set to 1. A thorough study of different initial values revealed that setting the κ, α parameters to one, μ to sample mean and β to be equal to the hazard rate yielded reasonably good results.

The algorithm was applied to the jasmine data, visualized in Figure 4.23. All the parameters were initialized as mentioned above.



***(a)** Result of running BOCPD with t-distribution*

***(b)** Result of running BOCPD with the parameters set to 1 (t-distribution)*

***Figure 4.24.** Results of running the BOCPD with different initial parameters*

As can be seen from Figure 4.24a there is a drop at 18 seconds. This algorithm shows good results if run on data with clear transient and stable phases. More results will be presented in the results section.

This is a well working algorithm, but it is very difficult to set up and explain some parameters. The main difficulty of this algorithm is the parameters. It is not obvious how to initialize the parameters and what they mean. Over time probabilities in the matrix converge to roundoff error. This can be avoided by replacing all values that are, for example, less than 1e-10 by zero.

The matrix R in Algorithm 5 was used for possible visualizations of intermediate results if needed. Note that using R-matrix is not mandatory. Moreover, not using the matrix R makes the algorithm more memory efficient. Algorithm 5 is run on each channel separately although there is a possibility to generalize it such that it can be run on all channels simultaneously. Each data point from a channel is handled separately as well. Here we keep the matrix $\mathbf{R}$ for visualization purposes. The algorithm is not optimized for using the $\mathbf{R}$-matrix. Values in the matrix $\mathbf{R}$ are likely to converge to round-off error.

The left-hand sides of 4.41 and 4.42 do not depend on the full history but only on the probabilities of the previous time step (Markov chain property). Thus we can use the vector $\mathbf{L}$ for saving all values from the previous step. Instead of using the matrix $\mathbf{R}$, decision about change points can be done by following data in the vector $\mathbf{l}$. If the index of the maximum element is 1 at some iteration this means that a change point is detected and the algorithm can be stopped. If data contains significant noise and the existence of the change point is obscured another way can be employed. Normally, if there are no change points, values in the vector L are growing from start to end at each time step. In case of clear change point, the first entry in the vector will have the maximum value.

Algorithm 5: Bayesian Change Point with t-distribution as UPM

```
Result: time step $i$ where distributions have changed
$\mathbf{R} \leftarrow \mathbf{0}$ initialize matrix R with zeros $n \times n$
$R[0,0] = 1$ the first value is one
$\mathbf{l} \leftarrow [1]$ create a vector for keeping information from previous step
$H = \frac{1}{100}$ initialize hazard rate
$\alpha_0 = \kappa_0 = 1$ initialize parameters of UPM distribution
$\mu_0 = \bar{\mu}$ mean is sample mean
$\beta_0 = H$ $\beta$ will be equal to the Hazard rate
$\mathbf{a} = \mathbf{b} = \mathbf{m} = \mathbf{k} = []$ initialize vectors for $\alpha, \beta, \mu, \kappa$ for saving all
  previous values
found = False
i = 1 counter
while found == False do
    $x_i \leftarrow$ next point
    $\pi = function$ get_predictive_probabilities($x_i, \mathbf{a}, \mathbf{b}, \mathbf{m}, \mathbf{k}$)
    $\mathbf{g} = \mathbf{l} \times \pi \times (1 - H)$ calculate growth probabilities
    $\mathbf{c} = \sum (\mathbf{l} \times \pi \times H)$ calculate change point probabilities
    $\mathbf{ng}$ = $[\mathbf{c}, \mathbf{g}]$ concatenate into one vector change and growth
      probabilities
    R[i,:] = $\frac{\mathbf{ng}}{\sum \mathbf{ng}}$ normalize and put into the R-matrix
    $\mu_{\mathbf{n}} = \frac{(\mathbf{k} \times \mathbf{m} + x_i)}{\mathbf{k}+1}$ update mean-value
    $\kappa_{\mathbf{n}} = \mathbf{k} + 1$ update kappa-value
    $\alpha_{\mathbf{n}} = \mathbf{a} + \frac{1}{2}$ update alpha-value
    $\beta_{\mathbf{n}} = \mathbf{b} + \frac{\mathbf{k}(x_i - \mathbf{m})^2}{2(\mathbf{k}+1)}$ update beta-value
    $\mathbf{a} = [\alpha_0, \alpha_{\mathbf{n}}]$ concatenate vector alpha
    $\mathbf{b} = [\beta_0, \beta_{\mathbf{n}}]$
    $\mathbf{m} = [\mu_0, \mu_{\mathbf{n}}]$
    $\mathbf{k} = [\kappa_0, \kappa_{\mathbf{n}}]$
    $\mathbf{l} = [C, G]$ concatenate for next iteration
end
Function get_predictive_probabilities($x_i, \mathbf{a}, \mathbf{b}, \mathbf{m}, \mathbf{k}$):
    df = $2 \times \mathbf{a}$
    loc = $\mathbf{m}$
    scale = $\sqrt{\frac{\mathbf{b}(\mathbf{k}+1)}{\mathbf{a} \times \mathbf{k}}}$
    return probability t(x,df, loc, scale)
return
```

In case of obscured change point, the growing sequence in the vector will be broken indicating a possible change point.

The algorithm keeps all previous parameters of underlying PDF in vectors $\mathbf{a}, \mathbf{b}, \mathbf{m}, \mathbf{k}$, which means that these vectors will grow in size as time goes on. Thus, the algorithm is not optimal for long data series. The DIGITS data-set contains time series measurements of 300 seconds length, which allows us to ignore this problem.

The first part of Algorithm 5 (above row 10) initializes all needed values:

- H - hazard rate, which expresses our prior beliefs about the rate of change point
- $\alpha_0, \kappa_0, \mu_0, \beta_0$ - initial parameters of the Normal-Gamma distribution
- $\mathbf{a}, \mathbf{b}, \mathbf{m}, \mathbf{k}$ - vectors for saving the parameters of the Normal-Gamma PDF at each iteration
- i - counter for defining time step where change point is found

The second part of the algorithm is the iterative process described above:

- Get predictive probability of the new point for each run-length (row 13) - here the generalized t-distribution was used by calling the function *get_predictive_probabilities()* defined in row 28
- Calculate growth probabilities (row 14) - multiply vector L with predictive probabilities and the inverse of the hazard value
- Calculate change point probabilities (row 15) - multiply vector L with predictive probabilities, the hazard rate, and sum all.
- Update parameters of PDF (rows 18 - 25)

5 RESULTS

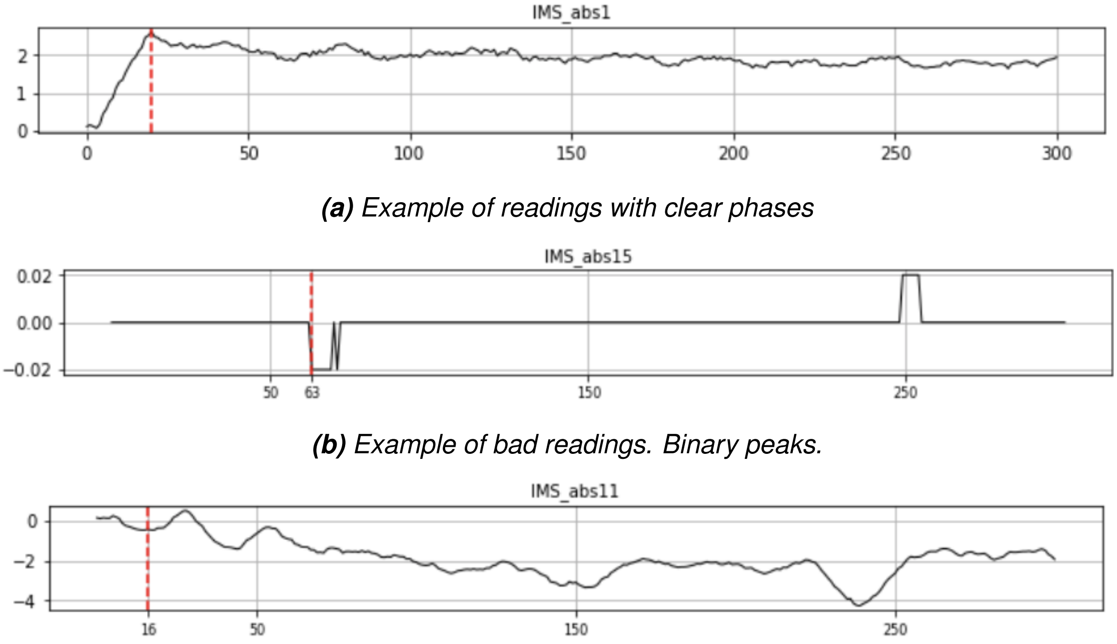
In this section results for three data sets will be presented as examples. The data set and the results from all 40 analyzed data sets can be found at `https://doi.org/10.5281/zenodo.4454381`. Here the exemplary analysis of one of the best data set, one of the worst data set, and a set with mixed data will be presented. A data set is characterized as good when there are clear transition and stable phases. These phases may be detected visually as well. A data set characterized as bad does not contain any clear phases. The problem of bad data set may arise from device insensitivity to certain scents, wrongly arranged measurement process, or internal errors of the ChemPro100i as seen in Figure 5.1b. It is assumed that the sensor with binary shift readings is corrupted. The wrong arrangement of measurements may have a significant impact. For example, if the time between collecting two data sets is too short then IMS signals might not return to the baseline 5.1. A bad data sets contain only fluctuations, that can not be explained and are, therefore, considered to be noise. A mixed data set contains good readings, bad readings, and readings with binary peaks.

Each change detection algorithm processes all 14 channels of a certain data set and the results are compared with ground truth values. Ground truth are change points, that are manually determined by visual inspection of the channel responses. The plots provided in this chapter demonstrate the results of running algorithms with the sliding window of size 10. The tables presented have the results of running algorithms with the sliding window of size 5, 10 and 15. For assessing the results of the detection algorithms the mean absolute error (MAE) scores were used. The MAE is calculated as

$$MAE(\mathbf{y}, \mathbf{x}) = \frac{1}{14} \sum_{i=1}^{14} |y_i - x_i|, \tag{5.1}$$

where $\mathbf{y}$ is a vector with ground truth values for each channel and $\mathbf{x}$ is a vector of detected change points for each channel.

The cases with binary peaks (Figure 5.1b and 5.1c) have ground truth points selected visually as well and this can falsify the results because ground truth points selected for measuring performance are subjective. Different people may have different opinions on where to place the actual change point. Especially for the data sets with unclear phase changes, this is a significant problem. However, using change point algorithms will reduce subjectivity in this process. If a set of parameters is determined and works well for a good data set, then the hypothesis is that it works well for a bad data set and that the change points it found can be trusted.



***(a)** Example of readings with clear phases*

***(b)** Example of bad readings. Binary peaks.*

***(c)** Example of readings without clear phases*

***Figure 5.1.** Examples of readings*

5.1 Summary of all data sets

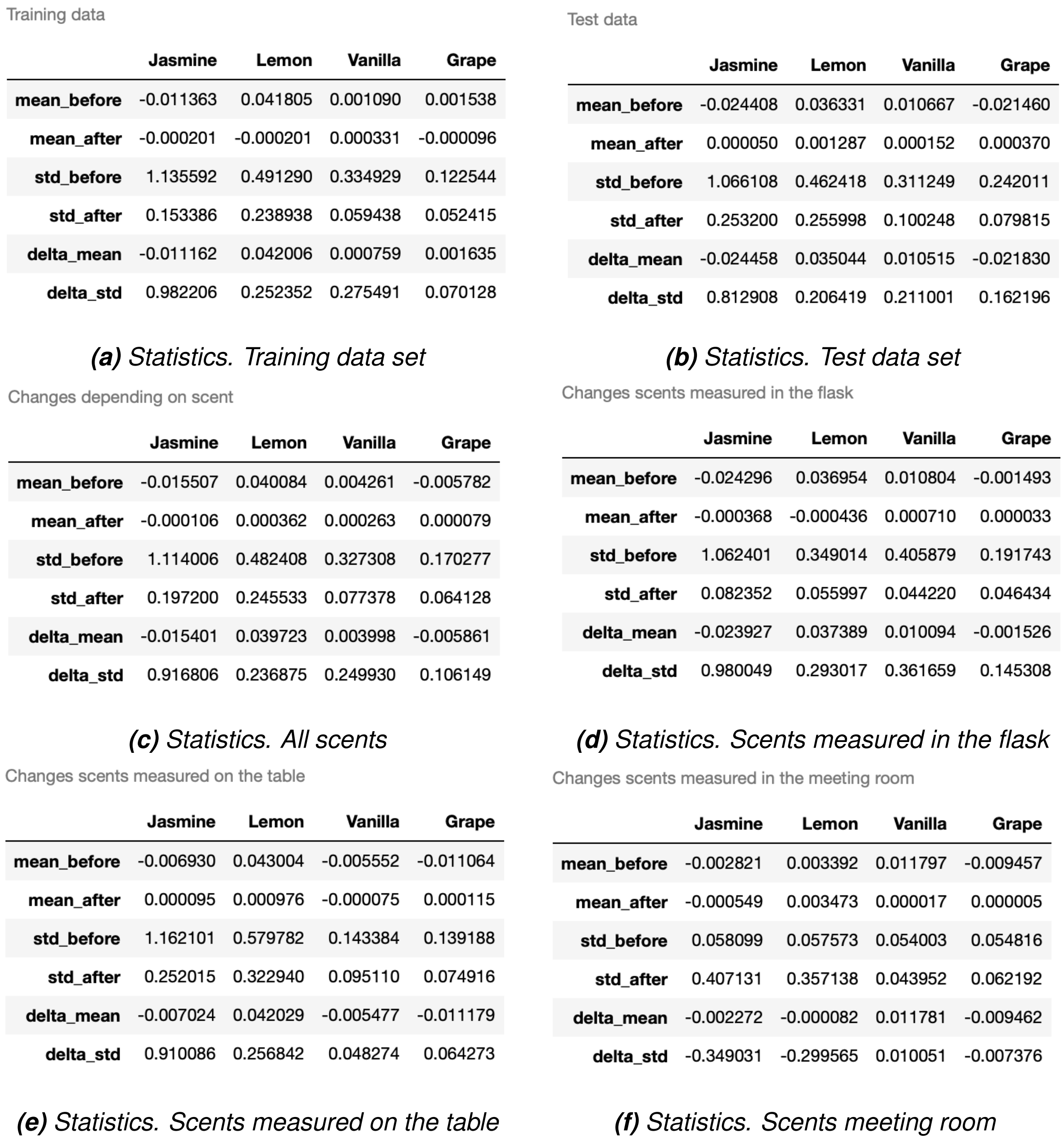
The data set contains 40 measurement sets from 4 scents. The scents are jasmine oil (1% concentration), lemon peel (grated peel of ripe lemon), vanilla (sliced dried vanilla fruit) and grape. The data sets were collected at Tampere University between October and November 2017. For the measurements the ion mobility based eNose ChemPro100i was used and data was sampled at 1 Hz.

The baseline is the measurement of ambient air in the room. The ambient air was measured for reducing noise in the measurements. The measurements were performed five times for each scent in the following sequence. The first 5 minutes were measured as a baseline. Subsequently, samples presented in the flask or on the table were measured for 5 minutes. Finally, ambient air was measured for 10 minutes to ensure that the IMS readings returned to the baseline.‘

Before implementing the change detection algorithms all data sets were summarized to get a big picture. The summary contains sample means and sample standard deviations before and after a change point. The change points were detected using the CUSUM with the moving window of size 5 for each channel separately. The statistics were calculated for all the data points belonging to the transition phase of the particular group and for all the data points belonging to the stable phase. For example, the change points were detected for the group "Training set". The sample mean was calculated for all the points belonging to all the transition phases in this group giving "mean_before". Similarly, sample mean was calculated for all the points belonging to all the stable phases giving "mean_after". The same approach was used for calculating "std_before" and "std_after" values.

There are two main statistics in the readings we can follow in order to detect a change. The readings at different phases show changes in the mean and the variance. Figure 5.2 shows several statistics for data grouped by measurement place and train/test sets. The data describes changes in the mean and variance before and after change points. Rows with names "mean_before" and "mean_after" demonstrate sample mean before the change point and after. Rows with names "std_before" and "std_after" contain sample standard deviations before and after. The rows "delta_mean" and "delta_std" show differences in changes before and after a change point.

If we compare rows "delta_mean" and "delta_std" the standard deviation demonstrates usually noticeable change compared to change in the mean. The statistics show as well scents, that often experience a very insignificant change. These are the vanilla and the grape measured on a table. However, scents measured from a flask show often more significant changes and have transient and stables phases.



Training data

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.011363	0.041805	0.001090	0.001538
mean_after	-0.000201	-0.000201	0.000331	-0.000096
std_before	1.135592	0.491290	0.334929	0.122544
std_after	0.153386	0.238938	0.059438	0.052415
delta_mean	-0.011162	0.042006	0.000759	0.001635
delta_std	0.982206	0.252352	0.275491	0.070128

(a) *Statistics. Training data set*

Test data

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.024408	0.036331	0.010667	-0.021460
mean_after	0.000050	0.001287	0.000152	0.000370
std_before	1.066108	0.462418	0.311249	0.242011
std_after	0.253200	0.255998	0.100248	0.079815
delta_mean	-0.024458	0.035044	0.010515	-0.021830
delta_std	0.812908	0.206419	0.211001	0.162196

(b) *Statistics. Test data set*

Changes depending on scent

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.015507	0.040084	0.004261	-0.005782
mean_after	-0.000106	0.000362	0.000263	0.000079
std_before	1.114006	0.482408	0.327308	0.170277
std_after	0.197200	0.245533	0.077378	0.064128
delta_mean	-0.015401	0.039723	0.003998	-0.005861
delta_std	0.916806	0.236875	0.249930	0.106149

(c) *Statistics. All scents*

Changes scents measured in the flask

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.024296	0.036954	0.010804	-0.001493
mean_after	-0.000368	-0.000436	0.000710	0.000033
std_before	1.062401	0.349014	0.405879	0.191743
std_after	0.082352	0.055997	0.044220	0.046434
delta_mean	-0.023927	0.037389	0.010094	-0.001526
delta_std	0.980049	0.293017	0.361659	0.145308

(d) *Statistics. Scents measured in the flask*

Changes scents measured on the table

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.006930	0.043004	-0.005552	-0.011064
mean_after	0.000095	0.000976	-0.000075	0.000115
std_before	1.162101	0.579782	0.143384	0.139188
std_after	0.252015	0.322940	0.095110	0.074916
delta_mean	-0.007024	0.042029	-0.005477	-0.011179
delta_std	0.910086	0.256842	0.048274	0.064273

(e) *Statistics. Scents measured on the table*

Changes scents measured in the meeting room

	Jasmine	Lemon	Vanilla	Grape
mean_before	-0.002821	0.003392	0.011797	-0.009457
mean_after	-0.000549	0.003473	0.000017	0.000005
std_before	0.058099	0.057573	0.054003	0.054816
std_after	0.407131	0.357138	0.043952	0.062192
delta_mean	-0.002272	-0.000082	0.011781	-0.009462
delta_std	-0.349031	-0.299565	0.010051	-0.007376

(f) *Statistics. Scents meeting room*

Figure 5.2. *Statistics for scents.*

5.2 Good data set

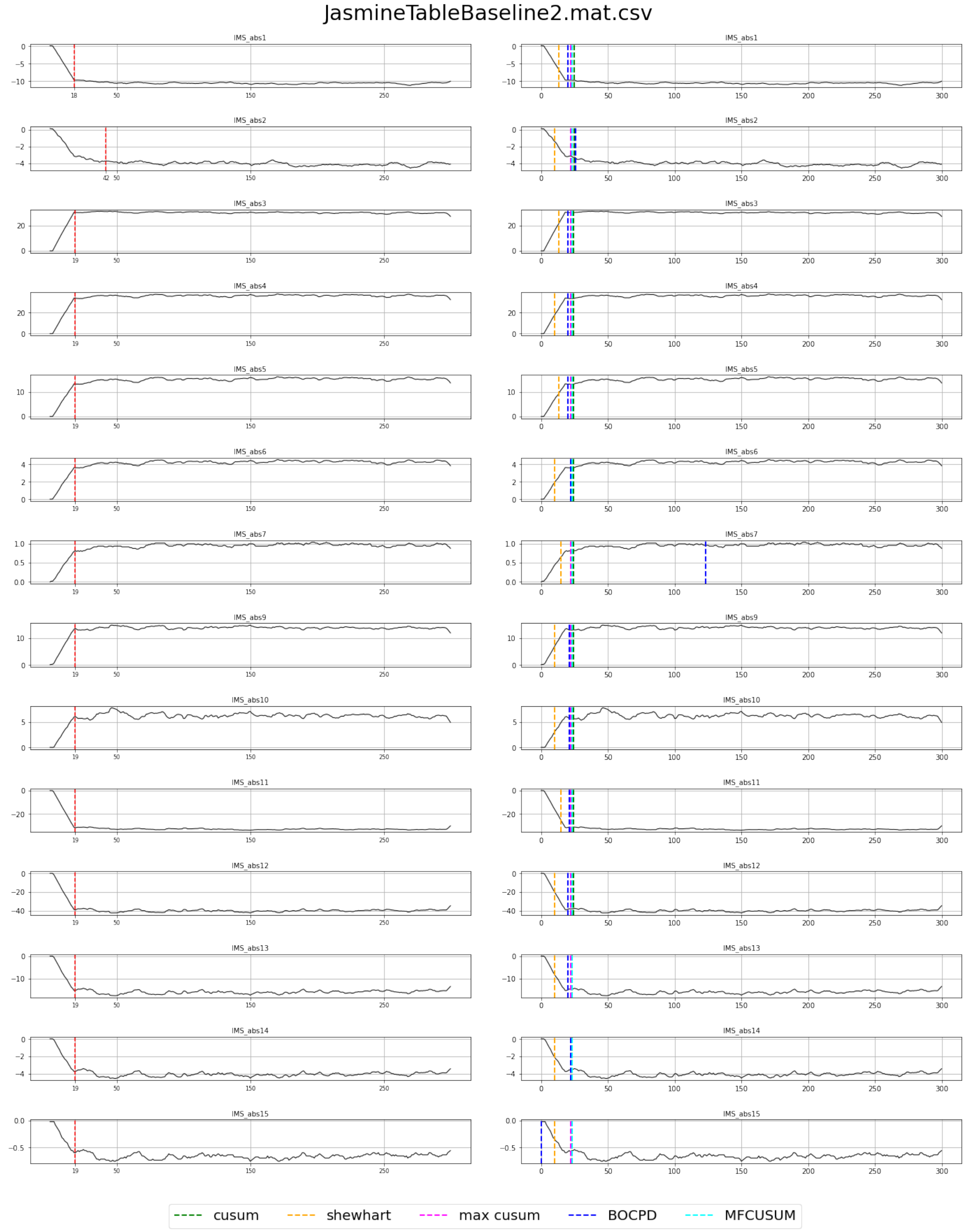


Figure 5.3. *Algorithms applied to jasmine table data set*

Figure 5.3 shows the results of all algorithms applied to the second jasmine data set measured on the table. The abbreviations used in the legends are:

- *shewhart* - Shewhart's Charts

- *cusum* - CUSUM
- *max mcusum* - Multivariate Max-CUSUM Chart
- *MFCUSUM* - Matrix Form CUSUM
- *BAYES* - Bayesian Online Change Point Detector

The red lines on the left column of the plots indicate ground truth change points. As can be seen, all the channels demonstrate clear phase change and all the algorithms found the change point comparatively well. The Shewhart Control Charts algorithm has detected a change point too early and the Bayesian Online Change point algorithm failed two times, but overall performances are very good.

reading	gt	window size	shewhart	cusum	mfcusum	max mcusum	bayes
IMS_abs1	18	5	10	22	21	8	20
		10	13	25	23	22	20
		15	19	28	24	22	20
IMS_abs2	42	5	5	22	21	8	26
		10	10	25	23	22	26
		15	15	29	24	22	26
IMS_abs3	19	5	5	22	21	8	20
		10	13	24	23	22	20
		15	18	27	24	22	20
IMS_abs4	19	5	5	22	21	8	20
		10	10	24	23	22	20
		15	15	27	24	22	20
IMS_abs5	19	5	10	22	21	8	20
		10	13	24	23	22	20
		15	15	27	24	22	20
IMS_abs6	19	5	5	22	21	8	22
		10	10	24	23	22	22
		15	19	27	24	22	22
IMS_abs7	19	5	11	22	21	8	123
		10	15	24	23	22	123
		15	19	27	24	22	123
IMS_abs9	19	5	5	21	21	8	21
		10	10	24	23	22	21
		15	15	26	24	22	21
IMS_abs10	19	5	5	21	21	8	21
		10	10	24	23	22	21
		15	15	26	24	22	21
IMS_abs11	19	5	10	22	21	8	21
		10	15	24	23	22	21
		15	19	27	24	22	21
IMS_abs12	19	5	5	21	21	8	20
		10	10	24	23	22	20
		15	15	26	24	22	20
IMS_abs13	19	5	5	21	21	8	20
		10	10	23	23	22	20
		15	15	25	24	22	20
IMS_abs14	19	5	5	21	21	8	22
		10	10	23	23	22	22
		15	15	25	24	22	22
IMS_abs15	19	5	5	21	21	8	0
		10	10	23	23	22	0
		15	15	24	24	22	0
mean run time (sec)			1.334e-03	6.455e-03	4.745e-04	9.224e-04	1.845e-01
mean relative run time			2.811	13.603	1.000	1.944	388.848

Table 5.1. *Change points found for each size of the moving window. Jasmine table set.*

Table 5.1 shows change points found by the algorithms for each size of the moving window and each channel. The *gt*-column in the table shows ground truth. The last row of the table shows the relative running time of each algorithm for all channels. Relative run time is calculated with respect to the fastest algorithm. The second last row shows direct run times. As can be seen from the table, the fastest and one of the closest to ground truth algorithm is Matrix Form CUSUM. The run times are generally very small and even performance of the Bayesian algorithm will not be a problem. The ChemPro100i is sampling at rate 1 Hz, which means that all algorithms, including the Bayesian algorithm, are able to check for change points in real-time, between consecutive samples.

moving window size	shewhart	cusum	mfcusum	max mcusum	bayes
5	14.07	3.86	3.43	12.57	11.29
10	9.21	5.79	5.14	4.29	11.29
15	4.36	7.79	6.00	4.29	11.29

Table 5.2. *MAE of each algorithm for jasmine table*

Table 5.2 shows the Mean Absolute Error of each algorithm for all sizes of the moving window. The Multivariate Max-CUSUM has the lowest MAE:s for window sizes 10 and 15, which means that its estimates are constantly closer to the actual change point for these window sizes. Given the information in these tables and plots, it can be stated that the larger the moving window for the Shewhart and Max CUSUM the better their performance. For the CUSUM and MFCUSUM increasing the size of the moving window decreases their accuracy. Each algorithm calculates first the initial mean and variance. The Multivariate Max-CUSUM calculates the initial mean vector and covariance matrix. As was explained in chapter 4 several points at the beginning of time series have large variability, which affects the initial covariance matrix. This causes significant sensitivity in the Multivariate Max-CUSUM. However, there are restrictions with the size of the moving window, because the transition phases are usually short. Since transition phases are usually 15 - 25 points long it is not feasible to use a moving window size of more than 20. Because the first several points of each time series have large variability, the length of the moving window needs to be 5 or longer. Out of the presented algorithms, the Matrix Form CUSUM and Multivariate Max CUSUM in case Jasmine data set are more suitable.

5.3 Bad data set

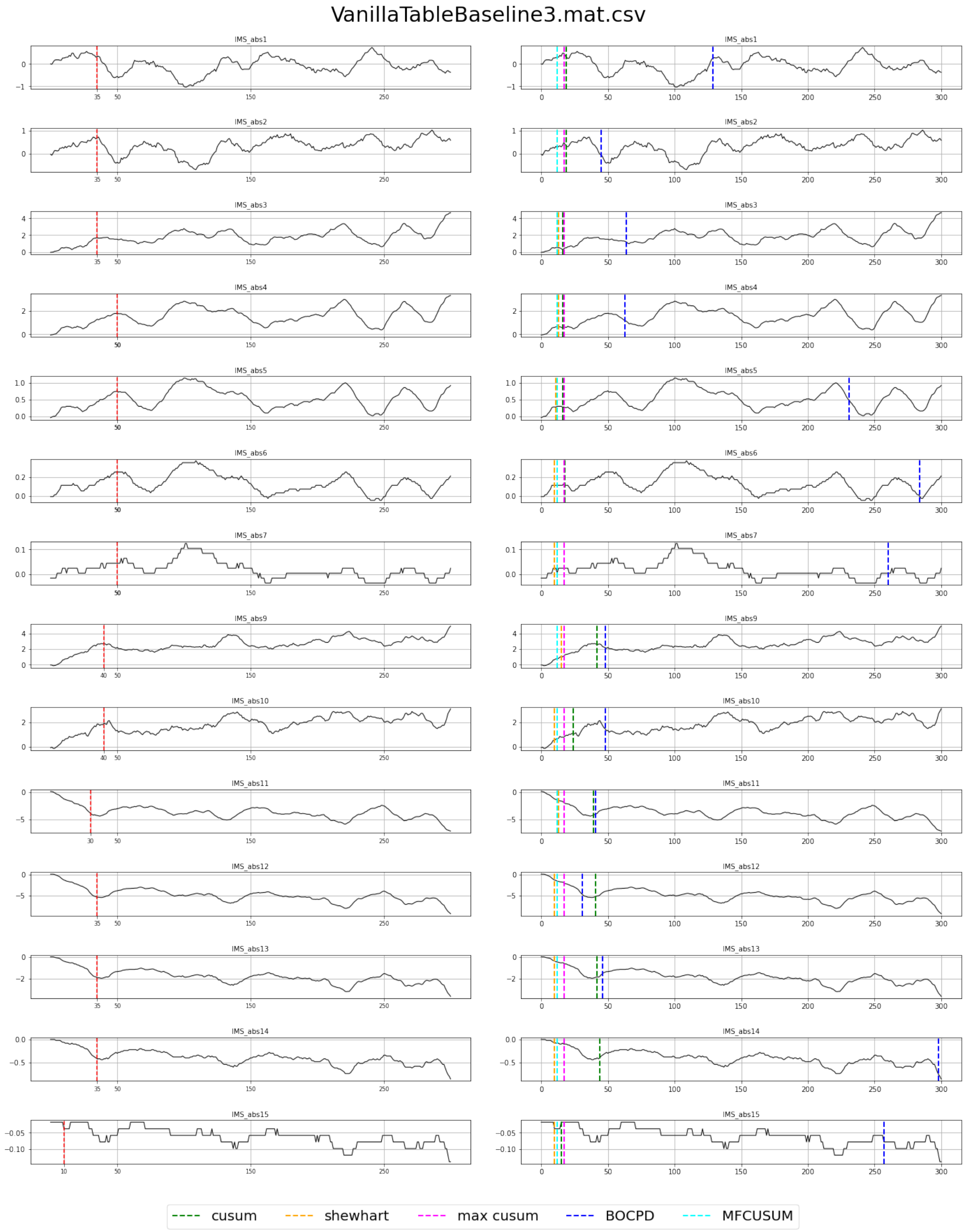


Figure 5.4. *Algorithms applied to vanilla table data set*

Vanilla measured from the table mostly does not have any clear phase transition points. It is very hard to determine change points for these readings. The ground truth points for this data set were selected subconsciously. As a result, the plots show (Figure 5.4) that the algorithms react mostly on local trend changes. Channels 7 and 15 have binary looking plots. However, there are slight trends so we can classify them as belonging to the bad data sets. Algorithms can detect change points on these channels as well.

reading	gt	window size	shewhart	cusum	mfcusum	max mcusum	bayes
IMS_abs1	35	5	7	7	7	11	129
		10	12	19	12	17	129
		15	17	19	15	21	129
IMS_abs2	35	5	8	12	7	11	45
		10	12	19	12	17	45
		15	15	37	15	21	45
IMS_abs3	35	5	10	13	7	11	64
		10	13	16	12	17	64
		15	16	24	15	21	64
IMS_abs4	50	5	5	14	7	11	63
		10	13	16	12	17	63
		15	16	22	15	21	63
IMS_abs5	50	5	5	13	7	11	231
		10	11	16	12	17	231
		15	16	21	15	21	231
IMS_abs6	50	5	5	13	7	11	284
		10	10	18	12	17	284
		15	15	21	15	21	284
IMS_abs7	50	5	5	12	7	11	260
		10	10	12	12	17	260
		15	15	21	15	21	260
IMS_abs9	40	5	5	41	7	11	48
		10	15	42	12	17	48
		15	16	45	15	21	48
IMS_abs10	40	5	10	27	7	11	48
		10	10	24	12	17	48
		15	15	27	15	21	48
IMS_abs11	30	5	12	36	7	11	41
		10	13	39	12	17	41
		15	19	42	15	21	41
IMS_abs12	35	5	5	40	7	11	31
		10	10	41	12	17	31
		15	15	44	15	21	31
IMS_abs13	35	5	5	40	7	11	46
		10	10	42	12	17	46
		15	15	45	15	21	46
IMS_abs14	35	5	5	16	7	11	298
		10	10	44	12	17	298
		15	15	46	15	21	298
IMS_abs15	10	5	5	0	7	11	257
		10	10	15	12	17	257
		15	15	0	15	21	257
mean run time (sec)			2.585e-03	1.552e-02	1.984e-04	9.008e-04	5.275e-01
mean relative run time			13.026	78.219	1.000	4.539	2658.041

Table 5.3. *Change points found for each size of moving window. Vanilla table set.*

moving window size	shewhart	cusum	mfcusum	max mcusum	bayes
5	31.29	20.00	30.86	27.00	94.50
10	26.50	17.36	26.14	21.86	94.50
15	22.86	15.29	23.57	18.43	94.50

Table 5.4. *MAE of each algorithm for vanilla table*

Table 5.3 demonstrates that Bayes Online Change Point algorithm fails on almost all channels and its running time is very long compared to the other algorithms. The MAE results from Table 5.4 tell us that the Multivariate Max-CUSUM yields again estimates the closest to the actual change points and is one of the fastest algorithms. The MAE scores in this table are greater than scores in the jasmine data set. This means that the algorithms perform worse. The accuracy of the algorithms decreased because the vanilla data set does not have clear phase changes. The other reason is the subjectivity of ground truths. As can be seen from Figure 5.3 variability of detected change points is significant. Mostly algorithms reacted on the first peak of readings. It is possible that algorithms responded on real change points that can not be visually detected or that the ChemPro did not react to this scent.

5.4 Mixed data set

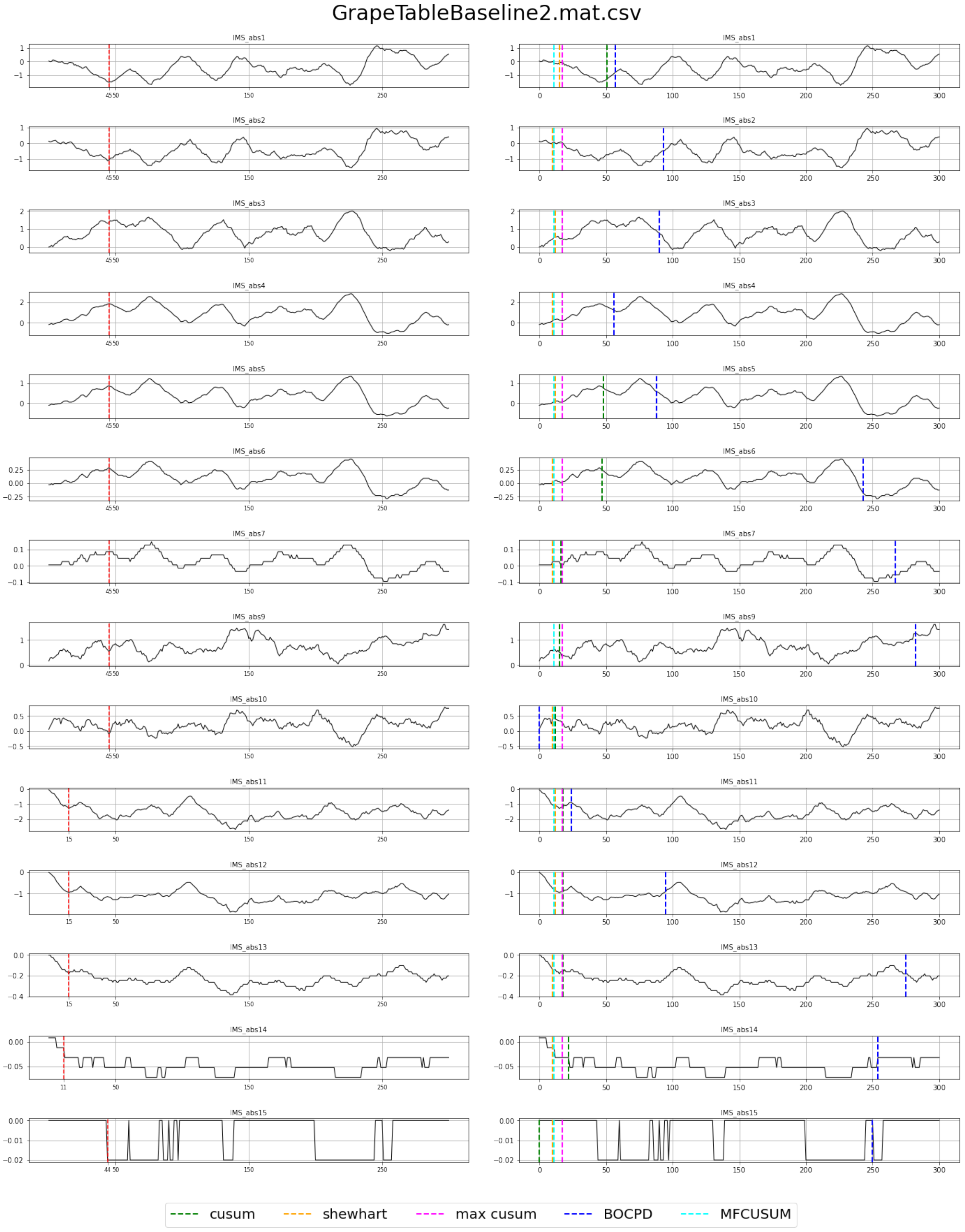


Figure 5.5. *Algorithms applied to grape table baseline 2 data set*

The plots in Figure 5.5 show the last exemplary data set in this section. Again there are no clear change points. It can be seen, furthermore, that the readings on the channels "IMS_abs14" and "IMS_abs15" are possibly corrupted. It is almost impossible to detect where the change happened especially in the case of the channels "IMS_abs14" and

"IMS_abs15". Since channels 14 and 15 have very clear binary shifted readings and other readings do not have clear change points, this data set was classified as mixed.

reading	gt	window size	shewhart	cusum	mfcusum	max mcusum	bayes
IMS_abs1	45	5	6	7	6	13	57
		10	15	51	11	17	57
		15	16	16	16	21	57
IMS_abs2	45	5	6	6	6	13	93
		10	10	11	11	17	93
		15	16	16	16	21	93
IMS_abs3	45	5	9	15	6	13	90
		10	12	17	11	17	90
		15	16	20	16	21	90
IMS_abs4	45	5	5	16	6	13	56
		10	10	17	11	17	56
		15	16	49	16	21	56
IMS_abs5	45	5	6	7	6	13	88
		10	12	48	11	17	88
		15	16	49	16	21	88
IMS_abs6	45	5	5	7	6	13	243
		10	10	47	11	17	243
		15	15	17	16	21	243
IMS_abs7	45	5	5	0	6	13	267
		10	10	16	11	17	267
		15	15	16	16	21	267
IMS_abs9	45	5	6	13	6	13	282
		10	11	15	11	17	282
		15	16	17	16	21	282
IMS_abs10	45	5	6	7	6	13	0
		10	10	12	11	17	0
		15	16	17	16	21	0
IMS_abs11	15	5	7	16	6	13	24
		10	12	18	11	17	24
		15	16	21	16	21	24
IMS_abs12	15	5	7	16	6	13	95
		10	12	18	11	17	95
		15	16	21	16	21	95
IMS_abs13	15	5	5	16	6	13	275
		10	10	18	11	17	275
		15	15	22	16	21	275
IMS_abs14	11	5	5	0	6	13	254
		10	10	22	11	17	254
		15	15	21	16	21	254
IMS_abs15	44	5	0	0	6	13	250
		10	10	0	11	17	250
		15	15	0	16	21	250
mean run time (sec)			6.252e-03	2.120e-02	1.707e-04	7.606e-04	6.819e-01
mean relative run time			36.623	124.197	1.000	4.456	3994.716

Table 5.5. *Change points found for each size of moving window. Grape table baseline 2 set*

moving window size	shewhart	cusum	mfcusum	max mcusum	bayes
5	30.50	27.50	30.07	23.36	118.50
10	25.07	18.36	25.07	20.79	118.50
15	21.29	19.79	21.21	19.07	118.50

Table 5.6. *MAE of each algorithm for grape baseline 2 table*

Tables (5.5 and 5.6) show that the detected change points are far from the actual ones.

This is caused by unclear phase changes and by the binary looking channels. The results demonstrate that the CUSUM, the Matrix Form CUSUM, and Multivariate Max-CUSUM performed almost equally well, especially if taking into account, that the ground truth change points may be shifted around.

5.5 Results review

The results over the entire data set are summarized in Table 5.7. The table contains names of all data sets, their quality, the best algorithm for each window size and its MAE score. The table shows that MAE for scents measured from flask are often better because the sealed flask contains only scented air. The scents measured on the table usually have more unstable readings because surrounding air containing other scents significantly impacts measurements, which cause more problems in change point detection. For details refer to [1] and [13]. The Matrix Form CUSUM and Multivariate Max CUSUM yielded the best results in many cases.

However, there are data sets classified as "bad" and "mixed", where ground truth change points are marked manually. The ground truth change points being marked manually give subjectivity, which increases MAE. This means, that the algorithms may potentially perform better with the "bad" and "mixed" data sets and MAE scores for these data sets are not very reliable. Thus the best algorithms for the good data sets are the CUSUM, the Matrix Form CUSUM and Multivariate Max-CUSUM. The algorithms are possibly detecting actual change points for these readings or they just guess because they need to find a change point. Since the algorithms are online they can run infinitely until a change point occurs. The ground truth points chosen by different people may differ significantly. This is a crucial part to choose the best algorithm and use it as a reference for further classification. We can assume that the best algorithm gives sufficient detection results even for the "bad" data sets. A small error in change point detection has no significant impact.

For scent classification different features can be used, such as variance, mean, derivatives. The length of transient phase can possibly uniquely characterize a scent. From the classification methods point of view, the best algorithm will provide similar features for "bad" data sets and similar features for "good" data sets. This difference of features makes it possible to apply classification algorithms. It is very important to have an automated change point detection system. This excludes subjectivity and thus increases the accuracy of change point detections.

There is one more problem with binary shifted readings. In this case, it is impossible to detect change points so we can exclude these channels from calculations. The decision about excluding channels from calculations can be made using, e.g. running variance. If the variance is zero or almost zero during several iterations or in the initial sample, this indicates that there is a probably binary channel. There is no information about why channels 7, 14, and 15 in the case of measurements from the table are often behaving in such a binary way. This part needs to be investigated.

	set name	quality	window size	best algorithm	MAE	set name	quality	window size	best algorithm	MAE	
Grape	GrapeFlaskBaseline1	mixed	5	max mcusum	1.286	Lemon peelFlaskBaseline1	good	5	cusum	4.286	Lemon peel
			10	max mcusum	3.429			10	cusum	5.643	
			15	mfcusum	1.143			15	cusum	7.214	
	GrapeFlaskBaseline2	good	5	max mcusum	3.571	Lemon peelFlaskBaseline2	good	5	max mcusum	2.714	
			10	max mcusum	2.714			10	max mcusum	2.714	
			15	mfcusum	4.000			15	shewhart	3.929	
	GrapeFlaskBaseline3	mixed	5	max mcusum	2.286	Lemon peelFlaskBaseline3	good	5	cusum	2.429	
			10	max mcusum	3.143			10	max mcusum	3.929	
			15	shewhart	3.500			15	shewhart	3.714	
	GrapeFlaskBaseline4	mixed	5	cusum	7.786	Lemon peelFlaskBaseline4	good	5	cusum	6.429	
			10	cusum	4.857			10	max mcusum	1.286	
			15	mfcusum	1.071			15	mfcusum	1.286	
	GrapeFlaskBaseline5	mixed	5	max mcusum	4.857	Lemon peelFlaskBaseline5	good	5	max mcusum	2.214	
			10	max mcusum	4.429			10	max mcusum	2.214	
			15	max mcusum	5.286			15	mfcusum	1.786	
	GrapeTableBaseline1	mixed	5	cusum	17.714	Lemon peelTableBaseline1	mixed	5	max mcusum	2.000	
			10	max mcusum	9.929			10	max mcusum	3.000	
			15	max mcusum	7.643			15	shewhart	4.071	
	GrapeTableBaseline2	bad	5	max mcusum	23.357	Lemon peelTableBaseline2	bad	5	cusum	2.857	
			10	cusum	18.357			10	max mcusum	3.643	
			15	max mcusum	19.071			15	mfcusum	3.643	
	GrapeTableBaseline3	bad	5	max mcusum	19.429	Lemon peelTableBaseline3	bad	5	cusum	4.571	
			10	cusum	24.714			10	mfcusum	0.000	
			15	max mcusum	17.429			15	max mcusum	1.000	
	GrapeTableBaseline4	bad	5	max mcusum	11.643	Lemon peelTableBaseline4	mixed	5	max mcusum	1.000	
			10	max mcusum	11.643			10	mfcusum	1.000	
			15	shewhart	11.286			15	shewhart	0.786	
	GrapeTableBaseline5	bad	5	cusum	24.143	Lemon peelTableBaseline5	mixed	5	max mcusum	5.643	
			10	max mcusum	13.643			10	mfcusum	7.643	
			15	max mcusum	13.071			15	max mcusum	4.071	
Jasmine	JasmineFlaskBaseline1	good	5	max mcusum	1.000	VanillaFlaskBaseline1	good	5	cusum	5.857	Vanilla
			10	bayes	2.786			10	cusum	7.000	
			15	shewhart	2.714			15	mfcusum	7.857	
	JasmineFlaskBaseline2	good	5	max mcusum	0.929	VanillaFlaskBaseline2	good	5	max mcusum	11.429	
			10	max mcusum	2.643			10	max mcusum	14.143	
			15	shewhart	2.357			15	max mcusum	12.714	
	JasmineFlaskBaseline3	good	5	max mcusum	2.000	VanillaFlaskBaseline3	mixed	5	max mcusum	4.071	
			10	max mcusum	1.000			10	max mcusum	8.357	
			15	max mcusum	3.000			15	shewhart	6.286	
	JasmineFlaskBaseline4	good	5	mfcusum	2.571	VanillaFlaskBaseline4	good	5	max mcusum	3.214	
			10	max mcusum	0.571			10	max mcusum	2.643	
			15	shewhart	3.071			15	max mcusum	3.214	
	JasmineFlaskBaseline5	good	5	cusum	9.571	VanillaFlaskBaseline5	good	5	cusum	11.571	
			10	mfcusum	4.000			10	max mcusum	6.357	
			15	max mcusum	0.000			15	cusum	6.286	
	JasmineTableBaseline1	good	5	max mcusum	2.571	VanillaTableBaseline1	mixed	5	max mcusum	18.643	
			10	max mcusum	3.429			10	max mcusum	4.929	
			15	shewhart	3.357			15	max mcusum	5.786	
	JasmineTableBaseline2	good	5	mfcusum	3.429	VanillaTableBaseline2	bad	5	max mcusum	21.929	
			10	max mcusum	4.286			10	max mcusum	9.929	
			15	max mcusum	4.286			15	max mcusum	9.929	
	JasmineTableBaseline3	good	5	max mcusum	1.000	VanillaTableBaseline3	bad	5	cusum	20.000	
			10	mfcusum	4.000			10	cusum	17.357	
			15	shewhart	3.286			15	cusum	15.286	
	JasmineTableBaseline4	good	5	max mcusum	0.929	VanillaTableBaseline4	bad	5	max mcusum	29.929	
			10	max mcusum	0.929			10	max mcusum	23.071	
			15	shewhart	2.500			15	cusum	22.071	
	JasmineTableBaseline5	good	5	max mcusum	1.000	VanillaTableBaseline5	mixed	5	max mcusum	27.286	
			10	max mcusum	1.000			10	max mcusum	26.571	
			15	shewhart	3.143			15	cusum	22.643	

Table 5.7. *Results over the entire data set*

6 DISCUSSION

The results in Section [5] show that change detection algorithms work well of more or less visible point of change. When a time series does not have a clear transition phase such simple algorithms do not perform well. However, we focused on simpler algorithms because one of the tasks was to select a range of techniques and implement them. For future research more sophisticated techniques will be studied for online change point detection. For example, algorithms using neural networks are an option[14]. The paper proposes using Bayesian Online Change Point Detection with a neural network. Another technique is to use time series analysis [15], specifically Auto Regressive Moving Average (ARMA) with Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) innovations models for online change point detection.

The problem with invisible phase changes may be solved by using more sophisticated preprocessing approaches. In this work, we used a derivative of time series readings also called "self lag differencing" of order 1. The task of this differencing is to remove seasonal changes from the time series. We could use other orders of differencing, but often short length of a transition phase does not provide us such freedom. The other differencing operation is Log Lag Differencing, which is not applied in this work. Its task is to stabilize the variance of the data, which may affect the results. In the theory section [3] we showed that the variance tends to change after the change point. One more possible approach to preprocess the data and detect a change point is to employ time series models such as ARMA. Since we have a moving window we can fit models to arriving sets of points and make decisions based on that. However, by removing the trend from the raw data series we might lose some valuable information. The difference operation may be applied to raw data to detect the existence of a trend.

The other problem is the performance measurement of the algorithms. In case of clear change points, which can be detected visually and used as ground truths, the algorithms perform well and MAE-values are reliable. In situations where we do not have certainty on where to place ground truth points, the ground truth points are assigned subjectively. This causes variability in the ground truth, which in turn affects the reliability of MAE-values. One of the main reasons for using change detection algorithms is to remove subjectivity from the process. The best performing algorithm has to be chosen and kept as a reference. Even if the chosen algorithm will detect change points with poor accuracy for the "bad" data sets it will provide similar results for all "bad" data sets. Among others, this will provide one more feature for further classification of scents.

The validity of readings affects results as well. Figure 5.1b shows how the channel IMS_abs15 provides binary readings. In this situation, it is not possible to determine the ground truth point. Moreover, such a channel is not suitable for being fed into algorithms. Such readings may signal corrupted sensors. However such channels were taken into account while calculating MAE-performance. There is no knowledge about what causes these channels to behave like that, thus there are no proposals on how to overcome this problem. The only reasonable way is to detect such channels and ignore them. These channels have to be investigated properly.

Sometimes IMS sensors do not react to certain scents or their concentrations. Knowledge about channels not reacting to presented scents can potentially increase detection accuracy. Excluding such channels from calculations will decrease the running time of the algorithms and increase accuracy. Detecting channels not reacting to scent can be implemented for example by following what values other channels get on average. The non-reacting channels are a problem particularly for the Multivariate MAX CUSUM algorithm because at the end of an iteration it yields a value, which is not related to a specific channel. However, prior knowledge about non-working channels can be potentially a feature for classification tasks if for different scents there are specific channels that do not react to a specific scent.

The questions asked in the Introduction part were:

1. Is it possible to find change points in the IMS readings by means of the change detection algorithms?
2. What is the computational cost?
3. How reliable these algorithms are?
4. What is the most effective algorithm?

It is possible to detect change points in case of existence of the change points in the readings. In case of unclear or non-existing change points, we have to rely on the detections provided by the algorithms.

The computational cost of all presented algorithms is very low compared to the frequency of sampling. It can be stated that all the presented algorithms consume less time than needed for sampling. The most costly algorithm in terms of time consumed is the Bayesian Online Change Point Detection algorithm, which still performs faster than the sampling.

Presented algorithms are reliable and provide trustworthy results. In case of clear change points, the algorithms perform well. In other cases, we have to rely on the results because ground truths contain only subjective information. Moreover, the algorithms provide detections of time steps, which usually indeed contain change points.

The most effective algorithm in terms of accuracy is the Multivariate Max-CUSUM. In terms of computational cost the most effective is the Matrix Form CUSUM. Choosing between these two the most effective turned out to be the Multivariate Max-CUSUM algorithm because it is multivariate as the Matrix Form CUSUM. It is also the most accurate algorithm.

REFERENCES

[1] Müller, P., Salminen, K., Nieminen, V., Kontunen, A., Karjalainen, M., Isokoski, P., Rantala, J., Savia, M., Väliaho, J., Kallio, P., Lekkala, J. and Surakka, V. Scent classification by K nearest neighbors using ion-mobility spectrometry measurements. *Expert Systems with Applications* 115 (2019), 593–606. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.08.042. URL: http://www.sciencedirect.com/science/article/pii/S0957417418305566.

[2] Raj, V. B., Singh, H., Nimal, A., Sharma, M. and Gupta, V. Oxide thin films (ZnO, TeO2, SnO2, and TiO2) based surface acoustic wave (SAW) E-nose for the detection of chemical warfare agents. eng. *Sensors and actuators. B, Chemical* 178 (2013), 636–647. ISSN: 0925-4005.

[3] Müller, P., Salminen, K., Kontunen, A., Karjalainen, M., Isokoski, P., Rantala, J., Leivo, J., Väliaho, J., Kallio, P., Lekkala, J. and Surakka, V. *Online Scent Classification by Ion-Mobility Spectrometry Sequences*. eng. fi=Tampereen ammattikorkeakoulu.

[4] Dodds, J. N. and Baker, E. S. Ion Mobility Spectrometry: Fundamental Concepts, Instrumentation, Applications, and the Road Ahead. *Journal of the American Society for Mass Spectrometry* 30.11 (2019). PMID: 31493234, 2185–2195. DOI: 10.1021/jasms.8b06240. eprint: https://pubs.acs.org/doi/pdf/10.1021/jasms.8b06240. URL: https://pubs.acs.org/doi/abs/10.1021/jasms.8b06240.

[5] Montgomery, D. C. *Introduction to statistical quality control, 6th edition*. John Wiley & Sons, Inc., 2009.

[6] Basseville, N. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[7] Zamora, D. and Blanco, M. Improving the efficiency of ion mobility spectrometry analyses by using multivariate calibration. *Analytica Chimica Acta* 726 (2012), 50–56. ISSN: 0003-2670. DOI: https://doi.org/10.1016/j.aca.2012.03.023. URL: http://www.sciencedirect.com/science/article/pii/S0003267012004400.

[8] PAGE, E. S. CONTINUOUS INSPECTION SCHEMES. *Biometrika* 41.1-2 (June 1954), 100–115. ISSN: 0006-3444. DOI: 10.1093/biomet/41.1-2.100. eprint: https://academic.oup.com/biomet/article-pdf/41/1-2/100/1243987/41-1-2-100.pdf. URL: https://doi.org/10.1093/biomet/41.1-2.100.

[9] Healy, J. D. A Note on Multivariate CUSUM Procedures. *Technometrics* 29.4 (1987), 409–412. ISSN: 00401706. URL: http://www.jstor.org/stable/1269451.

[10] Adams, R. P. and MacKay, D. J. C. Bayesian Online Changepoint Detection. (2007). arXiv: 0710.3742 [stat.ML].

[11] Rinne, H. *The Hazard rate : Theory and inference (with supplementary MATLAB-Programs)*. eng. Justus-Liebig-UniversitÃ¤t, 2014. URL: http://geb.uni-giessen.de/geb/volltexte/2014/10793.

[12] J.A.Grant. *BAYESIAN CHANGEPOINTDETECTION IN SOLAR ACTIVITY DATA*. 2014. URL: https://www.lancaster.ac.uk/postgrad/grantj/mastersdiss.pdf.

[13] Müller, P., Salminen, K., Kontunen, A., Karjalainen, M., Isokoski, P., Rantala, J., Leivo, J., Valiaho, J., Kallio, P., Lekkala, J. and Surakka, V. Online Scent Classification by Ion-Mobility Spectrometry Sequences. *Frontiers in Applied Mathematics and Statistics* 5 (July 2019). DOI: 10.3389/fams.2019.00039.

[14] Titsias, M. K., Sygnowski, J. and Chen, Y. Sequential Changepoint Detection in Neural Networks with Checkpoints. (2020). arXiv: 2010.03053 [cs.LG].

[15] Choi, H., Ombao, H. and Ray, B. Sequential Change-Point Detection Methods for Nonstationary Time Series. eng. *Technometrics* 50.1 (2008), 40–52. ISSN: 0040-1706.