

A Meta-Analysis of Metrics for Change Point Detection Algorithms



Matt Chapman

matthew.chapman@student.uva.nl

Spring 2017, 64 pages

Supervisor: Evangelos Kanoulas, Universiteit van Amsterdam
Host organisation: Buzzcapture B.V., <http://www.buzzcapture.com>



UNIVERSITEIT VAN AMSTERDAM
FACULTEIT DER NATUURWETENSCHAPPEN, WISKUNDE EN INFORMATICA
MASTER SOFTWARE ENGINEERING
<http://www.software-engineering-amsterdam.nl>

Contents

Abstract	3
Acknowledgments	4
1 Introduction	5
1.1 Problem Statement	5
1.2 Motivation	6
1.3 Document Structure	7
2 Related Work	8
2.1 Change Point Detection Algorithms & Their Applications	8
2.2 Algorithm Accuracy Evaluation	8
3 Background	10
3.1 Change Detection Algorithms	10
3.1.1 Pruned Exact Linear Time	10
3.1.2 Segment Neighbourhoods	11
3.1.3 Binary Segmentation	12
3.2 Algorithm Configuration	13
3.2.1 Penalty Scores	13
3.2.2 Distribution Assumptions	15
3.3 Evaluation Measures	15
3.3.1 F1 Score	15
3.3.2 Rand Index	16
3.3.3 Adjusted Rand Index	17
3.3.4 BCubed	17
4 Research Method	19
4.1 Evaluation Pipeline Construction	19
4.1.1 Calculation of Changepoint Locations	19
4.1.2 Calculation of Evaluation Measures	19
4.2 Measures Meta-Analysis	20
4.2.1 Experiment Listings	20
4.3 Comparison of Measures Based Upon Functional Requirements	23
4.4 Application of Algorithms to Real-World Data	23
4.4.1 Data Preparation	23
4.4.2 Execution	24
5 Research	26
5.1 Simulation Studies	26
5.1.1 Dependence on Sample Size	26
5.1.2 Impact of Data Preceding and Following a Known Change Point	27
5.1.3 Temporal Penalty	28
5.1.4 Penalisation for False Positives	29
5.1.5 Penalisation for False Negatives	30

CONTENTS

5.1.6	Number of Change Points in Fixed Length Data Stream	30
5.1.7	Number of Change Points in Variable Length Data Stream	31
5.2	Functional Requirements	32
5.3	Real-World Data Analysis	32
6	Results	35
6.1	Metric Behaviour	35
6.1.1	F1 Score	35
6.1.2	Rand Index	36
6.1.3	Adjusted Rand Index	36
6.1.4	BCubed F-Score	36
6.2	Fulfilment of Functional Requirements	37
6.3	Real-World Data Analysis	38
7	Results Discussion	39
7.1	Answers to Research Questions	39
7.2	Threats to Validity	41
7.2.1	Internal Validity	41
7.2.2	External Validity	41
7.2.3	Construct Validity	41
8	Conclusions	42
9	Future Work	44
	APPENDICES	49
A	Ground Truth Annotations	50
B	Real-World Change Point Detection Plots	51
C	Full Real-World Data Scorings	57
D	Bibliography Annotations	60

Abstract

Change point detection is a highly complex field that is hugely important for industries ranging from manufacturing and IT infrastructure fault detection, to online reputation management. There exists in this field a number of metrics that are widely used for proving the veracity of new or novel approaches. However, there is little consensus as to which metric(s) are most effective in this field. This thesis carries out research into the field of change point detection, carrying out a comparative study of the behaviours of various metrics using simulated data. An analysis using real-world data is also carried out, to investigate whether the aforementioned metrics disagree when used to rank change point detection approaches in order of accuracy.

In this thesis, it is found that many of the established metrics behave inconsistently when applied to change point detection problems, and exhibit properties that bring their usefulness and accuracy into question. It is also found that between certain metrics, there is no correlation or agreement in how algorithms are ranked according to accuracy.

Concluding, the study also shows that existing change point detection algorithms are not the most well suited methods for the use-case and requirements of the host organisation; that is, providing timely notifications of conversation volume increases to clients.

Acknowledgements

This thesis was only possible due to the contributions and assistance of a number of people. First thanks must go to Evangelos Kanoulas, the academic supervisor for this work. He always kept me on track, and was full of suggestions and insights into the data and experiments I was running. Without him, I have no doubt that at the time I am writing this, this thesis would still be in the planning stages.

Special thanks also go to Wouter Koot & Carina Dusseldorp at Buzzcapture. Wouter acted as my industry supervisor, and like Evangelos, was always on hand for support and suggestions. I very much look forward to continuing to work with and for him at Buzzcapture, following my graduation. Carina, as Head of Research at Buzzcapture, provided the data annotations that were invaluable for the smooth running of the experiments carried out on real-world data. Without her insight into the data sets that I collated, the experiments on real-world data would not have been possible to conduct in an accurate or unbiased manner.

I would also like to credit my friends Arend and Jetse, who have been excellent colleagues and friends throughout the duration of this course. Special thanks also goes to the erstwhile members of the ‘Beerability’ WhatsApp group, set up as the informal place for our course to whinge and moan about our work. It was the one place that I could go and not feel bad about my progress.

The most important thanks go to my girlfriend, Elisabeth. She is the one that put up with the sleepless nights, the whining about workload and the days I shut myself away to finish this work. Without her, I’m not sure I would have ever had the confidence to apply to do a Masters degree in the first place.

Finally, my parents. It would be remiss of me to submit this thesis without mentioning how much I appreciate the constant support (emotionally and otherwise) that they have supplied me with throughout this course. I told you I would do it, and now I have.

Chapter 1

Introduction

When asserting the veracity of an approach to a particular problem in software engineering, this is almost always achieved by utilising some sort of metric or measure. To give an example, when evaluating the quality of some new piece of software, this could be done through calculating measures such as McCabe Complexity [36] or unit test coverage.

The problem of detecting change points in a data stream or time series is no different. There exists within this field a number of metrics that may be used to prove the accuracy and effectiveness of a given approach, and these often fall into one of three categories: binary classification (e.g. F1 score), clustering (e.g. BCubed, or the Rand Index), or information retrieval (utilising scoring systems such as those utilised in the TREC 2016 Real Time Summarisation Track [48]).

Being that there are myriad ways to detect change points in a data stream, data set, or time-series, it is necessary to implement the application of some evaluation measure to back up an assertion that, for example, approach *A* is better than approach *B* for some data set. The application of these measures can, in some cases, result in more questions than answers, especially in situations where two different measures may disagree with the results of the aforementioned approaches *A* and *B*.

The purpose of this thesis is twofold: firstly, to conduct a meta analysis of evaluation measures, and how they perform in the domain of change point detection problems. Being that these measures are metrics designed for other problems that are not necessarily change point detection, it stands to reason that there are perhaps situations where families of metrics will disagree with each-other, or disagree with by-eye evaluation by domain experts. The second purpose is to evaluate the veracity of a number of existing change point detection algorithms when they are applied to data from social media.

For the purposes of transparency and to support the idea of ‘open science’, the full source code written for the experiments in this thesis, as well as the CSV files containing raw data before processing and the experiment results are made available in a Git repository hosted online. The full L^AT_EX source for this document is also made available in this repository.¹

1.1 Problem Statement

For the number of change point detection methods that exist, there is an almost equal number of ways to evaluate the results of the methods. There is little agreement between researchers on the ‘correct’ method to use, and as this research will show, there are problems that exist when utilising measures that have generally been widely accepted for this purpose.

Outside of the ‘normal’ applications of change point detection (for example, spotting the onset of ‘storm seasons’ in oceanographic data [31] or the detection of past changes in the valuation of a currency, such as Bitcoin [10]), there is also an application for change point detection as an ‘event detection’ mechanism, when applied to data such as that sourced from online social media platforms.

There exists methods of event detection in social media data utilising approaches such as term frequency counting, lexicon-based recognition and context-free-grammar algorithms. However, these

¹<https://github.com/matt-chapman/master-thesis>

approaches rely on (sometimes computationally expensive) analysis of the content of messages being posted on social media platforms (see, for example, Alvanaki et al. [2]). Being that change point detection algorithms, (and especially *online* change point detection algorithms - that is, change point detection approaches that operate on ‘streaming data’: data that has values constantly appended to it) can be utilised for the detection of past events, it stands to reason that these algorithms can also be utilised for event detection when applied to pre-computed data such as conversation volume or reach of a particular conversation. There certainly exists a requirement in the field of online reputation management to be able to inform businesses (in a timely manner) that a spike in conversation volume is occurring, and thus they may need to carry out some action to mitigate reputational damage in the case of negative sentiment. Therefore, this thesis intends to answer the following research question, from which a number of sub-questions have been formulated:

RQ Are existing metrics in the field of change point detection effective and accurate?

SQ1 In what way are existing metrics deficient when applied to change point detection problems?

SQ2 Do existing metrics agree on the ‘best’ approach when used to evaluate change point detection algorithms applied to real-world data?

SQ3 Is there a metric more suited than the others, for the purpose of evaluating change point detections according to functional requirements set forth by the host company?

SQ4 What would an ideal metric for evaluating change point detection approaches look like?

SQ5 Do metrics show that change point detection is a reasonable and effective approach for the use-case of the host organisation?

1.2 Motivation

This particular research is motivated specifically by the online reputation management sector. The business hosting this research project (Buzzcapture B.V.²) is a Dutch consultancy that provides online reputation management services to many other businesses throughout Europe. Recently acquired by Obi Group B.V.³, Buzzcapture is one of the largest and most widely engaged online reputation and webcare companies in Europe. Chief among the provided services is the BrandMonitor application, which, among other features, provides a rudimentary notification system for clients that is triggered once there is an absolute or relative increase in conversation volume (conversation volume being defined as the number of online messages or postings relevant to the client over a given time period).

Buzzcapture made a project available to students of the Universiteit van Amsterdam, wherein they would provide a method for more effectively providing these notifications, based on more than just an arbitrary threshold on conversation volume or some other computed metric. Upon accepting this project, research was carried out into the field of change detection algorithms, during which it was found that there was not a *single* accepted approach for evaluating measures.

Indeed, for every publication that described some novel change point detection algorithm, there was a slightly different approach for evaluating it, and proving the veracity of algorithm in a certain situation. Most publications made use of some sort of binary classification measure (for example, Qahtan et al. [39], Buntain, Natoli, and Zivkovic [10], and Pelecanos, Ryan, and Gatton [38]), while others had small variations on that theme, providing additional takes on the binary classification approach using methods such as *Receiver Operating Characteristic* curves (for example Fawcett and Provost [16] & Desobry, Davy, and Doncarli [13]). Additionally, there were publications that made use of clustering measures, calculated using a segmentation of the time series based on computed change points, such as that published by Matteson and James [35].

It is the intention of this thesis to not only answer the research questions set out in § 1.1, but also to provide a robust recommendation of a change detection methodology which could then eventually be implemented into the Brandmonitor tool to supply timely and relevant notifications to clients when a conversation concerning their brand exhibits a change in behaviour or otherwise ‘goes viral’.

²<http://www.buzzcapture.com>

³<https://www.obi4wan.com/>

1.3 Document Structure

This thesis document is split into a number of logical ‘chapters’, each focussing on a specific aspect of the work carried out.

Background Explanatory notes regarding the change point detection methods being utilised in this thesis (including an explanation of critical values such as penalty scores), as well as explanations of the evaluation methods being utilised in this thesis and how they are calculated.

Research Method A summary of how the research conducted in this thesis was carried out. This chapter contains a listing of the experiments that took place and explains how they were conducted.

Research A factual summary of findings for each experiment, including an answer as to which hypothesis held. This section does not contain a discussion of the results, rather preferring to concentrate on the facts as they appear.

Results A discussion of the experiment results, as well as the conclusions that could be gleaned from them.

Results Discussion A summary of answers for each of the research questions, including a brief discussion concerning any threats to validity for the project.

Conclusions A full summary of the context of this work, and the conclusions that this thesis contributes to the field.

Future Work A brief discussion of future work that could and should be carried out in order to further explore the results that this thesis contributes to the field. This includes work that can be carried out to further prove the results of the experiments carried out herein, as well as work that would build upon these results and provide additional insights and conclusions.

Chapter 2

Related Work

Change point detection is a large and complex field. This section is to provide a brief explanation of scientific literature that has been useful in the creation of this thesis.

2.1 Change Point Detection Algorithms & Their Applications

Change detection first came about as a quality control measure in manufacturing, and methods within this domain are generally referred to as *control charts*. Since the inception of approaches such as CUSUM [37] that provide the possibility for on-line evaluation of continuous data streams, change detection has grown as a field. With applications such as epidemic detection, online reputation management and infrastructure error detection, change detection is hugely useful both as an academic problem and in production systems of myriad application.

Over the years, many new approaches to change point detection have been proposed. Among these proposals are submissions from Desobry, Davy, and Doncarli [13], Kawahara and Sugiyama [25], and Downey [14], all of whom proposed novel methods of change point detection.

To give some examples of the application of change point detection algorithms, Tartakovsky and Rozovskii carried out research into their use in intrusion detection for IT systems [46], and Killick, Eckley, and Jonathan carried out a study in using change point detection algorithms as a method for identifying the onset of ‘storm seasons’ in oceanographic time series data [31].

The three main algorithms being utilised in the studies in this thesis were proposed by Killick, Fearnhead, and Eckley (Pruned Exact Linear Time [29]), Auger and Lawrence (Segment Neighbourhoods [5]) and Jackson et al. (Binary Segmentation [24]). All of these algorithms are described in more detail in § 3.

For a high level overview of change point detection algorithms, their implementations, benefits and drawbacks, the book *Detection of Abrupt Changes: Theory and Application* by Basseville and Nikiforov [7] fits this excellently.

2.2 Algorithm Accuracy Evaluation

In terms of evaluating change point detection algorithms, there have been a number of approaches. Papers such as those written by Buntain, Natoli, and Zivkovic [10] and Qahtan et al. [39] concentrate primarily on binary classification measures, while other authors such as Desobry, Davy, and Doncarli [13], Fawcett and Provost [16], and Kawahara and Sugiyama [25] utilise variations on this theme, preferring to concentrate on *Receiver Operating Characteristic* (ROC) curves to plot various confusion matrix outputs for different approaches in order to compare and contrast.

Downey published a paper utilising some interesting metrics, including mean delay before detection between algorithms, and also the probability of a false alarm occurring [14].

Work has also been carried out by Matteson and James utilising clustering measures (“A nonparametric approach for multiple change point analysis of multivariate data” [35]) - which is one of the reasons that clustering measures were chosen for this thesis as a focus.

For background information on the measures being utilised in this study, the following papers are the source material for the approaches detailed in § 3:

Rand Index “Objective Criteria for the Evaluation of Clustering Methods” by Rand [41]

Adjusted Rand Index “Comparing partitions” by Hubert and Arabie [23]

F1 Score “Machine literature searching VIII. Operational criteria for designing information retrieval systems” by Kent et al. [27]

BCubed “Entity-based cross-document coreferencing using the vector space model” by Bagga and Baldwin [6]

This is not an exhaustive list. There have been other approaches such as those published by Galeano and Peña [17], who compare and contrast two different change point detection by calculating the frequency and therefore likelihood of change point detection across several thousand different generated data sets. Additionally, for an example of case studies utilising the above algorithms, work has been published by Killick and Eckley in “changepoint: An R Package for Changepoint Analysis”, showing the application of the algorithms against real-world data [30].

A survey of evaluation methods for change point detection problems has been performed by Aminikhanghahi and Cook in “A survey of methods for time series change point detection” [4], though this work differs to this thesis in that it does not carry out a comparative evaluation of these metrics, nor does it suggest utilising clustering measures. This thesis differs considerably in this way - as it provides a comparison of measures and concludes with a summary of behaviours that these metrics exhibit in various situations, as well as including clustering measures in the experiments carried out.

Chapter 3

Background

3.1 Change Detection Algorithms

As the name suggests, change point detection methods are methods by which some *change* can be detected in some data, be it a change in mean, variance or some other measure. Change point detection also encompasses the field of anomaly or outlier detection.

This thesis work utilises three different change point detection methods, discussed forthwith.

3.1.1 Pruned Exact Linear Time

Pruned Exact Linear Time (*PELT*) is a modern change point detection method proposed by Killick, Fearnhead, and Eckley in “Optimal detection of changepoints with a linear computational cost”, in 2012 [29].

PELT is an *exact* method of computing change points, based on segmentation of data streams, in the same manner as Segment Neighbourhoods and Binary Segmentation, the two other algorithms being utilised in this work. PELT is a modification of the *Optimal Partitioning* method proposed by Jackson et al. in 2003 [24].

Jackson’s method has a computational cost of $O(n^2)$, while Killick’s PELT method improves on this with a computational complexity of $O(n)$ [29]. PELT achieves this by introducing a ‘pruning’ step in the algorithm (as the name suggests) in which as the algorithm iterates, it removes any values of τ (see Algorithm 1 for definition) that could not possibly be a minima of the operation in line 2 of Algorithm 1.

The PELT method works similarly to Segment Neighbourhood [5] and Binary Segmentation [24, 52] approaches (described in the following sections), in that it attempts to fit a model of segmentation of a given data set, minimising the cost of the segmentation to achieve an optimal distribution of segments and therefore change points.

Algorithm 1 describes a pseudocode implementation of the PELT algorithm, as described by Eckley, Fearnhead, and Killick in “Analysis of Changepoint Models” [15]

Algorithm 1: PELT Method for change point detection

Input: A set of data of the form, (y_1, y_2, \dots, y_n) where $y_i \in \mathbb{R}$.
 A measure of fit $C(\cdot)$ dependent on the data.
 A penalty constant β which does not depend on the number or location of change points.
 A constant K that satisfies $C(y_{(t+1):s}) + C(y_{(s+1):T}) + K \leq C(y_{t+1:T})$

Initialise: Let $n = \text{length of data}$ and set $F(0) = -\beta$, $cp(0) = \text{NULL}$

```

1 for  $\tau^* = 1, \dots, n$  do
2   Calculate  $F(\tau^*) = \min_{\tau \in R_{\tau^*}} [F(\tau) + C(y_{\tau+1:\tau^*}) + \beta]$ 
3   Let  $\tau^1 = \arg\left\{ \min_{\tau \in R_{\tau^*}} [F(\tau) + C(y_{\tau+1:\tau^*}) + \beta] \right\}$ 
4   Set  $cp(\tau^*) = [cp(\tau^1), \tau^1]$ 
5   Set  $R_{\tau^*+1} = \{\tau \in R_{\tau^*} \cup \{\tau^*\} : F(\tau) + C(y_{\tau+1:\tau^*}) + K \leq F(\tau^*)\}$ 
6 end

```

Output: the change points recorded in $cp(n)$

3.1.2 Segment Neighbourhoods

Proposed by Auger and Lawrence in 1989 [5], Segment Neighbourhoods (*SegNeigh*) is an example of an exact segmentation algorithm for the detection of change points in data. SegNeigh utilises dynamic programming to search the segmentation space (defined as the maximum number of change points in a given data stream) and then compute the cost function for every possible segmentation of the data. In this way, the location and number of change points can be computed exactly by taking the segmentation that returns the lowest cost function result.

Segment Neighbourhoods has a computational complexity of $O(n^2)$, significantly higher than that of PELT or Binary Segmentation. However, this additional cost in complexity is offset by improved performance when compared to Binary Segmentation, as shown by Braun, Braun, and Müller in “Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation” [9].

Algorithm 2 describes a pseudocode implementation of the SegNeigh algorithm, as described by Eckley, Fearnhead, and Killick in “Analysis of Changepoint Models” [15]

Algorithm 2: Generic Segment Neighbourhoods method for change point detection

Input: A set of data of the form, (y_1, y_2, \dots, y_n)
 A measure of fit $R(\cdot)$ dependent on the data which needs to be minimised.
 An integer, $M - 1$ specifying the maximum number of change points to find.

Initialise: Let n = length of data.
 Calculate $q_{i,j}^1 = R(y_{i:j})$ for all $i, j \in [1, n]$ such that $i < j$.

```

1 for  $m = 2, \dots, M$  do
2   for  $j \in \{1, 2, \dots, n\}$  do
3     Calculate  $q_{1,j}^m = \min_{v \in [1,j]} (q_{1,v}^{m-1} + q_{v+1,j}^1)$ 
4   end
5   Set  $\tau_{m,1}$  to be the  $v$  that minimises  $(q_{1,v}^{m-1} + q_{v+1,n}^1)$ 
6   for  $i \in \{2, 3, \dots, M\}$  do
7     Let  $\tau_{m,i}$  to be the  $v$  that minimises  $(q_{1,v}^{m-i-1} + q_{v+1, cp_{m,i-1}}^1)$ 
8   end
9 end

```

Output: For $m = 1, \dots, M$: the total measure of fit, $q_{1,n}^m$ for $m - 1$ change points and the location of the change points for that fit, $\tau_{m,1:m}$.

3.1.3 Binary Segmentation

Binary Segmentation [24, 52] (*BinSeg*) is a popular method for change point detection, widely utilised in the field. Binary segmentation is a method that recursively applies a single change point detection method. On the first iteration, if a change point is detected, the data is split around that change point (resulting in two data sets) and the change point method is run again on the two resulting data sets. This process is repeated such that many data set segments are created, and runs until no further change points are detected.

BinSeg is an *approximate* change point detection approach, and returns estimated change point locations - unlike PELT and SegNeigh - which return exact change point locations. It has the same computational cost as PELT, $O(n)$.

Algorithm 3 describes a pseudocode implementation of the BinSeg algorithm, as described by Eckley, Fearnhead, and Killick in “Analysis of Changepoint Models” [15]

Algorithm 3: Generic Binary Segmentation method for change point detection

Input: A set of data of the form, (y_1, y_2, \dots, y_n)
 A test statistic $\Lambda(\cdot)$ dependent on the data
 An estimator of change point position $\hat{\tau}(\cdot)$
 A rejection threshold C .

Initialise: Let $\mathcal{C} = \emptyset$, and $\mathcal{S} = \{[1, n]\}$

```

1 while  $\mathcal{S} \neq \emptyset$  do
2   Choose an element of  $\mathcal{S}$ , denote element as  $[s, t]$ 
3   if  $\Lambda(y_{s:t}) < C$  then
4     remove  $[s, t]$  from  $\mathcal{S}$ 
5   end
6   if  $\Lambda(y_{s:t}) \geq C$  then
7     remove  $[s, t]$  from  $\mathcal{S}$ 
8     calculate  $r = \hat{\tau}(y_{s:t}) + s - 1$ , and add  $r$  to  $\mathcal{C}$ 
9     if  $r \neq s$  then
10      add  $[s, r]$  to  $\mathcal{S}$ 
11    end
12    if  $r \neq t - 1$  then
13      add  $[r + 1, t]$  to  $\mathcal{S}$ 
14    end
15  end
16 end

```

Output: the set of change points recorded in \mathcal{C}

3.2 Algorithm Configuration

Change detection is an *unbounded* problem. Left without some system of constraint, the algorithm could theoretically run to infinity. Indeed, experiments carried out at the beginning of this research showed that one of the algorithms utilised in this research, PELT, when left unbounded, will detect every data point in the time-series as a change point. This result is *technically* correct, but not useful for our purposes. For this reason, the algorithms implement a penalty system, allowing for an optimal number of change points to be detected.

3.2.1 Penalty Scores

Penalty scores operate as a mechanism for optimising an unbounded problem such as the one being addressed here. Publications on the subject (such as “Efficient penalty search for multiple changepoint problems” by Haynes, Eckley, and Fearnhead [21]) define the problem as follows: Given time series data points y_1, \dots, y_n , the series will contain m change points such that their locations $\tau_{1:m} = (\tau_1, \dots, \tau_m)$, where $\{\tau_i \in \mathbb{Z} \mid 1 \leq \tau_i \leq n - 1\}$. τ_0 is assumed to be 0, and τ_{m+1} is assumed to be n . In this way we can state that a given change detection algorithm will split the time series into $m + 1$ segments such that segment i contains the points $y_{(\tau_{i-1}+1):\tau_i} = (y_{\tau_{i-1}+1}, \dots, y_{\tau_i})$. A *cost function* for a segmentation of the data set y_s can be defined as $C = (y_{s+1:t})$. This cost function defines a cost for a given segmentation containing points $y_{s+1:t}$. To illustrate, the cost function C , in the context of binary classification, returns a value that increases for an ‘incorrect’ classification and decreases for a ‘correct’ classification. More intuitively, C can be defined as a value that decreases when a change point is introduced into a segmentation being passed as an argument to C .

At this point it can be stated that the *constrained minimisation problem* is [Equation 3.1](#):

$$Q_m(y_{1:n}) = \min_{\tau_{1:m}} \left\{ \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i})] \right\} \quad (3.1)$$

Equation 3.1 is an equation showing the optimisation problem that change point detection presents. If the equation is taken intuitively, it shows that for a dataset with a *known* number of change points, the segmentation of the data around the change points can be effectively estimated by obtaining the segmentation that returns the minimum total cost based on the sum of cost function C .

However, the problem being solved by many change detection algorithms involves an unknown number of change points, at unknown locations. In this case we can estimate the output of Equation 3.2 to obtain the number and location of the change points:

$$\min_m \{Q_m(y_{1:n}) + f(m)\} \quad (3.2)$$

Equation 3.2 includes a *penalty function* $f(m)$ that increases proportionally with the number of change points m . Essentially, as the segmentation calculated as Q_m increases in size, so does the result of the penalty function $f(m)$. These two elements are at-odds with each other, presenting a problem that requires optimisation to correctly estimate the number and location of change points in the data set y_s . If $f(m)$ increases in a linear fashion with m , we can define the *penalised minimisation problem* in Equation 3.3:

$$Q_m(y_{1:n}, \beta) = \min_{m, \tau_{1:m}} \left\{ \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] \right\} \quad (3.3)$$

In Equation 3.3, β is the penalty function defined previously as $f(m)$.

There are a number of established approaches to calculating penalty values for unbounded problems (such as search or information retrieval for example), chiefly among which are Schwarz Information Criterion (SIC)/Bayesian Information Criterion (BIC) [43], Akaike Information Criterion (AIC) [1] and Hannan-Quinn [20]. Of these approaches, it is necessary to experiment to find the scheme that produces the ‘correct’ number of change points for a given dataset. The penalty schemes are defined as follows:

$$\text{SIC} = \ln(n)k - 2 \ln(\hat{L}) \quad (3.4)$$

$$\text{AIC} = 2k - 2 \ln(\hat{L}) \quad (3.5)$$

$$\text{HQC} = -2L_{max} + 2k \ln(\ln(n)) \quad (3.6)$$

Where n is the total number of observations, k is the number of *free parameters*¹ and \hat{L} is the maximum value of a likelihood function.

The penalty function returns a value which is represented by β in Equation 3.3, which then limits the number of possible segmentations returned by Q . In this way, varying the penalty function β can alter the results given by a change detection algorithm by increasing or decreasing the number of change points detected.

The **changeoint** package requires the provision of a penalty function for each algorithm, which has been chosen as ‘SIC’ for this research. This decision is based upon advice published in [15], in which it is stated that AIC (while popular in the field of change point detection as a penalty term) tends to asymptotically overestimate k .

The selection of an optimal penalty term for a given data set and change point detection algorithm is an open research question in itself. There exists a method, *Changepoints for a Range of Penalties* (CROPS) [21], in which a change detection algorithm (PELT, in this case) is repeatedly executed in such a fashion that allows for the plotting of the number change points detected against the penalty value used. When conducting analysis against a single, static data source (offline analysis), this method can be useful for ensuring optimal penalty term selection.

¹variables that are unable to be predicted by a given model

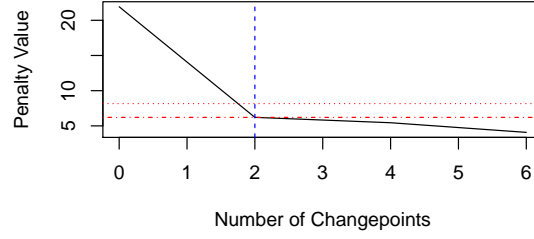


Figure 3.1: CROPS Algorithm Output

Figure 3.1 shows an example of the output of the CROPS algorithm, when executed against the ‘Rabobank’ data set used in the real-world data analysis. To read this output, the optimal penalty value should be chosen such that the plot at that point has begun to level off, showing a fast increase in the number of change points for penalty values lower than this. This example shows an optimal penalty value of about 6 (backed up by the algorithm output of a penalty value of 6.21 for 2 change points - signified by the dash-dot red line on the plot). The penalty value for the same data set computed with SIC is 8.16 (signified by the dotted red line on the plot), which also results in a total of two change points being detected. Because the PELT algorithm has a computational complexity of $O(n)$, if it is necessary to carry out an analysis such as this for a static data set, it would be easy to do so.

However, the utility of such a method for application to streaming (online) data sets is questionable.

3.2.2 Distribution Assumptions

It is also important to note that many change point detection methods make an assumption about the underlying distribution of the data being analysed. This distribution can be, for example, a normal distribution, poisson distribution or a gamma distribution. The selection of this distribution can have an effect on the ability of a given algorithm to detect change points.

To limit the scope of this research, an assumption is made that all of the data being analysed follows a normal distribution, to allow for a ‘like for like’ comparison of results. It is possible for some algorithms to use a CUSUM test statistic that makes no assumptions about data distribution, but this is not supported by all of the methods implemented in `change point`. It is also possible to specify a poisson or gamma distribution when utilising the mean/variance test statistic, but this has not been done in this research, again, in order to ensure a ‘like for like’ comparison of approaches.

3.3 Evaluation Measures

As briefly discussed in the introduction to this thesis, there are a number of pre-existing approaches for the evaluation of change detection methods. Here, the measures being evaluated are briefly explained:

3.3.1 F1 Score

This measure is utilised for testing accuracy in problems of binary classification. It considers two different measures, *precision* and *recall*, and takes a harmonic mean of the two measures to compute the final score. This measure is used in research by Qahtan et al. [39], Buntain, Natoli, and Zivkovic [10], and Pelecanos, Ryan, and Gatton [38] (among others) for the purposes of evaluating change point detection methods.

To calculate Precision and Recall, a *confusion matrix* such as that in Figure 3.2 is first constructed to provide values for the total number of true positives, false negatives, false positives and true negatives:

		Prediction Outcome		Totals
		P	n	
Actual Value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
Totals		P	N	

Figure 3.2: Confusion Matrix Example

Precision and Recall were proposed by Kent et al., in “Machine literature searching VIII. Operational criteria for designing information retrieval systems”, for the purposes of evaluating the accuracy of information retrieval methods [27].

Recall is computed as the number of correct positive results, divided by the number of positive results that should have been detected. *Precision* is computed as the number of correct positive results divided by the number of all possible positive results:

$$Precision = \frac{TP}{TP + FP} \quad (3.7)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.8)$$

The F1 score calculation (the harmonic mean of both recall and precision) can be described in general terms as in Equation 3.9 [42]:

$$F_1 = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3.9)$$

The F1 score was first proposed by Rijsbergen in his seminal work, *Information retrieval* [42].

As the F1 score is a binary classification measure, it can only be used to test the precision of an algorithm in a single domain, that is, was the change detected or not. Precision, Recall and F1 Score all provide a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$.

3.3.2 Rand Index

This measure is designed for computing the similarity between two clusters of data points. It is used for calculating the overall accuracy of a given clustering approach, when compared against a set of ground truth clusters. It was first proposed by Rand in “Objective Criteria for the Evaluation of Clustering Methods” [41]. It was used by Matteson and James in “A nonparametric approach for multiple change point analysis of multivariate data” for the purposes of evaluating change point detection approaches [35]. The Rand Index is defined as:

$$R = \frac{a + b}{a + b + c + d} \quad (3.10)$$

Given a set of data points S , partitioned (clustered) through two different methods, which shall be referred to as X and Y , the following can be defined:

- a = total number of pairs that were partitioned into the *same* subset by both X and Y
- b = total number of pairs that were partitioned into *different* subsets by both X and Y
- c = total number of pairs that were partitioned into the *same* subset by X and into a different subset by Y
- d = total number of pairs that were partitioned into the *same* subset by Y and into a different subset by X

Intuitively, it can be stated that $a+b$ is the total number of agreements between methods X and Y , while $c+d$ is the total number of disagreements. This calculation will return 0 for completely different clusters and 1 for identical clusters. The Rand Index provides a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$

3.3.3 Adjusted Rand Index

The Adjusted Rand Index is similar to the Rand Index, but adjusted to take into account the random chance of pairs being assigned to the same cluster by both approaches being compared. While the Rand Index is suited for comparing a segmentation method against a known-good *oracle* method, the adjusted index is also suited to comparing two differing approaches [35]. It was first proposed by Hubert and Arabie in ‘Comparing partitions’ [23]. It was utilised as a similarity measure for computed clusters created by segmentation of data sets by Matteson and James [35]. It is defined as:

$$R_{adjusted} = \frac{R - R_{expected}}{R_{max} - R_{expected}} \quad (3.11)$$

where $R_{expected}$ is defined as the expected Rand Index score for two completely random classifications of the given data [35] and R_{max} is defined as the maximum Rand Index value - generally 1. R is defined as the result provided by the ‘standard’ Rand Index.

The Adjusted Rand Index provides a score s such that $\{s \in \mathbb{R} \mid -1 \leq s \leq 1\}$. This is different to the score returned by the ‘basic’ Rand Index, in that it is possible for a negative value to be returned.

3.3.4 BCubed

BCubed is similar to the Adjusted Rand Index, in that it is adjusted to be appropriate for all constraints in a clustering problem. It was developed by Bagga and Baldwin in 1998 [6].

This measure was not found to be in use for the purposes of evaluating change point detection methods, but it was particularly interesting to investigate based on Matteson and James’s use of clustering metrics [35] for this purpose. It was decided that this measure would be included as an additional well-known clustering measure, to investigate whether it would perform any better than the established Rand and Adjusted Rand indices.

BCubed operates in a similar way to the F1 binary classification metric, in that it computes precision and recall values, before calculating a harmonic mean of the two metrics. BCubed also makes use of an additional *correctness* function defined as follows, where e and e' are equivalent elements in a ground truth and computed clustering respectively, L is a computed label and C is a ground truth category:

$$\text{Correctness}(e, e') = \begin{cases} 1 & \text{iff } L(e) = L(e') \leftrightarrow C(e) = C(e') \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

Equation 3.12 can be stated intuitively as returning 1 if and only if both the ground truth and computed clusterings place the element e into the same cluster, returning 0 in all other cases.

Precision and Recall can then be calculated as in Equation 3.13 and Equation 3.14.

$$\text{Precision} = \text{Avg}_e[\text{Avg}_{e'.C(e)=C(e')}[\text{Correctness}(e, e')]] \quad (3.13)$$

$$\text{Recall} = \text{Avg}_e[\text{Avg}_{e'.L(e)=L(e')}[\text{Correctness}(e, e')]] \quad (3.14)$$

To obtain the F score, a harmonic mean of Precision and Recall is taken in the same manner as the F1 score for binary classification:

$$F_{bc} = 2 \cdot \frac{1}{\frac{1}{Recall_{bc}} + \frac{1}{Precision_{bc}}} \quad (3.15)$$

The BCubed F-Score provides a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$, much like the Rand Index and F1 score for binary classification.

Chapter 4

Research Method

4.1 Evaluation Pipeline Construction

This thesis will analyse three different change detection algorithms: *Pruned Exact Linear Time* [29], *Binary Segmentation* [24] and *Segment Neighbourhoods* [5]. These will be referred to as *PELT*, *BinSeg* and *SegNeigh* respectively. The algorithms have been briefly discussed in a previous chapter (§ 3), along with the chosen critical values such as *penalty scoring* and *assumed underlying distribution*.

The algorithms are then applied to a collection of datasets falling into two categories: real-world conversation volume data taken from online social media as well as print media, TV and radio sources, and simulated data generated according to certain constraints which will also be discussed here.

The results provided by each technique will then be evaluated according to the following measures: Precision, Recall, F1 score, Rand Index, Adjusted Rand Index, BCubed Precision, BCubed Recall and BCubed F-Score. A meta-analysis will take place to evaluate the effectiveness of these methods when compared with each other and by-eye analysis from social media domain experts.

The method of evaluating the approaches will be developed using a combination of Python and R. R is a combined high-level programming language and environment created for the purpose of statistical computing [40].

Python is being utilised also due to the availability of relevant software packages for the purposes of evaluation measure calculation, as well as the author’s previous experience with the language.

4.1.1 Calculation of Changepoint Locations

changepoint is a powerful R package that provides a number of different change detection algorithms, along with various approaches to penalty values. **changepoint** offers change detection in mean, variance and combinations of the two, using the AMOC (*At Most One Change*), PELT (*Pruned Exact Linear Time*), Binary Segmentation and Segment Neighbourhood algorithms.

Changepoint was developed by Rebecca Killick and Idris A. Eckley and is provided free of charge under the GNU general public license [30].

4.1.2 Calculation of Evaluation Measures

For the calculation of Precision, Recall and F1 Score, the R package **caret** is being utilised. **caret** is an acronym for ‘Classification and Regression Training, and provides a number of tools for data manipulation, model creation and tuning, and performance measurements. Through the use of this package, it is possible to generate a *confusion matrix* based on algorithm output, and generate various performance measures based upon this. **caret** was written by Max Kuhn, and is provided free of charge under the GPL license [32].

For the calculation of the Rand Index and Adjusted Rand Index, the R package **phyclust** is being used. This package was developed by Wei-Chen Chen, for the purposes of providing a phyloclustering implementation. While this is not something being examined in this thesis, the package does provide an implementation of the Rand Index and Adjusted Rand Index metrics, which are relevant to this

research. This package is also provided free of charge under the GPL license [11].

Calculation of the BCubed precision, recall and f-score metrics is being carried out in Python, using the `python-bcubed` project. This is a small utility library for the calculation of BCubed metrics, developed by Hugo Hromic and provided under the MIT license [22]. In order to interface with this library via R, the package `rPython` is being used. This package provides functionality for running Python code and retrieving results in R, and was developed by Carlos J. Gil Bellosta. It is provided under a GPL-2 license [18].

Additionally, the `ROCR` package was utilised during the research phase of this project to generate Receiver Operating Characteristic (ROC) curves - a method used by some publications to evaluate change point detection methods. While ROC curves are not being utilised to produce the results published in this thesis, it was nonetheless an important part of the research, and provided useful insights into binary classification as a field and as a method of evaluation. `ROCR` was developed by Tobias Sing et. al. and is provided free of charge under a GPL license [45]

4.2 Measures Meta-Analysis

The experiments hereunder were developed to fulfil the requirement to answer the main research question, and a sub-question repeated here for readability:

RQ Are existing metrics in the field of change point detection effective and accurate?

SQ1 In what way are existing metrics deficient when applied to change point detection problems?

The findings of this research will also then be used to answer an additional sub-question:

SQ4 What would an ideal metric for evaluating change point detection approaches look like?

The research questions are to be answered using a series of experiments carried out using simulated data and algorithms. The experiments are designed to test the following criteria:

1. Dependence on sample size (otherwise referred to as time series length)
2. Impact of data preceding a known change point and its detection
3. Impact of data following a known change point and its detection
4. Ability to provide a temporal penalty for late, early or exact detections
5. Ability to penalise correctly for false positives
6. Ability to penalise correctly for false negatives
7. Impact of change point density in an analysed time series

4.2.1 Experiment Listings

For each experiment description, the sample size (otherwise known as time series length) is denoted by n , the true change point(s) in the time series is denoted as τ_n and any ‘detected’ change points are denoted by τ'_n . For each experiment, the changing variable (the iterator in the R source code) is denoted as i . For experiments where the computed change point(s) τ'_n are placed with some error, this error is denoted as $\Delta\tau$. $\Delta\tau$ can always be expected to adhere to $\{\Delta\tau \in \mathbb{R} \mid \Delta\tau \geq 0\}$ with the exception of Experiment 4.

For each experiment, a null hypothesis H_0 is defined, along with an alternative hypothesis, H_n .

Experiment 1: Increasing the ‘head’, variable length

This experiment involves increasing the ‘head’ (sample size prior to a single known change point) of a time series, prepending to and thus lengthening the time series. The time series contains a single known change point, and a detected change point provided by a pseudo-algorithm is placed such that $\Delta\tau = 5$.

For each prepended point, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 2.

The experiment runs such that $n = 55$ on commencement, and $n = 500$ upon completion, with each iteration adding a single data point such that $i = 445$ on completion. For each iteration, $\tau = i + 1$ and $\tau' = \tau + 5$.

Hypothesis 1 *The measures will be affected in some way by additional points before the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Null Hypothesis 1 *Neither additional points before a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

Experiment 2: Increasing the ‘tail’, variable length

This experiment involves increasing the ‘tail’ (sample size after a single known change point) of a time series, appending to and thus lengthening the time series. The time series contains a single known change point τ and a detected change point τ' provided by a pseudo-algorithm with a $\Delta\tau$ of 5.

For each appended point, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 3.

The experiment runs such that $n = 55$ on commencement, and $n = 550$ upon completion, with each iteration adding a single data point such that i runs from 2 to 500. For each iteration, $\tau = 51$ and $\tau' = 56$.

Hypothesis 2 *The measures will be affected in some way by additional points after the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Null Hypothesis 2 *Neither additional points after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

Experiment 3: Moving the true & detected change point

This experiment involves moving a known change point and a computed change point through a fixed length time series. The time series contains a single known change point τ and a detected change point τ' with $\Delta\tau = 5$, provided by a pseudo-algorithm.

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 2.

The experiment runs such that n is constant at $n = 505$. For each iteration, $\tau = i$ and $\tau' = i + 5$, for values of i from 5 to 500.

Hypothesis 3 *The measures will be affected in some way by additional points before or after the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Null Hypothesis 3 *Neither additional points before or after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

Experiment 4: Temporal Penalty Calculation

This experiment involves moving a detected change point τ' provided by a ‘pseudo-algorithm’ through a time series of fixed length, thus evaluating the ability of a measure to provide a temporal penalty for late, early, or exact detections.

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The experiment evaluates criterion 4.

The experiment runs such that n is constant at $n = 500$. For each iteration $\tau = 51$ and $\tau' = i$, where $i = 1$ at the commencement and $i = 500$ at completion, with τ' passing through τ such that $\tau = \tau'$. Thus, values of $\Delta\tau$ range from 50 to 500.

Hypothesis 4 *One or more measures will incorrectly issue scores for a detected change point occurring various distances from the true change point.*

Null Hypothesis 4 *All of the calculated metrics will correctly issue a penalty for late detections of a change point, in all instances.*

Experiment 5: Adding ‘false positive’ results

This experiment involves adding ‘false positive’ change point detections of various values of $\Delta\tau$ to a time series of fixed length with a single known change point τ .

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The Experiment evaluates criterion 4.

The experiment runs such that n is constant at $n = 500$. For each iteration $\tau = 51$ and $\tau' = i$, where $i = 51$ at the commencement of the experiment, $i = 55$ at the second iteration, and increases in increments of 5 until $i = 500$. This serves to move the detected change point through the time series, starting at the true change point and progressing to the end of the time series, ending at the final point of the time series.

Hypothesis 5 *The measures will not correctly issue penalties for false positive detections.*

Null Hypothesis 5 *All of the metrics will correctly penalise for false positive detections in a time series.*

Experiment 6: Removing ‘false negative’ results

Removing ‘false negative’ results (by way of explanation, adding correct detections) to a fixed length time series, thus increasing the number of correctly detected change points on each iteration.

The experiment begins with a time series of $n = 900$ with τ being situated at intervals of 100. As the experiment progresses, τ' points are added such that $\tau_n = \tau'_n$ for each iteration. Thus, adding a computed change point detection at the same place as a ground truth detection (a $\Delta\tau$ of 0)

This experiment evaluates criterion 6, by recalculating the scoring metrics upon each iteration.

Hypothesis 6 *The measures will not appear to correctly increase in value as false negatives are removed from the data set - showing instead inconsistent behaviour.*

Null Hypothesis 6 *Removing false negative results will result in a linear increase in score from each metric.*

Experiment 7: Adding both true and calculated change points, variable length

Adding both true change points and detected change points from a ‘pseudo algorithm’ that provides detections every 25 points to a time series, by appending to it, and thus increasing its length. The experiment runs from $n = 50$ until $n = 1000$ in increments of 50. Each τ and τ' is placed upon a new iteration such that each $\tau'_n = \tau_n + 2$, giving a $\Delta\tau$ value of 2. This experiment tests both criteria 1 and 7.

Hypothesis 7 *The measures will be affected in some way by change point density and data set length, showing an increase or decrease in score value as density and length increases.*

Null Hypothesis 7 *Metrics will be unaffected by change point density in a time series, nor will they be affected for changes in time series length.*

Experiment 8: Adding both true and calculated change points, fixed length

Adding both true change points and detected change points from a ‘pseudo algorithm’, to a time series of fixed length. The experiment runs with a static data set size of $n = 1050$, and adds two τ and τ' on each iteration, effectively ‘lifting’ a spike out of a static data set where all values are 0 at the start of the experiment. Each τ' is placed such that $\tau'_n = \tau_n + 2$ (a $\Delta\tau$ of 2, simulating a slightly late detection. This ensures that the measures do not report perfect detections throughout the experiment, thus making it impossible to spot how change point density in a fixed length data set affects the calculation of metrics.

This experiment fulfils criteria 7.

Hypothesis 8 *The measures will be affected in some way by changes in change point density. The hypothesis is that the value of the measures will increase or decrease as change point density increases.*

Null Hypothesis 8 *Metrics will be unaffected by change point density, maintaining a constant value.*

4.3 Comparison of Measures Based Upon Functional Requirements

As part of the research being carried out, a set of functional requirements are elicited from the host organisation. Based on these functional requirements the measures are evaluated and compared in such a manner that we can choose the ‘best’ measure for the use case of the host organisation. The previous research section (§ 4.2) provides the results necessary to compare the behaviour of certain metrics in certain situations, against the priorities of the host organisation.

For example, if the host organisation expresses that they require an approach with an absolute minimum of false positives, it is important to judge the application of these algorithms against real-world data, utilising measures that correctly penalise for false positive detections in a data stream.

This research is to answer the following research question:

SQ3 Is there a metric more suited than the others, for the purpose of evaluating change point detections according to functional requirements set forth by the host company?

A summary of the functional requirements gained as a part of this research can be found in § 6

4.4 Application of Algorithms to Real-World Data

4.4.1 Data Preparation

The experiment is being carried out using conversation volume data taken from Buzzcapture’s Brandmonitor application. In this context, conversation volume is defined as the number of postings pertaining to a particular topic or brand, over a given time period.

Brandmonitor harvests postings from various social media APIs (in addition to connections with print, TV and radio media) and makes them searchable through a bespoke interface as well as providing a number of useful metrics for reputation management.

In order to gain data for this experiment, Social Media Analysts from Buzzcapture were asked to provide query strings for the Brandmonitor application, showing a situation where they feel a client should have been informed of a distinct change in conversation volume. The data was then manually annotated by Buzzcapture’s Head of Research, with the points at which they believed a notification should be sent to a client due to a change in conversation volume. In this way, bias is avoided when establishing the ground truth for the data - the author of this work was not involved in this annotation process other than providing instructions for carrying out the annotation. The result of this exercise is then a set of time series’ showing the conversation volume over time for a given brand. This is accompanied by a separate data set containing indices at which changes *should* be detected - thus serving as the ground truth for this part of the experiment.

Once the query strings provided were executed, the corpus of test data included the following data sets, each pertaining to a specific brand:

- Dirk
- Dakota Access Pipeline
- Rabobank
- Ziggo
- Bol.com
- Jumbo
- Tele2
- Connexxion
- Kamer van Koophandel
- UWV

All data sets were trimmed such that the sample size equals 60 days for every set. The sets were exported from the BrandMonitor application in CSV format, with daily conversation volume totals. These CSV files, as mentioned in the introduction, are made available online.

4.4.2 Execution

To perform the experiment, an R script is executed that reads the CSV file containing the data points being analysed, and stores them in a data frame object. At this point, change point analysis is conducted using the following test statistics and algorithms:

Test Statistic	Algorithm
Mean	PELT
Mean	SegNeigh
Mean	BinSeg
Variance	PELT
Variance	SegNeigh
Variance	BinSeg
Mean & Variance	PELT
Mean & Variance	SegNeigh
Mean & Variance	BinSeg

Table 4.1: Test statistic & algorithm combinations used for real-world data analysis

Once all of the analysis methods have been executed successfully, the change points are extracted and used to compute the various scoring metrics, when the change points are compared against the established ground truth detections.

There are some assumptions made when handling this data. Firstly, for the binary classification measure, as the data being used in this study consists of daily conversation volume statistics, anything other than an *exact* detection when compared with the ground truth is considered a *failure*. Detections of a change a day after the true change point are too late for useful notifications to be sent in a production system. Detections prior to the true change point, while possibly useful in some way for predicting a future change (certainly in a production system), are also considered a *failure*. The clustering measures should not be affected by this assumption, as by their nature they should provide a more granular score for detections slightly before or after the ground truth change point.

Secondly, as discussed before, the algorithms being evaluated require some assumption to be made as to the probabilistic distribution of the data being analysed. Diagnostic histogram plots created prior to experimentation showed that the various data sets varied considerably in terms of the probability distribution of data points. It is possible for algorithms configured to use a change in mean as the trigger statistic to be configured to use a CUSUM test statistic that makes no assumption about the data distribution being analysed. Unfortunately, at the time of the experiments, the CUSUM test statistic was not supported for variance or mean/variance tests. The variance and mean/variance tests allow for the selection of other distributions such as poisson and gamma. As such, all algorithm runs were configured to assume a *normal* probability distribution. While this assumption may not hold for all of the data sets, it is necessary to take this step to ensure a like for like comparison as much as possible.

After all of the metrics are calculated and tabulated, it is possible to see which algorithm performed the ‘best’, and also carry out a comparison analysis to see which algorithms provided the most consistent results against all data sets.

This experiment is designed to answer the following research questions:

- SQ2** Do existing metrics agree on the ‘best’ approach when used to evaluate change point detection algorithms applied to real-world data?
- SQ5** Do metrics show that change point detection is a reasonable and effective approach for the use-case of the host organisation?

Chapter 5

Research

5.1 Simulation Studies

The simulation studies carried out are intended to compare each of the measures against the set of criteria described in § 4.2. The simulations are carried out according to the experiment descriptions previously explained, and the execution and results are presented here.

The scores given by ‘intermediate’ metrics used to calculate final metrics (recall and precision for both F1 Score and BCubed F-Score) are also plotted for completeness, though they will not be subject to the result discussion.

5.1.1 Dependence on Sample Size

Two of the experiments carried out in this thesis take particular note of sample size. Both involve increasing the sample size of the data, appending or prepending a number of points both prior to and following the ground truth change point. In each situation, a simulated algorithm detects a change n points after the ground truth, simulating detection that is relevant, though slightly late, to ensure that there that the baseline metric value at $t = 0$ is not equal to 1.

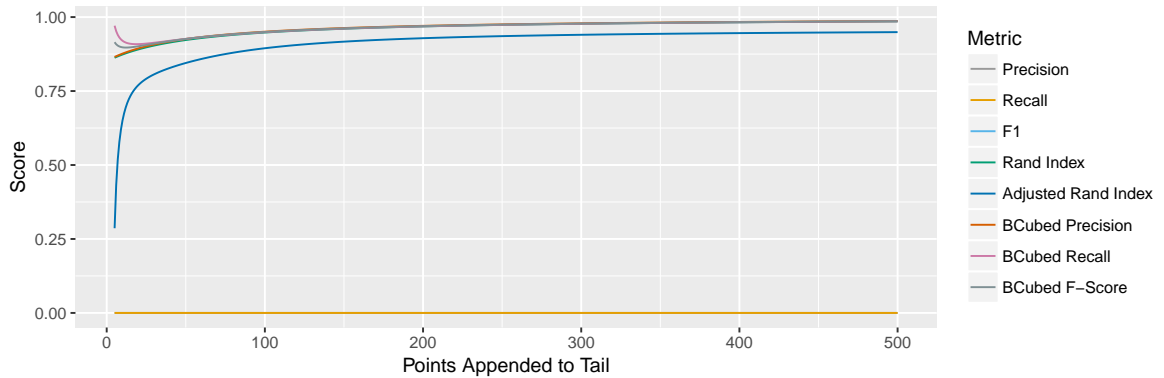


Figure 5.1: Tail-Append Results Plot

Figure 5.1 and Figure 5.2 show the results of this simulation study. Figure 5.2 demonstrates the value of metrics as the ‘head’ of the data is increased, while Figure 5.1 demonstrates the value of metrics as the ‘tail’ of the data is increased.

In this situation, we can see that both studies show a variation in metric score as the length of the data stream increases. For all of the clustering measures there is a clear increase in metric value as points are added. At $t = 0$, the clustering metrics appear to correctly penalise for a late detection, though this penalty decreases as t increases.

The Adjusted Rand Index shows a far more extreme temporal penalty for the change point detection,

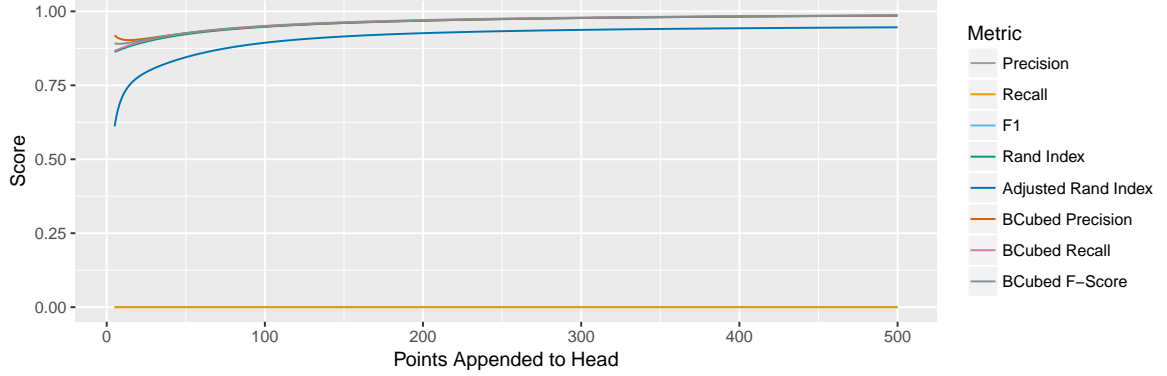


Figure 5.2: Head-Prepend Results Plot

which increases sharply at the beginning of the experiment, before maintaining a rate of increase comparable to the other clustering measures.

The binary classification measures of Precision, Recall and F1 maintain a constant value of 0 throughout the experiment, classifying the change point detection as a ‘missed’ detection and penalising accordingly. For clarity, the hypotheses are again stated here:

Hypothesis 1 *The measures will be affected in some way by additional points before the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Hypothesis 2 *The measures will be affected in some way by additional points after the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Null Hypothesis 1 *Neither additional points before a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

Null Hypothesis 2 *Neither additional points after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

For these experiments (1 & 2) the null hypothesis H_0 is rejected, and the alternative hypothesis H_1 is accepted.

5.1.2 Impact of Data Preceding and Following a Known Change Point

The experiment run for this criteria is similar to those run in § 5.1.1, but differs in one important manner: the size of the time series is maintained at a constant value throughout. As points are added to the ‘head’ of the data set, points are removed from the ‘tail’. In practice, this serves to move both the true and computed change points through the time series.

Figure 5.3 shows the results of the experiment. As with the previous experiment, the binary classification measures maintain a constant value of 0, with variation only being shown by the clustering measures. It is interesting to note that the change in all clustering measures aside from Adjusted Rand Index is minimal, with an almost constant value being held throughout the experiment, aside from when the change points are at the extreme ends of the time series.

The Adjusted Rand Index exhibits behaviour unique to itself, with a value approaching 0 as the change points approach either end of time series. From position 150 to 350, the Adjusted Rand Index holds an almost constant value.

From these results it can be inferred that it is only the Adjusted Rand Index that is effected in any meaningful way by the position of the change point in a fixed length data set. For readability, the hypotheses are again stated here:

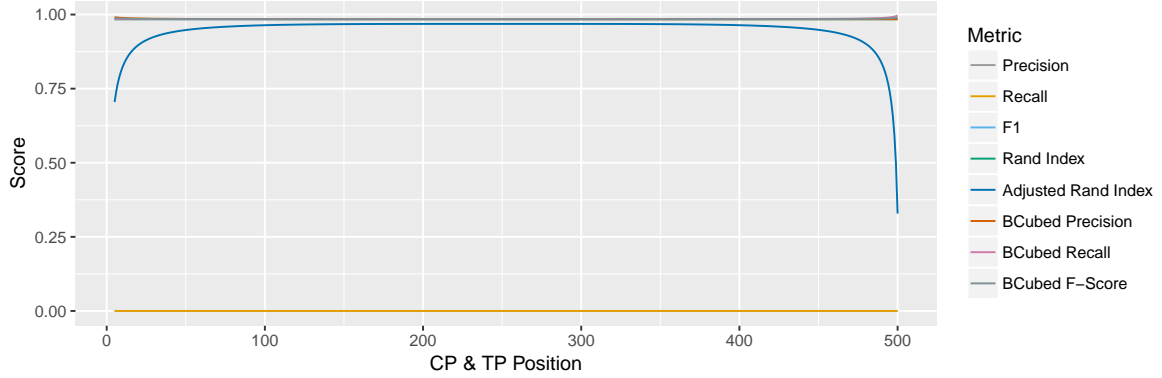


Figure 5.3: Fixed Length Append/Prepend Results Plot

Hypothesis 3 *The measures will be affected in some way by additional points before or after the change point. It is also hypothesised that data set length will have an affect on the metric value.*

Null Hypothesis 3 *Neither additional points before or after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.*

For this experiment (3), the null hypothesis H_0 fails to be rejected for several measures. The only situation in which the alternative hypothesis H_1 holds is with the Adjusted Rand Index, which shows a considerable range in values when the change point is at the extreme ends of the time series.

5.1.3 Temporal Penalty

This experiment makes use of a fixed length data set with a single, static, known change point. A computed change point is moved through the time series, beginning at the known change point, and ending at the end of the time series. Figure 5.4 shows a plot of the results of this experiment:

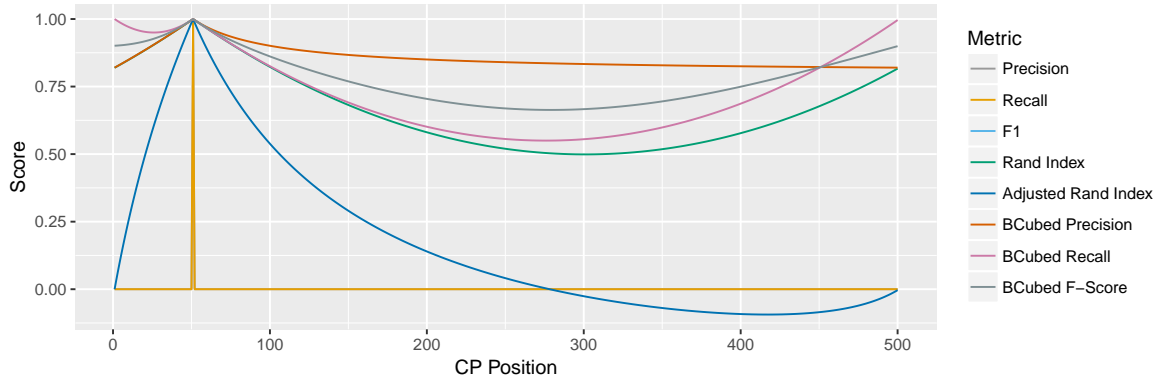


Figure 5.4: Temporal Penalty Results Plot

At the start of the experiment, when the computed change point and the true change point are positioned such that there is an early detection, all of the scores except for BCubed Recall appear to penalise for this. When the ground truth and the computed change point are equal, all of the measures return a value of 1 as expected. Once the computed change point is moved through the time series, the binary classification measures immediately return a value of 0, showing that none of the change points in the series (a single change point, in this case) have been detected.

All of the clustering measure values decrease as the computed change point moves away from the ground truth change point, though exhibit some strange behaviour as the computed change point

approaches the end of the time series. In all cases, the clustering measures show an increase in score as the computed point approaches the end of the time series, with the temporal penalty for a late detection reducing.

The Adjusted Rand Index once again exhibits behaviour unique to itself, dropping below 0 at approximately $t = 278$. This behaviour is expected from the Adjusted Rand Index, being that the adjusted nature of the metric (allowing for cluster classification occurring by chance) results in a metric capable of returning a value < 0 .

For readability the hypotheses of this experiment are repeated here:

Hypothesis 4 *One or more measures will incorrectly issue scores for a detected change point occurring various distances from the true change point.*

Null Hypothesis 4 *All of the calculated metrics will correctly issue a penalty for late detections of a change point, in all instances.*

The results of this experiment (4) cause the null hypothesis H_0 to be rejected. All of the measures show some issue with the application of a temporal penalty for late detections, and indeed also show issues with crediting algorithms for early detections. In this case, the alternative hypothesis H_1 holds.

5.1.4 Penalisation for False Positives

This experiment utilised a fixed length time series with a single ground truth change point, and a single computed change point that equals the ground truth change point. Successive false positives are added to the time series, moving from left to right across the time series.

Figure 5.5 shows the results of the experiment, with the score provided by each metric plotted against the number of false positives present in the data stream.

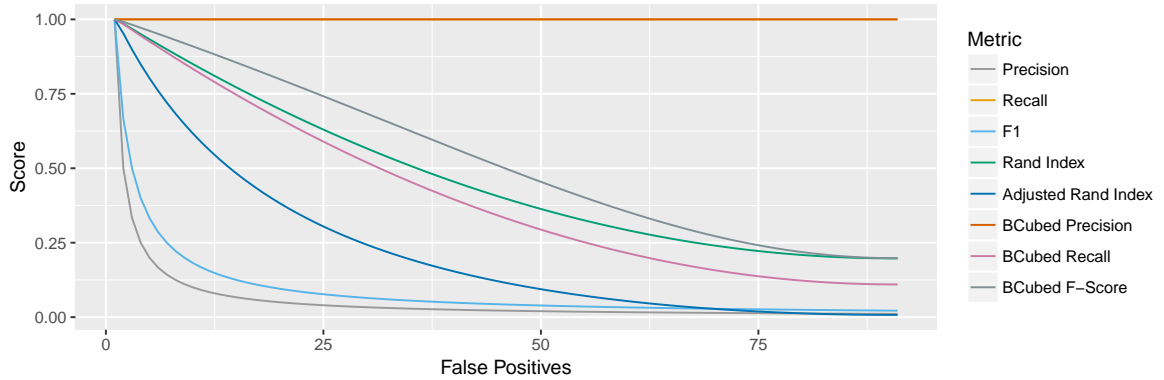


Figure 5.5: False Positives Results Plot

The results of this experiment show that almost all of the metrics behave in a similar fashion. The binary classification measure of Recall maintains a value of 1, while all of the other measures show a precipitous drop in score as the initial false positives are added. Precision and F1 scores for binary classification drop sharply, while the measures for clustering show a more gentle decrease in score. The Rand Index and BCubed F-Score end the experiment at approximately the same value, while the Adjusted Rand Index, F1 Score and Binary Classification Precision metrics also terminate at approximately the same value. For readability the hypotheses for this experiment are repeated here:

Hypothesis 5 *The measures will not correctly issue penalties for false positive detections.*

Null Hypothesis 5 *All of the metrics will correctly penalise for false positive detections in a time series.*

The results of this experiment cause the null hypothesis H_0 to hold, as all metrics behave correctly in this situation.

5.1.5 Penalisation for False Negatives

For this criteria evaluation, the experiment utilised a fixed length time series with multiple ground truth change points. As the experiment progressed, computed change points were added, equal to each of the ground truth change points. Figure 5.6 shows the results of the experiment, with the metric scores plotted against the number of false negatives present in the data stream.

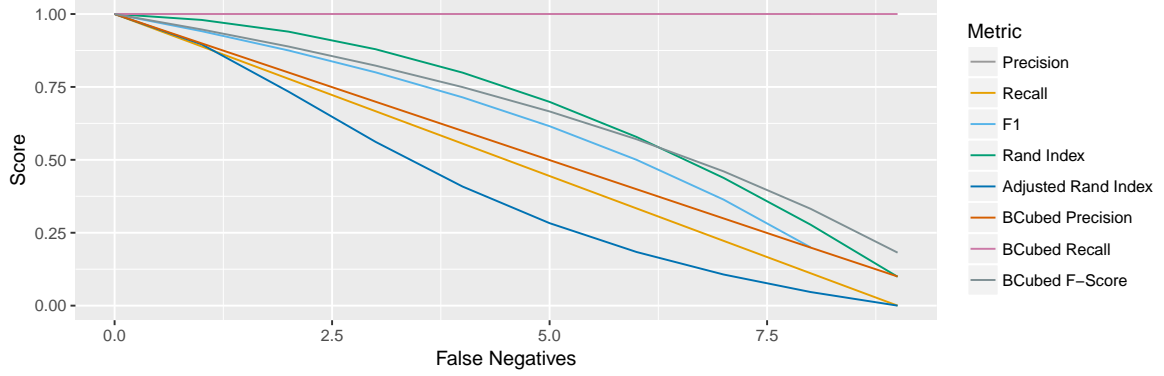


Figure 5.6: False Negatives Results Plot

The vast majority of measures see a consistent decrease in returned score as the number of false negatives increases. Indeed, once all of the ground truth change points have been correctly detected, all of the metrics return a score of 1. Only the Adjusted Rand Index returns a score of 0 after 10 false negatives are found in the data stream. As expected, BCubed Recall remained at a value of 1 for the duration of the experiment. For readability, the hypotheses for this experiment are repeated:

Hypothesis 6 *The measures will not appear to correctly increase in value as false negatives are removed from the data set - showing instead inconsistent behaviour.*

Null Hypothesis 6 *Removing false negative results will result in a linear increase in score from each metric.*

The results of this experiment cause the null hypothesis H_0 to be rejected, and the alternative hypothesis H_1 to hold.

5.1.6 Number of Change Points in Fixed Length Data Stream

This experiment adds change points to a data stream by ‘lifting’ slices of the data stream - adding a value to them to create two change points. This is carried out multiple times, each time adding two ground truth change points, and two computed change points 3 time units after the ground truth change points. Figure 5.7 shows the results of this experiment, plotting the score returned by the metric against the number of change points in the stream.

The results of this experiment show a drop in score provided by the Adjusted Rand Index and BCubed F-Score. The BCubed F-Score shows a linear rate of decline as change points are added to the stream, while the Adjusted Rand Index shows an increase in downward velocity towards the end of the experiment. All of the other metrics maintained a constant value of either 1 or 0, as expected. The hypotheses for this experiment are as follows:

Hypothesis 7 *The measures will be affected in some way by change point density and data set length, showing an increase or decrease in score value as density and length increases.*

Null Hypothesis 7 *Metrics will be unaffected by change point density in a time series, nor will they be affected for changes in time series length.*

The results of this experiment cause the alternative hypothesis H_1 to be accepted, and thus the null hypothesis H_0 is rejected.

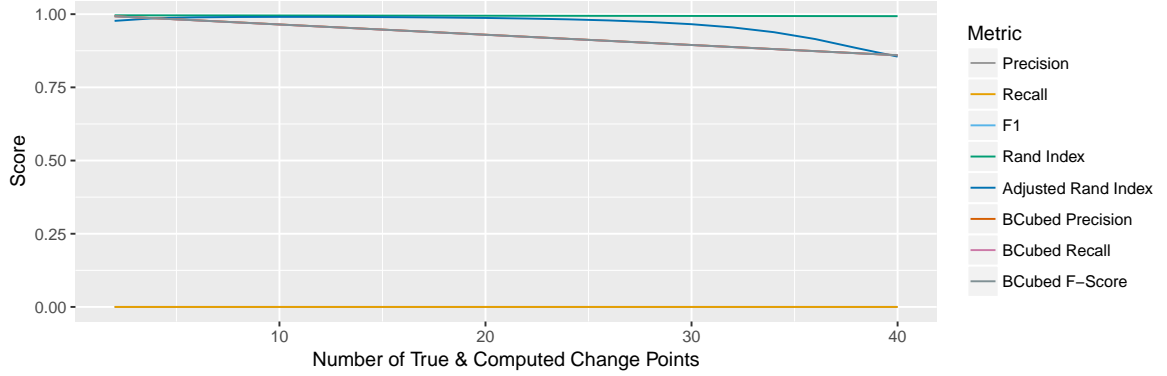


Figure 5.7: Change Point Density (Fixed Length) Results Plot

5.1.7 Number of Change Points in Variable Length Data Stream

This final experiment is similar to the previous one, in that it increases the number of change points in a data stream. However, the difference here is that in this case, data is appended to the data stream, also increasing the length of the stream as each successive change point is added. The results of this experiment are shown in Figure 5.8:

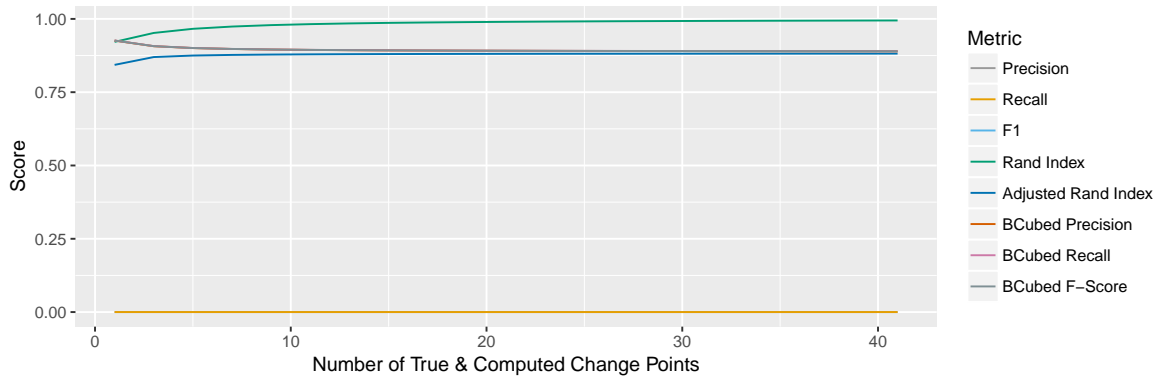


Figure 5.8: Change Point Density (Variable Length) Results Plot

The results of the experiment show some interesting behaviour in the Rand Index, Adjusted Rand Index and BCubed F-Score metrics. The Rand index and the Adjusted Rand Index increase in value as data points and change points are added to the time series, while the BCubed F-Score decreases in value. The Adjusted Rand Index begins at a slightly lower level than the Rand Index and BCubed F-Score metrics, while the Rand Index and BCubed F-Score metrics diverge almost immediately.

Aside from the large change in score value at the beginning of the experiment, the change in value for the Rand Index, Adjusted Rand Index and BCubed F-Score metrics is small. The hypotheses for this experiment are as follows:

Hypothesis 8 *The measures will be affected in some way by changes in change point density. The hypothesis is that the value of the measures will increase or decrease as change point density increases.*

Null Hypothesis 8 *Metrics will be unaffected by change point density, maintaining a constant value.*

In this experiment, the alternative hypothesis H_1 holds, allowing for the rejection of the null hypothesis H_0 .

5.2 Functional Requirements

This research was completed by carrying out a requirements elicitation process with the lead developer of the host company. Conversations and discussions were had throughout the execution of this project, and what follows is a listing of functional requirements that were gleaned as a part of this process:

- R1** The approach should not rely on arbitrary threshold values. It instead should adapt to the data that it is analysing.
- R2** False positives should not be considered wholly harmful. The consequence of a false positive in production is only that a notification is sent. That said, false positives should be kept to a reasonable level.
- R3** Late detections should be considered failure cases. Late detections are not useful to clients using the production system, as prompt detections are needed for effective damage control in situations where increased conversation volume is caused by a PR crisis.
- R4** Similarly, false negatives should also be considered failure cases. False negatives are defined as when the algorithm fails to flag a change point that would have been flagged by manual, by-eye analysis. False negatives prevent clients from acting upon changes in conversation volume in a timely manner.
- R5** The approach in it's first iteration should operate on conversation volume metrics provided by Brandmonitor, but should be extensible such that it can also be applied to other metrics.

These requirements were then used, along with the results of the simulation studies, to evaluate if one or more of the measures are best suited for evaluating the change detection methods used in the real-world data analysis study. A full summary of how the measures performed when compared with these requirements can be found in § 7.1

5.3 Real-World Data Analysis

As described in § 4.4, this experiment executed change detection algorithms against sets of real-world conversation volume data taken from online social media, print, TV and radio sources. The data was gathered and filtered to be relevant to certain brands (again, as listed in § 4.4) and then annotated with ground truth change points by social media analysts from the host organisation. The ground truth annotations can be found in Appendix A.

What follows is the results of that study. For each data set, the scores provided by the evaluation measures are tabulated and provided in Appendix C. The highest score(s) for each measure are highlighted for easy comparison and readability. Further discussion of the results and the extraction of conclusions will can be found in § 6 and § 7.1.

The final plots, including detected change points by each algorithm can also be found in Appendix B.

Analysis of the raw data used for Figure 5.9 shows that while there is a consensus on the best algorithm, some metrics provide higher scores for algorithms that perform objectively worse than others, when compared through by-eye analysis of the change point plots. Binary classification scores are universally low for all data sets.

Figure 5.9 shows a plot of average score values across all real-world data sets, from each algorithm. The error bars are drawn to show 1 standard deviation of error.

Taking the mean metric score for each algorithm across all of the experiments, we can rank the algorithms as shown in Table 5.1, using the four main metrics that this research was focussing on.

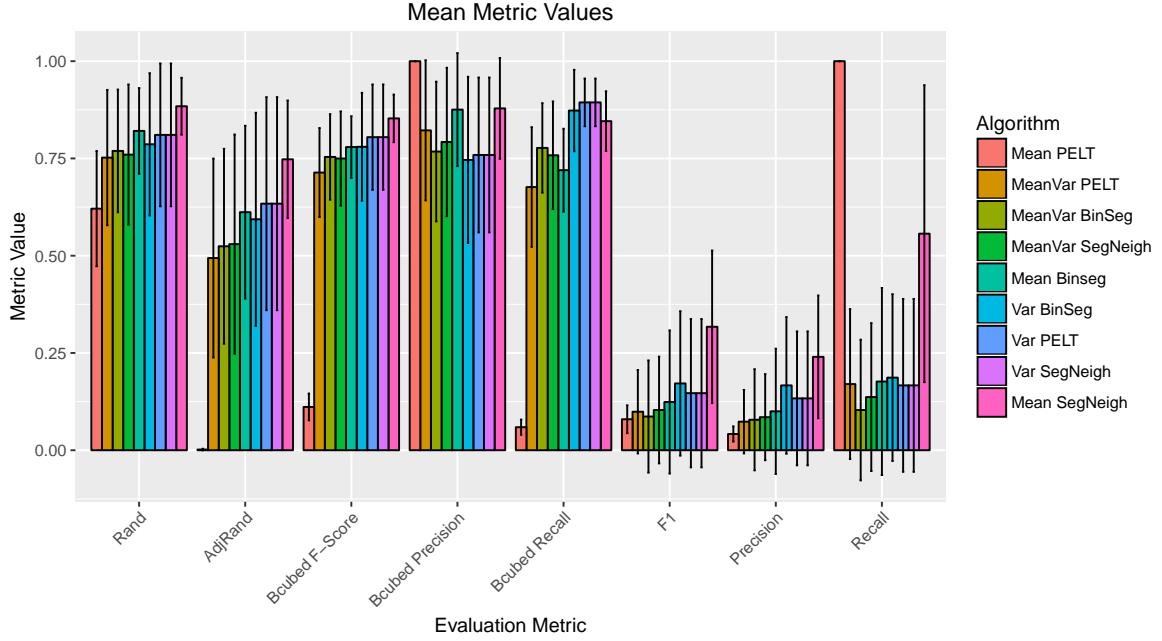


Figure 5.9: Average scores across all real-world data scenarios

Rank	F1 Score	Rand Index	Adjusted Rand Index	BCubed F-Score
1	Mean SegNeigh	Mean SegNeigh	Mean SegNeigh	Mean SegNeigh
2	Var BinSeg	Mean BinSeg	Mean BinSeg	Var PELT
3	Var PELT	Var PELT	Var PELT	Var SegNeigh
4	Var SegNeigh	Var SegNeigh	Var SegNeigh	Var BinSeg
5	Mean BinSeg	Var BinSeg	Var BinSeg	Mean BinSeg
6	MeanVar SegNeigh	MeanVar BinSeg	MeanVar BinSeg	MeanVar BinSeg
7	MeanVar PELT	MeanVar SegNeigh	MeanVar SegNeigh	MeanVar SegNeigh
8	MeanVar BinSeg	MeanVar PELT	MeanVar PELT	MeanVar PELT
9	Mean PELT	Mean PELT	Mean PELT	Mean PELT

Table 5.1: Algorithm Rankings

Here it is clear that while there appears to be a consensus on the ‘best’ algorithm in this limited trial, the remaining rankings vary somewhat. In order to analyse this, the *Kendall’s Tau* coefficient of ranking correlation is utilised [26]. This measure produces values of 1 for identical rankings, and -1 for fully different rankings. The results of this calculation are shown in Table 5.2.

With these values in Table 5.2, for each pair we can test the null hypothesis H_0 that $\tau = 0$ and there is no correlation between the rankings by computing the p -value for each τ . This gives results shown in Table 5.3.

The values shown in Table 5.3 demonstrate that for the comparison of the rankings provided for F1 and BCubed F-Score, we can reject the null hypothesis H_0 that they are uncorrelated, with a 95% confidence interval, as $p < 0.05$. Similarly, we can reject the null hypothesis for the ranks provided by the Rand Index and Adjusted Rand Index, with a confidence interval of 95%, as $p < 0.05$. This is not unexpected, as the Adjusted Rand Index is derived from the classic Rand Index, and in the case of these experiments we would not expect to see any deviation in rankings provided. The other pairs of rankings show a p value of > 0.05 and as such, the null hypothesis H_0 that these ranks are not correlated holds, with a 95% confidence interval.

	F1 Ranks	Rand Index Ranks	Adjusted Rand Index Ranks	BCubed F-Score Ranks
F1 Ranks	—	0.278	0.278	0.778
Rand Index Ranks	0.278	—	1	0.278
Adjusted Rand Index Ranks	0.278	1	—	0.278
BCubed F-Score Ranks	0.778	0.278	0.278	—

Table 5.2: Calculated Kendall’s Tau values between pairs of rankings

	F1 Ranks	Rand Index Ranks	Adjusted Rand Index Ranks	BCubed F-Score Ranks
F1 Ranks	—	0.3585	0.3585	0.002425
Rand Index Ranks	0.3585	—	5.51×10^{-6}	0.3585
Adjusted Rand Index Ranks	0.3585	5.51×10^{-6}	—	0.3585
BCubed F-Score Ranks	0.002425	0.3585	0.3585	—

Table 5.3: Calculated p values for Kendall’s Tau tests

Chapter 6

Results

6.1 Metric Behaviour

Behaviour	F1 Score	Rand Index	Adjusted Rand Index	BCubed F-Score
Credits correct detections	✓	✓	✓	✓
Penalises false negatives	✓	✓	✓	✓
Penalises false positives	✗	✓	✓	✓
Temporal penalty for late detections	✗	●	●	●
Unaffected by sample size	✓	✗	✗	✗
Unaffected by change point density	✓	✓	●	●
Unaffected by data before/after change point	✓	●	✗	●

Table 6.1: Metric Behaviour Summary

Table 6.1 shows a summary of how the metrics being tested in this study behaved when the experiments were run. Behaviours exhibited by a given metric are denoted by ✓, behaviours not exhibited as ✗, and behaviours that are exhibited with some caveats are denoted by ●.

6.1.1 F1 Score

The F1 score is a ‘traditional’ binary classification score, widely used for the evaluation of change point detection methods. Based upon the results of the experiments carried out, it is clear that this measure has some limitations in this domain.

Firstly, F1 score does not include any method for penalising a method based on late detections. Detections of a change point some number of points after the true change point may still be considered successful detections, depending on the functional requirements of the system that the method is implemented in. F1 Score returns a score of 0 for anything other than exact detections. However, the measure does properly penalise for false negatives. In a data set with two known change points, if one of these change points is successfully detected and the other not, the metric will return an F1 score of 0.5, which could be considered correct.

Secondly, F1 score does not appear to correctly penalise for false positives. In situations where a change point is detected some number of points after the true change point (so, as above, possibly a successful detection, albeit late), the metric continues to return a value of 0, even when false positives occur.

The F1 measure is, however, unaffected by changes in sample sizes in the experiments, and also exhibits no variation when the change point density in a given data set is increased or decreased.

6.1.2 Rand Index

The Rand Index is a clustering measure, calculated in these experiments by segmenting the data set around detected change points and treating all data points in each of these segmentations as a single cluster.

The Rand Index improves upon the F1 score, in that it successfully penalises for late detections, with some caveats. The Rand Index computed for a data set with a single late detection better reflects the true ‘accuracy’ of a method, but it behaves inconsistently when this detected change point is near to the beginning or end of a time series. In the experiment testing temporal penalty, the Rand Index successfully provides a decreasing value as the detected change point moves away from the true change point, but starts to increase as it approaches the end of the data set.

Taken intuitively, this means that if a method detects a change point late, the Rand Index only penalises up to a certain point. If the detection occurs towards the end of the data set, the Rand Index may return a score that is actually *higher* than that is returned if a method detects a change point late, but closer to the true change point. This is a clear limitation of the measure, and means it cannot be considered to completely fulfil this behaviour.

It is also clear, based on the results of these experiments, that the Rand Index is affected by the size of the data set, with a late detection being penalised higher in a small data set, than in a large. However, this difference is reasonably small - and it is safe to assume that in most situations, change detection methods will be compared in a like-for-like fashion, including ensuring that the test data sets are all of the same length.

6.1.3 Adjusted Rand Index

The Adjusted Rand Index is an additional clustering measure, calculated in a similar fashion to the ‘standard’ Rand Index, but takes into account the possibility that some items in a cluster may be accidentally, or randomly classified into that cluster.

The Adjusted Rand Index behaved in much the same fashion as the Rand Index in most situations, though exhibited different behaviour in more than one scenario. The Adjusted Rand Index is more affected by dataset size, which is especially apparent in the experiment where change point density is increased while also increasing the data set size.

This measure also reacted inconsistently when both the true change point and computed change point are moved through the time series - showing a large, steep drop in returned score at both the beginning and end of the data set.

Temporal Penalties were more effectively scored by the Adjusted Rand Index, however - as even though the score returned by the measure increased as the detected change point moved towards the end of the time series, the value returned by the Adjusted Rand Index did not become higher than 0 at any one point.

6.1.4 BCubed F-Score

As a clustering measure, it was expected that BCubed would behave in much the same way as the Rand and Adjusted Rand Indices. However, it does exhibit some minor differences.

The BCubed metric is less heavily affected by the parabolic curve behaviour exhibited by clustering metrics when plotting temporal penalty - though it is affected. The results returned by BCubed show less variation as the computed change point moves towards the end of the time series, but it shows the same curve with approximately the same velocity as the Rand Indices.

The BCubed F-Score also reacts differently to the Rand Index when change point density is increased with time series length, showing an almost linear drop, converging with the Adjusted Rand index at the end of the experiment, unlike the Rand Index, which remains constant.

When compared with the behaviour in a fixed length data set, the BCubed measure behaves in the opposite fashion to the Adjusted Rand Index. The scores start diverged, but quickly converge to the same value and remain almost constant towards the end of the experiment. BCubed also behaves in the opposite fashion to the ‘standard’ Rand Index, with the scores diverging from the beginning of the experiment.

6.2 Fulfilment of Functional Requirements

A summary of the functional requirements follows:

- R1** The approach should not rely on arbitrary threshold values. It instead should adapt to the data that it is analysing.
- R2** False positives should not be considered wholly harmful. The consequence of a false positive in production is only that a notification is sent. That said, false positives should be kept to a reasonable level.
- R3** Late detections should be considered failure cases. Late detections are not useful to clients using the production system, as prompt detections are needed for effective damage control in situations where increased conversation volume is caused by a PR crisis.
- R4** Similarly, false negatives should also be considered failure cases. False negatives are defined as when the algorithm fails to flag a change point that would have been flagged by manual, by-eye analysis. False negatives prevent clients from acting upon changes in conversation volume in a timely manner.
- R5** The approach in it's first iteration should operate on conversation volume metrics provided by Brandmonitor, but should be extensible such that it can also be applied to other metrics.

It is clear from the behavioural analysis of the measures, that there is not a single measure that best fits all of the functional requirement criteria for the purposes of scoring the algorithms being evaluated here. Taking each requirement in turn, the following is found:

- R1** None of the algorithms evaluated rely upon arbitrary threshold values. It is not possible in the context of this study, to choose a metric that evaluates on this criteria. That said, all of the algorithms tested fulfil this requirement regardless.
- R2** False positives were penalised harshly by almost all measures. If the measures are taken in their default forms, false positives are punished heavily when there are few of them, with the effect diminishing as their numbers increase.
- R3** This criteria is fulfilled, but with a caveat. All of the measures tested penalised for late detections, but also penalised for early detections - which intuitively should be considered a good thing in this use case.
- R4** This requirement is fulfilled by all of the major measures being evaluated. Each measure shows an *almost* linear relationship between score and the number of false negatives in a time series.
- R5** This requirement is incidental and was elicited as part of the requirements process, but does not have a bearing on how the methods were evaluated. As it stands, all of the methods can be applied to any arbitrary time-series data, where the values y_s are such that $\{y \in \mathbb{R}\}$.

6.3 Real-World Data Analysis

Intuitively, the results show that all of the measures agree that Segment Neighbourhoods utilising a mean test statistic, performed the ‘best’ in this limited study of real-world data sources.

However, it is important to note that the fact that there is no correlation between rankings (with the exception of Rand \leftrightarrow Adjusted Rand and F1 Score \leftrightarrow BCubed F-Score) suggests that this cannot be taken on face value alone, and the lack of correlation may imply that even though the 1st ranked algorithm is consistent across all rankings, it is simply not possible to state that all of the measures used agree that one measure was better than all of the others in these tests.

Further studies would need to be carried out, with a much greater sample size in terms of data sources, in order to gain a more accurate ranking of approaches.

Chapter 7

Results Discussion

7.1 Answers to Research Questions

Here, the research questions set out in § 1.1 are answered, based upon the results of the experiments.

SQ1: In what way are existing metrics deficient when applied to change point detection problems?

There are multiple ways in which the measures evaluated in this thesis are shown to be inappropriate for use for evaluating change point detection approaches.

The clustering measures evaluated (Rand, Adjusted Rand, BCubed) show that they are often affected by the size of the data set, or by the density of change points in the data set, when scoring change points detected with some error.

Clustering measures especially exhibited inconsistent behaviour when penalising for late (or early) detections. In the experiments carried out, the fact that the score *increases* as the detected change point approaches the end of the time series shows that these measures may be unreliable for providing scores for approaches in which detections are extremely late.

Binary Classification metrics performed well in simulation studies, fulfilling all but two of the criteria tested. However, when applied to real-world data and change points detected therein, they proved to be not as effective as the clustering measures.

SQ2: Do existing metrics agree on the ‘best’ approach when used to evaluate change point detection algorithms when applied to real-world data?

The rankings calculated after running the algorithms against a curated set of 10 different data sets, and the correlation coefficients calculated thereafter show that there is little agreement between the measures as to which method is the best for this particular application. The variation shown when plotting 1 standard deviation error bars also suggest that further studies would be necessary (with a much larger corpus of test data) to obtain useful data. As discussed in § 6, the lack of correlation between rankings of algorithms show that it is not possible to state (regardless of how the rankings appear) that the algorithm ranked highest by all of the measures is the ‘best’ for this particular application.

When the algorithms were applied to real-world data sets, they behaved inconsistently, to the point where no measure agrees on which approach is ‘best’ for all of the data sets evaluated. When the change point plots are examined by-eye, it is clear that the algorithms underperformed when used in this context. There were cases where the manual ground-truth annotations were *almost* correctly detected by the algorithms, but there were also many situations in which the algorithm output did not closely match the ground truth at all. Despite this, the metrics being used still scored the algorithms highly in many cases (assuming a score > 0.5 to be a ‘successful’ run).

SQ3: Is there a metric more suited than the others, for the purpose of evaluating change point detections, according to the functional requirements set forth by the host company?

When comparing the requirements against measure behaviour, the answer to this question was inconclusive. There was no single measure that fulfilled all of the functional requirement criteria for the selection of change point detection method for this domain. Indeed, it can also be stated that there is also not a single algorithm among those evaluated that fulfilled all of the functional requirements set forth by the host organisation. Concluding, the author asserts that the results of these experiments show that change point detection is perhaps not the optimal method for producing ‘smart’ notifications of changes in conversation volume, and a different approach not evaluated in this work may be more suited for this specific use-case.

SQ4: What would an ideal metric for evaluating change point detection approaches look like?

If the behaviours in § 6.1 are considered to be the ideal criteria for a measure to fulfil, then it is clear how we would expect an ‘ideal’ measure to be have. Based on experiences and evidence collected during this project, the following holds true.

An ideal measure should provide proper credit to correct detections, as well as correctly penalising for false positives and false negatives accordingly. It would be sufficient for scoring if there was a mechanism for defining the ‘relevance’ of a given change, perhaps by incorporating computation of the area under the curve for the signal around the change when devising the ground truth. This would then provide higher relevance scores for large changes, and lower relevance scores for small changes. In this way, an algorithm could be penalised in a lesser fashion for missing a detection of a less relevant change, and penalised more heavily for missing a detection of a very relevant change.

An ideal measure would be unaffected by the number of data points either side of the change. This could be achieved by taking a ‘slice’ of the signal around the ground truth change point, and computing a metric based on the location of the true change point and the detected change point within that slice. This would have the additional benefit of causing a metric to be unaffected by the data points either side of a detected or ground truth change point.

This would mean, however, that a method would have to be devised to calculate a final score that was unaffected by change point density. In situations where the ground truth is known, a process such as computing a harmonic mean of detection scores for each change point would be effective, but data with a single ‘badly’ detected change point would still receive a higher score than data with multiple change points with a small distance between the detection and true change point.

Ideally, there would be a linear relationship between the score, and the relative distance that a computed change point was discovered from the true change point. This was a property that was definitely not exhibited by the measures being evaluated in this study, and it is the believe of the author that this is one of the most vital properties for an ‘ideal’ measure for change point detection problems.

This is an area that would certainly benefit from further research - the other results of the studies carried out show that there is a need for a change point problem specific measure.

SQ5: Do metrics show that change point detection is a reasonable and effective approach for the use-case of the host organisation?

The answer to this question has proven inconclusive. There are situations in which the algorithms performed well, detecting change points close to those that constituted the ground truth - but there were also many situations in which the algorithms did not perform in the manner expected. Late, early, and missed detections were common, with algorithms occasionally showing many false positive detections throughout the data set. This poses two more questions: Was the performance of these algorithms an artefact of algorithm configuration? Or, perhaps, was the selection of penalty function not appropriate for this particular type of data? Taken at face value, the answer to the research question is ‘no’ - though taking a closer look at the data provides a more nuanced answer. In order to answer this question fully, further work would certainly need to be carried out.

7.2 Threats to Validity

It is not uncommon for an experiment or set of experiments to have some threats to validity. It is the intention of this section to detail the threats that have been identified at the time of writing.

7.2.1 Internal Validity

Internal validity is defined as the extent to which bias is avoided and ‘confounding variables’ are controlled to ensure the validity of the results.

The internal validity was well ensured during the experiments carried out in this thesis. Real-world data analysis was carried out by obtaining ground-truth annotations from an independent third-party with domain knowledge of social media data, and variables in the experiments were controlled to ensure the validity of the results. Variations of the experiments were carried out such that data-set length did not confound the tests. For example, when investigating the effect of change point density on metric values, the experiment was carried out twice. Once increasing the change point density by appending data to the data set, and once increasing the change point density by ‘lifting’ change points out of a static data set where all initial values were 0.

However, the fact that the simulation studies were designed in such a manner that the *objective* was to discover situations in which evaluation measures failed in some way, may mean that there is some element of experimenter bias in effect. This could be mitigated by carrying out additional studies (such as described in § 7.2.3) that cover more of the change point detection metric behaviour feature space, and thus also show situations in which they do behave well and/or correctly.

7.2.2 External Validity

External validity is defined as the extent to which the results of an experiment can be generalised or applied to other situations outside of the experimental setup.

In the studies carried out in this thesis, one external threat to validity is low sample size for the real-world data analysis. As 10 curated data sets were utilised for this part of the study, it can be said that there was not a large enough sample to be able to generalise the conclusions to situations outside of the scope of this thesis. This can be resolved in future work by carrying out experiments that make use of much larger and more varied data sets, in order to show that the conclusions of this thesis (with regards to the real-world data analysis and rankings therein) are also applicable outside of this work.

7.2.3 Construct Validity

Construct validity is defined as the extent to which an experiment fully tests the situations which it purports to be measuring.

In the experiments carried out in this thesis, experiments were carried out to evaluate the behaviour of certain metrics used to evaluate change point detection approaches. The conclusions of these experiments are valid, but could be improved by removing certain threats to construct validity.

If the various experiments carried out were plotted in terms of the feature space in which the problem resides, it can be stated that the methods by which behaviours were evaluated could be considered corner cases. That is, cases which demonstrate that the measures were deficient in some situations, but cases that nonetheless may not occur with a large amount of regularity in real life. This can be resolved by carrying out further experimentation with many more generated data sets with known distributions and properties that are closer to real-life examples of data, and evaluating behaviours that are more common within the feature space of change point detection algorithms and their evaluation.

Chapter 8

Conclusions

Change point detection algorithms are a widely used type of algorithm that have wide uses in various industries outside of statistics and computer science. In this thesis, several methods for evaluating these algorithms were extracted from existing work on the subject. Simulation studies were carried out to investigate the existence of situations in which these measures are flawed, when applied to change point detection measures. This question is relevant, as there is not a ‘made for measure’ metric for change point detection problems. Instead, measures for other types of problems (binary classification, clustering) are adapted and used for this purpose.

In these studies, it has been shown that several metrics that have been used to evaluate change point detection methods (for example, those used by Buntain, Natoli, and Zivkovic [10] and Matteson and James [35]) behave inconsistently in certain cases.

It was found that properties of the data being evaluated by a change point detection algorithm can have a marked effect on the ability of a metric to correctly evaluate. For example, it has been shown in the simulation studies that the length of the data-set, as well as the data preceding or following a known change point can disrupt the ability to gain insights into the accuracy of a method using the analysed metrics. The ability of multiple metrics to penalise a change point detection algorithm on the basis of late detections (or reward on the basis of early detections), has also been shown to be flawed.

While future work would be required to be able to state that the metrics analysed are *completely* unsuitable for use in this domain, the work carried out in this thesis does provide a sound basis for this further investigation to take place.

Experiments with real-world data, (carried out in a similar fashion to case-studies for the **changepoint** package [30] and the those executed by Buntain, Natoli, and Zivkovic [10]), were executed to further investigate the measures analysed in the simulation studies, and to ascertain whether or not these measures will disagree when ranking the performance of several change point detection approaches. These experiments also yield interesting results. To run these experiments, multiple algorithms were executed against curated data-sets, and their output compared to a known set of ‘ground truth’ change points.

All of the metrics appear to agree on which method performed with the most accuracy, but calculating a correlation co-efficient (in this case Kendall’s Tau [26]) shows that there is no correlation between both Rand Indices and F1 Score, and also between both Rand Indices and the BCubed F-Score. Through these calculations, it is shown that it cannot be assumed that there will be agreement between certain metrics when ranking change point detection algorithms according to accuracy. Taking into consideration the organisation’s functional requirements, none of the metrics evaluated fulfilled all of them, and thus one cannot be chosen to mathematically prove that one algorithm outperforms the others in terms of accuracy or the needs of the host organisation.

Finally, it has also been shown that change point detection algorithms are not the best approach to use for the use-case of the host organisation. There were situations in the real-world data studies that showed that the algorithms were generally ineffective at detecting changes in ‘postings’ volume for online and offline media. The algorithms evaluated often failed to detect changes in a timely manner when the change point plots in [Appendix B](#) are examined by-eye. Further work would be required to ensure that this low performance was not an artefact of algorithm configuration.

Chapter 9

Future Work

The results of this thesis open up numerous avenues for future research in the field. The validity issues discussed in § 7.2 suggest that changes to the experimental method could result in more interesting results.

It would be interesting to formulate the ‘ideal metric’ discussed in § 7.1 mathematically, and prove its veracity through simulation studies and analysis of real-world data, much like that which was carried out in this thesis. This could result in novel and useful research in the field of change point detection, especially considering the results of the simulation studies carried out in this thesis. To be able to not only state that existing measures are deficient in some ways, but also be able to suggest an *alternative* metric would be an excellent piece of additional work to carry out in the future.

Additional work should be carried out with variations on the calculation of the metrics discussed in this thesis. For example, this thesis operates under the assumption that anything other than an exact detection of a change point, matching the ground-truth for that data set is a failure case for the binary classification measures. It would be interesting and useful to carry out experiments that vary the ‘grace period’ afforded to algorithm detections (say, allowing for a detection n data points after the ground-truth to also be considered a correct detection) and allow for the detection of change points *early* - that is, not penalising (or even perhaps rewarding) early detections. Research conducted in this manner would certainly be useful for the use-case of the host organisation for this thesis.

Further research should also be undertaken using much larger data sets, and more numerous instances of these data sets. One of the main threats to validity for this project is the small sample size of the real-world data corpus, and this is something that, if resolved, would result in a considerable amount of results for analysis. Further, with a large enough test corpus, it would be possible to conduct statistical significance testing between the results, to evaluate exactly how much measures disagree with each other, if at all.

The selection of penalty functions for real-world data would also be an interesting avenue for future research. It is not reasonable for a client of the host company to have knowledge of information criterion functions, or to be able to select one in an interface for using change point detection as a notification system. Thus, it would be interesting to carry out work on a corpus of real-world data sets to determine if there is an optimal penalty function or optimal fixed value penalty for data of the kind collected by the host organisation.

Further to this, there is also a possibility to experiment with other algorithm configuration options such as minimum segment length, or the maximum number of change points to search for. Experimenting with these values was done informally as part of the knowledge gathering process for this thesis, but it would be interesting to carry out formal research in this area to evaluate if mis-configuration of these values in situations where you could not intuitively ‘know’ the correct values (in online data streaming situations, for example) would negatively affect the results, and to what degree.

While this thesis deals specifically with change point detection approaches, it would be valuable to compare the performance of these approaches with machine learning and natural language processing approaches. Analysis techniques such as sentiment analysis have been shown in the past to be good methods of detecting ‘events’ occurring on social media [2], and it may well be that those approaches are far more effective in this domain. The Text Retrieval Conference of 2016 carried out an evaluation

of real-time summarisation approaches [48], and supplied a number of scoring methods which were used as part of this evaluation. It would be useful to conduct a replication study utilising a corpus of data taken from the host company of this thesis and apply the approaches submitted to that track to it. In that way, the effectiveness of content summarisation studies could be evaluated as a method for detecting anomalous changes in subject, rather than just conversation volume.

Bibliography

- [1] Hirotugu Akaike. “A New Look at the Statistical Model Identification”. In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. ISSN: 15582523. DOI: [10.1109/TAC.1974.1100705](https://doi.org/10.1109/TAC.1974.1100705). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
- [2] Foteini Alvanaki et al. “EnBlogue: emergent topic detection in Web 2.0 streams”. In: *Proc. ACM SIGMOD International Conference on Management of Data* (2011), pp. 1271–1274. ISSN: 07308078. DOI: [10.1145/1989323.1989473](https://doi.org/10.1145/1989323.1989473). URL: <http://doi.acm.org/10.1145/1989323.1989473%7B%5C%7D5Cnpapers2:/publication/uuid/44E009D1-8574-49C1-A0AC-C2B247FE9043>.
- [3] Enrique Amigó et al. “A comparison of extrinsic clustering evaluation metrics based on formal constraints”. In: *Information Retrieval* 12.4 (2009), pp. 461–486. ISSN: 13864564. DOI: [10.1007/s10791-008-9066-8](https://doi.org/10.1007/s10791-008-9066-8).
- [4] Samaneh Aminikhanghahi and Diane J Cook. “A survey of methods for time series change point detection”. In: *Knowledge and Information Systems* 51.2 (2017), pp. 339–367. ISSN: 02193116. DOI: [10.1007/s10115-016-0987-z](https://doi.org/10.1007/s10115-016-0987-z). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5464762/pdf/nihms860556.pdf>.
- [5] Ivan E. Auger and Charles E. Lawrence. “Algorithms for the optimal identification of segment neighborhoods”. In: *Bulletin of Mathematical Biology* 51.1 (Jan. 1989), pp. 39–54. ISSN: 00928240. DOI: [10.1007/BF02458835](https://doi.org/10.1007/BF02458835). URL: <http://link.springer.com/10.1007/BF02458835>.
- [6] Amit Bagga and Breck Baldwin. “Entity-based cross-document coreferencing using the vector space model”. In: *Proceedings of the 17th international conference on Computational linguistics* -. Vol. 1. Morristown, NJ, USA: Association for Computational Linguistics, 1998, pp. 79–85. DOI: [10.3115/980451.980859](https://doi.org/10.3115/980451.980859). URL: <http://portal.acm.org/citation.cfm?doid=980451.980859>.
- [7] M. Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. 1993. ISBN: 0-13-126780-9. DOI: [10.1016/0967-0661\(94\)90196-1](https://doi.org/10.1016/0967-0661(94)90196-1). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.6896%7B%5C%7Drep=rep1%7B%5C%7Ddtype=pdf>.
- [8] S. Bersimis, S. Psarakis, and J. Panaretos. “Multivariate statistical process control charts: An overview”. In: *Quality and Reliability Engineering International* 23.5 (2007), pp. 517–543. ISSN: 07488017. DOI: [10.1002/qre.829](https://doi.org/10.1002/qre.829).
- [9] J. V. Braun, R. K. Braun, and H. G. Müller. “Multiple changepoint fitting via quasilielihood, with application to DNA sequence segmentation”. In: *Biometrika* 87.2 (June 2000), pp. 301–314. ISSN: 00063444. DOI: [10.1093/biomet/87.2.301](https://doi.org/10.1093/biomet/87.2.301). URL: <https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/87.2.301>.
- [10] Cody Buntain, Christopher Natoli, and Miroslav Zivkovic. “A Brief Comparison of Algorithms for Detecting Change Points in Data”. In: *Supercomputing*. 2014. URL: <https://github.com/cbuntain/ChangePointDetection>.
- [11] W. C. Chen. *Overlapping Codon Model, Phylogenetic Clustering, and Alternative Partial Expectation Conditional Maximization Algorithm*. 2011. URL: <http://gradworks.umi.com/34/73/3473002.html>.

- [12] Tamraparni Dasu et al. “Change (detection) you can believe in: Finding distributional shifts in data streams”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5772 LCNS (2009), pp. 21–34. ISSN: 03029743. DOI: [10.1007/978-3-642-03915-7_3](https://doi.org/10.1007/978-3-642-03915-7_3).
- [13] Frédéric Desobry, Manuel Davy, and Christian Doncarli. “An online Kernel change detection algorithm”. In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2961–2974. ISSN: 1053587X. DOI: [10.1109/TSP.2005.851098](https://doi.org/10.1109/TSP.2005.851098).
- [14] Allen B. Downey. “A novel changepoint detection algorithm”. In: *Applied Microbiology and Biotechnology* (2008), pp. 1–11. arXiv: [0812.1237](https://arxiv.org/abs/0812.1237). URL: <http://arxiv.org/abs/0812.1237>.
- [15] I.A. Eckley, P. Fearnhead, and R. Killick. “Analysis of Changepoint Models”. In: *Bayesian Time Series Models* January (2011), pp. 205–224. DOI: [10.1017/CB09780511984679.011](https://doi.org/10.1017/CB09780511984679.011).
- [16] Tom Fawcett and Foster Provost. “Activity monitoring: Noticing interesting changes in behavior”. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* 1.212 (1999), pp. 53–62. ISSN: 09258574. DOI: [10.1016/j.ecoleng.2010.11.031](https://doi.org/10.1016/j.ecoleng.2010.11.031). URL: <http://portal.acm.org/citation.cfm?id=312195>.
- [17] Pedro Galeano and Daniel Peña. “Variance Changes Detection in Multivariate Time Series”. 2004.
- [18] Carlos J Gil, Bellosta Maintainer, and Carlos J Gil Bellosta. *‘rPython’ Package Allowing R to Call Python*. 2015. URL: <https://cran.r-project.org/web/packages/rPython/rPython.pdf>.
- [19] Jeremy Ginsberg et al. “Detecting influenza epidemics using search engine query data.” In: *Nature* 457.7232 (2009), pp. 1012–4. ISSN: 1476-4687. DOI: [10.1038/nature07634](https://doi.org/10.1038/nature07634). URL: <http://www.ncbi.nlm.nih.gov/pubmed/19020500>.
- [20] E J Hannan and B G Quinn. “The Determination of the Order of an Autoregression”. In: *Journal of the Royal Statistical Society* 41.2 (1979), pp. 190–195. ISSN: 00359246. DOI: [10.2307/2985032](https://doi.org/10.2307/2985032). URL: <http://www.jstor.org/stable/2985032>.
- [21] Kaylea Haynes, Idris A. Eckley, and Paul Fearnhead. “Efficient penalty search for multiple changepoint problems”. In: (2014), pp. 1–23. arXiv: [1412.3617](https://arxiv.org/abs/1412.3617). URL: <http://arxiv.org/abs/1412.3617>.
- [22] Hugo Hromic. *python-bcubed*. 2016. URL: <https://github.com/hhromic/python-bcubed>.
- [23] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 01764268. DOI: [10.1007/BF01908075](https://doi.org/10.1007/BF01908075). URL: <http://link.springer.com/10.1007/BF01908075>.
- [24] Brad Jackson et al. “An algorithm for optimal partitioning of data on an interval”. In: *IEEE Signal Processing Letters* 12.2 (Sept. 2005), pp. 105–108. ISSN: 10709908. DOI: [10.1109/LSP.2001.838216](https://doi.org/10.1109/LSP.2001.838216). arXiv: [0309285](https://arxiv.org/abs/0309285) [math]. URL: <http://arxiv.org/abs/math/0309285%20http://dx.doi.org/10.1109/LSP.2001.838216>.
- [25] Yoshinobu Kawahara and Masashi Sugiyama. “Change-point detection in time-series data by direct density-ratio estimation”. In: *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), pp. 389–400.
- [26] M. G. Kendall. “A new measure of rank correlation”. In: *Biometrika* 30.1 (June 1938), pp. 81–93. ISSN: 0006-3444. DOI: [10.1093/biomet/30.1-2.81](https://doi.org/10.1093/biomet/30.1-2.81). URL: <https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/30.1-2.81%20http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/30.1-2.81%7B%5C%7D5Cnhttp://www.jstor.org/stable/2332226>.
- [27] Allen Kent et al. “Machine literature searching VIII. Operational criteria for designing information retrieval systems”. In: *American Documentation* 6.2 (Apr. 1955), pp. 93–101. ISSN: 0096946X. DOI: [10.1002/asi.5090060209](https://doi.org/10.1002/asi.5090060209). URL: <http://doi.wiley.com/10.1002/asi.5090060209>.

- [28] Daniel Kifer, Shai Ben-david, and Johannes Gehrke. “Detecting Change in Data Streams”. In: *Proceedings of the 30th VLDB Conference* (2004), pp. 180–191. ISSN: 00394564. DOI: [10.1016/0378-4371\(94\)90421-9](https://doi.org/10.1016/0378-4371(94)90421-9). arXiv: [9310008](https://arxiv.org/abs/9310008) [cond-mat].
- [29] R. Killick, P. Fearnhead, and I. A. Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* 107 (2011), pp. 500–1590. ISSN: 0162-1459. DOI: [10.1080/01621459.2012.737745](https://doi.org/10.1080/01621459.2012.737745). arXiv: [1101.1438](https://arxiv.org/abs/1101.1438). URL: <http://amstat.tandfonline.com/action/journalInformation?journalCode=uasa20><http://dx.doi.org/10.1080/01621459.2012.737745><http://arxiv.org/abs/1101.1438><http://dx.doi.org/10.1080/01621459.2012.737745>.
- [30] Rebecca Killick and Idris A. Eckley. “changepoint: An R Package for Changepoint Analysis”. In: *Journal of Statistical Software* 58.3 (2014), pp. 1–19. DOI: [10.18637/jss.v058.i03](https://doi.org/10.18637/jss.v058.i03). URL: <http://www.jstatsoft.org/v58/i03/>.
- [31] Rebecca Killick, Idris Eckley, and Philip Jonathan. “Efficient Detection Of Multiple Changepoints Within An Oceanographic Time Series”. In: (2011), pp. 4137–4142. URL: http://www.lancaster.ac.uk/~7B~7Djonathan/2011%7B%5C_%7DISI%7B%5C_%7DChangePoints%7B%5C_%7DTalk.pdf<http://eprints.lancs.ac.uk/56978/>.
- [32] Max Kuhn et al. *Caret: Classification and Regression Training*. 2012. URL: <https://cran.r-project.org/web/packages/caret/caret.pdf>.
- [33] Martin Kulldorff et al. “A space-time permutation scan statistic for disease outbreak detection”. In: *PLoS Medicine* 2.3 (2005), pp. 0216–0224. ISSN: 15491277. DOI: [10.1371/journal.pmed.0020059](https://doi.org/10.1371/journal.pmed.0020059).
- [34] Tze Leung Lai and Jerry Zhaolin Shan. “Efficient recursive algorithms for detection of abrupt changes in signals and control systems”. In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 952–966. ISSN: 00189286. DOI: [10.1109/9.763211](https://doi.org/10.1109/9.763211).
- [35] David S. Matteson and Nicholas A. James. “A nonparametric approach for multiple change point analysis of multivariate data”. In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI: [10.1080/01621459.2013.849605](https://doi.org/10.1080/01621459.2013.849605). arXiv: [1306.4933](https://arxiv.org/abs/1306.4933).
- [36] Thomas J. McCabe. “A Complexity Measure”. In: *IEEE Trans. Softw. Eng.* 2.4 (1976), pp. 308–320. URL: <http://juacompe.mrchoke.com/natty/thesis/FrameworkComparison/A%20complexity%20measure.pdf>.
- [37] E. S. Page. “Continuous Inspection Schemes”. In: *Biometrika* 41.1/2 (June 1954), p. 100. ISSN: 00063444. DOI: [10.2307/2333009](https://doi.org/10.2307/2333009). URL: <http://www.jstor.org/stable/2333009?origin=crossref>.
- [38] Anita M. Pelecanos, Peter A. Ryan, and Michelle L. Gatton. “Outbreak detection algorithms for seasonal disease data: a case study using Ross River virus disease.” In: *BMC medical informatics and decision making* 10.1 (2010), p. 74. ISSN: 1472-6947. DOI: [10.1186/1472-6947-10-74](https://doi.org/10.1186/1472-6947-10-74). URL: <http://www.biomedcentral.com/1472-6947/10/74>.
- [39] Abdulhakim A. Qahtan et al. “A PCA-Based Change Detection Framework for Multidimensional Data Streams”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pp. 935–944. DOI: [10.1145/2783258.2783359](https://doi.org/10.1145/2783258.2783359). URL: <http://dl.acm.org/citation.cfm?id=2783258.2783359>.
- [40] R Core Development Team. *R: a language and environment for statistical computing*, 3.2.1. R Foundation for Statistical Computing. Vienna, Austria, 2015. ISBN: 3-900051-07-0. DOI: [10.1017/CB09781107415324.004](https://doi.org/10.1017/CB09781107415324.004). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3). URL: <https://www.r-project.org/>.
- [41] William M. Rand. “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the American Statistical Association* 66.336 (Dec. 1971), pp. 846–850. ISSN: 0162-1459. DOI: [10.1080/01621459.1971.10482356](https://doi.org/10.1080/01621459.1971.10482356). URL: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356>.
- [42] C. J. van Rijsbergen. *Information retrieval*. Butterworths, 1979, p. 208. ISBN: 0408709294. URL: <https://dl.acm.org/citation.cfm?id=539927>.

- [43] Gideon Schwarz. “Estimating the Dimension of a Model”. In: *The Annals of Statistics* 6.2 (1978), pp. 461–464. ISSN: 0090-5364. DOI: [10.1214/aos/1176344136](https://doi.org/10.1214/aos/1176344136). URL: <http://projecteuclid.org/euclid.aos/1176344136>.
- [44] D. Siegmund and E. S. Venkatraman. “Using the generalized likelihood ratio statistic for sequential detection of a change-point”. In: *The Annals of Statistics* 23.1 (1995), pp. 255–271. ISSN: 0090-5364. DOI: [10.1214/aos/1176324466](https://doi.org/10.1214/aos/1176324466).
- [45] T. Sing et al. “ROCR: visualizing classifier performance in R”. In: *Bioinformatics* 21.20 (Oct. 2005), pp. 3940–3941. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/bti623](https://doi.org/10.1093/bioinformatics/bti623). URL: <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti623>.
- [46] Alexander G. Tartakovsky and Boris L. Rozovskii. “A Nonparametric Multichart CUSUM Test for Rapid Intrusion Detection”. In: *Proceedings of Joint Statistical Meetings* (2005), pp. 7–11.
- [47] Alexander G. Tartakovsky et al. “Detection of intrusions in information systems by sequential change-point methods”. In: *Statistical Methodology* 3.3 (2006), pp. 252–293. ISSN: 15723127. DOI: [10.1016/j.stamet.2005.05.003](https://doi.org/10.1016/j.stamet.2005.05.003).
- [48] Text Retrieval Conference 2016. *TREC 2016 Evaluation Guidelines*. URL: <https://treccrts.github.io/TREC2016-RTS-guidelines.html> (visited on 07/06/2017).
- [49] Dang-Hoan Tran, Mohamed Medhat Gaber, and Kai-Uwe Sattler. “Change Detection in Streaming Data in the Era of Big Data: Models and Issues”. In: *ACM SIGKDD Explorations Newsletter - Special issue on big data* 1 (2014), pp. 30–38. ISSN: 1931-0145. DOI: [10.1145/2674026.2674031](https://doi.org/10.1145/2674026.2674031).
- [50] Alan S. Willsky and Harold L. Jones. “A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems”. In: *IEEE Transactions on Automatic Control* 21.1 (1976), pp. 108–112. ISSN: 15582523. DOI: [10.1109/TAC.1976.1101146](https://doi.org/10.1109/TAC.1976.1101146).
- [51] Yi Xu et al. “Pattern change discovery between high dimensional data sets”. In: *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11* (2011), p. 1097. DOI: [10.1145/2063576.2063735](https://doi.org/10.1145/2063576.2063735). URL: <http://dl.acm.org/citation.cfm?doid=2063576.2063735>.
- [52] Yi-Ching Yao. “Estimation of a Noisy Discrete-Time Step Function: Bayes and Empirical Bayes Approaches”. In: *The Annals of Statistics* 12.4 (Dec. 1984), pp. 1434–1447. ISSN: 0090-5364. DOI: [10.1214/aos/1176346802](https://doi.org/10.1214/aos/1176346802). URL: <http://projecteuclid.org/euclid.aos/1176346802>.

Appendix A

Ground Truth Annotations

What follows is a graphical visualisation of the *ground truth* change points as annotated. Each ground truth change point is shown as a vertical dashed blue line.

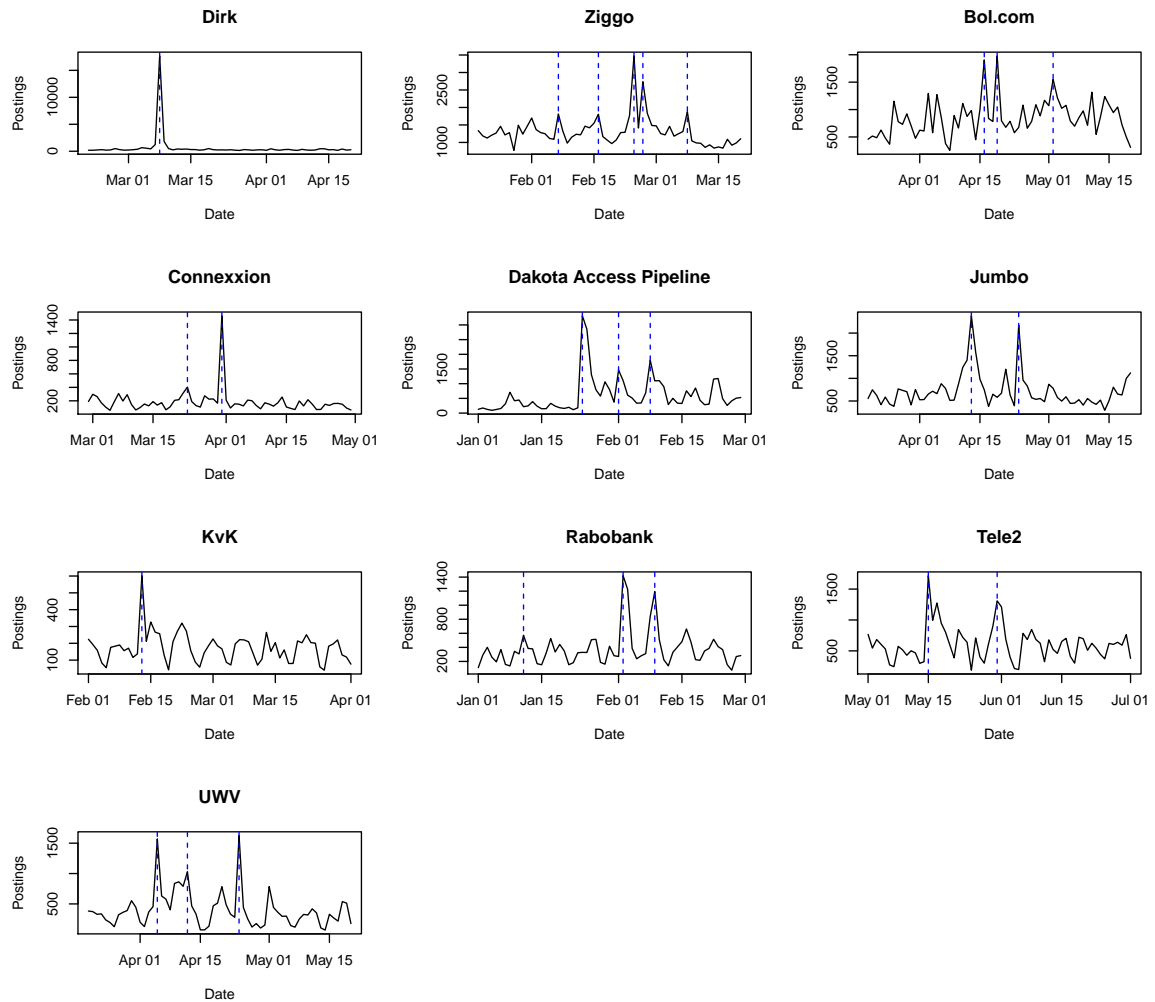


Figure A.1: Unadjusted Ground Truth Plots

Appendix B

Real-World Change Point Detection Plots

These plots show a graphical visualisation of change point detections when change point detection algorithms are applied to real-world data sets. Algorithms detecting changes in *mean* show horizontal red bars at the current mean between change points, while algorithms detecting changes in *variance* or *mean & variance* show the change points as vertical red bars. The *ground truth* is indicated by a vertical dashed blue line.

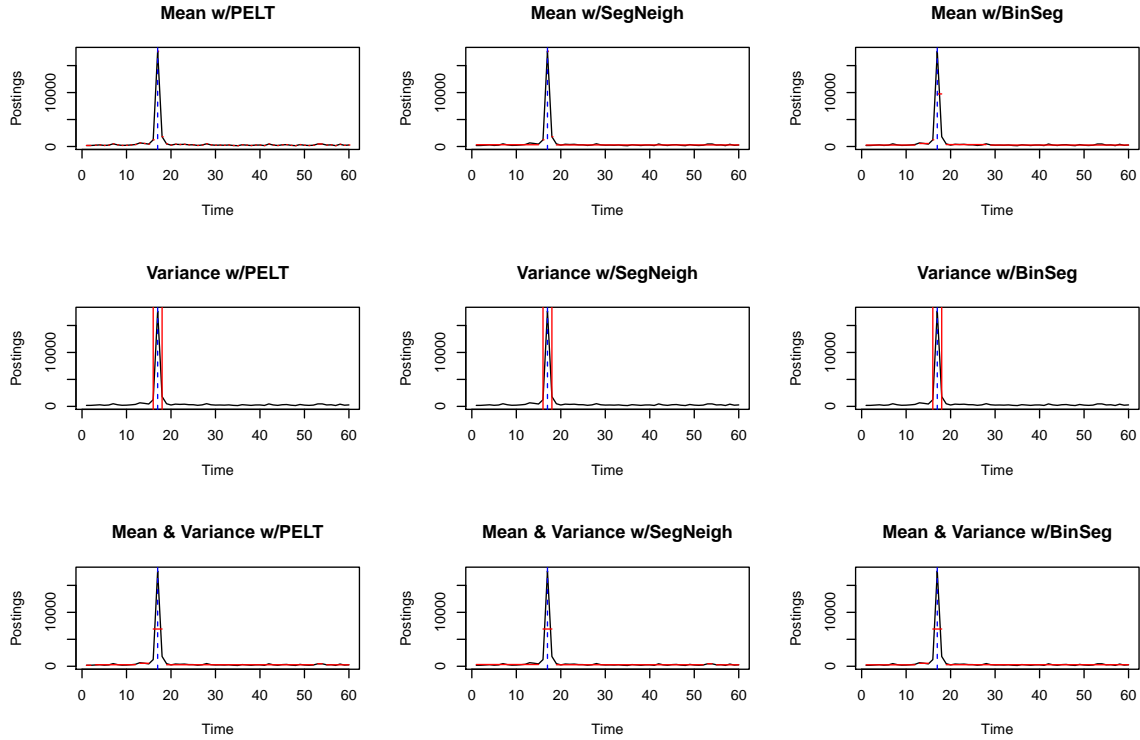


Figure B.1: ‘Dirk’ Change Point Detections

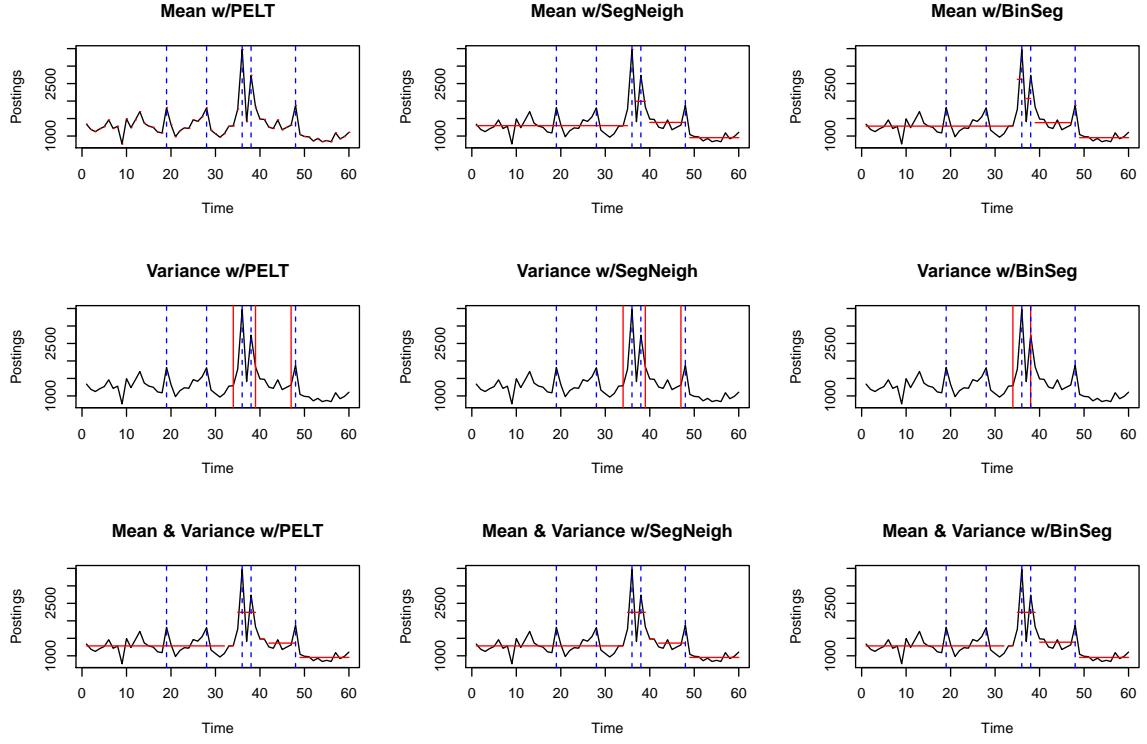


Figure B.2: 'Ziggo' Change Point Detections

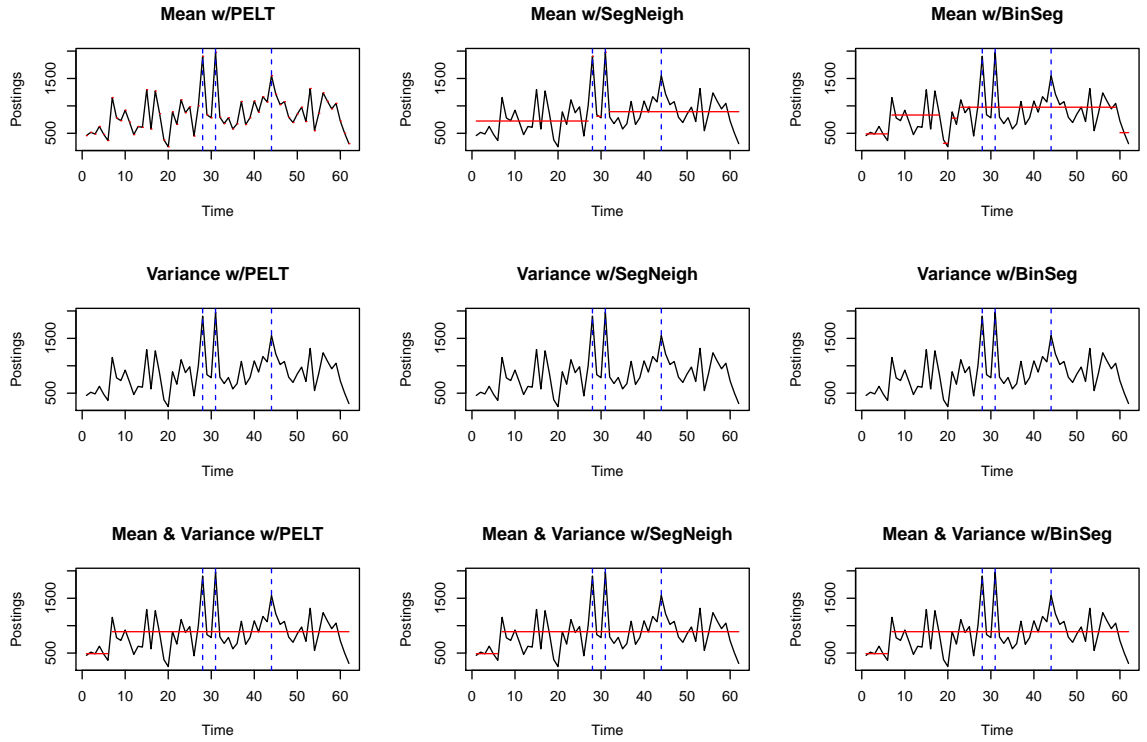


Figure B.3: 'Bol.com' Change Point Detections

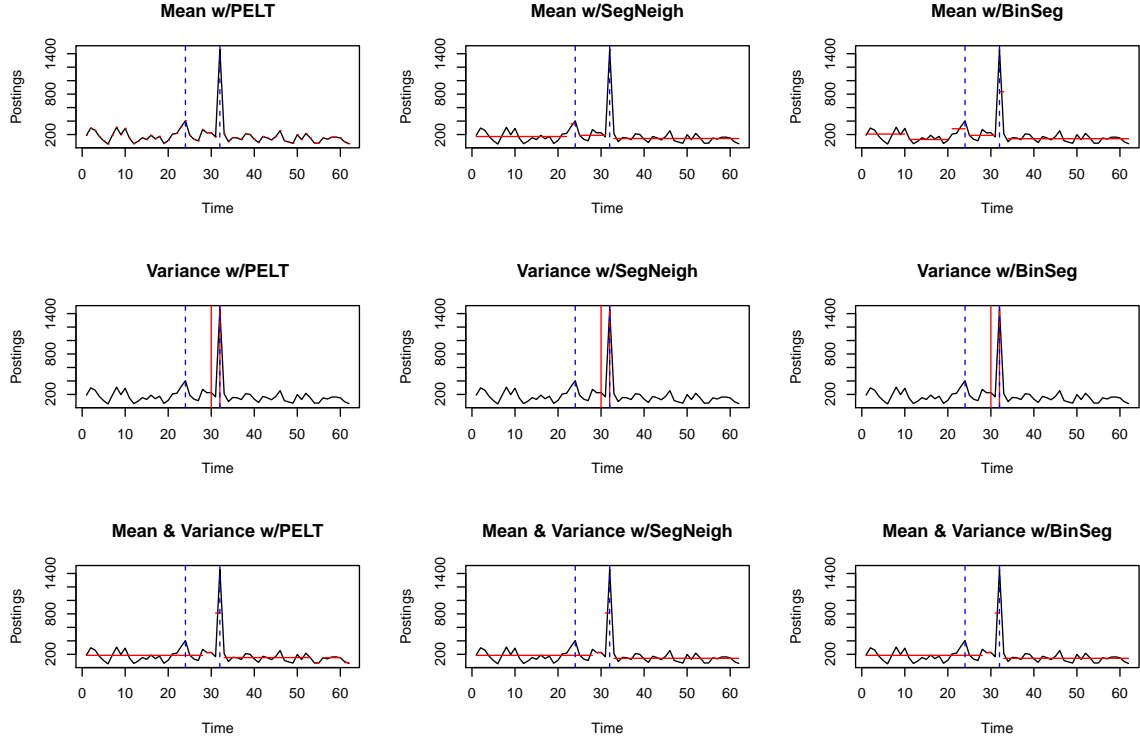


Figure B.4: 'Connexxion' Change Point Detections

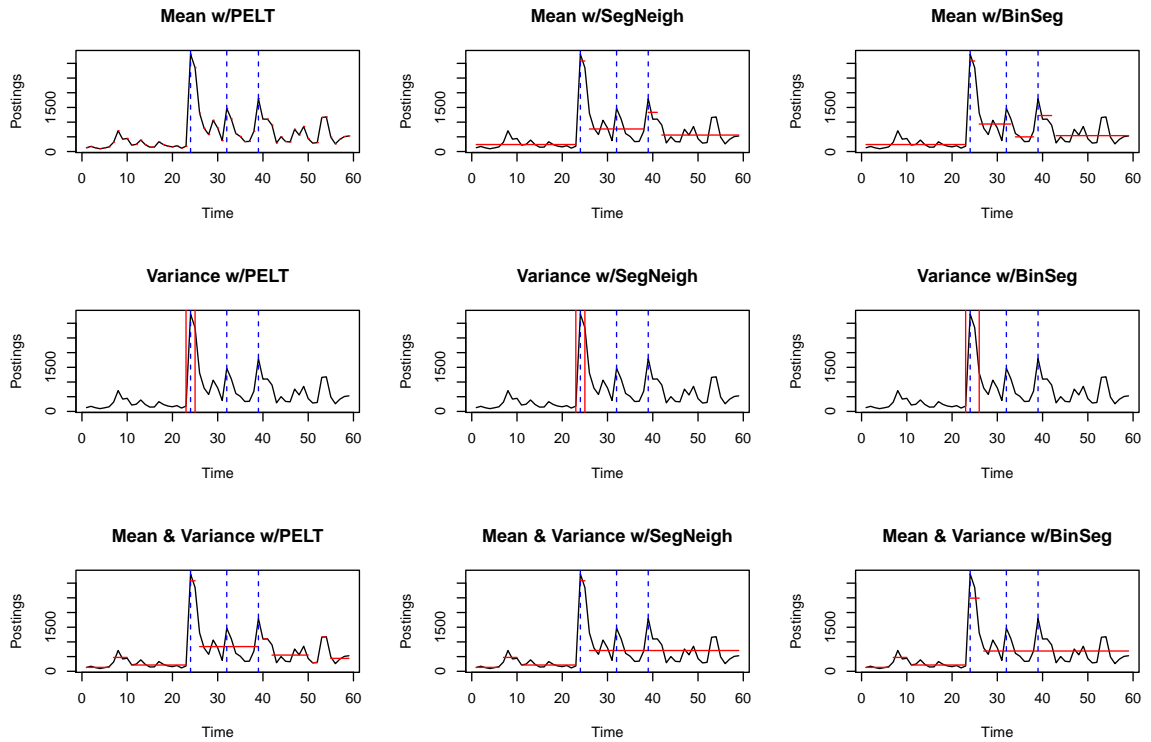


Figure B.5: 'Dakota Access Pipeline' Change Point Detections

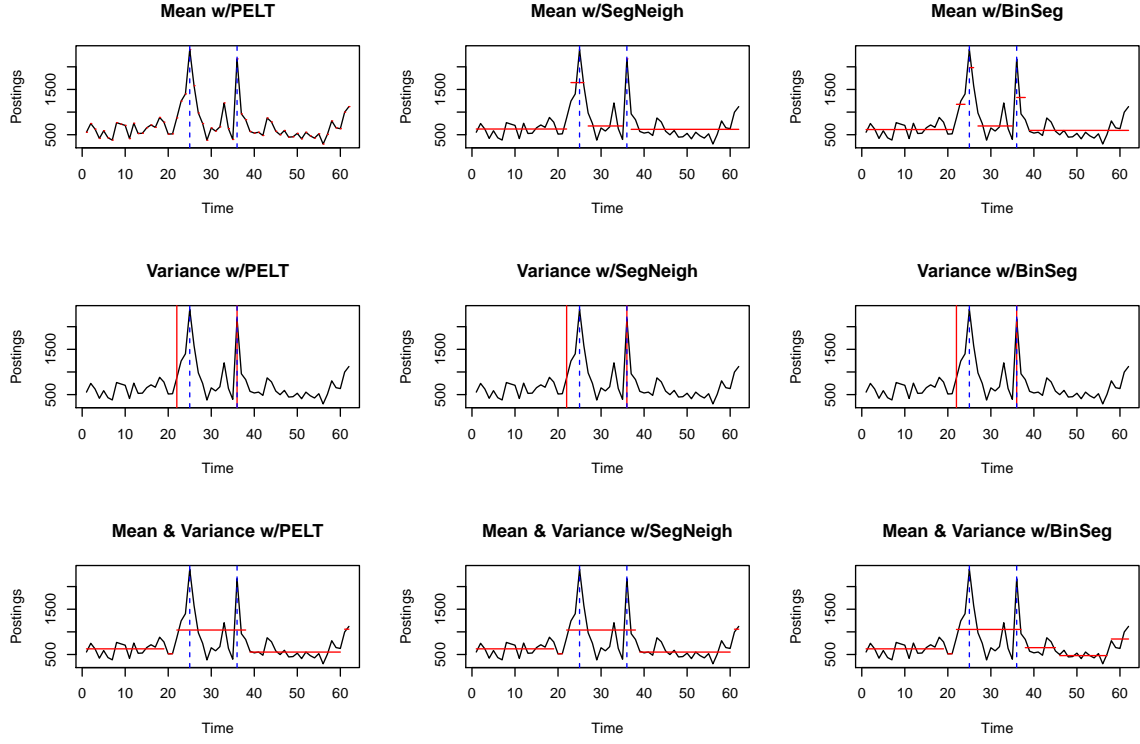


Figure B.6: 'Jumbo' Change Point Detections

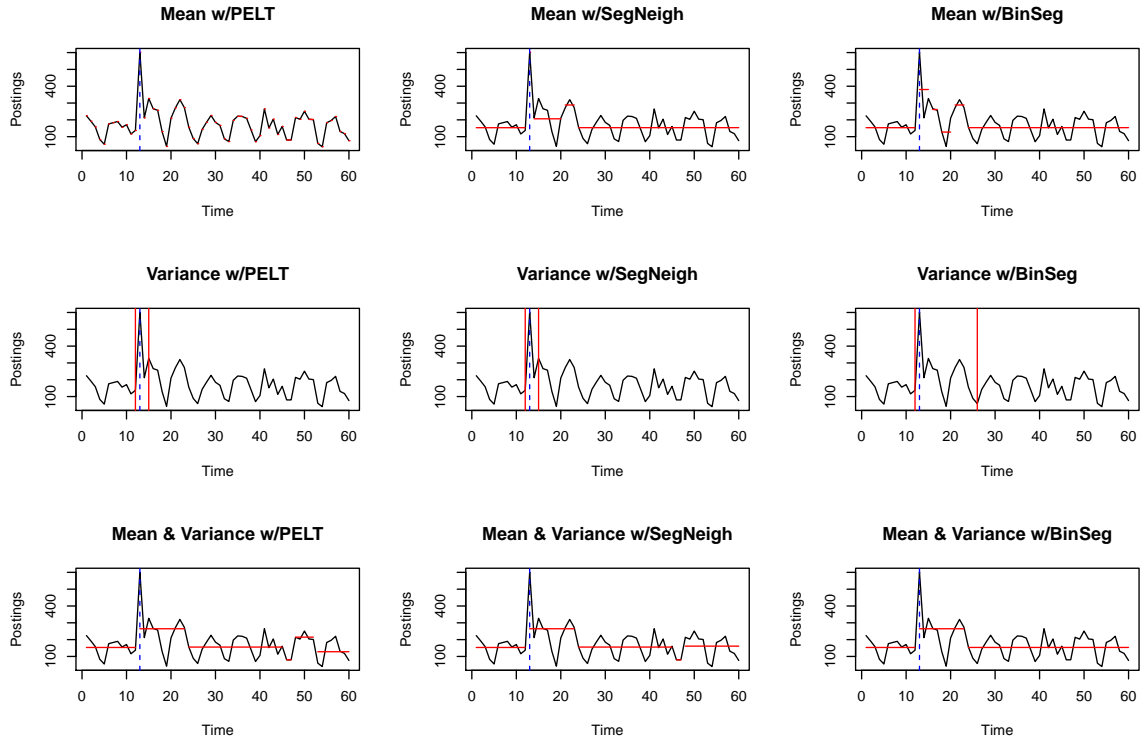


Figure B.7: 'Kamer van Koophandel' Change Point Detections

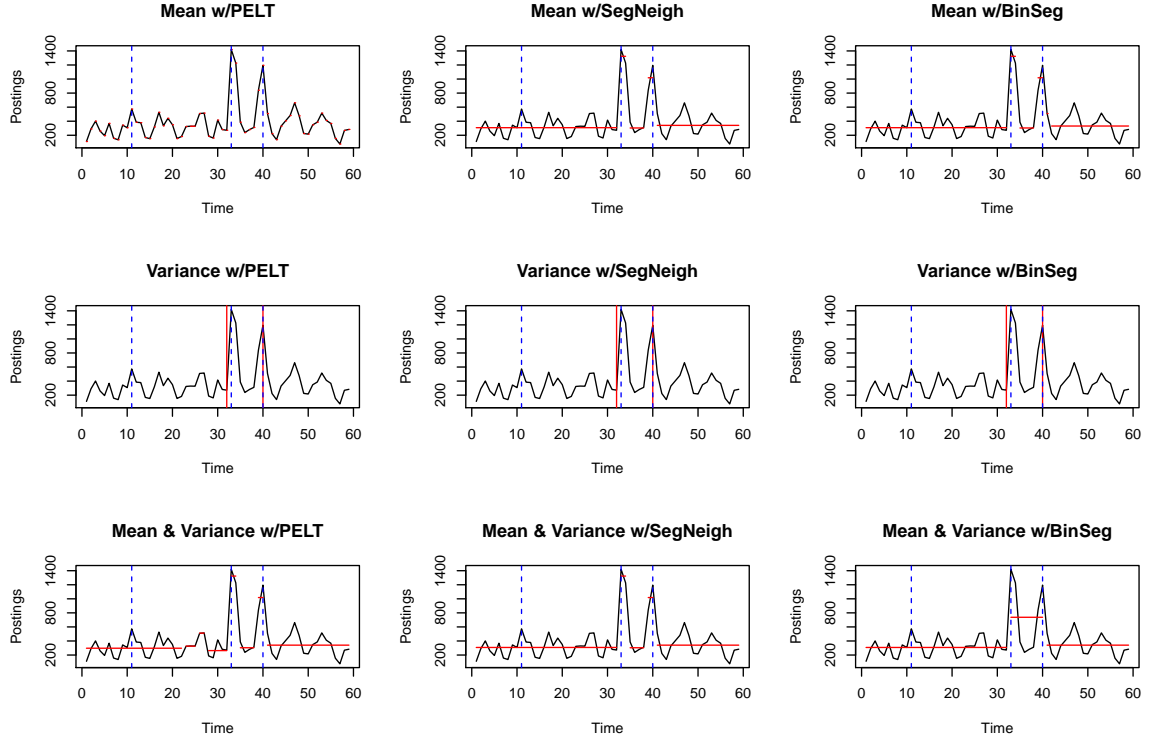


Figure B.8: 'Rabobank' Change Point Detections

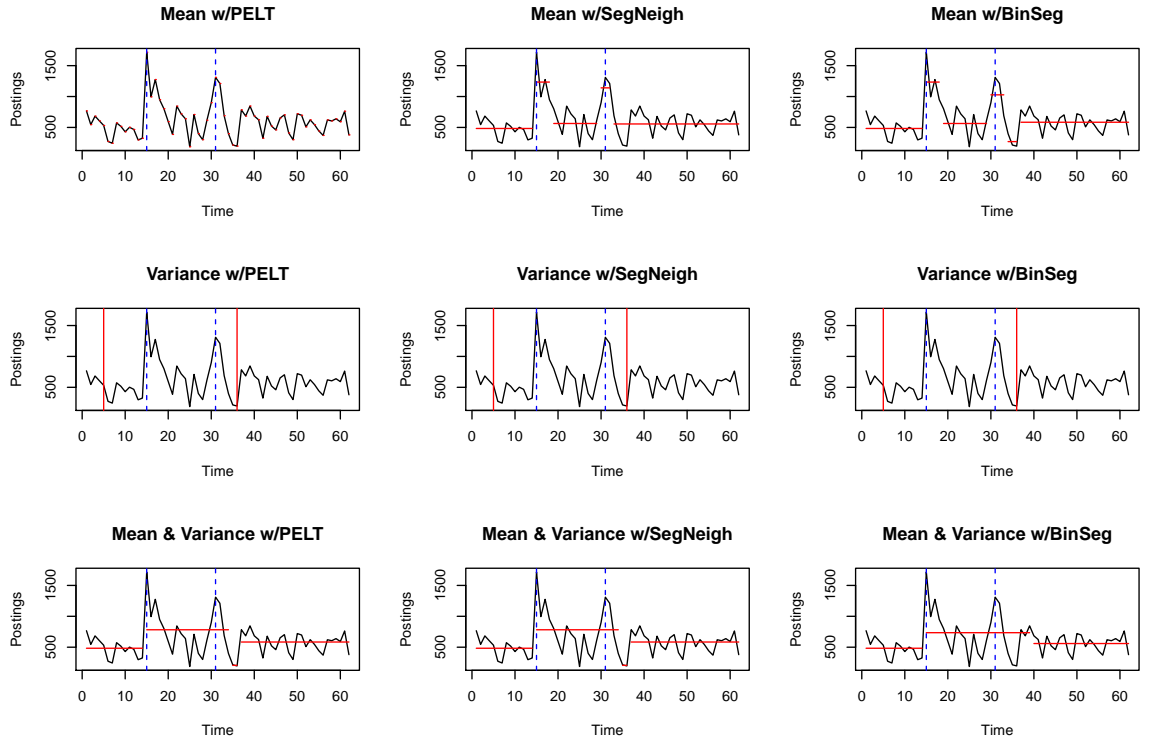


Figure B.9: 'Tele2' Change Point Detections

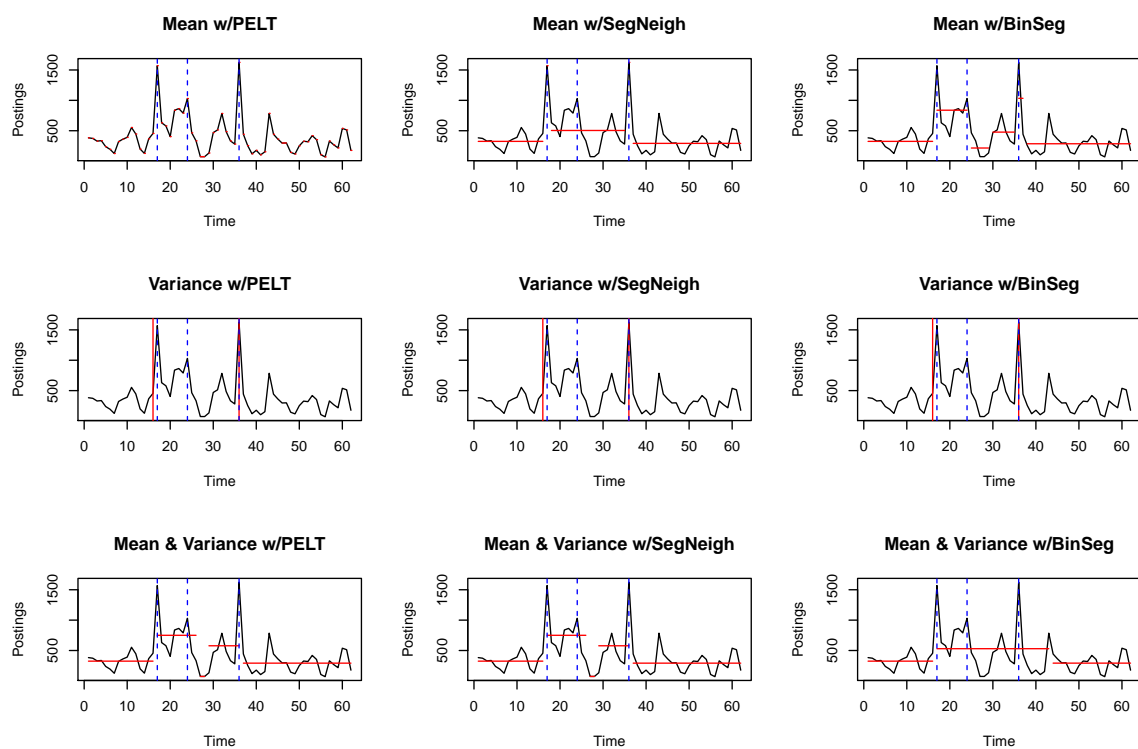


Figure B.10: 'UWV' Change Point Detections

Appendix C

Full Real-World Data Scorings

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.916384181	0.831326108	0.977777778	0.808522727	0.885131649
MeanVar SegNeigh	0	0	NA	0.774576271	0.57039828	0.977777778	0.662121212	0.789569859
MeanVar PELT	0	0	NA	0.750282486	0.528564301	0.977777778	0.572537879	0.722194622
Var BinSeg	0	0	NA	0.942937853	0.883671294	0.983333333	0.904356061	0.942192569
Var SegNeigh	0	0	NA	0.942937853	0.883671294	0.983333333	0.904356061	0.942192569
Var PELT	0	0	NA	0.942937853	0.883671294	0.983333333	0.904356061	0.942192569
Mean BinSeg	0	0	NA	0.73559322	0.503863154	0.983333333	0.564015152	0.716858425
Mean SegNeigh	0.2	1	0.333333333	0.93559322	0.8691606	1	0.875189394	0.933441067
Mean PELT	0.017241379	1	0.033898305	0.398305085	0.000746372	1	0.034090909	0.065934066

Table C.1: Scores for ‘Dirk’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0.2	0.2	0.2	0.819774011	0.54185782	0.656344086	0.855897436	0.742954366
MeanVar SegNeigh	0.2	0.2	0.2	0.785875706	0.477814329	0.619393939	0.842564103	0.713945385
MeanVar PELT	0.166666667	0.2	0.181818182	0.811864407	0.515766487	0.656344086	0.809230769	0.724812967
Var BinSeg	0.333333333	0.2	0.25	0.733333333	0.418172701	0.457575758	0.919230769	0.611004825
Var SegNeigh	0	0	NA	0.782485876	0.477512609	0.588624709	0.862564103	0.699738779
Var PELT	0	0	NA	0.782485876	0.477512609	0.588624709	0.862564103	0.699738779
Mean BinSeg	0.5	0.6	0.545454545	0.798305085	0.510612723	0.672727273	0.889230769	0.765974212
Mean SegNeigh	0.4	0.4	0.4	0.784745763	0.492259321	0.633660131	0.910064103	0.747117038
Mean PELT	0.083333333	1	0.153846154	0.807344633	0	1	0.1	0.181818182

Table C.2: Scores for ‘Ziggo’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
MeanVar SegNeigh	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
MeanVar PELT	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
Var BinSeg	0	0	NA	0.332099418	0.002293403	0.341618191	0.969439728	0.505207652
Var SegNeigh	0	0	NA	0.332099418	0.002293403	0.341618191	0.969439728	0.505207652
Var PELT	0	0	NA	0.332099418	0.002293403	0.341618191	0.969439728	0.505207652
Mean BinSeg	0	0	NA	0.584346906	0.100269497	0.594594595	0.584795322	0.589654248
Mean SegNeigh	0.4	0.666666667	0.5	0.851930196	0.685228297	0.756503642	0.916871031	0.829002955
Mean PELT	0.048387097	1	0.092307692	0.6811211	0	1	0.064516129	0.121212121

Table C.3: Scores for ‘Bol.com’ data set

APPENDIX C. FULL REAL-WORLD DATA SCORINGS

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0.25	0.5	0.333333333	0.924907456	0.844343916	0.890083632	0.888137357	0.889109429
MeanVar SegNeigh	0.25	0.5	0.333333333	0.924907456	0.844343916	0.890083632	0.888137357	0.889109429
MeanVar PELT	0.142857143	0.5	0.222222222	0.812268641	0.590695386	0.890083632	0.666493236	0.762229907
Var BinSeg	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	0.920395421	0.881900364
Var SegNeigh	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	0.920395421	0.881900364
Var PELT	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	0.920395421	0.881900364
Mean BinSeg	0.166666667	0.5	0.25	0.87678477	0.727378477	0.983870968	0.677560738	0.802479376
Mean SegNeigh	0.4	1	0.571428571	0.958223162	0.910909287	1	0.881650681	0.937103459
Mean PELT	0.033898305	1	0.06557377	0.607086198	0.004865842	1	0.05450052	0.10336746

Table C.4: Scores for ‘Connexxion’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.7106955	0.351202345	0.667180277	0.682527985	0.674766871
MeanVar SegNeigh	0	0	NA	0.698421975	0.334534305	0.655034895	0.703714426	0.678502647
MeanVar PELT	0.1	0.333333333	0.153846154	0.779661017	0.378254169	0.86440678	0.476111521	0.614022242
Var BinSeg	0	0	NA	0.790765634	0.562324673	0.667180277	0.884443977	0.760601126
Var SegNeigh	0	0	NA	0.77849211	0.543596211	0.655034895	0.905630417	0.760213636
Var PELT	0	0	NA	0.77849211	0.543596211	0.655034895	0.905630417	0.760213636
Mean BinSeg	0	0	NA	0.928696669	0.819906546	0.927966102	0.770037197	0.841657276
Mean SegNeigh	0	0	NA	0.920514319	0.805466429	0.850934376	0.818463347	0.834383069
Mean PELT	0.051724138	1	0.098360656	0.701344243	0.002735198	1	0.06927045	0.129565817

Table C.5: Scores for ‘Dakota Access Pipeline’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.776837652	0.464020988	0.881048387	0.548984468	0.676462612
MeanVar SegNeigh	0	0	NA	0.837123215	0.627783501	0.859582543	0.703106332	0.773510247
MeanVar PELT	0	0	NA	0.837123215	0.627783501	0.859582543	0.703106332	0.773510247
Var BinSeg	0.333333333	0.5	0.4	0.935483871	0.85755302	0.923963134	0.884259259	0.903675299
Var SegNeigh	0.333333333	0.5	0.4	0.935483871	0.85755302	0.923963134	0.884259259	0.903675299
Var PELT	0.333333333	0.5	0.4	0.935483871	0.85755302	0.923963134	0.884259259	0.903675299
Mean BinSeg	0	0	NA	0.905341089	0.782728087	0.962365591	0.744311394	0.839408606
Mean SegNeigh	0.2	0.5	0.285714286	0.941300899	0.868296668	0.975806452	0.828540784	0.896163916
Mean PELT	0.032258065	1	0.0625	0.639344262	0	1	0.048387097	0.092307692

Table C.6: Scores for ‘Jumbo’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.752542373	0.525490081	0.96969697	0.679861111	0.799316213
MeanVar SegNeigh	0	0	NA	0.551412429	0.241900508	0.96969697	0.432638889	0.598328306
MeanVar PELT	0	0	NA	0.528813559	0.214187614	0.96969697	0.404861111	0.571227361
Var BinSeg	0	0	NA	0.710169492	0.458374641	0.969047619	0.629861111	0.763477488
Var SegNeigh	0	0	NA	0.915254237	0.818859861	0.977777778	0.874305556	0.923151273
Var PELT	0	0	NA	0.915254237	0.818859861	0.977777778	0.874305556	0.923151273
Mean BinSeg	0	0	NA	0.736158192	0.502540289	0.977777778	0.654166667	0.783886525
Mean SegNeigh	0.2	1	0.333333333	0.746327684	0.519210875	1	0.665277778	0.798999166
Mean PELT	0.01754386	1	0.034482759	0.327118644	0.001636731	1	0.035416667	0.068410463

Table C.7: Scores for ‘Kamer van Koophandel’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0.333333333	0.333333333	0.333333333	0.849795441	0.667534638	0.740705303	0.935439137	0.826760168
MeanVar SegNeigh	0.2	0.333333333	0.25	0.845119813	0.653321966	0.753417168	0.867642527	0.806505494
MeanVar PELT	0.125	0.333333333	0.181818182	0.821157218	0.536777118	0.805488297	0.650385208	0.719674713
Var BinSeg	0.333333333	0.333333333	0.333333333	0.849795441	0.667534638	0.740705303	0.935439137	0.826760168
Var SegNeigh	0.333333333	0.333333333	0.333333333	0.849795441	0.667534638	0.740705303	0.935439137	0.826760168
Var PELT	0.333333333	0.333333333	0.333333333	0.849795441	0.667534638	0.740705303	0.935439137	0.826760168
Mean BinSeg	0.166666667	0.333333333	0.222222222	0.834599649	0.625980308	0.753417168	0.837134052	0.793072438
Mean SegNeigh	0.2	0.333333333	0.25	0.845119813	0.653321966	0.753417168	0.867642527	0.806505494
Mean PELT	0.051724138	1	0.098360656	0.715955582	0.002935273	1	0.069337442	0.129682997

Table C.8: Scores for ‘Rabobank’ data set

APPENDIX C. FULL REAL-WORLD DATA SCORINGS

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.799048123	0.561286116	0.803870968	0.753312212	0.777770817
MeanVar SegNeigh	0	0	NA	0.86938128	0.711729799	0.891935484	0.801699309	0.844413523
MeanVar PELT	0	0	NA	0.86938128	0.711729799	0.891935484	0.801699309	0.844413523
Var BinSeg	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Var SegNeigh	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Var PELT	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Mean BinSeg	0	0	NA	0.868852459	0.699871621	0.943548387	0.678715438	0.789515055
Mean SegNeigh	0	0	NA	0.925965098	0.835786141	0.954301075	0.785570276	0.861754013
Mean PELT	0.032258065	1	0.0625	0.626123744	0	1	0.048387097	0.092307692

Table C.9: Scores for ‘Tele2’ data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.779481756	0.488322974	0.709677419	0.779793907	0.743085305
MeanVar SegNeigh	0.2	0.333333333	0.25	0.947117927	0.869205369	0.925806452	0.84192055	0.881873135
MeanVar PELT	0.2	0.333333333	0.25	0.947117927	0.869205369	0.925806452	0.84192055	0.881873135
Var BinSeg	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Var SegNeigh	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Var PELT	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Mean BinSeg	0.166666667	0.333333333	0.222222222	0.939185616	0.845868213	0.955645161	0.798611111	0.870099604
Mean SegNeigh	0.4	0.666666667	0.5	0.931782126	0.838129718	0.862007168	0.909124851	0.88493927
Mean PELT	0.05	1	0.095238095	0.705975674	0.005045764	1	0.068399044	0.128040257

Table C.10: Scores for ‘UWV’ data set

Appendix D

Bibliography Annotations

What follows is an annotated bibliography of works that were read as part of the preparation for this research. It is not an exhaustive list, merely a set of papers, books and other publications which are relevant to this research or provided some insight into how best to carry it out. Full bibliographical citations are provided for each text.

- [10] Cody Buntain, Christopher Natoli, and Miroslav Zivkovic. “A Brief Comparison of Algorithms for Detecting Change Points in Data”. In: *Supercomputing*. 2014. URL: <https://github.com/cbuntain/ChangePointDetection>

One of the first papers on the subject read by the thesis author, this paper discusses and compares three different types of change detection algorithm: the *Likelihood Ratio Test* (LRT), the *Cumulative Sum* (CUSUM) test, and the *Kernel-based Change Detection* (KCD) algorithm. The authors of this paper also kindly provided source code implementations of these algorithms, once contact was established. The algorithms were applied to several data sets, such as historical Bitcoin valuations and data from structural stress sensors.

- [25] Yoshinobu Kawahara and Masashi Sugiyama. “Change-point detection in time-series data by direct density-ratio estimation”. In: *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), pp. 389–400

This paper proposes what the authors call a “novel non-parametric change detection algorithm”[25]. It is applied alongside four other approaches, against three different data sets generated by models borrowed from other research papers on time-series data change detection. The approaches are evaluated according to accuracy rate and degree, though not for performance or other relevant metrics.

- [28] Daniel Kifer, Shai Ben-david, and Johannes Gehrke. “Detecting Change in Data Streams”. In: *Proceedings of the 30th VLDB Conference* (2004), pp. 180–191. ISSN: 00394564. DOI: [10.1016/0378-4371\(94\)90421-9](https://doi.org/10.1016/0378-4371(94)90421-9). arXiv: [9310008](https://arxiv.org/abs/9310008) [cond-mat]

This paper, much like the one written by Kawahara and Sugiyama, presents a novel change detection algorithm for evaluation. However, this particular paper differs in that the presented algorithm is also useful for estimation of the detected change. They also discuss the use of a ‘two window’ approach to applying change detection algorithms to data streams, so as to limit memory usage in practice. This part in particular is relevant to the business case of the organisation hosting this thesis.

- [17] Pedro Galeano and Daniel Peña. “Variance Changes Detection in Multivariate Time Series”. 2004

This paper served as the basis of two of the algorithms tested in the paper written by Buntain, Natoli, and Zivkovic. This paper also provides its own evaluations of the approaches contained therein.

- [13] Frédéric Desobry, Manuel Davy, and Christian Doncarli. “An online Kernel change detection algorithm”. In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2961–2974. ISSN: 1053587X. DOI: [10.1109/TSP.2005.851098](https://doi.org/10.1109/TSP.2005.851098)

Desobry, Davy, and Doncarli write concerning an implementation of a Kernel Change Detection (KCD) algorithm that is considered *online* - that is, applicable to moving and updating data sets such as those that are processed by the host organisation of this thesis. This is particularly important, as any final implementation of a change detection method in the production systems of the thesis host company will be handling online data.

This paper makes use of Receiver Operating Characteristic curves to show the accuracy of change point detection algorithms, which is a variation on standard binary classification measures concerning false/true positives/negatives.

- [47] Alexander G. Tartakovsky et al. “Detection of intrusions in information systems by sequential change-point methods”. In: *Statistical Methodology* 3.3 (2006), pp. 252–293. ISSN: 15723127. DOI: [10.1016/j.stamet.2005.05.003](https://doi.org/10.1016/j.stamet.2005.05.003)

This paper discusses applying a CUSUM (Cumulative Sum) approach for detection of network intrusions. The approach taken by Tartakovsky et al. was intended to research the possibility of change detection while maintaining a low rate of false alarms. The idea was to apply an algorithm for this purpose that would not need to take into account pre-change and post-change models in order to be effective.

- [46] Alexander G. Tartakovsky and Boris L. Rozovskii. “A Nonparametric Multichart CUSUM Test for Rapid Intrusion Detection”. In: *Proceedings of Joint Statistical Meetings* (2005), pp. 7–11

A collection of papers and discussions by the same authors as the above papers, expanding somewhat on the problem they tackled and the solutions found.

- [35] David S. Matteson and Nicholas A. James. “A nonparametric approach for multiple change point analysis of multivariate data”. In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI: [10.1080/01621459.2013.849605](https://doi.org/10.1080/01621459.2013.849605). arXiv: [1306.4933](https://arxiv.org/abs/1306.4933)

Discussion of an ‘offline’ (that is, applying an algorithm to a fixed data-set as opposed to processing moving ‘live’ data) approach to change detection in multi-variate data. The authors carry out a simulation study to compare various approaches to this problem and present their results.

- [44] D. Siegmund and E. S. Venkatraman. “Using the generalized likelihood ratio statistic for sequential detection of a change-point”. In: *The Annals of Statistics* 23.1 (1995), pp. 255–271. ISSN: 0090-5364. DOI: [10.1214/aos/1176324466](https://doi.org/10.1214/aos/1176324466)

A paper on using a *Generalized Likelihood Ratio* test for the detection of change points in a data set. An old piece of text that pre-dates social media, yet is still useful in explaining how the GLR test can be used for the detection of change points. The GLR approach is compared against standard CUSUM tests.

- [50] Alan S. Willsky and Harold L. Jones. “A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems”. In: *IEEE Transactions on Automatic Control* 21.1 (1976), pp. 108–112. ISSN: 15582523. DOI: [10.1109/TAC.1976.1101146](https://doi.org/10.1109/TAC.1976.1101146)

Much like the paper written by Siegmund and Venkatraman, this paper pre-dates social media as a data source to which change detection algorithms could be applied. It is however, a useful look into how change detection algorithms have progressed and improved over the years. This paper is primarily focussed on uses of change detection in automated controls (being that it was published in the *IEEE Transactions on Automatic Control*), but is still a useful resource to understand how change detection algorithms can be evaluated.

- [34] Tze Leung Lai and Jerry Zhaolin Shan. “Efficient recursive algorithms for detection of abrupt changes in signals and control systems”. In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 952–966. ISSN: 00189286. DOI: [10.1109/9.763211](https://doi.org/10.1109/9.763211)

Another paper concerning change point detection in control systems, Lai and Shan write primarily regarding Generalised Likelihood Ratio tests, and how these can be configured, specifically with regards to window size.

- [8] S. Bersimis, S. Psarakis, and J. Panaretos. “Multivariate statistical process control charts: An overview”. In: *Quality and Reliability Engineering International* 23.5 (2007), pp. 517–543. ISSN: 07488017. DOI: [10.1002/qre.829](https://doi.org/10.1002/qre.829)

Control charts are a mechanism used in various industries to decide whether a given process is ‘in control’ or ‘out of control’. In this way, control charts are used to discover outlying or anomalous results in a given data set, or inform operators of a sudden change in the data. This makes control charts particularly relevant to my research.

This particular paper discusses various approaches to control charts, and the methods behind their operation.

- [14] Allen B. Downey. “A novel changepoint detection algorithm”. In: *Applied Microbiology and Biotechnology* (2008), pp. 1–11. arXiv: [0812.1237](https://arxiv.org/abs/0812.1237). URL: <http://arxiv.org/abs/0812.1237>

Downey discusses his creation of a ‘novel change detection algorithm’ that can also be used for “...predicting the distribution of the next point in the series.”[14]

He discusses in some detail the difference between online and offline change detection algorithms, and compares his implementation of a new algorithm with implementations of existing and established algorithms.

- [31] Rebecca Killick, Idris Eckley, and Philip Jonathan. “Efficient Detection Of Multiple Changepoints Within An Oceanographic Time Series”. In: (2011), pp. 4137–4142. URL: http://www.lancaster.ac.uk/%7B~%7Djonathan/2011%7B%5C_%7DISI%7B%5C_%7DChangePoints%7B%5C_%7DTalk.pdf%20http://eprints.lancs.ac.uk/56978/

This paper carries out research similar in some ways to that being conducted in this thesis. Change detection methods are briefly explained and then applied to time series data from an oceanographic study. The ability for change detection approaches to correctly identify the onset and end of ‘storm seasons’ - which form the ground truth in this study.

- [35] David S. Matteson and Nicholas A. James. “A nonparametric approach for multiple change point analysis of multivariate data”. In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI: [10.1080/01621459.2013.849605](https://doi.org/10.1080/01621459.2013.849605). arXiv: [1306.4933](https://arxiv.org/abs/1306.4933)

A comparison study of several change point detection methods. This study utilises the Rand Index as a measure for comparing algorithm output against a ground truth, and the Adjusted Rand Index for computing the similarity between two computed outputs. The approaches are then applied to various real-world data sets from fields such as genetics and finance. This paper formed the basis for the decision to concentrate mainly on the performance of clustering metrics in the field of change point detection evaluation.

- [29] R. Killick, P. Fearnhead, and I. A. Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* 107 (2011), pp. 500–1590. ISSN: 0162-1459. DOI: 10.1080/01621459.2012.737745. arXiv: 1101.1438. URL: <http://amstat.tandfonline.com/action/journalInformation?journalCode=uasa20><http://dx.doi.org/10.1080/01621459.2012.737745><http://arxiv.org/abs/1101.1438><http://dx.doi.org/10.1080/01621459.2012.737745>

The paper that initially proposed the PELT algorithm for change point detection. The paper mainly concentrates on the computational cost of the PELT algorithm, comparing to other, existing methods of change point computation. Some analysis on the basis of algorithm accuracy is carried out, though limited to the use of binary classification measures.

- [38] Anita M. Pelecanos, Peter A. Ryan, and Michelle L. Gatton. “Outbreak detection algorithms for seasonal disease data: a case study using Ross River virus disease.” In: *BMC medical informatics and decision making* 10.1 (2010), p. 74. ISSN: 1472-6947. DOI: 10.1186/1472-6947-10-74. URL: <http://www.biomedcentral.com/1472-6947/10/74>

A paper written concerning possible approaches for the detection of disease outbreaks. This publication was not interesting in terms of the approaches used, but rather in how they were evaluated. This paper restricts itself to the use of binary classification measures, tallying true/false positives and true/false negatives.

- [39] Abdulhakim A. Qahtan et al. “A PCA-Based Change Detection Framework for Multidimensional Data Streams”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pp. 935–944. DOI: 10.1145/2783258.2783359. URL: <http://dl.acm.org/citation.cfm?id=2783258.2783359>

An evaluation of change point detection methods, using traditional binary classification measures. The measure results are presented purely in the form of true positives, late detections, false positives and false negatives, and thus are easily digested and if necessary translated into the usual terms of recall, precision and F1.

- [3] Enrique Amigó et al. “A comparison of extrinsic clustering evaluation metrics based on formal constraints”. In: *Information Retrieval* 12.4 (2009), pp. 461–486. ISSN: 13864564. DOI: 10.1007/s10791-008-9066-8

A detailed evaluation of BCubed as a measure for document clustering. It was based off of this paper that BCubed was chosen as an additional metric to test. BCubed has a good reputation for accuracy, and this paper backs up the ability for BCubed to be effective with detailed comparisons of it’s constraints and ability to perform correctly in various situations.

- [2] Foteini Alvanaki et al. “EnBlogue: emergent topic detection in Web 2.0 streams”. In: *Proc. ACM SIGMOD International Conference on Management of Data* (2011), pp. 1271–1274. ISSN: 07308078. DOI: 10.1145/1989323.1989473. URL: <http://doi.acm.org/10.1145/1989323.1989473><http://dx.doi.org/10.1145/1989323.1989473><http://www.acm.org/publication/uuid/44E009D1-8574-49C1-A0AC-C2B247FE9043>

An alternative method of achieving the business goals of the host organisation of this thesis, “EnBlogue: emergent topic detection in Web 2.0 streams” provides an explanation and example implementation of a method for detecting changes in *conversation topic* as opposed to conversation volume.

- [7] M. Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. 1993. ISBN: 0-13-126780-9. DOI: 10.1016/0967-0661(94)90196-1. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.6896%7B%5C%7Drep1%7B%5C%7Dtype=pdf>

The seminal text on change point detection methods and evaluation. This text is considerably old at the time of writing, but is still an invaluable source of insights and explanations as to how change point detection works, and the criteria by which they should be evaluated.

- [12] Tamraparni Dasu et al. “Change (detection) you can believe in: Finding distributional shifts in data streams”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5772 LCNS (2009), pp. 21–34. ISSN: 03029743. DOI: [10.1007/978-3-642-03915-7_3](https://doi.org/10.1007/978-3-642-03915-7_3)

An interesting text discussing the application of distributional shift detection as a change detection measure. The approach is not necessary applicable to the work being carried out in this thesis, but it is interesting to read nonetheless - considering the differences between this approach and those tested in this research.

- [19] Jeremy Ginsberg et al. “Detecting influenza epidemics using search engine query data.” In: *Nature* 457.7232 (2009), pp. 1012–4. ISSN: 1476-4687. DOI: [10.1038/nature07634](https://doi.org/10.1038/nature07634). URL: <http://www.ncbi.nlm.nih.gov/pubmed/19020500>

Another study in the same vein as [38], this time carried out by Google. This study uses data streams to attempt to predict influenza outbreaks with surprising effectiveness. Here, a log-likelihood approach is used to determine the probability that a random visit by a doctor is related to an illness like influenza.

- [33] Martin Kulldorff et al. “A space-time permutation scan statistic for disease outbreak detection”. In: *PLoS Medicine* 2.3 (2005), pp. 0216–0224. ISSN: 15491277. DOI: [10.1371/journal.pmed.0020059](https://doi.org/10.1371/journal.pmed.0020059)

An additional study in the same category as [19] and [38], it is interesting to this project due to it’s use of observed/expected cases to prove the veracity of their prediction model, in the same manner as binary classification scores are utilised in other publications.

- [49] Dang-Hoan Tran, Mohamed Medhat Gaber, and Kai-Uwe Sattler. “Change Detection in Streaming Data in the Era of Big Data: Models and Issues”. In: *ACM SIGKDD Explorations Newsletter - Special issue on big data 1* (2014), pp. 30–38. ISSN: 1931-0145. DOI: [10.1145/2674026.2674031](https://doi.org/10.1145/2674026.2674031)

A paper discussing the importance of change point detection techniques when applied to *big data* scenarios. No comparison study is carried out, but the paper does serve to espouse the importance of change point detection in scenarios where the sheer *volume* of data may serve to make many approaches impractical.

- [51] Yi Xu et al. “Pattern change discovery between high dimensional data sets”. In: *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11* (2011), p. 1097. DOI: [10.1145/2063576.2063735](https://doi.org/10.1145/2063576.2063735). URL: <http://dl.acm.org/citation.cfm?doid=2063576.2063735>

This study compares various approaches for the application of event detection in data streams. It provides experiments evaluating approaches for topic change detection in documents and news streams, and also event detection in surveillance video streams. In a similar fashion to [2], it focuses mainly on using analysis techniques to spot changes in topic in documents as opposed to simply measuring volume.