Benchmarking of Facebook's Prophet, PELT and Twitter's Anomaly detection and automated deployment to cloud

Master Thesis

by

Siddhartha Srivastava

1710540

MSc Computer Science

Data Science and Smart Services

Enschede

Graduation Committee

Dr Maurice Van Keulen, University of Twente

Riccardo Vincelli, KPMG

Dr Doina Bucur, University of Twente

UNIVERSITY OF TWENTE.

KPMG

Prefix

I would like to thank my company supervisor, Riccardo Vincelli, i thank you for providing me with the topic for this master thesis that piques my interest and connects data engineering and data science, making it a perfect fit for my degree in data science. And a further thanks for supporting me along the way with guidance and good advice. I also thank Youri Wessie and Ralph Urlus from the company, who guided my along the way whenever i needed help.

A thank you goes to the University of Twente, for offering this interesting Master program, and allowing me to do an internship abroad. Of course, I would also like to thank my two supervisors Dr. Maurice Van Keulen and Dr Doina Bucur for providing helpful feedback and support. Finally and most importantly, I would like to thank my family and friends, who have always shown great support, especially my parents without whom none of this would have been possible.

Contents

List of Figures

List of tables

Abstract

We perform benchmark and do a comparison of three algorithms, namely Facebook's Prophet and PELT, which are changepoint detection algorithms, and Twitter's Anomaly Detection, which is an anomaly detection algorithm, to see which one is better. The benchmarking is done over synthetic and real datasets, and they have been chosen to accommodate as many real world cases as possible. The metric chosen to compare them are Accuracy, Error rate, Specificity, Precision, Recall and F-measure. Out of these metrics, Precision, Recall and F-measure have been given more weight because they are dependent on number of true positives detected, the points that actually are anomalies/changepoint, which is what we are interested in. Less importance is given to Accuracy, Error rate and Specificity as they are dependent on number of true negatives, the points that are not changepoints/anomalies, which we are less interested in. Run time of the algorithms is also taken into account. We found that PELT is better than Prophet in terms of Precision, Recall and F-measure, and also it is faster than Prophet. Twitter's Anomaly Detection works best on real data. One of the algorithms, PELT, was deployed to cloud over Kubernetes, a container orchestration engine.

1 Introduction

The work presented in this thesis has a main purpose to introduce and critically assess two changepoint detection and one anomaly detection algorithm to help better understand which algorithm works better under what circumstances, and get a picture of the true nature of underlying data.

In today's age, we are able to record and store more data than ever. It can be useful to know if there are any breaks or anomalies in the data, and most importantly why are they there. Detecting them is getting one step closer to understanding the why. For example, take data generated by machines in industries. A change or an outlier in data might indicate failure or change in activity of systems.

1.1 Changepoints

A point in data series before and after which there is a change in one or more of statistical properties is called as a *changepoint*. There are several types of changepoints, there could be a changepoint in data when there is a change in mean, change in variance, change in mean and variance. In a non-stationary signal every point is a changepoint, some can be significant while others might be not).

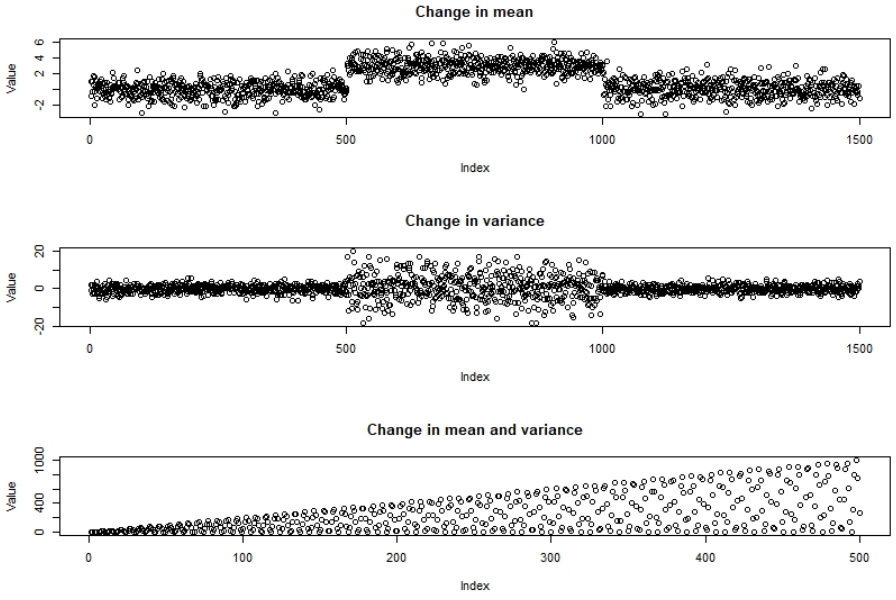

Figure 1: Data with changepoints, 1^{st} figure shows change in mean, 2^{nd} figure shows change in variance, 3^{rd} figure shows change in mean and variance

Figure 1 shows 3 types of synthetic data, first figure shows data with change in mean, 2nd figure shows data with change in variance and 3rd figure shows data with change in mean and variance. They all have changepoints, before and after which statistical properties of data change.

Changepoint detection has applications in fraud detection[1], for example when a credit card is stolen and there is a spike in spending habit, intrusion detection in computer networks [2], for example when a system is breached and it sends or downloads huge amounts of or sensitive data. It is also used in signal segmentation in data stream [3], which means decomposing a signal into stationary segments. Finally it is used in fault detection in engineering systems [4], where changepoints can be seen as abrupt changes in the system's behavior. These abrupt changes could be faults, and timely detection helps improve availability and reliability of these systems.

There are several types of changepoint detection methods. They fall into two main categories: supervised and unsupervised. Supervised techniques, for example, include Decision Tree, Nearest neighbor, Support Vector Machines, Naive Bayes etc. They learn from labeled training data, which points are changepoints and which are not.

Unsupervised methods find out hidden patterns in data, which is not labeled. They usually divide the data into segments, and then find out changepoints using the individual segment's statistical properties. Some methods use likelihood ratios, in which a point is picked, and the probability distributions of segments before and after that point is determined. If the probability distributions are different, that point is marked as a changepoint. Another approach is clustering based anomaly detection, where points are grouped into clusters. If two consecutive points are not in the same cluster, that point can be marked as a potential changepoint candidate. Unsupervised methods include Likelihood ratio methods like CUSUM and PELT[5].

1.2 Anomalies

Anomalies are instances in data that do not conform to a pattern or normal behavior. They can often convey information that is useful, critical, actionable and beneficial to businesses.

Figure 2 shows an example of anomalies in data. *Anomalies* are marked in red.

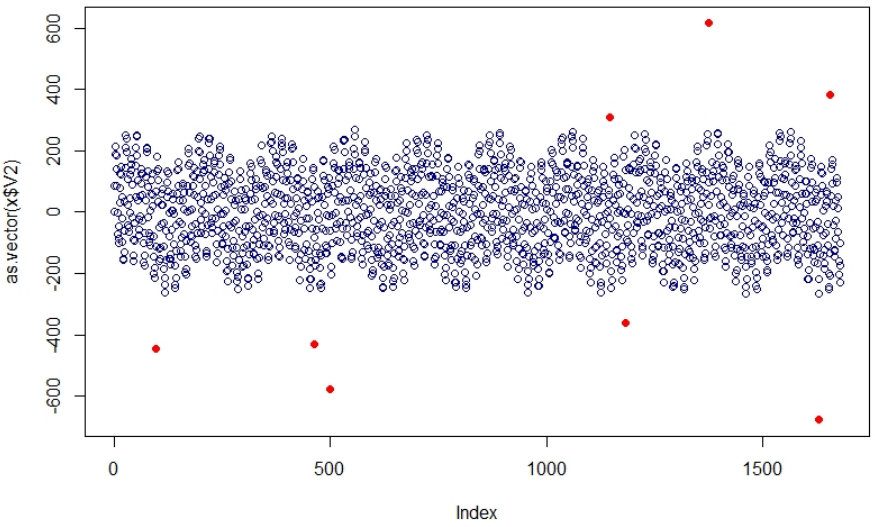

Figure 2: Anomalies marked in red, in sample data

There are multiple types of anomalies. Several are listed below [6].

a) **Point anomalies**: A single data point is said to be a point anomaly if it is considerably different from the rest of the dataset. Those points that lie on the extremes fall under this category. The example in Fig 2 shows point anomalies.

b) **Collective anomalies**: The data points that are anomalous if taken in a group, but non-anomalous individually are called collective anomalies.

c) **Contextual anomalies**: Data points that are abnormal in a particular context but normal without that context are called contextual anomalies.

It is difficult to come up with a definition of an anomaly that accounts for every deviation from a normal or standard behavior. Mostly because anomalies differ from application to application. It becomes difficult to generalize normal and abnormal behavior that covers different data types and domains.

Anomaly detection has applications in various domains. For example, in intrusion detection, a compromised machine may be sending out information to a host that does not have permission [7]. An intrusion makes the system behave differently, which is why anomaly detection can be used in this domain. Anomalies in an MRI scan may show presence of a tumor [8], fault detection in

mechanical systems [9], or a disease outbreak could be detected [10]. Anomaly detection can be used in credit card fraud detection [11], where anomaly is an unusual number of purchase transactions, purchasing items that have never been bought before, etc. They would come up as point anomalies. Another application is anomaly detection in sensor networks [12], where anomalies would come up as faults or intrusion detection.

There are several methods of anomaly detection. One is classification based anomaly detection. In this method, similar to changepoint detection, a classifier runs on a training data with anomalies labeled in it. It then classifies, based on the model created during training phase, points as anomalous and non-anomalous on the test data. It can be divided into one-class classifier and multi-class classifier. Examples are techniques that are neural networks based, which can be both single class or multi-class[13,14], Bayesian networks based [15], support vector machines based or rule based. Then there are nearest neighbor based techniques that compute the distance between two data points, which is usually euclidean distance. These can be of two types, one that computes distance to k^{th} nearest neighbor, and one the uses relative density. Then there are clustering based techniques that group data into clusters, and it is an unsupervised technique. Normal data points belong into a cluster, and anomalies belong outside the cluster. Finally there are statistical anomaly detection techniques, that assume a stochastic model, and points in high probability regions of stochastic model are marked as normal, whereas points in low probability regions are marked as anomalies[16].

1.3 Problem Statement

A lot of literature exists on various types of changepoint detection and anomaly detection algorithms, but there is a gap on their comparison. Not a lot of literature exists on their benchmarking, one that gives an idea on which algorithm fares better on what kind of dataset and under which circumstances. This thesis aims to conduct a comprehensive benchmark of three unsupervised algorithms, Facebook's Prophet [1] and PELT [2] which are changepoint detection algorithms, and Twitter's anomaly detection [3] , which is an anomaly detection algorithm. The reason Prophet was chosen is because it was the business requirement of the company. For the second changepoint detection algorithm, there were several candidates, like E-divisive and Bayesian online changepoint detection algorithm. The reason PELT was chosen was because it is an offline algorithm, unlike Bayesian online changepoint algorithm, and E-divisive takes a long time to run even on small datasets, which made it not suitable for practical purposes. Then the selected algorithm's performance is compared by using standard evaluation metrics like Accuracy, Precision, Recall and Error rate. Benchmarking and comparison is done for Prophet and PELT, whereas only benchmarking is done for Anomaly Detection. We conducted a comparison for Prophet and PELT because there were two candidate algorithms to compare

each other with. Benchmarking was done only in the case of Twitter's Anomaly Detection because there was not a second algorithm to compare this to. Then, PELT was deployed to the cloud, with the aim of automating the deployment process. By automation it is meant that the deployment process should be as convenient, configuration free and flexible as possible so that any algorithm can be deployed, and once deployed the algorithm should give the results (changepoints or anomalies) when fed data.

1.4 Research question

Resulting from the facts stated in the previous section, in this thesis, changepoint detection and anomaly detection frameworks are identified and implemented for solving the task of detecting changepoints and anomalies in data. The frameworks are applied to different data types, namely synthetic data with varying mean and variance, stationarity, random walk data as well as real world data. This is done to establish which framework and data is best-suited for the task of detection of changepoints and anomalies given a wide range of real world scenarios where changepoints and anomalies could occur.

The main research question can be formulated as follows:

How do the selected changepoint and anomaly detection frameworks fare to different types of data containing changepoints and anomalies, and how can the deployment process be automated?

Several sub-questions are needed to answer the main research question above:

Sub-Question 1: *What kind of datasets cover scenarios of changepoints and anomalies that occur frequently in real world datasets?*

Sub-Question 2: *How does each framework perform against synthetic and real world datasets with changepoints and anomalies?*

Sub-Question 3: *What evaluation metrics can be used to compare these frameworks?*

Sub-Question 4: *How can a selected framework be deployed to cloud towards automation?*

1.5 Research method and evaluation metrics

The algorithms would be fed data, both synthetic and real world, to see how well they perform. The datasets would be of different kinds, and synthetic datasets would have artificial changepoints and anomalies injected into them. Each dataset would represent most of the real world scenarios where anomalies and changepoints could occur. Then a confusion matrix would be created from which evaluation metrics, namely Accuracy, Precision, Recall and Error rate would be derived. Based on these metrics the performance of the changepoint detection algorithms would be compared, and for anomaly detection algorithm it is determined how well it performs under each scenario.

1.6 Literature gap

A lot of literature has been devoted to different types of changepoint detection techniques and anomaly detection techniques. Anomaly detection and changepoint detection are inherently different, they both do different things, their approach is different. Anomaly detection detects anomalies or outliers, which by definition do not conform to a pattern, whereas changepoint detection techniques detects changes in data stream, which by definition mean data before a changepoint and after a changepoint has different statistical properties. This thesis presents a comprehensive benchmark of three frameworks, two of them being changepoint detection frameworks and one being anomaly detection framework. We chose two different types of algorithms because initial idea was to come up with a benchmark which can compare one anomaly detection algorithm against a changepoint detection algorithm, but for the scope of the thesis, we couldn't derive it. By comprehensive it is meant that most real world cases are covered where changepoint and anomalies can be present, and the benchmarking is done while keeping the parameters at default. This is because keeping the parameters at default would give results that are generalized, and not specific to any dataset. Some datasets are more suited to anomaly detection techniques and some to changepoint detection techniques. This is why a real world dataset was included that has changepoints as well as anomalies, which also brings us one step closer to comparing these two techniques.

1.7 Report organization

The remainder of the report is organized as follows. Section 2 goes through related work, several other types of changepoint and anomaly detection algorithms, and applications of anomaly detection and changepoint detection. Section 3 goes through background of the selected techniques. Section 4 goes through the methodology, the research method, evaluation metrics and tool selection. Section 5 goes through different types of data, how they were generated and why were they selected. It also goes through real world dataset. Section 6 goes through implementation of the algorithms over the selected datasets, then their comparison and discussion. Section 7 goes through automation and

deployment of PELT over Kubernetes. Section 8 goes through limitations and future work, and section 9 is conclusion.

2 Related Work and Background

This section goes through background and various types of changepoint detection and anomaly detection algorithms, and what categories they fall under.

2.1 Background

2.1.1 Changepoints versus Anomalies

Anomaly points "The items, events or observations that don't conform to an expected pattern or other items in the dataset" [4].

Change points "An intervention, that may lead to change of the original series. Statistical properties of data before and after that change are different" [5].

Changepoints are subset of anomalies. Anomalies can be outliers, something which is out of the ordinary, and this encompasses changepoints.

2.1.2 Online vs Offline

Changepoint and anomaly detection algorithms can be online or offline [5]. Offline algorithms take into account the whole dataset at once, and then detects changepoints/anomalies. Online algorithms on the other hand, work on real-time or streaming data, taking into account new points as they come, detecting changepoints/anomalies as soon as it occurs.

2.1.3 Background of synthetic data

This subsection goes through background of synthetic data used in the work, namely stationary data, non-stationary data and sine wave.

1) **Stationary data**: Stationary data is used in anomaly detection in the thesis, because certain data, especially time series data can be stationary. It can be divided into weakly stationary and strongly stationary data [6].

a) Weakly stationary data: In weakly stationary data, the mean, variance, and covariance of data does not change over time [6]. We have used weakly stationary data in the thesis.

b) Strongly stationary data: In strongly stationary data, the distribution of random variables in the data is the same [6].

If $(x_t : t \in Z)$ and $k \in R$ then

x_1, x_2, x_3.... have the same distribution function,

and the statistical distribution of (x_1, x_2, x_3) is same as distribution of $(x_{1+k}, x_{2+k}, x_{3+k})$

2) **Non-stationary data**: A non-stationary series is one in which the statistical properties of data change over time [7]. The statistical properties can be mean, variance, or any other property. We have used non-stationary data in changepoint detection, since strictly speaking every point is a changepoint in non-stationary series, so it makes an interesting candidate for changepoint detection.

If $(x_t : t \in Z)$ and $k, n \in R$ then

mean and variance of (x_1, x_k) is not equal to mean and variance of (x_k, x_n)

3) **Sine wave**: A sine wave is a series with periodic oscillation, and a combination of multiple sine waves can exhibit non-periodic amplitude and phase. Certain time series data exhibit this behavior, which is why it has been included in the thesis, as an example in anomaly detection.

The sine wave included in this thesis is a combination of two sine waves with different phase and amplitude [8]. A sine wave can be defined as

$y(t) = Asin(\omega t + \phi)$

Where A is the amplitude

ω is angular frequency

ϕ is phase

2.2 Related work: Anomaly detection algorithms

There are a variety of techniques that can be used for anomaly detection. They mostly fall under the category of statistical, supervised and unsupervised methods.

2.2.1 Statistical Techniques

Early techniques for anomaly detection were statistical techniques, and there a lot of outlier detection methods designed. Hodge and Austin [9] divide statistical techniques into four categories, parametric, non-parametric, semi-parametric and proximity based. Parametric methods are useful when size of data is large and model depends upon number of dimensions rather than number of observations. For example, least squares regression [10]. Non-parametric methods are useful when data has a unknown distribution, or multiple distributions. Semi-

parametric methods apply local distributions to data, instead of relying on a single distribution. Proximity based techniques try to compute the distance between points. For example, k-nearest neighbor [11], where the euclidean distance is compared to determine if a point is anomalous, or k-means [12] which tries to minimize the sum of squares in a cluster of points.

2.2.2 Supervised anomaly detection

A dataset with anomalous and non-anomalous points labeled in it is required as training data for supervised anomaly detection. Examples are Artificial Neural networks, Bayesian Networks, rule based classifiers, etc [4]. It is important for training data to cover as much of normal and anomalous behavior as possible, so that the algorithm performs good on test data. If the different attributes of data are essential to the model, they should be covered in the training data. Examples include classification techniques, which can be one class or multi-class techniques. They include neural networks [13], which can be both one class or multi-class, then Bayesian networks [14], which is a multi-class classifier. Another technique is support vector machines [15], which learns the region of boundaries of normal instances, and points away from the boundary are anomalous.

2.2.3 Unsupervised anomaly detection

These methods learn from the data itself in an unsupervised manner, they do not need a pre-classified training data. The outlier detection is done without having any prior knowledge of data [9]. An example of unsupervised anomaly detection is clustering [16], it divides the data into groups. An anomalous point is usually located away from the cluster. The distance between the points is usually euclidean distance. Another way it detects anomalies is by calculating the density of the clusters. Normal points are located in high density clusters and anomalous points are located in low-density clusters [16]. Another method is nearest neighbor based anomaly detection [11]. It either compute the distance between a point and its k-th nearest neighbor or computes the relative density of each data point to calculate its anomaly score.

2.3 Related work: Changepoint detection

Techniques for changepoint detection are categorized the same way as anomaly detection algorithms. They are supervised, unsupervised, and statistical techniques.

2.3.1 Supervised techniques

Supervised changepoint detection techniques learn from training data with points labeled as changepoints and not-changepoints. They in turn can be classified into binary classifiers and multi-class classifiers. They need a training data that

sufficiently represents cases where a point may be a changepoint, so as to capture the diversity. Examples of this category are decision trees, naive Bayes, Bayesian net, support vector machines, nearest neighbor, hidden markov model, conditional random field, and Gaussian mixture model [5]. In binary classifiers, a sequence with all the changepoints represents one class, and non-changepoints represent another class. Examples of this include support vector machines, naive Bayes, logistic regression[5].

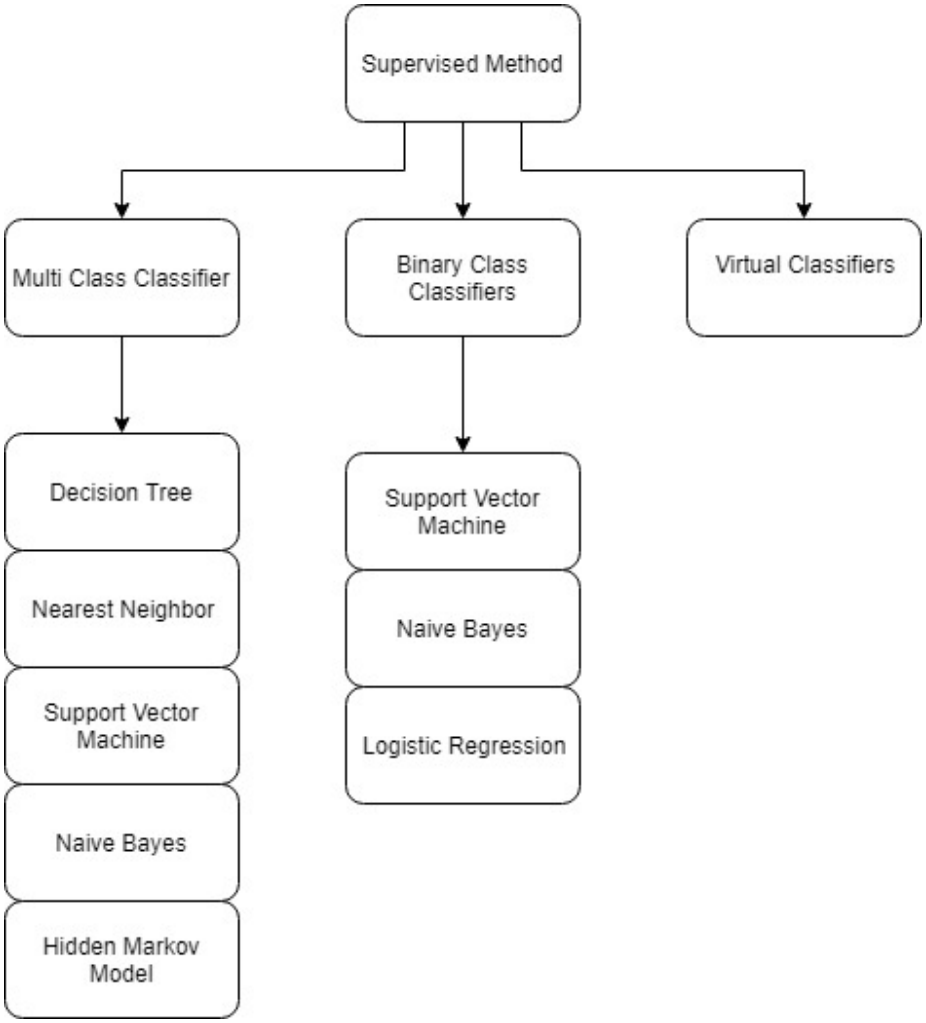

Figure 3: Supervised methods for changepoint detection [5]

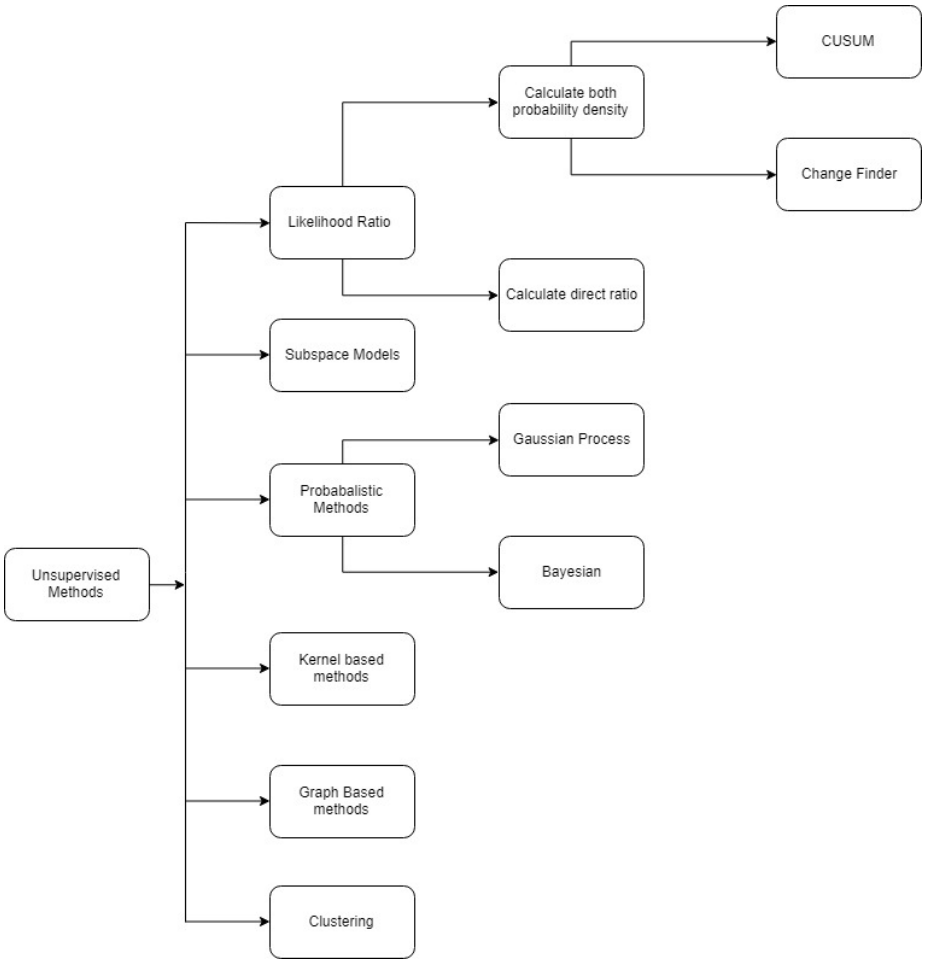

Figure 4: Unsupervised methods for changepoint detection [5]

2.3.2 Unsupervised techniques

Unsupervised techniques discover hidden patterns in un-labeled data. They segment the data, and find changepoints based on statistical properties of data. There is some overlap between unsupervised methods and statistical methods. Examples include likelihood ratio, probabilistic methods, graph based methods, and clustering methods [5]. Unsupervised techniques can be useful in certain cases because they do not require training data. Some methods use likelihood ratio [17], if the probability distributions of data before and after a candidate changepoint are different, it is a changepoint. Clustering methods [18] assume that data within clusters is identically distributed. If data at time t is is from a different cluster from point at time $t + 1$, there is a changepoint between them.

2.3.3 Statistical Techniques

These techniques have an overlap with some unsupervised techniques, like likelihood ratio and probabilistic methods [5,17]. These include techniques like binary segmentation [19], whose approach is recursive. It applies a changepoint detection technique to a dataset, divides it into two parts, one before and one after changepoint, and applies changepoint detection technique to both segment, and finally applies recursion. Then there are dynamic programming based methods [20], which aim to minimize a cost function on segments of data. Then there are non-parametric techniques [21] which are used when the underlying distribution is not known. An example is e-divisive algorithm [22], which divides the data into segments, then applies cost function that maximizes euclidean distance between two segments.

3 Selected techniques and their background

In this section i will go through the background of each framework, Facebook's Prophet, PELT and Twitter's Anomaly Detection.

3.1 Twitter's Anomaly Detection

Twitter's package is built for detecting anomalies in time series data that exhibit heavy seasonality and trends [3] This is because Twitter is a social platform in which trending events are captured. It works as follows.

3.1.1 Terminologies

Twitter decomposes a time series into three separate series, Seasonal, Residual and Trend, and then applies Extreme Studentized Deviate (ESD) [23] to detect anomalies. Several statistical methods that are used in the process are mentioned below.

1) Extreme Studentized Deviate (ESD)

In a dataset that follows a normal distribution, Extreme Studentized Deviate can be used to detect one or multiple anomalies. The number of anomalies that have to be detected have to be given as an input to ESD. Since by definition, anomalies are data points that do not conform to the patterns of majority of data, they have to be less than 50% of the data, that is why the maximum number of anomalies that can be supplied is 49% of the whole data.

Let the number of anomalies be k. Then, based on that k, the following statistic is calculated for each point in the time series.

$$C_k = \frac{max_k|x_k - \bar{x}|}{s}$$

where, $\bar{x}$ and s denote the mean and variance of the time series X.

A point from k is marked as anomaly or outlier if it is greater in comparison than the threshold which is determined from the following equation. If the point is greater, the threshold is computed again after removing that point from the dataset.

$$\lambda_k = \frac{(n-k)t_{p,n-k-1}}{\sqrt{(n-k-1+t^2_{p,n-k-1})(n-k+1)}}$$

This process is repeated the k times, equaling the number of anomalies given as input. Eventually, the anomalous becomes lower than the threshold.

2) Median and Median Absolute Deviation

A statistically robust median and median absolute deviation are introduced to replace ESD, this is because they are robust against presence of high number of anomalies, and ESD is based on mean and standard deviation, which are sensitive to the presence of anomalies. There is a direct correlation between the number of anomalies and the amount of effect they have on usefulness of mean and standard deviation.

For a univariate data set $X_1, X_2, ..., X_n$, MAD is defined as the median of the absolute deviations from the sample median. Formally,

$$MAD = median_i(|X_i - median_j(X_j)|)$$

Presence of high number of anomalies does not have an effect on MAD [24].

3) Moving Averages

Moving average is what the name suggests, an average that changes with respect to selected consecutive data points in dataset. One type of moving average is simple moving average (SMA), which is defined as:

$$SMA_t = \frac{x_t + x_{t-1} + ... + x_{t-(n-1)}}{n}$$

This type of average gives the average of selected t points out of n. Equal weights are given to each point.

Another type of moving average is exponentially weighted moving average (EWMA) [25], which can be defined as

$$EWMA_T = \begin{cases} y_t = x_t, & t = 1, \\ y_t = \alpha(x_t) + (1-\alpha)y_{t-1}, & t > 1 \end{cases}$$

In exponentially weighted moving average, different weights are assigned to different data points, which can be useful in certain situations.

4) Seasonality and STL

ESD and other anomaly detection techniques assume a unimodal distribution, whereas there is no guarantee that the time series data can only be unimodal, it can be multimodal as well. That is why these techniques would not work with certain types of data, those with multi-modality

The solution to this is series decomposition using STL [26]. An existing time series (X) can be decomposed into seasonal (S_X), trend (T_X), and residual (R_X) components. Out of these, residual component is unimodal, and now anomaly detection techniques like ESD can be applied to it.

For time series decomposition, the algorithm first determines the trend T_X by using a moving average filter, and then subtracts it from the time series X. The seasonal component S_X is then determined by taking mean of all the data points in the time series, and that is subtracted from X as well. What is then left is residual, R_X, which is unimodal.

$$R_X = X - T_X - S_X$$

3.1.2 Anomaly detection algorithm: Seasonal ESD

Presence of seasonality and a multimodal distribution in the nature of time series data prevented techniques like ESD to detect anomalies.

To counter these problems, an algorithm was proposed, called Seasonal-ESD. It applies an STL-variant to extract residual component, then applies ESD to detect anomalies. It has the advantage that it detects both global anomalies, and local anomalies masked by seasonality.

STL Variant

Normal STL decomposition ended up producing anomalies not originally present in original time series. To rectify this problem, the STL-variant subtracts median of the time series data, instead of trend component.

$$R_X = X - S_X - \tilde{X}$$

X being the time series, S_X the seasonal component and $\tilde{X}$ the median of the time series.

Using STL variant it is possible to detect local anomalies masked by seasonality.

S-ESD Limitations

The advantage of S-ESD is that it can detect both local and global anomalies,

Algorithm 1 S-ESD algorithm [3]

Require: X = a time series
n = number of observations in X
k = max anomalies (number of iterations in ESD)

$k \leq$ (n * 0.49)

Output: an anomaly vector where each element is a tuple (timestamp, observed value)

Steps

1: Extract seasonal component S_X using STL variant
2: Compute median $\tilde{X}$

Compute Residual
3: $R_X = X - S_X - \tilde{X}$

Detect anomalies vector X_A using ESD
4: X_A = ESD(R,k)

return X_A

whereas its disadvantage is that if a high number of anomalies are present, S-ESD does not work well.

3.1.3 Seasonal Hybrid ESD (SH-ESD)

Seasonal Hybrid ESD (S-H-ESD) uses the robust statistical measures Median and MAD as described above. They are particularly useful when the number of anomalies are high, which usually reduces the effectiveness of mean and standard deviation, which are used in S-ESD, because high values of mean and standard deviation can cause true anomalies to pass off as not-anomalous. Although, it should be noted that S-H-ESD takes longer to run.

3.2 Facebook's Prophet

Prophet is mainly a time series forecasting tool [1]. It has the added functionality to detect changepoints. Similar to Twitter's Anomaly Detection, it decomposes a time series into 3 components, seasonal, trend and holidays (instead of residual which was the case in Twitter's model). The model can be shown as:

$$y(t) = g(t) + s(t) + h(t) + \epsilon t$$

g(t) is the trend function to capture non-seasonal changes, s(t) to capture seasonal changes, and h(t) to capture the effect of holidays on the time series. ϵt captures any changes not accommodated by trend, holidays or seasonal components in the time series.

Changepoints are detected in trend component, similar to Twitter's anomaly detection where they were detected in residual component. Trend is composed of two parts: a saturating growth model, and a piecewise linear model.

Nonlinear saturating growth

Growth in Prophet is modeled as it happens in nature, non-linearly with a carrying capacity that becomes constant after time. This growth takes the form of a sigmoid curve [27]. It is modeled using logistic growth model [28], which is

$$g(t) = \frac{C}{1 + exp(-k(t - m))}$$

where C is the carrying capacity, k is the growth rate, and m an offset parameter.

This equation does not fully capture the growth at Facebook, first reason being that the carrying capacity, that is the number of users that have access to internet, is not constant, it is varying, so that is replaced from C to C(t). Another factor is that growth rate is also varying, number of users getting access to internet can drastically increase or decrease due to a number of reasons. So that has to be varying as well.

Prophet defines changepoints to capture changes in trend. So the logistic growth model becomes a piecewise logistic growth model, and it is defined as

$$g(t) = \frac{C(t)}{1 + exp(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma)))}$$

where

$$a_j(t) = \begin{cases} 1, & if\ t \geq s_j, \\ 0, & otherwise \end{cases}$$

Linear Trend

A piecewise linear model [29] is used where growth is linear, and trend is modeled as:

$$g(t) = (k + a(t)^T\delta)t + (m + a(t)^T\gamma)$$

where k is the growth rate, δ the rate adjustments, m is the offset parameter.

Automatic Changepoint Selection

The changepoints could be manually specified, or it can be automatically selected by Prophet. A vector rate of adjustments is defined, $\delta \in R^S$ where the changes happen, in both the models, piecewise logistic growth model and piecewise constant growth model. Changepoints are detected through passing these points in δ through Laplacian distribution ($\delta_j \sim Laplace(0, \tau)$), where τ controls the flexibility of growth rate.

Algorithm 2 Prophet algorithm [1]

Require: C = carrying capacity
k = growth rate
m = offset parameter
$a(t)^T\delta$ = cumulative growth till changepoints s_j

Steps

1: Model growth similar to growth in natural ecosystems, i.e. logistic growth model

$$g(t) = \frac{C}{1 + exp(-k(t-m))}$$

2: Incorporate trend changes in the growth model by explicitly defining changepoints where the growth rate is allowed to change.

$$\text{growth rate at t } = k + a(t)^T\delta$$

3: Modify the original logistic growth model to incorporate trend changes for non-linear, saturating growth as

$$g(t) = \frac{C(t)}{1 + exp(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma)))}$$

and for linear growth as

$$g(t) = (k + a(t)^T\delta)t + (m + a(t)^T\gamma)$$

4: Define $\delta \in R^S$ where points in δ are rate of adjustments in g(t).

5: Extract changepoints by putting δ through Laplace distribution

3.3 Pruned Exact Linear Time

PELT is based on the algorithm Optimal Partitioning Dynamic Programme [20], but involves a pruning step within it. Optimal partitioning dynamic programme uses dynamic programming to achieve that. It looks at the last changepoint,

dividing the data into two segments, one before the (last) changepoint and one after the changepoint. It then calculates the cost function (usually log likelihood) of both the segments, adds them up, then adds a constant β to guard against overfitting. The cost of both the segments is then compared to the cost of the whole segment, and if the cost of sum of both individual segments (and constant β) is less than the cost of whole segment, the point is considered as a changepoint. Then, it looks at the segment prior to the last changepoint, finds another (last) changepoint, and repeats the process. Using recursion to calculate cost function of segments, the changepoints are determined.

PELT [2] prunes the search space of potential last changepoints. Instead of considering all the time points prior to the current time point as potential last changepoint locations, PELT instead considers a subset of these time points. Specifically those that are no more than β away from the optimal at previous time points. In this way at each iteration it prunes the list of potential previous time points, only keeping those that are within β of the optimal.

More formally, let $y_{1:n} = y_1, ..., y_n$ be an ordered sequence of data, with m number of changepoints, $\tau_{1:m} = (\tau_1 .... \tau_m)$. Changepoint positions are between 1 and n - 1. The m changepoints divides the sequence into m+1 segments, and i^{th} segment will contain data $y(\tau_{i-1} + 1) : \tau_i$. C is the cost function, which in this case is twice the negative log likelihood [30], but other cost functions can also be used, like quadratic loss [31] or cumulative sums. β is the penalty to guard against overfitting, which can be Akaike's information criterion [32] or Schwarz information criterion [33].

Optimal partitioning looks to minimize, using a search method, the following.

$$\sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta]$$

Let F(s) denote the minimization of the above equation for data $y_{i:s}$ and $\tau_s = \tau_0 < .... < \tau_{m+1} = s$ be a vector of all possible segmentations with m changepoints. Finally, set F(0) to be the $-\beta$. Then,

$$F(s) = \min_{\tau \epsilon \tau_s} \{ \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] \}$$

This step represents dividing the whole data segment into smaller segments and calculating the minimum cost of individual segments.

$$F(s) = \min_{t} \{ \min_{\tau \epsilon \tau_t} \sum_{i=1}^{m} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] + C(y_{(t+1):n}) + \beta \}$$

This step represents division of data segment into two parts after determining the (last) changepoint of the data, one before the changepoint which will again be divided in two parts, and one after the changepoint.

$$F(s) = \min_{t} \{F(t) + C(y_{(t+1):n}) + \beta\}$$

This step shows that minimization of F(s) can be represented as F(t) and cost function of last data segment. This is dynamic programming, dividing the dataset into two parts, applying minimization of cost function to each part, dividing dataset again and repeating.

This provides a recursion which gives the minimal cost for the data $y_{1:s}$ in terms of the minimal cost for data $y_{1:t}$ for $t<s$. This recursion can be solved in turn for s = 1,2,...n. The optimal partitioning dynamic program can be shown in terms of algorithm as following:

Algorithm 3 Optimal partitioning algorithm [20]

Require: A set of data of the form $y_1, y_2, ...y_n$ where $y_i \epsilon$ R.
A cost function C
A penalty constant β that guards against overfitting.

Initialize

Let n be the length of the data, and set F(0) = -β, cp(0) = NULL

Iterate

for τ^* = 1,...,n

1: Calculate $F(\tau^*) = min_{0 \leq \tau < \tau^*}[F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta]$

2: Let $\tau^\wedge = arg\{min_{0 \leq \tau < \tau^*}[F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta]\}$

3: set $cp(\tau^*) = (cp(\tau^\wedge), \tau^\wedge)$

return the changepoints recorded in cp(n).

PELT removes those values of changepoints (τ) which can never be minima from the minimization performed at each iteration in optimal partitioning algorithm. It makes use of the following condition for that [64].

It is assumed that when introducing a changepoint into a sequence of observations the cost, C, of the sequence reduces. More formally, it is assumed there exists a constant K such that for all $t<s<T$,

$$C(y_{(t+1):s}) + C(y_{(s+1):T}) + K \leq C(y_{(t+1):T})$$

Then if

$$F(t) + C(y_{(t+1):s}) + K \geq F(s)$$

holds, at a future time $T>s$, t can never be the optimal last changepoint prior to T. It means that if there is a changepoint s between changepoint t and last point, t can never be a good candidate to be selected as a changepoint at that time, t is discarded and search space is pruned.

Algorithm 4 PELT algorithm [2]

Require: A set of data of the form $y_1, y_2, ...y_n$ where $y_i \epsilon$ R.
A cost function C.
A penalty constant β that guards against overfitting.
A constant K

Initialize

Let n be the length of the data, set F(0) = $-\beta$, $cp(0) = NULL$, $R_1 = \{0\}$

Iterate for $\tau^* = 1,...,n$

1: Calculate $F(\tau^*) = min_{0\leq\tau<\tau^*}[F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta]$

2: Let $\tau^\wedge = arg\{min_{0\leq\tau<\tau^*}[F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta]\}$

3: set $cp(\tau^*) = (cp(\tau^\wedge), \tau^\wedge)$

4: Set $R_{\tau^*+1} = \{\tau \epsilon R_{\tau^*} \cup \{\tau^*\} : F(\tau) + C(y_{\tau+1:\tau^*}) + K \leq F(\tau^*)\}$

return the changepoints recorded in cp(n).

4 Methodology

This section contains an explanation of the methodology that was used to structure this study, followed by the explanation of evaluation metrics on which the model comparison will be focused. The chapter ends with the tool selection.

4.1 Research method

The main goal of the thesis is to conduct a benchmark, to compare the performance of the frameworks. First step is to prepare the data, with known anomalies and changepoints. Different types of datasets are prepared, which represent most of the scenarios where anomaly detection and changepoint detection can be applied, so as to encapsulate most probable scenarios of the real world. A real world dataset with known anomalies is also found. Each dataset containing known anomalies and changepoints is fed into the framework which will give the results, anomaly detection framework specifies which points are the anomalies and changepoint detection framework specifies which points are changepoints. Then a confusion matrix is prepared, which contains true positives, true negatives, false positives, and false negatives. Then the performance is evaluated based on different metrics as mentioned below. To summarize, following steps are needed to describe complete research method. They are based on CRISP-DM standard [34]

a) Problem understanding: This phase considers project objectives from a requirements perspective. Generated insights are transformed into a changepoint detection or anomaly detection problem definition.

b) Data understanding: During the data understanding phase, data is initially generated with known anomalies and changepoints (anomalies and changepoints are injected manually) and analyzed to get first insights and for accomplishing familiarity with the data.

c) Data preparation: This phase entails all of the steps undertaken to generate the final dataset which will serve as input to the changepoint and anomaly detection frameworks.

d) Implementation: In this step the data is fed as an input to the different changepoint and anomaly detection frameworks.

e) Evaluation: After implementation, the framework's performance has to be evaluated and compared. It is important to assess whether the goals, defined during the business understanding phase, are met.

f) Deployment: In order to actually benefit from the framework it needs to be deployed. This requires for the framework to be integrated in systems and fed with data, in order to gain valuable insights.

4.2 Evaluation metrics

This section describes the evaluation metrics used to compare the different frameworks. Resulting from the business understanding, the evaluation will be based on performance metrics.

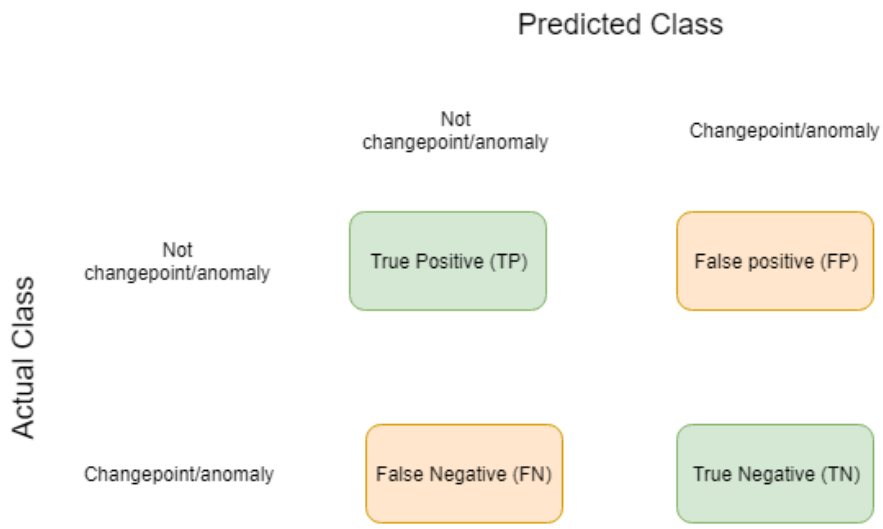

Figure 5: Confusion matrix, showing true positives, true negatives, false positives and false negatives.

Various measures exist to assess and compare the performance of frameworks on anomaly detection and changepoint detection task. These metrics are based on the confusion matrix, from which one can derive the correctly predicted cases, indicated in green, called true positives and true negatives. These are the cases where a changepoint/anomaly existed and was detected and changepoint/anomaly did not exist and was not detected. Also, the wrongly predicted cases can be identified, as indicated in orange, the false negatives, and the false positives, where a changepoint/anomaly existed but none was detected or where no anomaly/changepoint existed but was detected.

From the confusion matrix, various performance metrics can be derived. The most common metrics are accuracy and error. Accuracy describes the percentage of correct results, whereas the error rate is the number of wrongly classified results. Most models aim at achieving a high accuracy, or equivalently a low error rate. Accuracy is not always a good measure, especially not for all datasets. Better estimators which provide more information about the type of error, are precision, recall, and the F1-score. Precision, also called positive predict value, is the number of true positives divided by the number of all positive classified cases. The recall, also called sensitivity, is the number of true positives divided

by all positives in the data set. In the F1-score both recall and precision are considered equally. Another very important measure is the specificity which stands in contrast to the sensitivity and measures the proportion of negatives that are correctly identified as such.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$ErrorRate = \frac{Number\ of\ wrong\ predictions}{Total\ number\ of\ predictions} = \frac{FP + FN}{TP + TN + FP + FN}$$

$$Precision = \frac{True\ positives}{Total\ number\ of\ positive\ predictions} = \frac{TP}{TP + FP}$$

$$Recall = \frac{True\ positives}{False\ negatives\ + True\ positives} = \frac{TP}{TP + FN}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Specificity = \frac{True\ negatives}{True\ negatives + False\ positives} = \frac{TN}{TN + FP}$$

There is no static rule to estimate which metric is best suited for a anomaly/changepoint detection task, instead, it depends on the use case. For the case of this study, it is less important to identify every non-anomalous/non-changepoint instance. It is more important that the ones which are identified as anomalies/changepoints. This fact is expressed through the precision and the recall, which are therefore the most important measures. Nevertheless, most preferable are frameworks that also consider the accuracy and error rate, since there are situations, where it is important to find all cases belonging to the negative class.

4.2.1 Run time of frameworks

It is also important to measure the run time of the frameworks, how much time they take to perform the anomaly/changepoint detection task. This metric is not very important but cannot be ruled out, as it can give the estimate that even though an algorithm is better than the other at the given task, if the run time to analyze the data is too high, it is better to go with a not so accurate but faster alternative.

4.3 Tool selection

R is used for implementing the changepoint/anomaly detection algorithms. R is a general-purpose high-level programming language. It is used throughout the

statistics community also due to its many libraries that contain tools for easy data manipulation. The libraries used would be Prophet by Facebook, Anomaly Detection by Twitter and Changepoint by Dr. Rebecca Killick. These libraries contain the mentioned techniques for anomaly/changepoint detection. All these are official packages by the aforementioned parties.

5 Data description and understanding

This section goes through why the data was included, what kind of real world scenarios it covered, and understanding of the data. To have a comprehensive benchmark, both synthetic data and real world data was used. Several types of synthetic data were generated, each suited to anomaly detection or changepoint detection, and real world data was used as well. They are as follows.

5.1 Changepoint detection data

Different types of synthetic data generated are data with change in mean, change in variance, non-stationary data, data with change in mean and variance, and two synthetic datasets by Yahoo. These are as follows

5.1.1 Change in mean

Data with changing mean represent changepoints, intersection of data points with different means is exactly the point at which the change occurs. It looks like figure 6.

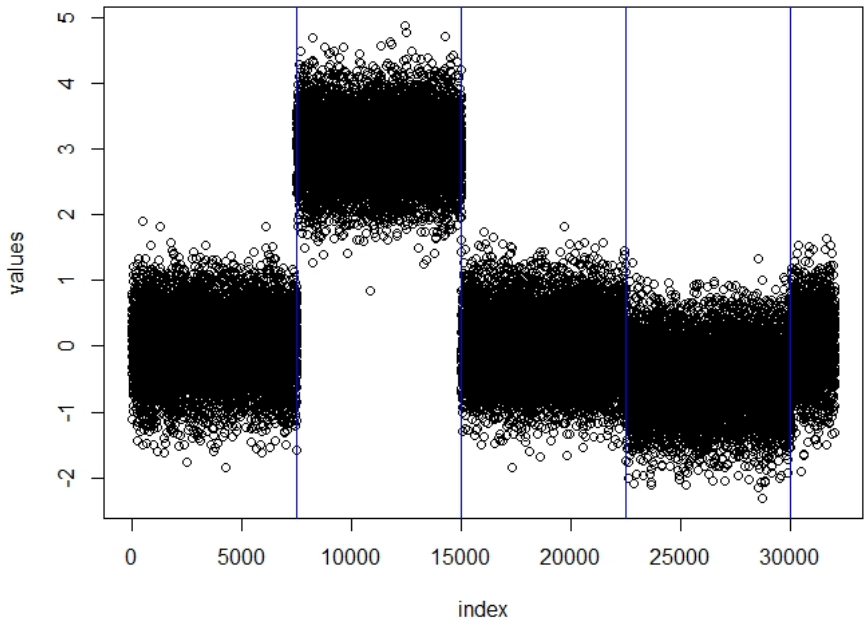

Figure 6: Data with change in mean, changepoints marked in blue.

In this example, data generated consisted of 35904 points, there is a change in mean after every 7500 points. The first 7500 points have a mean of 0, second 7500 points have a mean of 3, third 7500 points have a mean of 0, fourth 7500 points have a mean of -0.5, and last remaining points have a mean of 0.

5.1.2 Change in variance

Data with changing variance also represent changepoints. Changepoints are at the intersection of data points with different variance. The dataset looks like figure 7.

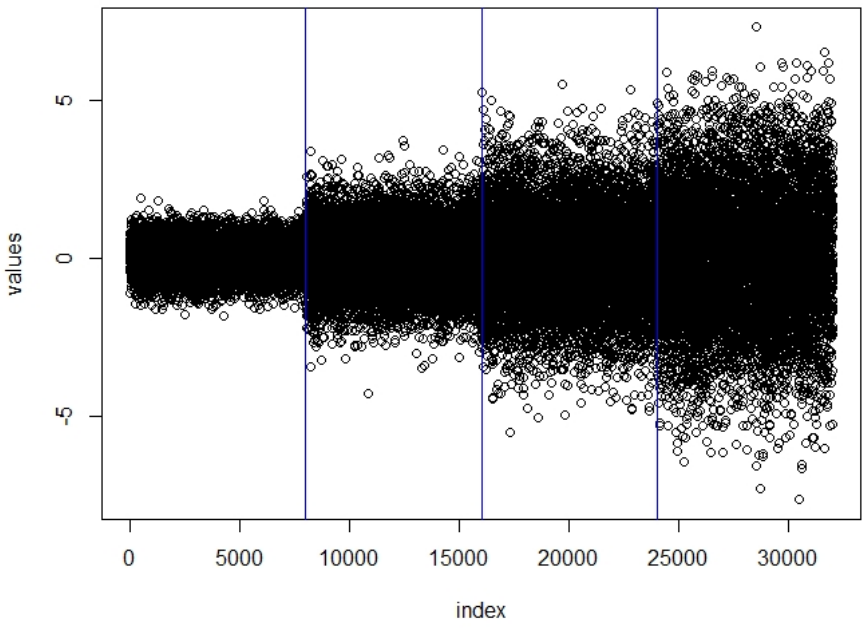

Figure 7: Data with change in variance, changepoints marked in blue.

The dataset consists of 32000 points, first 8000 points have a variance of 0.5, next 8000 points have a variance of 1, next 8000 points have a variance of 1.5, and last remaining points have a variance of 2.

5.1.3 Data with change in mean and variance

This dataset is a combination of data with change in mean, change in variance, and change in mean and variance. It contains a total of 7060 points. The first changepoint is at index 1000, when the mean changes. The second changepoint

is at index 2000, when the mean changes again. The next 4 changepoints are at 3000, 3500, 4000, 4500 where the variance changes. The next changepoint is at 6000, where the mean and variance change, and 6030, where the mean and variance change again. It looks like figure 8.

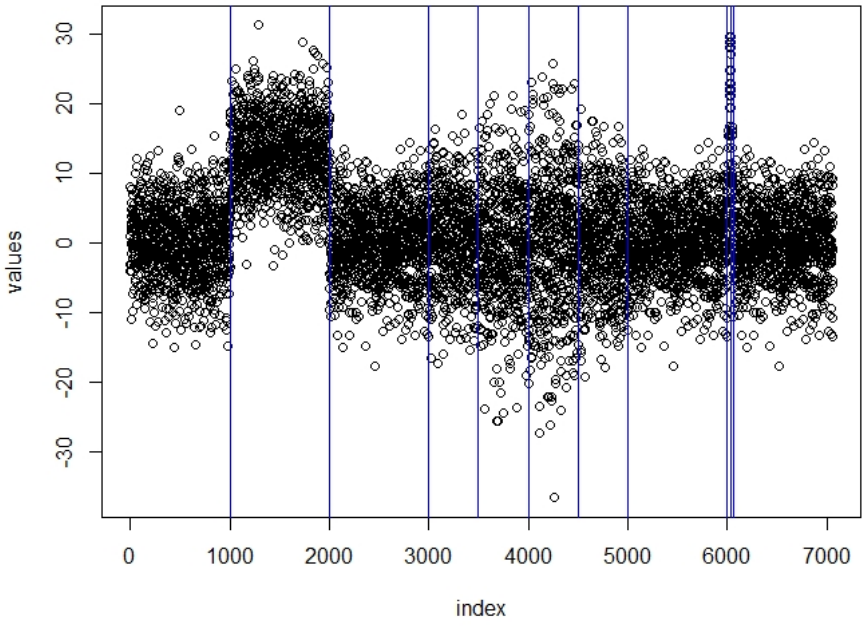

Figure 8: Data with change in mean, variance, mean and variance.

5.1.4 Yahoo Synthetic Data 1

This dataset has been provided by Yahoo, it is a synthetic dataset, that contains 4 changepoints. It consists of 1680 points. It looks like the figure 9.

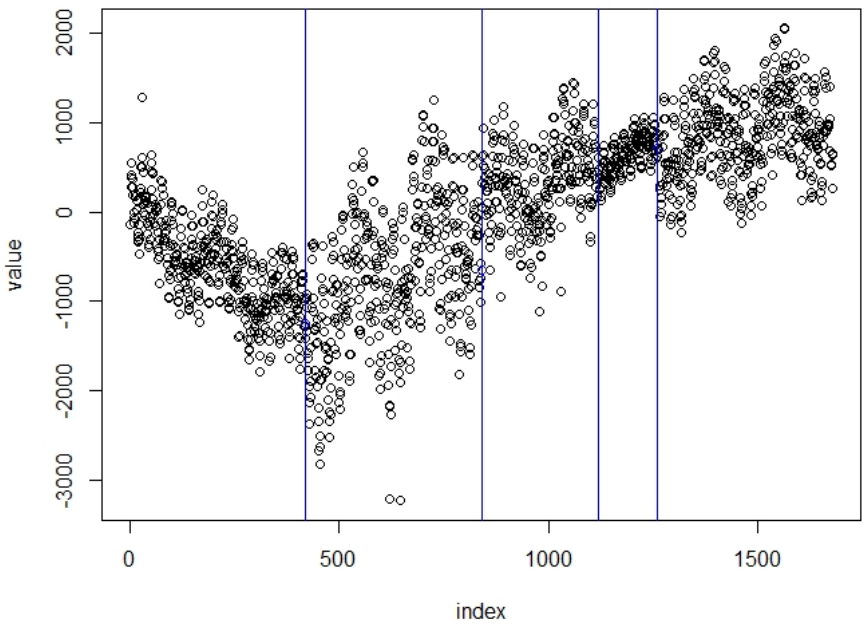

Figure 9: Synthetic Data by Yahoo. Changepoints marked in black

5.1.5 Yahoo Synthetic Data 2

This is another dataset that has been provided by Yahoo, it is a synthetic dataset, that contains 4 changepoints. It consists of 1680 points. It looks like figure 10.

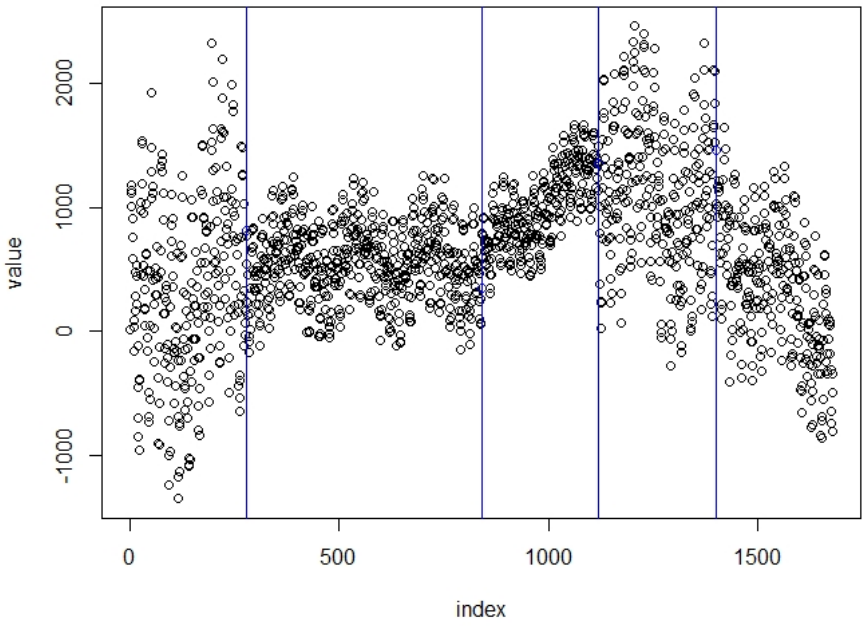

Figure 10: Synthetic data by Yahoo. Changepoints marked in black.

5.1.6 Non-stationary data

In non stationary data the mean and variance of the dataset are not static, they vary, along with other statistical properties. Which means every point is a changepoint, there is no concrete definition of a changepoint in non-stationary data. To have a frame of reference of changepoints for this data, the definition of changepoint is, if the mean of a segment of data is different from the mean of another segment, there is a changepoint between them. But, then there is a question of different by how much. To deal with this, three cases are considered. In the first case, the data is divided into really small number of segments, 3. This dataset contains 32000 points, and it looks like figure 11.

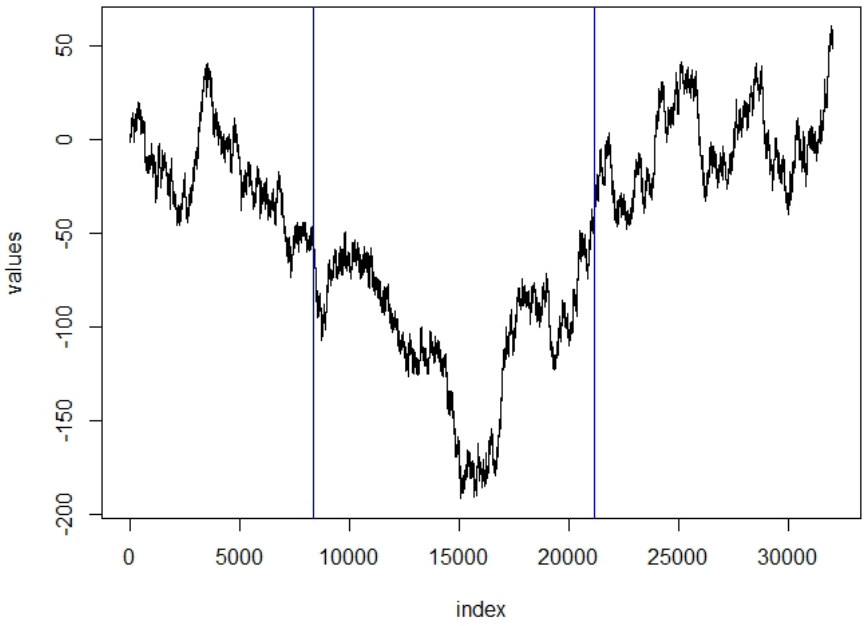

Figure 11: Non-stationary data with 2 changepoints.

In the second case, the dataset has been divided into 12 segments, with 11 changepoints. Which is more reasonable. It looks like figure 12.

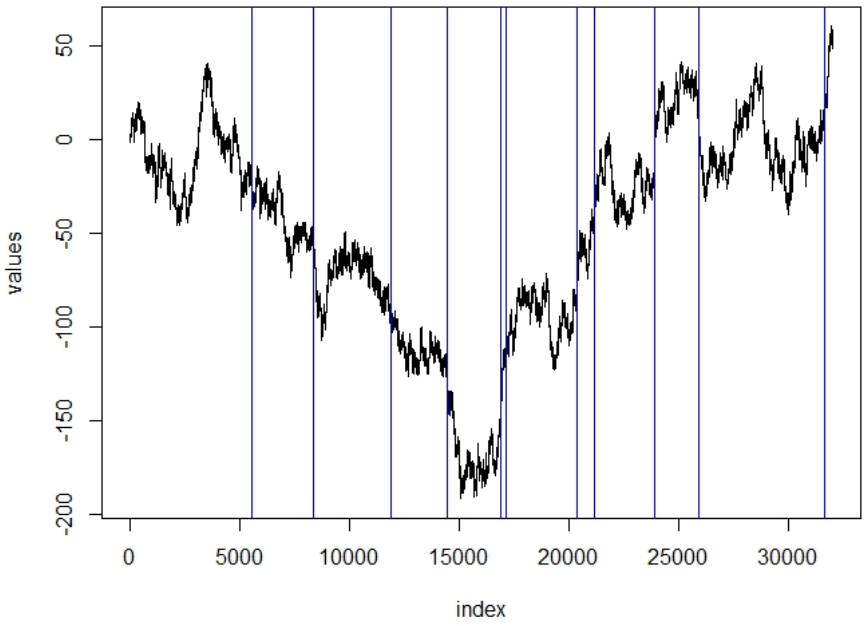

Figure 12: Non-stationary data with 11 changepoints.

In the last case, the data has been divided into a huge number of segments, 50, with 49 changepoints. It looks like figure 13.

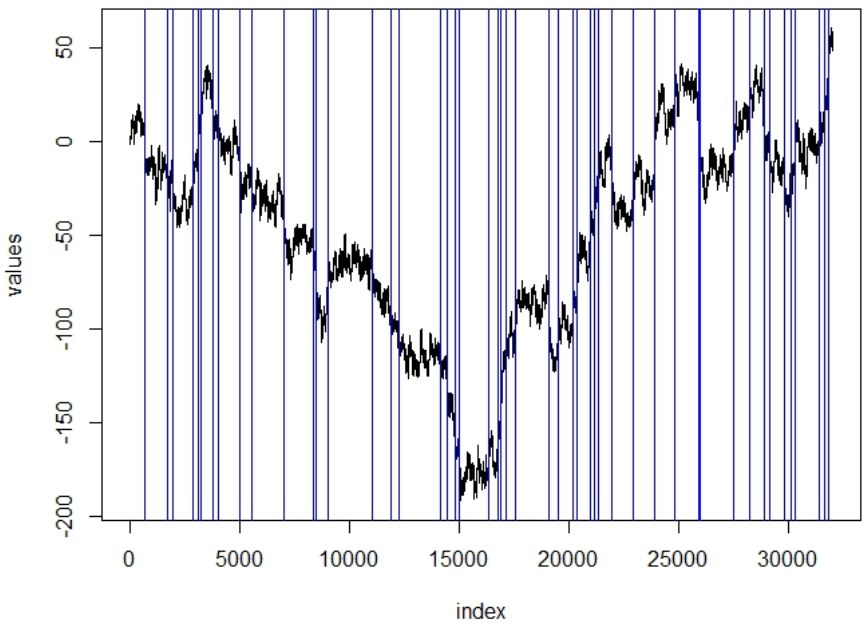

Figure 13: Non-stationary data with 49 changepoints.

Each case will serve as a frame of reference to how many actual changepoints exist in the dataset, and how the confusion matrix has to be derived.

5.2 Anomaly detection datasets

Several synthetic and one real world dataset has been used as a dataset for anomaly detection. Two synthetic and one real world datasets were used from Yahoo. Other datasets are outliers, stationary and sine wave. Some details about them are as follows.

5.2.1 Outliers

This dataset contains 32000 points. It is a constant stream of data with constant mean of 0. There are 200 anomalies injected into it, and are at least 3 standard deviation and at most 5 standard deviation away from the mean of the dataset. 3 is used as a lower bound because it is pretty widely used standard in outlier detection [45], usually points 3 standard deviations away from a certain statistic is considered an anomaly. 5 is chosen as an upper bound because we wanted to

make it challenging for the algorithm to detect anomalies, as a very high value of say 7 or 10 would result in points that can be easily detected. Number of injected anomalies are 200. It looks like figure 14.

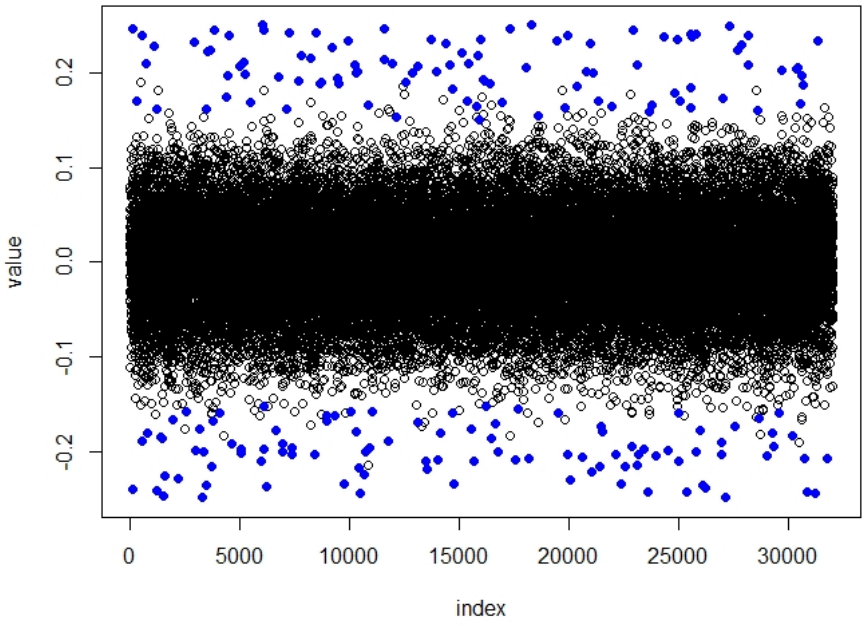

Figure 14: Outliers dataset, anomalies marked in blue.

5.2.2 Stationarity

A stationary data has a constant mean and variance, it doesn't vary over time. This dataset looks more like a real world data. It contains 32000 points. There are 200 anomalies, and they have been injected in the following way. 200 random points are chosen, and a window has been created 50 points prior to that point, to 50 points after that point. Then the mean and standard deviation of the segment in that window has been calculated, and anomalies are injected 3.5 to 5.2 standard deviations away from the mean of that segment, in both directions, positive and negative. The logic behind 3 and 5.2 is same as before, 3 standard deviations is widely used standard, more than which a point is considered an anomaly, and 5.2 is used so as make the anomaly detection process challenging for the algorithm. The data looks like figure 15.

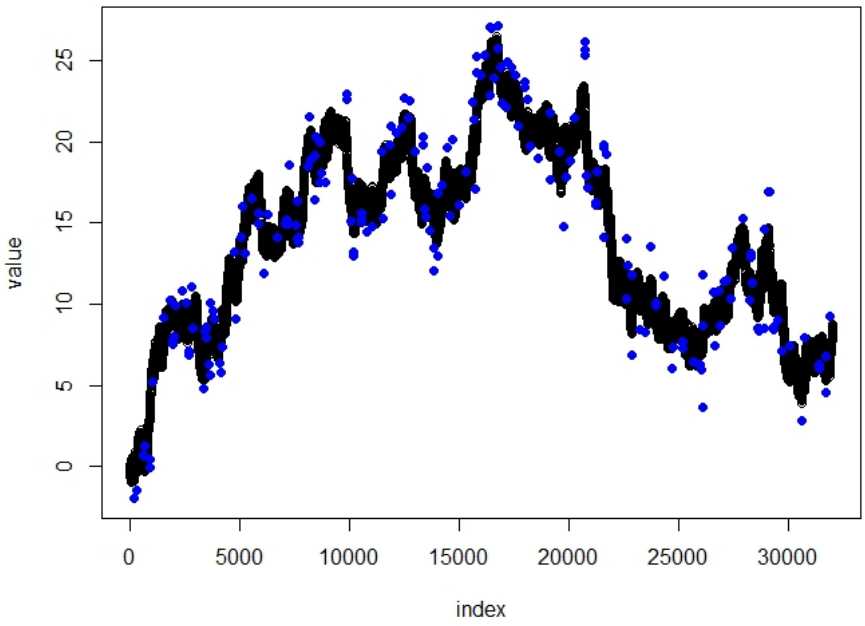

Figure 15: Stationary dataset, anomalies marked in blue

5.2.3 Sine Wave

This data is a sinewave, it contains 14000 points, and 200 anomalous points. Similar to stationary data, this data was divided into segments by randomly choosing 200 points, and creating a window 50 points prior and after that point, computing mean and variance of that segment, and injecting anomalies 3 to 5 standard deviations of that segment, away from the mean of that segment. The assumption is same as before, anomalies lie at least 3 standard deviations away, and 5 is chosen to make the anomaly detection task challenging. It looks like figure 16.

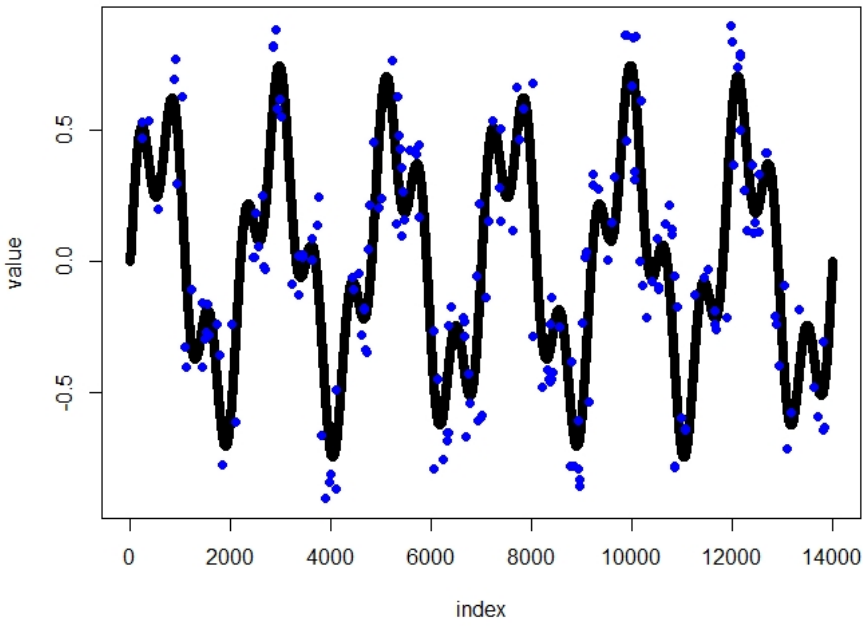

Figure 16: Sine wave, anomalies marked in blue

5.2.4 Yahoo Synthetic Data 1

This data has been taken from Yahoo, it is synthetically generated, and has 9 anomalies injected into it. It contains 1680 points, and looks like figure 17.

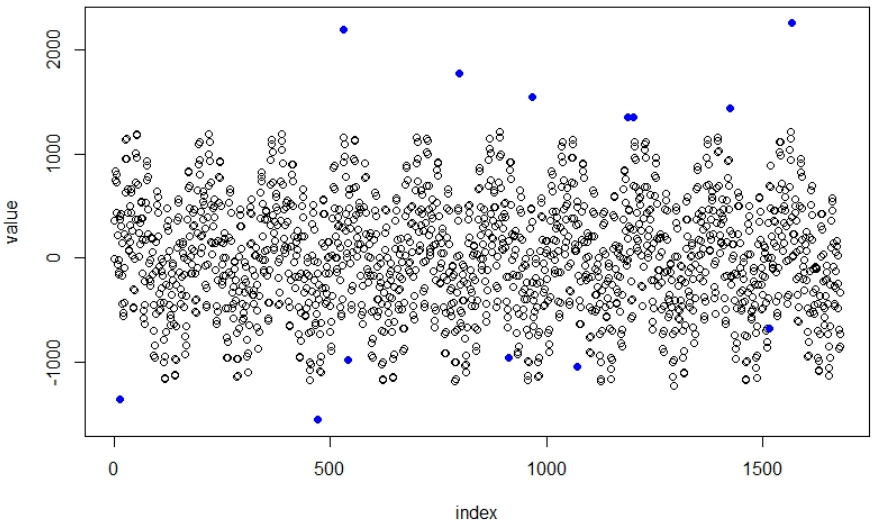

Figure 17: Synthetic data by Yahoo, anomalies marked in blue

5.2.5 Yahoo Synthetic Data 2

This data has also been taken from Yahoo, it is synthetically generated, and has 9 anomalies injected into it. It contains 1680 points, and looks like figure 18.

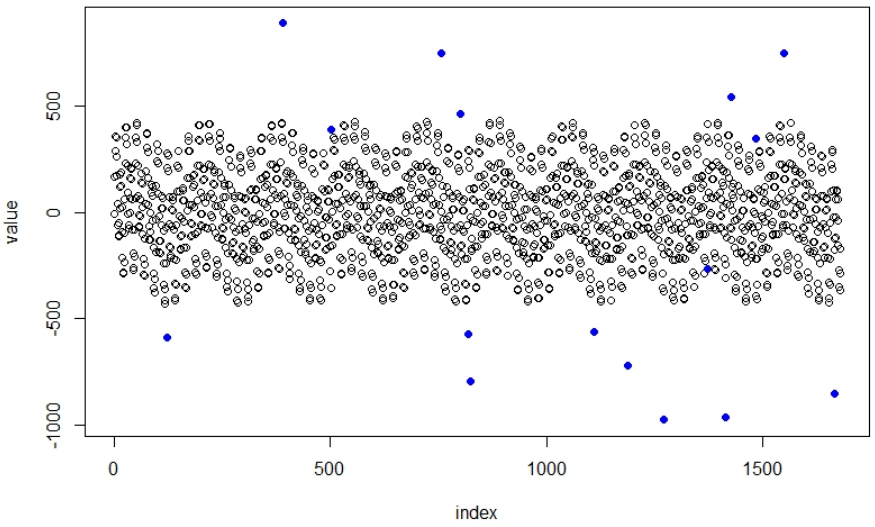

Figure 18: Synthetic data by yahoo, anomalies marked in blue

5.2.6 Real world data by Yahoo

This data has also been taken from Yahoo, it has 1461 data points, and it has 16 anomalies that have been marked by humans. It looks like figure 19.

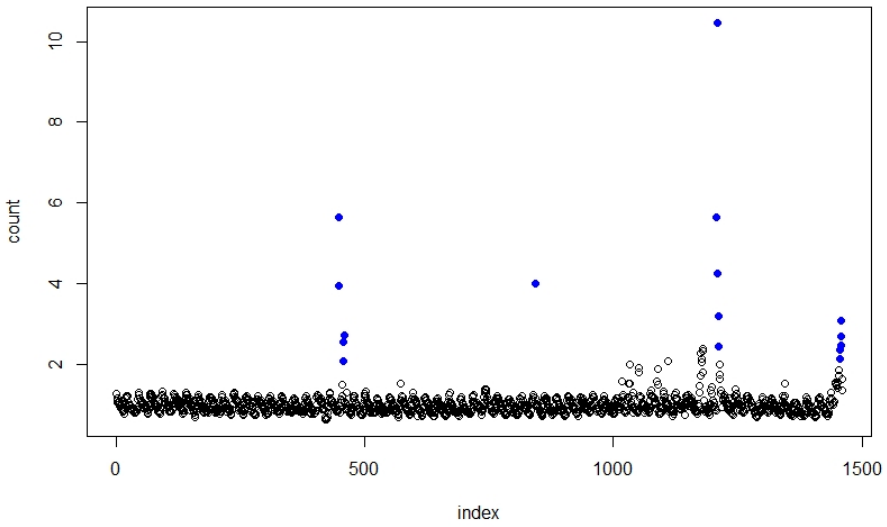

Figure 19: Real world data by Yahoo, anomalies marked in blue

6 Experiments and results

This section goes through implementation of each framework against each dataset. It is to be noted that every framework has been used with its default configuration, no changes were made to modify it, so as to keep the implementation generalized. Otherwise the implementation would have been too specific, which was to be avoided. The reason why parameters were kept at default to generalize the implementation is because generalization gives a wider scope for comparison to another implementation of the same algorithm with another dataset. If the parameters were tuned, the results could not be directly comparable, and they would be comparable to another implementation with the same parameters in it's implementation.

6.1 Changepoint's PELT

PELT, in its application, accepts following arguments.

a) Data: Input data

b) Penalty: The penalty method to guard against overfitting. This is the β value that is added to the cost function of individual segments. There is a choice of SIC (Schwarz information criterion), BIC (Bayesian information criterion), MBIC (Modified Bayes Information Criterion), AIC (Akaieki's information criterion), Asymptotic and manual. Default is MBIC, which was used here.

d) Method: Which algorithm to use for changepoint detection, namely PELT, BinSeg (binary segmentation), SegNeigh (segment neighborhood). There is no default, it has to be specified.

g) test.stat: Assumed test statistic/distribution of the data. Accepts Normal, Exponential, Gamma and Poisson. Default is Normal.

h) Minseglen: Minimum segment length, the number of observations between changepoints. Default for PELT is 1.

Application of PELT on the aforementioned datasets is as follows.

6.1.1 Dataset with change in mean

PELT was applied on dataset with change in mean, and the changepoints were detected as follows. There are 4 changepoints in total. Figure 20 shows the results.

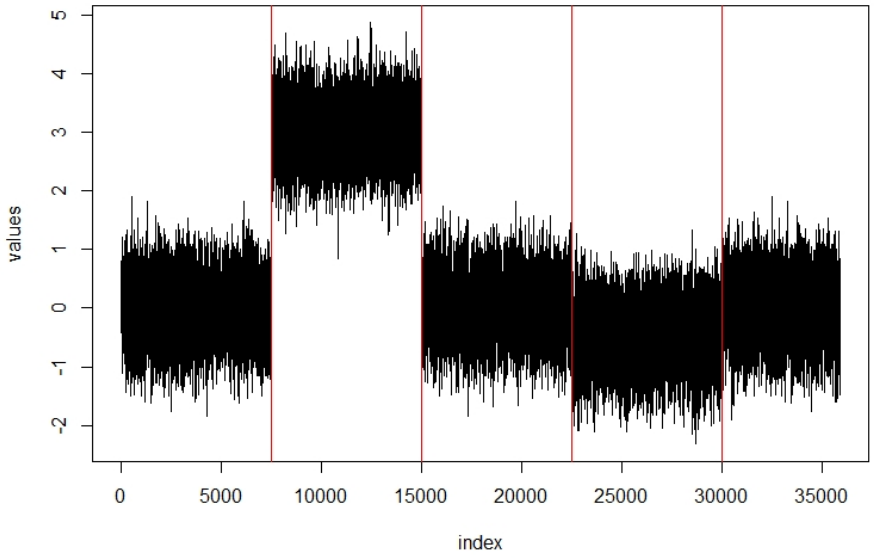

Figure 20: PELT on data with change in mean, changepoints detected are marked in red

PELT detects all the 4 changepoints accurately, with no false positives.

6.1.2 Dataset with change in variance

PELT was applied on dataset with change in variance, and the changepoints were detected as follows. There are 3 changepoints in total. Figure 21 shows the results.

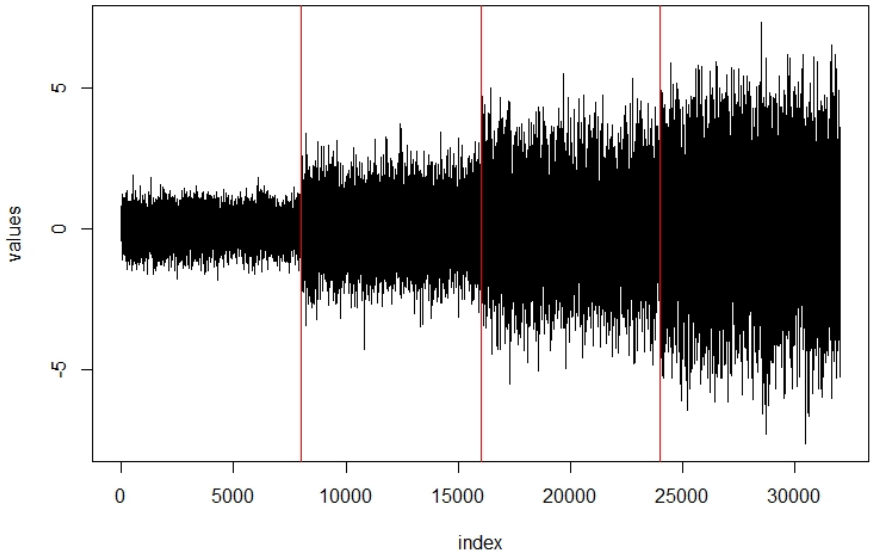

Figure 21: PELT on data with change in variance, changepoints marked in red.

PELT detects all 3 changepoints accurately, with no false positives.

6.1.3 Dataset with change in mean, variance, and mean and variance

PELT was applied to dataset with change in mean, variance, mean and variance, and the changepoints were detected as follows. There are 10 changepoints in total/ Figure 22 shows the results.

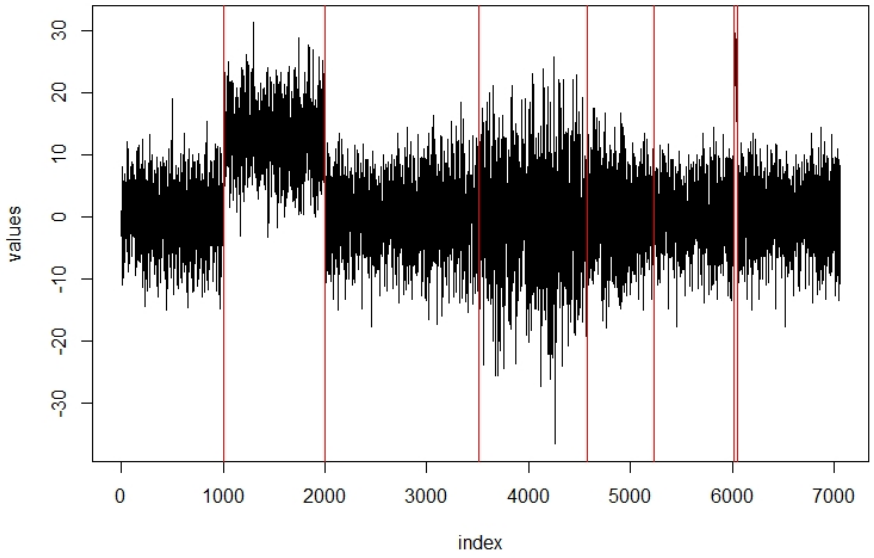

Figure 22: PELT on data with change in mean, variance, mean and variance, changepoints marked in red

PELT detects 7 changepoints correctly, but misses 3. Two of them are between data with change in variance, and one between data with increasing mean and variance and decreasing mean and variance. There were no false positives detected.

6.1.4 Synthetic dataset 1 by Yahoo

PELT was applied to Synthetic dataset by Yahoo, which has 4 changepoints, and results are shown in figure 23.

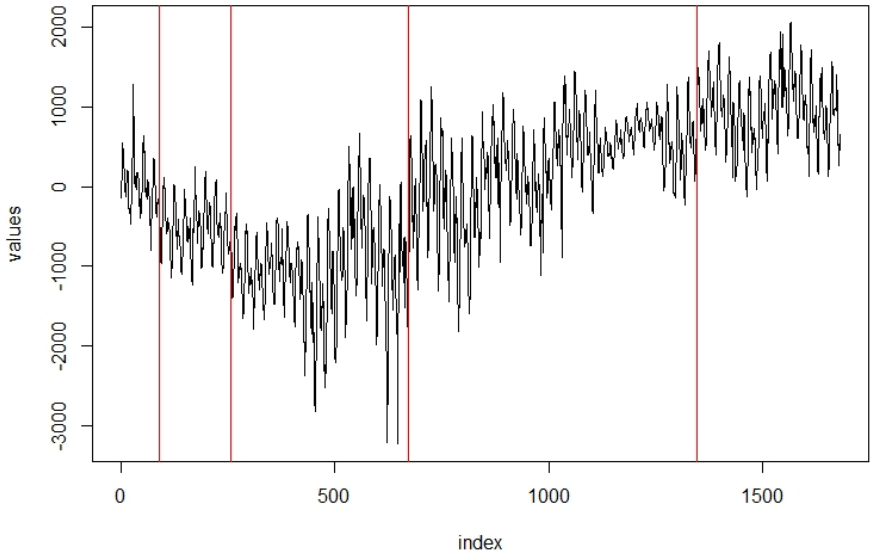

Figure 23: PELT on synthetic data 1 by Yahoo. Changepoints marked in red

Out of the 4 changepoints, no one is detected correctly. 4 false positives are detected.

6.1.5 Synthetic dataset 2 by Yahoo

PELT was applied on second synthetic data by Yahoo, which again has 4 changepoints, and they were detected as shown in figure 24.

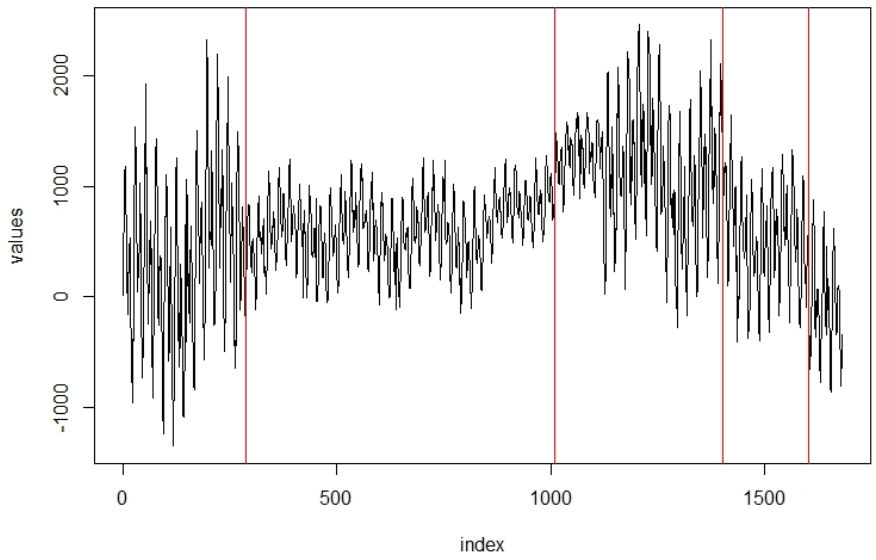

Figure 24: PELT on synthetic data 2 by Yahoo. Changepoints marked in red

PELT detected 2 changepoints correctly, but the other two were not detected. Also, 2 false positives were detected as well.

6.1.6 Non-stationary data

PELT was applied on non-stationary data and the changepoints were detected as shows in figure 25.

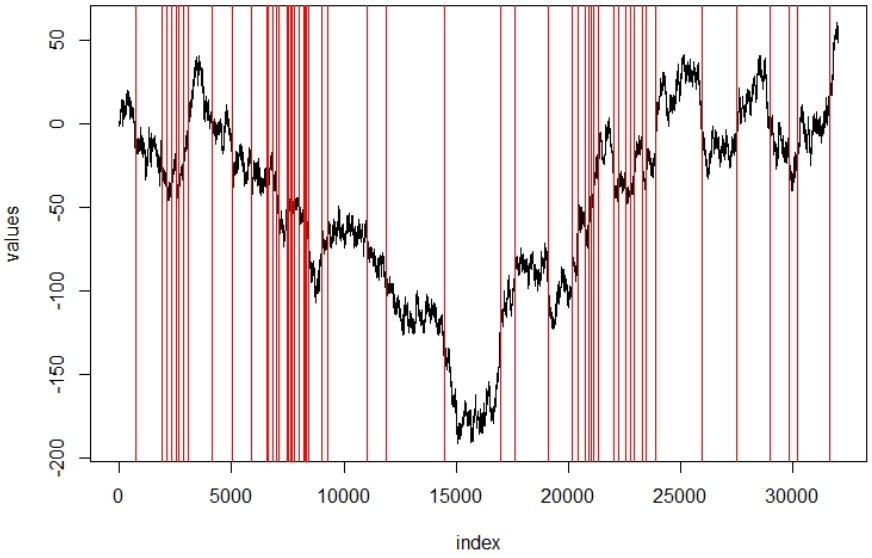

Figure 25: PELT on non-stationary data. Changepoints marked in red

PELT detected a lot of changepoints in non-stationary data, but due to the nature of non-stationary data, there is no correct statistical interpretation of where the changepoints are in the data, because every point is a changepoint in non-stationary data. This is because by definition, the mean, variance and other statistical properties of data are different between two sets of points. So, in order to have a minimal definition of relevant changepoints, human interpretation of where the changepoints are has been chosen. But, that may vary from person to person, so three three ground truths to number of true positives, false positives, true negatives and false negatives have been selected here. One of them is, there are 2 changepoints in the data, another one is, there are 11 changepoints in the data, and last one is, there are 49 changepoints in the data. To start with, if the ground truth is considered as there are 2 changepoints, PELT detected those 2 changepoints, but 53 false positives were detected as well. When the ground truth is considered as there are 11 changepoints in the data, PELT detects 7 of them correctly, with 44 false positives and 4 false negatives. When the ground truth is considered as the dataset having 49 changepoints, PELT detects 22 of them correctly, with 27 false positives and 26 false negatives.

6.2 Facebook's Prophet

Prophet accepts the following arguments in its application.

a) Data: Input data

b) Changepoint.range: Range of data upon which changepoint detection method has to be applied. 0.1 means method is to be applied to 10% of data. 1 means method is to be applied to 100% of data.

c) Add_changepoints_to_plot: This method adds the changepoints to the data and plots them.

Several parameters, which do not have to be specified by default, are as follows.

d) Growth: linear' or 'logistic' to specify a linear or logistic trend

e) Changepoints: List of dates at which to include potential changepoints (automatic if not specified)

f) n_changepoints: If changepoints in not supplied, you may provide the number of changepoints to be automatically included

g) Changepoint_prior_scale: Parameter for changing flexibility of automatic changepoint selection

Application of Prophet on datasets is as follows.

6.2.1 Data with change in mean

Prophet was applied to data with change in mean and the results are as follows. There are 4 changepoints in total, as shown in figure 26.

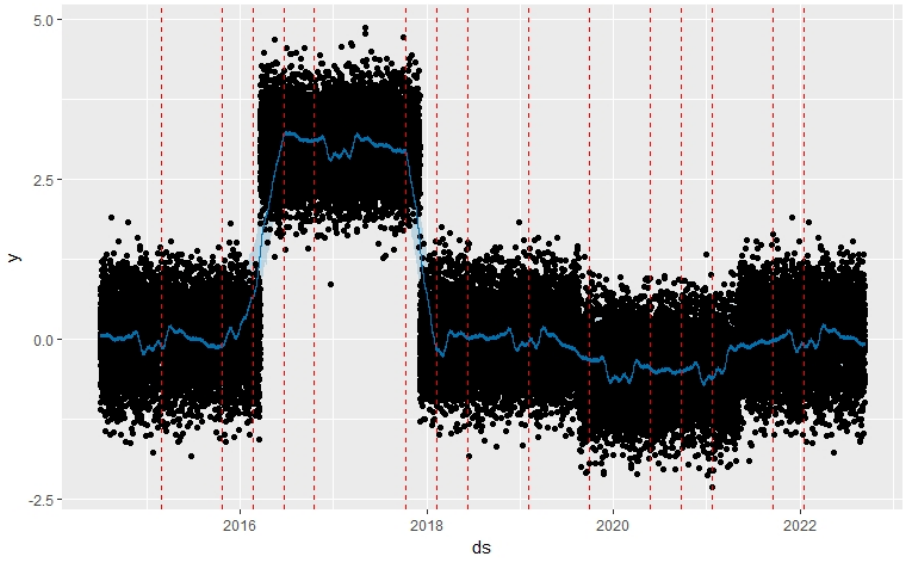

Figure 26: Prophet on data with change in mean, changepoints marked in red.

Out of 4 changepoints, 2 are detected correctly, 13 false positives are detected, and 2 points which in ground truth are changepoints are not detected. Results shown in figure 26.

6.2.2 Data with change in variance

Prophet was applied to data with change in variance, and changepoints are as follows. There are 3 changepoints in total, results shown in figure 27.

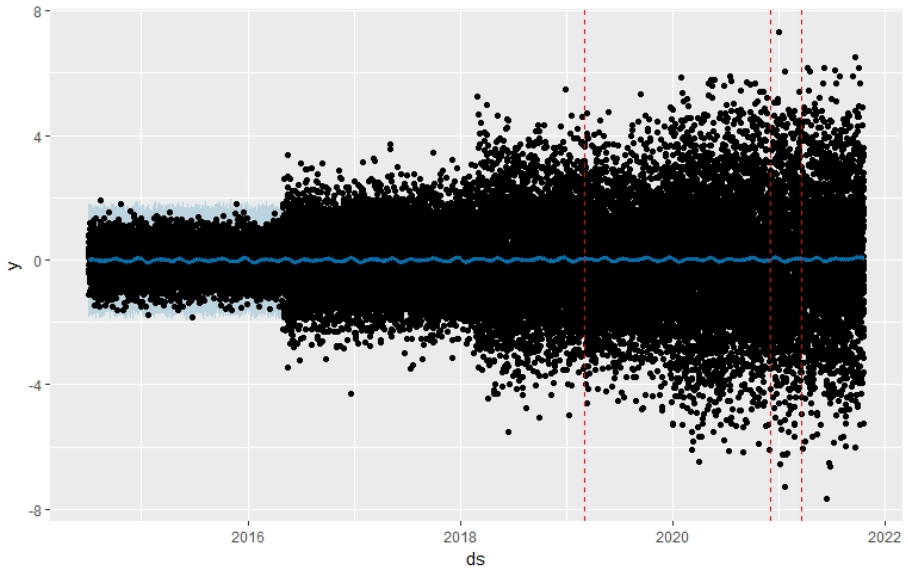

Figure 27: Prophet on data with change in variance, changepoints marked in red.

Out of three changepoints, none were detected correctly. 3 false positives were detected.

6.2.3 Data with change in mean, variance, mean and variance

Prophet was applied to data with change in mean, variance, mean and variance, and results are as follows.. There are 10 changepoints in total, results shown in figure 28.

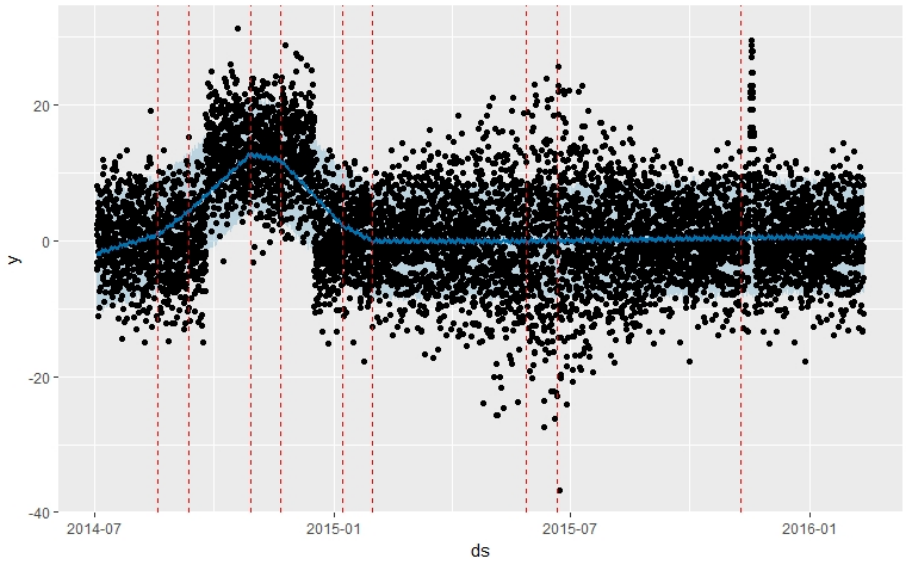

Figure 28: Prophet on data with change in mean, variance, mean and variance. Changepoints marked in red.

Out of 10 changepoints, 1 was detected correctly. 8 false positives were detected. 9 points which in ground truth are changepoints were not detected.

6.2.4 Synthetic Dataset 1 by Yahoo

Prophet was applied to one synthetic data by Yahoo, and results are as follows. There are 4 changepoints in this dataset, results shown in figure 29.

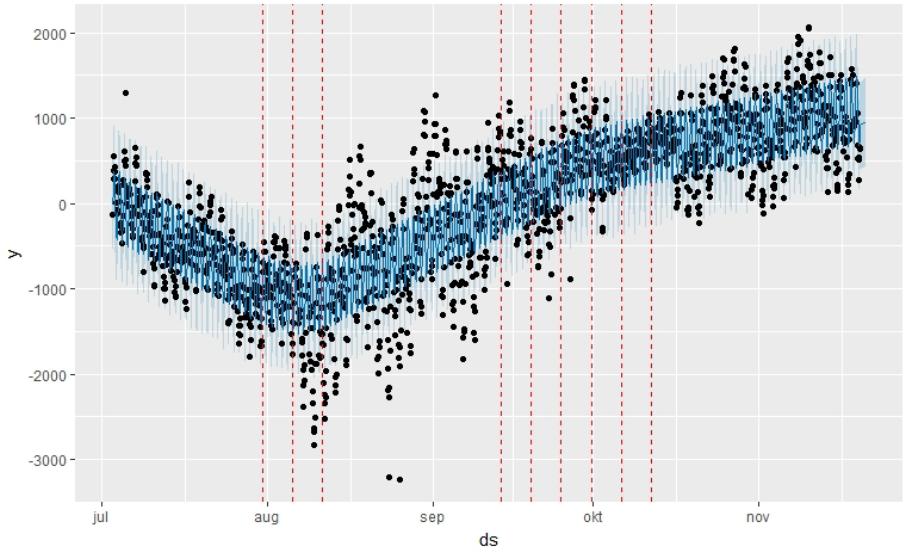

Figure 29: Prophet on synthetic data 1 by Yahoo. Changepoints marked in red.

Out of 4 changepoints, 2 were detected correctly, and 2 points which in ground truth are changepoints are not detected. 7 false positives were detected.

6.2.5 Synthetic Dataset 2 by Yahoo

Prophet was applied to another synthetic data by Yahoo, and results are as follows. There are 4 changepoints in this dataset, results shown in figure 30.

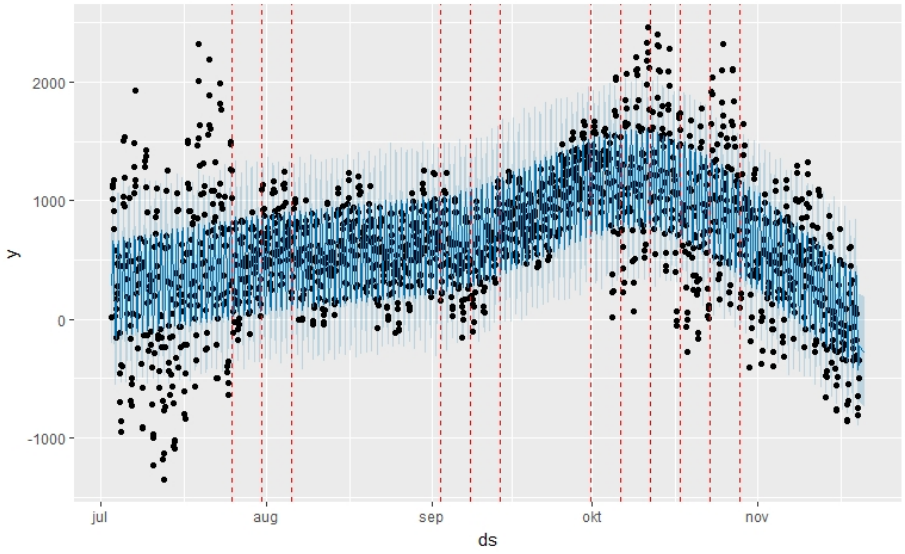

Figure 30: Prophet on synthetic data 2 by Yahoo. Changepoints marked in red.

Out of the 4 changepoints, all 4 of them are detected correctly. But, 8 false positives are detected as well.

6.2.6 Non-stationary data

Prophet was applied to non-stationary dataset, and changepoints detected are as follows, results shown in figure 31.

Figure 31: Prophet on non-stationary dataset. Changepoints marked in red.

Prophet detected a lot of changepoints in non-stationary data, but due to the nature of non-stationary data, there is no correct statistical interpretation of where the changepoints are in the data, because every point is a changepoint in non-stationary data. 3 ground truths to number of true positives, false positives, true negatives and false negatives have been selected here. One of them is, there are 2 changepoints in the data, another one is, there are 11 changepoints in the data, and last one is, there are 49 changepoints in the data. To start with, if the ground truth is considered as there are 2 changepoints, Prophet detected those 0 changepoints, and 2 false positives were detected. When the ground truth is considered as there are 11 changepoints in the data, Prophet detects 3 of them correctly, with 9 false positives. When the ground truth is considered as the dataset having 49 changepoints, Prophet detects 8 of them correctly, with 17 false positives and 41 false negatives.

6.3 Twitter's anomaly detection

Anomaly detection, in its application, accepts the following parameters.

a) data: Input data

b) max_anoms: Maximun number of anomalies to be detected.

c) direction: Direction in which to detect anomalies. Both means detect anomalies in both upwards direction and downwards direction.

d) only_last: This parameter specifies if the anomalies have to be determined in last 24 hours (or any other value supplied).

e) plot: This parameter specifies whether anomalies are to be plotted.

Anomaly detection is inherently different than changepoint detection. In case of Twitter anomaly detection, the amount of anomalies have to be manually specified. This can range from 1% to 49.9%/. To perform a comprehensive benchmark, three thresholds are selected, starting from 1% to 49%. This would encompass all the points that could be counted as anomalies, with the anomalies that are detected when threshold is 1% have more weight, and anomalies that are detected when the threshold is 49% have less weight. This is why in this thesis for each dataset, the amount of anomalies supplied are 1%, 25%, and 49%. Each of them are described below.

6.3.1 Outliers dataset

Anomaly detection was applied to dataset with outliers, that contains 200 anomalies, and the anomalies detected are as follows.

2) Anomalies when threshold is 1%, 25% and 49% are detected as shown in figure 32.

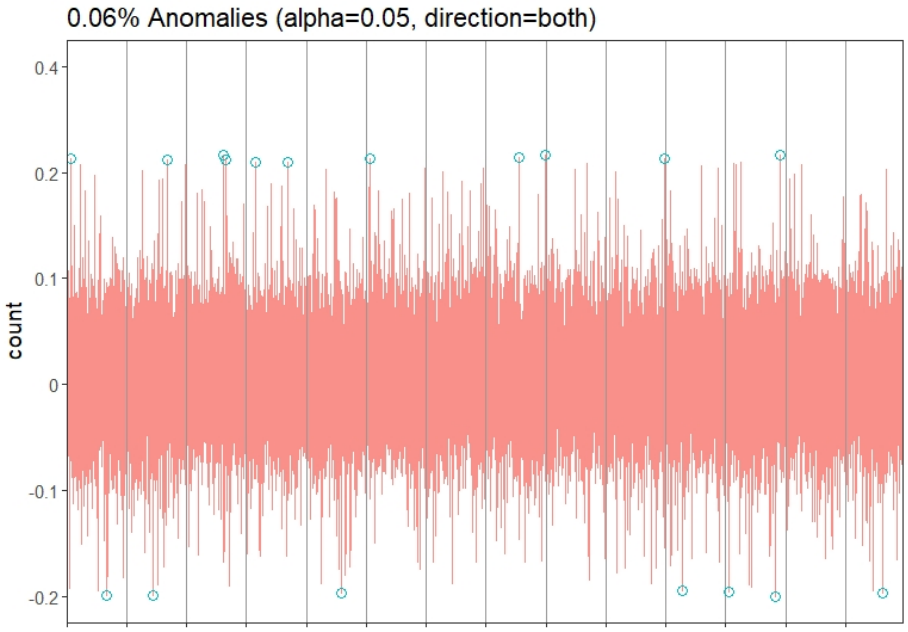

Figure 32: Twitter AD on data with outliers, anomalies marked in blue

When the threshold is 1%, 25% or 49%, out of 200 anomalies, 18 of them are detected correctly, and rest of them are not detected. The results were the same at every threshold.

6.3.2 Sine wave

Anomaly detection was applied to a sine wave, that contains 200 anomalies, and no anomalies were detected at the threshold of 1%, 25% and 49%.

6.3.3 Stationary data

Anomaly detection was applied to a stationary dataset, that contains 200 anomalies, and no anomalies were detected at the threshold of 1%, 25% and 49%.

6.3.4 Synthetic data 1 by Yahoo

This synthetic dataset by Yahoo contains 9 anomalies. When the threshold is 1%, 25% or 49%, anomalies detected are shown in figure 33.

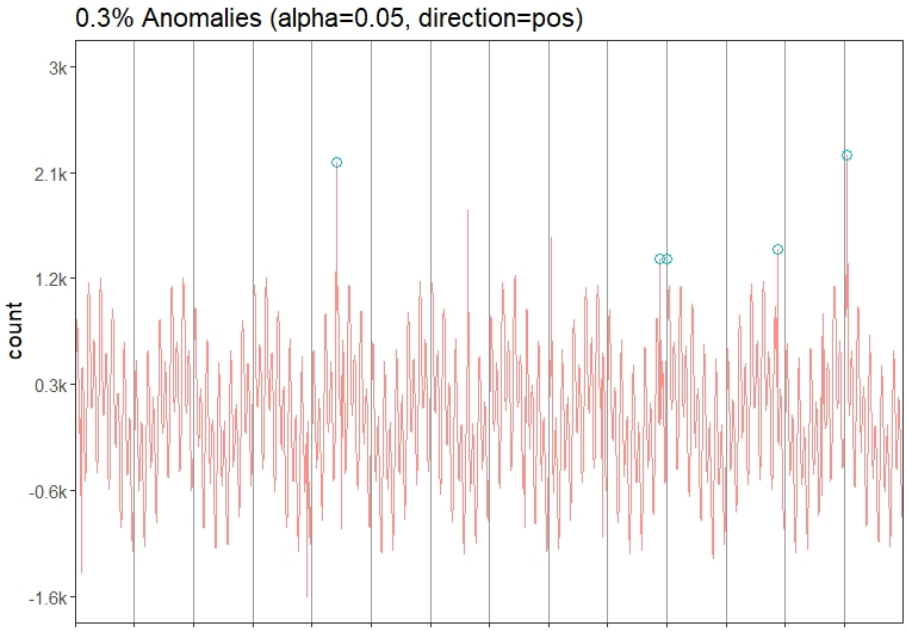

Figure 33: Twitter AD on synthetic data 1 by Yahoo. Anomalies marked in blue

Out of 9 anomalies, 5 of them are detected correctly, and 4 of them are not detected.

6.3.5 Synthetic data 2 by Yahoo

This is another synthetic dataset by Yahoo that contains 9 anomalies. When the threshold is 1%, 25% or 49%, anomalies detected are shown in figure 34.

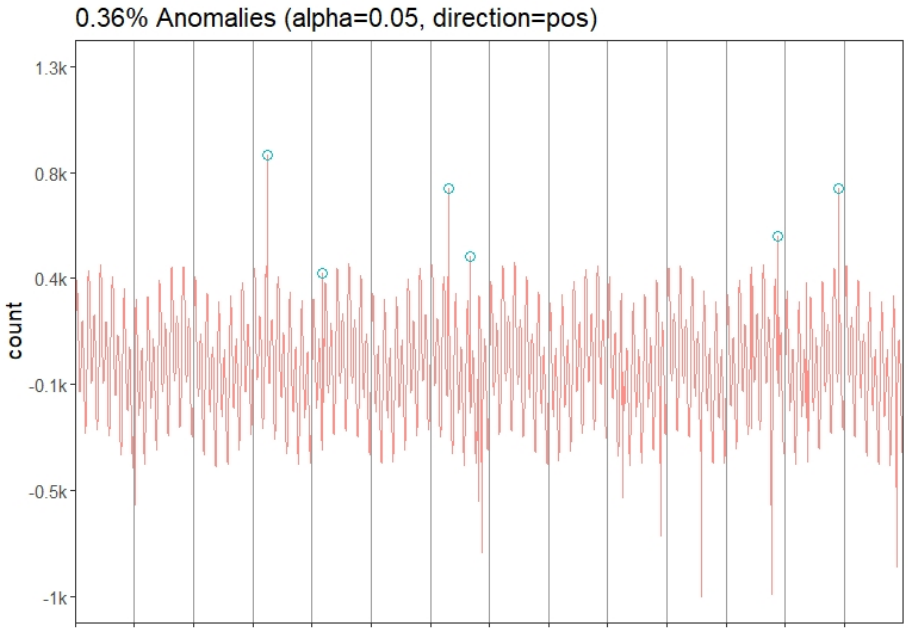

Figure 34: Twitter AD on synthetic data 2 by Yahoo, anomalies marked in blue

Out of 9 anomalies, 6 of them are detected correctly, and 3 of them are not detected.

6.3.6 Real world Yahoo! data

This dataset had anomalies marked by humans, contains 1461 points and 16 anomalies, and when anomaly detection was applied to this dataset, following anomalies were detected.

1) Anomalies when threshold is 1%

When the threshold was set to 1%, following anomalies were detected, as shown in figure 35.

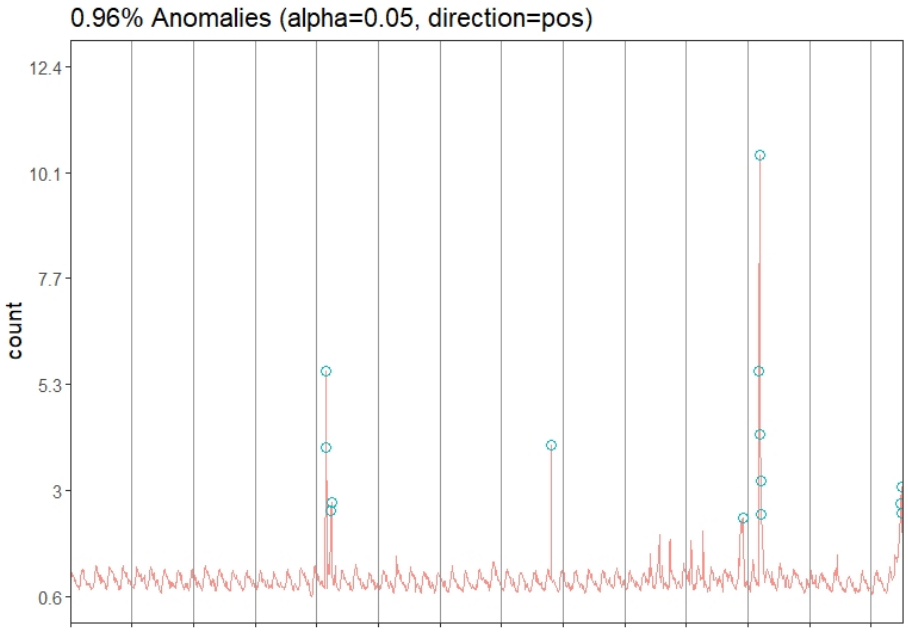

Figure 35: Twitter AD on real world dataset, threshold 1%, anomalies marked in blue

Out of 16 anomalies, 13 of them were detected and no false positives were detected.

2) Anomalies when threshold is 25% and 49%.

When the threshold was set to 25% and 49%, following anomalies were detected, as shown in figure 36.

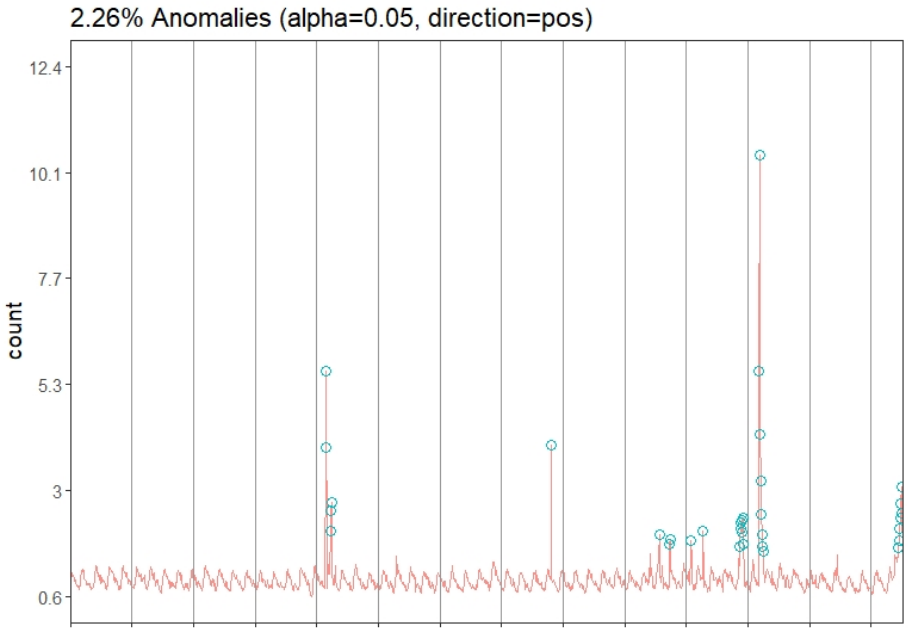

Figure 36: Twitter AD on real dataset, threshold 25% and 49%, anomalies marked in blue

In both the cases, 16 out of 16 anomalies are detected, but 17 false positives are detected as well.

6.4 Comparison

6.4.1 PELT and Facebook's Prophet

Following are the tables of performance metrics of PELT and Prophet, metrics being Accuracy, error rate, precision, recall, specificity and f-measure. For non-stationary data, the three cases are included, when there are 2 changepoints, when there are 11 changepoints and when there are 49 changepoints.

Prophet and PELT							
Dataset	Algorithm	Accuracy	Error rate	Precision	Recall	Specificity	F-measure
Change in Mean	Prophet	0.999582	0.000418	0.13333	0.5	0.999638	0.210526
	PELT	1.0	0	1	1	1	1
Change in Variance	Prophet	0.999812	0.000187	0	0	0.999906	N/A
	PELT	1	0	1	1	1	1
Mix	Prophet	0.997594	0.002405	0.11111	0.1	0.998866	0.105263
	PELT	0.999291	0.000708	0.875	0.63	0.999858	0.736842
Yahoo Synthetic 1	Prophet	0.994665	0.005334	0.22222	0.5	0.995840	0.3076923
	PELT	0.995249	0.004750	0	0	0.997619	N/A
Yahoo Synthetic 2	Prophet	0.995260	0.004739	0.33333	1	0.995249	0.5
	PELT	0.997621	0.002378	0.5	0.5	0.998808	0.5
Non-stationary	Prophet (2)	0.99987	0	0	0	0.999937	N/A
	PELT (2)	0.99834	0.001653	0.03636	1	0.998346	0.070175
	Prophet (11)	0.999031	0.000968	0.25	0.12	0.999718	0.162162
	PELT (11)	0.998502	0.001497	0.13725	0.63	0.998626	0.225806
	Prophet (49)	0.998188	0.001811	0.32	0.16	0.999468	0.216216
	PELT (49)	0.99834	0.001654	0.448	0.45	0.999155	0.453608

6.4.2 Twitter's Anomaly detection

In this table, outliers dataset has been included as it was meant or an anomaly detection algorithm. Performance results are as follows.

Twitter's Anomaly Detection							
Dataset	Threshold (%)	Accuracy	Error rate	Precision	Recall	Specificity	F-measure
Outliers	1,25,49	0.994312	0.005687	1	0.09	1	0.165137
Sine Wave	1,25,49	0.985714	0.014285	N/A	0	1	N/A
Stationary	1,25,49	0.993752	0.006252	N/A	0	1	N/A
Synthetic 1	1,25,49	0.997619	0.002380	1	0.555555	1	0.714285
Synthetic 2	1,25,49	0.998212	0.001785	1	0.666666	1	0.8
Real Yahoo Data	1	0.997946	0.002053	1	0.8125	1	0.896551
	25, 49	0.988497	0.011502	0.4848	1	0.988	0.653061

Figure 37 shows the average precision, recall and f-measure of Pelt and Prophet on every dataset except non-stationary data.

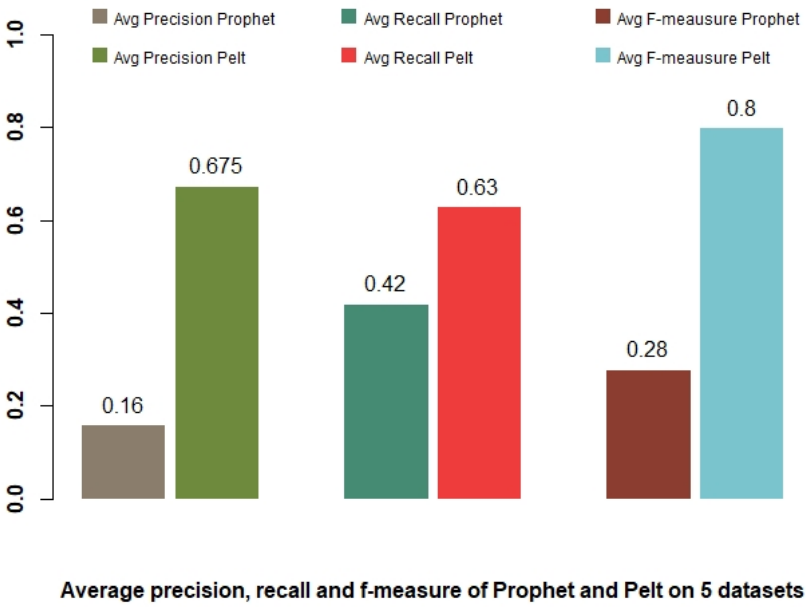

Figure 37: Average precision, recall and f-measure of Prophet and PELT on 5 datasets, except non-stationary data

From the plot, it can be seen that PELT is the clear winner when it comes to all the metrics. It means that it detects true changepoints far better than Prophet, and detects less false negatives than prophet.

Figure 38 shows the average precision, recall and f-measure of Pelt and Prophet on non-stationary data.

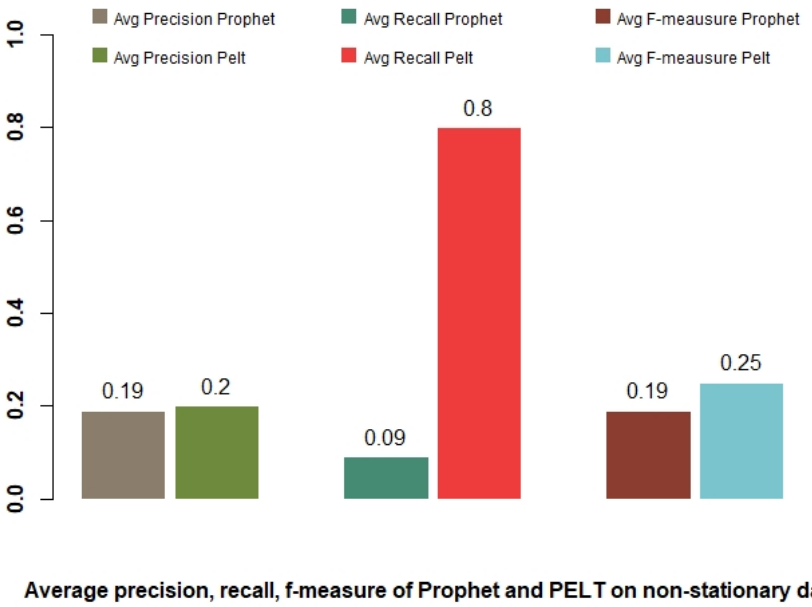

Figure 38: Average precision, recall and f-measure of Prophet and PELT on non-stationary data

From the plot it can be seen that PELT is better than Prophet on non-stationary data. The precision is almost the same in both cases, but recall is where PELT performs better. This shows that PELT detects far less false negatives than Prophet.

Figure 39 is the plot of precision of Twitter's anomaly detection over all the datasets. Precision for sine wave and stationary data were not available, so they have not been included in the plot.

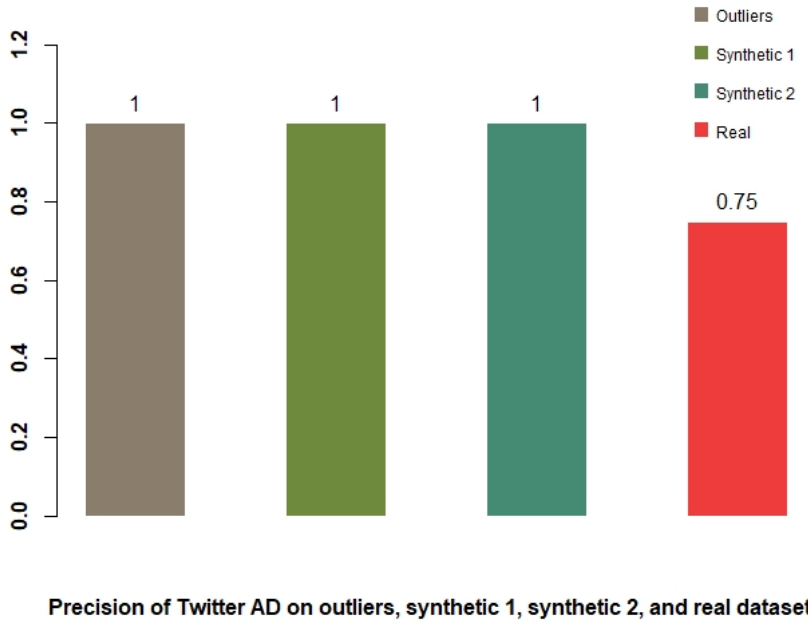

Figure 39: Precision of Twitter's AD on outlier, synthetic 1, synthetic 2 and real datasets.

From the plot it can be seen that the precision is a 100% in the case of outliers data, and synthetic data 1 and 2. No false positives were detected, and points which in ground truth were anomalous have been detected as anomalies. Twitter does good on real data as well, with a precision of 75%.

Figure 40 is a plot of recall of Twitter's anomaly detection for all datasets except sine curve and stationary data, because this metric for those 2 datasets are not available.

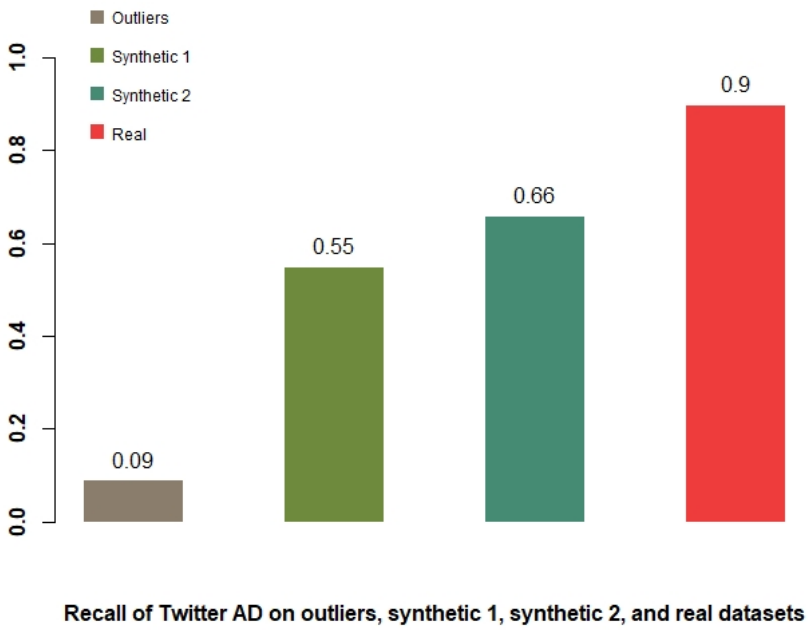

Figure 40: Recall of Twitter's anomaly detection for outliers, synthetic data 1, synthetic data 2 and real world data

From the graph it can be seen that recall is highest in the case of real world data, which means no false negatives were present. Recall is lowest in the case of outliers data, which means lot of false negatives were present. Synthetic datasets both have a recall of 0.5 and 0.6, which means the performance of the algorithm was mediocre, as many false negatives were present as true positives.

Figure 41 is an image of F-measure of Twitter's anomaly detection, the data for sine wave and stationary signal was not available, so they have not been included in the plot.

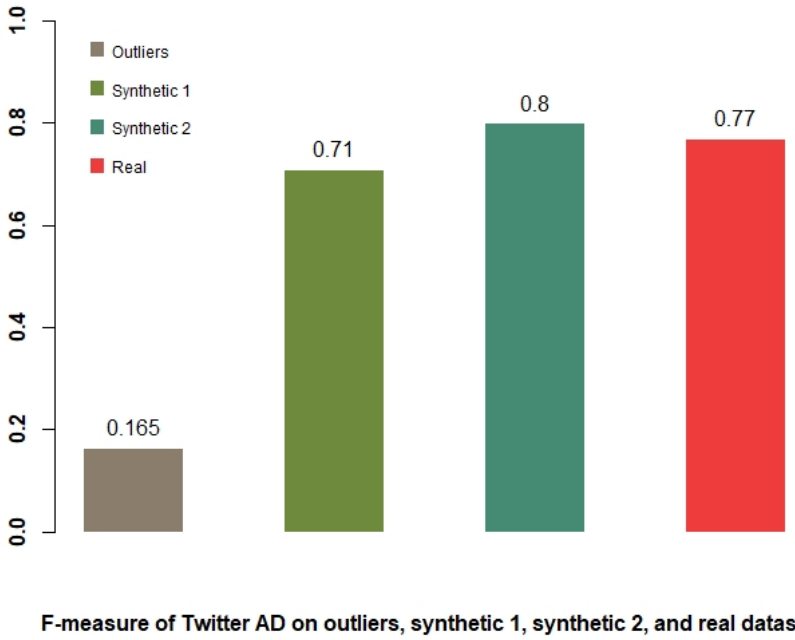

Figure 41: F-meausre of Twitter's AD on outlier, synthetic 1, synthetic 2 and real datasets

From the plot it can be seen that the F-measure is almost the same for synthetic and real datasets, and the value is around the same, also, pretty high. Given that precision and recall for these datasets was high, it is natural F-measure is high as well, since F-measure is dependent on precision and recall. This metric is low for outliers dataset, given the recall for outliers dataset was really low, it dragged the F-measure to a low value.

Run time of algorithms was also taken into account. The algorithms were run on a local machine with an Intel i5 CPU with 4 cores clocked at 2.4 GHz, 8 gigabytes of RAM, Windows 10 enterprise operating system system, and Intel HD graphics with 128 MB of memory.

Figure 42 is a plot of runtime of Prophet on all datasets with increasing number of points. Time is in seconds.

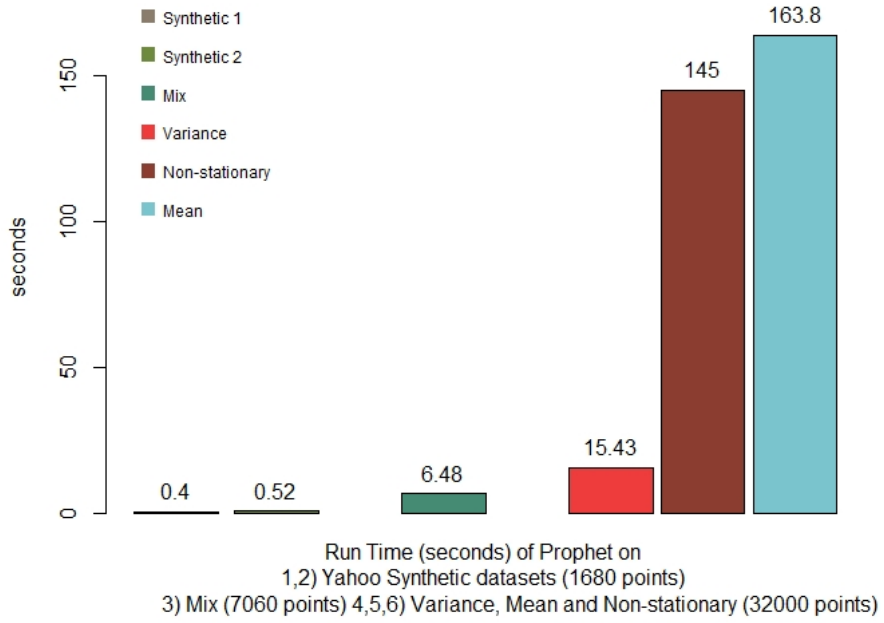

Figure 42: Run time of Prophet on datasets with increasing number of points

Figure 43 is a plot of runtime of PELT on all datasets with increasing number of points. Time in seconds.

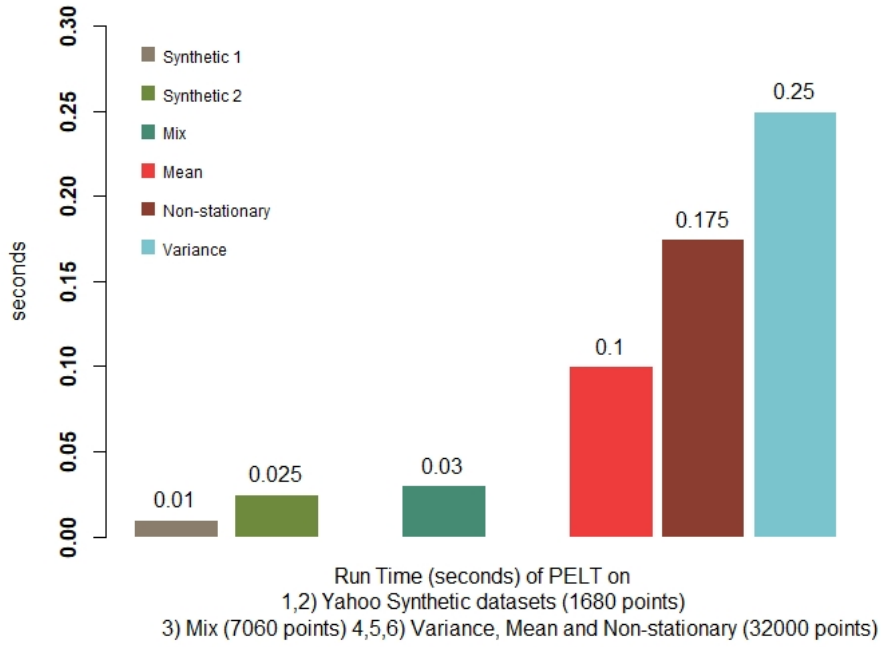

Figure 43: Runtime of PELT on datasets with increasing number of points

Figure 44 is a plot of runtime of Twitter's Anomaly detection on all datasets with increasing number of points.

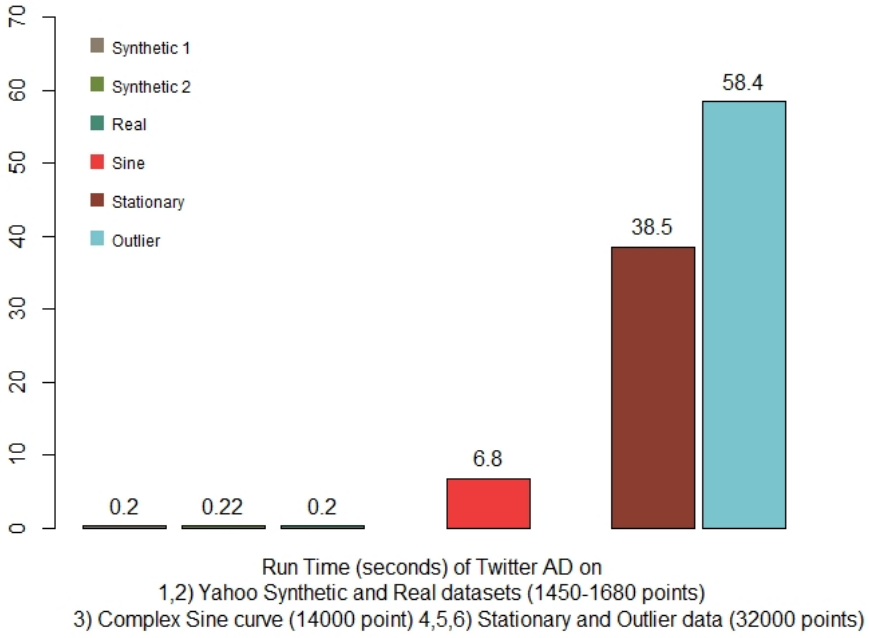

Figure 44: Runtime of Twitter's anomaly detection on all datasets with increasing number of points

From the plots it can be seen that the runtime of algorithms increases with increase in data points. Run time of Twitter's AD is 10 times more than run time of PELT in case of Yahoo synthetic datasets, and run time of Prophet is two times the run times the run time of Twitter's AD for the same datasets. When datasets with more than 30,000 points are considered (change in mean, non-stationary data in case of changepoint data, stationary and outlier data in case of anomaly detection data), Prophet is the slowest, taking more than 100 seconds, Twitter's AD is second slowest, taking an average of 45 seconds, and PELT is the fastest, taking just 0.2 seconds. Overall, PELT is the fastest algorithm among the three.

6.5 Discussion

The results are evaluated for every algorithm, and for every dataset. The metrics were chosen as discussed in section 5.2, Precision, Recall, Error rate, Accuracy, Specificity and F-measure. Most important metrics are Precision, Recall and F-measure. Least important metrics are Accuracy, Error rate, and Specificity. This is because of true negatives. In this collection of datasets, number of true

negatives are really high, but we are not interested in those as there can be a large number of points which can not be either anomalies or changepoints. We are interested in true positives, the points detected as anomalies/changepoints which in ground truth are changepoints/anomalies. Accuracy, Error rate and Specificity makes use of true negatives, which is why they do not provide a lot of useful information. Precision, Recall and F-measure do not make use of those values, and that is why they provide useful information. We are interested in true positives more than true negatives because we want to see how good an algorithm is at picking up anomalies/changepoints, not how good is the algorithm at detecting non-anomalous/non-changepoints.

The tables show the results of every framework upon every dataset. Overall, it can be seen that Accuracy is very high, Error rate is really low, and Specificity is high, which is because of the true negatives.

In case of Facebook's Prophet, the values of accuracy and specificity are really high, and error rate is really low, and these metrics for every dataset are really close to each other, because Prophet does a really good job at detecting true negatives, and these metrics are dependent on true negatives, and because there are a high number of true negatives present. But we are more interested in true positives, ground truth changepoints that were detected, and the metrics precision, recall, and f-measure are dependent on true positives. Highest precision prophet has is in the case of synthetic data 2 by Yahoo, and non-stationary data when the number of changepoints are set to 49. Highest recall is in the case of dataset with change in mean, and synthetic dataset 1. Highest recall is in the case of synthetic data 1.

In case of PELT, from the table it can be seen that it performs best in case of data with changing mean, and changing variance. It detects all the changepoints perfectly, with no false positives. It performs worst in the case of synthetic data 2, in which it detects half the changepoints correctly, and half of them are not detected. If we take into account accuracy, error rate and specificity into account, those values are almost the same for each dataset, stemming from the fact that there are huge number of true negatives. Overall speaking, it performs better than Prophet, which can also be seen from the plots, the average precision, recall and f-measure is always greater than Prophet's.

In case of Twitter's Anomaly detection, specificity and accuracy are really high, and error rate is really low, which means that it is really good at detecting those points that are not anomalies. In fact specificity is actually 1 in a lot of cases. When it comes to true positives, it is really good at detecting anomalies in datasets with outliers, synthetic datasets 1 and 2, and real dataset when threshold is 25% and 49%. But, it detects false positives as well. It does a really bad job at datasets sine wave and stationary data, where it wasn't able

to detect any anomalies at all. Precision and recall in case of these datasets are not available. Overall, it does best in case of real data.

7 Deployment and Automation

7.1 Options

It was desired that one of the algorithms be deployed to cloud with as much ease as possible, laying the groundwork for future deployments. PELT was deployed on cloud over Kubernetes. There were a lot of options for the deployment of the algorithm. These include, H2O, which "provides an open-source, in-memory, distributed, fast, and scalable machine learning and predictive analytics platform" [35]. Second option was OpenScoring.io [36], which provides a predictive model markup language, which converts a machine learning model in Python or R to XML based PMML format, which is interchangeable to other formats. Third option was Kubeflow, which "is targeted at leveraging the scheduling and management ability of Kubernetes, to support mainstream machine learning frameworks as a platform" [37]. Fourth option was ONNX, which "provides a definition of an extensible computation graph model, as well as definitions of built-in operators and standard data types" [38]. It is to be noted that H2O is n IDE, and ONNX is a serialization format.

Initially, Tensorflow serving was selected to deploy PELT over Tensorflow serving. Tensorflow is described as " "an interface for expressing machine learning algorithms, and an implementation for executing such algorithms" [39]. Tensorflow has stateful dataflow graphs. It is a directed graph where nodes represent Operations and edges represent Tensors.

The reason Tensorflow was chosen is because, it is scalable, it provides a ready to deploy serving base upon which an algorithm can be deployed. After deployment, the models can be accessed through CURL over a server. Apart from that, Tensorflow is flexible and portable, because of the clear separation between interface and implementation. Tensorflow supports GPU's and is scalable, and supports parallelization, by making subgraphs. And, Tensorflow is the most asked about package on Stackoverflow in its category since its release, and it was the most forked repository on Github in 2015 [40]. This clearly shows that it is a popular choice among many, it has been tested, and comes with a guarantee from Google.

While deploying PELT onto Tensorflow serving, we encountered an issue, which was a limitation from Tensorflow's side. PELT is written in Python an Numpy, and Tensorflow only accepts models that are written with Tensorflow operators, otherwise it would not be able to make a directed graph. There did exist a function, *py_func*, but Tensorflow does not allow programs with *py_func* to be converted into directed graphs. An alternative was to write PELT in Tensorflow, but that wasn't chosen, and there was another alternative, to deploy PELT over Kubernetes as a service.

Finally, PELT was packed in a docker container, which was deployed to Kubernetes, and the algorithm was accessed as a service. The code can be found in Appendix B. Docker and Kubernetes are explained below.

The reason is docker and Kubernetes are chosen is that Kubernetes deploys the application which can be accessed as a service, just like in the case of Tensorflow serving, we can HTTP to it and it will send the response back. We get the exact same functionality without having to write the algorithm in Tensorflow.

7.2 Docker

Containerization or OS level visualization is a technique where the OS kernel supports multiple isolated user space environments, called as *Containers* [41]. Docker uses its own containerization engine, called as *libcontainer*, to create containers [42]. Docker consists of a docker daemon, which sits on top of the OS, which is responsible for creating container environments. It also manages containers, images, networking, volumes and listens for requests through REST API. The daemon creates the containers and fills it with applications and runtimes using a *Dockerfile*. The applications that have to be installed in a container can be specified in *Dockerfile* and docker daemon will fill it in. The dockerfiles are hosted on Docker registry, from where they can be pulled.

7.3 Kubernetes

Kubernetes is a container orchestration for deploying, scaling and managing containers across machines [43]. Kubernetes groups containers into a virtual entity called as pods, which have their own IP address. Pods have their shared storage and networking, and specifications to build containers. Pods are grouped into an entity called as nodes, and pods can switch from node to node due to reasons being scalability, crashing of pods etc. Nodes can be grouped to form a cluster which can be exposed by Kubernetes master. The architecture of Kubernetes is shown in figure 45.

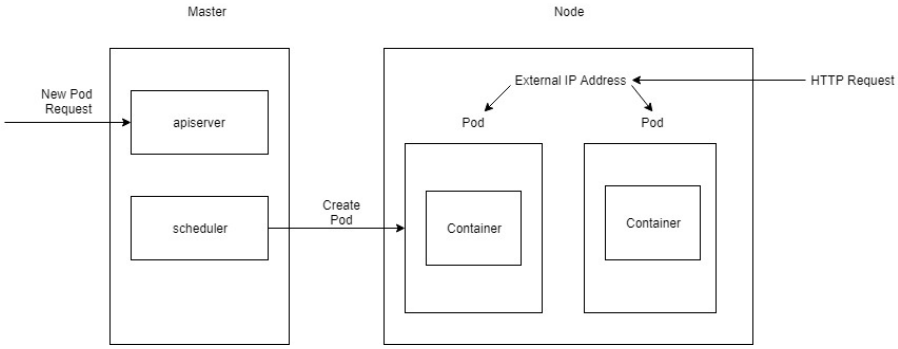

Figure 45: Overview of Kubernetes architecture. Pods have their external IP address, and application on containers can be access by external HTTP requests [44]

The above figure shows the architecture of Kubernetes. In our case, the application, PELT is containerized on docker, and that container is deployed to Kubernetes. That container resides in a pod, and that pod is accessible to outside world through external IP address, which accepts HTTP requests.

8 Limitations and Future work

The thesis tries to include most types of the datasets, which contains situations that can be found in real world datasets. But, it is not all encompassing. That is its one limitation. To expand upon the future work, more types of datasets can be included, both synthetic and real world. Synthetic datasets can include mixing and matching of different datasets suited to changepoint detection with anomaly detection geared datasets, which can serve needs of both. Datasets covered here have both point anomalies and group anomalies. More datasets with group anomalies can be introduced.

In case of changepoint detection frameworks, evaluating the significance of the detected change point is an important issue for unsupervised methods, which includes Prophet and PELT. These two methods compare change scores with a threshold value to determine whether change occurs or not. Selecting the optimal threshold value is difficult. These values may be application dependent and they may change over time. Developing statistical method to find significant change point based on previous values may offer greater autonomy and reliability.

These frameworks are not perfect. There lies scope for their improvement. For example, in case of changepoint detection algorithms, Prophet detects a lot of false positives, which can be seen in case of almost every dataset. PELT detects anomalies as changepoints, though it can be argued whether they are changepoints or not because of the definition of changepoints. Twitter's Anomaly detection detects non-anomalous points as anomalies in certain datasets, which can be seen in case of data with changing mean, and real world dataset. This can be improved as well.

9 Conclusion

This section concludes the report by answering the sub questions and the main research question.

Sub question 1: What kind of datasets cover scenarios of changepoints and anomalies that occur frequently in real world datasets?

Several types of changes can occur in data, most of which are change in mean, change in variance, change in mean and variance, data with naturally occurring changes like non-stationary data. Anomalies that can occur are point anomalies and group anomalies, and both have been covered in datasets like outliers, stationarity, sine wave, two synthetic datasets and one real world dataset by Yahoo. The datasets are separate, different for changepoint detection and different for anomaly detection.

Sub question 2: How does each framework perform against synthetic and real world datasets with changepoints and anomalies?

In case of changepoint detection frameworks, PELT was better overall when compared to Prophet. Prophet performed best with synthetic data by Yahoo, in terms of Precision, Recall and F-measure, and performed worst in case of non-stationary data when frame of reference was 2 changepoints, where it could not detect any changepoint. PELT performed best in case of data with change in mean and change in variance, not just in case of Precision, Recall and F-measure, but Accuracy, Error rate and Specificity. It performs worst in case of non-stationary data with frame of reference as 2 changepoints, where it detects just 2 changepoints, but detects a lot of false positives. But, overall PELT was better than Prophet, detecting more true positives than Prophet in almost every dataset. Anomaly detection did worst in case of sine wave and stationary data, where no anomalies were detected, and does best in case of real data by Yahoo. It gives a mediocre performance in case of synthetic datasets by Yahoo and outliers dataset, where it predicts most of the anomalies correctly, but gives false positives as well.

Sub question 3: What evaluation metrics can be used to compare these frameworks?

The evaluation metrics that have been chosen are Accuracy, Error rate, Precision, Recall, Specificity and F-measure. Out of these, Precision, Recall and F-measure are the most important ones as they are dependent on true positives, because we are interested in those points that are anomalies/changepoints, and not those that are not changepoints/anomalies. Accuracy, Specificity and F-

measure are dependent on true negatives, which, have a high value in almost all datasets, which is why they are not really that important. These metrics have an importance when the number of false positives is really high, as can be seen in case of outliers dataset, when Anomaly detection was applied to it.

Sub question 4: How can a selected framework be deployed to cloud towards automation? The algorithm is containerized, and that container is deployed on Kubernetes, which accepts external HTTP requests, and we can send data to the algorithm and it sends the results back. This functionality is similar to Tensorflow serving, which was initially selected, but we did not go through with it because it required the application to be rewritten in Tensorflow.

Research Question: How do the selected changepoint and anomaly detection frameworks fare against different types of data containing changepoints and anomalies, and how can this process be automated?

The frameworks were applied to a different variety of datasets containing changepoints and anomalies, to cover as many real world scenarios as possible. A real world dataset has also been included. Some framework work with some dataset the best, whereas the performance on other datasets is not that good. To compare their performance several benchmark metrics have been used, namely Accuracy, Error rate, Precision, Recall, Specificity and F-measure. Out of these, Precision, Recall and F-measure are the most important ones as they depend on the number of true positives, detecting which is the main task of these frameworks. The study can be expanded further by including more, different types of datasets with anomalies and changepoints. Finally, one algorithm, PELT is deployed to cloud over Kubernetes, which accepts data over HTTP and sends the response back, which completes the automation part.

10 References

[1] Taylor SJ, Letham B (2017a) Forecasting at scale. Am Stat. https://doi.org/10.1080/00031305.2017.1380080

[2] Killick, R., Fearnhead, P., & Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590-1598.

[3] Hochenbaum, J., Vallis, O. S., & Kejariwal, A. (2017). Automatic anomaly detection in the cloud via statistical learning. arXiv preprint arXiv:1704.07706.

[4] Chandola, V., Banerjee, A., Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.

[5] Aminikhanghahi S, Cook DJ, Knowledge and Information Systems Volume 51 Issue 2, May 2017 Pages 339-367

[6] https://en.wikipedia.org/wiki/Stationary_process

[7] [Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... & Liu, H. H. (1998, March). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences (Vol. 454, No. 1971, pp. 903-995). The Royal Society.]

[8] https://en.wikipedia.org/wiki/Sine_wave

[9] Hodge, V., Austin, J. (2004). A survey of outlier detection methodologies. Artificial intelligence review, 22(2), 85-126.

[10] Quinn, J. A., & Sugiyama, M. (2014). A least-squares approach to anomaly detection in static and sequential data. Pattern Recognition Letters, 40, 36-40.

[11] Upadhyaya, S., & Singh, K. (2012). Nearest neighbour based outlier detection techniques. International Journal of Computer Trends and Technology, 3(2), 299-303.

[12] Yassin, W., Udzir, N. I., Muda, Z., & Sulaiman, M. N. (2013, August). Anomaly-based intrusion detection through k-means clustering and naives bayes classification. In Proc. 4th Int. Conf. Comput. Informatics, ICOCI (No. 49, pp. 298-303).

[13] Ryan, J., Lin, M. J., & Miikkulainen, R. (1998). Intrusion detection with neural networks. In Advances in neural information processing systems (pp. 943-949).

[14] Heard, N. A., Weston, D. J., Platanioti, K., & Hand, D. J. (2010). Bayesian anomaly detection methods for social networks. The Annals of Applied Statistics, 4(2), 645-662.

[15] Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB journal, 16(4), 507-521.

[16] Portnoy, L. (2000). Intrusion detection with unlabeled data using clustering (Doctoral dissertation, Columbia University).

[17] Dette, H., & Gösmann, J. (2018). A likelihood ratio approach to sequential change point detection. arXiv preprint arXiv:1802.07696.

[18] Allahyari, S., & Amiri, A. (2011). A clustering approach for change point estimation in multivariate normal processes. In Proceedings of the 41st International Conference on Computers & Industrial Engineering (Vol. 3843).

[19] Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. The Annals of Statistics, 42(6), 2243-2281.

[20] Jackson Brad, Sargle Jeffrey D, Barnes David, Arabhi Sundararajan, Alt Alina, Gioumousis Peter, Gwin Elyus, Sangtrakulcharoen Paungkaew, Tan Linda, Tsai Tun Tao (2005). An algorithm for optimal partitioning of data on an interval. IEEE, Signal Processing Letters, 12(2), 105-108.

[21] Zhou, Y., Fu, L., & Zhang, B. (2017). Two non parametric methods for change-point detection in distribution. Communications in Statistics-Theory and Methods, 46(6), 2801-2815.

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

[23] Bernard Rosner. On the detection of many outliers. Technometrics, 17(2):221-227, 1975.

[24] Peter J Huber and Elvezio Ronchetti. Robust statistics. Wiley, Hoboken, N.J., 1981.

[25] James M. Lucas and Michael S. Saccucci. Exponentially weighted moving

average control schemes: properties and enhancements. Technometrics, 32(1):1-12, 1990.

[26] Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition. Journal of Official Statistics, 6(1), 3-73.

[27] https://en.wikipedia.org/wiki/Sigmoid_function

[28] https://en.wikipedia.org/wiki/Logistic_function

[29] https://en.wikipedia.org/wiki/Piecewise_linear_function

[30] https://en.wikipedia.org/wiki/Likelihood_function

[31] https://en.wikipedia.org/wiki/Loss_function#Quadratic_loss_function

[32] Sakamoto, Y., Ishiguro, M., & Kitagawa, G. (1986). Akaike information criterion statistics. Dordrecht, The Netherlands: D. Reidel, 81.

[33] Cavanaugh, J. E., & Neath, A. A. (1999). Generalizing the derivation of the Schwarz information criterion. Communications in Statistics-Theory and Methods, 28(1), 49-66.

[34] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999). The CRISP-DM user guide. In 4th CRISP-DM SIG Workshop in Brussels in March (Vol. 1999).

[35] http://docs.h2o.ai/h2o/latest-stable/h2o-docs/architecture.html

[36] https://openscoring.io/

[37] https://github.com/kubeflow

[38] https://github.com/onnx/onnx

[39] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). Tensorflow: A system for large-scale machine learning. In Osdi (Vol. 16, pp. 265-283).

[40] Rao, D. (2016). The unreasonable popularity of tensor ow. http://deliprao.com/archives/168

[41] David Bernstein. "Containers and cloud: From lxc to docker to kubernetes".In: IEEE Cloud Computing 1.3 (2014), pp. 81-84.

[42] libcontainers: https://github.com/opencontainers/runc

[43] Cloud Native Computing Foundation. Kubernetes url: https://kubernetes.io/

[44] https://en.wikipedia.org/wiki/Kubernetes

[45] Hekimoglu, S., and Koch, K. R. (2000). How can reliability of the test for outliers be measured. Allgemeine Vermessungs-Nachrichten, 107(7), 247-53.

11 Appendix A

11.1 Confusion Matrix

Following is a table of true positives, true negatives, false positives and false negatives of both changepoint detection and anomaly detection frameworks.

Confusion Matrix					
Anomaly Detection					
Dataset	Threshold	TP	TN	FP	FN
Outliers	1,25,49	18	31800	0	182
Sine wave	1,25,49	0	13800	0	200
Stationary	1,25,49	0	31800	0	200
Synthetic 1	1,25,49	5	1671	0	4
Synthetic 2	1,25,49	6	1671	0	3
Real	1 25,49	13 16	1445 1445	0 17	3 0
Facebook's Prophet					
Dataset	Threshold	TP	TN	FP	FN
Change in Mean		2	35900	13	2
Change in Variance		0	31997	3	3
Mix		1	7050	8	9
Synthetic 1		2	1676	7	2
Synthetic 2		4	1676	8	0
Non-stationary	2 11 49	0 3 8	31998 31989 31951	2 9 17	2 22 41
Changepoint's PELT					
Dataset	Threshold	TP	TN	FP	FN
Change in Mean		4	35900	0	0
Change in Variance		3	31997	0	0
Mix		7	7050	1	4
Synthetic 1		0	1676	4	4
Synthetic 2		2	1676	2	2
Non-stationary	2 11 49	2 7 22	31998 31989 31951	53 44 27	0 4 26

11.2 Appendix B

The Dockerfile used to create the docker image is shown below

```
FROM ubuntu:latest

RUN apt-get update
RUN apt-get install -y software-properties-common
RUN add-apt-repository ppa:deadsnakes/ppa
RUN apt-get update
RUN apt-get install -y python3.5
#RUN python3.5 -m pip --version

RUN apt-get install -y wget
RUN wget https://bootstrap.pypa.io/get-pip.py
RUN python3.5 get-pip.py

RUN python3.5 -m pip install -U changepy
RUN python3.5 -m pip install -U numpy
RUN apt-get install vim -y
# Create a new system user
RUN useradd -ms /bin/bash changepoint

# Change to this new user
USER changepoint

WORKDIR /home/changepoint/

COPY CPDetection.py /home/changepoint/
COPY data.csv /home/changepoint

ENTRYPOINT ["python3.5","CPDetection.py","data.csv"]
```

In order to deploy it to Kubernetes, following commands were used.

To build the docker image,

docker build -t gcr.io/$PROJECT_ID/changepoint:v1 .

To push the image to google registry,

docker push gcr.io/$PROJECT_ID/changepoint:v1

To deploy application on Kubernetes,

kubectl run changepoint-deployment –image=gcr.io/$PROJECT_ID/changepoint:v1 –port 8080

To run the container

kubectl exec pod_name – bash -c "python3.5 CPDetection.py data1.csv"