

Sameh Faidi

# Finding Anomalous eNodeBs

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

15 August 2018

## PREFACE

I will start by thanking everyone who encouraged me to start this master's degree after a long break from organized education, about 20 years. I have enjoyed going to the Metropolia school classes and doing the assignments and home works. I was very excited and motivated in the first semester of the program and enjoyed, particularly, the big data and cloud computing classes. I started the thesis work in the middle of the second semester and I gained great momentum to finalize 60-70 % of the work. Then I had a period of less motivation during the spring and summer of 2018. It was very challenging to resume the thesis work after the holiday season, but I knew that the only way forward is to finish what I started and finally I resumed the work by the start of the autumn and could finalize everything on time.

My main challenges were related to keeping up the motivation level specially after taking a break from study. I have learned a lot through the thesis work and I have also refreshed some of my previous knowledge that I have not used for long period.

Helsinki, 15.08.2018

Sameh Faidi

Author(s) Title	Sameh Faidi Finding anomalous eNodeBs
Number of Pages Date	52 pages + 3 appendices 18 August 2018
Degree	Master of Engineering
Degree Program	Information Technology
Specialization option	Networking and Services
Instructor(s)	Antti Koivumäki, Principal Lecturer
<p>A typical telecommunication operator could easily have over 10,000 eNodeBs. Huge amount of data and logs are collected from these network elements in daily basis. Existing tools are used to analyze the collected data and generate reports about the status and health of the network. The amount of reports and the information in each is overwhelming which make it virtually impossible for the maintenance team to find problems in their radio network. This lead to many cases where problems go undetected which might result in service degradation and eventually revenue loss.</p> <p>The objective of this thesis was to use machine leaning, particularly anomaly detection, to rank the eNodeBs in the order of their probability of being anomalous using KPIs data. Maintenance teams can save time by focusing on the short list of top anomalous eNodeBs. By performing further investigation and analysis on the anomalous network elements, maintenance teams will be able to apply any required changes and fixes before problems escalate and cause service degradation and eventually loss of revenue.</p> <p>Three different anomaly detection methods were applied to a selected sub set of the KPIs time series, HW related KPIs. The methods and their results were compared and evaluated based on their advantages and disadvantages. The result of this thesis shows that the distance based with custom references method is the most suitable for detecting anomalous eNodeBs as it requires the least number of hyper parameters and it does not seem to be sensitive to the choice of selected threshold.</p>	

## Table of Contents

Preface

Abstract

List of Figures

List of Tables

List of Abbreviations

1	Introduction	1
1.1	Mobile Radio Network	1
1.2	Preventive services business	3
1.3	Research Problem	5
1.4	Objective and Outcomes	6
1.5	Result Verification	7
1.6	Study Outline	7
2	Application Field	8
2.1	eNodeB Overview	8
2.2	Current Analysis of Logs	9
2.2.1	Stability Faults Analysis	9
2.2.2	Active Faults Analysis	9
2.3	Monitoring and Measurement System in LTE	10
2.4	KPI Based Analysis	12
2.4.1	Throughput Degradation Detection	12
2.4.2	Accessibility and Retainability Checks	12
2.4.3	Sleeping Cells Detection	12
2.4.4	Live Use Cases	13
2.5	KPI Data	15
2.5.1	Obtaining the Data	15
2.5.2	Data Pre-processing	15
2.5.3	Visualization and Exploratory Data Analysis (EDA)	18
2.5.4	Clustering of KPIs	20
2.5.5	Selection of KPIs	21
2.5.6	Seasonality in KPIs	22
3	Machine Learning (ML)	23
3.1	Cross Validation	24
3.2	Feature Engineering	25

3.3	Feature Scaling	25
3.4	Clustering Methods	26
3.4.1	Hierarchical Clustering	26
3.4.2	Partitional Clustering	27
4	Anomaly Detection	28
4.1	Supervision Level	29
4.1.1	Semi-supervised Learning	29
4.1.2	Unsupervised Learning	29
4.2	Type of Anomaly	30
4.3	Anomaly Detection Methods	30
4.3.1	Nearest Neighbours Methods	31
4.3.2	Statistical Methods	31
4.3.3	Clustering Methods	31
4.3.4	Density Based	31
4.3.5	Distance Based Methods	33
4.4	Anomaly Detection in Time Series	34
4.4.1	Stationary Time Series	34
4.4.2	Time series decomposition	36
4.4.3	Anomalous Time Series	37
4.5	Azure Time Series Anomaly Detection	38
5	eNodeB Anomaly Detection	40
5.1	Probability Density Function (PDF)	40
5.1.1	Results	41
5.2	Clustering Based Anomaly Detection	43
5.2.1	Results	44
5.3	Distance Based with Custom References	46
5.3.1	Discovery of KPI Optimal Values	46
5.3.2	Discovering Anomalies	47
5.3.3	Results	48
6	Comparison and Conclusion	50
6.1	Comparison of Used Methods	50
6.1.1	Difference from Kumpulainen Approach	51
6.2	Comparison with Threshold Based Methods	52
6.3	Verification of Results	52
6.4	Follow up and Next Steps	52

## References

### Appendices

Appendix 1. Azure Anomaly Detection

Appendix 2. PDF Anomaly Detection

Appendix 3. Clustering Anomaly Detection

## List of Figures

Figure 1.1 3G network layout	10
Figure 1.2 4G network layout	10
Figure 1.3 Mobile data traffic forecast	11
Figure 1.4 Nokia preventive care infrastructure	12
Figure 2.1 Three sectors eNodeB modules	16
Figure 2.2 Measurement community	19
Figure 2.3 CPU usage graphs before and after the issue was resolved	21
Figure 2.4 Success ratio for location update KPI before and after the issue is resolved	22
Figure 2.5 Clustering of eNodeB KPIs using K-Means method	28
Figure 2.6. Ratio based KPIs distribution	30
Figure 2.7 Real values KPIs distribution	30
Figure 3.1 Standard machine learning pipeline	32
Figure 3.2 Hierarchical clustering	35
Figure 4.1 Stationary and non-stationary time series	42
Figure 4.2 Time series with 99% confidence of being stationary	43
Figure 4.3 Time series with 99% confidence of being stationary	44
Figure 4.4 Time series with 95% confidence of being stationary	44
Figure 4.5 Decomposition of a ratio based KPI	44
Figure 4.6 Decomposition of a real value based KPI	45
Figure 4.7 Time series with many peaks (black dots), a change point (red dot) and a trend point (yellow dots)	46
Figure 4.8 Time series with many dips (black dots)	47

Figure 5.1 Two dimensions anomalous data points with 0.005 threshold	49
Figure 5.2 Two dimensions anomalous data points using 3.0 threshold	49
Figure 5.3 kpi_2 graph for enb_9 using PDF method	50
Figure 5.4 kpi_2 graph for enb_26 using clustering method and parameters (3, 0.8, 20)	53
Figure 5.5 Combined KPIs graph for enb_26 using clustering method and parameters (3, 0.9, 20)	53
Figure 5.6 Combined KPIs graph for enb_9 using 1.15 threshold	57
Figure 6.1 Ratio based KPI time series for enb_9	62



## List of Tables

Table 2.1 Active Faults Analysis Severity	17
Table 2.1 Active Faults Analysis Severity	17
Table 2.2 Performance Monitoring Functions	19
Table 2.3 KPI data before pre-processing	24
Table 2.4 Time Series for enb1, cell1 and kpiA	24
Table 2.5 Time Series for enb1, cell1 and kpiB	24
Table 2.6 Time Series for enb1, cell2 and kpiC	24
Table 2.7 Time series before aggregation	25
Table 2.8 Time series after aggregation	25
Table 2.9 EDA findings	27
Table 2.10 Missing values analysis	27
Table 2.11 eNodeB HW related KPIs	29
Table 5.1 Five top anomalous eNodeBs using PDF	50
Table 5.2 Five least anomalous eNodeBs using PDF	50
Table 5.3 input matrix for the clustering method	51
Table 5.4 Five top anomalous eNodeBs using clustering with parameters (3, 0.8, 20)	52
Table 5.5 Five least anomalous eNodeBs using clustering with parameters (3, 0.8, 20)	52
Table 5.6 Five top anomalous eNodeBs using clustering with parameters (3, 0.9, 20)	53
Table 5.7 Data for one cell and using 2 KPIs	55
Table 5.8 Euclidean distance for each time point for one cell	55
Table 5.9 Five top anomalous eNodeBs using distance-based method with 1.15 threshold	56

Table 5.10 Five least anomalous eNodeBs using distance-based method with 1.15 threshold	56
Table 5.11 Five top anomalous eNodeBs using distance-based method with 1.1 threshold	57
Table 6.1 Comparison of the three used anomaly detection methods	59

## List of Abbreviations

3G	Third Generation (mobile network)
4G	Fourth Generation (mobile network)
WCDMA	Wideband Code Division Multiple Access
WBTS	WCDMA Base Transceiver Station
eNodeB	Evolved Node B (abbreviated as eNodeB or eNB)
KPI	Key Performance Indicator
LTE	Long Term Evolution
PM	Performance Monitoring
QoS	Quality of Service
EPC	Evolved Packet Core
RF	Radio Frequency
std	Standard Deviation
SVM	Support Vector Machines
ML	Machine Learning
FE	Feature Engineering

# 1 Introduction

## 1.1 Mobile Radio Network

Mobile telecommunication networks have evolved a lot through the years in terms of capacity, performance and general end user experience. This in turn enabled service providers to take advantage of the improved network and offer end users a wider range of applications and services which feed into the motivation of improving the performance of the network even more.

A typical telecommunication network consists of the following three main parts (1):

1. User Equipment (UE): This can refer to any mobile phones or tables or any other mobile enabled device.
2. Radio Network Subsystem (RNS): This is the part that provides and manages the air interface between UE and the Core Network.
3. Core Network: The core network provides all the central processing and management for the system. It is equivalent to the Network Switching Subsystem in earlier GSM networks and it consists of circuit switched and packet switched elements.

The Radio part of the network has evolved and is different in 3G network compared to 4G (and later LTE) networks. In 3G network we have the following elements (1):

1. WBTS (sometimes referred to as Node B): This is the hardware that is connected to the mobile phone network and communicates directly with mobile handsets. In contrast with GSM base stations, WBTS uses WCDMA as the air interface technology. As in all cellular systems, the WBTS contains radio frequency transmitter(s) and receiver(s) used to communicate directly with mobile devices, which move freely around it. In 3G networks, the WBTSs have minimum functionality, and are controlled by an RNC (Radio Network Controller).
2. Radio Network Controller (RNC): It is a governing element and is responsible for controlling the WBTSs that are connected to it. The RNC carries out radio resource management, some of the mobility management functions and is the point where encryption is done before user data is sent to and from the mobile. The RNC connects to the circuit and packet switched network (Core Network).

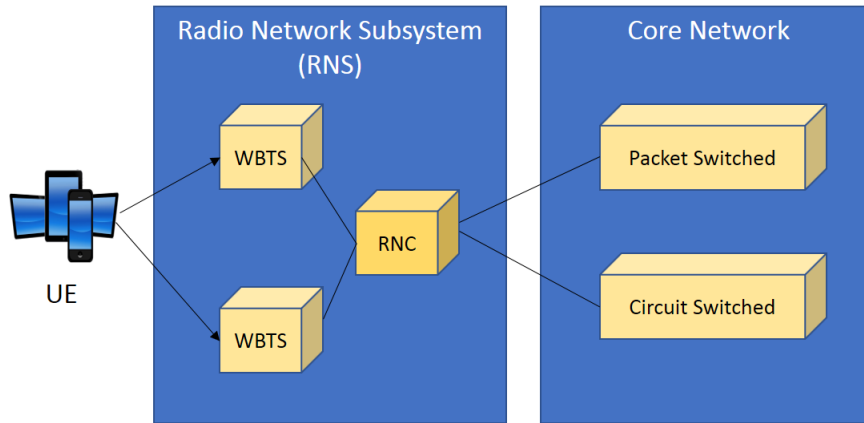


Figure 1.1 3G network layout

In 4G and LTE networks the architecture was simplified and flattened and the radio part of the network only consist of the Evolved Node B (abbreviated as eNodeB or eNB). eNodeB is the hardware that is connected to the mobile phone network that communicates directly and wirelessly with mobile handsets (UEs). eNodeB embeds its own control functionality, rather than using an RNC (Radio Network Controller) as does the WBTS in 3G network (2).

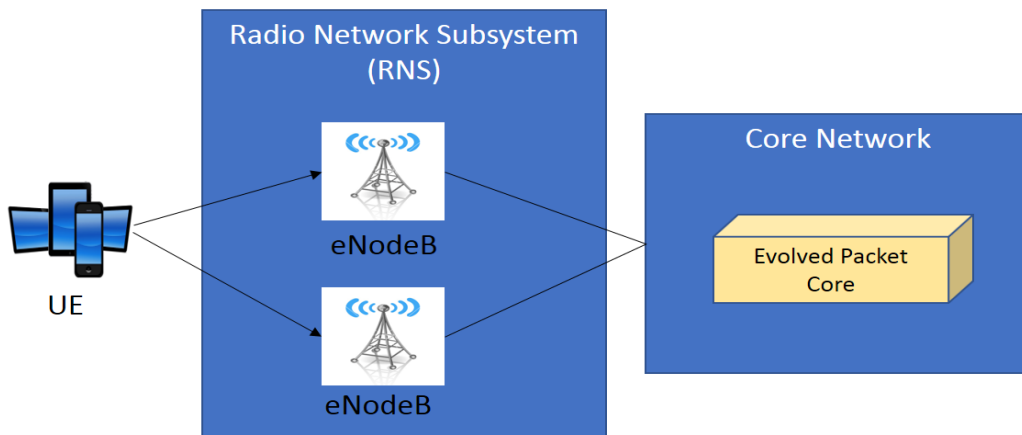


Figure 1.2 4G network layout

The focus of this thesis is the eNodeB network elements because typically telecommunication networks contain huge amount of them and they could be source of problems for the network operations.

## 1.2 Preventive services business

The usage of mobile networks is still increasing at a very high rate. According to estimates, the amount of data traffic in mobile networks will increase from 7 exabytes per month in 2016 to 49 exabytes per month in 2021. At the same time, end users are becoming more and more attached to their various mobile gadgets engaging in different daily mobile activities such as social media, e-commerce and video streaming. This put more pressure on mobile network providers for maintaining their network to the highest possible quality and ensuring the end user experience is seamless. This means that operators must spend bigger portion of their budget on services for continuous network monitoring that help to increase their network stability and reliability (3).

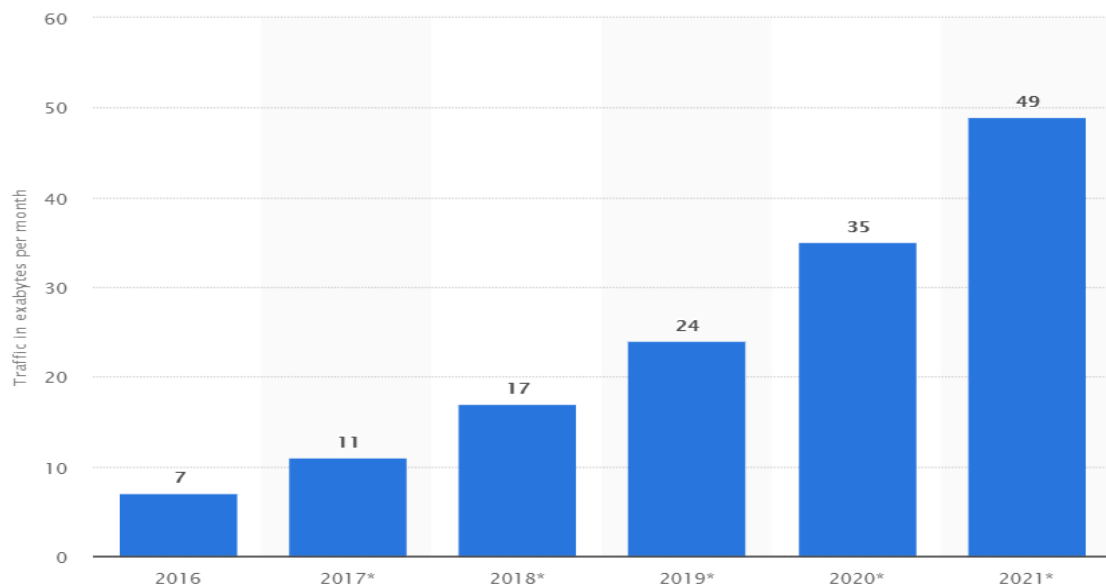


Figure 1.3 Mobile data traffic forecast (3)

Nokia provides preventive services for the telecommunication operators who purchase its mobile network equipment. The main workflow of preventive services consists of data collection, data transfer and data analysis. The target of the services is to detect errors and potential problems in the network early on and to enable the operators own maintenance teams to react to specific events and apply corrections if needed to avoid any loss of connectivity or service degradation that could possibly lead to end user dissatisfaction and eventually losing revenue (4).

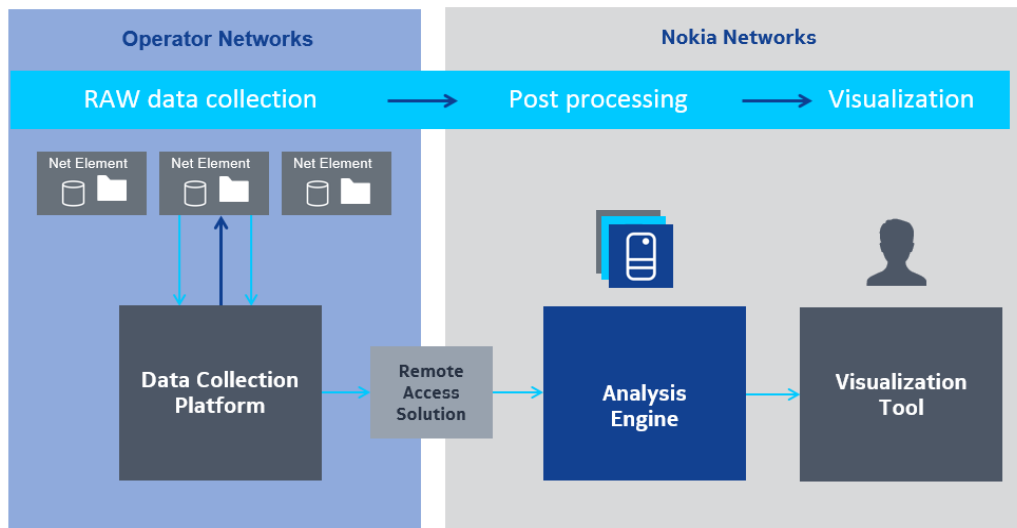


Figure 1.4 Nokia preventive care infrastructure (4)

### 1.3 Research Problem

A typical telecommunication operator could easily have over 10,000 radio base stations network elements (WBTSs and eNodeBs). Huge amount of data is collected from these network elements in daily basis. The collected data ranges from alarms logs, event logs, configuration files and various performance monitoring counter files. Existing tools are used to analyze the raw files and the result of the analysis is presented to the operator in a form of a web-based reports. The huge amount of reports and huge amount of information in each one of them, make it virtually impossible for the operators to find problems in their radio network. This lead to many cases where problems go undetected which might result in service degradation and eventually revenue loss.

The radio network elements produce a wide range of counters data that could be used to monitor its performance and quality of service. The counters are grouped into administrative entities called measurements. The counters are the building blocks for key performance indicators (KPIs). A KPI is basically a formula that consists of one or several counters. The underlying formulas for calculating KPIs are company confidential and not publicly available. The KPIs are used to create top-level reports, which indicate the network performance and functionality.

Counters are usually collected in predefined intervals of 15, 30 or 60 minutes and over a long period of time (Time Series). Experts usually defines thresholds for counters and KPIs that could be used to trigger alarms. However, this method is not optimal as it relies on hard and fixed values that could become irrelevant with new trend in the traffic or a change in the network configuration.

An alternative to threshold-based analysis is to use machine learning and specifically anomaly detection methods to find the points in time when a counter or a KPI value deviate from its normal range. Anomaly detection could be applied to one counter at a time (univariate analysis), or to many counters at once (multivariate analysis). The latter will most likely provide higher prediction capabilities as it looks at combination of many aspects of the network element at the same time.



## 1.4 Objective and Outcomes

The objective of this thesis is to develop an anomaly detection method to find the top anomalous eNodeBs in LTE networks. The result of this method will be a short list of network elements that the operator maintenance teams can focus their analysis and troubleshooting effort on.

The number of counters collected from a network element is enormous, therefore it will be not feasible to use them for anomaly detection. On the other hand, KPIs are aggregated from many counters and they present understandable and easy to interpret aspects of the network elements. In this thesis, various anomaly detection methods will be applied to a selected list of KPIs data collected over longer period of time (Time Series) from eNodeB network elements.

The result will be the number of anomaly detected in each network elements. The network elements with the highest amount of anomaly will be presented to operation and maintenance teams as a potential target for further investigation. By providing a list of top problematic network elements we achieve two folds benefit. First, maintenance team can save time by focusing their effort on the short list of network elements (a list of a top 10 or 100 compared to the complete network of over 10,000 network elements) and secondly, by performing further investigation and analysis on the anomalous network elements, maintenance teams will be able to apply any changes or fixes before issues escalate further and cause service degradation and affect the end user experience and potentially cause loss of revenue for the operator.

## 1.5 Result Verification

The result of this thesis will be a short list of network elements that have the highest amount of anomalous KPIs values. The intuition is that these network elements are more likely to be problematic and that maintenance teams should manually investigate these network elements to make sure they are functioning correctly. Anomaly detection is usually an unsupervised learning method where the ground truth is not known. If the ground truth was known, then the problem would be handled as a binary classification that give the probability of each data point being anomalous. The verification would be straight forward in that case and a metric like Logarithmic Loss or AUROC (Area Under the Receiver Operating Characteristic curve) could be used to measure the performance of the predictions with concrete accuracy. In real life problems, and in this thesis, a practical way to verify the detected anomalies is to use expert and domain knowledge. In addition, logs and related alarms could be investigated manually to find out if any interesting events took place at or around the detected anomaly time.

## 1.6 Study Outline

In **chapter two**, an overview of the non-KPI related analysis currently available for the WBTS and eNodeB network elements is presented. It also presents the KPIs data and its statistical properties through Exploratory Data Analysis. Finally, few use cases of current threshold-based analysis using KPIs is presented. **Chapter three** introduces machine learning and different concepts related to it such as feature engineering, feature scaling and clustering methods. In **chapter four**, anomaly detection methods are outlined and explained in detail. In this chapter also, a ready-made anomaly library from Microsoft Azure is used to discover anomalies the eNodeBs KPIs. **Chapter five** is where the actual thesis solution is presented. In this chapter, three different methods are used to discover anomalies and produce the list of top anomalous eNodeBs that fulfil the objective of this thesis. In the **last chapter** of this thesis, the anomaly detection methods and their results will be compared. The target of this chapter is to draw conclusions regarding the use of anomaly detection for KPIs data in eNodeBs. Finally, the chapter will conclude by presenting some ideas for what can be done as a follow up work in this research area.

## 2 Application Field

### 2.1 eNodeB Overview

The network base station in LTE networks, known as eNodeB, performs the tasks of the NodeB and Radio Access Controller (RNC) combined in previous releases of telecommunication networks. The target behind this simplified architecture is to reduce the latency of all radio interface operations. An eNodeB consists of a tower and several antennas attached to the tower. In many cases, three antennas are used and they are separated by 120 degrees. In more dense sites, more antennas could be used to increase the capacity and to be able to serve more users. The antennas mounted on the tower are continuously scanning for UEs to serve. The eNodeB together with the UE are referred to as E-UTRAN (2).

An eNodeB provides the following two main functions (2):

1. Sends and receives radio transmission to all connected UEs using the analogue and digital signal processing functions of the LTE air interface
2. Controls the low-level operations of all the connected UEs by sending them signalling messages such as handover commands.

A typical deployment of eNodeB consists of baseband system module and one or multiple radio modules that handle the transmit and receive of RF signals. Each radio unit is connected to an antenna that serves a specific direction, and thus forming a sector or a cell (2).

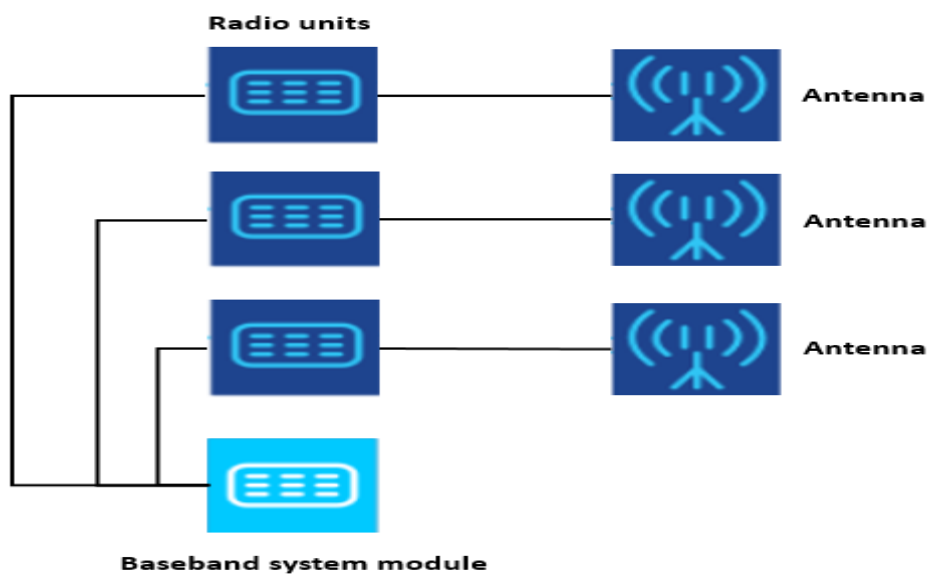


Figure 2.1 Three sectors eNodeB modules (2)

## 2.2 Current Analysis of Logs

In this chapter, the current analysis performed on the eNodeB network elements is explained. The analysis is done using logs collected from the network elements and will be used in the context of this thesis as a background information and as means of verification of the anomaly detection methods. Two types of non-KPI related analysis will be presented. Firstly, the stability faults analysis and secondly the active faults analysis of the eNodeB.

### 2.2.1 Stability Faults Analysis

The purpose of this analysis is to find stability related faults detected during the eNodeB startup or runtime. The finding from this analysis is a list of faults with the timestamp, reporting unit and severity if available. The list of faults to be checked are determined by technology experts. In case, none of the faults is found then no finding is presented. In all cases, the approximate startup time of the eNodeB is shown in the finding. In some cases, a site reset may occur because of a fault and in these cases the log files used to collect the faults will be lost and therefore such an analysis will not be possible (5).

### 2.2.2 Active Faults Analysis

Active faults of eNodeB showing the timestamp, fault id, fault name and fault source. This is done by checking a specific list of faults that technology experts specified to be worthy of checking from the active alarms log file. The end user is presented with a table that contains the list of faults and their meaning and possible suggestion of how to mitigate their negative impact on the operation of the network element. The severity of the finding from this analysis is decided based on the unit status in case such a fault would occur (6). The mapping is done in the following way:

Unit Status	Severity
Out of order	Critical
Degraded	Major
Working	Minor

Table 2.1 Active Faults Analysis Severity (6)

### 2.3 Monitoring and Measurement System in LTE

In this chapter the generic concept of performance monitoring system for eNodeB network elements is explained. This will lead us to the understanding of the KPIs data that will be used in this thesis as an object for applying anomaly detection methods.

The main target of a mobile telecommunication operator is to be profitable. To do that it is required to provide a flawless network services and mobile experience to the end users in low or reasonable costs. Monitoring and performance measurement make the operator job much easier in achieving these goals and will support the operator in many related tasks and processes (7). Performance monitoring enable the collection of information about the following (7):

- Intensity of networks traffic
- Traffic distribution (if it is spread out evenly, or concentrated in certain spots)
- Events happening in certain spots of the network (and how often do they occur)
- Planning efficiency (that is, if the instructions are fulfilled, or when any additional changes are needed)
- Locations where frequent failures are reported
- Subscriber behaviour (if it corresponds with the assumed model)

Using performance monitoring, the operator can have a clear view on the performance, capacity and quality of the network and hopefully improve its users' satisfaction and loyalty.

Key areas where performance monitoring can be used for (7):

- Network planning
- Acceptance and verification
- Traffic model verification
- Benchmarking
- Troubleshooting
- Monitoring of QoS
- Network optimization

Performance monitoring can be divided into functions and sub-functions. Examples are given in the table below:

Function	Sub-function
Measuring	<ul style="list-style-type: none"> <li>• Performance measurements: <ul style="list-style-type: none"> <li>○ Counters</li> <li>○ Counter based KPIs</li> </ul> </li> <li>• Threshold based PM alarms</li> </ul>
Tracing	<ul style="list-style-type: none"> <li>• Subscriber and equipment trace</li> <li>• Cell traffic</li> <li>• External interface trace</li> </ul>

Table 2.2 Performance Monitoring Functions (7)

The Measuring function is about gathering various performance data from the network elements. The lowest level data is called Counters. KPIs are aggregated from counters and technology experts usually are responsible for defining the formula for deriving KPIs from low level counters. The KPIs are usually grouped into entities called Measurement areas that represent a specific aspect of the performance of the network element (7).

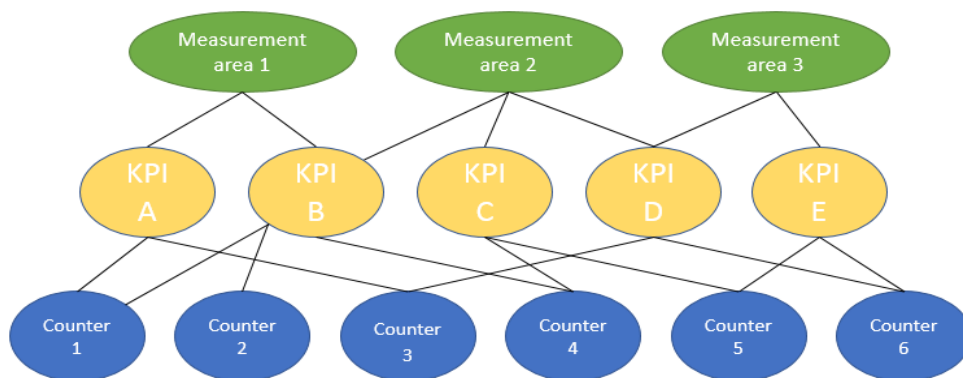


Figure 2.2 Measurement community

The Tracing function of the monitoring is about collecting tracing records for certain events like calls or subscribers related operations and can help the operator to directly detect network and service failures in their network (7).

## 2.4 KPI Based Analysis

In this chapter current analysis performed on eNodeB based on counter and KPIs data is explained.

### 2.4.1 Throughput Degradation Detection

Specific KPIs and counters are checked to detect conditions impacting the throughput performance achieved by the end users in cases when the accessibility is not impacted.

For this analyser, KPIs affecting the call setup successful ratio is checked against a high threshold that indicates a good setup rate. Secondly, a counter that represents the throughput is compared to a low threshold value that indicates low throughput. If the combined condition is found in several consecutive hours then a flag is raised to the operator. The severity of the flag is correlated to how many hours the condition is present in the network element (8).

### 2.4.2 Accessibility and Retainability Checks

In this analyser, four different KPIs are checked against specific thresholds and a flag is raised if any of the KPIs is found to have values higher or lower than the specific threshold for several consecutive hours. The severity of the flag is correlated to how many hours the KPI was found at the too high or too low value (9).

### 2.4.3 Sleeping Cells Detection

Cells in eNodeB could become inactive or otherwise unable to serve user traffic even when they are on-air. In many cases, there are no faults or alarms that indicate such an abnormal state of cells, sleeping state (10).

This analysis focuses on two groups of KPIs to detect sleeping cells. First group is related to access attempts and the second to volume of traffic. In both cases, a cell is considered in sleeping state if the KPI value goes under a defined threshold for longer periods of time (10).

To account for natural variation in traffic during the day and in different days of the week, the analysis is capable of excluding certain hours of the day and days of the week from being checked against the defined threshold. This is necessary as some cells will naturally have very low, or no traffic at all, during night time while other cells might be deployed in sites where no user traffic exists at all during weekend. The used threshold and

the excluded hours and days are configurable and could be customized per telecommunication operator to account for variation in different networks and countries (10).

Applying anomaly detection method to the sleeping cells related KPIs would most likely not yield to finding problems in the eNodeB as a huge spike or dip in the KPI could be completely expected due to natural variation of traffic. However, this set of KPIs could be used in the verification of the anomaly detection findings.

#### 2.4.4 Live Use Cases

KPI based analysis, and specifically threshold based, is used as part of Nokia preventive care services offered to telecommunication operators with Nokia equipment. Two live use cases of using the threshold-based analysis are presented and observations are made regarding the strength and weakness of this method that would be compared with anomaly detection method.

##### CPU Usage KPI

Experts specified three thresholds for the CPU usage KPI that should be monitored. If the CPU usage exceeds any of the thresholds values then an action by the maintenance team should be taken. The top critical threshold was defined at 95% which means CPU usage higher than this would cause unpredictable behaviour and might result in interruption to the end user operations (11).

The maintenance team noticed that the upper threshold was exceeded and they took an action by manually checking if any unnecessary processes are running in the system. The unnecessary processes were stopped and shortly after that the CPU usage returned to normal (11).



Figure 2.3 CPU usage graphs before and after the issue was resolved



## Location update Success Ratio KPI

In this case, a KPI that indicates the success ratio for location updates was monitored against a set of thresholds. As the KPI had a cyclic or daily seasonal pattern, the thresholds had to be adjusted accordingly. The maintenance team noticed that the KPI values were under the threshold and an action was taken to apply a fix by reinitializing a link (11).

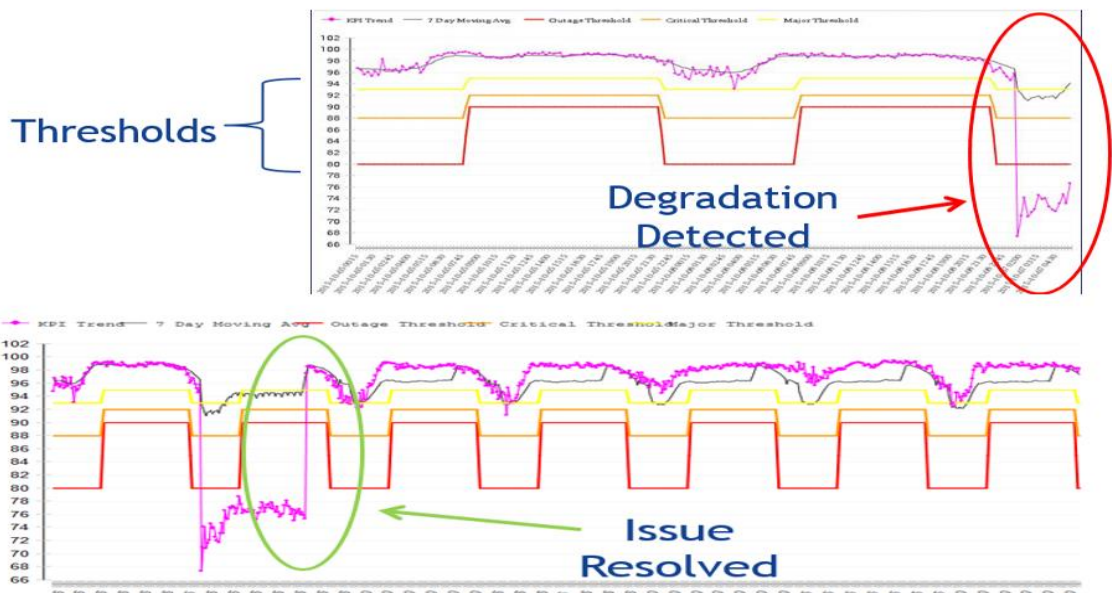


Figure 2.4 Success ratio for location update KPI before and after the issue is resolved

As can be seen from the live use cases, the threshold-based analysis was very successful in catching a real problem in the network. The maintenance team's fast reaction and the deployment of a quick fix could resolve the issues and ensure that the impact is minimized and that normal operations are resumed. However, defining thresholds for KPIs is not usually a trivial work and requires a deep knowledge of the technology in general and the operator-specific configuration. The thresholds are defined at KPI level and problems that could be caused by changes in multiple KPIs are not detected.

## 2.5 KPI Data

### 2.5.1 Obtaining the Data

The data collection is done at the network element level. The counters are saved and aggregated in predefined intervals (15 mins, 60 mins or less often). This is usually a configurable setting that might vary from one operator to another.

In our current setup, the counter data is collected from the operator network in daily basis. Each daily package for one eNodeB contains several files per each aggregation period. For example, there will be 24 files if the aggregation period is 60 mins. The files are processed at Nokia side into tabular format that includes the following information:

- Timestamp
- Counter name and value
- The cell ID in the eNodeB.

The cell ID correspond to a radio module in the eNodeB that is connected to a physical antenna mounted in a tower. After the parsing, the counter data is used to calculate a set of defined KPIs using formulas defined by technology experts to measure certain aspects of the performance and quality of service of the eNodeB.

### 2.5.2 Data Pre-processing

The KPI data is extracted from the database and several pre-processing steps are applied to the data before the actual analysis is done. The target of the pre-processing is to obtain time series data for each combination of a KPI, eNodeB and Cell ID aggregated in the same frequency.

#### 2.5.2.1 Time Series extraction

The KPI data is stored in a tabular format with each row presenting the value of a specific KPI for a specific cell at a specific timestamp. In this step we will extract all the values that belong to the same cell and KPI to a separate table. Example of the data before and after pre-processing is show in the following tables.

Timestamp	eNodeB	KPI ID	Cell ID	Value
T1	enb1	kpiA	cell1	1
T1	enb1	kpiB	cell1	2
T2	enb1	kpiA	cell1	1
T2	enb1	kpiB	cell1	2
T2	enb1	kpiC	cell2	3
T3	enb1	kpiA	cell1	1.1
T3	enb1	kpiB	cell1	2.1
T3	enb1	kpiC	cell2	3.1

Table 2.3 KPI data before pre-processing

Timestamp	Value
T1	1
T2	1
T3	1.1

Table 2.4 Time Series for enb1, cell1 and kpiA

Timestamp	Value
T1	2
T2	2
T3	2.1

Table 2.5 Time Series for enb1, cell1 and kpiB

Timestamp	Value
T2	3
T3	3.1

Table 2.6 Time Series for enb1, cell2 and kpiC

In addition, unnecessary fields are removed from the data and the type of some fields are corrected. For example, the Timestamp is converted to datetime format and the value is converted to float.

### 2.5.2.2 Time Aggregation

As the counter data could be saved and collected at different frequency (15 mins, 60 mins or less often), it will be important to perform aggregation on the data so that all time series have the same frequency. One-hour frequency is selected for the aggregation as most of the KPIs seems to have at least one value in any given hour and with less frequency we might not capture important information. When a KPI has more than one value in an hour, then the mean of the values is used for that hour. Example of time series data before and after time aggregation:

Timestamp	Value
09:00	100
09:15	95
09:30	98
09:45	93
10:00	99
10:15	95
10:30	90
10:45	85

Table 2.7 Time series before aggregation

Timestamp	Value
09:00	Mean (100, 95, 98, 93) = 96.5
10:00	Mean (99, 95, 90, 85) = 92.25

Table 2.8 Time series after aggregation

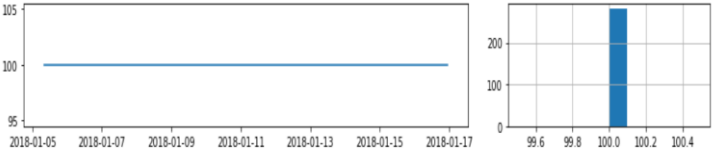
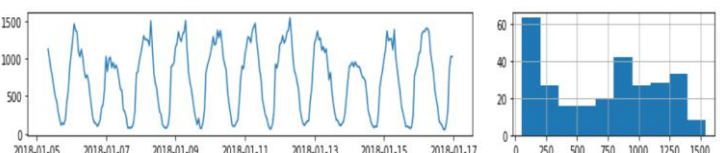
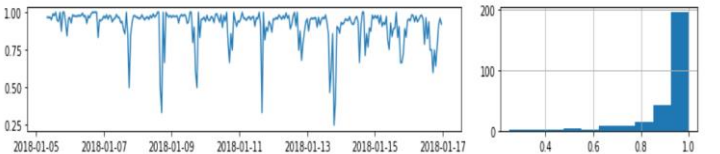
### 2.5.3 Visualization and Exploratory Data Analysis (EDA)

In this chapter, the properties and statistical attributes of the KPIs time series generated by the pre-processing step will be explored. Different aspects of the network element performance are covered by each KPI and therefore the total amount of KPIs could be overwhelming. Understanding the KPIs statistical properties and attributes is essential in developing any meaningful analysis as in most cases specific methods will only be effective if the distribution of the data is of certain type.

The following characteristics of each time series will be examined:

1. Basic statistical metrics like mean, median, min, max, std and skewness
2. Visualization as a time series
3. Distribution through histograms
4. Missing values analysis

A summary of the findings from the Exploratory Data Analysis is shown in the below table.

Few of the KPIs had constant value over time (zero standard deviation)	 <p>The figure shows a time series plot on the left and a histogram on the right. The time series plot has a y-axis from 95 to 105 and an x-axis from 2018-01-05 to 2018-01-17. A single horizontal blue line is plotted at the value 100. The histogram has a y-axis from 0 to 200 and an x-axis from 99.6 to 100.4. It shows a single, very narrow bar at the value 100.0, reaching a height of approximately 200.</p>
Daily and weekly seasonality can be noticed in some KPIs with higher and lower values based on the hour of the day and the day of the week.	 <p>The figure shows a time series plot on the left and a histogram on the right. The time series plot has a y-axis from 0 to 1500 and an x-axis from 2018-01-05 to 2018-01-17. It displays a clear periodic pattern with peaks around 1400 and troughs around 200. The histogram has a y-axis from 0 to 60 and an x-axis from 0 to 1500. It shows a distribution with multiple peaks, with the highest peak around 200 and another significant peak around 800.</p>
Not all KPIs have normal distribution. Many had a long tail to one or both sides.	 <p>The figure shows a time series plot on the left and a histogram on the right. The time series plot has a y-axis from 0.25 to 1.00 and an x-axis from 2018-01-05 to 2018-01-17. It shows a signal fluctuating between 0.25 and 1.00 with several sharp downward spikes. The histogram has a y-axis from 0 to 200 and an x-axis from 0.4 to 1.0. It shows a distribution that is heavily skewed to the right, with a very high frequency of values near 1.0 and a long tail extending towards 0.4.</p>

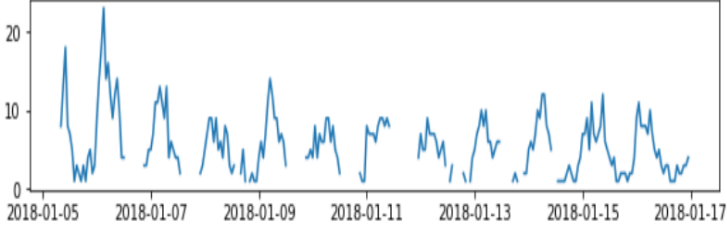
<p>Considerable amount of missing values is noticed in large amount of KPIs.</p>	
<p>Most of the KPIs are stationary which mean in general the mean of the KPI value does not change over the observation time.</p>	

Table 2.9 EDA findings

### 2.5.3.1 Missing values

The missing values are caused by either data collection fault or errors during the registration of counters at the eNodeB level. Understanding the prevalence of missing data in the KPI time series is crucial when selecting the analysis method. Two aspects are examined with regards to missing value in hourly aggregated KPI time series. The number of KPIs time series affected by missing values and secondly the amount of missing values in the affected time series.

Average percentage of KPIs with any missing values	Average percentage of missing values in these KPIs
41% (Calculated from a random 10 eNodeBs)	58% (Calculated from a random 10 eNodeBs)

Table 2.10 Missing values analysis

As can be seen from the figures in the above table, it will not be a good idea to use all the KPIs as about 40% of them have some missing values. In average when a KPI is missing values then it is missing large portion of the total values (close to 60%). This emphasis the need for a KPI selection method that will be investigated in the next chapters.

## 2.5.4 Clustering of KPIs

Thousands of counters are collected from eNodeB network element every day. The number of calculated KPIs, though smaller, is still very high and in most operators, there could be hundreds of KPIs to be examined and analysed. The nature of the data, the KPI time series in this case, will influence the selected analysis method. For example, some methods might be more sensitive to missing values while others might produce better results for normally distributed time series. Instead of manually grouping the KPIs, which would be very time consuming, clustering technique could be applied to separate the KPIs into groups with similar statistical properties.

Several clustering methods are available. K-Means is selected in this case as it is one of the most common in this field and very easy to use. It is usually a good practise to experiment with several number of clusters. In this case, three experiments were performed with three, five and eight clusters. Five clusters seem to give reasonable results.

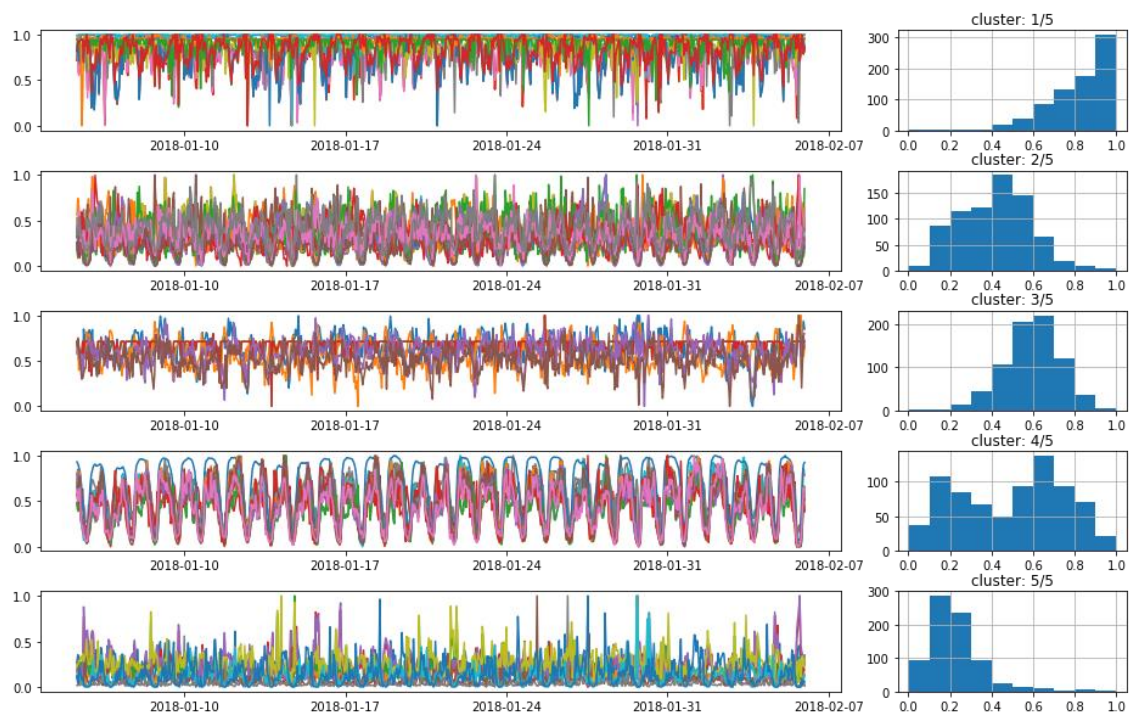


Figure 2.5 Clustering of eNodeB KPIs using K-Means method

The K-Means with five clusters seem to group the KPIs nicely into five groups with a lot of common statistical properties for KPIs in each group. The following could be observed:

- Cluster 1: majority of KPIs are negatively skewed (long left tail)
- Cluster 2 and 3: majority of KPIs have normal distribution which could be slightly skewed to either side.

- Cluster 4: majority of KPIs have a clear cyclical pattern and their histogram are multimodal.
- Cluster 5: majority of KPIs are positively skewed (long right tail)

### 2.5.5 Selection of KPIs

KPIs selection could be done based on their statistical properties as presented in the previous chapter. Another approach for KPI selection could be their performance measurement area. A target group of KPIs could be related to quality of service, capacity or hardware (7).

Hardware related KPIs, in particular, have been analysed using traditional threshold-based methods. Applying anomaly detection method to the same set of KPIs would give a good ground for comparison.

Three groups of hardware related KPIs have been identified and are presented in the following table (12).

Group	KPI	Impact
1	Total E-UTRAN RRC Connection Setup Success Ratio	could indicate and lead to a general hardware failure
	E-UTRAN E-RAB Setup Success Ratio, QC1	
	E-UTRAN E-RAB Drop Ratio, RAN View	
2	E-UTRAN HO Success Ratio, inter eNB X2 based	could indicate and lead to interface (card) failure
	E-UTRAN average PDCP Layer Active Cell Throughput UL	
	E-UTRAN average PDCP Layer Active Cell Throughput DL	
3	E-UTRAN PDCP SDU Volume DL	they have indirect impact that indicate failure in radio modules that could lead to a dip in traffic
	E-UTRAN PDCP SDU Volume UL	

Table 2.11 eNodeB HW related KPIs (12)



### 2.5.6 Seasonality in KPIs

Seasonality in time series data refers to changes in distribution during different period of times. A time series could have hourly or daily seasonality that cause the KPI to have different values during different hours of the day or different days of the week. Seasonality, in many cases, reflects normal changes in traffic in the mobile network. The following graphs shows the daily and hourly distribution of hardware related KPIs. A random 50 eNodeBs were selected and each KPI was aggregated to hourly level using the mean value for the hour. The first five KPIs are ratio based where 100% is usually the optimal value, success ratio for example. For these KPIs, each bar represents the minimum value for the hour or day of week among the 50 network elements. The remaining KPIs have real numbers as values and in this case, each bar represents the mean value of the hour or day of week among the 50 network elements.

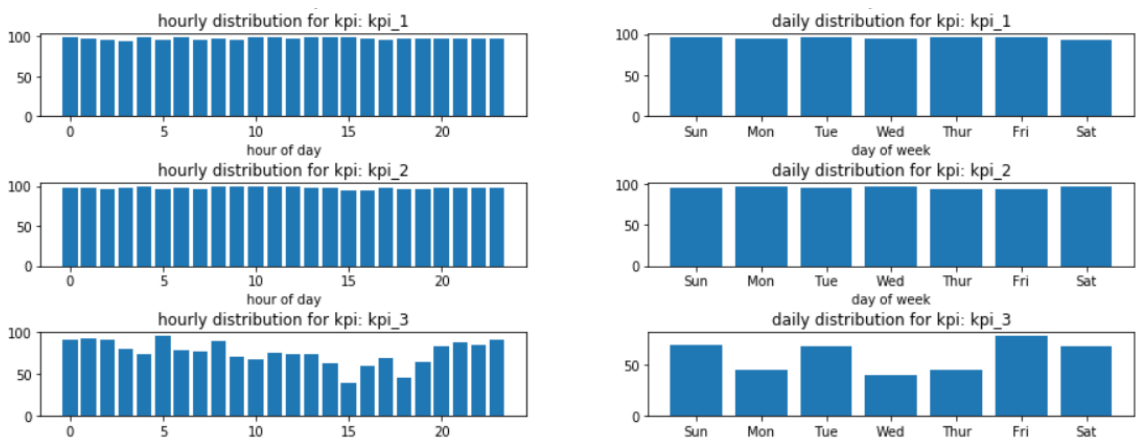


Figure 2.6. Ratio based KPIs distribution

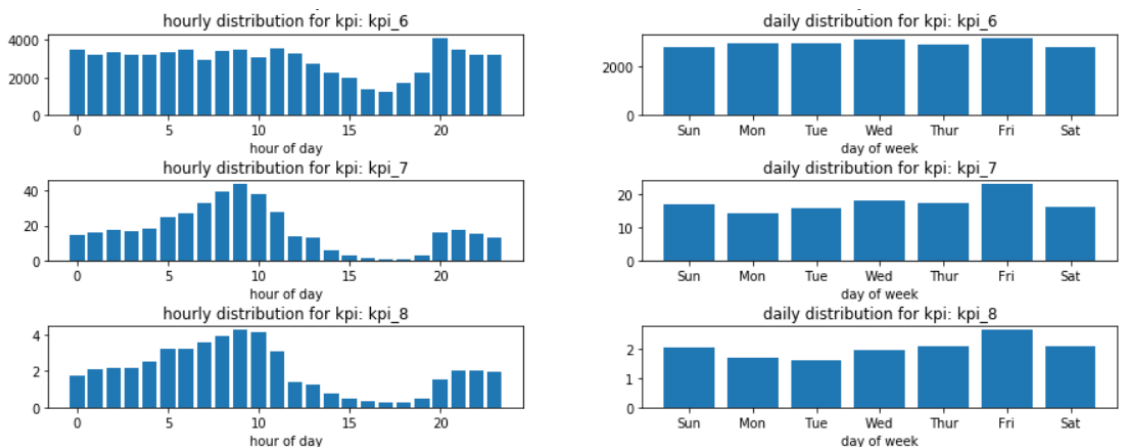


Figure 2.7 Real values KPIs distribution

### 3 Machine Learning (ML)

Machine learning (ML) continue to change the world and is becoming a standard addition to many services and applications workflow to improve the end user experience. Machine Learning is the process of training a model to perform a task on new unseen data without being explicitly programmed to do so. The data used for training the model is called training data, while the new unseen data is called test data (13).

A machine learning workflow usually starts with a problem in hand that is usually partially solved by other means. For example, a search engine that return a list of matching products in response to a user query. The search engine might use a technique like inverted index to return the result of a query. Inverted index is a concept that allows for a fast full-text searches. This traditional technique has its short coming as it is not able to capture the user intent and contextual meaning of a query. ML comes to the rescue and a binary classification model could be plugged to the search engine workflow to return the matching products, obtained by the inverted index method, ordered by their relevancy to the user search query (14).

In recent years, ML started to be considered main stream and its usage expanded to more and more applications. This was influenced by the increasing knowledge in the field and the hundreds of possible libraries and tools that could be used in a fast, easy and affordable manner to integrate to an existing application or service. ML is used in many of our everyday life activities, even when we don't know it. From social media, search engines, stock trading, spam and fraud detection, forecasting, voice recognition, image processing and computer vision and increasingly in the automobile industry (15).

A key factor in the success of any ML algorithm is the data used for training the model. The more data the model sees the better it will generalize to new unseen data. Also, the quality and diversity of the data play an important rule on the final performance of the model.

An example pipeline for a machine learning problem is displayed in the following diagram:

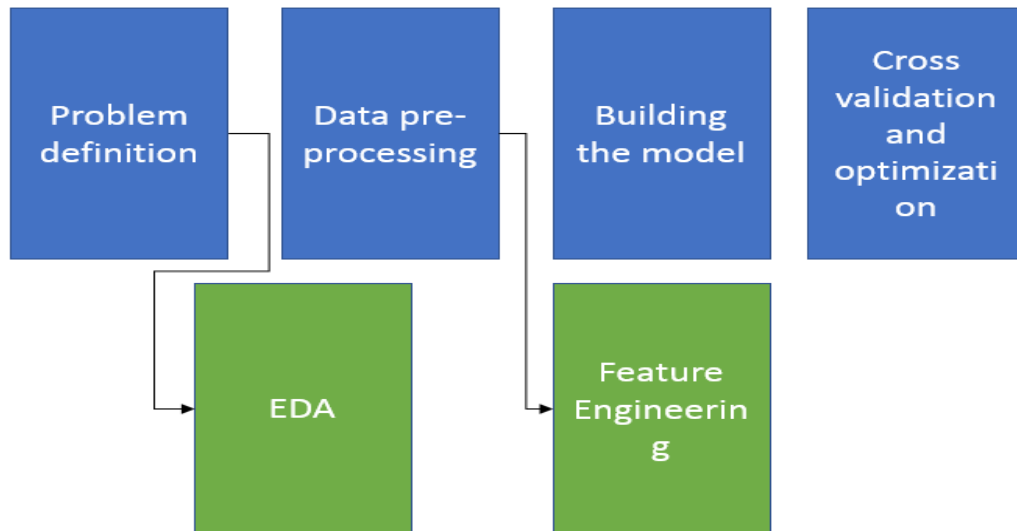


Figure 3.1 Standard machine learning pipeline

Once a business or technical problem is defined, then data gathering and pre-processing could start. Typical activities in data pre-processing includes (13):

1. Data cleaning like removing invalid data and handling of missing fields.
2. Data transformation like grouping and aggregation
3. Scaling or Normalization

Exploratory Data Analysis (EDA), an optional step, helps to get insight about the data and the general problem and in many cases, will create inspiration for new ways of solving the problem.

### 3.1 Cross Validation

When building a ML model, it is crucial to get feedback about the performance or goodness of the model early on and before applying it to real new unseen data. Cross validation is the process of dividing the data set into training and validation sets. The model is trained on the training set and the accuracy is evaluated on the validation set. Cross validation is mainly used in supervised learning because the predictions of the model can be measured against the given ground truth of the training dataset (16).

K-Fold cross validation, a common cross validation technique used in applied machine learning, usually start by shuffling the training data set and splitting it to k equally sized subsets. The model is then executed k times (folds). In each fold, one of the k subsets is used as a validation set and the remaining (k-1) subsets as a training set. The accuracy of the model is then calculated as the average accuracy obtained from all the folds (16).

### 3.2 Feature Engineering

Feature Engineering (FE) is the science and art of extracting new features from the given raw data, usually using field knowledge and accumulated ML experience, to improve the accuracy of the ML model. A feature tells a story or define one aspect of the problem that would help the model to do the required task. It can be given directly in the raw data or can be derived through feature engineering. As an example, a model that is supposed to predict prices of houses based on their size, number of rooms, location and floor number (all given features), would most likely benefit from new engineered feature like the ratio of a house size compared to the average size of houses in the same location. Another example would be to extract the hour, day of week from a given date feature (16).

Even with the wide spread of deep learning and other automated feature generation methods, manual feature engineering and data preparation in general, is still a main differentiator in the performance and accuracy of ML models. Therefore, data scientists often spend considerable amount of time on data preparation and FE before building a model (17).

### 3.3 Feature Scaling

Scaling, also known as normalizing, refers to the operation that we perform on features to change the scale or range of its values. Usually this is done during the data pre-processing step and, depending on the nature of the model, it could have great effect on the performance or accuracy (16).

The features in a data set could have different range of values. For example, the eNodeB KPIs could be presented as a percentage (0-100) or as absolute numbers that represent something like throughput or lost packets counts which could have a very large value range. Scaling is used to prevent features with larger values from dominating the ML model. This is a concern especially if the model uses distances between data points like in clustering where the distance will largely depend on the feature with greater values (16).

In applied machine learning, two common scaling techniques are usually used.

1. Standard scaling: A feature is scaled to have zero mean and one standard deviation.

$$x' = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

2. Min-Max scaling: A feature is scaled to have values in the range from zero to one.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

### 3.4 Clustering Methods

Clustering is an unsupervised learning method that target to group or categorize similar samples of data together. Clustering can be used in exploratory data analysis to find correlation and useful connections in the data set. It could also be used to reduce the amount of data. In anomaly detections, clusters with relatively low number of members could indicate a suspicious incident while samples the farthest to its own cluster centre might be considered a local anomaly (18).

Clustering methods can be divided into two groups: Hierarchical clustering and Partitional clustering.

#### 3.4.1 Hierarchical Clustering

Hierarchical clustering separate samples of data into distinct groups based on some similarity or distance measure between the samples. Similarity between data points could be measured by the distance that separate them or by their correlation. A simple distance measure is the Euclidean distance (18). The Euclidean distance between point  $x$  and  $y$  in dimension  $n$  is defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Hierarchical clustering could be divisive or agglomerative. In divisive clustering the algorithm starts by one cluster that contains all the data points. It then proceeds to recursively split each cluster to two least similar clusters until it ends up with clusters that has only one data point. Agglomerative, on the other hand, starts with each data point in its own cluster and proceeds to merge the clusters that are most similar together until it ends up with one cluster that contains all data points. In both methods, the splitting and merging is presented by a hierarchy called dendrogram. The final clusters are obtained by cutting

the dendrogram at specific level. This usually require field knowledge and experimenting with several possible levels (18).

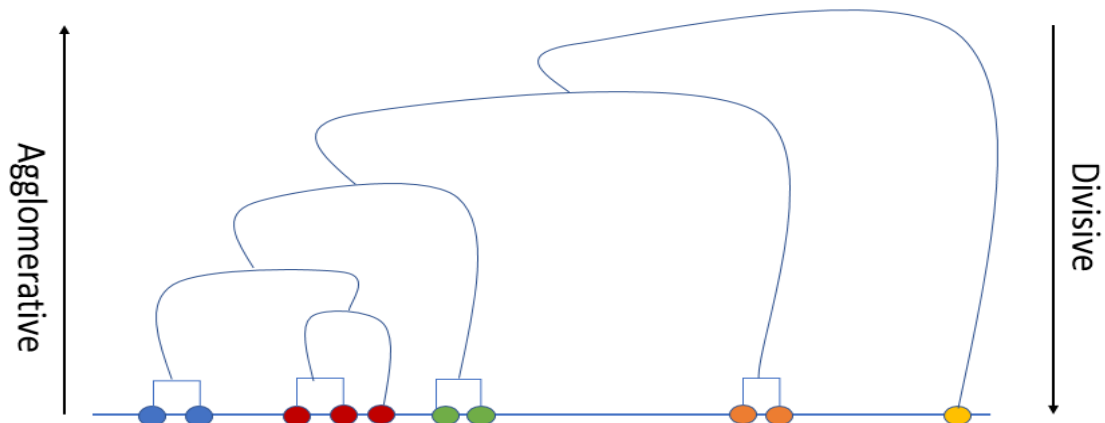


Figure 3.2 Hierarchical clustering

At each step of splitting and merging of clusters, the distances between all possible clusters need to be re-calculated. The distance between two clusters, with one or more data points in each, could be calculated with three different ways (18):

1. Single linkage: the distance between the closet data points in each cluster
2. Complete linkage: the distance between the farthest data points in each cluster
3. Average linkage: the average distance between each data point in the first cluster to each data point in the second cluster

This makes hierarchical clustering, especially when used with average linkage, not very practical for large data sets as it requires huge computational power.

### 3.4.2 Partitional Clustering

The K-Means is probably the most common partitional clustering method. This iterative method starts with K clusters or centroids chosen randomly or otherwise selected by the user. The distance from each data point to each centroid is calculated and data points are assigned to the closest cluster. Each centroid location is then recomputed as the average of the data points assigned to it. The whole process is repeated until data points cluster assignment does not change anymore (19).

Partitional clustering complexity is much less than hierarchical clustering as distances are only calculated from data points to the clusters centroids which make it more feasible to be used especially for larger data set.

## 4 Anomaly Detection

Anomaly detection, also referred to as outlier detection, is a technique of machine learning that addresses the problem of automatically finding patterns and events in data that does not conform to the normal state or behaviour. This is a complex problem as data patterns evolve and change and new trends can emerge at any point of time. Traditionally, anomaly detection was used to remove outliers from the data before applying other machine learning algorithms like classification as these algorithms were sensitive to outliers. Later, researchers started to be interested in the anomalies itself as it usually points to suspicious events in the data and could help businesses to detect problems early on. There are various applications for anomaly detection. For example, it could be used in real time monitoring of sensors, network or resource usage. Other applications are fraud and intrusion detection and recently in the medical field (20).

There are many definitions of an anomaly. All of them agree on the following two characteristics of an anomaly (20):

- An anomaly should have different attributes and features compared to most of other data points.
- Anomalous data points are rare compared to the normal data points.

In a broader sense, anomaly detection is not restricted to machine learning. For example, companies could create dashboards and reports for most important metrics and investigate the charts manually to find outliers (21). This manual work would prove to be expensive and not scalable to higher number of metrics like the case we have with eNodeB KPIs. Another approach would be to use rule-based analysis and specifically set thresholds for each KPI. Setting thresholds for hundreds of KPIs is time consuming and require deep understanding of the general technology and the specific configuration of the telecommunication network where the KPI is used. Thresholds might also need to be adjusted when a new release is rolled out or new sales campaign that affect the traffic pattern is launched. This leads us to anomaly detection by machine learning that will support and complement the existing threshold-based analysis systems.

## 4.1 Supervision Level

The supervision level refers to the degree of availability of labels for the training data. The supervised learning is not discussed as it is not relevant in the context of anomaly detection for this thesis work because fully labelled KPI data is not available.

### 4.1.1 Semi-supervised Learning

In the context of anomaly detection, semi-supervised learning could be used when the training data set could be verified to be free of any outliers. Field experts could manually check the training data and make sure it does not have any outliers. Once this is done then an algorithm like one-class Support Vector Machines (SVM) could be trained on this data and when presented by unseen data the model will be able to classify samples into normal or abnormal labels. In practice the model will give each sample a probability score for being normal. A threshold could be derived by empirical methods to finally decide which samples are indeed the outliers (20).

Semi-supervised learning would deliver better performance compared to unsupervised learning, but it is a challenge to have a clean data set that is outlier free. An automatic way to determine if the training data is not associated with any problems would add an edge and will make this approach practically possible.

### 4.1.2 Unsupervised Learning

In unsupervised learning, contrary to semi-supervised, the training data is not guaranteed to be outlier free (13). In the following chapters, different techniques for detecting anomaly in such a data set will be explained. Two approaches could be chosen when applying unsupervised learning using KPI time series data for eNodeB:

1. Univariate analysis: In this method, the ML model looks at each KPI by itself, learning its pattern and generating a list of anomalies for each KPI. The benefit of univariate analysis is that it is easy to scale up when it comes to computational power. The disadvantage, is that it does not tell the complete story for what have happened, and the amount of anomaly generated might be huge (21).



2. Multivariate analysis: In this method, the model looks at all, or a selected group of, the KPIs at once which gives the model a stronger signal for detecting anomaly (21). Scaling or normalization of the KPIs is crucial to the success of a multivariate method as the KPIs could have different range of values and without scaling, the KPIs with larger values will dominate the results while other KPIs will not have any significant effect (16). Obviously, it is hard to scale up a multivariate method and therefore in this thesis a KPI selection method is applied.

#### 4.2 Type of Anomaly

In any data set, different types of abnormalities or outliers could be found that would define the type of the problem or event that caused the deviation from the norm. **Global** point anomaly refers to a sample in the data set that is clearly far from all other samples and its statistical properties greatly differ from the others. **Local** anomaly is usually associated with a sample that is abnormal compared to a sub-group of the data while might be considered normal in the global context of the data. Another type of anomalies is small clusters of samples that are close to each other but far enough from the rest of the data to be considered anomalous (20).

Anomalies are not always obvious and in general anomaly detection methods would assign a score or probability for being anomalous to data points instead of binary classifying them.

In the KPIs time series data, an anomaly is any unexpected change in the pattern of the time series (21). Many of the KPIs time series have hourly and daily pattern with a clear cycle of ups and downs based on the hour of the day and day of week. A High CPU usage during peak hours might not be considered unusual, while slightly high value in the off-peak hours would be treated as anomaly. This type of anomaly is called **contextual** anomaly. The context in the case of KPIs time series can be introduced into the anomaly detection model by including the hour of the day and day of week as new features (20).

#### 4.3 Anomaly Detection Methods

In this chapter the anomaly detection methods are categorized into four different groups.

#### 4.3.1 Nearest Neighbours Methods

This is a global anomaly detection method where each data point is assigned a score based on the average distance to its K nearest neighbours. On other cases, the distance could be to  $K_{th}$  nearest neighbour instead of the average of all K nearest neighbours. The higher the score or the distance the more likely the data point to be anomalous. Data points with low average distance are close to other data points and are considered normal (20).

The choice of K depends highly on the nature of the data and usually several values are tested, and field knowledge is used to determine the best choice.

#### 4.3.2 Statistical Methods

In statistical methods, a probability function is constructed to model the data and samples with low probability are considered outliers. The probability density function (pdf) provides an estimate of the probability of a sample having certain value by deriving it from smaller and larger values (22).

If a variable x has a normal or gaussian distribution and it has a mean value of  $\mu$  and a standard deviation of  $\sigma$  then its pdf can be presented with the following formula (22):

$$pdf(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

#### 4.3.3 Clustering Methods

Clustering methods can be used to find global or local anomalous data points. In some cases, one cluster will be very different from all the other clusters and would usually contain small number of data points. In this case the micro cluster could form a global anomalous series of data points that can indicate an incident in the data set. On the other hand, the farthest data points from the centre of a normal cluster could be considered a local anomaly for that cluster (20).

#### 4.3.4 Density Based

In these methods, the density of a data point is determined by the average distances to its K nearest neighbours or to the  $K_{th}$  nearest neighbour. If a data point is close to its neighbours, then it is considered to be in a high-density region. Different measurements could be used to calculate the distance like Manhattan or Euclidean (23).

### **Local Outlier Factor (LOF)**

The density of a data point is compared to the density of its  $K$  nearest neighbours. If a data point is in a significantly lower density area compared to its neighbours, then it is highly probable that it is an outlier (23).

### **Influenced Local Outlier (INFLO)**

The density of an examined data point is compared to the densities of its  $K$  nearest neighbours and to an additional set of neighbours. The additional set, also called inverted list of neighbours, are the data points that have the examined data point as one of its  $K$  nearest neighbours. This is particularly useful in detecting anomaly for data points on the border between two regions with significant difference in density. LOF would assign a high anomaly score for these data points while INFLO would also consider the inverted neighbours from the less dense region and produces smaller and more realistic anomaly score (20).

### **Local Outlier Probability (LoOP)**

The distinction of outliers into global and local is usually based on the context where the outlier data point is considered. A data point is globally anomalous, if it was considered in the context of the complete data set. On the other hand, a data point is a local anomaly if it was considered in a local context such as the  $K_{th}$  closest neighbours. (24)

Local outlier methods usually assign a score of outlierness to data points as opposed to binary classification. The Local Outlier Probability technique attempts to solve the problem of interpreting the score produced by outlier detection methods. In many cases, the score is not consistent among different data sets or even within the same dataset but in different local clusters or regions. For example, a data point far from its cluster centroid will have higher score, distance, value in sparse clusters compared to denser clusters. The LoOP provides normalization method to transform the outlierness score to probability in the range of  $[0, 1]$  which can be interpreted as the probability of a data point being anomalous. This probability score is consistent among the complete data set and even for new data sets (24).

#### 4.3.5 Distance Based Methods

Distance based methods are special case of clustering methods. In the former methods, a data point anomaly is determined by its distance to a reference point that is considered optimal compared to the centre of a cluster in the clustering methods. *Kumpulainen* presented an effective and simple method for discovering anomalies in 3G BTS cells using a handful of manually picked KPIs with daily aggregated values. In this sub chapter, *Kumpulainen* approach is presented and it is also mentioned in following chapters for comparison purposes (25).

Instead of linearly scaling the KPI to a range from zero to one, *Kumpulainen* utilized expert knowledge scaling to account for the non-linearity in the interpretation of the KPI values. As an example, A KPI that measure a success ratio would have 100% (scaled to 1) as the optimal value. Instead of scaling a success ratio of 85% to 0.85, as would have been the case in linear scaling, *Kumpulainen* used expert knowledge and scaled the value to 0.2 to account for the fact that an 85% success ratio is considered very poor (25).

For each BTS cell, the Euclidean distance between the previous day KPIs scaled values and the optimal values, a vector of ones, is calculated. The cells with highest values are considered the most anomalous (25).

Advantages of *Kumpulainen* approach:

1. Simple and very easy to understand.
2. Easy to scale as it does not require huge computational power.
3. Expert knowledge provided meaningful insights into scaling the KPIs and therefore interpreting the results are easy and usually provides great insights.

Disadvantages and possible improvement for *Kumpulainen* method:

- 1- While expert knowledge was an advantage, it is also a challenge as acquiring expert knowledge is usually very hard and with the scope of hundreds of KPIs it become virtually impossible. In this next chapter, an automatic way of discovering optimal values for KPIs is presented as an alternative.
- 2- Using aggregated daily averages of the KPIs means that problems that occur during the day might get unnoticed. Increasing the granularity of the KPIs data to hourly bases, as done in this thesis, could provide better coverage.

- 3- Seasonality of the KPIs are not taken into account in terms of natural variation during the hours of the day and also changes based on the day of the week.

#### 4.4 Anomaly Detection in Time Series

A Time Series is a collection of data points over a period of time with certain frequency (26). In the KPI data, an example frequency of time could be one hour and the data points are the value of the KPI at each hour.

Anomaly detection in time series target to find three kinds of issues (27):

1. Outlier: a data point which greatly differ in value than the other data points
2. Change point: a data point where the previous and next data points are greatly different.
3. Anomalous time-series: are time-series that are greatly different than other time-series

In the context of this thesis, a novel way for finding anomalous time series or KPIs would be to compare the same KPI among a larger set of eNodeBs as generally a KPI should behave similarly among different eNodeBs.

##### 4.4.1 Stationary Time Series

The study of the stationary state of a time series is important in building a model that fits the time series. A time series is stationary if its statistical properties does not change over time. Statistical properties are defined as the mean (average), standard deviation and autocovariance. Autocovariance refers to the covariance of the time series at different points of time. For a time series to be stationary, it should have a covariance that is not time dependent (28).

Below are examples of stationary and non-stationary time series (28):

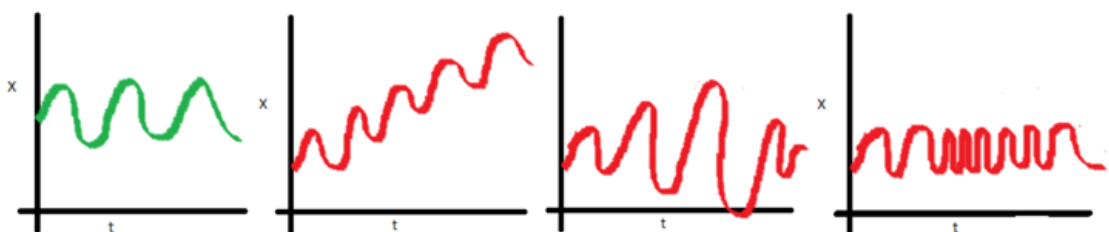


Figure 4.1 Stationary and non-stationary time series

When a time series is non-stationary because the mean is not constant over time then the time series is considered to have *trend*. The trend could be positive or negative depending if the values of the time series increase or decrease over time (26).

Seasonality is another attribute of time series and it is defined as regular changes that occur in time series at specific frequency of time. For example, the values of a time series could peak around the mid of the day and dip into low values around the mid of the night. The seasonality could have different frequency like hourly, daily, weekly, monthly, ..., etc (26).

The eNodeB KPIs were studied to check their stationary state and the results confirmed, to a good degree, that the KPIs time series are stationary in general. For this experiment, the HW KPIs and another random 40 KPIs for a random 20 eNodeBs were investigated. For each time series, a total of 271, the rolling mean, the rolling standard deviation and the Dicky-Fuller test were executed (29).

The Dicky-Fuller test returns three critical values corresponding to the confidence degree of 99%, 95% and 90% of the time series being stationary. In addition, it returns the actual value for the time series test. The results of the test are interpreted by comparing the test value to the confidence thresholds. If the signed value of the test is smaller than a confidence threshold, the time series is said to be stationary with a confidence degree equal to that threshold (29).

The graphs below show the rolling mean, rolling standard deviation and the Dicky-Fuller test for some of the KPIs time series. All the investigated time series were stationary with a 95% or higher confidence rate.



Figure 4.2 Time series with 99% confidence of being stationary

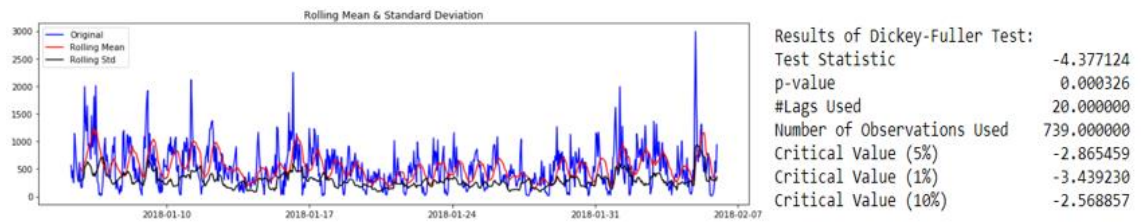


Figure 4.3 Time series with 99% confidence of being stationary

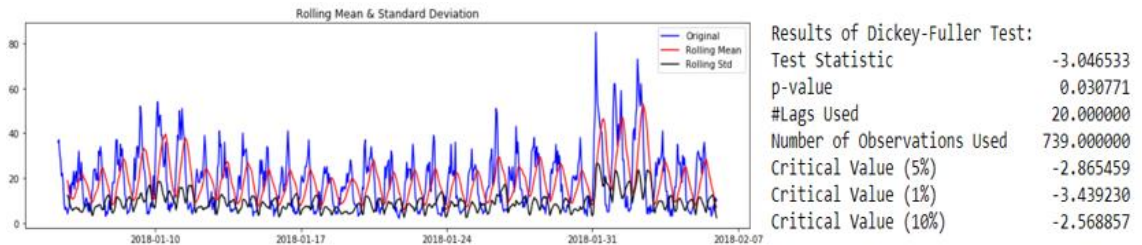


Figure 4.4 Time series with 95% confidence of being stationary

#### 4.4.2 Time series decomposition

The decomposition of time series refers to the separation of the time series into the following components: *trend*, *seasonality* and *residual*. Residual is what is left in the time series after extracting the trend and seasonality (26).

Removing trend and seasonality enables better study of the actual patterns in a time series. It is also used in time series forecasting where the actual forecasting is applied to the residual part of the time series and, at the end, the trend and seasonality are added back to construct the final forecasted time series (26).

Decomposition was applied to HW KPIs of ten randomly selected eNodeBs. The below graphs show the original time series, trend, seasonality and residual plots for a couple of the KPIs.

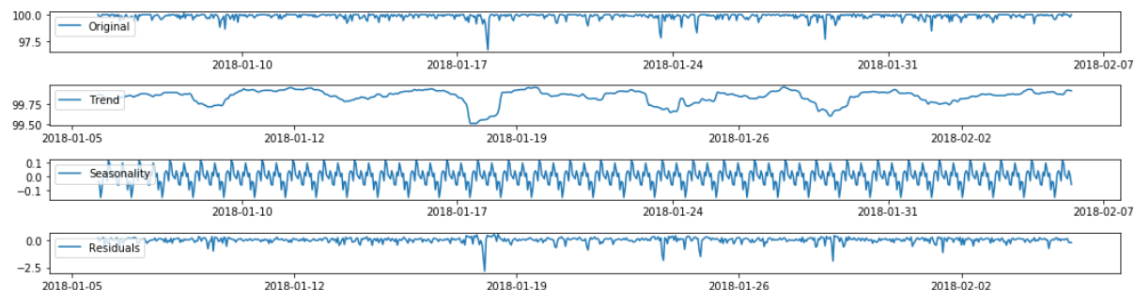


Figure 4.5 Decomposition of a ratio based KPI

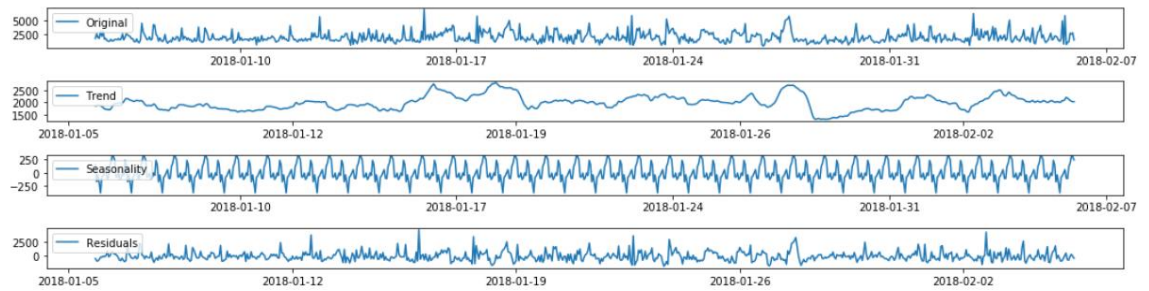


Figure 4.6 Decomposition of a real value based KPI

#### 4.4.3 Anomalous Time Series

In studying time series, two different types of unexpected behavior could be investigated. A time series could be explored for possible *point anomaly* where specific point in time deviate from what is considered normal at this time, considering any possible seasonality. Another way of finding anomaly is to consider a group of similar time series at once and find the one that deviate greatly from the otherwise similar time series (27).

Finding similar time series could be done using field knowledge or automatically by using clustering methods. In the context of this thesis, the most natural way of finding similar time series is by grouping them by KPI id.

To find the anomalous time series, the distance to the mean, or median, of the all other time series in the same group should be calculated. A threshold could be found empirically, to define the distance that indicates anomaly. If the distance between a time series and the mean of the group of time series it belongs to is higher than the defined threshold, the time series is considered anomalous (27).

In chapter 5, both approaches of point anomaly and anomalous time series will be used and compared to detect the most anomalous eNodeBs. An eNodeB could be considered anomalous if it has a lot of anomalous points of time for the selected KPIs or if it has a lot of KPIs that are considered anomalous compared to the other eNodeBs.



#### 4.5 Azure Time Series Anomaly Detection

This chapter presents an example of ready-made anomaly detection software by Microsoft Azure cloud. It is an API built with Azure machine learning and can be used to discover unexpected behaviour in time series data (30).

The API can discover three kinds of unexpected behaviour in a time series (30):

1. Upward or downward trend: detects the point in time when a time series starts experiencing an upward or downward trend.
2. Level changes: detects the point in time when a time series range or level of values changes.
3. Point anomaly: detects the points in time when the time series takes in values either too high or too low.

For this experiment, the following data preparation steps were taken:

1. HW related KPIs and another 40 randomly selected KPIs for a 20 randomly selected eNodeBs are initially examined, a total of  $40 * 20 = 800$  time series.
2. Each KPI time series is then aggregated on hourly basis.
3. To ensure data continuity, any time series with missing values is removed from the experiment, a total of 218 time series are left.
4. Each time series is then fed to the Azure API (<http://anomalydetection-aml.azurewebsites.net>) for anomaly detection.

Below are few diagrams of the result of the anomaly detection by the Azure API (more graphs can be found from Appendix 1):

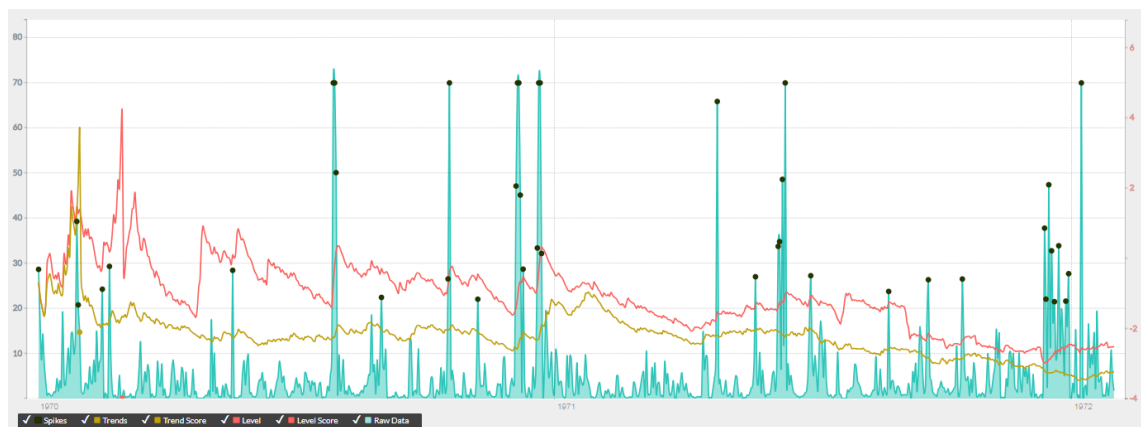


Figure 4.7 Time series with many peaks (black dots), a change point (red dot) and a trend point (yellow dots)

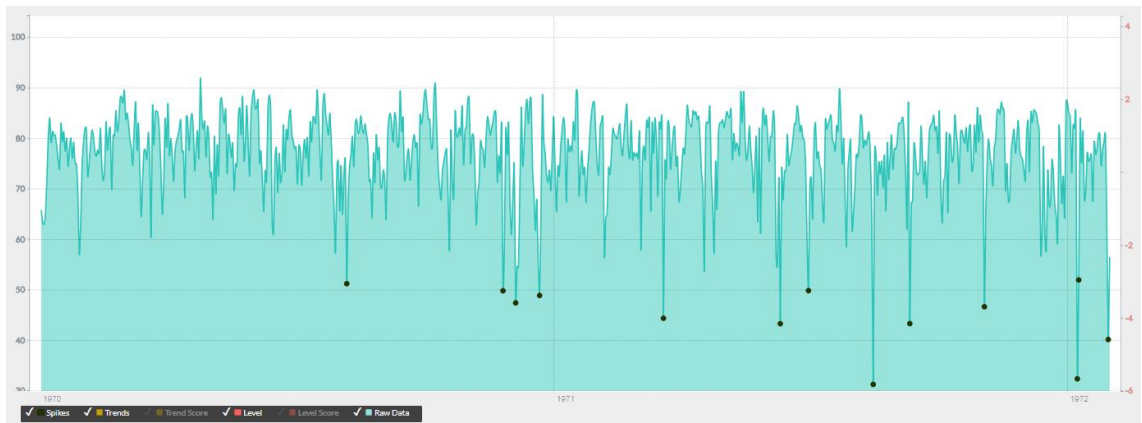


Figure 4.8 Time series with many dips (black dots)

## 5 eNodeB Anomaly Detection

This is the chapter where the actual solution is presented and the results are outlined and compared.

eNodeB KPIs time series will be used to detect anomaly with granularity of one hour. If a KPI has more than one value in one hour, then the average of the values will be used for that hour. KPIs are then scaled to range (0,1) using min-max scaling to prevent KPIs with higher value ranges from dominating other KPIs. The analysis is done on cell level. The cells are ranked by the count of anomalous data points they have. Cells with high anomalous data points should be flagged and investigated further. Finally, the anomalous data points from one eNodeB are clustered to discover incident types in the eNodeB level.

The first three methods in this chapter all fall under the umbrella of finding anomalous data points or points in time in the KPI data. In chapter 5.5, on the other hand, an alternative method for finding anomalous eNodeBs is presented. In the latter method, anomalous time series as a whole is used as a criterion for deciding the final rank of anomaly of the eNodeBs.

### 5.1 Probability Density Function (PDF)

Anomalous data points, in this method, are the data points that have low probability based on the multivariate gaussian distribution. A threshold need to be determined to decide the probability limit to consider a data point anomalous. A too high threshold would result on a large number of discovered anomalous while a too low limit would mean missing some data points that are actually anomalous. A threshold was derived based on several experiments to accomplish a good balance.

For this method, HW related KPIs were used. All KPIs time series were aggregated to one-hour interval. If a KPI had higher granularity than one hour, then the mean value of the KPI is used for the hour.

Secondly, the hourly aggregated time series were investigated for missing values. Missing values can skew the statistical properties of a time series and hence they were removed from the scope of applying this method. After the filtering step, a total of 34 eNodeBs and 6 HW KPIs were available for investigation. The same procedure was repeated for all the available cells in each eNodeB.

To account for seasonality, each KPI time series was decomposed to trend, seasonality and residual signals. The PDF algorithm was applied to the residual part of the KPI.

### 5.1.1 Results

The following graphs show anomalous data points in two dimensions by plotting only 2 KPIs. The plots are generated using two different values for the threshold to demonstrate how using higher value could result in over sensitivity of finding anomaly.

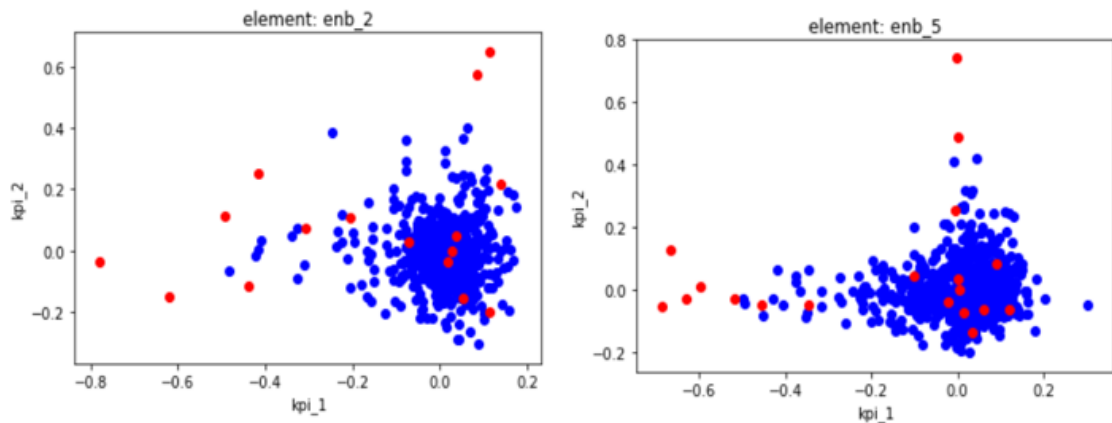


Figure 5.1 Two dimensions anomalous data points with 0.005 threshold

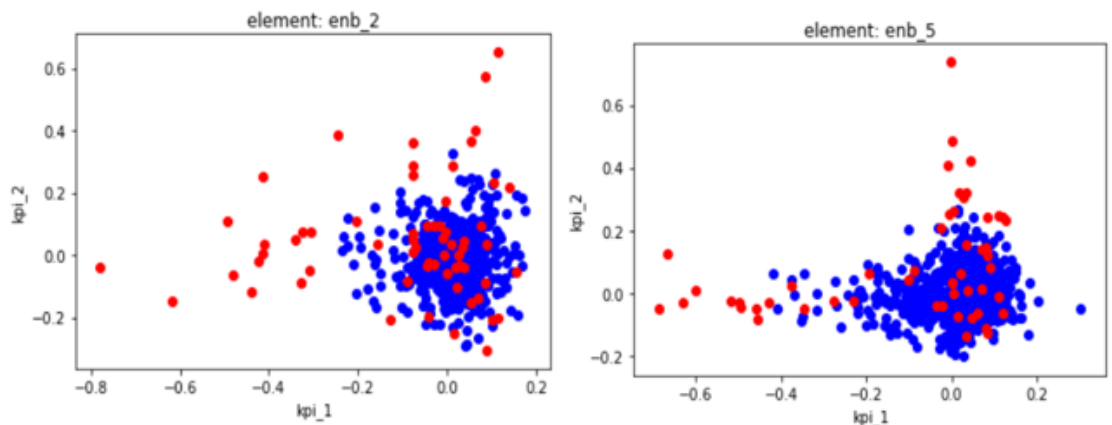


Figure 5.2 Two dimensions anomalous data points using 3.0 threshold

The red data points in the middle of the graphs are due to the fact that anomaly was calculated based on 6 KPIs while the plot only represented 2 KPIs.

Using 0.005 threshold, the eNodeBs cells had between 7 and 23 anomalous data points. The total number of data points was 760 that account to about one-month period. Top and least 5 anomalous eNodeBs are shown in the below tables:

eNodeB	Cell	Anomalous data points count
enb_9	0	23
enb_21	0	22
enb_6	0	19
enb_2	0	19
enb_16	0	19

Table 5.1 Five top anomalous eNodeBs using PDF

eNodeB	Cell	Anomalous data points count
enb_1	0	7
enb_22	0	9
enb_15	0	11
enb_18	0	11
enb_23	0	12

Table 5.2 Five least anomalous eNodeBs using PDF

Examining the individual KPI graphs for the most anomalous eNodeB, *enb\_9*, shows indeed some strange behaviour toward the end of the observation period. In particular, a KPI related to throughput, that could indicate failure in the radio module and a dip in traffic, shows abnormal range of values in the last few days of the observation period. About half of the discovered anomalies, using the PDF method, are found at the same suspicious period.

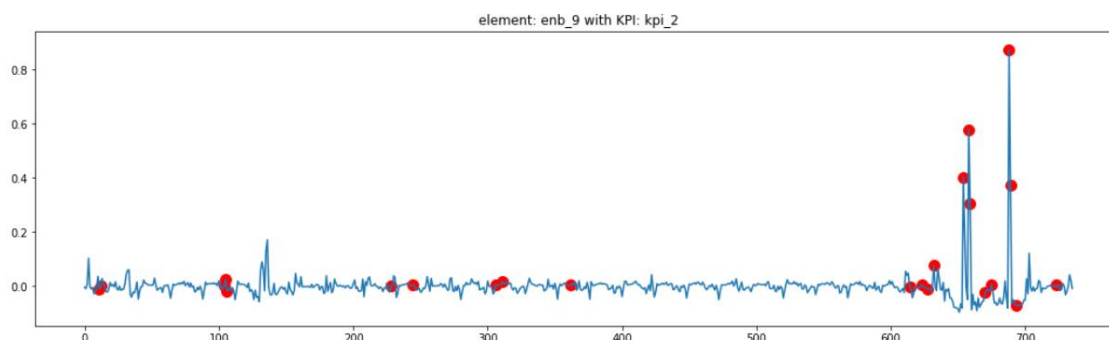


Figure 5.3 kpi\_2 graph for enb\_9 using PDF method

Combined and individual KPI graphs for the most and least anomalous eNodeBs can be found from Appendix 2.

## 5.2 Clustering Based Anomaly Detection

In this method, clustering using K-Means is applied at cell level. The time context is introduced to account for natural variation in the KPIs by creating two new features, columns, for the hour of the day and the day of the week (dow).

<b>Time</b>	<b><math>kpi_1</math></b>	<b><math>kpi_2</math></b>	<b>...</b>	<b><math>kpi_m</math></b>	<b>Hour</b>	<b>dow</b>
t1	$val_{11}$	$val_{12}$	...	$val_{1m}$	12	1
t2	$val_{21}$	$val_{22}$	...	$val_{2m}$	13	1
...	...	...	...	...	14	1
tn	$val_{n1}$	$val_{n2}$	...	$val_{nm}$	15	1

Table 5.3 input matrix for the clustering method

The logic for finding the anomalous data points and the three parameters that should be determined empirically are:

1. The number of clusters. Three, five and eight clusters were tested.
2. The threshold for the distance from a cluster centroid. For each data point, the distance to its cluster centroid is calculated. As distances could be in different scale in different clusters, using a relative value should be considered. Normalizing the distance by the average or maximum distance in a cluster is used. The data points that are more far than the chosen threshold from their cluster centroid are considered anomalous.
3. The threshold for small cluster size. In case a cluster is way too small, then all the data points in that cluster are considered anomalous. Alternatively, the distances to the nearest normal size cluster could be calculated and used for determined anomaly.

### 5.2.1 Results

Using 3 clusters, 0.8 as a threshold for the distance from cluster centroid and 20 as the limit for considering a cluster too small, the eNodeBs cells had between 3 and 34 anomalous data points. Top and least 5 anomalous eNodeBs are shown in the below tables:

eNodeB	Cell	Anomalous data points count
enb_26	0	34
enb_9	0	22
enb_5	0	20
enb_27	0	18
enb_22	0	17

Table 5.4 Five top anomalous eNodeBs using clustering with parameters (3, 0.8, 20)

eNodeB	Cell	Anomalous data points count
enb_1	0	3
enb_6	0	3
enb_19	0	3
enb_30	0	4
enb_28	0	4

Table 5.5 Five least anomalous eNodeBs using clustering with parameters (3, 0.8, 20)

Some similarities can be noticed between the anomalous eNodeBs found using the PDF method and the clustering method. For example, the least anomalous eNodeB, enb\_1, was the same in both methods. Also, the most anomalous eNodeB found by the PDF method ended up as the second most anomalous eNodeB in the clustering method. However, the rest of the top and least lists are significantly different as the algorithms works differently and the choice of the hyper parameters like thresholds and count of clusters greatly affects the results of the latter algorithm.

Examining the individual KPI graphs for the most anomalous eNodeB, enb\_26, shows that the clustering method, with the used hyper parameters, might have been too sensitive and has generated ways too many false positive predictions. Example in the below graph:

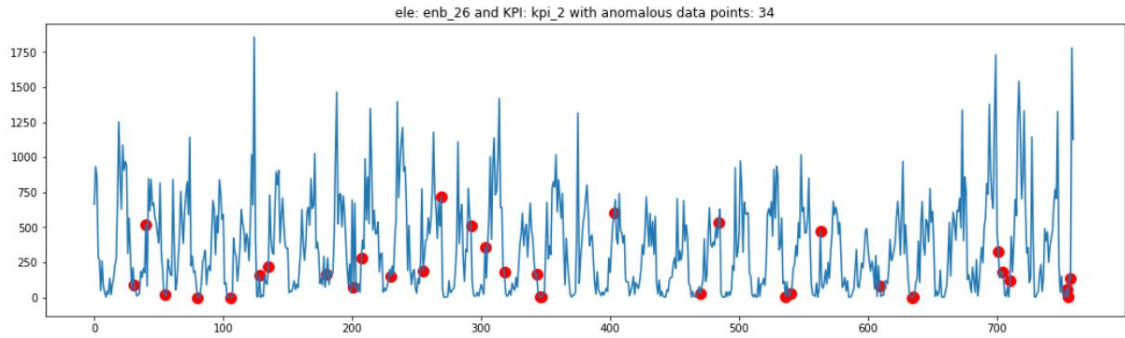


Figure 5.4 kpi\_2 graph for enb\_26 using clustering method and parameters (3, 0.8, 20)

Increasing the threshold for the relative distance to 0.9 seems to help in decreasing the sensitivity. Below is the combined graph for all the KPIs for the same eNodeB, enb\_26, using the higher threshold of 0.9:

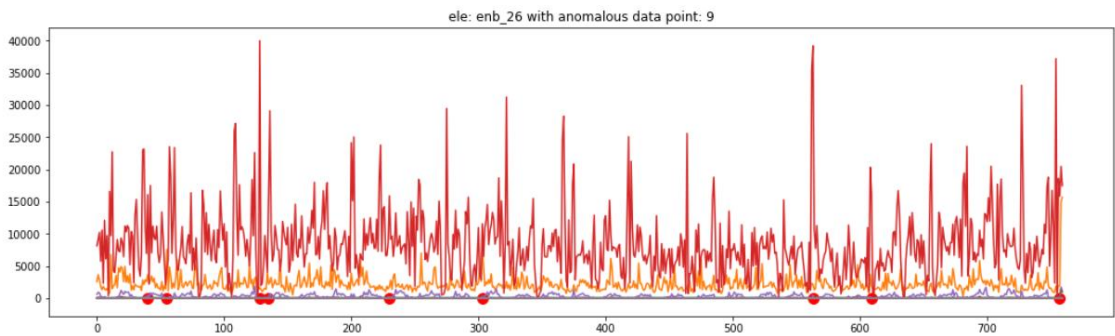


Figure 5.5 Combined KPIs graph for enb\_26 using clustering method and parameters (3, 0.9, 20)

Using the higher threshold has also changed the list of the top anomalous eNodeBs, placing enb\_9 at the top which agree with the result from the PDF method from the previous chapter.

eNodeB	Cell	Anomalous data points count
enb_9	0	17
enb_18	0	10
enb_26	0	9
enb_22	0	8
enb_7	0	8

Table 5.6 Five top anomalous eNodeBs using clustering with parameters (3, 0.9, 20)

More graphs for the top and least anomalous eNodeBs can be found from Appendix 3.



### 5.3 Distance Based with Custom References

In this method, the distance of each data point from a reference value is calculated. The reference or optimal values have two characteristics. Firstly, they are found automatically and without any expert knowledge. Secondly, the optimal values are context aware to account for natural variation of KPI values over the hour of the day and over the different days of the week.

#### 5.3.1 Discovery of KPI Optimal Values

The target of this step is to discover automatically an optimal value for each KPI for each time data point. Introducing the context of time into the model is a novel way to account for expected variation in the KPI values that were discovered during the exploratory data analysis that was presented in the previous chapters.

The automatic discovery of the optimal values is another advantage for this method as expert knowledge is usually hard to obtain and as the number of KPIs grows, it becomes practically impossible to define optimal values for all important KPIs, let alone defining optimal values for different time context.

This technique makes the following two assumptions:

1. A KPI usually has the same pattern and distribution cross different cells and eNodeBs.
2. Outliers in a KPI time series are rare.

Based on these two assumptions, the optimal values for each KPI is driven from the KPI values for all the cells and eNodeBs. Technically, the optimal value for each KPI at each time point is calculated as the median of the KPI values across all the cells in all the eNodeBs at that time point. Using the median instead of mean would reduce the effect of outlier values.

### 5.3.2 Discovering Anomalies

The result from the previous step is a vector of optimal values for each KPI. The length of the vector is equal to the count of time data points. The next step is to work on a cell level and multivariate space to calculate the distance from optimal value for each time data point for each cell.

To illustrate the method, the below table present the data for one cell and using only two KPIs,  $kpi_a$  and  $kpi_b$ , where,  $val_{tn_a}$  is the actual value for  $kpi_a$  at time  $n$ .

Time	$kpi_a$	$kpi_b$
t1	$val_{t1_a}$	$val_{t1_b}$
t2	$val_{t2_a}$	$val_{t2_b}$
...	...	...
tn	$val_{tn_a}$	$val_{tn_b}$

Table 5.7 Data for one cell and using 2 KPIs

The optimal values for these two KPIs are presented with the following two vectors:

$$opt_a = (opt_{t1_a}, opt_{t2_a}, \dots, opt_{tn_a})$$

$$opt_b = (opt_{t1_b}, opt_{t2_b}, \dots, opt_{tn_b})$$

Where,  $opt_{tn_a}$  is the optimal value for  $kpi_a$  at time  $n$ .

The Euclidean distance is then calculated between each time point ( $val_{tn_a}$ ,  $val_{tn_b}$ ) and the optimal values at the same time point ( $opt_{tn_a}$ ,  $opt_{tn_b}$ ). This is illustrated in the following table.

Time	Distance
t1	$\sqrt{(val_{t1_a} - opt_{t1_a})^2 + (val_{t1_b} - opt_{t1_b})^2}$
t2	$\sqrt{(val_{t2_a} - opt_{t2_a})^2 + (val_{t2_b} - opt_{t2_b})^2}$
..	....
tn	$\sqrt{(val_{tn_a} - opt_{tn_a})^2 + (val_{tn_b} - opt_{tn_b})^2}$

Table 5.8 Euclidean distance for each time point for one cell

The result for each cell is a list of absolute distances from the optimal values. A threshold should be used to find the minimum distances to consider a data point anomalous. Then the cells would be ranked by the number of anomalous data point, the data points where the distance from optimal was greater than the threshold.

### 5.3.3 Results

Different thresholds were tested and a value of 1.15 seemed to give the most reasonable results and a good sensitivity level. Top and least 5 anomalous eNodeBs are shown in the below tables:

eNodeB	Cell	Anomalous data points count
enb_9	0	18
enb_8	0	15
enb_6	0	12
enb_33	0	8
enb_22	0	6

Table 5.9 Five top anomalous eNodeBs using distance-based method with 1.15 threshold

eNodeB	Cell	Anomalous data points count
enb_16	0	0
enb_10	0	0
enb_34	0	0
enb_15	0	1
enb_2	0	1

Table 5.10 Five least anomalous eNodeBs using distance-based method with 1.15 threshold

Lowering the threshold increases the count of anomalous data points found but does not seem to, significantly, change the order of anomalous eNodeBs. Below is the top 5 anomalous eNodeBs using a threshold of 1.1.

eNodeB	Cell	Anomalous data points count
enb_9	0	36
enb_8	0	30
enb_6	0	21
enb_24	0	14
enb_33	0	13

Table 5.11 Five top anomalous eNodeBs using distance-based method with 1.1 threshold

Examining the individual KPI graphs for the most anomalous eNodeB, enb\_9, shows that this method of anomaly detection, as expected, does not perform very well in finding individual anomalous data points as many completely normal looking data points are flagged with red in figure 5.6. The method, instead, focus on finding data points that are significantly different compared to other eNodeBs.

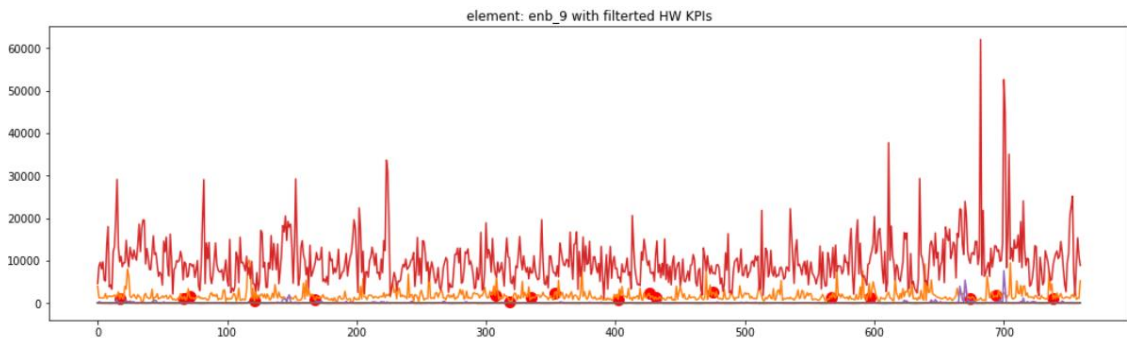


Figure 5.6 Combined KPIs graph for enb\_9 using 1.15 threshold

## 6 Comparison and Conclusion

### 6.1 Comparison of Used Methods

Three methods were used to find the top anomalous eNodeBs using the KPI time series data. All three methods have agreed on the top anomalous eNodeB, enb\_9, and the list of the top 5 and top 10 anomalous eNodeBs had a lot of common eNodeBs between the three methods but was, noticeably, different using the clustering method.

Each method had a number of parameters to be tuned or adjusted which presented the complexity of the method and its ability to be easily applied to new data set. The clustering method was the most complex as it had three different parameters that needed to be adjusted which are: the number of clusters, the distance from a cluster centroid and finally the size of small cluster. On the other hand, the distance based with custom references, was the least complex as it had only one parameter to adjust which is the threshold for the distance from the optimal values.

Another aspect for comparison is the sensitivity of the method in regard to the choice of used parameters and thresholds. The PDF and clustering methods were found to be extremely sensitive to the choice of used threshold. For example, it was clear how changing the threshold for the distance from cluster centroid, in the clustering method, had changed the order of the top anomalous eNodeBs. While, the distance based with custom references method had the least sensitivity to the value of the used threshold.

The first two methods focused on finding point anomaly that are greatly different than the other points in the context of the same eNodeB. On the other hand, the distance based with custom references method, focused on finding anomalous data points that are greatly different than other points considering all other eNodeBs.

The below table summarize the advantages and disadvantages of the three used methods:

Method	Advantages	Disadvantages
PDF	<ol style="list-style-type: none"> <li>1. Easy to implement</li> <li>2. Good in finding anomalous data points</li> </ol>	<ol style="list-style-type: none"> <li>1. Sensitive to the choice of parameters</li> </ol>
Clustering	<ol style="list-style-type: none"> <li>1. Easy to implement</li> <li>2. Good in finding anomalous data points</li> </ol>	<ol style="list-style-type: none"> <li>1. Too many parameters to tune</li> <li>2. Sensitive to the choice of parameters</li> </ol>
Distance based with custom references	<ol style="list-style-type: none"> <li>1. Easy to implement</li> <li>2. Only one parameter to tune</li> <li>3. Not sensitive to the choice of parameters</li> <li>4. Can discover non-obvious anomalies</li> </ol>	<ol style="list-style-type: none"> <li>1. Assume a KPI time series cross all eNodeB has similar distribution</li> </ol>

Table 6.1 Comparison of the three used anomaly detection methods

#### 6.1.1 Difference from Kumpulainen Approach

In chapter four, Kumpulainen approach for anomaly detection was explained (25). The main differences between the distance based with custom references method, presented in the previous chapter, and Kumpulainen method are:

1. Higher resolution for the KPIs is used. All KPIs time series are aggregated on hourly basis. In Kumpulainen approach, one daily value for each KPI was used. Using hourly aggregation ensure that patterns and changes of the KPI during the day are not overlooked.
2. Time context is introduced in calculating the optimal or reference values. Instead of having one reference or optimal value for a KPI, an optimal value per hour and day is used. This is important because of the natural and expected variation in KPI values based on hour of day and day of week.
3. The optimal values are computed automatically instead of based on expert knowledge. This is a big advantage because obtaining expert knowledge for each KPI is practically difficult. Automating enable the usage of more KPIs which would improve the overall accuracy of detecting anomalies.

## 6.2 Comparison with Threshold Based Methods

As can be seen from the below graph, using a threshold for the ratio based HW KPIs would not lead to find any anomaly as the value of the KPI does not drop below 98% during the whole observation period for the most anomalous eNodeB, enb\_9, which confirms that using machine learning is a better choice for this problem.

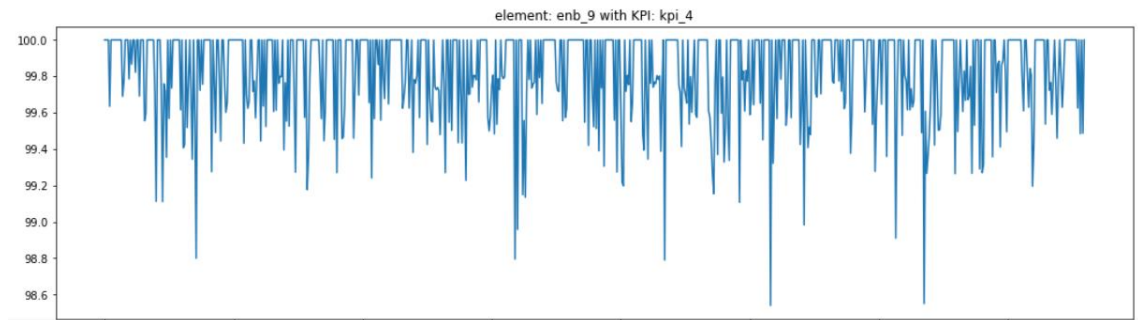


Figure 6.1 Ratio based KPI time series for enb\_9

## 6.3 Verification of Results

The verification process was challenging as logs and other information related to the eNodeBs were missing from the time frame of the studied KPIs. However, two factors contributed, to a good extent, to confirming the findings presented in the previous chapter:

1. Manual inspection of the KPIs time series graphs confirms that the most anomalous eNodeBs found in the previous chapter, enb\_9, has indeed a very abnormal pattern at the end of the observation period. In particular, a KPI that is related to throughput had a lot of anomalous data points in the last few days of the studied period.
2. The three used anomaly detection methods, though all different in the way they work and find anomaly, still agreed on the top anomalous eNodeB which, indirectly, confirms the suspicious abnormality observed in the KPIs graphs of that eNodeB.

## 6.4 Follow up and Next Steps

A follow-up research could focus on extending the work of this thesis by:

- Including higher amount of eNodeBs
- Using different or bigger set of KPIs
- Once the anomalous data points are discovered in a cell level, a clustering method could be applied to all the anomalous data points cross all cells for the

same eNodeB. This could be useful in discovering incidents in eNodeB level as abnormality in multiple cells are most likely to be correlated.

Another possible approach for finding anomalous eNodeBs, in a future research, is to calculate the distance of each KPI time series from the average of the same KPI cross all other eNodeBs. This would require using greater amount of KPIs than what was used in this thesis. HW related KPIs. The eNodeBs are then ordered by the amount of anomalous KPIs. This approach has the advantage of being able to detect anomaly of a KPI even if it does not have many anomalous data points as it considers the total distance of all the data points to the reference value.



## References

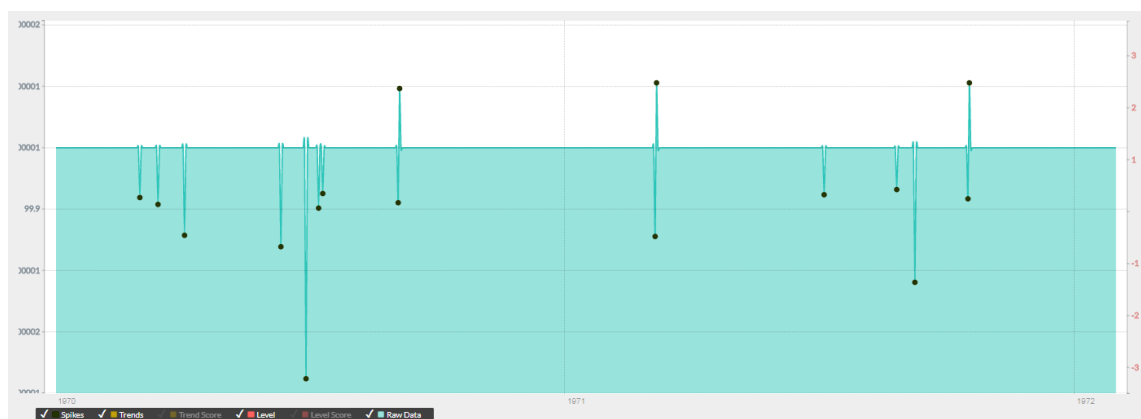
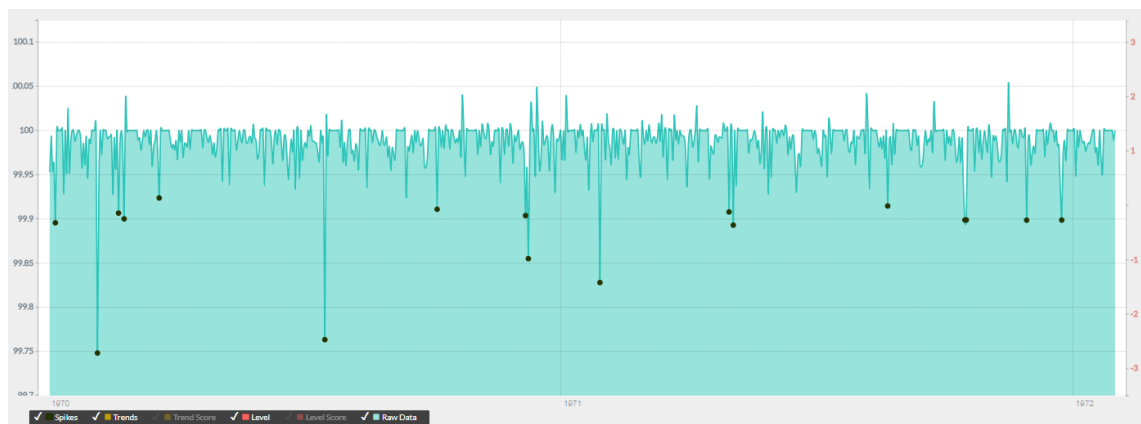
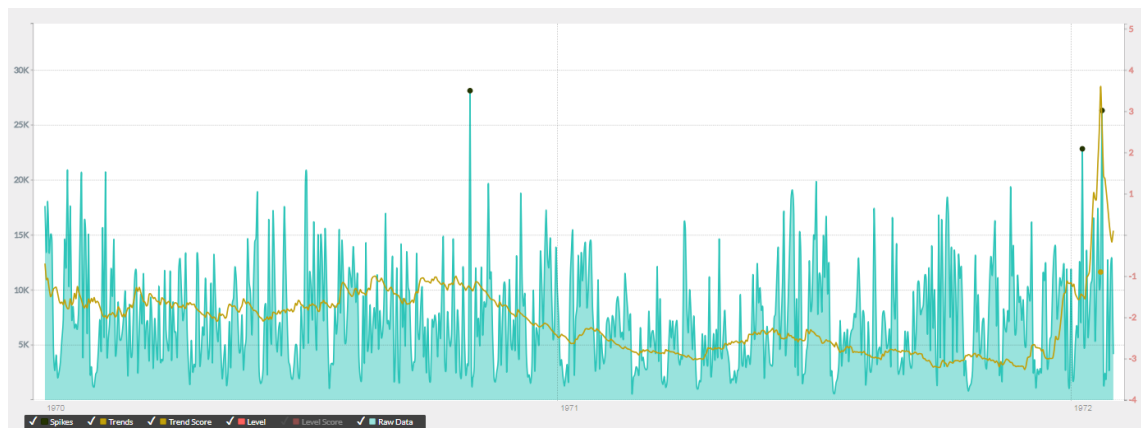
1. Nokia. Architecture of LTE network. [Online].; 2018 [cited 2018 08 13. Available from:  
[https://skylabx.int.net.nokia.com/product/LTE\\_Radio\\_Access/4/release/LTE\\_18/12/document/LTE\\_System\\_Description/ciKygf/topic/204216479\\_\(Internal\\_Nokia\\_Network\)\)](https://skylabx.int.net.nokia.com/product/LTE_Radio_Access/4/release/LTE_18/12/document/LTE_System_Description/ciKygf/topic/204216479_(Internal_Nokia_Network))).
2. Nokia. Flexi Multiradio BTS. [Online].; 2018 [cited 2018 February. Available from:  
[https://skylabx.int.net.nokia.com/product/Flexi\\_EDGE\\_GSM-R\\_BTS/60/release/Flexi\\_EXR5\\_2/146/document/BSS\\_Description/diGD3i/topic/DN03541632\\_4\\_\(Internal\\_Nokia\\_Network\)](https://skylabx.int.net.nokia.com/product/Flexi_EDGE_GSM-R_BTS/60/release/Flexi_EXR5_2/146/document/BSS_Description/diGD3i/topic/DN03541632_4_(Internal_Nokia_Network)).
3. www.statista.com. Global mobile data traffic 2016-2021. [Online].; 2016 [cited 2018 February. Available from: [www.statista.com](http://www.statista.com).
4. Nokia. Preventive Services Marketing Presentation. Presentation. Espoo.; Nokia Global Services; 2018.
5. Nokia. AHC use case - Stability fault history. Specification. Espoo.; 2017.
6. Nokia. AHC use case LTE - Active Fault Analysis for eNodeB. specification. Espoo.; 2017.
7. Nokia. Monitoring and Measuring System in LTE RAN. 2014..
8. Nokia. AHC use case - abnormal throughput. 2017..
9. Nokia. AHC use case - Accessibility and retainability. 2017..
10. Nokia. Counter Checks Sleeping Cells. 2014..
11. Nokia. Proactive Care Services Use Cases. 2016..
12. Nokia. Description of eNodeB HW related KPIs [email exchange with KPI expert in Nokia].; 2018 [cited 2018 05.
13. Bell J. Machine Learning: Hands-On for Developers and Technical Professionals. 1st ed.: Wiley; 2014.
14. Haubert E. OpenSource Connections, What is Learning To Rank? [Online].; 2017 [cited 2018 08 13. Available from:  
<https://opensourceconnections.com/blog/2017/02/24/what-is-learning-to-rank/>.
15. Ng AYT. Coursera, Machine Learning by Andrew Ng Stanford University, Lecture 1. [Online]. [cited 2018 08 13. Available from:  
<https://www.coursera.org/learn/machine-learning>.

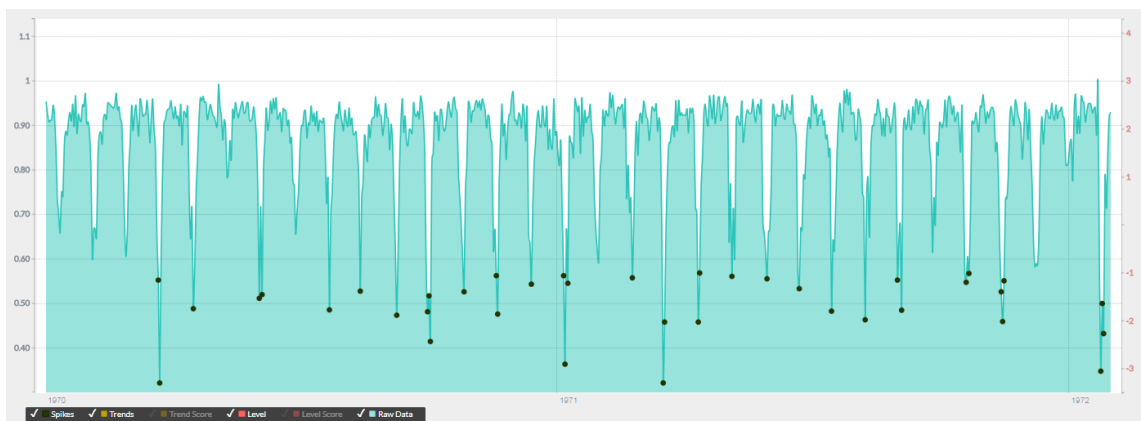
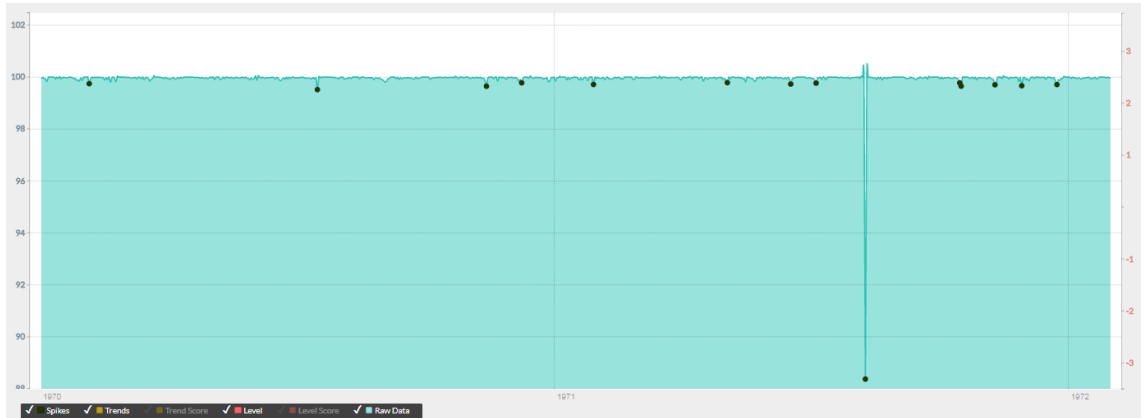
16. Raschka S. Python Machine Learning: Packt Publishing; 2015.
17. Sarkar D. towardsdatascience. [Online].; 2018 [cited 2018 February. Available from: <https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b>.
18. Berkhin P. Survey of Clustering Data Mining Techniques. 2018..
19. Hu Q, Wu J, Bai L, Zhang Y, Cheng J. Fast K-means for Large Scale Clustering. 2017 November.
20. Goldstein M, Uchida S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. 2016. Public Library Of Science PLOS.
21. Anodot. Ultimate Guide to Building A Machine Learning Anomaly Detection System, Part1: Design Principles. www.anodot.com. 2017.
22. Ng AYT. Coursera, Machine Learning by Andrew Ng Stanford University, Lecture 15. [Online]. [cited 2018 07 20. Available from: <https://www.coursera.org/learn/machine-learning>.
23. Breunig M, Kriegel HP, Ng T, Sander J. LOF: Identifying Density-Based Local Outliers. 2000 June; 29(2).
24. Kriegel HP, Kröger , Schubert E, Zimek A. LoOP: Local Outlier Probabilities. 2009 November.
25. Kumpulainen P. Anomaly Detection for Communication Network Monitoring Applications. 2014. PHD thesis, Tampere University of Technology.
26. Aarshay J. analyticsvidhya, A comprehensive beginner's guide to create a Time Series Forecast. [Online].; 2016 [cited 2018 February. Available from: <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/>.
27. Laptev N, Amizadeh S, Flint I. Generic and Scalable Framework for Automated Time-series Anomaly Detection. 2018..
28. TAVISH S. analyticsvidhya, A Complete Tutorial on Time Series Modeling in R. [Online].; 2015 [cited 2018 07 13. Available from: <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>.
29. Zaiontz C. www.real-statistics.com, Real Statistics Using Excel. [Online]. [cited 2018 07 16. Available from: <http://www.real-statistics.com/time-series-analysis/stochastic-processes/dickey-fuller-test/>.

30. Kirpal A, Ericson G, Martens J, Rohm W. www.microsoft.com, Machine Learning Anomaly Detection API. [Online].; 2017 [cited 2018 07 19. Available from: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/apps-anomaly-detection-api>.
31. Anodot. Ultimate Guide to Building A Machine Learning System, Part 2: Learning Normal Time Series Behaviour. www.anodot.com. 2017.
32. Anodot. Ultimate Guide to Building A Machine Learning System, Part 3: Correlating Abnormal Behavior. www.anodot.com. 2017.
33. Dunning , Friedman. Practical Machine Learning, A New Look at Anomaly Detection. First Edition ed. Loukides M, editor.; 2014.
34. Nokia. Networks Integrated Data Dictionary. 2017..

## Azure Anomaly Detection

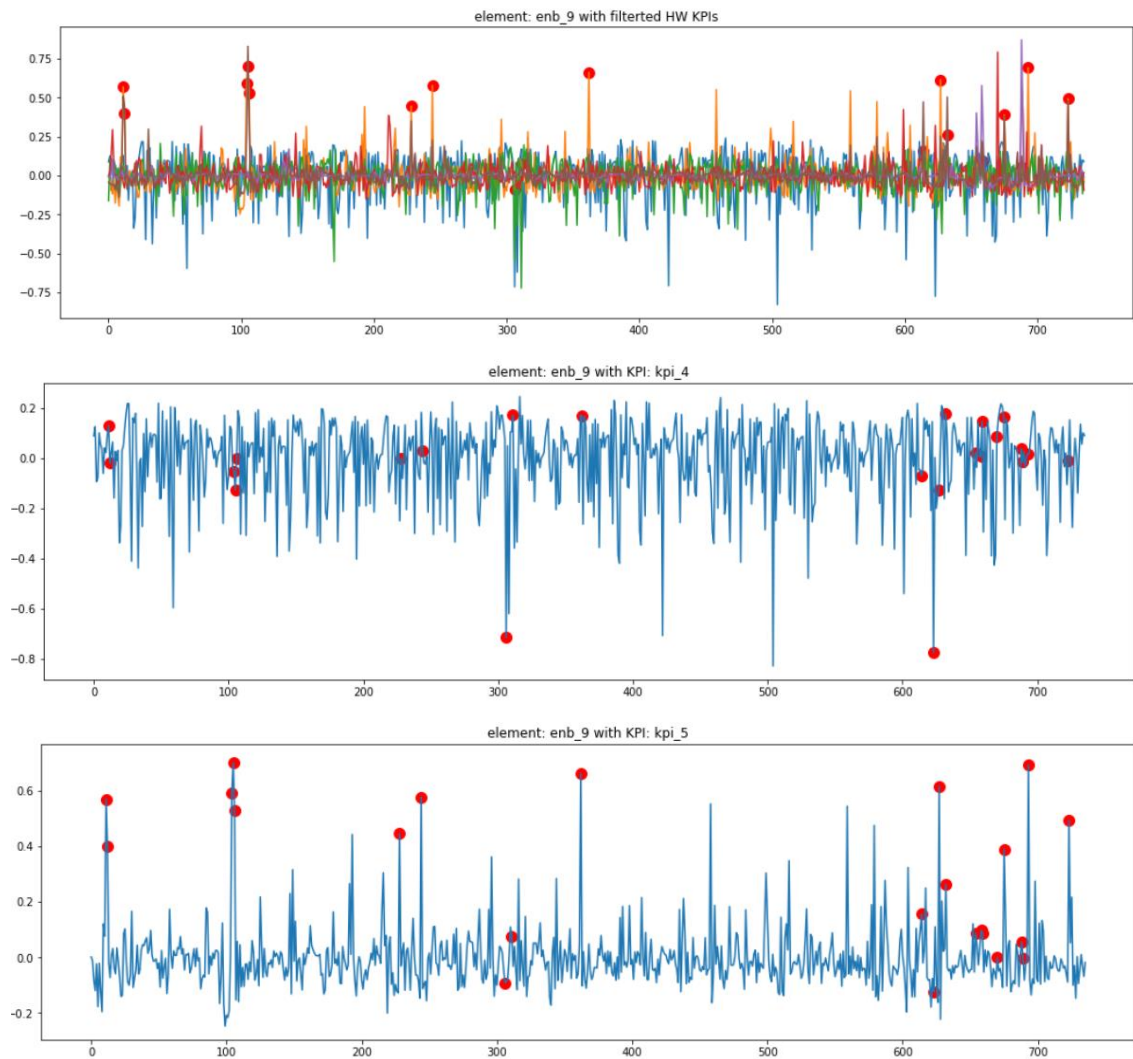
Azure API for anomaly detection was used to discover peaks/dips, level changes and trend points in the eNodeB KPIs. Below are additional graphs that were generated using this API for combination of (eNodeB, KPI):

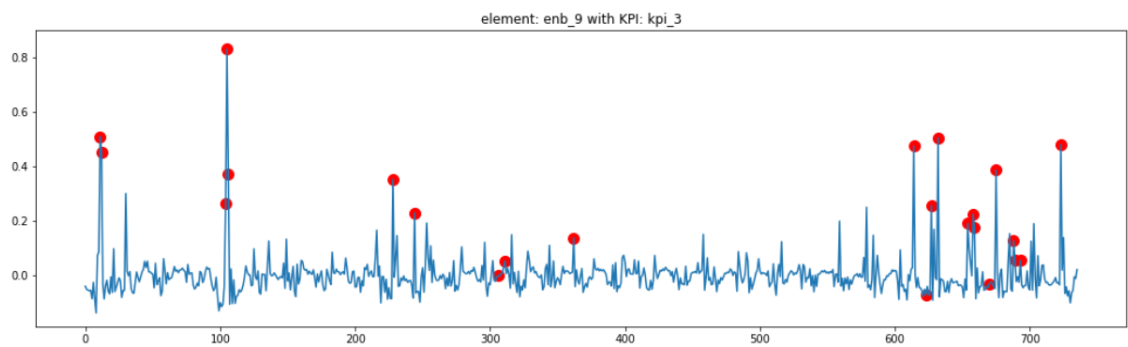
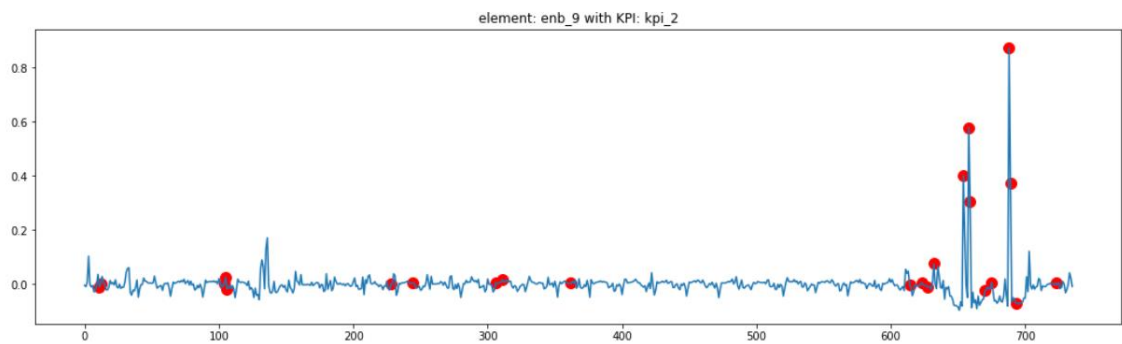
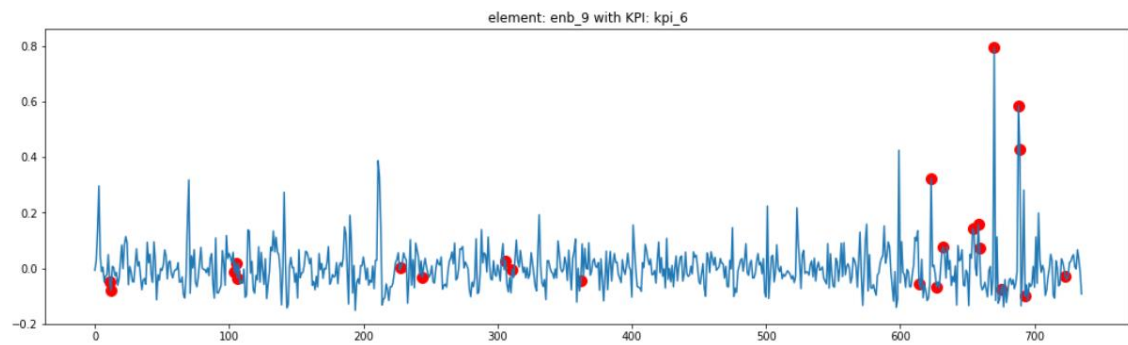
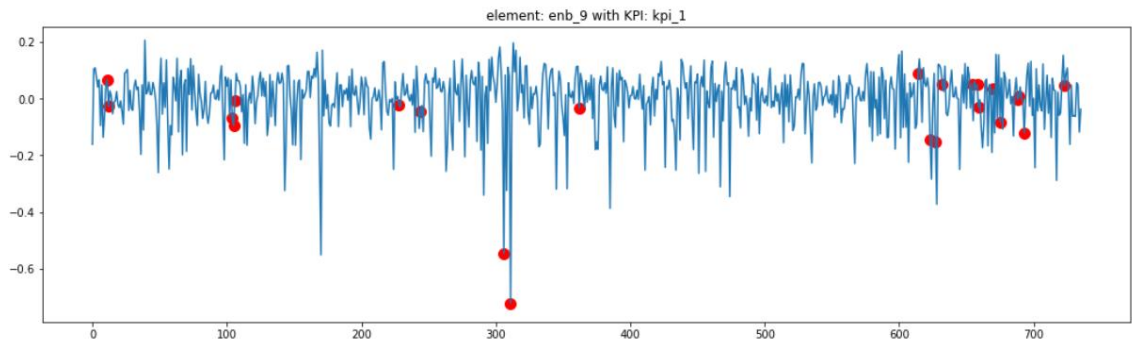




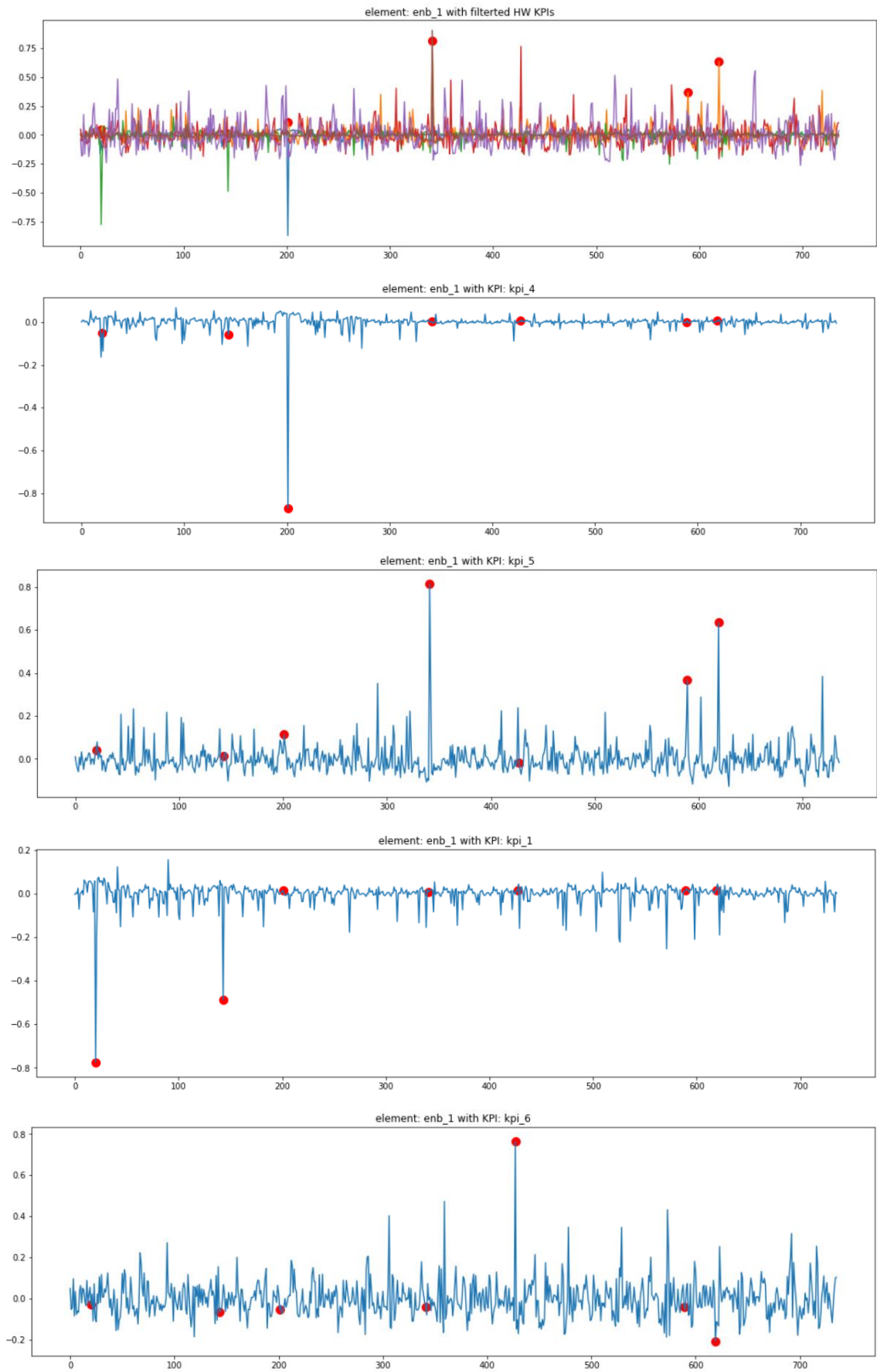
## PDF Anomaly Detection

In this appendix, the KPIs time series for the top (23 data points) and least (7 data points) anomalous eNodeBs are presented.

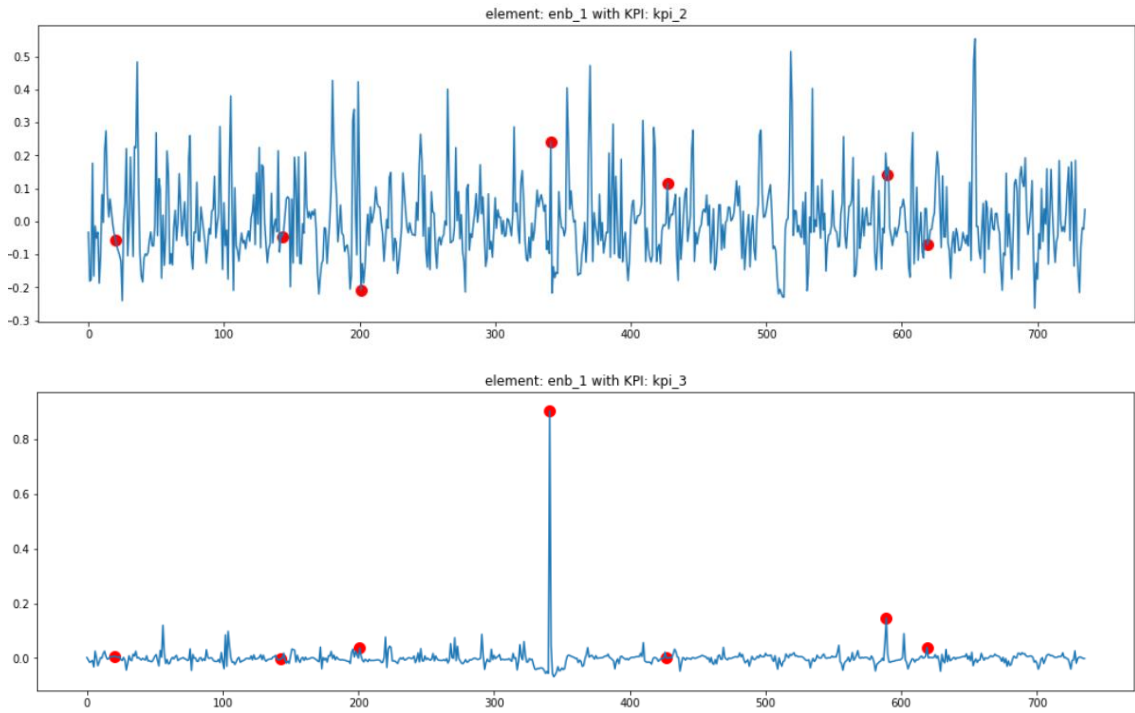




Top anomalous eNodeB KPI graphs using PDF



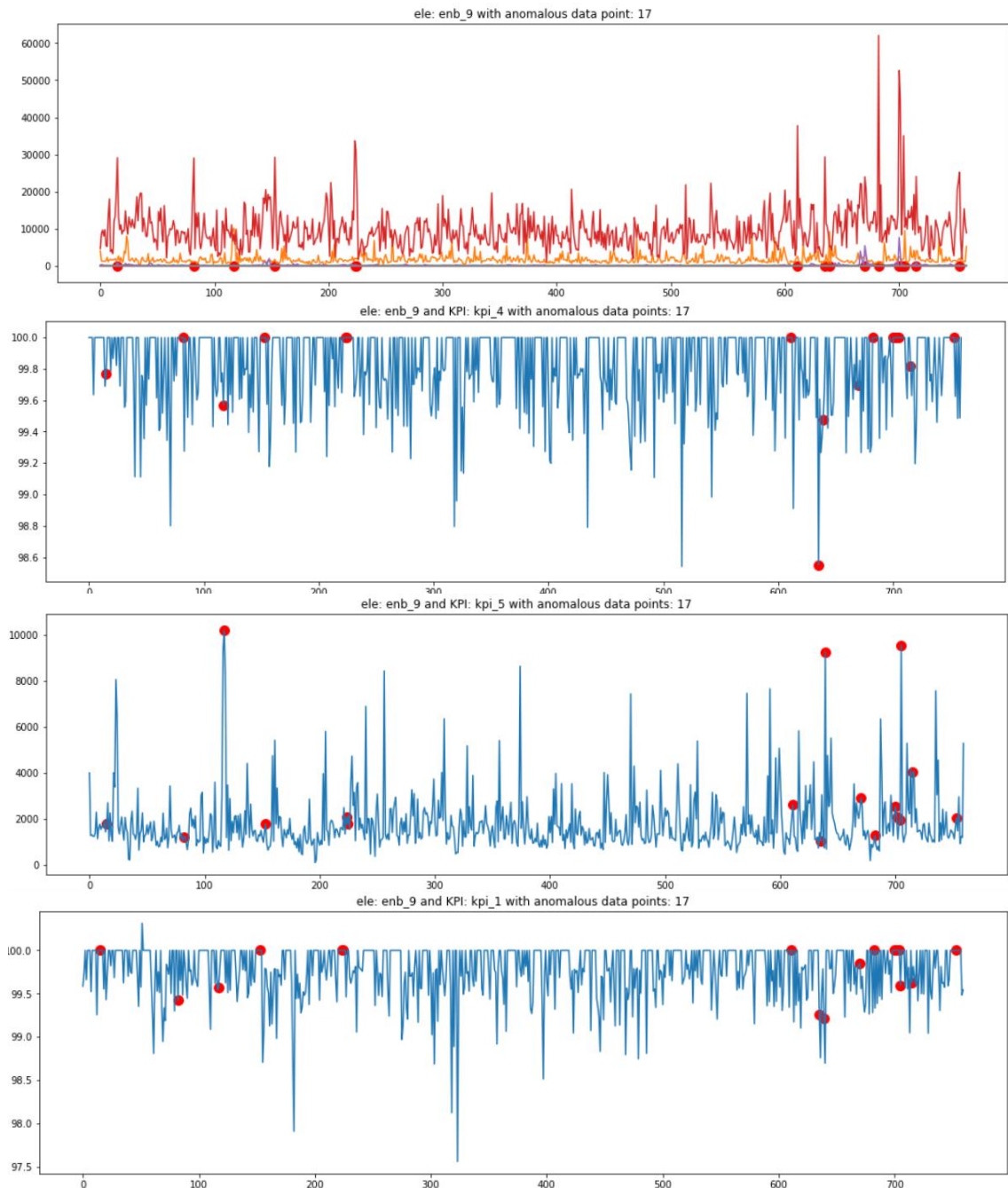


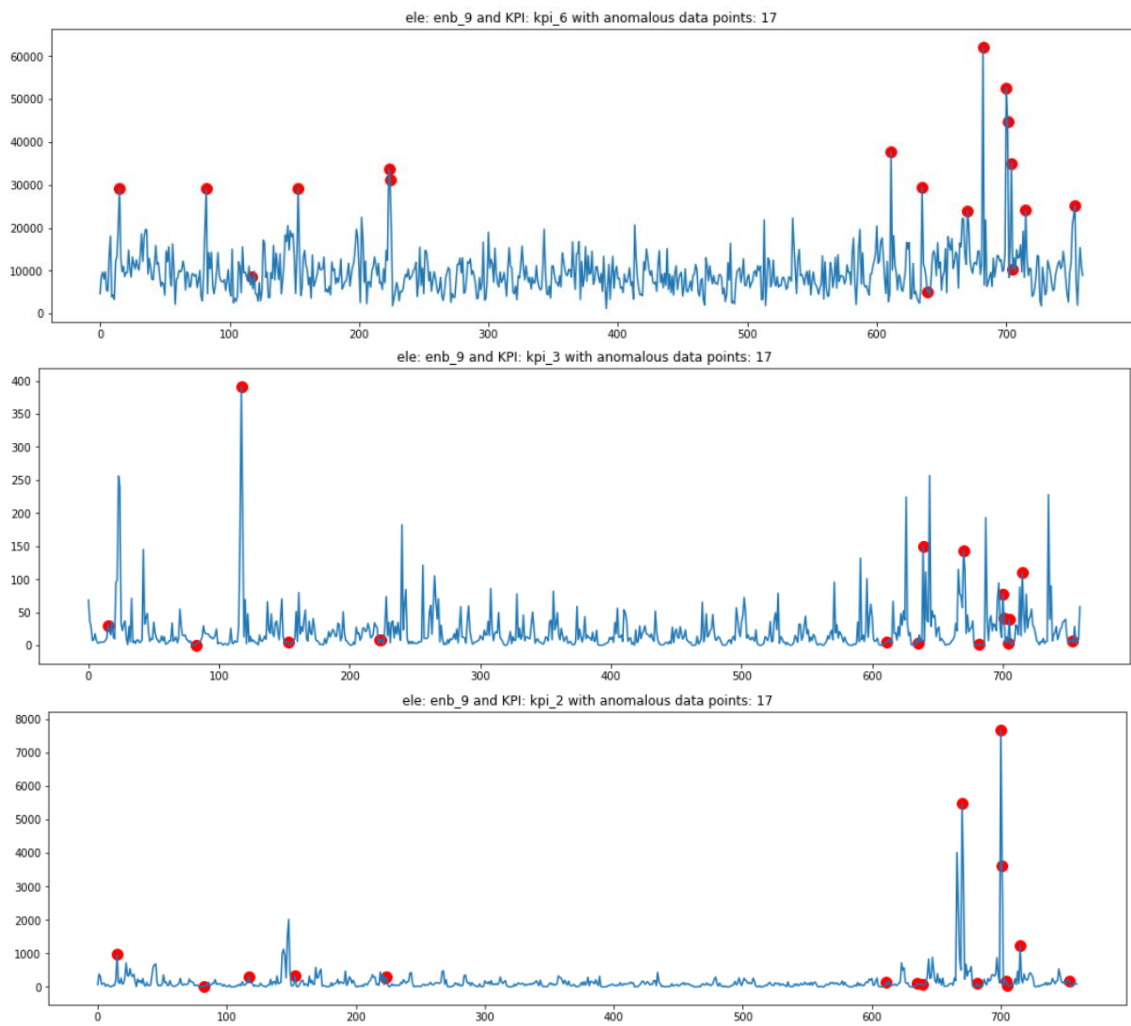


Least anomalous eNodeB KPI graphs using PDF

## Clustering Anomaly Detection

In this appendix, the combined and individual KPI graphs for the top anomalous eNodeB, using the hyper parameters (3, 0.9, 20), are presented.





Top anomalous eNodeB KPI graphs using clustering with parameters (3, 0.9, 20)