UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING & INFORMATICS

LONG TERM EVOLUTION ANOMALY DETECTION AND ROOT CAUSE ANALYSIS FOR DATA THROUGHPUT OPTIMIZATION

By

Simon Mbogo Wanjiru

P52/11311/2018

Supervised By: Dr. Evans Miriti

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF MASTERS OF SCIENCE IN COMPUTATIONAL INTELLIGENCE IN THE SCHOOL OF COMPUTING AND INFORMATICS OF UNIVERSITY OF NAIROBI

July 2020

DECLARATION

This research project is my original work and has not been submitted to any other university for academic award.

Sign…… Date…….

SIMON MBOGO WANJIRU

P52/11311/2018

This project has been submitted with my approval as the appointed University supervisor.

Sign…… Date…….

DR. EVANS MIRITI

ABSTRACT

There is a growing demand for data which is driven by high number of smartphones, applications and traffic demand. Network operators have tried to provide enough capacity and meet the data speeds that the customer needs. This has led to introduction of new technology and expansion of the mobile networks making it complex to manage. Detecting anomalies that affect data throughput/speeds and investigating the root causes in mobile networks is challenging as mobile environments are increasingly complex, heterogeneous, and evolving.

There is need to automate network management activities to improve network management processes and prevent revenue loss. Self-Organizing network is a standard introduced by third Generation Partnership Program (3GPP) to automate network management. However, the standard is still not fully developed.

This project focused on implementing an anomaly detection and root cause analysis model that helps in the process of data throughput optimization in Long-term evolution (LTE) networks. The model used Density Based Spatial Clustering of Applications with Noise (DBSCAN) for anomaly detection, K-Nearest Neighbour (KNN) for root cause analysis and real network performance data from a Kenyan Operator.

Proposed anomaly detection model achieved a silhouette coefficient of 0.451 showing a good separation of existing clusters in the dataset and was able to detect anomalies with both positive and negative impact on data throughput. The root cause analysis model achieved an accuracy of 94.59% and was able to identify the root cause of detected anomalies that had a negative impact on data throughput.

ACKNOWLEDGEMENTS

I would like to express my great appreciation and gratitude to Dr. Evans Miriti my project supervisor, for his patient guidance, valued advice and constructive critiques during the planning and development of this research project. His generosity with his time and expert guidance is highly appreciated. I also extend my thanks to my other lecturers at the School of Computing and Informatics, University of Nairobi for their advice and assistance during my studies.

I would also like to thank Mr. Ronald Moenga, my manager and immediate supervisor at work for his support during my studies and research project. His professionalism in handling my split responsibilities for work and study is deeply appreciated. I would also like to appreciate my work colleagues for their understanding and support during my studies.

Finally, I would like to express my deep gratitude to my wife Pauline Muthoni for her unconditional encouragement and support throughout my studies and her patience and understanding during the busy and unpredictable work-study schedules.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

2G	- Second Generation
3G	- Third Generation
3GPP	- Third Generation Partnership Programme
4G	- Fourth Generation
ANR	- Automatic Neighbour Relations
BER	- Block Error rate
BTS	- Base Transceiver Station
CA	- Carrier Aggregation
CAK	- Communications Authority of Kenya
CSSR	- Call Set up Success Rate
DBSCAN	- Density-Based Spatial Clustering of Application with Noise
DL CQI	- Downlink channel quality index
EAB-SVM	- Ensemble of Ada booster with Support Vector Machine
eNBs	- eNodeBs
ETSI	- European Telecommunications Standards Institute
HSPA+	- High Speed Packet Access Plus
KNN	- K-Nearest Neighbour
KPI	- Key Performance Indicator
LSTM	- Long Short-Term Memory
LTE	- Long Term Evolution
NPS	– Net Promoter Score
OPEX	- Operating Expenditure
PCI	- Physical Cell ID

PRB - Physical Resource Block

QAM -Quadrature Amplitude Modulation

QoE - Quality of Experience

QoS - Quality of Service

RAN - Radio Access Network

RCA - Root Cause Analysis

RF - Radio Frequency

RL - Reinforcement Learning

RNN - Recurrent Neural Network

RSRP - Reference Signal Receive Power

RSRQ - Reference Signal Receive Quality

RxLev - Receive level

RxQual - Receive Quality

SINR - Signal to Interference Noise Ratio

SOM - Self-Organizing Map

SON - Self-Organizing Networks

SQI - Speech Quality Index

SVM - Support Vector Machine

1 CHAPTER ONE: INTRODUCTION

1.1 Background

The Communications Authority of Kenya (CAK) is the agency tasked with licensing of all communications systems and services in Kenya. One of the industries that falls under communications industry is telecommunications. The development and implementation of strategies and policies for the telecommunications services also falls under the mandate of CAK. CAK is also tasked with safeguarding consumers of telecommunications services with regards to the quality and variety of services offered as wells as prices charged for the services. To ensure good quality of service to the consumers, the service providers are required to continually monitor their performance to make sure that they meet their commitment as set out in their licenses and are complying with the provisions of the Kenya Information and Communications Act, 1998 and the Kenya Communications Regulations, 2001.

According to CAK, (2017), there are three main components that constitute the quality of the Information Communication Technology (ICT) service which includes, overall network performance (quality of the network infrastructure), End-to-End (quality of service) and Quality of Experience (QoE). The overall network performance is determined by conducting a performance assessment of the network between two network interfaces referred to as network counters obtained directly from the access and/or the core network.

There are three service types used in network quality assessment done by CAK which are, voice, sms and data/internet. CAK measures quality of service using eight key performance indicators (KPIs) which includes voice and packet call completion rate, set up success rate (CSSR), drop rate, setup time, and block rate, voice call speech quality, handover success rate, and Receive Level (RxLev). A service provider is deemed compliant if they attain an aggregate of 80% or above. In the event of failure by a service provider, penalties and/or other sanctions are applied on an annual basis. Service providers are fined sum equivalent to 0.15% of their gross annual income for non-compliance (Communication Authority of Kenya, 2017).

The measures of quality used by CAK overlook data/internet experience in terms of data speeds, yet this is an important KPI with the ever-increasing data demand. Mobile data demand is growing at a high rate every day. Close to 300 million people started using wireless network internet for in

2018, raising the number people using wireless network in the world to more than 3.5 billion people. This has driven acceleration in network rollout by network operators, e.g. in African coutries in the south of Sahara desert, third generation infrastructure deployment grew by 7% from 2017 to 2018, covering more than 80 million new wireless network consumers, (Bahia & Suardi, 2019). The accelerated network expansion introduces complexities in maintaining network infrastructure quality. One of the key data KPI is data throughput which is the rate of end to end message/packet transfer over a transmission channel. Anomalies in this KPI directly affect a user internet experience.

Detecting anomalies and investigating their root causes in mobile networks is a challenging exercise as mobile environments are increasingly complex, heterogeneous, and evolving. This is an important exercise currently as telecom operators are faced with the need to switch from technical quality requirements to providing good user experience. This trend is driven by the increasing number of network devices, applications and the high demand of mobile network services, (Eirini et al., 2015). A slow data connection can lead to customer churn.

Mobile networks collect high volume of performance data that can be used to monitor and analyze the network performance. Radio Access Network (RAN) on occasion experience anomalous events e.g. capacity constraints, transmission faults, random hardware/software failures, traffic shifts, network availability, handover problems and interference among others that significantly affect network performance as well as customer experience. These anomalous events can be complex to detect and diagnose through manual approach by radio network optimization and field engineers. To maintain a good radio network performance, hundreds of KPIs and alarmas from a high number of base stations ought to be monitored continuously. Once an anomaly is detected, network engineers could spend a significant amount of time and effort carryout root cause analysis (RCA) manually before the problem can be remedied. Manual root cause analysis requires experienced Engineers who are domain experts and can take a long time to diagnose the problem, (Zhang et al., 2018). Use of Machine Learning can help in automating such an exercise.

1.2 Problem Statement

Today's mobile network should provide a reliable and fast network connection. However, the mobile network is complex and many anomalies affecting a network connection are difficult to detect, diagnose and resolve. Some network anomalies like transmission faults, traffic shifts,

capacity constraints and network availability have a direct impact on revenue and customer experience and requires modern detection and root cause analysis approaches.

Digital transformation is happening in many industries and the telecom industry is at the forefront of this transformation. One of the main reasons for digital transformation is process improvement and reducing Internet Technology (IT) complexity. Self-organizing networks (SON) is driving this change in the radio network. SON is a group of use cases for automatic network configuration, network optimization, diagnosis and healing of wireless networks and is defined in third Generation Partnership Program (3GPP) specifications (TS 32.500). Some scholars have contributed to SON functions developments by designing models that automate network management functions. Mismar & Evans (2019) proposed a deep reinforcement learning (RL) model to automate fault management in mobile network. Mismar and Evans model learns how to improve downlink(DL) Signal to Interference Noise Ratio (SINR) by exploring and exploiting various alarm corrective actions. Zhang et al., (2019) proposed self-organizing wireless radio access network (SORA) system enhanced with deep learning. Zhang et al. solution was divided into four core components which included KPI monitoring, anomaly prediction, diagnosis of predicted anomalies and self-healing. In Zhang et al. work, only the first three components were implemented. Self-healing was left for future works. Anomaly detection component in Zhang et al. solution predicts anomaly or normal cell state based on the current state of collected critical KPIs.

Although many use cases are available in SON with significant benefits in the network, the field is still underdeveloped and more advanced use cases are either not yet developed or have incomplete implementations (Grondalen et al., 2012).

1.3 Main objective

The aim of this project is to detect and diagnose the root cause of network anomalies in the radio access network.

1.3.1 Specific Objectives

1. To detect radio network anomalies in a one-month optimization cycle through adoption of a machine learning algorithm.

2. To determine the root cause of network anomalies in a one-month optimization cycle through adoption of a machine learning algorithm.
3. To implement and test machine learning model to detect and diagnose network anomalies.

1.4 Significance

The project contributes towards the development and improvement of optimization processes and adoption of best network optimization practices in radio network. It also contributes towards the development of user-friendly data optimization tools which can be used by network maintenance and optimization Engineers in their network management activities. Designed model focuses on automating data throughput optimization process, an activity that takes a lot of Engineer's quality time, and addresses the problem of long data optimization time/cycle and high optimization cost. Other benefits include proactive network optimization, fast and efficient customer complaint resolution, and improvement on Net Promoter Score (NPS).

2 CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

Despite mobile network operators' race to create better and larger networks to meet the mobile data crunch, the demand for data is growing faster than the carriers can keep up with. As mobile technology improves, customers use more data and the faster the speeds provided by the mobile operators, the more consumers swallow it up and demand more. That is great for carriers because data usage is monetized, but it presents major challenges in network maintenance and optimization. The additional revenue is great, but if speeds degrades, customers look for other service providers that can provide a more reliable connection (Mobolize, 2014).

According to European Telecommunications Standards Institute (ETSI), capacity limitations of the second generation (2G) network due to high demand for multimedia services (voice plus data) and demand for higher data rates triggered the introduction of third generation (3G) network. Theoretical maximum speed for 2G network is 384kbps. The first release of 3G network standard had a top speed of 2Mbps but enhancements were done later (later 3G releases) to improve the data speeds to 42 Mbps through combining two carriers together with High speed Packet Access (HSPA+)). The increasing demand for data later led to introduction of fourth generation (4G)/Long Term Evolution (LTE) network which is more spectrum efficient as compared to 3G technology and provides higher data speed (throughput) of up to 1Gbps (theoretical). Bandwidth scalability in LTE enables operators to use multiple channels to achieve higher peak data rates, (Mohana *et al*., 2014). Additional features help in improving data throughput: with 2x2 multiple input multiple output (MIMO), 256 Quadrature Amplitude Modulation (QAM) and 4x20 MHz Carrier Aggregation (CA) throughput of up to 800 Mbps is possible. Carrier aggregation (CA) is a technology where multiple carriers across the available spectrum are combined to create a wider bandwidth channel for increased network data speeds and more network capacity, (Kiwoli, 2017).

2.2 Radio access network optimization

Radio access network (RAN) optimization is a set of activities that are required in an existing mobile network to improve or maintain network performance. These activities are carried out to make the system work more effectively by preventing and resolving network anomalies while at the same time avoiding unnecessary investments in costly infrastructure. RAN Optimization

consists of several activities which includes performance monitoring, drive testing and Radio Frequency (RF) optimization. Network performance monitoring is a day to day network activity to assess, analyze and track wireless network performance. Network operators use equipment vendor counters and defined KPIs to gauge and maintain performance of the RAN.

Drive testing is a means of assessing and measuring the capacity, coverage and Quality of Service (QoS) of a wireless network. It involves using a vehicle equipped with a radio network drive test equipment that detect and logs various physical and logical parameters of wireless network in a given network geographical area. This measures the real network user experience of would experience in a specific area, which helps the operators to make directed network changes to improve network coverage and quality for better user experience.

Radio Frequency (RF) Optimization is a network activity through which different base station parameters are adjusted to improve the coverage and quality of signal in an area. Many radio frequency parameters like Reference Signal Receive Power (RSRP), Reference Signal Receive Quality (RSRQ), Receive Level (RxLev), Receive Quality (RxQual), packet drop rate, mean opinion score, signal to noise ratio, throughput etc. are constantly monitored and necessary changes proposed to keep these parameters within agreed targets set by the mobile operator (Usuah, 2017).

Mobile operators use data collected from network monitoring systems (NMSs), drive tests, customer feedback/complaints, network probes and other systems to address anomalies reported by the users or detected while monitoring the network. Collected statistics and other network data is used to detect and analyze the root cause of anomalous events like dropped calls or slow internet speeds. Most of optimization activities are done manually although tools are being introduced to automate these activities. SON is a standard introduced for automation of network management activities.

2.2.1 Anomaly detection in the radio access network

Anomaly detection is the method of identifying observations that do not follow an anticipated pattern within a given data set (Adrian and Niclas, 2017). Alvarez (2018) identifies three categories of anomalies as follows;

a) **Point anomaly:** an isolated data occurrence that is considered as anomalous when compared to values of the rest of the data.

b) **Contextual anomaly**: an isolated data occurrence that is considered as abnormal only in a specific situation, but not otherwise.
c) **Collective anomaly**: is a group of data occurrences considered as abnormal when compared to the entire dataset. In collective anomaly, the isolated data occurrences are anomalous if and only if they occur together.

2.2.2 Root cause analysis

The health of a wireless network can be diagnosed based on a number of observed key performance indicators in the same way that a doctor diagnoses a patient based on the symptoms he presents with. During performance monitoring, network anomalies are detected and investigated to get the root cause. The root cause analysis is conducted by experts who use their knowledge of the correlations between key performance indicators and the status of the network. Indicators used in network anomalies diagnosis includes performance counters, network nodes alarms, key performance indicators, traces and customer complaints (Palacios et al, 2016).

2.2.3 Network optimization methods

Manual method

Operators set target network KPIs to maintain a good radio network performance. Network Engineers continuously monitor a high number of KPIs from thousands of base stations to detect anomalies and identify inefficient use of resources. Engineers rely on equipment vendor NMSs and spreadsheets to gauge and maintain performance of the network. Once an anomalous KPI is detected, network engineers perform root cause analysis after which the problem can be remedied.

Automated method

Wireless networks are getting more complex with heterogeneity and this makes network management expensive, time consuming, labor intensive and error prone if done manually. According to Third Generation Partnership Project (3GPP) TS 32.500 version 11.1.0 Release 11, SON was developed to reduce the operating expenses (OPEX) linked to the management of the high number of network elements in wireless network. The network infrustructure can be from a single or mixed vendor. Automation of some planning, optimization and configuration activities

by using SON use cases can reduce network operator operating expenses by reducing costly manual network management activities.

Self-Organizing Networks (SON)

According to 3GPP, SON use cases can be divided into three main groups: Self-Configuration, Self-Optimization and Self-Healing. SON has three architectures which includes centralized, distributed and a mix solution.

i. Self-configuration

This is an automated configuration of newly integrated base station whose power, transmission and physical site parameters like physical cell id are configured automatically leading to quick site planning and integration.

ii. Self-optimisation

Self-optimization functions are encompassed in 3GPP Release 9 and it includes the optimization of network capacity, cell coverage, mobility and management of unwanted, interfering radio signals. Coverage and Capacity Optimization makes it possible to automatically correct capacity constraints depending on slowly changing network environment, such as recurrent changes.

iii. Self-healing

The features for the automatic anomaly detection, resolution of network faults and automated parameter canges are specified in 3GPP Release 10.

Below are some examples of SON functions.

- Coverage and capacity optimization.
- Physical cell identifier (PCI) planning and configuration.
- Automatic neighbor relations (ANR).
- Mobility robustness optimization (MRO).
- Energy saving management.

2.3 Clustering algorithms

Clustering is used to deal with the data structure spliting in unfamiliar area. Clustering algorithms are constructed based on distance (dissimilarity) and similarity. When dealing with

quantitative data structures, distance is the best measure of resemblance among data. On the other hand, similarity is best measure when dealing with qualitative data structures. Clustering algorithms are divided into several categories which includes, clustering based on partitioning, hierarchy, density, distribution among others (Xu and Tian, 2015).

2.3.1 Density based clustering

The concept behind this type of clustering algorithms is that a contigous region of high point density in a data space is considered to as a cluster and data in that region is in that cluster, (Xu and Tian, 2015). This approach is based on density and connectivity functions. Typical algorithms include: Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Density-based Clustering (DenClue) and Ordering Points to Identify The Clustering Structure (OPTICS).

Density-Based Spatial Clustering of Applications with Noise

DBSCAN is a density-based clustering algorithm that works through checking for areas of high density and assigning them to clusters. Points in less dense regions are not included in the clusters and are considered as noise. DBSCAN algorithm needs two parameters:

- Epsilon(eps): this indicates the maximum distance between any two points for them to be considered to be in the same cluster. This therefore indicates that that if the separation between two points is equal to or lower than the given value (eps), then the points are considered neighbors. If epsilon is too small, many points will not be clustered and will be considered as outliers while if epsilon is too large, the majority of the points will fall in one cluster.
- Minpoints: this refers to the least number of points needed to form a dense region e.g. if the value of this parameter is set to 4, then a minimum of 4 points are required to form a cluster. Typically, minimum points are obtained from the number of dimensions (D) in the dataset, as minpoints $\geq D + 1$ with bigger values considered better for datasets with undesirable data as it forms more notable clusters. The minimum acceptable value for the minimum point parameter is three, but with a higher number of points in the dataset, its advisable to have a larger value of minpoints parameter.

The values of the two DBSCAN parameters depend on the kind of data and the expectation of the model. The parameters can be provided by an expert who is very familiar with the data set to be clustered. When DBSCAN is fitted with data, it labels data points as core, border or noise points. The core points have at least minimum number of points within the epsilon distance. On the other hand, border points those points in a cluster that are not dense enough to be core points. For this reason, they can end up in different clusters on different runs. Points that are not labelled as core or border points are labelled as noise (Celik et al., 2011).

DBSCAN is attractive in that it is efficient and is robust to the presence of noise within data. It also requires few parameters. It however doesn't work well with high dimension spaces.

2.4 Classification algorithms

Classification algorithms use training dataset to come up with boundary conditions that can be applied to a dataset to discover target classes. A classification algorithm (classifier) is used to map input data to a specific categories. Common types of classification algorithms include: K-Nearest neighbor, linear classifiers, quadratic classifiers, support vector machines, learning vector quantization, decision trees, and neural networks. K-Nearest neighbor was used in this project.

2.4.1 K-Nearest Neighbor (KNN)

KNN is a classification algorithm that classifys new cases based on their correlation to other nearby cases or data points (Theobald, 2017). KNN is a simple to implement and interprete machine learning algorithm that can be employed to classification and regression problems. The algorithm assumes that cases or data points with similar properties in different datasets will exist close to each other (Kostas, 2018). KNN is therefore used to identify the class of new cases or data points using labelled training data through identifying the k-nearest neighbors of unlabelled new cases or data points.

The most common distance measures for KNN are Euclidean and Manhattan both of which measure the distance between observation xa and xb for all j features. These are determined as follows

$$\sqrt{\sum_{j=1}^{P}(x_{aj} - x_{bj})^2}$$

Euclidean formula

$$\sum_{j=1}^{P}|x_{aj} - x_{bj}|$$

Manhattan formula

The main advantage of KNN is its ability to provide good performance over high dimension data and has a pretty fast training phase. It is however quite slow in the testing phase.

2.5 Related work

Machine learning has been acknowledged as a great tool to drive digital transformation in mobile network and introduces intelligence in the network to achieve automated healing in case of a network anomaly. Never the less, some major challenges have been identified with regard to its practical applications for self-healing. These challenges include; data insufficiency, data imbalance, non-real-time response, cost insensitivity, and merging data from different sources, (Zhang et al., 2019). A number of machine learning algorithms and approaches have previously been used in anomaly detection and root cause analysis.

Alvarez (2018) used random forest to detect LTE resource consumption anomalies and neural network for root cause analysis. He used random forest technique known as Gini importance or Mean decrease impurity to rank performance counters used to predict resource consumption and Quantile Regression Forest which is a modification of random forest to predict the resource consumption in LTE. Predicted resource consumption is compared with real consumption and based on the relationship, its determined whether there is an anomaly in the data. For root cause analysis, Alvarez used auto encoder, a special type of feed forward neural network which uses unsupervised learning. Auto encoder here is used to identify performance monitoring (PM) counters potentially correlated to anomalies and thus the root cause of the anomaly. The approach used by Alvarez assumes availability of labelled data for detection phase. Unfortunately, this is not always the case since experts focus on resolving network faults and are not concerned in root cause of the problem or recording how a fault was detected.

Andrades et al. (2016) proposed a model based on self-organizing map (SOM) to automatically identify LTE faults. The system has two phases which are design and exploitation phases. At design stage, the system automatically identifies several clusters that represents different faults. To obtain good clusters and label them correctly, an method based on the behavior of analyzed performance measurements of each cluster was used. Base station cells were clustered based on the correlation between cell KPIs and fault causes, and performance evaluation was done using silhouette coefficient. The clusters obtained had to be labeled by an expert with the identified causes. This is the first phase which is the diagnosis system. In the second phase, exploitation phase, SOM was used to associate a fault to its cause by classifying a base station's state according to the manifestation of its symptoms. Unlike Alvarez, Andrades et al uses an unsupervised learning which is not limited to labelled data and can detect new anomalies.

Josefsson (2017) also used SOM for both anomaly detection and root cause analysis for video-on-demand (VoD) cloud service. Unlike Andrades et al. (2016) he used both supervised and unsupervised versions of Self Organizing Maps (SOM) and compared them to ascertain which performs better than the other. The supervised SOM implementation used was originally proposed by Ron Wehrens while the unsupervised version was originally proposed by Dean. The key difference between the two is the number of maps used. Supervised version uses 2 maps while the unsupervised version uses 1 map. He used an experimental setup to generate data by creating a testbed that mimic a video-on-demand cloud service. The testbed was designed such that it allowed disperate input similuations and fault introduction into the system to simulate different system utilization usage scenarios. He injected three types of faults (memory hog, Central Processing Unit (CPU) hog and disk hog) and recorded fourteen features for root cause analysis. This resulted to a labelled dataset which was used to train both versions of SOM models. For the fault localization, he used dissimilarity measure to identify the root cause. He compared the features of normal and anomalous samples and the features with the largest difference were selected as the likely cause of an anomaly. Supervised learning approach make use of available labelled data thus performing better than unsupervised learning approach like the one used by Andrades et al (2016).

Time series analysis have also been used for anomaly detection. Eamrurksiri (2017) used Markov switching autoregressive model to detect anomalies in CPU utilization and any change

that affect the CPU utilization in the test environment. Structural changes identified by Markov switching model and used for detecting anomalies are assumed to follow unobserved Markov chain in the time series data used. In addition to time series analysis, he also proposed another approach that uses a hierarchical clustering algorithm to identify various change point locations in time series data. The approach is called E-divisive method and is based on a non-parametric analysis. Evaluation for this model was done through use of simulated dataset since the data used in the analysis contains no ground truth. The simulated dataset was also used to compare Markov switching autoregressive model with the E-divisive model. From the evaluation done, Markov switching autoregressive model performed better than E-divisive model in detecting changes.

While still using time series data from a cellular network, Mamuna & Juhavalimakia (2018) used KNN to classify test data. They carried out anomaly detection in two phases. First phase involved use of one-class support vector machine (SVM) to predict anomalies in key performance values that were range based. In the second phase, a model that used Long Short-Term Memor (LSTM) architecture of deep learning Recurrent Neural Network (RNN) was then applied to predict key performance indicators values that were profile based, and thus indirectly predicting profile based anomalies. The intersection between the profile based and range based predicted anomalies produced the final list of anomalies.

Scholars have also used deep learning model in anomaly detection. Mismar & Evans (2019) used deep reinforcement learning to automate fault management in mobile network. Proposed model learns how to improve downlink SINR by exploring and exploiting various alarm corrective actions. The model resides in the base station which is a good idea since a fault is localized to a base station, but the model may not detect anomalies beyond the base station e.g. core network or transmission faults at aggregation points.

Zhang et al. (2019) proposed a self-organizing wirelss radio access network model strengthened by use of deep learning. Proposed solution was divided into four core components which included KPI monitoring, detection of anomalies, identifying the root cause of those anomalies and automated healing. Only the first three components were implemented while Self-healing was left for future works. The approach for KPI monitoring component was to divide cell KPIs into several sets that shows the system status from the following angle; retainability, network access, availability, integrity, mobility and base station speeds, which are then stored in a

database. For anomaly detection, the system predicts anomaly or normal cell state by analyzing the current state of fetched critical key performance indicators. Critical KPIs are identified with the help of an expert while prediction is based on CNN+convLSTM algorithms. After predicting an anomaly, a marked point is passed to the root cause analysis component which uses both supervised and unsupervised learning. Supervised classification made use of auto-encoder and decision tree while unsupervised clustering made use of auto-encoder and agglomerative clustering. Self/automated healing using reinforcement learning is the last module to change specific parameters to clear identified anomaly.

Ensemble learning is widely used in detecting anomalies and identifying their root causes. Palacios et al. (2016) proposed an Ensemble of AdaBooster with support vector machine based component classifier technique to detect the network intrusions and monitor the activities of the network node and classify it as either normal or anomalous. Wang et al. (2019) was also able to use ensemble learning to automatically diagnos network failures in a heterogenious network. The final diagnosis was obtained by combining all base classifications to improve performance. The fault-diagnosis system had two stages: a design stage and an exploitation stage. Data preprocessing was done in the design stage where data imbalance problem was resolved by using over-sampling technique. The resulting data was then reorganised into subgroups of two classes to which a weight vector was added. The subgroups were then used to train the base classifiers which were used to classify each input case and final result achieved through a union and majority vote in the exploitation stage. Ensemble learning approach produces a model that improves quality of predictions through stacking, that reduces bias through boosting and improves stability and accuracy through bagging.

2.6 Summary of gaps

Of all the SON functionalities, Self-healing (anomaly detection, diagnosis and automatic adjustment) is the most challenging especially diagnosis, and therefore is the least developed. There are no commercial systems that perform automatic diagnosis with enough reliability to convince network operators (Khatib, 2017). Some of the major challenges with regard to the real-world applications of machine learning systems for self or automated healing were highlighted by Zhang et al. (2019) and includes; data imbalance, cost insensitivity, data insufficiency multi-source data fusion, , and non-real-time response.

The choice of learning algorithm to be used for anomaly detection or diagnosis introduces some limitations in the model. Supervised and semi supervised learning methods used by Khatib (2017) and Alvarez (2018) for anomaly detection limits the system to detecting previously known anomalies only. The approach also faces the problem of insufficient amount of labelled data since experts focus on resolving network anomalies and are not concerned in root cause of the problem or recording how a fault was detected.

Once a model is created, where it resides can also limit its performance. Mismar & Evans (2019) model resides in the base station and may not detect anomalies beyond the base station e.g. core network or transmission faults at aggregation points. Majority of proposed models (Mismar & Evans (2019), Alvarez (2018), Eamrurksiri (2017), and Khatib (2017)) focuses on use of performance data for anomaly detection and only a few models use alarms or configuration data. Some authors like Mdini et al. (2018) and Ciocarlie et al. (n.d), were not able to connect the component to diagnose the root cause of anomalies to the anomaly detection component and plan to do so in their future works. To deal with this challenge, Wang et al. (2019) and Palacios et al. (2016) successfully used ensemble learning approach which also improved model performance.

2.7 Proposed solution

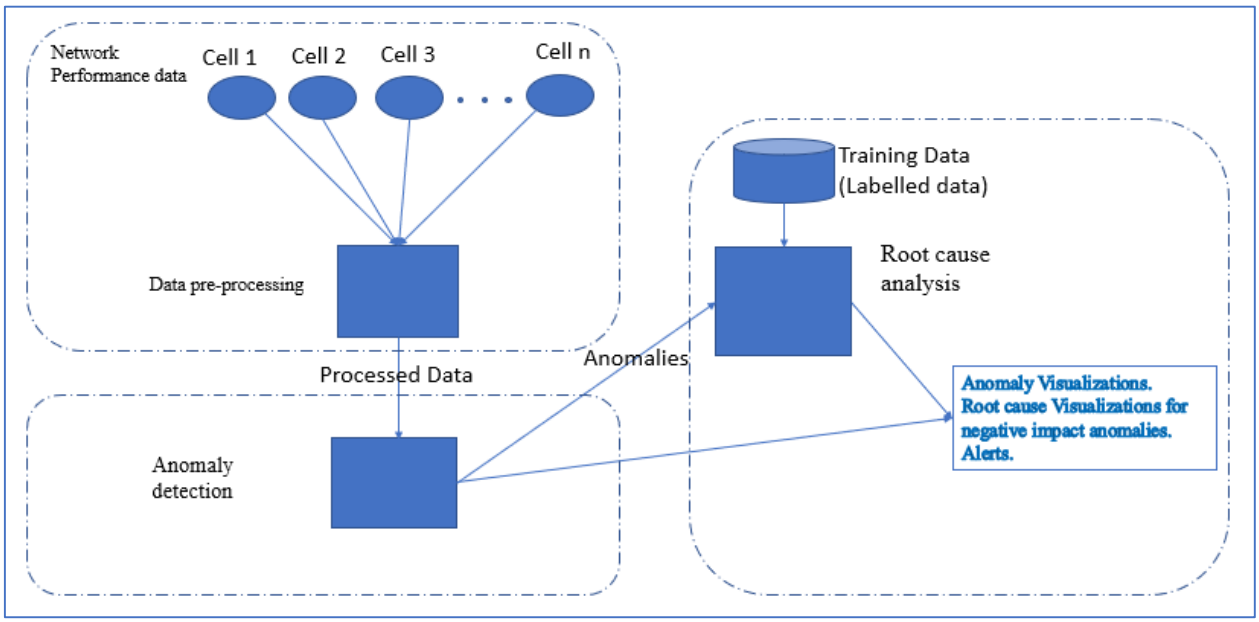


Figure 1: Anomaly detection and root cause analysis process.

The proposed model has three modules. In the first module, time series network performance data for cells in chosen network cluster was pre-processed and passed to anomaly detection module. The first task in the data processing was to remove missing and invalid data. Data was then anonymized to replace the real name of a cell with an auto-generated cell name. The last task in this module was to filter out the key metric, average user throughput, and group it to 28 dimensions/features used in anomaly detection.

The second module used DBSCAN algorithm to detect anomalies. This clustering algorithm was used because it is robust to noisy data. The two parameters used in DBSCAN were generated from the input data. This was through preparation of histograms using minimum distances between samples and number of neighbors per cell. The output of this module was clusters and anomalies. Clustering done in this module helped in improving the sensitivity of the module to anomalies and reducing false anomaly detection. Once anomalies were detected, root cause analysis module was activated. A visualization was also generated for all the cells showing the different clusters and trends for normal and anomalous cells.

In the last module, root cause analysis was done for any cell picked with anomaly. The module took advantage of labelled performance data to train the model. K-nearest neighbor algorithm was used in this phase. KNN was chosen as it provides good performance over multidimensional data and it's simple and fast algorithm during the training phase. The only parameter in this module was determined using cross-validation technique. The output of this module was root cause for each anomaly which can be used to generate network optimization/anomaly correction recommendation. Visualizations were generated for those anomalies that affect data throughput negatively.

3 CHAPTER THREE: RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

Research methodology refers to the specific approach used to identify, choose, process, and analyze information on a specified topic. It involves scientific/systematic process where every single part of the process is based on thought reasoning, Dadhe (2016). This section deals with how the data was gathered and analyzed. This project focused on detection of anomalies that affect LTE throughput and root cause analysis of those anomalies using machine learning algorithms.

3.2 Research design

Research design is a general plan of the research study to be done, highlighting a overall statement of the methods to be employed (Dadhe, 2016).

3.2.1 Quantitative research design

Quantitative approach involves generating data in quantitative form that can go through a thorough quantitative analysis (Dadhe, 2016). Quantitative methods place emphasis on unbiassed measurements and the analysis of the collected data. This project focused on creation and testing of a model in an experimental setup thus the choice of quantitative research design. Data used in this project was queried from historical mobile network performance data from a Kenyan operator.

Design steps

1. Data Collection
2. Pre-processing the data
3. Training the model
4. Performance evaluation and reporting the performance.

3.2.2 Data collection

Secondary data was used in this project. Long term evolution (LTE) historical performance data was extracted from one Kenyan Operator database after seeking necessary approvals from the operator. Imanager performance reporting system (PRS) was used to extract data from performance database servers and labelled throughput anomalies root cause data was extracted from network optimization team shared folder.

3.2.3 Data pre-processing

Data preparation is a key exercise in machine learning. It helps in organizing data in a form that is ready for processing by machine learning models and prepares features that lead to best model performance. Python programming language was used in data pre-processing and model design. Anaconda distribution was used as a Python data science platform in this project. The distribution is available for free and provides access to a great variety of Python packages that facilitate data science tasks. Below are some of the ones that were used in this project.

- **NumPy** – is a library used for processing arrays in Python. It adds high performance objects and tools for processing arrays.
- **Pandas –** is also a Python library built on top of NumPy that provides data structures and operations for processing numerical tables and time series data.
- **Scikit-learn –** is a machine learning library for Python built on top of SciPy.

Missing data

Base transceiver station (BTS) cell samples in the data with missing values, invalid entries like NA and negative numeric value were dropped. This operation was performed using python pandas library on each data set after which a new table with clean data was created.

Data anonymization

Toom (2018) defines anonymization as the operation of changing data into a form that does not identify individuals. It is usually done to protect the identity of research subjects or individuals and is critical to ethical research. In this project, anonymization was applied to direct identifiers of the chosen operator's cell names. The cell names of collected data was divided into two main parts, site name part and cell id part separated by a hyphen. The site name part was replaced with the word "site" which was followed by an autoincrementing integer (X) for each unique original site name. The hyphen was replaced with an underscore after which the word "cell" plus an underscore was inserted between the underscore and the cell id (Y). The resulting anonymized cell name had the format "siteX_Cell_Y".

Feature selection

Feature selection helps in reducing the features to a subset that can have the same predictive performance like performance when the full set of features are used. It reduces complexity in the

model and make it easier to understand. In this project, the features were reduced to 6 features through use of domain knowledge. These features are: throughput, data traffic, packet retransmission, downlink channel quality index (DL CQI), physical radio bearer (PRB) Utilization and Transmission radio network layer KPI. Throughput is the key metric for anomaly detection while the others are the main features that affect throughput in LTE.

Data normalization

Data normalization was done to avoid issues related to training models with features that have wildly varying ranges. Normalization was performed on data set used in root cause analysis. Since the five features takes different ranges, there is need to normalize the data to improve the training process. Min-max technique was used to normalize the data to the range [0,1]. The following formula was applied to each feature value in the dataset:

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

Where x is x_1, x_2, x_n and z_i is the ith normalized data.

3.2.4 Model training

Before the process of identifying the root cause of throughput degradation in the network, the system had to detect that an actual anomaly was present. DBSCAN was used for anomaly detection and once an anomaly was detected, KNN was used for root cause analysis.

Table 1: Sample DBSCAN modelling data

Date	Cell_Name	4G_PS_traffic_Volume(GB)	AvgUserThroughput(Mbps)(Mbps)	DL_PRB_Usage_Rate(%)	L.E-RAB.Fail Est.TNL	L.Traffic.DL.AM.Retrans.TxPackets(packet)	DL_CQI
6/14/2020	Site1_Cell_0	86.7077	11.3413	33.4541	88	1735110	9.027
6/21/2020	Site1_Cell_0	73.8486	8.1519	32.4937	52	1818899	8.9092
6/4/2020	Site1_Cell_0	69.5632	11.5424	29.3117	0	1609008	9.0438
6/5/2020	Site1_Cell_0	72.6653	10.2321	30.439	0	1615526	9.1097
6/21/2020	Site1_Cell_1	88.0132	1.8757	48.0953	161	2496300	7.7719
6/14/2020	Site1_Cell_1	83.8891	6.6517	42.5041	65	1957277	7.911
6/21/2020	Site1_Cell_2	40.7067	20.9512	11.7075	37	645536	10.2923
6/14/2020	Site1_Cell_2	46.2813	21.1434	13.5065	36	577089	10.4569

Table 1 above shows sample dataset used in anomaly detection model training in the system.

DBSCAN model training

Four weeks network cluster data, 5426 rows of data, with throughput as the only metric was organized in such a way that each cell formed a data point while each day's average throughput formed a feature/dimension. With N number of cells in a network cluster (i.e. N time series) and four weeks daily average data, there was an N x 28 data array. There were therefore N points that were clustered. This data was fitted to the model for training. The two DBSCAN parameters (epsilon and minpoints) depend on the kind of data and the expectation of the model. To determine epsilon, minimum separation between each data point to all other data points was calculated and the resulting values used to generate a histogram. The minimum distance under which majority of the data points lie was chosen as the Epsilon. Once the epsilon was determined, it was used to calculate the number of neighbors for each data point which was then used to generate another histogram. The frequency of neighbors in the histogram was then used to determine the minpoints or the value under which the neighbors were considered to be few.

Clustering steps

The clustering steps used in this project were adapted from Babul et al., (2015) whose work focused on analyzing DBSCAN clustering method on disparate datasets using a tool called weka.

The algorithm starts by selecting a random base station cell and retrieves its neighboring cells information using the epsilon parameter.

1. If selected base station cell has minimum neighboring cells within epsilon neighborhood, cluster formation starts, else the point is labelled as noise. The cell can however be found later within the epsilon neighborhood of another base station cell and made part of that cluster.
2. If the cell is found to be a core point, a cluster is formed, and all other core point cells within the epsilon neighborhood are added together with their epsilon neighborhood.
3. The above process ends once a density-connected cluster is found.
4. The whole process is repeated with a new base station cell which can be a part of a new cluster or labelled as noise once all the cells are visited.

Cells labelled as noise were considered anomalous.

KNN model training

Detected anomalous cells were passed to the last module for root cause analysis. Labelled data was acquired from a Kenyan operator and since the amount of normal data was much more than abnormal data which generates imbalanced data, the data was randomized, and under-sampling used to reduce data points to 7066 which were used as the training data for the model. The features used in KNN model training were data traffic, packet retransmission, DL CQI, PRB Utilization and Transmission radio network layer KPIs. Training data was extracted from labelled performance data and divided into training (80%) and testing set (20%). The value of k was determined through use of cross validation. KNN algorithm was then applied to the training dataset and validated using the test dataset. A model was created using different values of k (odd numbers from 3 to 13), prediction done for the test data and accuracy checked. The value of k that gave the maximum accuracy was chosen. Euclidean distance formula was used as the distance measure.

The new data for this module was anomalous cell performance data with the same features as those used in KNN model training.

Classification steps using KNN

1. Load the labelled training data.

2. Choose value of k as indicated above using cross validation.
3. For each point in the data, calculate its distance from each row of training data using Euclidean distance formula. Store the index of each row and the distance in an ordered collection.
4. Sort them based on the distance value from the smallest to the largest. Choose the top k rows from the sorted collection
5. Assign a label of the selected k rows based on the most frequent label of these rows

3.2.5 Performance evaluation and reporting the performance

There are two types of performance evaluation based on information available about the dataset to be used and these are external and internal evaluation. External evaluation is used when previous information about the data set is available. Examples of external evaluation performance measure are; F-measure, accuracy and normalized mutual information. Internal performance measures like silhouette index, partition coefficient, Dunn index and separation index makes use of the dataset itself for performance evaluation. In this project, Silhouette evaluation method was used for clustering phase and since previous information about the data set was available, Accuracy was used for the root cause analysis phase.

Silhouette coefficient

The silhouette index is a measure of how similar an object is to its own cluster compared to other clusters. It makes use of clustering properties, cohesion and separation to assess the effect of clustering. Below formulae is used to calculate silhouette coefficient of a point i.

$$S(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Where: *a (i)* describes cohesion in a cluster as it is the mean distance between data point i and other data points within the same sub-cluster; *b(i)* describes separation between sub-clusters as it is the mean distance from data point i to all other data points in the nearest sub-cluster (Duan et al., 2019).

Silhouette Coefficient has a range of -1 to +1. Larger values indicate better separation between clusters (Duan et al., 2019).

Accuracy

The percentage of correctly predicted samples to total number of samples in classification problems is referred to as accuracy. (Duan et al., 2019). It is the ratio of correct predictions to sum of all predictions made, presented as a percentage.

Accuracy = correct predictions / aggregate predictions * 100

To avoid bias, k-fold cross validation technique was used to calculate the accuracy.

K-fold cross validation technique has the following steps.

1. Randomize the dataset.
2. Divide the dataset into k groups.
3. For each unique group: take the group as hold out data set and remaining groups as training data set. Fit the model on the training set and evaluate it on the hold out data set. Store the evaluation result and discard the model.
4. Repeat above process k times.
5. Get the average of k results to give a single performance measure.

Calculated performance measures were used to report performance of the proposed model.

4 CHAPTER FOUR: RESULTS, DISCUSSIONS AND CONCLUSIONS

4.1 Data preprocessing

4.1.1 Missing data

Data rows with missing attributes or invalid attributes e.g. #DIV/0!, NIL, n/a, /0, and negative numeric value were dropped. The data was then anonymized by replacing the real cell names with autogenerated cell names. A base station (BTS) cell name has two main parts separated by a hyphen. The first part is the BTS name while the second part is a single digit number between 0 and 9. Table 1 below indicates how data anonymization was done using dummy BTS and Cell names.

Table 2: Anonymizing data

BTS Name	Cell ID	Cell Name	Anonymized Cell Name
Nairobi_University	0	Nairobi_University-0	Site1_cell_0
Nairobi_University	1	Nairobi_University-1	Site1_cell_1
Nairobi_University	2	Nairobi_University-2	Site1_cell_2
Mamlaka_Hostel	0	Mamlaka_Hostel-0	Site2_cell_0
Mamlaka_Hostel	1	Mamlaka_Hostel-1	Site2_cell_1
Mamlaka_Hostel	2	Mamlaka_Hostel-2	Site2_cell_2

The table has two BTS, Nairobi_University and Mamlaka_Hostel. Each BTS has three cells (0,1 and 2) and thus, three cell names. Each BTS name was replaced by an autogenerated name in the format site'X', the hyphen was replaced with an underscore while a cell Id was replaced with an autogenerated name in the format cell_'Y'. An integer X was introduced in the site name part to differentiate one site from the other. Interger Y in the autogenerated cell name is the cell id.

4.2 Detecting anomalies using DBSCAN Algorithm.

The DBSCAN algorithm detects anomalies by identifying points that do not fall in any of the generated clusters and uses two parameters. In the proposed model, the two parameters were chosen through generation of histograms visualizations using anomaly detection dataset. For epsilon, the distance from each observation to the nearest neighbor was calculated and a histogram was generated as shown in Figure 2 below.

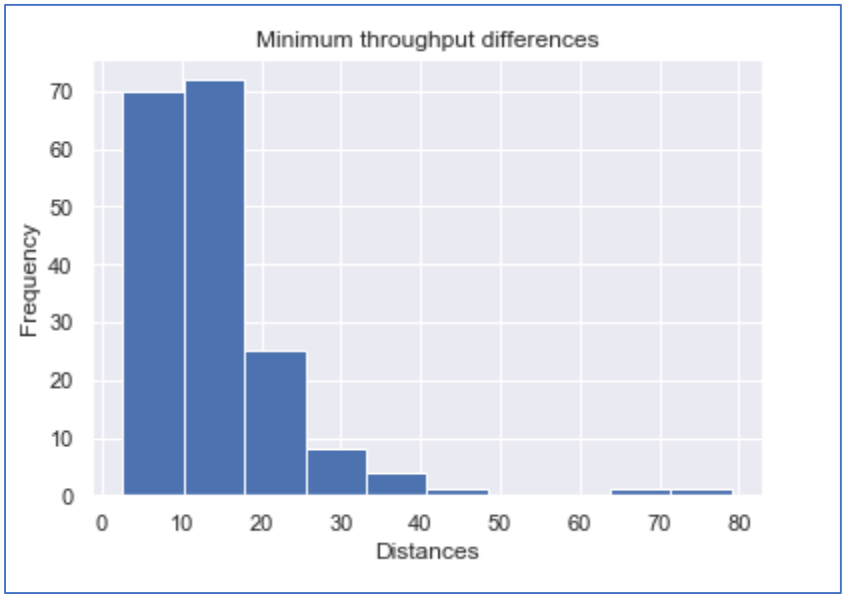


Figure 2: Minimum throughput differences histogram

The histogram indicated that the minimum distance between each point to all other points for majority of points lie within bins 0 and 1. A matplotlib function was used to get the value of the left-hand edge of the third bin which was 17.88. This value was used as the initial value of epsilon. Once the epsilon was chosen, the number of points that lie within each point's epsilon-neighborhood was calculated and a histogram was generated (see Figure 3).

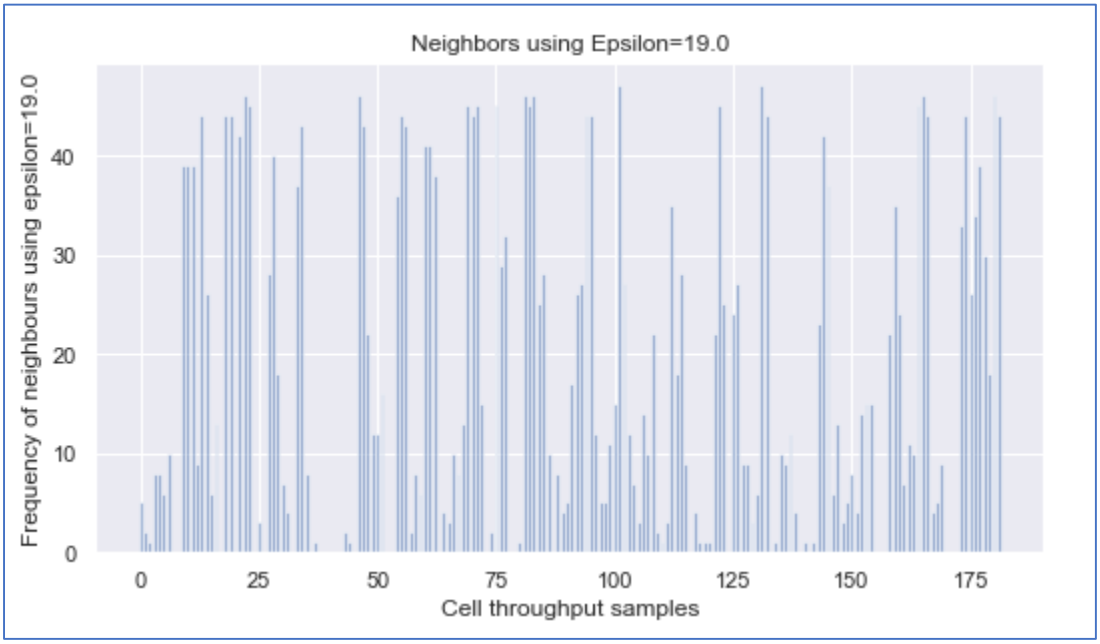


Figure 3: Neighbours using chosen epsilon.

The histogram (Figure 3) shows the number of neighboring points per observation within using a minimum distance (epsilon) of 19. Figure 3 indicated that the majority of points had more than ten (10) neighbors. Different values of epsilon within the neighborhood of 17.88 and different values of minpoints within the neighborhood of 9 were used as DBSCAN parameters and silhouette index calculated. The value Nineteen (19) as the epsilon and a value of ten (10) as the minimum points parameter produced the best value of silhouette index and were used as DBSCAN parameters.

Applying an epsilon of 19 and minimum points of 10 for the dataset, two clusters were generated, and 53 anomalous cells detected. The first cluster had a throughput range from 1Mbps to 20Mbps while the second cluster had a throughput range from 16Mbps to 34Mbps. The two clusters show a separation between cells performance on average user throughput in terms of number of users, allocated channel bandwidth and channel quality. The cluster with higher throughput indicates cells with smaller number of users, larger bandwidth or good channel quality. On the other hand, cluster with lower throughput indicates cells with high number of users, smaller bandwidth or poor channel quality.

4.2.1 Evaluating the model

To evaluate the quality of the model's prediction, silhouette coefficient was calculated using silhouette score function defined in the metric package in sklearn library. Silhouette Coefficient ranges from -1 to +1. A large value indicates good separation between the clusters, (Duan et al., 2019). A silhouette coefficient of 0.451 was achieved (see Figure 4) which indicated a good separation between the two clusters. A cell that did not fit any of the two clusters was labelled as an anomaly.

```
Epsilon is  19.0
Minimum Points parameter is  10
Estimated number of clusters: 2
Estimated number of noise points: 53
Silhouette Coefficient: 0.451
```

Figure 4: DBSCAN Clustering results

4.2.2 Detected anomalies

Figures 5 and 6 shows cells with anomalies in red.

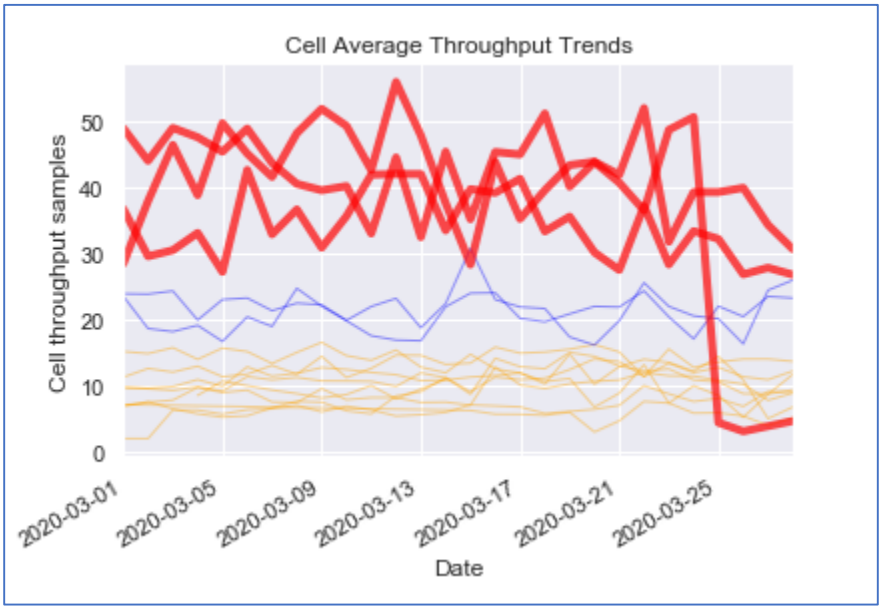


Figure 5: Detected anomalies 1

In Figure 5, two of the cells were outliers with a very high throughput compared to other cells. Unlike the cells in the two clusters, the two cells' throughput varied with large values from one day to the next. The third cell had a step reduction in throughput on 25th March which was due to a fault on site.

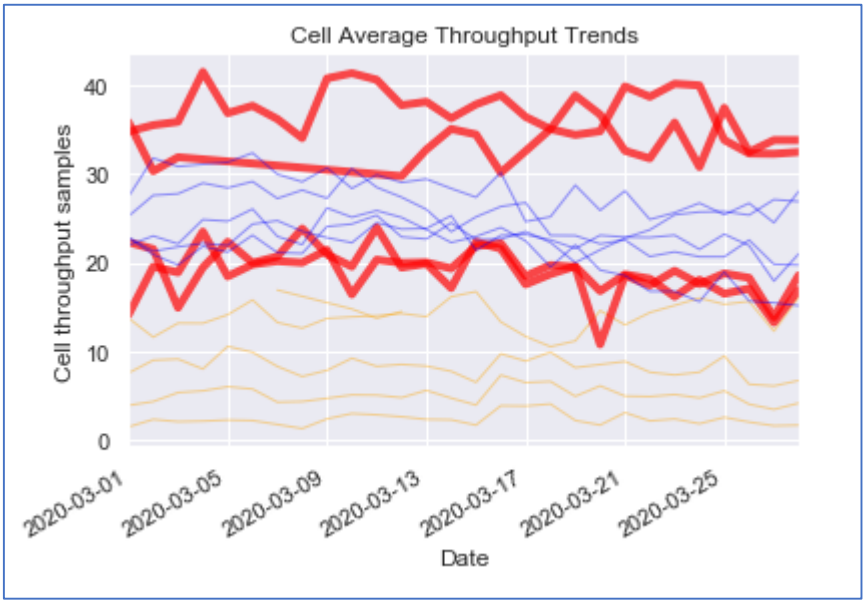


Figure 6: Detected anomalies 2

In Figure 6, there are 4 cells with anomalies two of the cells are outliers with high throughput while the other two cells had a consistently degrading throughput from 16th March.

4.3 Root cause analysis

K-nearest neighbour has one parameter k which was determined through cross-validation. The entire labelled dataset was split into training and test set using a ratio of 80:20. Six different values of k were used, and accuracy calculated. Classification accuracy is the percentage of correctly predicted samples to total number of samples in the classification problem. Table 7 below shows the result of using different values of k. The value of k = 5, that resulted to the best accuracy was chosen.

Table 3: Cross-validation to determine k

Value of k	Accuracy (%)
3	96.4
5	99.96
7	99.73
9	99.94
11	99.86
13	99.23

4.3.1 Performance evaluation for root cause analysis model

The training data was shuffled and then split into 5 and 10 groups for a 5-Fold and 10-Fold cross validation. These values were chosen because they have previously been shown to empirically yield test error rate estimates that do not suffer from bias or high variance. Each unique group was used as hold data while the other groups were used as training data. This ensured that data used for training and testing were non-overlapping and thereby reporting unbiased test results. The model was then fit with the training data and evaluated using the test data generating tables 3 and 4.

Table 4: 5-Fold cross validation

Folds	Accuracy (%)
Fold 1	94.34
Fold 2	94.76
Fold 3	94.62
Fold 4	94.62
Average Accuracy	**94.59**
Standard Deviation	**0.15**

Table 5: 10-fold cross validation

Folds	Accuracy (%)
Fold 1	94.48
Fold 2	94.9
Fold 3	94.62
Fold 4	94.19
Fold 5	94.48
Fold 6	94.9
Fold 7	94.48
Fold 8	94.33
Fold 9	94.05
Fold 10	95.04
Average Accuracy	94.55
Standard Deviation	0.31

Applying the 5-fold cross validation, the model had an accuracy of 94.59% while the 10-fold cross validation had an accuracy of 94.99%. This shows good quality of the model predictions. The standard deviation for 5-fold cross validation was 0.15% while that of 10-fold cross validation was 0.31%. Both had a low standard deviation indicating stable model performance and thus, a good model performance.

4.3.2 Applying the model on new data

After choosing five (5) as the value of k, anomalous cells performance data (see table 6) was used as new data in the root cause analysis module. Out of the fifty-three (53) anomalous cells, forty-six (46) cells were predicted to have normal performance while seven (7) were predicted to have negative impact anomalies. For all the cells predicted to have a negative impact anomaly, visualizations were created showing the KPI which indicates the cause of anomaly.

Table 6: Sample new normalized data

Cell_Name	4G_PS_traffic_Volume(GB)	DL_CQI	DL_PRB_Usage_Rate(%)	L.E-RAB.FailEst.TNL	L.Traffic.DL.AM.Retrans.TxPackets(packet)
Site10_Cell_0	0.135380398	0.593779798	0.170992806	0	0.107979316
Site15_Cell_0	0.865193944	0.625900513	1	0	1
Site15_Cell_1	0.678276456	0.576639401	0.940894557	0	0.719901425
Site8_Cell_2	0.573441888	0.693216625	0.592952407	1	0.530281046
Site22_Cell_6	0.061805428	0.706573985	0.17935784	0	0.057526856
Site16_Cell_0	0.202144259	0.720865426	0.216308392	0	0.222368854

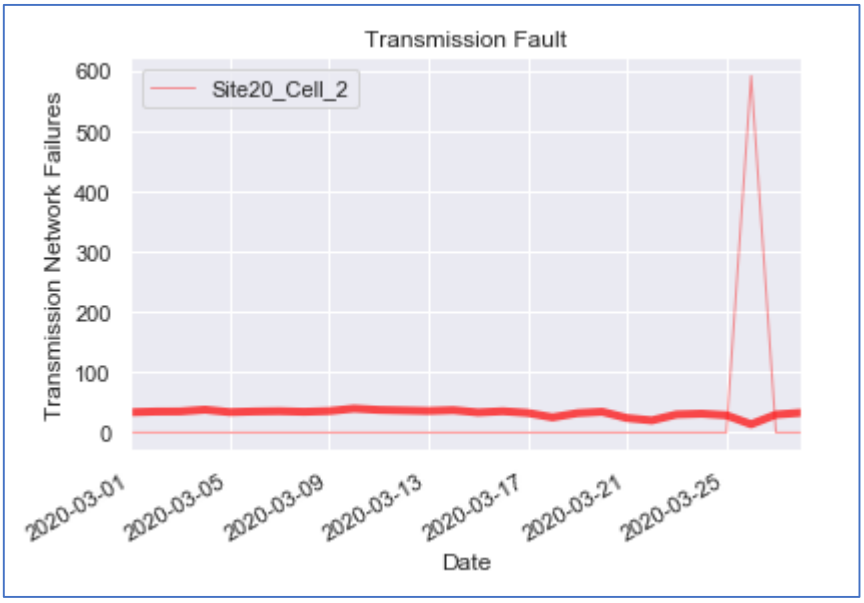


Figure 7: Transmission network failure

Figure 7 shows a cell with a spike in transmission network link failures which caused a degradation in user throughput.

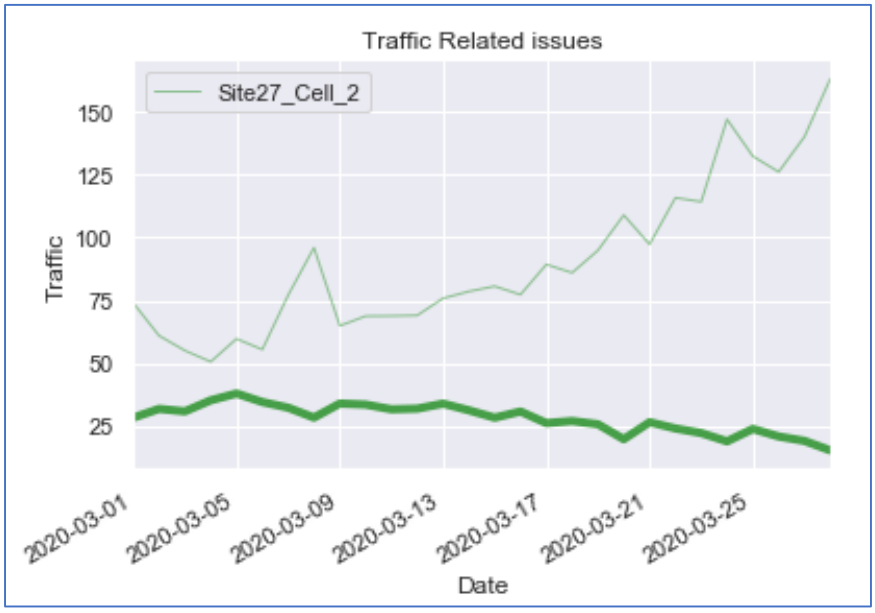


Figure 8: Traffic related anomalies

Figure 8 shows a cell which experienced an increase in traffic from 14th March causing a degradation in average user throughput.

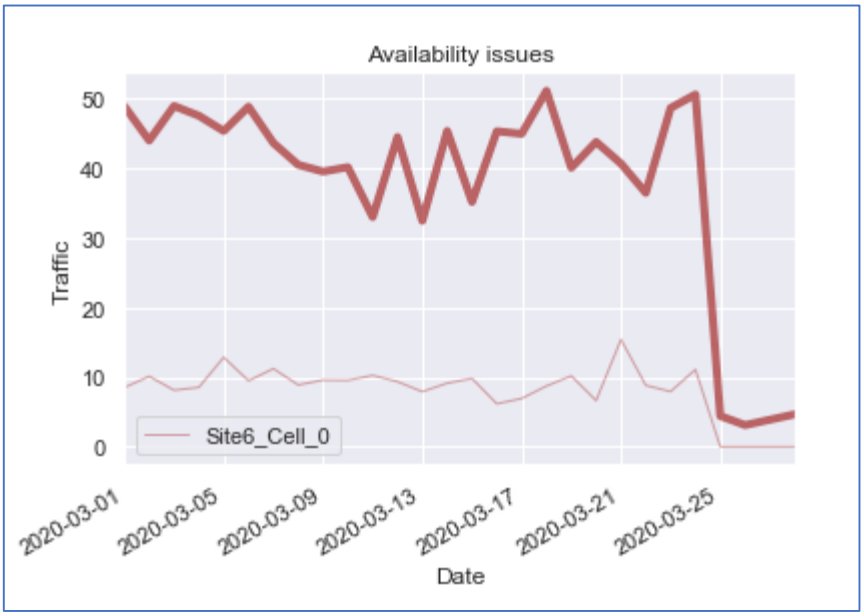


Figure 9: Availability related anomaly

Figure 9 shows a cell that experienced an availability problem on 25th March causing a step reduction in average user throughput.

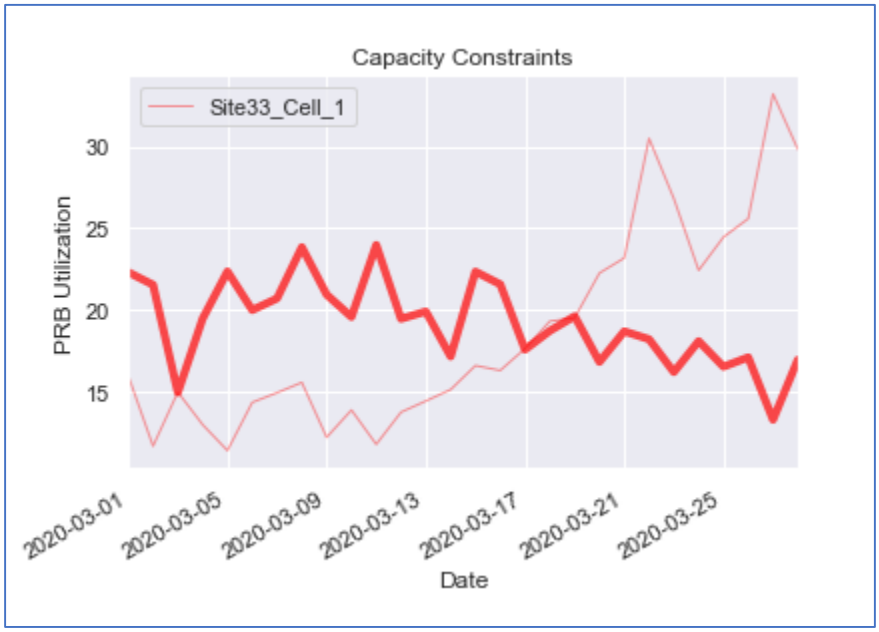


Figure 10: Capacity Constraints anomaly

Figure 10 shows a cell whose physical resource block (PRB) utilization increased from 14th March causing a degradation in average user throughput.

4.4 Discussions

Use of machine learning has been identified as a great tool to introduce intelligence to network management and to develop self/automated healing tools. This project implemented a model to detect anomalies and identify the root cause of those anomalies. The model was able to detect anomalies affecting data throughput and diagnose the root cause that resulted in the degradation in data throughput.

The proposed model anomaly detection adopted an unsupervised learning algorithm similar to approaches used by Andrades et al. (2016) and Josefsson (2017). However, the implementation used in this model is different. While Andrades system worked by clustering cells based on the correlation between cell KPIs and fault causes, proposed model clustered cells based on the target metric trends in time dimension. Both approaches can detect known and unknown anomalies. However, in Andrades model, obtained anomalous clusters had to be labeled with identified causes by experts, who are scarce and expensive. To ensure that a fault cause is not misdiagnosed, Andrades used silhouette index and percentile-based approach for border root causes. Average silhouette index for each diagnosis was calculated and used to evaluate the

quality of each root cause analysis result. The root cause with a higher silhouette index was selected. This approach led to reduction of diagnosing error from 1% to 0.77%. To eliminate the need for an expert in the training phase of the proposed model, DBSCAN was used to cluster cells based on the key metric trends and any cell that didn't fall in any cluster was labelled as anomalous. A silhouette index of 0.451 was achieved in proposed model showing a good quality of obtained clusters. The proposed model anomaly detection function performed as expected and was able to detect anomalies with both positive and negative impact on data throughput. It was also able to detect anomalies that affect user experience without the data throughput crossing set thresholds used in manual anomaly detection methods.

Root cause analysis in the proposed model adopted supervised learning algorithm (KNN). The model takes advantage of available labelled data to learn the pattern of anomalies based on cell KPIs. Once the model is trained, it is used to identify the root cause of detected anomalies. Past researches have shown good performance of supervised learning approach for root cause analysis. Josefsson (2017) used both supervised and unsupervised approaches in his root cause analysis model and on comparing results from both approaches, supervised approach performed better. Alvarez (2018) also used supervised learning in his anomaly detection and root cause analysis, although data labelling was done internally in his system. Kostas (2018) compared 7 supervised learning algorithms in his system for anomaly detection and KNN had the best performance of 97% accuracy. Other algorithms had an accuracy of: Multiple Layer Perceptron (MLP) 83%, Naïve Bayes 86%, QDA 86%, Random forest 94%, AdaBoost 94% and ID3 95%. The proposed model used historical data labelled by Engineers in their network optimization activities which helped to speed up root cause analysis. The model had a performance of 94.59% accuracy. The use of supervised learning in root cause analysis is limited in that it relies on previously known causes and can misdiagnose unknown anomalies. However, this can be remedied by visualizations generated after root cause analysis which can help Engineers identify such cases. Anomaly detection and root cause analysis models were linked up in the application phase whereby the output of anomaly detection was used as new data for the root cause analysis model after training the model.

Mobile networks function relatively well most of the active time and a failure or network degradation appear with a low probability. Therefore, normal data is often much more than

abnormal data thus generating imbalanced data for classification problems. To deal with this problem in the proposed model, under-sampling technique was used whereby normal data was randomized and a certain amount was removed. Randomization was done to prevent loss of some important information in majority classes, which is a problem of using under-sampling technique. To deal with imbalanced data problem, Zang et al. (2019) and Wang et al. (2019) proposed the use of over-sampling minority class data was increased. Over-sampling, on the other hand, may result in over-fitting due to the duplicating operations of minority class samples.

4.4.1 Achievements

This project was able to automate the process of detecting and diagnosing the root cause of those anomalies that affect LTE data throughput. The model if integrated in the network can create efficiency in network optimization through offloading the work of detecting and diagnosing LTE data throughput anomalies from the network Engineers.

Whereas the manual process focuses on identifying anomalies by using thresholds, this model can detect anomalies that affect user experience without the data throughput crossing set thresholds. An example is a transmission fault that degrades average base station cell throughput from 20Mbps to 10Mbps. Such a degradation may not be picked through manual process which focuses on identifying cells with less than 1Mbps on average user throughput (worst cells), yet the degradation affects experience of users served by that base station. Once such an anomaly is detected, the root cause module can diagnose the cause and generate a visualization after which the network Engineer will focus on resolving the transmission fault.

With such a model, network issues can be picked proactively and resolved before a customer complains thus improving network promoter score.

4.5 Conclusion and recommendations

In this project, a machine learning model that aims to detect anomalies affecting data throughput in LTE and root cause analysis of anomalies affecting data throughput negatively was proposed. The model makes use of real network KPIs and labelled root cause data from a Kenyan operator, unsupervised learning for anomaly detection and supervised learning to diagnose network anomalies. The model emulates the manual process followed by radio network Engineers to detect and diagnose network anomalies.

Results show the effectiveness of the proposed anomaly detection and root cause analysis model. It was able to detect anomalies which had both positive and negative impact on data throughput. From the results, those data points that had very high throughput and didn't fall in any cluster were outliers and considered as positive anomalies. Anomalies with negative impact were those that caused a reduction in data throughputs like the one caused by capacity constraints.

To improve the quality of predictions for the root cause analysis module, there is need to sensitize network Engineers on the need to properly record root cause of new anomalies in the network. With both positive and negative impact anomalies' labelled data, the model will be able to diagnose both type of anomalies in future.

The next steps for future research direction would be to use reinforcement learning to resolve anomalies that do not need physical intervention on site through configuration robots in the network. This will allow the module to work autonomously for the self-healing functionality.

5 REFERENCES

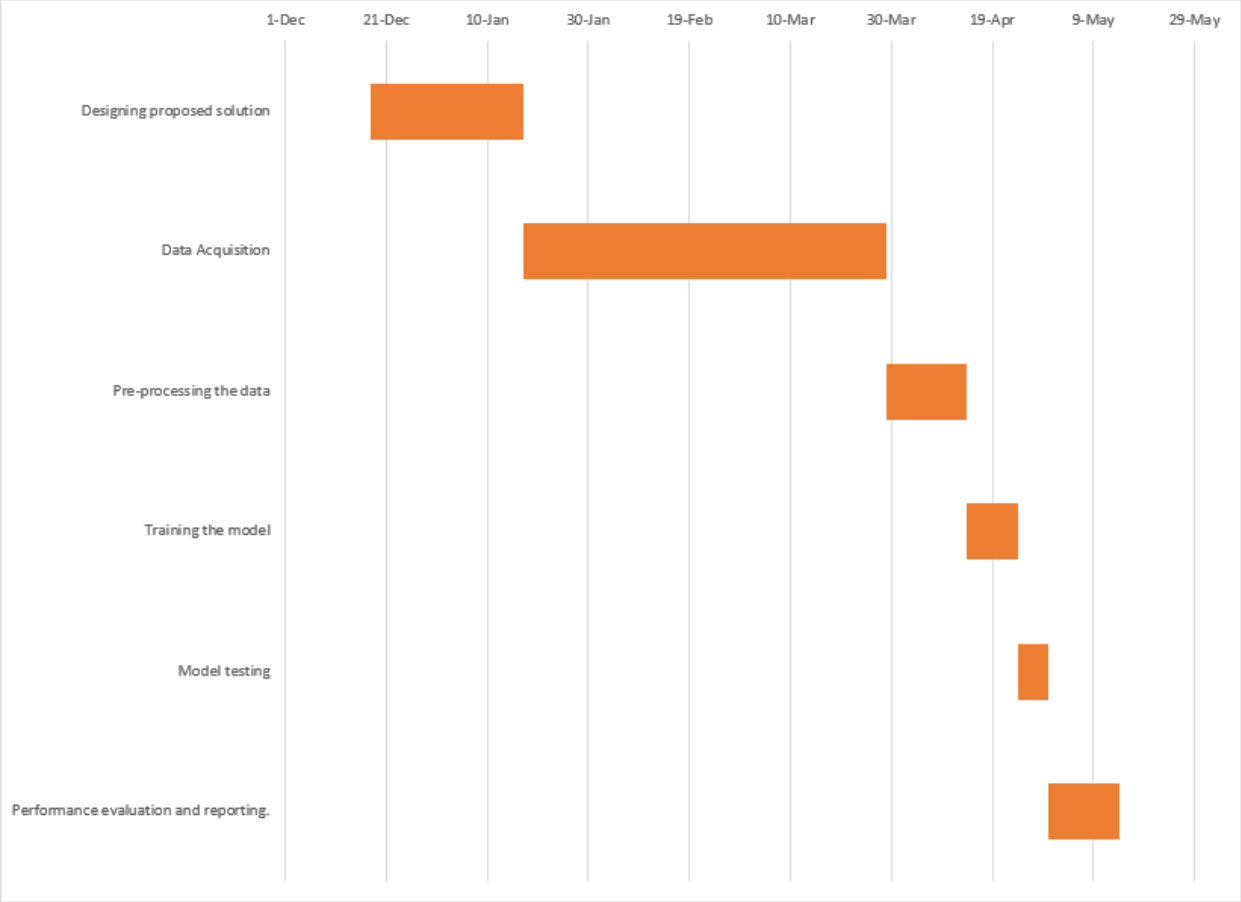
1. Zhang, W., Ford, R., Cho, J., Zhang, C. J., Zhang, Y., & Raychaudhuri, D. (2018), Self-Organizing Cellular Radio Access Network with Deep Learning. Retrieved from https://www.semanticscholar.org/paper
2. Khatib, E. J. (2017). Data Analytics and Knowledge Discovery for Root Cause Analysis in LTE Self-Organizing Networks. Retrieved from https://www.semanticscholar.org.
3. Mdini, M. & Simon, G. & Blanc, A. & Lecoeuvre, J. (2018). ARCD: A Solution for Root Cause Diagnosis in Mobile Networks. Retrieved from https://www.semanticscholar.org.
4. Ciocarlie, G. F., & Novaczki, S., & Sanneck, H. (n.d.). Detecting Anomalies in Cellular Networks Using an Ensemble Method. Retrieved from https://www.researchgate.net.
5. Alvarez, S. L. (2018). Anomaly Detection and Root Cause Analysis for LTE Radio Base Stations. Retrieved from https://pdfs.semanticscholar.org.
6. Andrades, G., Muñoz, P., Serrano, I., & Barco R, (2016). Automatic root cause analysis for LTE networks based on unsupervised techniques. Retrieved from https://www.researchgate.net
7. Mobolize. (2014). Can Mobile Networks Keep Up with Data Demand? Retrieved from https://www.mobolize.com/2014/09/30/can-mobile-networks-keep-up-with-demand/
8. Eirini, L., Dimitris, T., Nikos, P., & Lazaros, M. (2015). Quality of Experience Management in Mobile Cellular Networks: Key Issues and Design Challenges. IEEE Communications Magazine. 53. 10.1109/MCOM.2015.7158278. Retrieved from https://www.researchgate.net/publication
9. Zhang, C., Patras, P., & Haddadi, H. (2018). Deep Learning in Mobile and Wireless Networking: A Survey. ArXiv, abs/1803.04311. Retrieved from https://www.academia.edu/
10. Grondalen, O., & Osterbo, O., (2012). Benefits of Self-Organizing Networks (SON) for Mobile Operators. Retrieved from https://www.hindawi.com/journals/
11. Josefsson, T. (2017). Root-cause analysis through machine learning in the cloud. Retrieved from http://www.diva-portal.org
12. Eamrurksiri, A. (2017). Applying Machine Learning to LTE/5G Performance Trend Analysis. Retrieved from https://pdfs.semanticscholar.org

13. Mamuna, A. S. M., & JuhaValimakia, J. (2018). Anomaly Detection and Classification in Cellular Networks Using Automatic Labeling Technique for Applying Supervised Learning. Retrieved from https://www.sciencedirect.com
14. Zhang, T., Zhu, K., & Hossain, E. (2019). Data-Driven Machine Learning Techniques for Self-healing in Cellular Wireless Networks: Challenges and Solutions. Retrieved from https://arxiv.org
15. Mismar, F. B. & Evans, B. L. (2019). Deep Q-Learning for Self-Organizing Networks Fault Management and Radio Performance Improvement. Retrieved from https://arxiv.org
16. Palacios, D., Khatib, E., & Barco, R. (2016). Combination of multiple diagnosis systems in Self-Healing networks. Retrieved from https://www.sciencedirect.com
17. Wang, Y., Zhu, K., Sun, M., & Deng, Y. (2019). An Ensemble Learning Approach for Fault Diagnosis in Self-Organizing Heterogeneous Networks. Retrieved from https://ieeexplore.ieee.org
18. Adrian, G. R., & Niclas, O. N. (2017). Detecting Network Degradation Using Machine Learning: Predicting abnormal network behavior with anomaly detection. Retrieved from https://schlieplab.org
19. Kostas, K. (2018). Anomaly Detection in Networks Using Machine Learning. Retrieved from https://www.researchgate.net
20. Theobald, O. (2017). Machine Learning for Absolute Beginners. 2nd Edition. Retrieved from https://www.scribd.com.
21. Zhang, W., Ford, R., Cho, J., Zhang, C. J., Zhang, Y., & Raychaudhuri, D. (2019). Self-Organizing Cellular Radio Access Network with Deep Learning. Retrieved from https://www.researchgate.net
22. Dadhe, A. (2016). Research Methodology. Retrieved from https://www.scribd.com/book/332959095/Research-Methodology
23. Communication Authority of Kenya, (2017). Framework for the assessment of service quality of telecommunication systems and services. Retrieved from https://ca.go.ke/
24. Bahia, K., & Suardi, S. (2019). The State of Mobile Internet Connectivity 2019. Retrieved from https://www.gsma.com/

25. Celik, M., Dadaser-Celik, F., & Dokuz, A., (2011). Anomaly Detection in Temperature Data Using DBSCAN Algorithm. INISTA 2011 - 2011 International Symposium on INnovations in Intelligent SysTems and Applications. 10.1109/INISTA.2011.5946052.
26. Xu, D. & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms: A survey. Retrieved from https://doi.org/10.1007/s40745-015-0040-1
27. Duan, H., Wei, Y., Liu, P. & Yin, H. (2019). A Novel Ensemble Framework Based on K-Means and Resampling for Imbalanced Data. Retrieved from https://doi.org/10.3390/app10051684
28. Kiwoli, L., Sam, A. & Manasseh, E. (2017). Performance analysis of carrier aggregation for various mobile network implementations scenario based on spectrum allocated. Retrieved from https://arxiv.org/
29. Mohana, H. K., Mohankumar, N. M., Devaraju, J. T. & Swetha. (2014). Effect of Bandwidth Scalability on System Performance in the Downlink LTE Systems. Retrieved from https://www.researchgate.net/
30. Babur, I. H., Ahmad, J., Ahmad, B. & Habib, M. (2015). Analysis of dbscan clustering technique on different datasets using weka tool. Retrieved from https://www.researchgate.net/

6 APPENDICES

6.1 Project Schedule



6.2 Budget

Budget Line	Item	Budget Description	Units	Unit Measure	Cost per unit	Amount
1	**Tools**					
1.1	Python	Programming language	25	MB	0.12	3
1.2	Anaconda IDE	Development environment	80	MB	0.12	10
2	**Transport**					
1.1	College	Meeting supervisor and group members	20	Trip	800	16,000
1.2	Mobile Operator	Visits to the mobile operator	12	Trip	800	9,600
2	**Internet expenses**	Cost of internet access	5	month	3,900	19,500
	Grand Total					**45,113**