3G and 4G Paging Success Rate based Mobile Network Anomaly Detection using Supervised and Unsupervised Learning

by

Md Rakibul Ahasan
20166055

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
April 2022

Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rakib

Md Rakibul Ahasan

20166055

Approval

The thesis "3G and 4G Paging Success Rate based Mobile Network Anomaly Detection using Supervised and Unsupervised Learning" submitted by

1. Md Rakibul Ahasan (20166055)

Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on April 09, 2022.

Examining Committee:

External Examiner:
(Member)

Mohammad Shamsul Arefin, Ph.D.

Professor
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET)

Internal Examiner:
(Member)

Md Khalilur Rhaman, Ph.D.

Associate Professor
Department of Computer Science and Engineering
Brac University

Internal Examiner:
(Member)

Md. Ashraful Alam, Ph.D.

Associate Professor
Department of Computer Science and Engineering
Brac University

Supervisor:
(Member)

Md. Golam Robiul Alam, Ph.D.

Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Amitabha Chakrabarty, Ph.D.

Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi , Ph.D.

Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In a mobile network, there are a lot of data that can provide network detail about network efficiency, robustness, and availability. A type of data is mobile network performance data obtained from the key performance indicators (KPI) or the key quality indicators (KQI). An integral part of mobile network monitoring is it monitor any unusual pattern in the performance data. The pattern detection or anomaly detection use case from performance data is essential for mobile operators because it detects issues in the network that are not possible to detect by the network alarms. A machine learning-based anomaly detection model is most common nowadays. This thesis demonstrates a supervised and unsupervised machine learning-based anomaly detection model. The base data set is paging success rate performance data of day-level and hourly-level granularity. Secondly, a comparative analysis is present over various anomaly detection models. Thirdly, the data used in this paper has an imbalance scenario and how the re-sampling technique can affect the outcome of the anomaly detection model. Lastly, one supervised machine learning recommends mobile network anomaly detection. However, implementing supervised machine learning over a large data set is more computational because it requires ground truth determination. On the other hand, unsupervised machine learning will cluster various data volumes without any prerequisite. If proper tuning is in place on this model, it will give an efficient anomaly detection. Another aspect of this thesis is to identify unsupervised machine learning that is best suited for mobile network anomaly detection. To do that a benchmarking approach is performed over three unsupervised machine learning, and these are K-means, DBSCAN, and HDBSCAN. The thumb rule of the benchmark follows as converting the unsupervised machine learning output into a classification problem and then measuring the model performance. The deep learning implication of anomaly detection in 4G network performance data exercise in this thesis and an autoencoder used to see how it performs in anomaly detection with moderate accuracy.

Keywords: Anomaly Detection, Supervised learning, KPI, Mobile Networks,SMOTE, Unsupervised learning, K-means, DBSCAN, HDBSCAN, Mobile network anomaly detection, Autoencoder.

Acknowledgement

Firstly, all praise to the Great Allah for whom my thesis have been completed without any major interruption.
Secondly, to my advisor Md. Golam Robiul Alam sir for his kind support and advice in my work. He helped me whenever I needed help.
Thirdly, My sincere gratitude to the judging panel of IEEE UEMCON 2021, 2021 IEEE SPICSON, and ICAEEE 2022. Though my paper not accepted there, all the reviews they gave helped me a lot in my later works.
Fourthly, ICISET 2022 and the judging panel where my paper "Benchmarking Effective Unsupervised Machine Learning for Mobile Network Anomaly Detection" got accepted.
Fifthly, TENSYMP2022 and the judging panel where my paper "Supervised Learning based Mobile Network Anomaly Detection from Key Performance Indicator KPI Data" got accepted.
And finally to my family without their throughout support it may not be possible. With their kind support and prayer I am now on the verge of my graduation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The importance of mobile networks is increasing day by day because they can provide support in the medical sector, distance learning, fire services, and many more, and high reliability, speed, and robustness when serving the end-user is the prerequisite. A mobile network is self-organizing [13], consisting of multiple embedded systems and those that support all the services underlying. To maintain those service availability and other qualities of service factors, mobile operators monitor the whole network objects alarm and performance. An alarm from a network object indicates a direct service interruption or a warning of possible service interruption. Nevertheless, an alarm is essential for the understanding of service interruption. However, alarms are not providing network analysis for hidden issues causing service interruption. In such cases, performance management [6] is widely used. The monitoring of performance management supports a wide array of issue identification.

1.2 Research Problem

In a mobile network, their various domains, each of these domains contains many network objects. The network object generates counters, and the KPI and KQI are measured. A mobile operator continuously monitors the network performance data and uses automation to reach a decision. The most efficient anomaly detection relies on performance management. Let's consider a network object which is following the usual data pattern. An anomaly is considered if there is a sudden change or spike found in the pattern. Such anomalies might lead to a critical issue and cause a catastrophic problem. Several types of research are ongoing based on anomaly detection models and the types of machine learning are supervised machine learning, unsupervised machine learning, and semi-supervised learning.
There is various use of data set that can be beneficial. One circumstance shows how the quality of experience KPI [17] from the mobile terminal is further classified into a scoring system and benefits suggestive promotion of mobile terminal. In a network, a computer or machine is an integral part and possesses a lot of traffic data. An anomaly detection model is independent of history baseline data and focuses on generating false positive data applied to computer traffic data. A lot of research on mobile networks [29] used the supervised learning algorithm IP-OCSVM and used a modified decision function to categorize the detection anomaly set. Data possesses

lots of information, hence the optimum selection of machine learning algorithms is essential for a data set. Statistical anomaly detection [5] is another way that changes the data more dynamically. For example, it is impossible to get a satisfactory result of anomaly detection in raw data without any exploratory data analysis or statistical Implication. A thumb rule for supervised machine learning-based anomaly detection is mandatory. First, do some exploratory data analysis (EDA) and Statistical Implications [8] on the data and then apply the base algorithm. Then tune on the parameter and so on. There are other aspects of the data set while classification is imposed from the anomaly detection model. There are cases where the data is not balanced enough. Here Synthetic Minority Over-sampling Technique (SMOTE) [27] is an efficient technique that works on the imbalance scenario and improves the performance of the machine learning-based mobile network anomaly detection model. Time Series analysis is very much co-related with the data set, and research [15] depicted that if the disruption is in time series data with a sequential hypothesis test, anomalies are possible to identify. Most of the related work described only one type of Machine learning implication on the data set and improving the machine learning output using feature engineering or data mining. Very little research works on the subset of the performance data set, and supervised machine learning can be helpful in such type of data. Chapter 3 focused on identifying an anomaly detection model based on supervised learning and how it works better on different granularities for network performance data.

Due to the large volume in size of a mobile network, it is prone to different types of problems. Early detection of this problem will greatly increase service availability. Hence, anomaly detection is very much important in a mobile network. A Network consists of an internal domain and an external domain that produces a large number of data. This data is very much useful for machine learning-based anomaly detection and it is done by a combination of unsupervised learning K-means and decision tree (DT) supervised learning[12]. But it is not mentioned why K-means is chosen as unsupervised machine learning. In another research, a similar combination of K-means and decision tree is used to predict the anomaly in computer traffic[7]. But K-means is a centroid and flat clustering system and this clustering continue until the overall cluster reaches convergence. Also, it is attempting to consider all the data points in the data set to create the cluster, which is not an ideal approach for anomaly detection through clustering. Another type of unsupervised machine learning is density-dependent. Flat density-based clustering is DBSCAN and hierarchical density-based clustering is HDBSCAN. Both of these clusterings are very effective while dealing with noise in a data set. Such noise is further categorized as an anomaly. Research has taken place where DBSCAN performed in temperature data[9]. Another anomaly was detected in-network sensor data from the wireless domain by DBSCAN along with SVM[22]. HDBSCAN[23] is also used to detect anomalies in the energy domain. As explained anomaly detection is very much important for the mobile network, however for a dataset there is no direct rule on which machine learning is better to use. Due to the large data size of mobile network data, unsupervised machine learning is the best fit. Current research depicted that for different domain energy, computer traffic, wireless network unsupervised learning is used. In this chapter 4, the performance data set of a mobile network will be used, and through a benchmark approach, it will identify efficient unsupervised learning for anomaly detection.

PCA is well known for dimensionality reduction, but it is not able to identify the anomaly. In [14] demonstrated how autoencoders is detecting anomalies where linear PCA fails and does not require any complex computation. Autoencoders are unsupervised learning which is trained from normal or abnormal behavior on the data and classify the test data accordingly. Anomaly detection by autoencoder is used in High-Performance Computing Systems [[25] where training is done over the normal data because it improves accuracy varies from 88 percent to 96 percent. In wireless sensor network (WSN) [20] autoencoders are used in two different levels of anomaly detection one is in wireless sensor and another is in the IoT cloud level. Here it demonstrated high detection accuracy and a low false alarm rate. In mobile wireless networks, the performance data is at the time series level and autoencoders have proven efficient for time series DDoS cyber-attack [30] in detecting anomalies.In a mobile network, the data size is huge because it serves a huge customer base. To serve the customer at different domain levels monitoring on anomaly is necessary. One of these is the spectrum that is allocated to the user during connecting to the network. During this allocation procedure autoencoders, [16] are useful because autoencoders are unsupervised learning and do not require any pre-requisite of data labeling. There are various types of the autoencoder is available and those are used as well for anomaly detection for example convolutional autoencoder [18]
All in all, there is no end to research found that work on base mobile network performance data and perform benchmarking or comparative approach which is useful for a mobile operator for its anomaly detection mechanism. This thesis aims to analyze this question and provide a viable recommendation for supervised and unsupervised learning.

1.3 Research Objectives

The main research objective of this thesis is to find a viable mobile network anomaly detection model based on supervised or unsupervised learning. During this process, the base network data is mobile network performance data. Usually, the performance data is derived from various network element of the mobile network and provide a deep insight into the network apart from network fault. The focused objective of this thesis is:

1. This thesis demonstrate how supervised and unsupervised machine learning behave over mobile network performance data and provide an efficient anomaly detection model.

2. Mobile network performance data generates in different granularity like hourly level, day level, or monthly level. This thesis focuses on these different granular performance data and how different machine learning is used to determine an anomaly.

3. This thesis also talks about the individual data size and imbalance property of mobile network performance data. This thesis demonstrates how it is overcome by SMOTE.

4. Subject matter expert level validation is also exercised in this thesis as a new approach for mobile network anomaly detection.

5. Variable autoencoder architecture and threshold influence observed over mobile network anomaly detection.

1.4 Research Methodology

This thesis is set to produce an anomaly detection model for mobile network performance data using different machine learning. From the vast area of supervised and unsupervised machine learning few have been chosen and a comparative approach related to supervised and unsupervised learning has been proposed. The performance data feed from a real network is troublesome work and requires complex integration. Hence there is a segment of 3G and 4G performance data taken from real network data and used in this research.

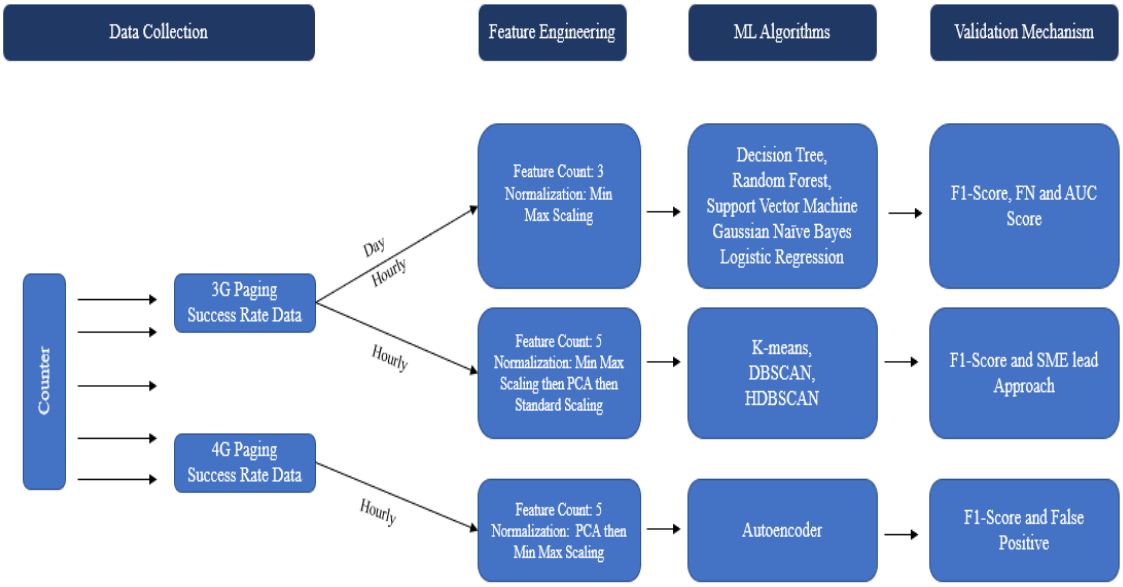

Figure 1.1: Overall Methodology

This mobile network performance data is collected in two different granularity one is day level data and another is hourly level data. Both this data is collected from real network and based network element is from the core domain. Fig 1.1 details the overall methodology of this thesis work. The main verticals are data collection, feature engineering, machine learning implementation, and validation as applicable. This thesis is also divided into three levels of work, anomaly detection by supervised learning or by unsupervised learning, or by autoencoder. Each of these divisions has a separate feature count in the data set and different machine learning algorithms implication in 1.1.
The validation of each model to get supervised, unsupervised, and autoencoder machine learning is done from the confusion matrix, F1-Score, and AUC scores. The false negative and the false positive are mostly used from the confusion matrix.

1.5 Scope and Limitation

This thesis aims to provide an anomaly detection model of paging success rate data of a mobile network. The data model is designed to work on numerical data or real

values such as paging data and its features. The model will not be able to deal with textual data.
Furthermore, In this thesis, the detection of anomalies of mobile network performance data is explored in the core network data only. But in the mobile network, there are other domains such as transport, access domain, etc which are not explored for this thesis anomaly detection model.
Finally, the proposed machine learning for anomaly detection has not been implemented in the real network though the data set is from the real network.

1.6 Document Outline

In Chapter 2 has the details of mobile network KPI understanding and in-depth idea of different types of paging in the mobile network and its features. Derive the anomalies in mobile paging data and how it effects the customer.
The details of a comparison of various supervised learning-based anomaly detection over a mobile network are in Chapter 3.
The Benchmarking of different unsupervised machine learning for mobile network anomaly detection is described in chapter 4.
Chapter 5 has given a brief overview of how a neural network-based autoencoder is useful for mobile network anomaly detection. The overall outcome of the thesis describes in the chapter 6 conclusion section.

Chapter 2

Mobile Network KPI Understanding

2.1 Mobile Network KPI Understanding

The performance data set is the base data set of this thesis. Performance data is spread over various network domains. In this thesis, a paging success rate a subset of performance data will be used. To understand the paging and its importance, need to understand the call establishment procedure in a mobile network domain. This is described in below Fig 2.1.

2.1.1 Paging success rate Network KPI

Let us consider a customer 'A' is giving a call to customer 'B'. A call from customer 'A' is first tagged with a BTS1 and BTS1 communicating with BSC1 for channel request and channel allocation is done from BSC1. After that call setup request was given to MSC from BTS1 and MSC started the call proceeding. During this time MSC is sending a paging request to all the BSC so that the customer 'B' and its associated BSC2 and BTS2 can find it. Once the paging request is successful a corresponding paging response will be sent from BSC2 to MSC. Upon having that MSC complete the call setup procedure with BTS2, A call between caller party 'A' and called party 'B' is established.
The term paging is defined in a mobile network as an MSC is giving paging towards the BSC or RNC and vice versa. This is very much important to monitor the paging success rate KPI because it is directly associated with the call flow and call establishment. If the paging success rate is decreasing for a mobile network, meaning customer is not able to communicate with each other. The paging success rate is associated with paging response and paging failure[4]. The paging response time is when a BSC/RNC sends a paging response, in response to a paging request from MSC. The paging failure is the failure count when MSC receives an error from BSC/RNC. In this thesis, a month of an hourly data set of paging success rate KPI data will be used. Associated features are as below:

1. PSR, This is denoted as the overall paging success rate.
2. PA, This is denoted the number of paging attempts is given from one node to another node.

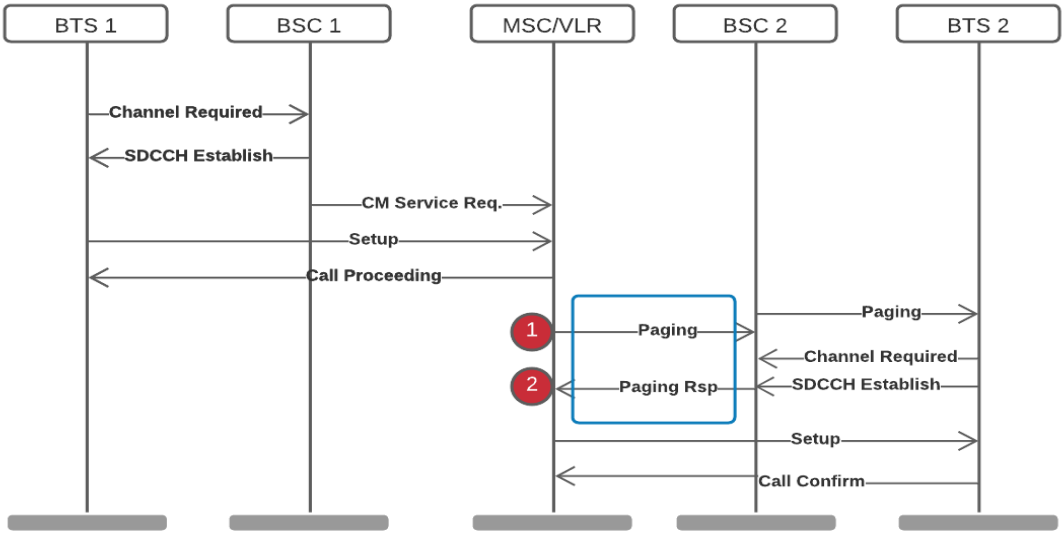

Figure 2.1: Mobile Call Establishment Procedure

3. PF, This is known as the paging fail count between one node to another node.
4. FTPSR, This is denoted the first paging success rate when in a network multiple paging attempt is configured.
5. FPA, This is denoted the number of first paging attempt count.

2.1.2 4G network Paging success rate Network KPI

In the above section the paging criteria is described between MSC and BSC. It is also mentioned paging is require for various domain. In Fig 2.2 a typical 4G network is described where the key components are User Equipment (UE), eNodeB equivalent to base station of 3G network, Mobility Management Entity (MME), Home Subscriber Server (HSS) equivalent to HLR in traditional GSM network, Serving Gateway (SGW) and PDN Gateway (PGW). MME is the main component which is used to authenticate UE in Robi network. It also used to track the UE and select suitable SGW and PGW which is appropriate for the UE.
To do deep learning autoencoder based anomaly detection S1-MME PS Paging success rate KPI and its respective feature is used. The main features are described in below.

1. PPSR, This is denoted as the overall 4G network paging success rate.
2. PPRT, This is denoted the request time of paging given from the MME to UE
3. PPST, This is denoted as the success time of the number of paging given from MME to UE
4. PPFT, This is denoted as the success time of the number of paging given from MME to UE.
5. PPD, This is denoted as the delay observed during the paging time.

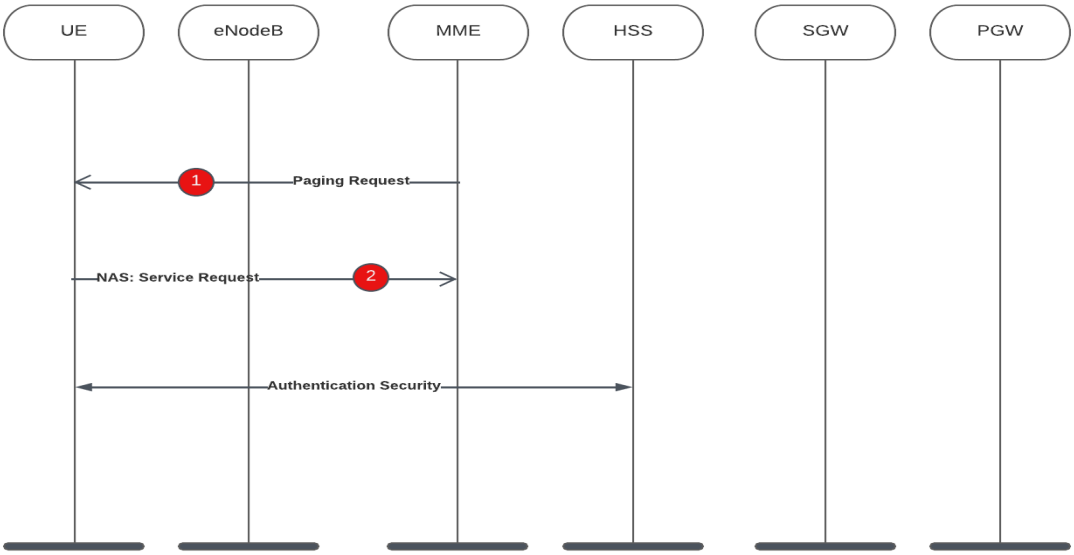

Figure 2.2: 4G network Paging success rate Network KPI

The PS Paging success rate is derived from having the PS Paging success time divided by PS Paging request time. The PS Paging success time denotes Measure the total number of NAS:Service Request messages at measurement point 2 in Fig 2.2.And PS Paging Requests measure the total number of Paging messages at measurement point 1.

2.1.3 Anomalies in Mobile Network Performance Data

Let's discuss what is considered to be anomalies in mobile network paging success rate data. Usually, the paging success rate is derived from the different counter of 3G and 4G network elements, and formulas are derived in eq 2.1 and 2.2. A core feature of the base 3G paging success rate is mentioned below. However, it also needs to mention in some mobile networks instead of one paging multiple paging is configured.

1. PA, This is denoted the number of paging attempts is given from one node to another node.

2. PF, This is known as the paging fail count between one node to another node.

$$3GPagingSuccessRate = \frac{PA}{PA + PF} \tag{2.1}$$

On the other hand in 4G, PS paging success rate derive the how 4G subscriber or mobile is efficiently authenticated with MME of 4G network. The key feature of measuring the 4G PS paging success rate is mentioned below:

1. PPRT, This is denoted the request time of paging given from the MME to UE

2. PPST, This is denoted as the success time of the number of paging given from MME to UE

$$4GPSPagingSuccessRate = \frac{PPST}{PPRT} \tag{2.2}$$

The 3G or 4G paging success rate data is quite huge and generated in different granularity daily level and hourly levels. Before implementing any machine learning on those data sets ground truth determination is required. To that, the whole data set has asses by a subject matter expert and after assessment one formula is provided. It is described in eq 2.3.

$$Thresholdof Anomalies = maxPSR - 5 \tag{2.3}$$

Though this equation looks simple the inheritance of this formula is vast. Because the threshold needs to define for the individual network object level, not all the network objects. The reason is each network object is serving a different customer base and has different configurations, hence each network object has to be treated differently to detect an anomaly. In below table has details of a network object paging success rate data for 5 days to give an understanding of the anomaly. The total count of the paging data set is 120 and features are PSR, PR, PF, FTPSR, and FTPR. According to eq 2.3, the max PSR value of 5 days paging success rate data is 92.70 and the threshold of anomalies is 87.70. Meanings to say as per the subject matter expert provided formula any paging success rate data less than 87.70 will be considered as an anomaly in the ground truth.

Table 2.1: Normal Data Set of Paging Success Rate data

PSR	**PA**	**PF**	**FTPSR**	**FTPR**
90.89	87272	8737	87.26	83786
91.29	45117	4300	88.12	43592
91.28	29303	2798	88.52	28418
91.23	24557	2359	88.68	23870
90.91	27725	2772	87.98	26831

Table 2.2: Anomaly Data Set of Paging Success Rate data

PSR	**PA**	**PF**	**FTPSR**	**FTPR**
87.14	125475	18511	82.63	118988
86.89	421110	63538	82.77	401158
86.96	568658	85273	82.15	537253
86.96	561601	84193	81.16	530636
87.33	404493	58640	83.14	385095

Tables 2.1 and 2.2 provide the view of normal data and anomaly data in the paging success rate data as per the subject matter expert provided formula and this is applicable to 3G and 4G paging success rate data. Just one important point if the data duration increase then maxPSR need to adjust and the threshold will be tuned automatically.

Chapter 3

Supervised Learning based Mobile Network Anomaly Detection from Key Performance Indicator (KPI) Data

3.1 Research Gap Details

There are two types of machine learning supervised machine learning and unsupervised machine learning.In this chapter it is demonstrated how supervised machine learning is used in mobile network anomaly detection.The base data set is paging success rate data and the key contribution of this chapter is as follows:

1. Mobile network performance data size is huge, and it is measured in different granularity yearly, monthly, hourly and minute levels. But none of the research is available where multiple granular data is used. In this chapter, mobile network performance data is day-level, and hourly-level data and observed how to supervise machine learning performs over those data set.

2. The mobile network runs on absolute precision to ensure the highest network availability. Hence the deviation between anomalous and normal data is massive. Such deviation indicates an imbalance scenario and biases the machine learning outcome. SMOTE is a method used to overcome this scenario which is also a groundbreaking contribution of this chapter.

The following sections of this chapter will have the details of the mobile network and which network performance data is part of the chapter. The next sections have details of the methodology of identifying one supervised learning for anomaly detection model. The next part will have the details of the performance data, its type, and ground truth details. The last section has the complete comparative analysis of all the supervised learning decision tree (DT), random forest (RF), support vector machine (SVM), gaussian naïve Bayes (GNB), and logistic regression (LR). The basis of comparative analysis is F1-Score, confusion matrix, AUC of ROC.

3.2 Supervised Machine learning

Supervised machine learning is a type of machine learning where the ground truth is the pre-requisite. There are various type of supervised learning but in this chapter decision tree (DT), random forest (RF), support vector machine (SVM), gaussian naïve Bayes (GNB), and logistic regression (LR) will be briefly discussed.

3.2.1 Decision Tree

A decision tree [21]is a type of supervised learning which is used to solve classification and regression problem but mostly prominent in the classification problem. As name suggested it is a tree shape architecture where the feature is represented by the internal node, outcome is represented by the leaf nodes and the decision rules are determine by the branches. The main to component of the decision tree is the decision node and leaf node. The decision node which is the trigger point of a branch and leaf node is the end node which does contain a decision but not a branch.The decision tree is like building a tree and it uses Classification and Regression Tree algorithm (CART) to build the tree. The algorithm of decision tree first start from the decision node and comparing the attribute in the decision node, create branches and reach to the leaf node. The brief algorithm is as follows:

1. Begin the tree with the root node which contains the complete dataset.
2. Identify a attribute in the dataset which contain the most meaningful insight using Attribute Selection Measure (ASM).
3. Divide the dataset into subsets that contains possible values for the best attributes.
4. Generate the decision tree node, which contains the best attribute.
5. Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node..

The main advantage of decision tree is that it is very simple to understand and take yes/no decision to move further kind of like a human brain. However it is affected by over-fitting and solved by Random Forest machine learning.

3.2.2 Random Forest

The Random Forest (RF) machine learning is the advance version of decision tree machine learning algorithm. It is based on the concept of ensemble learning where the output is taken from multiple decision tree on the same dataset. As name suggested the RF [1] algorithm which contains multiple decision tree over the subset of same dataset and taken the average to improve the accuracy.The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. However there are some assumption to get the best output of random forest

1. The features selection on the subset of dataset must have some actual value so that RF classifier can predict an actual output rather then a guessed output.

2. The co-relation between the subset of dataset should be as low as possible.

The random forest work of two phases, first it creates a N number of decision tree and second phase it combine the output. Major steps are.

1. Select random data points from the dataset.
2. Build the decision trees associated with the selected data points
3. Choose the number N for decision trees that you want to build.
4. Repeat Step 1 and 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes

The main two advantages of RF algorithm is it is capable of handling large datasets with high dimensionality and enhances the accuracy of the model and prevents the overfitting issue.

3.2.3 Support Vector Machine

Support vector machine(SVM) is a type of supervised machine learning which is also popular in the classification problem. The main objective of SVM [2] is to create the best decision boundary in the data set into classes so that any new data can also be classified easily. The term for the decision boundary is known as hyperplane. As name suggested SVM used points/vectors to create the hyperplane. SVM is widely used for face detection, image detection or text categorization. There are two type of SVM

1. Linear SVM: If the data set is completely classified with a straight line then it is called the linear SVM.
2. Non-Linear SVM: If there is non-linearity in the dataset then the dataset is not classified by a straight line. For such cased non-linear SVM is used.

There are a set of tuning parameter which will determine the best hyperplane. Those are as follows,
Regularization: This parameter signifies how much you want to misclassify the training data. For large values regularization the optimization will choose a smaller-margin hyperplane and for low regularization value it will consider larger-margin separating hyperplane.
Gamma: The parameter gama signifies the with how many feature dataset the hyperplane is created. If gamma value is low it takes the far data and if the gamm value is high it only take the closer data set to build a hyperplane.
Margin: A good margin is one where this separation is larger for both the classes.

3.2.4 Gaussian Naive Bayes

Gaussian Naive Bayes [24] is a supervised machine learning which is based on bayes theorem. It is mainly a probabilistic classifier because it predict on the basis of probability of the object. There are two generic terms used in the algorithm Naive and Bayes.
Naive: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.
Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.
Gaussian Naive Bayes is basically meant for binary or multi-class classification. For cases when you have a majority class and a minority class, the prior probabilities of the majority class will most definitely dominate the minority class (for e.g. 0.99 ¿¿ 0.01) and thus most of the time, all the data points may be classified as member of majority class. The characteristics of Naive Bayes classifier is as follows:

1. The Naive Bayes method makes the assumption that the predictors contribute equally and independently to selecting the output class.

2. Although the Naive Bayes model's assumption that all predictors are independent of one another is unfeasible in real-world circumstances, this assumption produces a satisfactory outcome in the majority of instances.

3. Naive Bayes is often used for text categorization since the dimension of the data is frequent rather large.

3.2.5 Logistic Regression

From the categorical dependent variable the logistic regression(LR) predict the output. The output of logistic regression [3] is discrete value and lies between 0 to 1. Logistic regression is a type of supervised learning which is similar to linear regression but instead of regression analysis it is used to solve classification problem. In the logistic regression a 'S' shaped logistic function is used which predict max values between 0 and 1. The logistic regression has the ability to provide probabilities and classify new data using continuous and discrete data set.
The main parameter of logistic regression is the logistic function and the assumptions is as below:

1. In the logistic regression the logistic function is knows a sigmoid function and it converts the real value within the range of 0 and 1.

2. In the logistic regression a threshold concept is used. Anything beyond threshold denoted as 1 and below the threshold will be denoted as 0.

Last but not the least the dependent variable must be categorical in nature and independent variable should not have multi-correlation.

3.3 Methodology of Supervised Machine learning anomaly detection model

In this chapter, supervised learning performs over the performance data set. There is two part of the overall work. The first part is to prepare a data set with human

intuition, implying data normalization and re-sampling wherever applicable. The second one is imposing several supervised machine learning over the data set and performing a comparative analysis.

3.3.1 Data Preparation

At the initial stage, KPI data consider for 30 days. Then data set's ground truth is determined by a subject matter expert of performance management of the mobile network. After finishing the ground truth determination, a filter imposes over the overall data set. The filter criteria are to select the network objects which has the ground truth anomaly count of more than five Fig 3.1. In the Day-level data, the number of network objects is six, and for an hourly level, the number of network objects is five. Once this initial data set is ready, A min-max data normalization exercises on both types of data set. This normalization converts the data to the same scale for efficient machine learning outcomes. Also, there is another work of analyzing the imbalance scenario, and if it's found, SMOTE implies to make the data set the balance for the next course of action. In the end, two types of data will be ready for both day and hourly level data. Those are normalized data, and another is re-sample data.

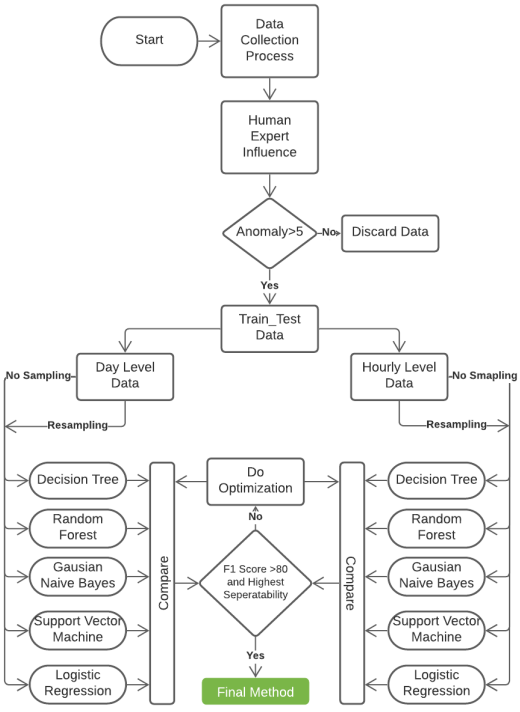

Figure 3.1: End to End Methodology

3.3.2 Model Implementation and Validation

In both normalized and re-sampled day level and hourly level data, all the supervised learnings are implemented one by one shown in Fig 3.1. After finishing the implementation accuracy, precision, recall, and F1-Score are saved. Then a ranking is done based on all supervised machine learning F1-Score. A slightly different

ranking is done based on the false-negative (FN) value that resides in the confusion matrix. Another validation step that is followed in this chapter is AUC score raking. As it demonstrated the area under the curve and receiver operator characteristic plot of machine learning means the model performance in terms of separability. After combining all ranking results of F1-Score and AUC, supervised learning is recommended for anomaly detection of mobile network performance data.

3.4 Data Details and Feature Description

The paging success rate KPI is the performance data set of this chapter. This KPI measures between network objects which are MSC and BSC or RNC. It denotes when an MSC declares a mobile is in the coverage area, pages the mobile, and connects to the mobile trying to connect. There are associated counters and KPIs, which are tightly coupled with paging success rate KPI considered as key features of the data set. The performance data is calculated in different aggregation levels yearly, monthly, daily, or hourly levels. The daily and hourly level paging success rate is used in this chapter. The total duration of the data set is one month. A subset of seven days data of day level data is shown in Fig 3.2. A day-level data is an average aggregation of hourly-level data. However, such average aggregation removes most of the data variance from the data set. From Fig 3.2 only a few of the outliers are identified to network objects. Due to this, hourly level data is also considered for anomaly detection model analysis.

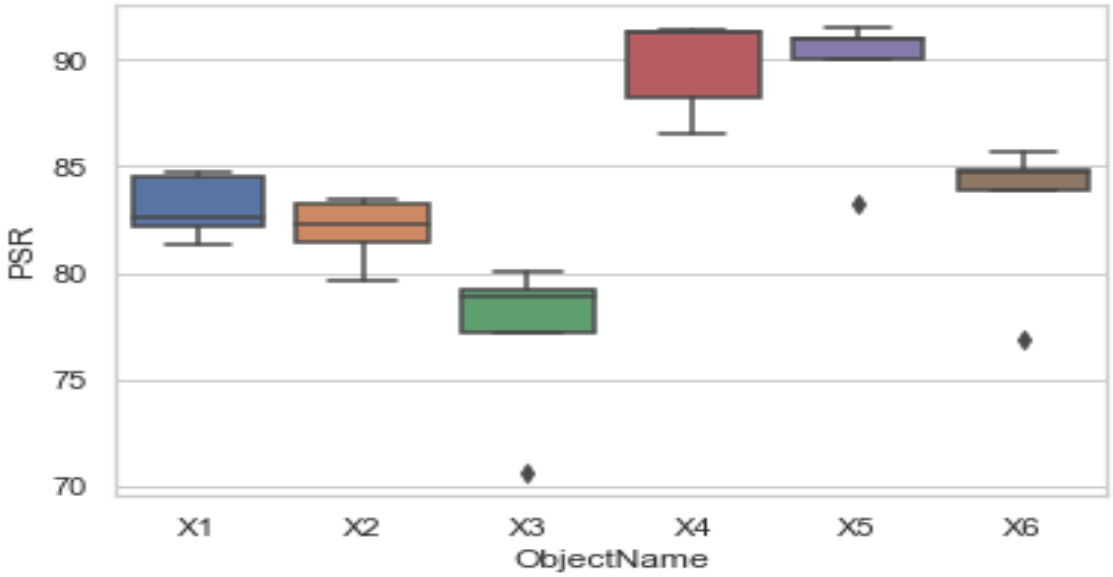

Figure 3.2: Outlier on A Subset of Day Level Data

Fig 3.4 it is showing a subset of hourly level data of 24 hours. The hourly level data is very much insightful. It reports the paging success rate data every hour. If anomaly detection is done on an hourly basis, mobile operators will have much more time to detect those anomalies. The number of outlier to network objects is also significant showing in Fig 3.4.Both day level and hourly level sample shown in the above figures have a data point, but it does not say which point is anomalous and which are not. As supervised machine learning requires ground truth for the

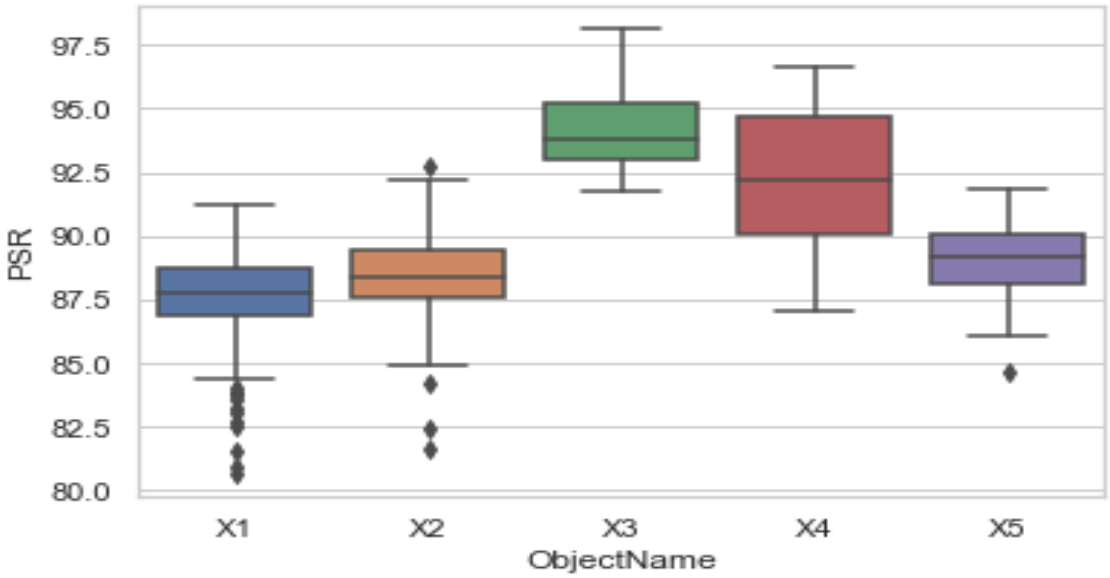

Figure 3.3: Outlier on A Subset of Hourly Level Data

baseline anomaly detection model, the whole data set is leveled by a subject matter expert. This human influence helped to get a leveled and base data set ready for applying multiple supervised learning for anomaly detection.

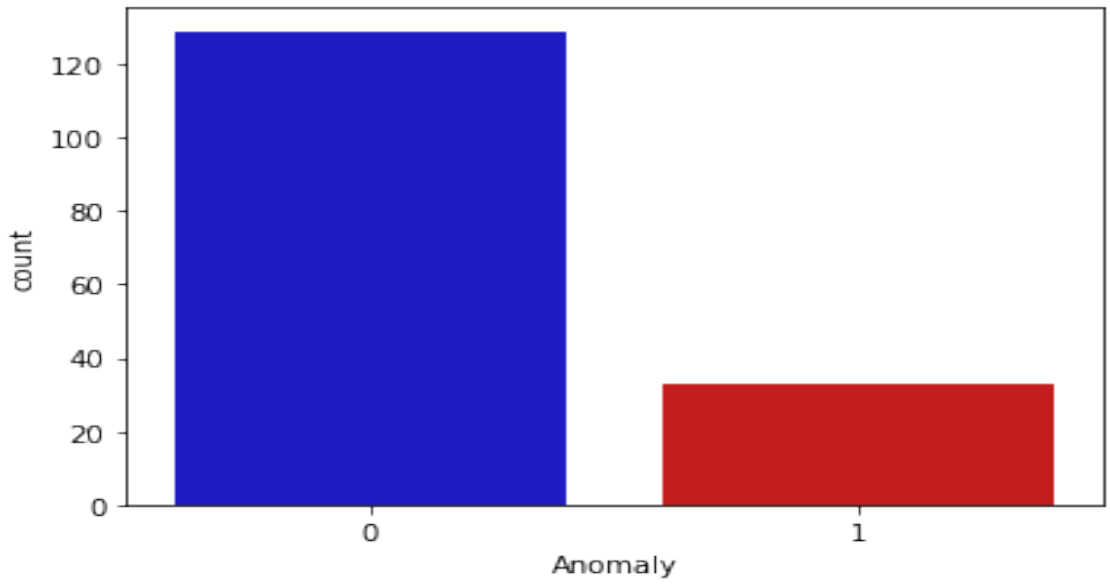

Figure 3.4: Imbalance Ratio of Day Level Data

Now, look on to the imbalance ratio of both the data set of day-level and hourly-level. This ratio is determined to the ground truth anomalous and not anomalous data. In the day-level data, the ratio is very high. In Fig 3.4 it is showing 80 percent data is not and 20 percent data is anomalous. To improve it, SMOTE applies for better anomaly detection outcomes. But the hourly level data is more balanced. From Fig 3.5 the imbalance ratio is 62 percent of not anomalous data and 42 percent is the anomalous data. This is quite a good ratio of imbalance however, still SMOTE is implied to observe the performance scenario of anomaly detection.

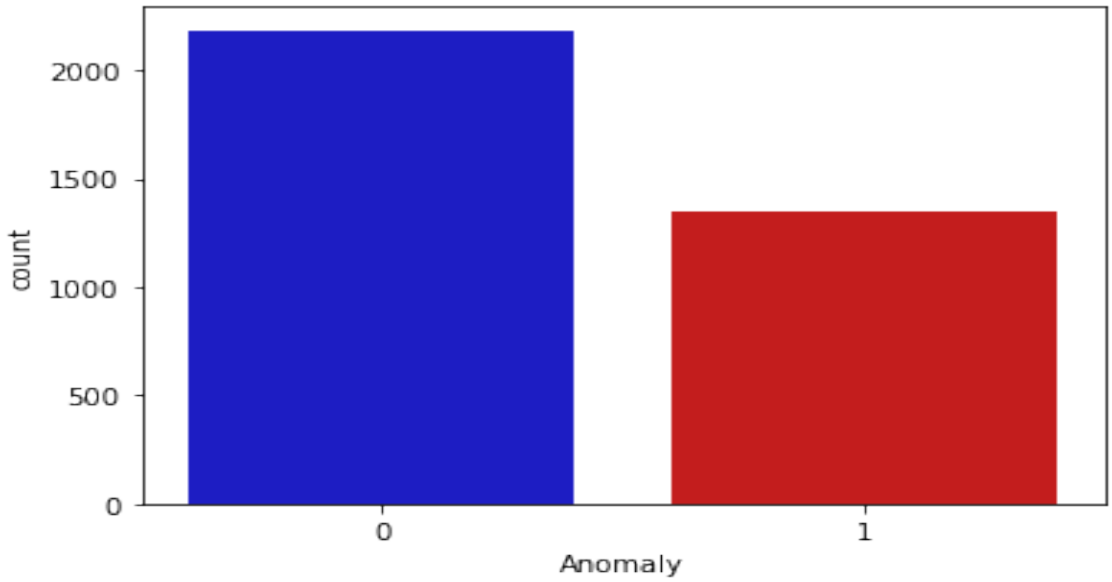

Figure 3.5: Imbalance Ratio of Hourly Level Data

3.5 Result Discussion of Supervised Machine Anomaly Detection

The performance measurement of a machine learning algorithm is not straightforward. Let us consider a set of data where it is depicting that 98 percent of data denote no anomaly. Now if machine learning is built to produce a result of no anomaly state. In that case, the machine learning algorithm will have 98 percent of accuracy. But the problem remains for the anomalous data which is detected by the algorithm. Hence accuracy is not the exact matrix for evaluating supervised machine learning algorithms. The most popular and practiced one's are confusion matrix, precision, recall, F1-score, or AUC for evaluation.

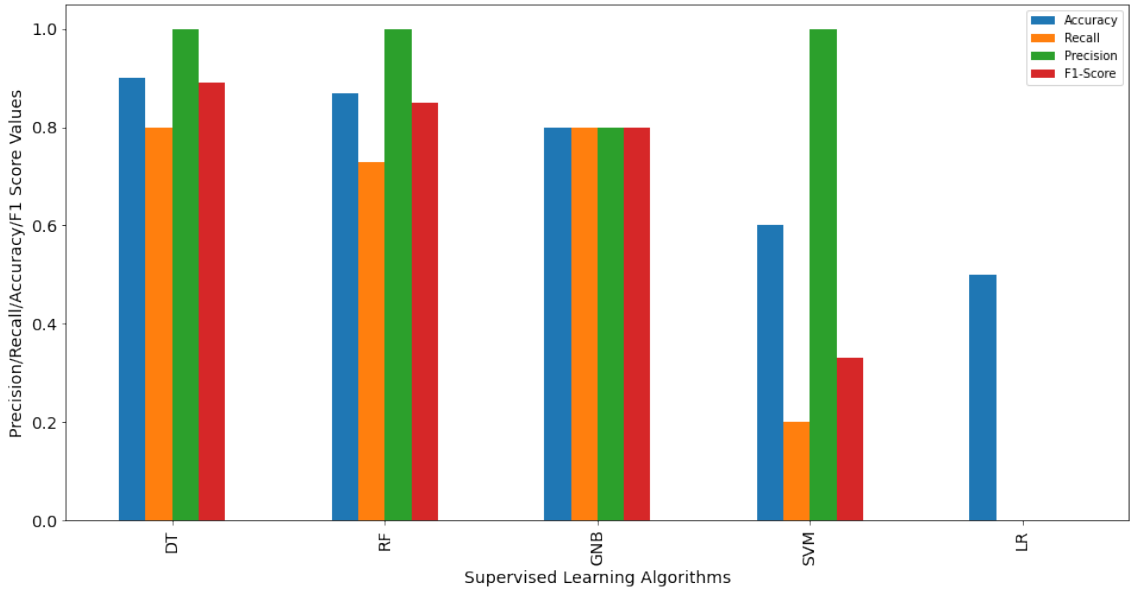

Figure 3.6: F1-Score Ranking for Day Level Data

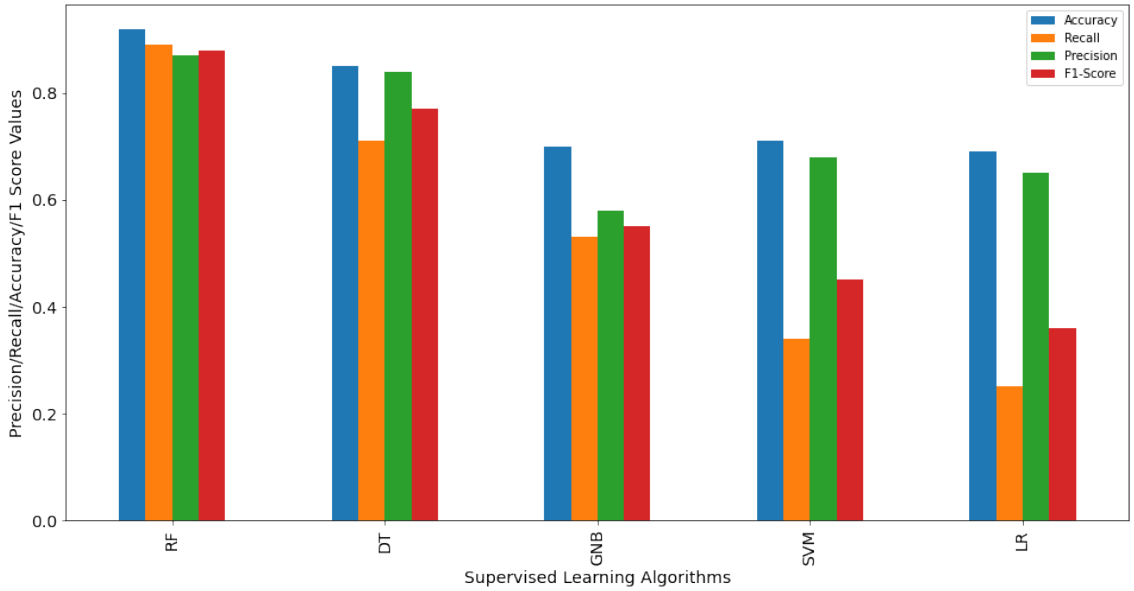

Figure 3.7: F1-Score Ranking for Hourly Level Data

Fig 3.6 is showing that after applying the proposed methodology over day level data, most of the supervised learning has outperformed in terms of F1-Score. Furthermore, after applying the proposed methodology over hourly level data, only the random forest (RF) algorithm has met the criteria, further is in Fig 3.7.

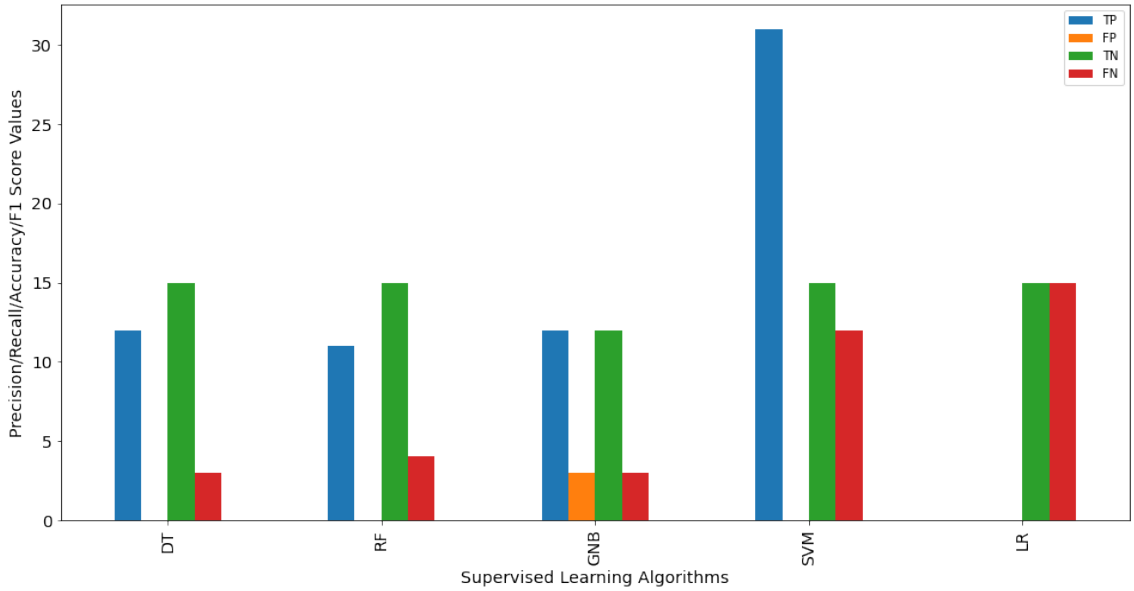

Figure 3.8: Confusion Matrix of Day Level Data

Now let's look into more detail of the confusion matrix that is the basis of precision, recall, and F1-Score of the machine learning algorithm. Among all the parameters of the confusion matrix, FN is a parameter that denotes the anomalies which are an anomaly but predicted as not an anomaly. This parameter is important because it signifies anomalies that are predicted as not anomalies by the machine learning algorithm.
Fig 3.8 and Fig 3.9 have the details of false-negative for all the machine learning algorithms of both data types day level and hourly level. While looking at the table

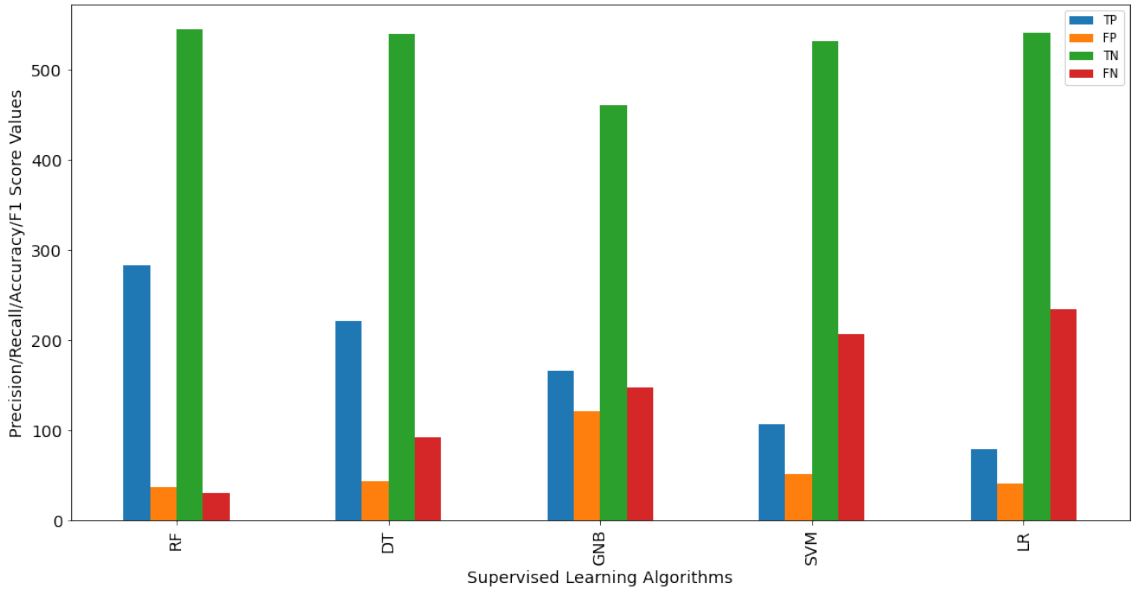

Figure 3.9: Confusion Matrix of Hourly Level Data

random forest has the lowest FN value 30 among others however, in day-level data decision tree has the lowest FN value of 3.

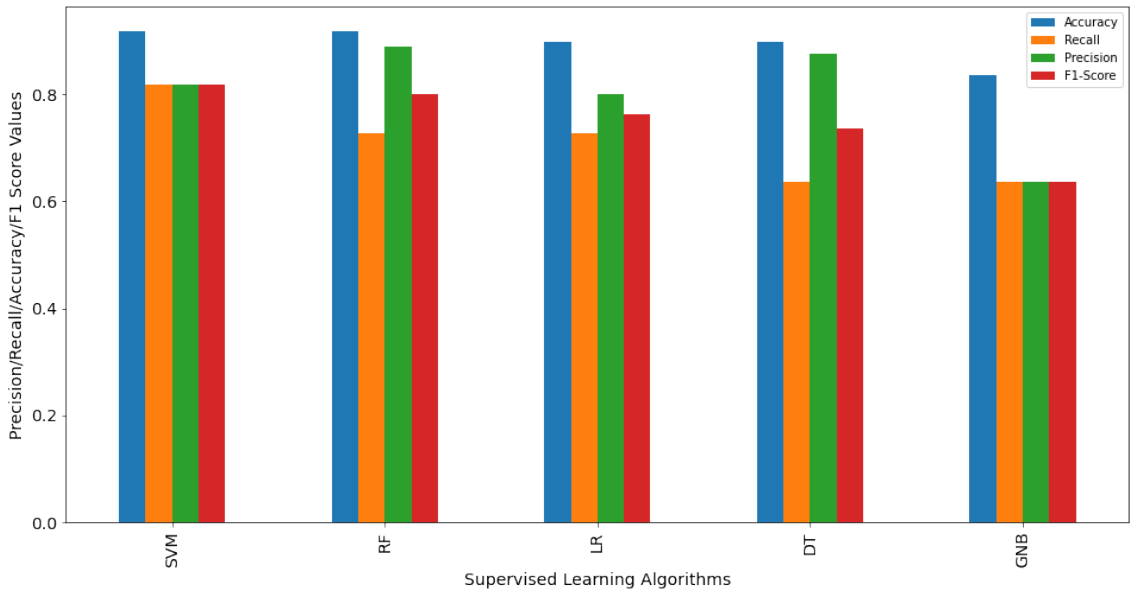

Figure 3.10: SMOTE Implication on Day Level Data

As explained earlier, SMOTE implication is observed over the data set, and during this implication, F1-Score analysis will be discussed. In Fig 3.10 SVM has performed better in the day level data whereas in normalized data SVM has not performed well. Here is an important outcome is drawn, data should be properly balanced so that right supervised model is identified for the network anomaly detection model. On the other hand, though hourly level data's imbalance ratio is quite ok, however SMOTE is implied over the hourly level data. It is shown in Fig 3.11 where RF also being superior on the overall data scenario.
There is another type of evaluation which is called AUC. AUC denotes the area

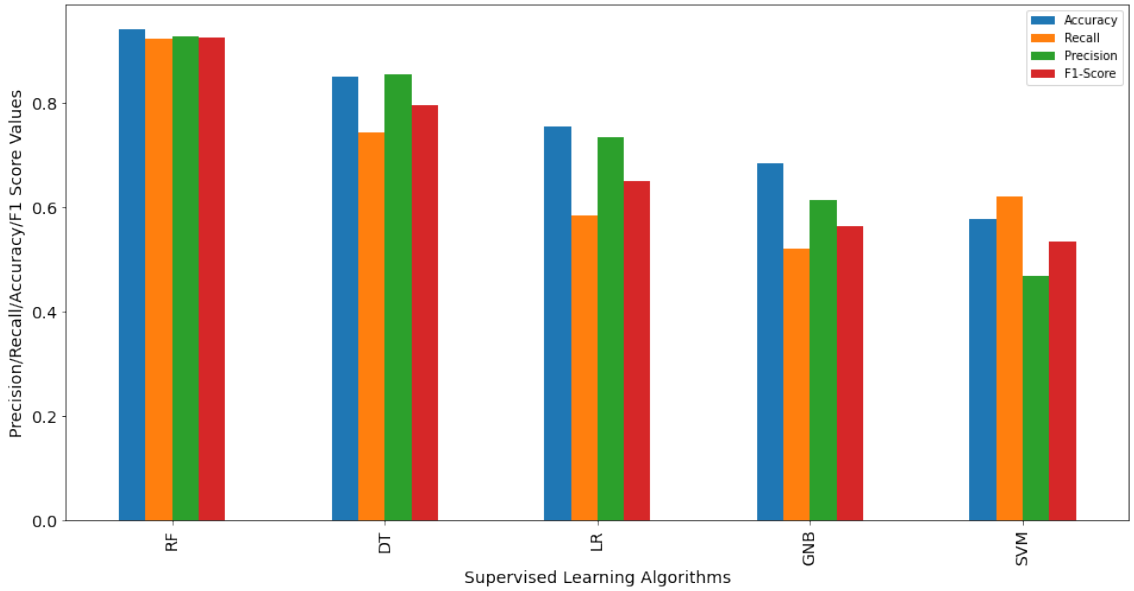

Figure 3.11: SMOTE Implication on Hourly Level Data

under the curve, meaning whether the model classifies 0 as 0 and 1 as 1. It's also called separability.

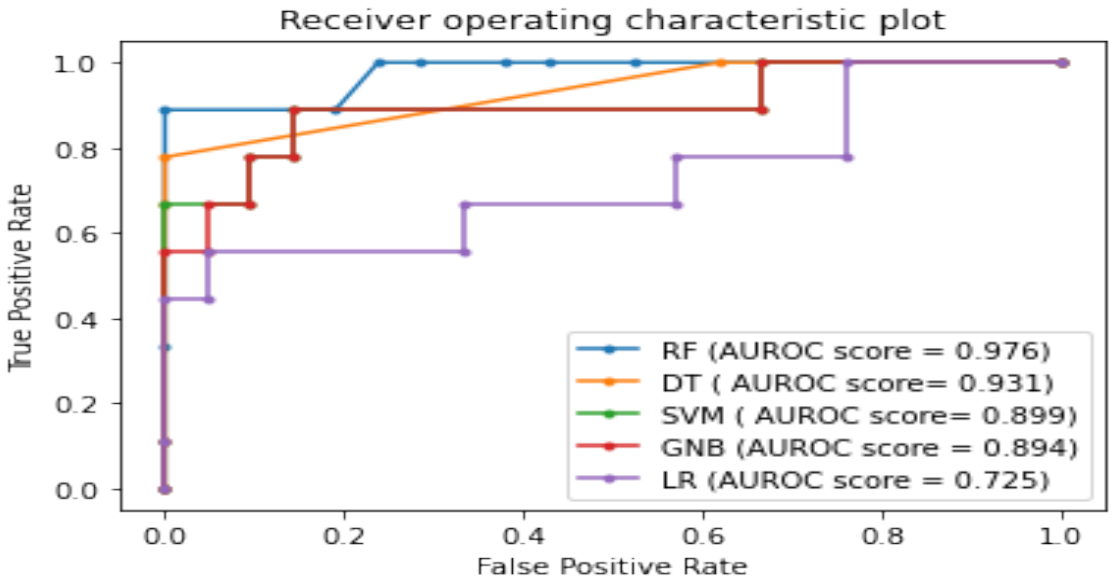

Figure 3.12: Day level Receiver Operating Characteristics Curve

The higher the AUC value is dignified better is the model. In Fig 3.12 and Fig 3.13, the AUC score for random forest (RF) is close to 1. The value is 0.974 for day-level data and 0.973 for hourly-level data. That seems promising and means the RF algorithm has good separability in detecting an anomaly. The same is shown in the plot, both day-level, and hourly-level date set. In the overall analysis, the evaluation criteria are F1-Score and AUC scores.
In conclusion for any type of paging success rate data, random forest (RF) has outperformed among all the supervised learning. And the ground truth data is imbalanced in such case SMOTE is recommended to apply and a support vector

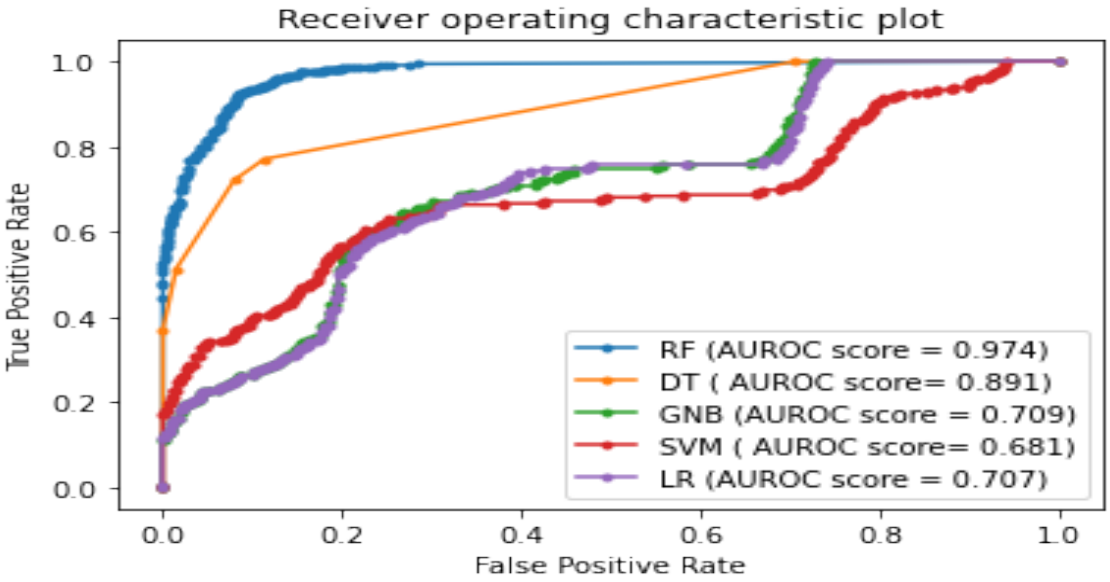

Figure 3.13: Hourly Level Receiver Operating Characteristics Curve

machine (SVM) is the recommended one to use in the anomaly detection model.

Chapter 4

Benchmarking Unsupervised Machine Learning for Mobile Network Anomaly Detection

4.1 Research Gap Details

The anomaly detection (also referred to as outlier detection and sometimes as novelty detection) is generally understood to be the identification of rare items, events or observations which deviate significantly from the majority of the data and do not conform to a well defined notion of normal behaviour. The main focus of this chapter is lying on unsupervised learning and anomaly detection models. The contribution of this chapter is as follows:

1. It briefly describes the mobile network data set, identifies the efficient normalization step that makes the data set ready for exercising unsupervised machine learning based anomaly detection.

2. Multiple unsupervised learning K-means, DBSCAN, and HDBSCAN implications will observe over the mobile network data set. During this implication, necessary parameter tuning of unsupervised machine learning algorithms will exercise.

3. In terms of benchmarking unsupervised machine learning, a series of validation will imply. There is usual validation in terms of accuracy, precision, recall, and F1-Score. However, the data set is fragmenting in different levels such as monthly, weekly, anomalous data. But the most important one is on the data set chosen by a mobile network subject matter expert.

4. After the series of validation, one unsupervised machine learning is recommended for mobile network performance anomaly detection.

In this chapter, three unsupervised machine learning will be used and their basic features and work step will be described in the unsupervised machine learning section. In the later section benchmark step for the anomaly detection model will be discussed and the following section data description and the type of normalization will be elaborated. In the final section of the result discussion, the overall research outcome will be shown and how benchmarking of different unsupervised machine learning is achieved by different levels of validation approach.

4.2 Unsupervised Machine Learning

Unsupervised machine learning is a type of machine learning in which does have a pre-requisite of label information. This converts the overall data into multiple clusters based on the data characteristics and by unsupervised machine learning model parameters. In this chapter three unsupervised machine learning will be discussed, those are K-means, DBSCAN, and HDBSCAN. This clustering algorithm has its model parameter to cluster the data.

4.2.1 K-means Clustering

The K-means is the most interactive unsupervised algorithm. It works by determining a random k centroid at first in the data set and starting updating the centroids. This update is continuing until the total cluster scenario reaches equilibrium. As a distance measurement within the data set using euclidean distance [19]. The basic algorithm will be as follows for an N number of features within the data point:

1. Randomly pick 'k' value as a centroid in the data set
2. Assign each of the values of the feature to the nearest centroids, measured by euclidean distance. 'k' clusters are determined
3. Estimate the centroids in 'k' cluster. as the mean of feature, value is assigned to the cluster
4. Go back to step 2 if convergence is not achieved with the condition new cluster is significantly different from the previous cluster

The K-means clustering is completely an iterative approach and works until overall cluster convergence is achieved. Dynamic anomaly case [11] K-means is used however in this chapter that cluster will be considered as an anomaly that has to lowest observation.

4.2.2 DBSCAN Clustering

The DBSCAN clustering stands for Density-based Spatial Clustering of Applications with Noise. This unsupervised machine learning works with data density, clustering, and noise. The detected noise is categorized as an anomaly. Based on the density of the data set DBSCAN creates the clusters. Data setpoints are categorized into three, core points, border points, and noise [10]. A core point is a point determined by the conditions from where a cluster meets the criteria of minimum samples. The boundary points are measured by two conditions, one is the number of neighbor data points of a boundary point must be less than minimum samples and the boundary point should be a neighborhood to a core point. Lastly, noise points are the points that are neither core points nor boundary points. The simplest DBSCAN algorithm is as follows:

1. Classify all the data points as per DBSCAN model parameter
2. Determine noise and categorize into an anomaly.

3. Assign cluster to the core points.

There is another aspect, the DBSCAN algorithm is highly sensitive to its parameter. Two important parameters determine the outcome of the DBSCAN algorithm. One is epsilon and another is min samples. The epsilon is the measure of the neighborhood and min samples stands for the number of minimal sample points within a cluster. In the DBSCAN algorithm, optimal epsilon value identification is mandatory. The optimal value of epsilon is measured from a plot where is y-axis has the distance and the x-axis represents an array of i where all the data points reside. If everything works fine, then an elbow will be shown in the graph. The elbow point corresponding to the y axis value is the optimal epsilon value.

4.2.3 HDBSCAN Clustering

The HDBSCAN is an extended version of DBSCAN clustering. It is a hierarchical clustering approach rather than a flat one[28]. It converts the DBSCAN clustering to a hierarchical one and does a flat clustering to find the stability of the cluster[26]. It works on some defined steps and is supported by multiple distance metrics. Key steps are:

1. Density-based space transforming.
2. Using distance metrics construct a minimum spanning tree.
3. Using connected component build cluster hierarchy
4. Condense the cluster hierarchy based on minimum cluster size.
5. Find and extract stable cluster

The identification of anomalies in the data is like DBSCAN. Here the noise identified by HDBSCAN will be considered as an anomaly.

4.3 Methodology of Unsupervised Machine learning based Anomaly Detection

Performance data is a combination of multiple counters generated from network elements. Based on the formula, the counter is converted to KPI. A KPI deterioration means there is an issue in the mobile network and customer experience will be affected. Hence KPI's needs to monitor cautiously. Any anomaly of those KPI's needs immediate reporting. In this chapter, a couple of unsupervised machine learning will exercise on performances data and it will be benchmarked by converting it into a classification problem. Two major steps are described below:

4.3.1 Unsupervised Machine Learning Implication

The total period of the data set is a month and data granularity is hourly level data. At first, the raw data is collected, there is a sequence of data normalization will be implied. First is min-max scaling, then PCA analysis, and last is standard

scaling. In the next step, unsupervised machine learnings will be implied. During this implication, different parameter tuning will be performed on the unsupervised machine learning model demonstrated in Fig 4.1.

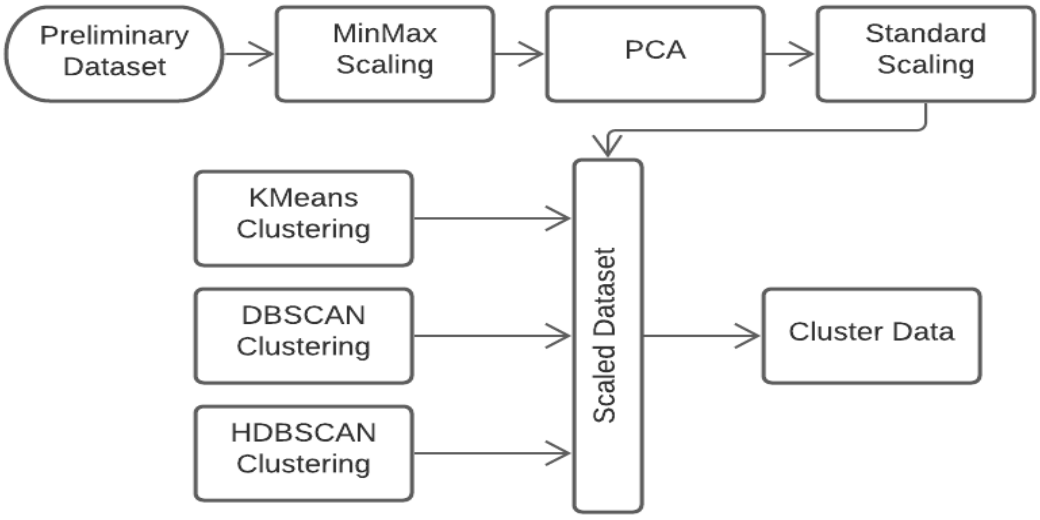

Figure 4.1: Normalized Data to Cluster Data

Now it is required to measure the model performance of the clustered data. To do that, a conversion is imposed so that clustered data change into a classification problem. That is dependent on the properties of the unsupervised machine learning algorithm. For example, in DBSCAN and HDBSCAN, the cluster label of '-1' will be considered an anomaly and further labeled to '1'. The rest considered as '0' means no anomaly. On the other hand, for K-means bottom 5 cluster has been chosen as an anomaly which has the lowest observation

4.3.2 Validation

There are a couple of ways to measure the model performance of unsupervised machine learning. The most used one is to covert the cluster data into a classification and then compare it with the ground truth. The performance data set used in this chapter has its ground truth prepared. The F1-Score, recall, precision, and accuracy are measured in the usual way.
To further enhance the validation and be in line with benchmark outcome, monthly data is categorized to a different level at Fig 4.2.At first unsupervised learning will be implied over the hourly data set of a month. Then the monthly data set will be divided into four weeks data sets. Those named T1, T2, T3, and T4. Individual validation of those data set will perform and then benchmarked. Thirdly as ground truth data is already in hand hence next validation of the cluster will be performed only for the data sets which are anomalous in the ground truth. Over this data set unsupervised machine learning will be implemented and benchmarked accordingly. Lastly, the overall unsupervised learning outcome will be cross-validated by a subject matter expert.

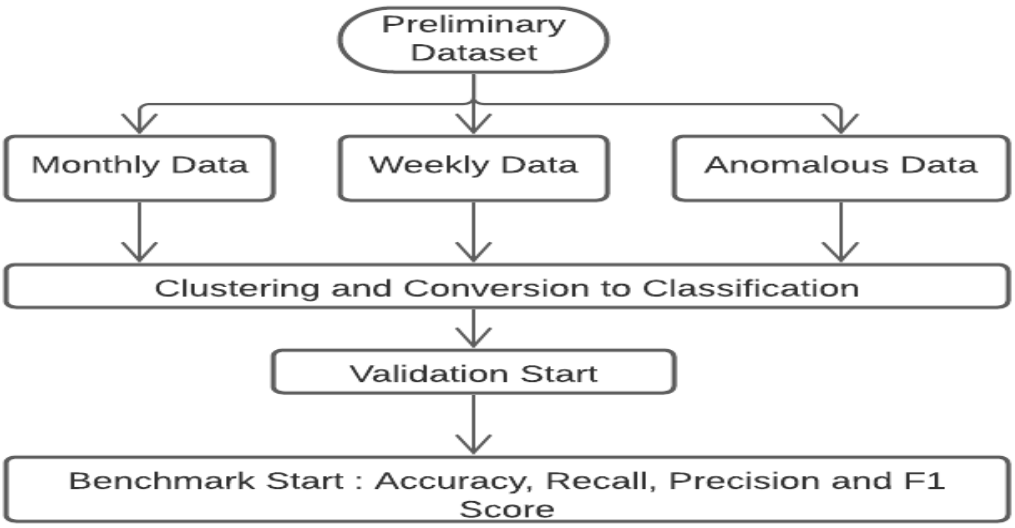

Figure 4.2: Validation Workflow

4.4 Data Normalization Details

It is clearly understood, each of the features under the data set has lots of information. If unsupervised machine learning is implemented over raw data, it might not provide the expected outcome. This is due to the data set is not appropriately normalized. To do that some normalization is recommended to implement before applying unsupervised machine learning.

4.4.1 Normalization

Machine learning tends to find a pattern within the data sets by correlating multiple features. However, the issue arises when there is a drastic scale change of the features. In such case, features are required to be under the same scale to have a better machine learning outcome. That is called scaling. Min-max scaling is one of the scaling techniques which will scale the value of the feature between 0 and 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{4.1}$$

This method takes the maximum and minimum feature value of the data set, after that scale the feature value as per the above equation.

4.4.2 Standardization

Standardization is converting a feature value by subtracting from its mean and then scaling following standard deviation. This type of scaling is known as a unit variance. The basis is, this will transform the features in such a way so that the mean value will be 0 and the standard deviation of 1. Standardization is recommended to be implemented over the data set which is zero-centric data.

$$x' = \frac{(x - \overline{x})}{\sigma} \tag{4.2}$$

4.4.3 Principle Component Analysis

In machine learning, principal component analysis (PCA)reduces the dimension of data. The PCA creates a principal axis on data and keeps rotating the axis. Each axis denotes a different co-variance. The first principle component corresponds to the highest number of variances. From the below equation of feature x,y,z the PC1, and PC2 are determined. The PC1 is a linear combination to determine the magnitude and the direction of the maximum variance in the data set. The PC2 has less variance and is not co-related with PC1.

$$PC1 = w_{1,1}(x) + w_{2,1}(y) + \cdots + w_{n,1}(z) \tag{4.3}$$

$$PC2 = w_{1,2}(x) + w_{2,2}(y) + \cdots + w_{n,2}(z) \tag{4.4}$$

If PCA is implemented on the data set, it reduces the features number. This statistical model decides as like as a human though it has limitations, it is very much affected by the outlier. Hereafter, it is recommended to use normalization on the data set and then apply PCA. The explained variance also looks prominent, which means it captured the maximum variance. For component 2 the explained variance value is 0.992908.

4.5 Result Discussion

As per the validation step, the unsupervised machine learning outcome has been converted to a classification problem. Then model performance management matrix is calculated for accuracy, precision, recall, and F1-Score. Later benchmarking is completed comparing model performances using K-means, DBSCAN, and HDBSCAN unsupervised machine learning outcome First approach is to see the unsupervised learning implication on a period of one-month hourly level data The key parameter are as below and is applicable for overall benchmarking.

- In the K-means clustering approach, 5 lowest observation cluster is chosen as anomaly.
- The DBSCAN clustering approach parameters are, eplison value=0.09 and minsamples=5. The noise will be considered as anomaly.
- The HDBSCAN clustering distance metrics is 'manhattan' for noise identification and will further categorize into an anomaly.

In Fig 4.3, it is showing the HDBSCAN model has outperformed all other unsupervised machine learning, however DBSCAN F1-Score 0.49 which is also close to HDBSCAN. To identify one unsupervised machine learning for mobile network anomaly detection some additional validation approach has been considered.The next approach is to categorize a month's network performance data into four weeks denoted as T1, T2, T3, and T4. Each of those individual data set has enough anomaly data points and legitimate data points. Unsupervised machine learning's then imposed. However, in this approach, DBSCAN and HDBSCAN performed almost similarly for T1, T2, T3, T4 which does not help select a single unsupervised machine learning. Details are in Fig 4.4.

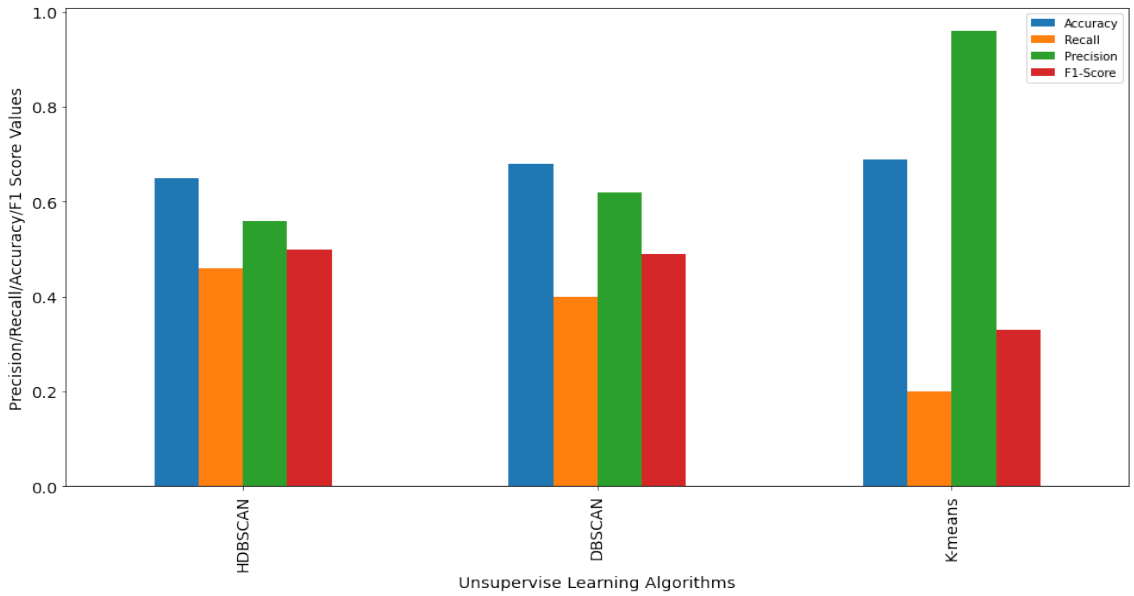

Figure 4.3: Benchmarking of Monthly Level

Above two approach does not give the clear indication whether DBSCAN or HDBSCAN to use in the mobile network anomaly detection model. Hence another approach has taken. As described in validation section, ground truth is already known for the dataset used in this chapter. A separate data set is prepared where only anomalous data is present.

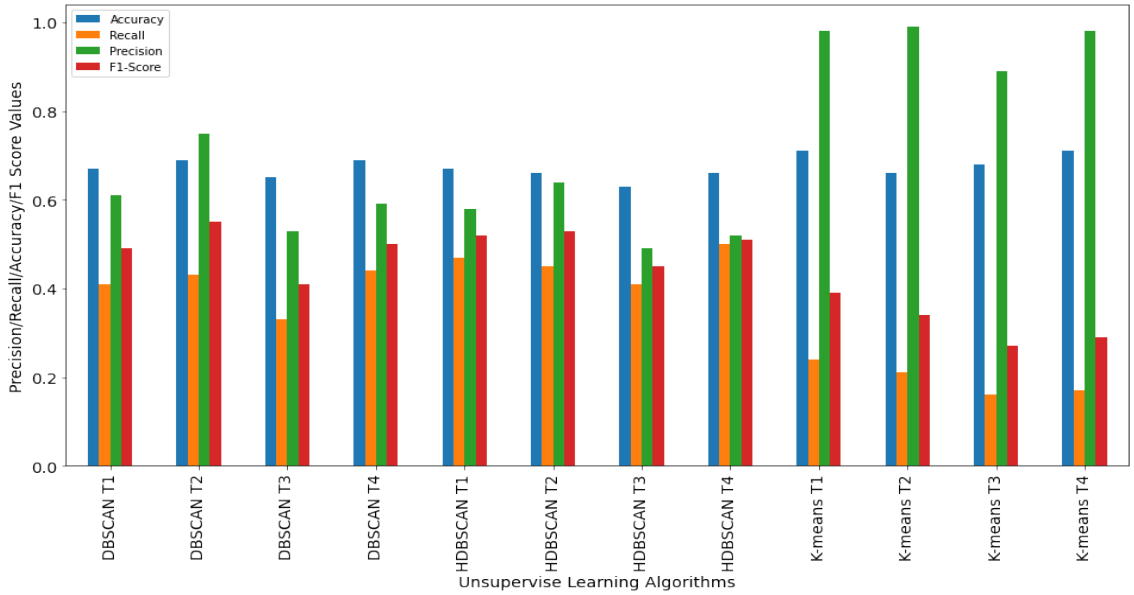

Figure 4.4: Benchmarking of Weekly Level

From Fig 4.5 during F1-Score analysis, it is observed ultimately on anomalous data set HDBSCAN has outperformed all other clustering algorithms. The F1-Score is quite impressive 0.63 whereas DBSCAN is 0.57 and K-means is 0.33. Nevertheless, the unsupervised machine learning outcome is not straightforward. To make it more effective as subject matter expertise has been considered and as per the guideline another anomaly data set has been prepared where the paging success rate value

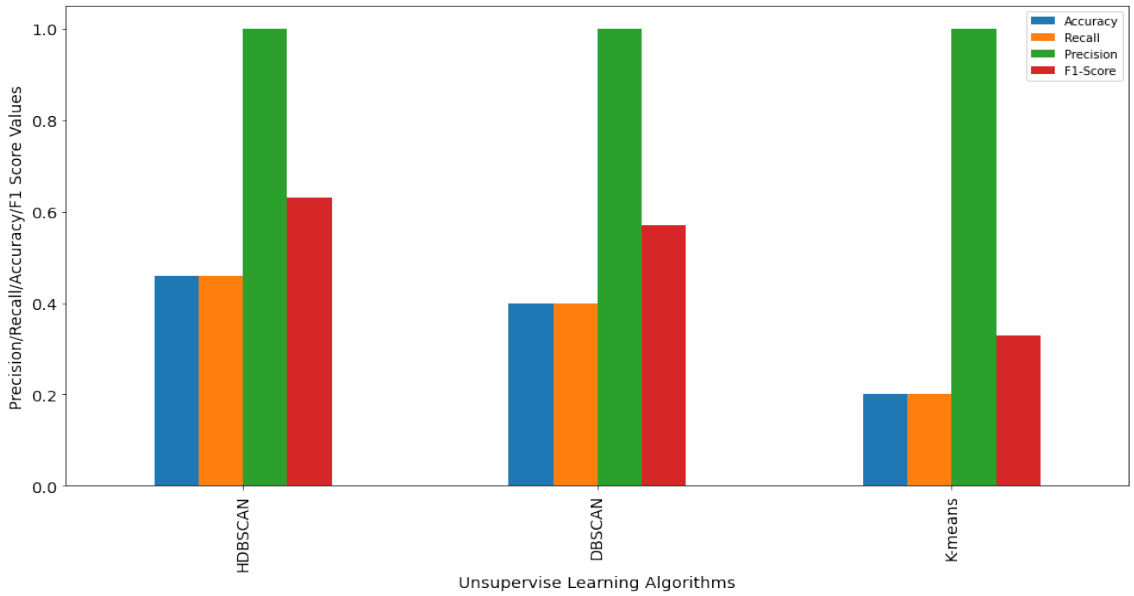

Figure 4.5: Benchmark of Anomalous Data Level

relies on between 85.00 to 95.00.

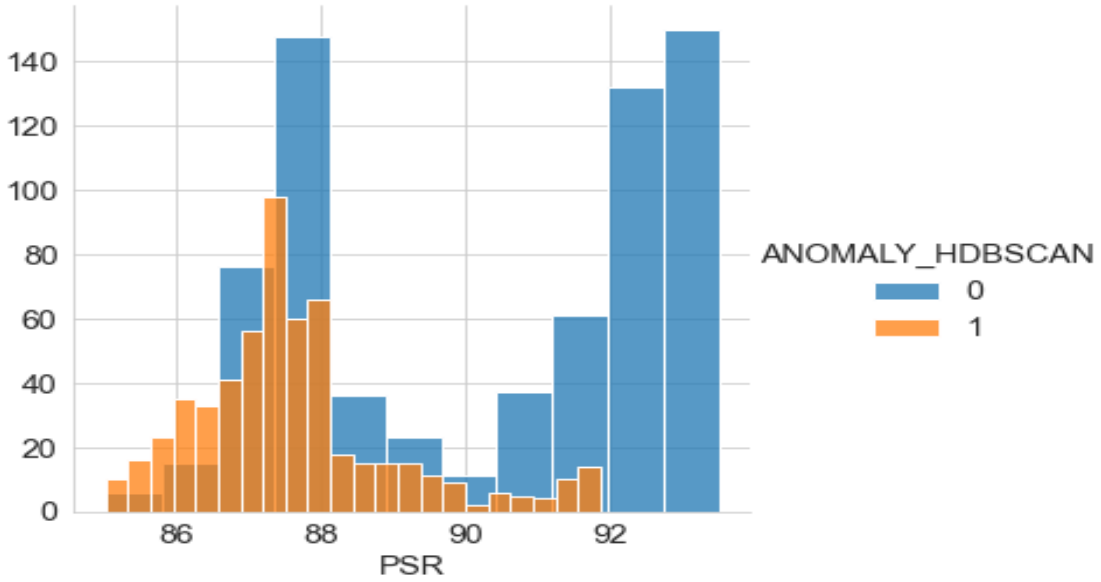

Figure 4.6: HDBSCAN Anomaly Detection

From Fig 4.5 during F1-Score analysis, it is observed ultimately on anomalous data set HDBSCAN has outperformed all other clustering algorithms. The F1-Score is quite impressive 0.63 whereas DBSCAN is 0.57 and K-means is 0.33. Nevertheless, the unsupervised machine learning outcome is not straightforward. To make it more effective as subject matter expertise has been considered and as per the guideline another anomaly data set has been prepared where the paging success rate value relies on between 85.00 to 95.00.
Over that data set, another filter has been proposed to consider only the anomalous data. This data set count is around 1257. Now it is observed that 45 percent anomaly can detect by HDBSCAN whereas only 36 percent can detect by the DB-

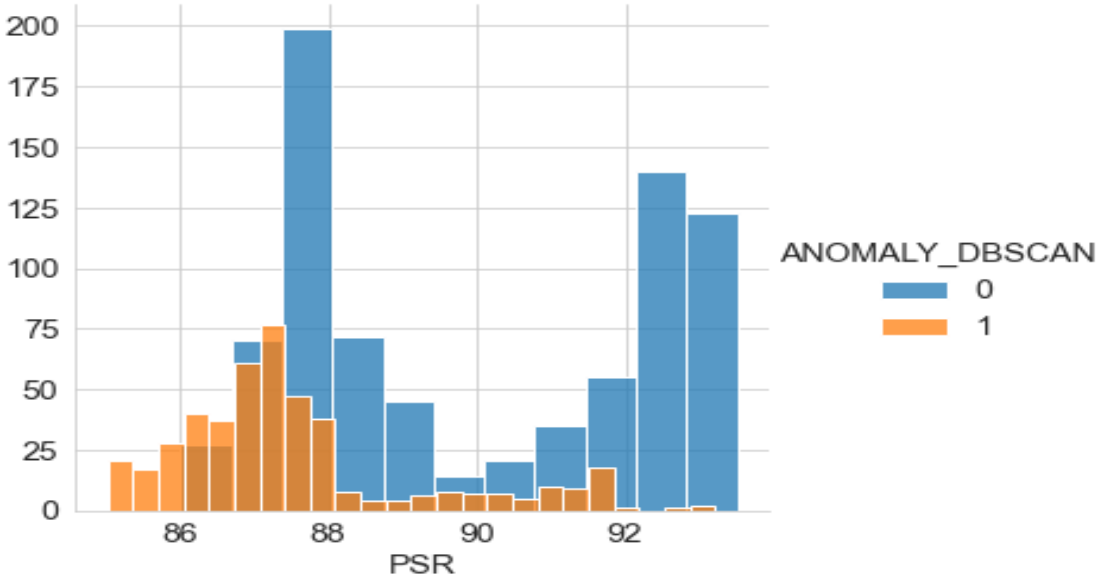

Figure 4.7: DBSCAN Anomaly Detection

SCAN algorithm. This demonstration of filter data made subject matter experts certain about HDBSCAN is best suited for network anomaly detection. Details is in Fig 4.6 and Fig 4.7.Now from F1-Score and subject matter expert analysis, it is evident that unsupervised machine learning is useful for mobile network anomaly detection. Among three of the unsupervised learning used in this chapter, hierarchical clustering HDBSCAN outperformed all others and is recommended for mobile network performance anomaly detection.

Chapter 5

Deep learning Autoencoder based Anomaly Detection Model on 4G Network Performance Data

5.1 Research Gap Details

It is evident from the above chapters anomaly detection is a prominent use case in a mobile network. However, in an earlier chapter, it is discussed how unsupervised learning is helping for mobile network anomaly detection. In this chapter, the focus will on neural-network-based autoencoders and how it is helping with anomaly detection. The key contribution of this chapter is as follows:

1. There is no notable research on anomaly detection that has found that work on S1-MME PS Paging success rate in hourly level data.

2. The 4G network performance data possess the same data inheritance and neural network-based autoencoder is efficient when data details are the same. This area is not explored in the current research.

5.2 Neural Network Autoencoder

Autoencoder is a type of neural network that tends to mimic output from input. The thump rule is followed by the autoencoder which compresses the input into latent space and reconstructs the output. Autoencoder falls under the category of unsupervised machine learning also known as feature extraction algorithm. Autoencoders' input is diverse but not limited to text, speech, image, or video. Let's look at some of the basic properties of autoencoder:

1. Autoencoder is an input-dependent machine learning method. Like it is only able to compress and reconstruct the data on which it has been trained. For example, an autoencoder is trained to reconstruct an image of a tree however the same autoencoder is not able to reconstruct the image of an animal.

2. Autoencoders has lossy compression means that the decompressed outputs are degraded compared to the original inputs.

3. Autoencoders are learned automatically from data examples, which is a useful property: it means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input. It doesn't require any new engineering, just appropriate training data.

The basic architecture of the neural network-based autoencoder used in the chapter is in below Fig 5.1. It possesses two-part the encoder and the decoder.

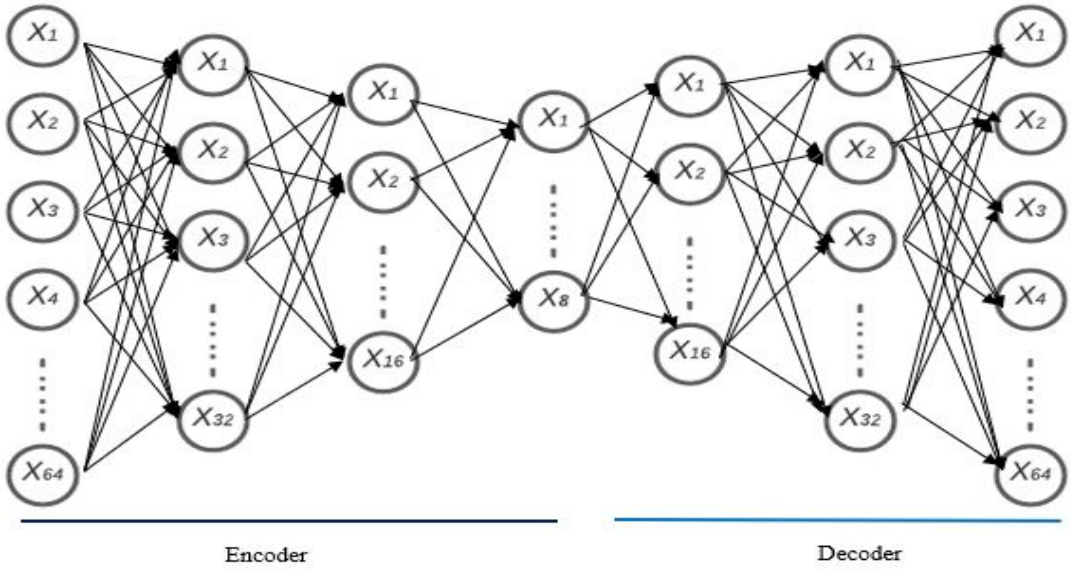

Figure 5.1: A typical Autoencoder Architecture

Encoder: This part of the network encodes or compresses the input data into a latent space representation. The compressed data typically looks garbled, nothing like the original data.
Decoder: This part of the network decodes or reconstructs the encoded data(latent space representation) back to the original dimension. The decoded data is a lossy reconstruction of the original data.
Despite the strong details to work on the same data set, it has some bottlenecks in real-world implications. As a compression method, they don't perform better than its alternatives, for example, jpeg does photo compression better than an autoencoder. The main use case of autoencoder can be mentioned below:

1. Data denoising
2. Dimensionality reduction: Visualizing high-dimensional data is challenging, it can be overcome by an autoencoder.
3. Variational Autoencoders (VAE): This is a more modern and complex use-case of autoencoders that use the probability distribution of input data.

5.3 Methodology of Autoencoder Based Anomaly Detection

Autoencoder is unsupervised learning based on a neural network that can reproduce the input. However, the data properties of training data and test data should be

the same. In this chapter, the base data set is 4G network PS Paging Success rate KPI data. The anomaly detection use case of this KPI is important because if it degrades the user will not be able to authenticate with MME and not be able to use the 4G network.

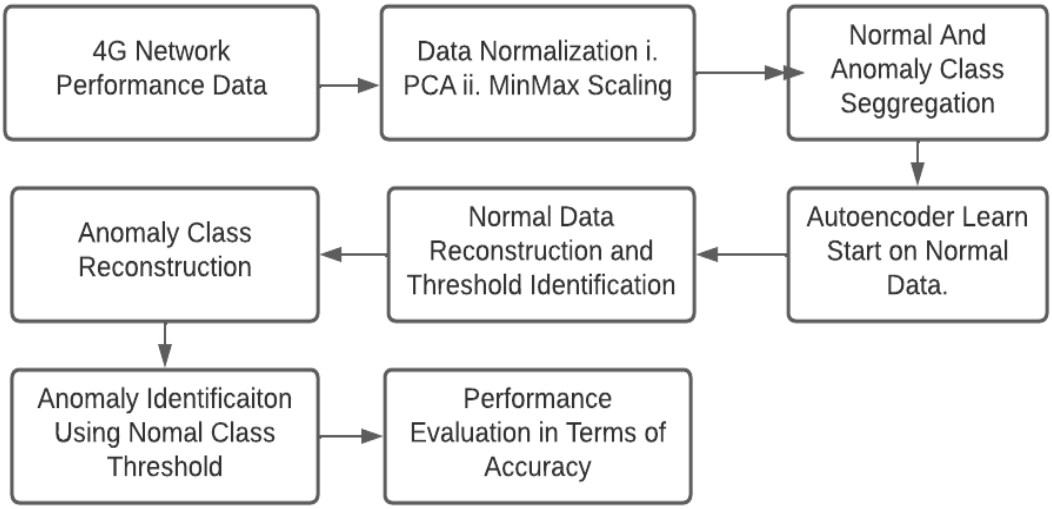

Figure 5.2: Model of Autoencoder Base Anomaly Detection

Fig 5.2 has illustrated the overall methodology used in this chapter. In the initial step, the 4G PS Paging Success rate data is collected for 10 days, and on that data, the necessary normalization technique is implied. The data normalization order is first principle component analysis (PCA) and then the min-max scaling. Once the normalized data set is ready, two subsets are created from the data set. The subsets are the normal class and the anomaly class. After this subset creation, the autoencoder will be trained from the normal class data and normal data is constructed for threshold identification. This threshold is dependent on the system demand and a combination of normal data mean and standard deviation. In the last step, the anomaly class data is reconstructed and if the reconstructed data is beyond the threshold then the data will be considered an anomaly. Lastly, the accuracy of the overall model is determined for the anomaly class.

5.4 4G Performance Data Details

As mentioned earlier for 18 days PS paging success rate data is used for the autoencoder-based anomaly detection. The total sample count is 1736. This dataset will be further divided into normal class and anomaly class. The normal class data count is 1578 and the anomaly class data count is 158. The imbalance between a normal class and an abnormal class is visible and it is good for the encoder. Because the more it trained over the normal data, the better it will be able to reconstruct the data. There are five features those are PS Paging Success Rate(PPSR), PS Paging Request Times(PPRT), PS Paging Success Time(PPST), PS Paging Failure Time (PPFT), and PS Paging Delay (PPD). The PS paging success rate is derived from PS Paging Success Time(PPST) divided by PS Paging Request Times(PPRT)

Fig 5.3 and Fig 5.4 have the details f the first three features of each of the subsets of the dataset. It also describes the possible co-relation between the normal class and the anomaly class. In the normal class, the beginning state of the features is inclined from high to a low value and the maximum value is in the high state.

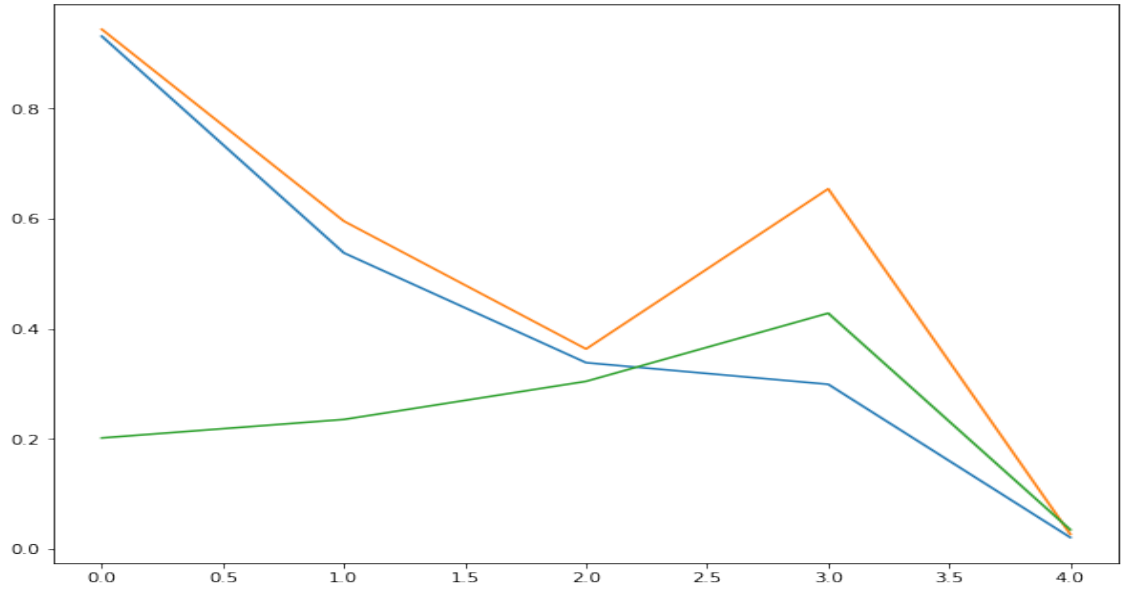

Figure 5.3: First 3 features of Normal Class Subset

On the other hand, the anomaly subclass of the data set is inclined from low to high value, and from both the graph it is visible that there is very less co-relation between anomaly class and normal class. This low co-relation is good because it will make the autoencoder reconstruct the normal and anomaly class better.

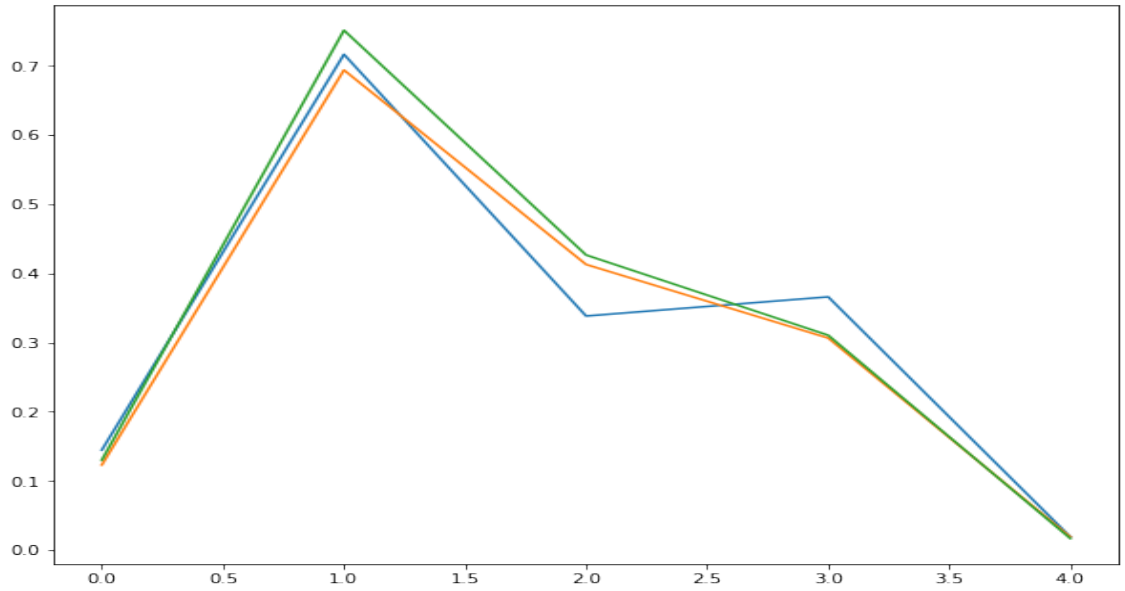

Figure 5.4: First 3 features of Anomaly class Subset

5.5 Result Discussion

Autoencoder is a lossy compression mechanism hence in normal and anomaly class reconstruction the output will not be exact as the input. During the training of the autoencoder, the parameter of loss monitoring is mean square base and the optimizer is adam. The number of epochs is 50 and the batch size is 128 for training. In terms of validation loss based monitor during the training and early stopping criteria are imposed. In the reconstruction error plotting of the normal class data, most of the data is lies between 0.000 to 0.050. Using the mean and standard deviation of the normal class data threshold is determined. The threshold is calculated using one standard deviation for nominal understanding of autoencoder based anomaly detection. The threshold value from the normal class data is 0.0313.

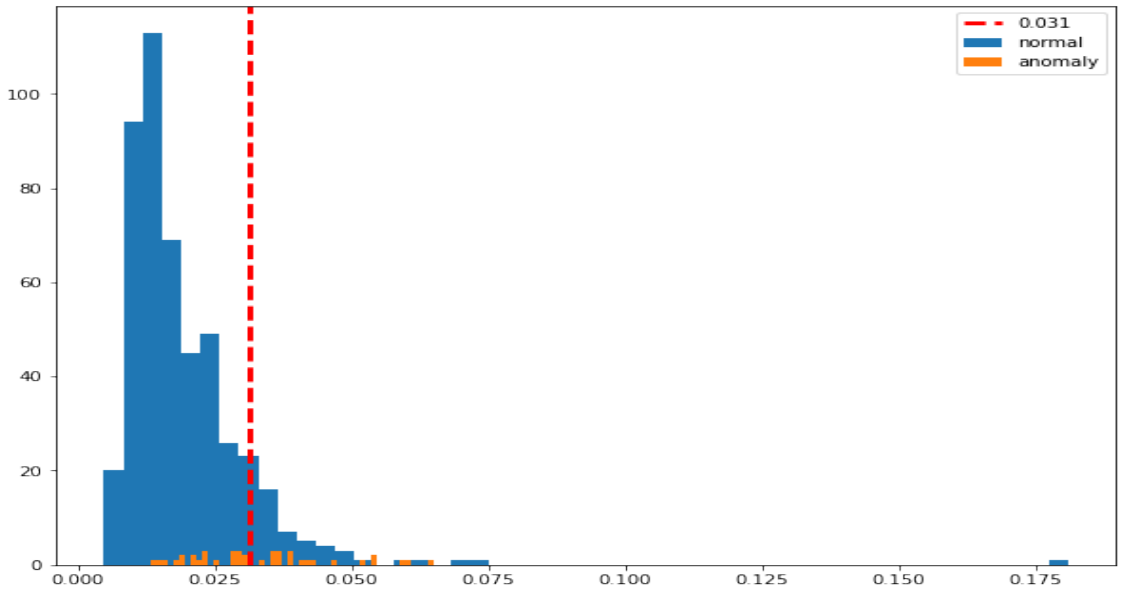

Figure 5.5: Anomaly Detection With Respect to Normal Data Threshold

In the anomaly class representation it is seen that the data is spread on the value from 0.01 to 0.06. In value level comparison it is difficult to detect the anomaly in the anomaly class. To do that a graphical comparison is done where both normal class and anomaly class are plotted together and a fine line is drawn for the normal class threshold. Any value beyond the normal threshold value of either normal class or anomaly class is considered an anomaly. However, this drawing value of the fine line of threshold is not adequate for proper anomaly detection which is illustrated in Fig 5.5
Various factors are driving the outcome autoencoder-based anomaly detection model. The top two variant factors are:

1. Neuron count and hidden layer count in the encoder and decoder level.
2. Threshold identification of normal data considering one standard deviation or two standard deviation

Having those factors in mind another observation has put in overall autoencoder lead work. There are two sets of data normal and anomaly which are trained through the

autoencoder and finally, the anomaly is detected by the threshold got from normal data training. An important observation of the above factor on the normal data. Parameters are:

1. Neuron count starting from 32 to 8 in encoder and 8 to 32 in decoder level or The hidden layer is 1. The threshold of normal data varies from one standard deviation or two standard deviations. Corresponding labelling are "H=1 and T1=1std" and "H=1 and T1=2std"

2. Neuron count starting from 64 to 8 in encoder and 8 to 64 in decoder level or The hidden layer is 2. The threshold of normal data varies from one standard deviation or two standard deviations.Corresponding labelling are "H=2 and T1=1std" and "H=2 and T1=2std"

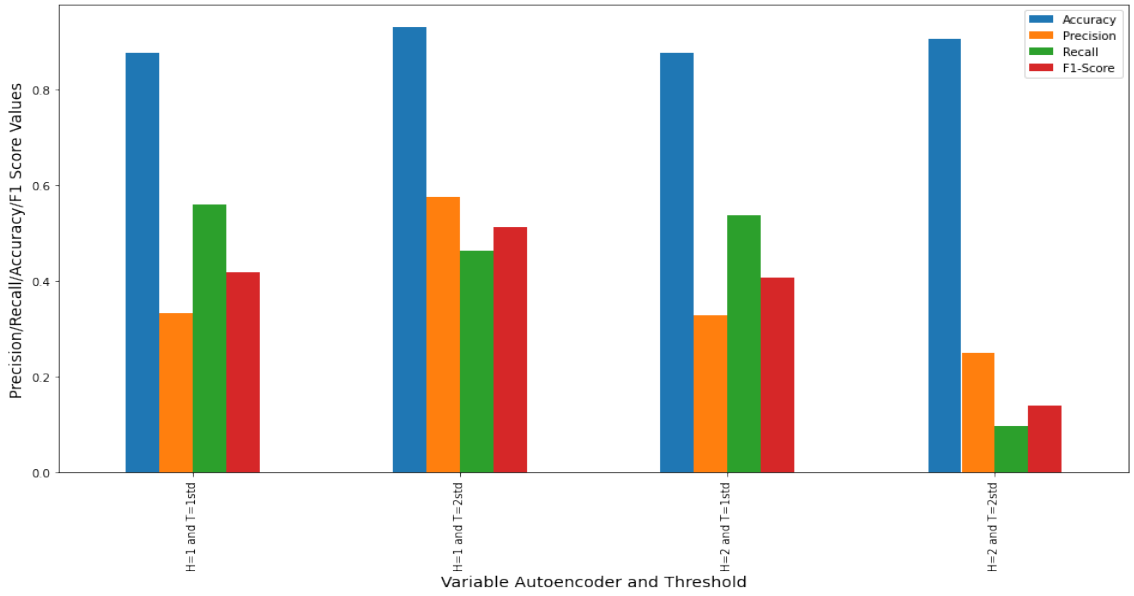

Figure 5.6: F1-Score Visualization for Variable Autoencoder Arch.

The Fig 5.6 visualizing the F1-Score outcome of variable autoencoder architecture. It is clearly visible that most of the auencoder acchitecture provide the efficient anomaly detection however "H=2 and T=2std" achitecture provide the least result. The most efficient autoencoder architecture in terms of F1-Score is "H=1 and T=2std". The details of structure is neuron count starting from 32 to 8 in encoder and 8 to 32 in the decoder level, hidded layer count is 1 and threshold identification by using two standard deviation. The F1-Score is 0.51 which is quite promising.
To more sure about the outcome of this paper, another set of validation approach is exercised. This is based on false positive (FP) value. Here it is also shown despite all other case in "H=1 and T=2std" having the lowest false positive value which is 14. This means this autoencoder architecture denoting not anomaly as anomaly is the lowest possible way.
In nutshell autoencoder is efficient in detecting anomaly in ps paging success rate data set and its variant factors is helping to get the best and optimal anomaly detection model. In the focused and comparative method of F1-Score and false positive base gives more efficient outcome. Most efficient was denoted for the autoencoder

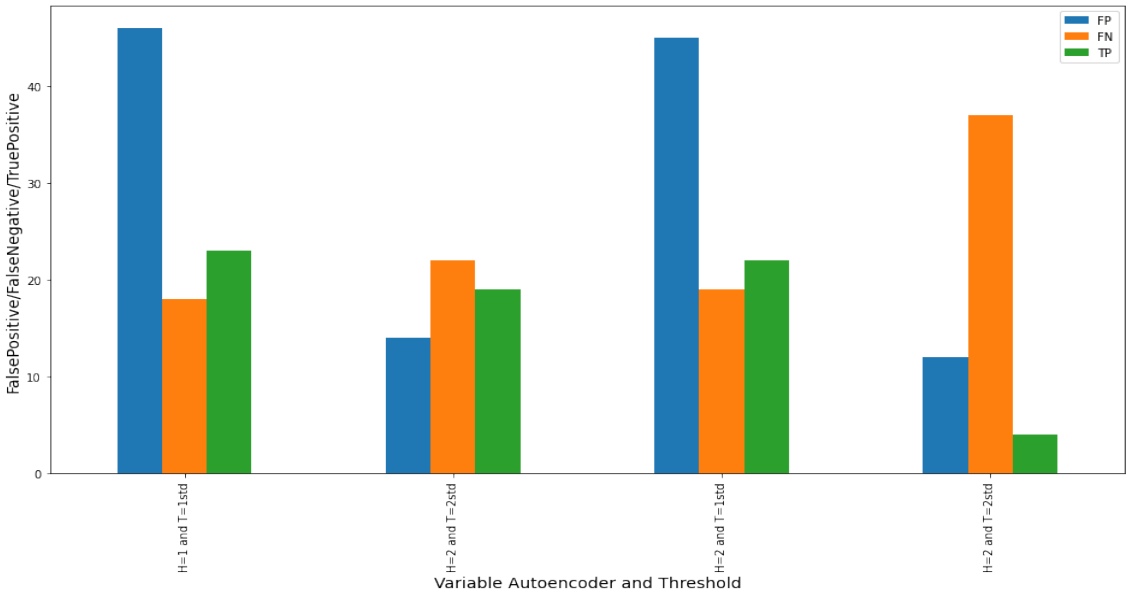

Figure 5.7: False Positive Visualization for Variable Autoencoder Arch.

having hidden layer count is 1 and threshold is identified using the second standard deviation.

Chapter 6

Conclusion

This thesis has described many broader aspects of anomaly detection of a mobile network. Chapter 4 has discussed day level and hourly level types of data and how different supervised machine learning influenced the outcome of anomaly detection. Secondly, it showed why supervised machine learning gave better detection of an anomaly. Finally, it has described the necessity to test multiple supervised machine learning implications irrespective of a dataset. That will help to choose a supervised learning method for the different datasets. The outcome of chapter 4 is that random forest supervised machine learning and support vector machines are efficient for mobile network performance anomaly detection. Chapter 5 has demonstrated how multiple unsupervised learning is used over mobile network performance data and what optimization technique will lead to efficient anomaly detection. Because of using unsupervised machine learning, there is no requirement for ground truth determination. It has also been benchmarked, among multiple unsupervised machine learning which provides a better network anomaly detection. It has been done by converting unsupervised machine learning outcomes to a classification problem, and the model accuracy has been evaluated just like a classification problem. The end outcome identifies as HDBSCAN has outperformed all other unsupervised machine learning algorithms for mobile network anomaly detection. Chapter 6 demonstrates how autoencoder is used in mobile network anomaly detection and provide moderate accuracy in anomaly detection scenario.

Bibliography

[1] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324.

[2] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," vol. 2049, Jan. 2001, pp. 249–257. DOI: 10.1007/3-540-44673-7_12.

[3] J. Peng, K. Lee, and G. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research - J EDUC RES*, vol. 96, pp. 3–14, Sep. 2002. DOI: 10.1080/00220670209598786.

[4] J. Veerasamy, J. Jubin, and S. Kodali, "Practical approach to optimize paging success rate in cdma network," *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3, 1353–1358 Vol. 3, 2005.

[5] Q. Wu and Z. Shao, "Network anomaly detection using time series analysis," ser. ICAS-ICNS '05, USA: IEEE Computer Society, 2005, p. 42, ISBN: 0769524508.

[6] S. Zhang, R. Zhang, and J. Jiang, "A performance management system for telecommunication network using ai techniques," in *Proceedings of the 2008 Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX*, ser. DEPCOS-RELCOMEX '08, USA: IEEE Computer Society, 2008, pp. 219–226, ISBN: 9780769531793. DOI: 10.1109/DepCoS-RELCOMEX.2008.32. [Online]. Available: https://doi.org/10.1109/DepCoS-RELCOMEX.2008.32.

[7] Y. Yasami and S. P. Mozaffari, "A novel unsupervised classification approach for network anomaly detection by k-means clustering and id3 decision tree learning methods," *The Journal of Supercomputing*, vol. 53, pp. 231–245, 2009.

[8] C. H. Yu, "Exploratory data analysis in the context of data mining and resampling," *International Journal of Psychological Research*, vol. 3, Jun. 2010. DOI: 10.21500/20112084.819.

[9] M. Celik, F. Dadaser-Celik, and A. S. Dokuz, "Anomaly detection in temperature data using dbscan algorithm," *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 91–95, 2011.

[10] Z. Chen and Y. Li, "Anomaly detection based on enhanced dbscan algorithm," *Procedia Engineering*, vol. 15, pp. 178–182, Dec. 2011. DOI: 10.1016/j.proeng.2011.08.036.

[11] L. Han, "Using a dynamic k-means algorithm to detect anomaly activities," in *2011 Seventh International Conference on Computational Intelligence and Security*, 2011, pp. 1049–1052. DOI: 10.1109/CIS.2011.233.

[12] A. Muniyandi, R. Ramachandran, and R. Rajaram, "Network anomaly detection by cascading k-means clustering and c4.5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, Dec. 2012. DOI: 10.1016/j.proeng.2012.01.849.

[13] O. Aliu, A. Imran, M. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *Communications Surveys Tutorials, IEEE*, vol. 15, pp. 336–361, Jan. 2013. DOI: 10.1109/SURV.2012.021312.00116.

[14] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *MLSDA'14*, 2014.

[15] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2015.11.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515002891.

[16] Q. Feng, Z. Dou, C. Li, and G. Si, "Anomaly detection of spectrum in wireless communication via deep autoencoder," in *CSA/CUTE*, 2016.

[17] M. Li, H. Wei, and H. Liao, "Mobile terminal quality of experience analysis based on big data," in *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, 2016, pp. 241–245. DOI: 10.1109/ISCIT.2016.7751629.

[18] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 193–198, 2017.

[19] K. Sirait, T. Tulus, and E. Nababan, "K-means algorithm performance analysis with determining the value of starting centroid with random and kd-tree method," *Journal of Physics: Conference Series*, vol. 930, p. 012 016, Dec. 2017. DOI: 10.1088/1742-6596/930/1/012016.

[20] T. Luo and S. Nagarajany, "Distributed anomaly detection using autoencoder neural networks in wsn for iot," May 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422402.

[21] H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, pp. 74–78, Oct. 2018. DOI: 10.26438/ijcse/v6i10.7478.

[22] H. Saeedi Emadi and S. M. Mazinani, "A novel anomaly detection algorithm using dbscan and svm in wireless sensor networks," *Wireless Personal Communications*, vol. 98, Jan. 2018. DOI: 10.1007/s11277-017-4961-1.

[23] P. Wang and M. Govindarasu, "Anomaly detection for power system generation control based on hierarchical dbscan," Sep. 2018. DOI: 10.1109/NAPS.2018.8600616.

[24] S. Xu, "Bayesian naïve bayes classifiers to text classification," *Journal of Information Science*, vol. 44, no. 1, pp. 48–59, 2018. DOI: 10.1177/0165551516677946. eprint: https://doi.org/10.1177/0165551516677946. [Online]. Available: https://doi.org/10.1177/0165551516677946.

[25] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," in *DDC@AI*IA*, 2019.

[26] I. Ghamarian and E. A. Marquis, "Hierarchical density-based cluster analysis framework for atom probe tomography data.," *Ultramicroscopy*, vol. 200, pp. 28–38, 2019.

[27] M. Luo, W. ke, Z. Cai, A. Liu, Y. Li, and C. Cheang, "Using imbalanced triangle synthetic data for machine learning anomaly detection," *Computers, Materials Continua*, vol. 58, pp. 15–26, Jan. 2019. DOI: 10.32604/cmc.2019.03708.

[28] C. Malzer and M. Baum, *A hybrid approach to hierarchical density-based cluster selection*, Nov. 2019.

[29] Y. Lu, J. Wang, M. Liu, *et al.*, *Semi-supervised machine learning aided anomaly detection method in cellular networks*, Jan. 2020. DOI: 10.36227/techrxiv.11634720.

[30] M. Salahuddin, M. F. Bari, H. Alameddine, V. Pourahmadi, and R. Boutaba, "Time-based anomaly detection using autoencoder," Nov. 2020. DOI: 10.23919/CNSM50824.2020.9269112.