UNIVERSITY OF GLOUCESTERSHIRE

CT7205

MACHINE LEARNING AND OPTIMIZATION

ASSESSMENT REPORT

Data-Driven Insights into 4G and 5G Networks Using Machine Learning and Optimization Techniques

TEJASRI SRINIVASAKUMAR - S4328731

MSc Data Science

April 2025

Table of Contents

List of Tables

List of Figures

INTRODUCTION

The exponential growth in mobile communication technologies has led to a vast accumulation of data generated by 4G and 5G networks. These datasets hold immense potential for uncovering patterns, optimizing network operations, and improving service quality (Kousias *et al.*, 2023). This project, titled "Data-Driven Insights into 4G and 5G Networks Using Machine Learning and Optimization Techniques", aims to harness this rich data to derive actionable insights, enhance model performance, and contribute to intelligent network management. By leveraging real-world datasets from 4G and 5G environments, the study applies various machine learning methods including clustering and classification, along with Genetic Algorithm-based optimization to explore, model and fine-tune predictive models for signal quality analysis and network behaviour segmentation.

The 4G dataset consists of telemetry data from a Long-Term Evolution (LTE) network. It includes technical features that reflect signal strength, radio conditions, network configurations, and spatial parameters. Key features include RSRP (Reference Signal Received Power), RSRQ (Reference Signal Received Quality), SINR (Signal-to-Interference-plus-Noise Ratio), Cell tower location data such as cellLongitude, cellLatitude, Altitude, Temporal, and mobility features, such as UTC, Speed, and Distance. It also includes several engineered features such as distance_w and RSRP_diff, used to capture weighted distance from the user to the base station and differences in signal levels. The 5G dataset captures parameters under a New Radio (NR) network configuration. It contains more advanced metrics from 5G architecture, which are related to synchronization signals and channel quality, Beam-related data, reflecting beamformed signal transmission strategies in 5G, and Band and frequency information relevant to 5G NR operation (Ali *et al.*, 2022).

This project explores and preprocesses a 4G dataset, addressing missing values, outliers, skewness, and correlations before applying clustering algorithms like K-Means and Agglomerative Clustering to identify network behaviour patterns. It also uses the 5G dataset for binary and multi-class classification with models such as Logistic Regression and Random forest, supported by feature importance and SHAP analysis for interpretability. Finally, a Genetic Algorithm is applied to optimize k-means clustering which enhances the performance through evolutionary tuning techniques.

The dual focus on 4G and 5G datasets allows comparative exploration of performance characteristics while providing a foundation to improve resource allocation and signal coverage assessment in next-generation networks (Ma *et al.*, 2020). Through systematic preprocessing, model evaluation, and optimization, this work illustrates how data science techniques can be used to solve real-world telecommunication challenges, offering a data-driven foundation for future network intelligence.

TASK 1- Data Preprocessing and Exploration

For task 1, the 4G dataset was chosen due to its comprehensive structure and the presence of diverse data quality challenges, including missing values, outliers, categorical and continuous features, and a multi-class target variable. This variety provided a robust foundation to demonstrate a full spectrum of data preprocessing and exploration techniques, ensuring a meaningful foundation for downstream modelling.

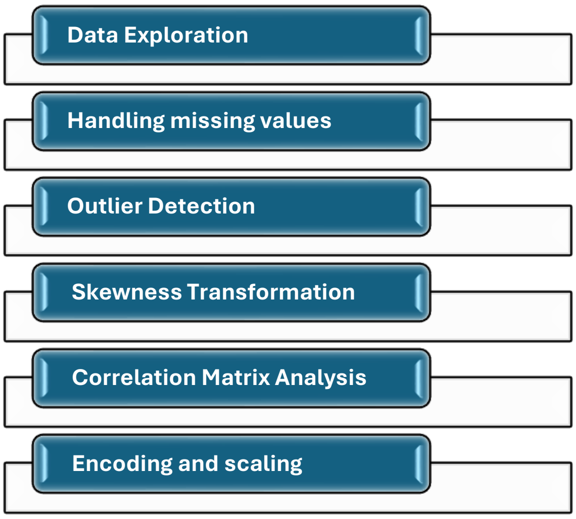


Figure 1 - Flowchart of Data Preprocessing

1.1 Data Exploration

Data preprocessing and exploration are foundational steps in any machine learning or data science workflow. These steps ensure that the raw data is converted into a clean, structured, and interpretable format suitable for analysis or modelling (Saraswat and Raj, 2022). This project started by importing essential libraries like pandas for data manipulation, numpy for numerical operations, seaborn and matplotlib for visualization. Then, the 4G passive measurement data is loaded from a CSV file into a Data frame named df_4g for further analysis.

General information of dataset:

The 4G dataset contains **527,540 rows and 27 columns**, capturing detailed radio access network measurements. The columns having **integer** data types are 'Unnamed: 0', 'EARFCN', 'PCI', 'CellIdentity', 'eNodeB.ID', 'n_CellIdentities' and 'Band'. The columns having **float** data types are 'UTC', 'Latitude', 'Longitude', 'Altitude', 'Speed', 'Frequency', 'Power', 'SINR', 'RSRP', 'RSRQ', 'cellLongitude', 'cellLatitude', 'cellPosErrorLambda1', 'cellPosErrorLambda2' and 'distance'. The columns having **object** data types are 'Date', 'Time', 'MNC', 'scenario' and 'campaign'.

The **initial data inspection** involves examining the first five rows to gain an understanding of the structure and types of features present. Date, Time, and UTC features provide timestamps for when data was captured. Latitude, Longitude, Altitude, and Speed indicate the device's location and movement, collected during drive tests. Columns such as EARFCN, PCI, RSRP, RSRQ, SINR, Power, and Band represent key radio frequency parameters, essential for network performance analysis. eNodeB.ID and CellIdentity identify the serving base station and cell. The scenario and campaign fields denote the context of data collection and the campaign from which it originated.

The **summary statistics** is another important data exploration step which gives descriptive statistics for all numerical features and provides perceptions into central tendency, spread and overall distribution of variables which would help to analyse the data structure and identify potential preprocessing needs (Molin, 2021). The key observations are the RSRP and SINR values fall within expected industry ranges, confirming the integrity of signal strength and quality data. Features like Power, SINR, and RSRQ show a moderate to wide variance, suggesting the presence of heterogeneous network conditions.

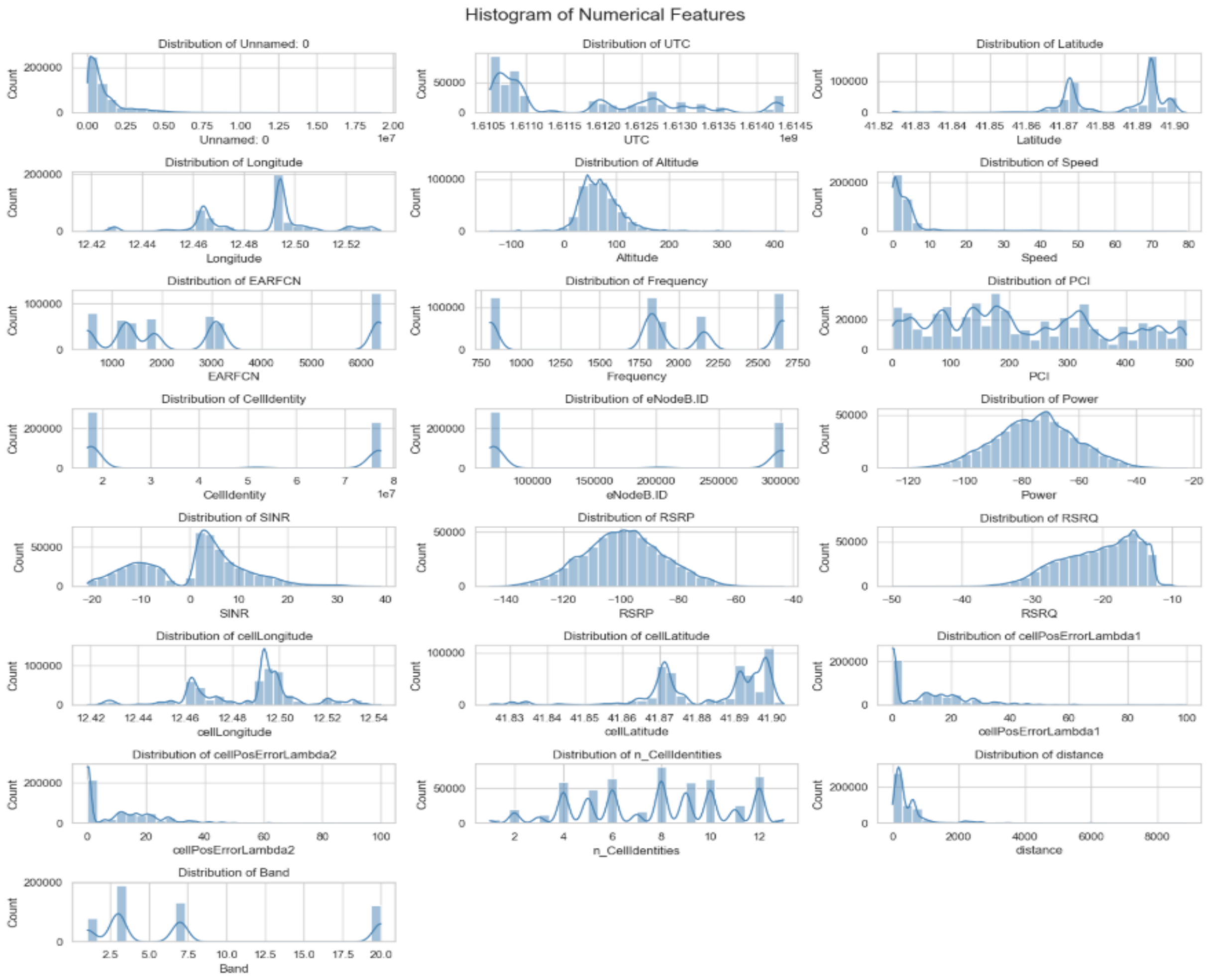


Figure 2 - Visualisation of Numerical Features

The figure above presents histograms for the numerical features in the 4G dataset. Each subplot represents the distribution of a specific numerical variable, providing a visual understanding of the data spread, central tendencies, and potential anomalies. This figure helps to understand the nature of different features and detect skewness and outliers.

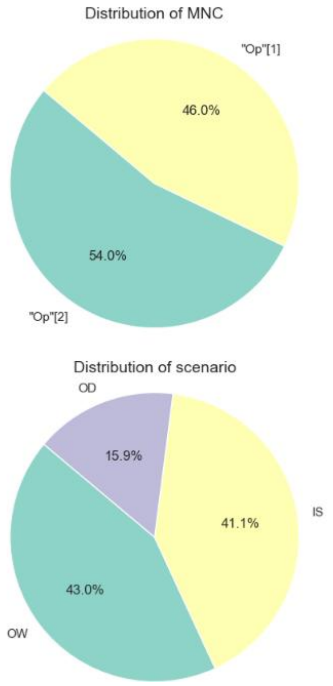


Figure 3 - Visualisation of Categorical Features

The figure above illustrates the distribution of two important categorical features in the dataset using pie charts: MNC (Mobile Network Code) and scenario.
Distribution of MNC shows the proportion of different Mobile Network Codes (MNC), which represent different mobile operators in the dataset "Op"[2]: 54.0% and "Op"[1]: 46.0%. The distribution of scenarios represents different testing or usage conditions:

- OW (Outdoor Walk) has 43.0%.
- IS (Indoor Static) has 41.1%.
- OD (Outdoor Drive) has 15.9%.

This visualisation helps to understand class balance and identify underrepresented categories.

1.2 Handling Missing Values

Missing values are common in real-world datasets and if left untreated can lead to biased model results, and errors in statistical analysis. Therefore, overseeing them properly is a crucial preprocessing step. This study identified the columns having the missing values and their percentages.

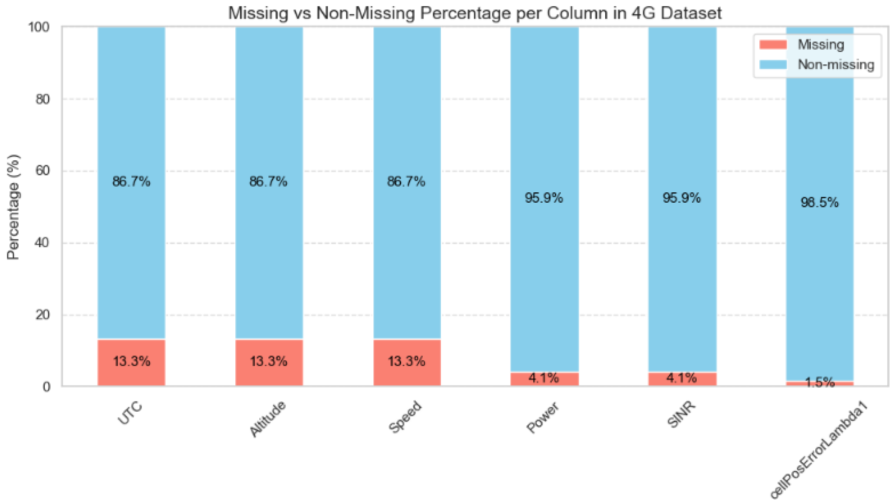
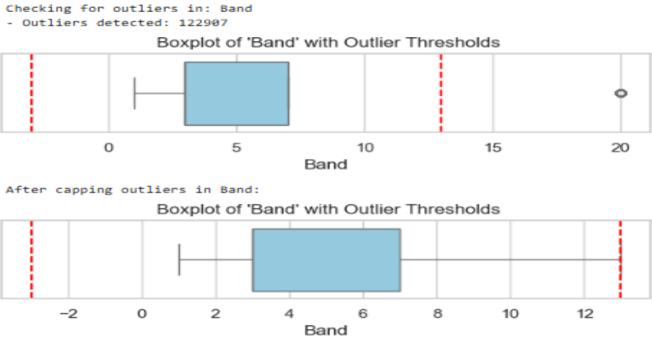


Figure 4 - Missing vs non-missing percentages per column

The above figure represented the columns with null values with their percentages of missing and non-missing values which it used to analyse the quantity of missing values. The columns like UTC, Altitude and Speed had 13.3% (70339 counts) missing values which were the highest of all columns. To manage these missing values, this project splits the feature based on data types as numerical columns which have continuous or discrete numerical values and categorical columns which have label-based values. Next imputation methods were applied using SimpleImputer. Missing values were imputed using the mean of the respective feature for numerical features, and the mode (most frequent value) of the respective feature for categorical columns (Çetin and Yıldız, 2022). Lastly, the duplicate rows and rows having null values were dropped from data frame df_4g.

1.3 Outlier Detection

Outlier detection is used to identify the deviated data points from the majority of the data and the outlier summary illustrates the outliers of each column in the 4G dataset where the outlier count represents a number of data points considered outliers for the respective feature (Boukerche *et al*., 2020). The outlier % represents the proportion of outliers compared to the total values for the features. This analysis was crucial for understanding anomalies and preparing for further processing.



```
Checking for outliers in: Band
- Outliers detected: 122907
```

Boxplot of 'Band' with Outlier Thresholds
0
5
10
15
20
Band

```
After capping outliers in Band:
```

Boxplot of 'Band' with Outlier Thresholds
−2
0
2
4
6
8
10
12
Band

Figure 5 - Band Feature before and after capping Outlier.

The above figure illustrated the number of outliers in the Band feature before capping which had the highest outliers of 23% and it managed the anomalies using outlier capping which adjusted the anomaly to the closest values within the defined range after capping. This would improve model accuracy and support robust analysis.

1.4 Skewness Transformation

The skewness evaluates how asymmetrical the data distribution of features is. This method is used to detect data quality issues and to enhance statistical accuracy.

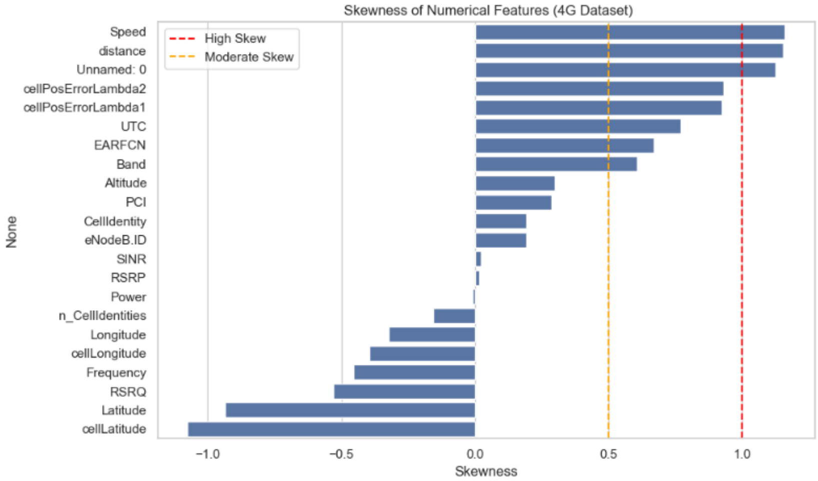


Figure 6 - Skewness of Numerical Features

The image above highlighted the features with high skewness using thresholds which were Speed and distance. The positive skew illustrates the data is skewed right towards higher values and the negative skew illustrates the data is skewed left towards lower values (Gallaugher *et al.*, 2020).

Detected skewness in features is then categorised into log-transformable (positive values) and power-transformable (negative values) which are applied transformations for normalisation and rechecked after skewness treatment improving the data distribution for better analysis.

1.5 Correlation Matrix Analysis

A correlation heatmap is used to visually represent the relationships between variables. It helps to detect redundant and strongly correlated variables (Easaw *et al.*, 2023).

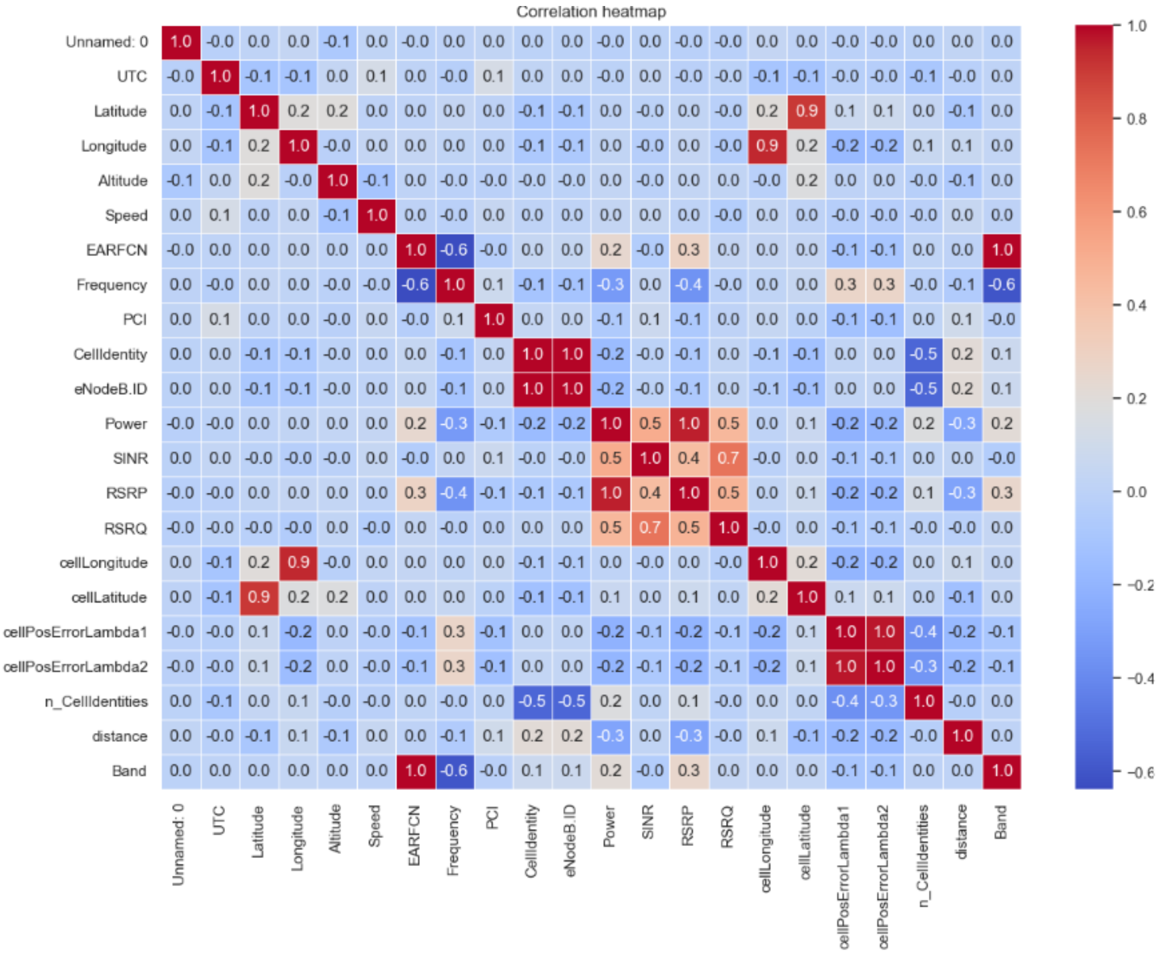


Figure 7 - Correlation Heatmap

From the above heatmap, the strongly correlated features are eNodeB.ID – CellIdentity, RSRP- Power, cellLongitude- Longitude, cellLatitude- Latitude and cellPosErrorLambda2-cellPosErrorLambda1 and Band- EARFCN.

1.6 Encoding and scaling

Encoding is done by Label encoder for categorical columns like MNC, scenario and campaign which assign unique numerical values to categories as the machine learning model works with only numerical data. Scaling ensures fairness and improves model accuracy which is done by Standard scalar for numerical columns as each features have different ranges (Ahsan *et al.*, 2021).

The Date and time feature has a specified format which makes it hard to extract information. Therefore, day, hours, weekdays, and months are extracted from Date and time features dropped from the originals and created new features for better analysis. Finally, the 4G dataset is cleaned and prepared for further analysis.

TASK 2- Clustering Analysis

Clustering is an unsupervised machine learning technique, that groups unlabelled data points based on similarities (Amato and Di Lecce, 2023). This helps to uncover hidden patterns of data and detect anomalies. In telecommunications, 4G clustering reveals distinct types of user signal behaviours and network zones without labels.

2.1 Dataset Selection

In task 2, the 4G dataset was selected due to its diverse signal metrics for unsupervised learning. The quality metrics like SINR, RSRP, RSRQ and contextual variables speed and distance exhibited significant variations suitable for clustering algorithms.
On the other hand, the Latency Tests dataset was not selected as its columns centred on latency values for game servers; the NB-IoT dataset lacks diversity, suffered from significant sparsity and missing values in domain features; Throughout test dataset was excluded as it focused on active measurements which reflected more on network performance testing rather than from user.

2.2 Clustering Techniques

Hypothesis Statement:
The distinct clusters exist within 4G network signal data based on features like SINR, RSRP, RSRQ, speed, distance and hour. These clusters show mobile patterns, varying network quality zones, and user experience levels.
This project focused on the distinct features which would give actionable network segmentations for telecommunication and used four clustering algorithms on the 4G dataset which visualised the clusters and interpreted the cluster results.

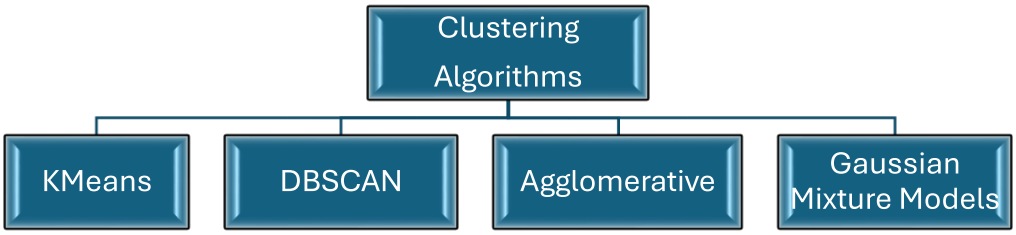


Figure 8 - Clustering Algorithms

This study explored four different clustering algorithms to ensure a more comprehensive and insightful analysis rather than limiting the investigation to a minimum of two algorithms (Rodriguez *et al.*, 2019). By evaluating K-Means, DBSCAN, Agglomerative Clustering, and Gaussian Mixture Models, a more robust comparison was achieved across various performance metrics. This approach not only highlights the strengths and limitations of each method when applied to the 4G dataset but also supports a more data-driven selection of the most suitable clustering technique for the given context.

K-means Clustering

K-means Clustering divides the unlabelled data points into distinct k number of clusters based on the attributes where each cluster is illustrated by its centroid (Sinaga and Yang, 2020). Based on Euclidean distance, each data point is allocated to the closest cluster centroid. K-means is used to discover patterns in datasets. The 4G dataset is used to group users by signal quality, identify abnormal traffic patterns, and allocate bandwidth efficiently.

This project used k-means to split signal data into distinct clusters based on features like SINR, RSRP, RSRQ, Speed, distance and hour. This algorithm is suitable for large datasets because of its simplicity in showing clear groupings (Roy, 2024). This study initialised clusters k to 3, which were then clustered as 1 with 270317 data points, 0 with 153761, and 2 with 103462 data.

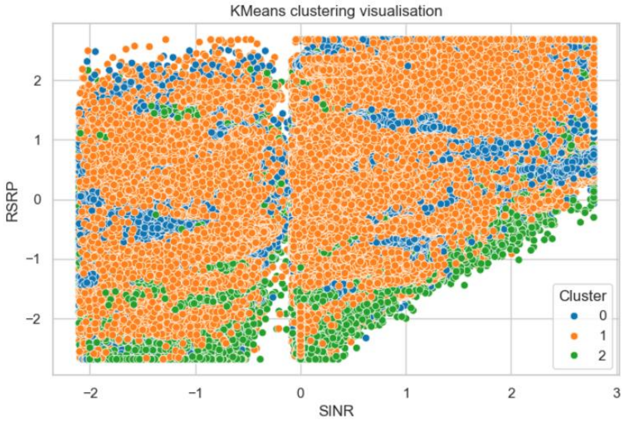


Figure 9 - Kmeans Clustering Visualisation

The above scatter plot represented the output of the K-means algorithm applied to features SINR on the x-axis and RSRP on the y-axis. Cluster 1 (orange) dominated most of the middle range of the two features which had moderate to similar signal characteristics, Cluster 0 (blue) appeared in higher-density horizontal bands at specific RSRP values and moderate-to-high SINR ranges possibly representing specific coverage patterns and Cluster 2 (green) represented lower regions of both features indicating weaker signals. By this visualisation, it detected signal regions of both features and clusters visually separated well. Therefore, the hypothesis was validated in terms of discovering signal-based grouping.

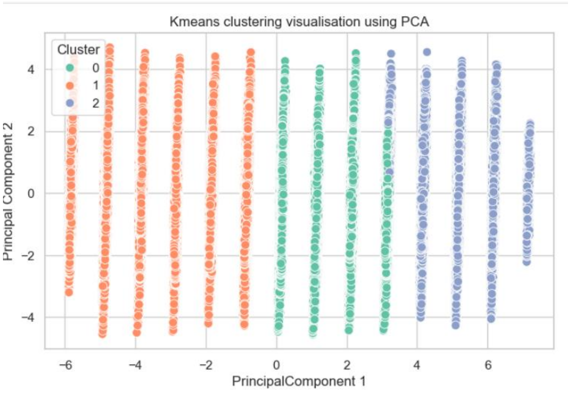
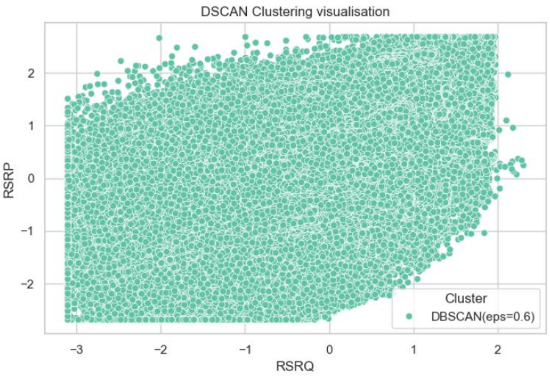


Figure 10 - Kmeans Visualisation using PCA

The above figure illustrated clustering using principal component analysis which reduced the high-dimensional data into 2D. The clusters appear linearly separated and have minimal overlap between clusters (Ding and He, 2004). This study tried using features like scenario and campaign but resulted in overlapping of clusters.

DBSCAN Clustering

Density-Based Spatial Clustering of Applications with Noise groups data points that are closely packed as high density and detect noise well as it lies alone in low-density regions (Deng, 2020). This study used this algorithm as the 4G dataset had non-linear and irregularly shaped clusters. It would be useful in real-world cellular data with signal drops.



v

Figure 11 - DBSCAN Visualisation

The figure represented the DBSCAN visualisation which used an epsilon value of 0.6 which was the neighbourhood size for forming clusters and a minimum sample size of 5. However, this algorithm was not suitable for 4G as it did not cluster the data points well.

Agglomerative Clustering

Agglomerative is hierarchical clustering like tree structure which groups the closest pairs reflecting signal quality and mobility where high-level cluster represents broad zones like urban or rural and lower-level cluster represents weak or strong signals (Ackermann *et al.*, 2012). The study used this algorithm to cluster the 4G dataset based on its hierarchy to identify the segmentations on regions as DBSCAN failed to do that.

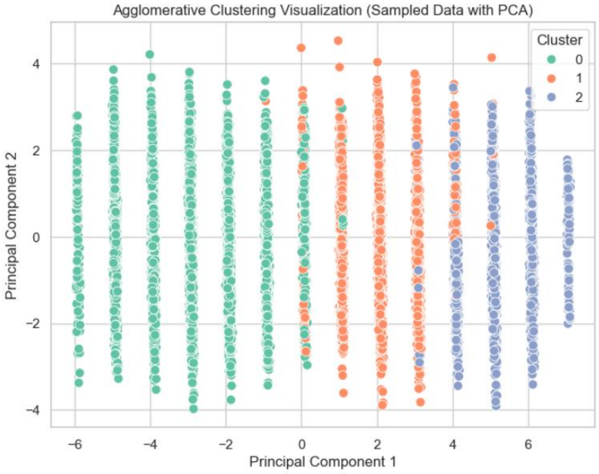


Figure 12 - Agglomerative Clustering Visualisation

The above figure illustrated the agglomerative clustering after applying PCA, which showed a good clustering in the horizontal direction but had a bit of overlap between clusters 0 and 1. The hierarchy worked well by clear progression from left to right that was from cluster 0 to cluster 1 and to cluster 2. This study used this analysis to understand user mobility patterns and PCA ensured high-dimensional signal data transformed into interpretable visualisation.

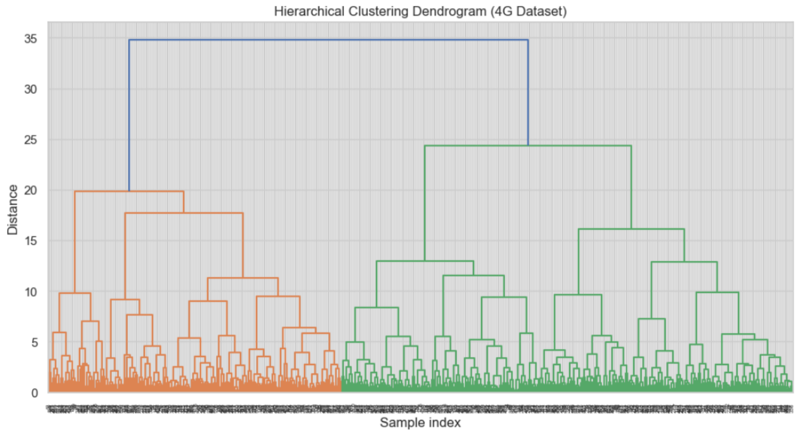


Figure 13 - Hierarchical Clustering Dendrogram

The above dendrogram showed progressive merging of closely related clusters based on distance and added interpretability to the clustering process (Nazari *et al.*, 2015). The tight groupings at the bottom indicated homogeneous signal zones, while larger vertical merges showed more diverse boundary conditions.

Gaussian Mixture Model (GMM) Clustering

The Gaussian mixture model is used to detect cluster overlapping by soft clustering, which provides probabilities to each point for being part of various clusters (Zhang et al., 2021). This clustering supports soft boundary regions by its probabilistic nature. This study used this algorithm to manage complex cluster shapes rather than k-means clustering and used it for outlier detection.

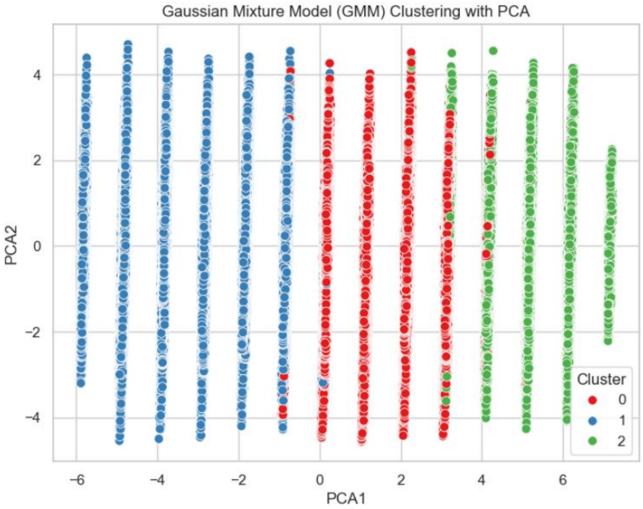


Figure 14 - Gaussian Mixture Model Visualisation

The above figure represented the hard clustering visualisation using the Gaussian mixture model with PCA where it had three clusters. It was hard-labelled clusters which had clear separation and cleanly divided by PCA 1 which holds most of the clustering signal. It captured subtle distribution shapes which was good for real-world noise 4G data. But there was a tiny overlap around cluster boundaries. Hard clustering would force data points to fit only one cluster but in real-world data, some data points belong to multiple clusters which would be seen in soft clustering.

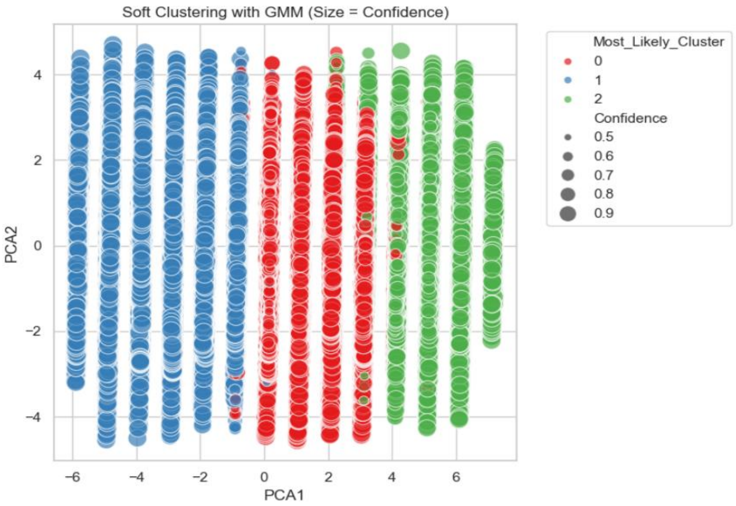


Figure 15 - Soft Clustering with Gaussian Mixture

The above figure represented the soft clustering visualisation using the Gaussian mixture model with PCA. It had three most likely clusters which were clusters with the highest probability by GMM. Confidence (probability) indicated the size of each point of the model's assignment where larger points would denote higher confidence, and smaller points would denote more uncertainty. The plot had well-separated clusters where PCA 1 captured variance used for clustering and PCA 2 showed within-cluster variance. This study used this visualisation to analyse how confident the model was about each cluster point. It identified ambiguous data points and spotted overlap between clusters which was more transparent than hard clustering (Gogebakan, 2021). It was ideal for representing network signal zones with shared and transitional characteristics.

2.4 Comparison and Evaluation

To evaluate the clustering techniques applied to the 4G dataset, this study used Silhouette scores and the Calinski-Harabasz Index (CHI). Silhouette score is used to evaluate how well the clusters are separated and stick together which quantifies how data points work on its cluster and with other clusters (Amato and Di Lecce, 2023). It should range between -1 to 1. Calinski-Harabasz index score is used to measure the ratio of distinct separation of clusters and compactness within clusters where high scores indicate distinct clusters.

Clustering Performance Metrics

Algorithm	Silhouette Score	Calinski-Harabasz Index
Kmeans	0.294	7204.741
GMM	0.287	6931.701
Agglomerative	0.289	5328.707
DBSCAN	0.012	656.937

Table 1 - Clustering Performance Metrics

Evaluation Based on metrics:

1. **Kmeans Clustering:-** It was the best-performing algorithm based on both Silhouette and Calinski-Harabasz scores where it showed clear and well-separated clusters. Cluster 0 and 1 were average but cluster 2 showed weaker signal with high speed and distance which indicated poor user experiences farther from the tower. This aligned well with the hypothesis that performance tiers are based on signal quality and mobility patterns.
2. **Gaussian Mixture Model (GMM):-** It was second and close to kmeans with a slightly lower score, but the cluster showed smooth and probabilistic transitions. Here cluster 2 degraded network conditions.

3. **Agglomerative Clustering:-** It had a reasonable silhouette score, but lower CHI compared to k-means and GMM. It shows less separation in signal features and suggests overlapping groups.
4. **DBSCAN:-** It had the lowest silhouette score of 0.012 and CHI score of 651.937 which identifies 133 micro-clusters and many with noisy data. It was inconsistent signal separation and failed to capture the structure suggested in the hypothesis.

Finally, the Kmeans and GMM model emerged as the best performing techniques with balanced Silhouette score and Calinski-Harabasz scores and DBSCAN as the worst techniques for the 4G dataset.

2.5 Interpretation of Clusters:

This project used the groupby keyword to get a summary table which grouped all clusters in each algorithm to interpret the clusters.

Kmeans Clustering interpretation:-

Cluster	Signal Metrics (SINR, RSRP, RSRQ)	Speed & Distance	Hour
0	Highest among all	Low	~14 (Afternoon)
1	Moderate across all	Average	~10 (Morning)
2	Lowest metrics	High speed & distance	~17 (Evening)

Table 2 - Kmeans Clustering Interpretation

From the above table, this study interpreted that cluster 0 had high SINR, RSRP, and RSRQ values representing stationary users in strong signal zones such as urban centres during peak hours; cluster 1 had average values for most features representing typical daytime users, balanced mobility, and signal; and cluster 2 had low signal metrics with higher speed and distance representing high-mobility users commuting through low-coverage areas. Kmeans successfully aligned the hypothesis by separating signal quality and mobility into clearly interpretable segments, proving its strengths in groups based on the chosen features.

Gaussian Mixture Model interpretation:-

Cluster	Characteristics	Interpretation
0	Strong signal, low speed	Urban or indoor users
1	Balanced across all	Regular users
2	Low signal, high movement	Outdoors or fast-moving vehicles

Table 3 - Gaussian Mixture Model Interpretation

GMM produced very similar clusters to kmeans but with soft boundaries. The probabilistic assignment allowed more nuanced separation. Cluster 0 had higher SINR and RSRP with low speed and distance, reflecting strong coverage zones during peak hours at 14. Cluster 1 had balanced signal values at hour 10, indicating stable network conditions. Cluster 2 had low SINR and RSRP but high Speed and distance, aligning with areas of user mobility at the 17th hour. GMM reaffirmed that signal and speed patterns meaningfully reflect network quality and user behaviour and aligned with the hypothesis.

Agglomerative Clustering interpretation:

Cluster	Characteristics	Interpretation
All	Very small variation	High uniformity, likely fails to separate high-risk areas
0 & 1	Slightly better signal	Urban or average zones
2 & 3	Minor signal drop or speed rise	Marginal performance degradation

Table 4 - Agglomerative Clustering Interpretation

From the table, agglomerative clustering used a hierarchical method which returned four tight clusters with subtle feature differences. Cluster 0 and 1 had relatively balanced signal metrics with slight variations, representing similar network zones or user behaviours. Clusters 2 and 3 had negative values in speed and distance, indicating weaker or less dynamic mobility patterns. It partially aligned with the hypothesis while it showed internal structure and failed to expose extreme outliers and weak signal zones.

DBSCAN Clustering interpretation:-

It generated 133 micro-clusters many based on local densities. It struggled with the 4G dataset due to its sensitivity to density. It did over-segmentation, making mobility insights less consistent and poorly aligned with the hypothesis. Due to noise and varying densities in the 4G dataset, DBSCAN fragmented data too much, making interpretation difficult.

Clustering Quality Summary:-

This project summarised the performance of all clusters with their cluster count, how they were separated, quality on mobility and the rankings of all four cluster algorithms used in the 4G dataset.

Algorithm	Cluster Count	Signal Differentiation	Mobility insight	Evaluation
K-Means	3	Strong	Good	Best
GMM	3	Good	High variation	Very good
Agglomerative	4	Moderate	Low	Average
DBSCAN	133	Poor	Noisy	Not suitable

Table 5 - Clustering Quality Summary

From the above table, this study interpreted that Kmeans effectively separates the data into distinct signal groups with minimal overlap and gave good insights into mobility. While the Gaussian model reflected mixed user behaviours and dynamic signal conditions. Agglomerative clustering is moderate and average in mobile insights as hierarchical clustering would not capture dynamic signals. DBSCAN was not suitable for the 4G dataset, had poor differentiation, and no mobility insights. This comparison helped to understand what each feature represented and map these statistical differences into actionable insights (e.g., time-of-day effects, and mobility influence on signal quality).

Final Evaluation: Alignment with Hypothesis

The hypothesis stated that clustering would reveal groups based on signal and mobility, which would highlight network performance tiers. Kmeans and GMM both effectively extracted interpretable clusters that mapped cleanly to real-world conditions. Agglomerative offered limited insights, while DBSCAN failed to deliver meaningful segmentation due to the data’s lack of density-based structure.

TASK 3- Classification

Classification is a supervised learning which is used to categorise data into labelled groups based on given attributes (FY *et al.*, 2017). It has four types- Binary classification, Multi-class classification, Multi-label classification and Imbalanced classification. In this project, only two classifications are used, that are binary classification which classifies data points into two labelled categories and multi-class classification which classifies data into more labelled categories.

3.1 Dataset selection

This project selected the 5G dataset for classification as it had a rich set of target variables which included multiple well-labelled signal quality indicators like SSS-SINR, SSS-RSRP, and PBCH-RSRP which would be suitable for both binary and multi-class classifications. High feature density for input features allows for meaningful model training. The 5G dataset captured the real-world signal measurements, making the classification model relevant for deployment scenarios.

The other datasets were not selected because the Latency test dataset did not have detailed signal quality metrics, the NB-IoT dataset had inconsistent temporal coverage, and the Throughput test dataset centred around download and upload rather than signal quality. These datasets were not suitable for classification based on signal quality metrics.

3.2 Data preprocessing

The 5G passive measurements dataset was loaded into the data frame as df_5g. It contains **8144644 rows** and **37 columns**. The columns of 5G are 'Unnamed: 0', 'Date', 'Time', 'UTC', 'Latitude', 'Longitude', 'Altitude', 'Speed', 'EARFCN', 'Frequency', 'Band', 'PCI', 'SSBIdx', 'SSS-SINR', 'SSS-RSRP', 'SSS-RSRQ', 'SSS-RePower', 'MNC', 'DM_RS-SINR', 'DM_RS-RSRP', 'DM_RS-RSRQ', 'DM_RS-RePower', 'PBCH-SINR', 'PBCH-RSRP', 'PBCH-RSRQ', 'PBCH-RePower', 'PSS-SINR', 'PSS-RSRP', 'PSS-RSRQ', 'PSS-RePower', 'SS_PBCH-SINR', 'SS_PBCH-RSRP', 'SS_PBCH-RSRQ', 'SS_PBCH-RePower', 'scenario', 'distance_w', 'campaign'.

Data preprocessing is essential for classification techniques. The 5G dataset was cleaned which handled missing values by imputing median value for numerical null value columns respectively and mode value was imputed for categorical null value columns, respectively. Numerical features were scaled and categorical featured were encoded for classification analysis.

The 5G dataset used in this study consists of high-frequency signal measurements gathered from multiple components of the 5G network infrastructure. The recorded measurements were signal strength, quality metrics, and distance-based parameters. It had multiple observations representing different environmental conditions, network scenarios, and user equipment behaviours. The dataset was structured with consistent formatting and minimal missing values. The dataset was split 70:30 for training and testing, with stratified sampling to maintain class balance. These characteristics made it well-suited for classification tasks, allowing a focus on model performance and feature importance.

3.3 Binary Classification

The goal of this project's binary classification is to predict the signal quality based on physical radio measurements based on the 5G dataset. This study trained the classification model to determine whether 'SSS-SINR' (Secondary Synchronization Signal – Signal to Interference plus Noise Ratio) indicates low or high-quality signals. The binary target variable is **Signal_Quality_Binary** which is derived from the '**SSS-SINR'** column where 1 represents good signal quality (SSS-SINR > 0) and 0 represents poor signal quality (SSS-SINR ≤ 0). This allowed the project to frame classification problems where the model would learn from signal strength and quality features to predict network performance.

Selected Input Features :

Feature	**Description**
SSS-RSRP	Received Signal Received Power for SSS
SSS-RSRQ	Received Signal Received Quality for SSS
DM_RS-RSRP	Demodulation reference signal power
DM_RS-RSRQ	Demodulation reference signal quality
PBCH-RSRP	Power of Physical Broadcast Channel
PBCH-RSRQ	Quality of Physical Broadcast Channel
distance_w	Weighted distance from the cell tower

These feature collectives captured both the strength and quality of different signal channels and the proximity of users to network infrastructure which were key determinants of SINR.

Hypothesis Statement:
"Signal quality as measured by SSS-SINR can be reliably predicted using signal strength and distance related measurements such as RSRP, RSRQ and distance to the cell tower."
Null Hypothesis (H_0): Signal strength features (RSRP, RSRQ) have no predictive power in classifying SSS-SINR signal quality. Alternative Hypothesis (H_1): There is a significant relationship between signal features and signal quality, and classification models can predict this accurately.

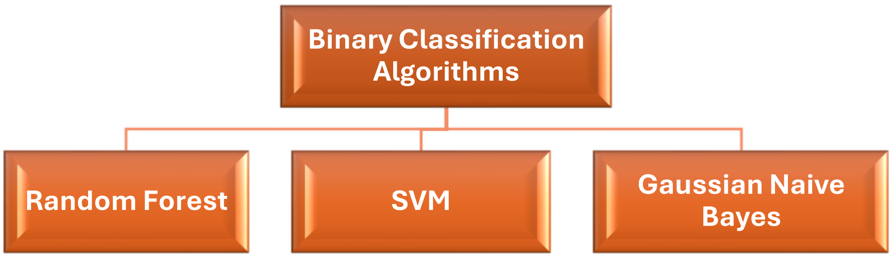


Figure 16 - Binary Classification Algorithm

This project applied three classification algorithms for both binary and multi-class tasks to ensure robust comparison, improve model accuracy, and select the best-performing classifier for the 5G dataset.

Random Forest Binary Classifier:
Random forest classifier is a versatile and powerful algorithm for classifying data points into two categories in terms of signal quality like good or poor. It works by building an ensemble of decision trees and integrating their predictions where each tree independently predicts the class labels (Pal, 2005). This project used this algorithm as it would handle high-dimensional data, robust to outliers and provide feature-importance insights for interpretability.

The key parameters are the number of estimators =100 which built 100 decision trees and random state 42 which ensured the reproducibility of results. The model was trained on X_train and y_train learning patterns that associated the selected signal features with binary signal quality labels.

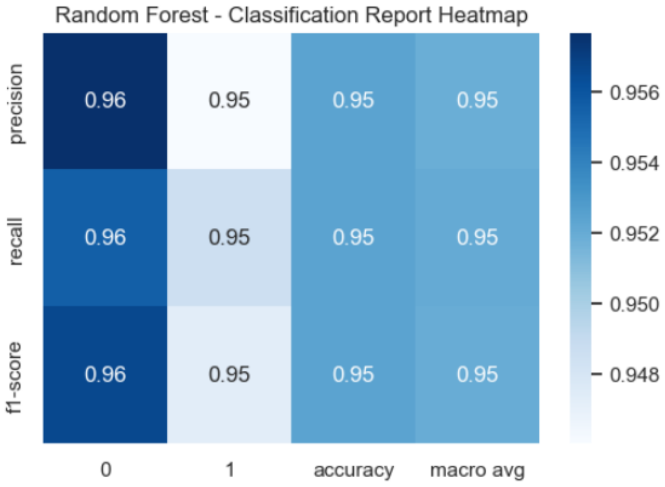


Figure 17 - Random Forest Binary Classification Report Heatmap

The above heatmap illustrated the quality of random forest classifier model predictions by using **classification_report()**. The classification metrics were **Precision** which indicates predicted positives, **Recall** indicates actual positives, **F1-score** is the harmonic mean of precision and recall, **Accuracy** refers to overall correct predictions and **Macro average** refers to the average of metrics across classes (Vujovic, 2021). This model worked well with an accuracy of 95% which was high. In terms of precision, the model correctly predicted class 0 at 96% of the time while class 1 at 95% of the time. In terms of recall, it captured 96% of all actual class 0s and 95% of all actual class 1s. In terms of F1-score, the model balanced between precision and recall which were 0.96 for class 0 and 0.95 for class 1.

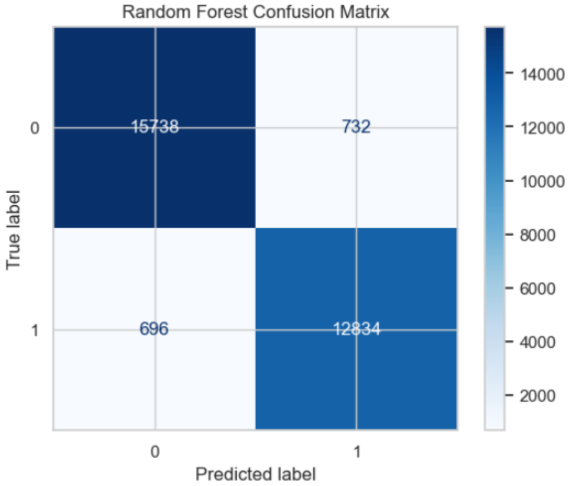


Figure 18 - Random Forest Binary Classifier Confusion Matrix

The above figure confusion matrix explained actual vs predicted classifications made by random forest binary classifier. For class 0: 15738 out of 16470 were correctly identified with ~95.6% and for class 1: 12834 out of 13530 were correctly identified with ~94.9% (Patro and Patra, 2014). It also had some misclassifications where 732 false positives and 696 false negatives. This project concluded that the model balanced error distribution and was consistent across both classes. This model successfully aligned with the hypothesis of selected signal features sufficient to differentiate signals.

SVM Binary Classifier:

Support Vector Machine (SVM) uses optimal hyperplane to separate the data points into two categories distinctly. It identifies a boundary that effectively separates two classes and is robust against overfitting (Saravanan and Sujatha, 2018). This study used this algorithm as it is effective for high-dimensional data and handles both linear and non-linear data.

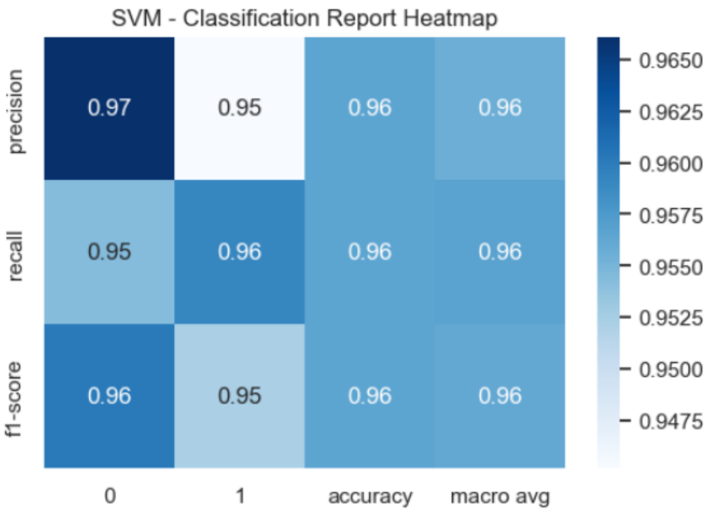


Figure 19 - SVM Binary Classification Report Heatmap

The above heatmap of the SVM classification report had an accuracy of 96%. In terms of precision, class 0 predicted 97% and class 1 of 95% correctly. For recall, the model captured actual class 0 of 96% and class 1 of 96% time. F1 scores of 95% and 96% indicated consistent performance across both classes. This result confirmed that the hypothesis of signal features was predicted well and distinguished.

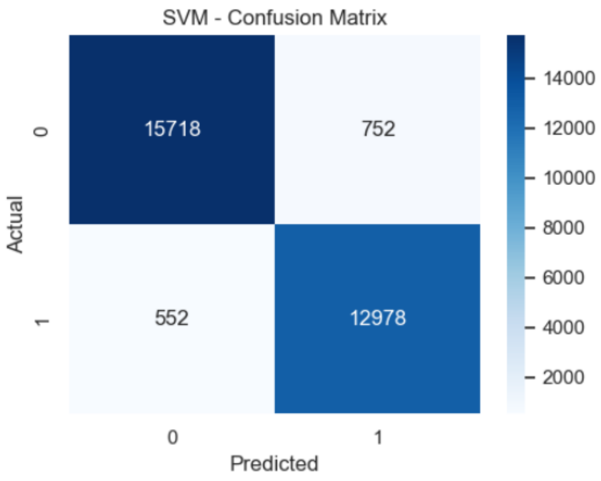


Figure 20 - SVM Binary Classifier Confusion Matrix

The above confusion matrix of SVM represented the model predicting Signal_Quality_Binary where it correctly predicted 15718 for class 0 of ~95% and 12978 for class 1 of ~ 95.9%. It had 752 false positives where it predicted a good signal, but it was actually poor and 552 false negatives where it predicted a poor signal, but it was actually good. But overall SVM classifier was well-balanced and performed robustly on binary signal quality.

Gaussian Naive Bayes Binary Classifier:

Gaussian NB classifier is a classification algorithm based on Bayes's theorem. It models continuous features using normal distribution and it also calculates the probability of class label with the input features, and it assumes all attributes are conditionally independent which should not affect other feature's probability (Ontivero-Ortega *et al.*, 2017). This study used this algorithm as it is ideal for signal strength classification and faster for large datasets.

This model predicted well with an accuracy of 92% and precision for class 0 correctly predicted 91% and 95% for class 1. Recall 96% captured for class 0 and 88% for class 1. F1 score, good balance of both precision and recall but slightly lower for class 1 (91%) than class 0 (93%). This model was suitable for real-time interpretable models.

3.4 Evaluation of Binary Classifiers

This study compared the evaluation metrics of all binary classifiers to identify which algorithm was suited best for the 5G dataset and how it well aligned with the hypothesis.

Metric	Random Forest	SVM	Gaussian NB
Accuracy	**0.95**	**0.96**	**0.92**
Precision (0/1)	**0.96 / 0.95**	**0.97 / 0.95**	**0.91 / 0.95**
Recall (0/1)	**0.96 / 0.95**	**0.95 / 0.96**	**0.96 / 0.88**
F1-Score (0/1)	**0.96 / 0.95**	**0.96 / 0.95**	**0.93 / 0.91**

Table 6 - Binary Classifiers Comparison

From the above table of comparison, this study interpreted that the **Support Vector Machine (SVM)** performed the best in terms of accuracy at 96% and had balanced precision and recall across both classes. It significantly distinguished good vs poor signal quality and aligned well with the hypothesis that signal metrics could clearly define the class boundaries. Second, comes the **Random forest,** with an accuracy of 95% which was also an impactful classification, and it effectively learned the importance of features like RSRP, RSRQ and distance. Lastly, **Gaussian Naive Bayes**, with an accuracy of 92% was good but lagged due to its assumption of feature independence. The reduced recall showed it struggled to detect signals which were less suitable for this domain and partially misaligned with the hypothesis (Starovoitov and Golub, 2020). By comparing three algorithms, this project came to know which model suited the real-world signal data.

3.5 Feature Importance of Binary Classification

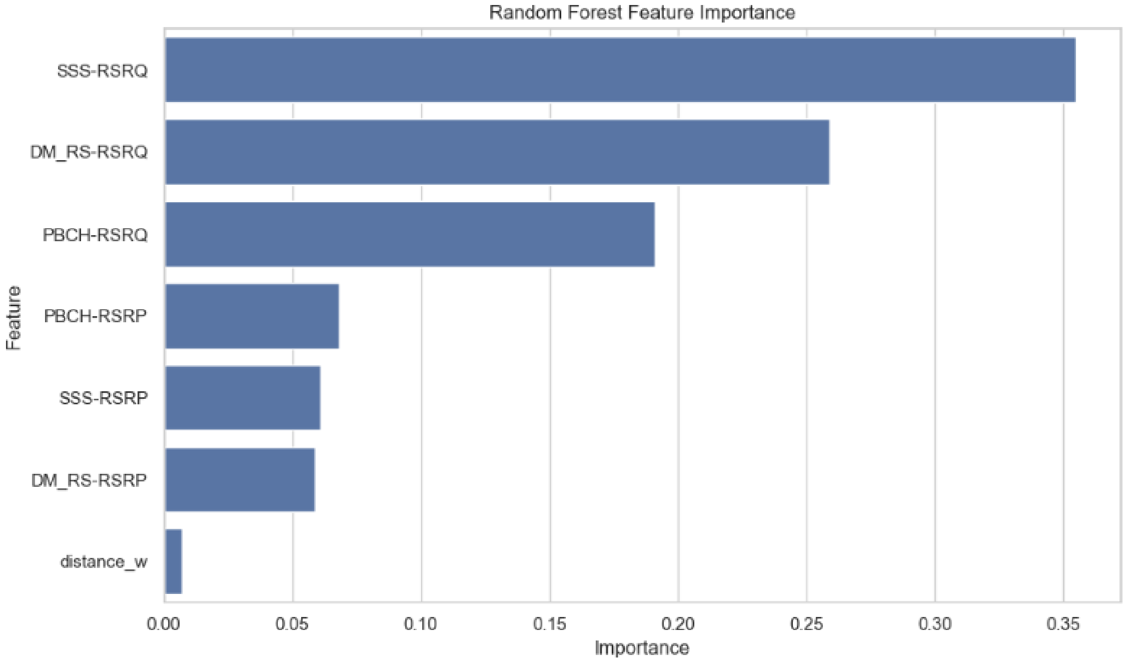


Figure 21 - Random Forest Binary Classifier Feature Importance

While all models are evaluated for classification performance, feature importance can only be analysed for the Random Forest model due to its superior interpretability and alignment with the project hypothesis (AlSagri and Ykhlef, 2020). Other models like SVM and Gaussian NB do not offer built-in interpretability or have less interpretive strength. The above random forest feature importance plot illustrated that SSS-RSRQ had ~0.34 importance score which indicated the most critical feature indicating signal quality degradation. A poor RSRQ usually correlates with interference or weak signal. Secondly, DM_RS-RSRQ had a ~0.28 score which was a strong indicator of signal reception quality. Thirdly, PBCH-RSRQ of ~0.22 captured broadcast channel reception. Lastly, distance_w was the least important of 0.01 which might have a minor indirect influence. This study summarises that RSRQ features dominated the prediction task, supporting the idea that the quality of the signal not just power plays a larger role in determining SINR, thus aligning with the hypothesis.

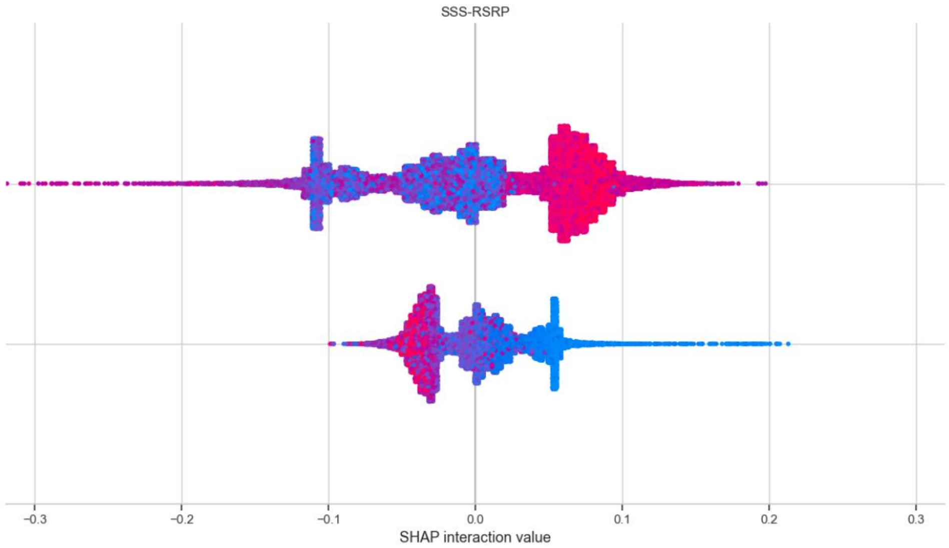


Figure 22 - Random Forest SHAP plot of SSS-RSRP

The SHAP (SHapley Additive exPlanations) summary plot for SSS-RSRP showed how values of this feature contributed to model predictions across the test set: Red dots(high values) on the positive SHAP side indicated that higher SSS-RSRP values increased the likelihood of the signal being classified as *good*. Blue dots (low values) on the negative SHAP side showed that lower values contribute to *poor signal* prediction (Nohara *et al.*, 2021). The spread and shape of the SHAP values suggest non-linear interactions, where certain ranges of SSS-RSRP had more predictive power than others. This interaction confirmed that even features with lower Gini importance (like SSS-RSRP) still had nuanced effects, especially in combination with others, highlighting the value of feature diversity in the model. Thus, findings aligned closely with the hypothesis and multiple feature types (synchronization, demodulation, broadcast) collaboratively define signal conditions.

3.6 Multi classification

Multi-class classification is used to classify data into more distinct categories and each data point belongs to only one class (Grandini *et al.*, 2020). The multi-class classification focuses on predicting the signal quality class of a 5G connection using a set of signal reference metrics. The target variable is **Signal_Quality_Multi** derived from **SS_PBCH-RSRP** which is divided into three quantiles representing low(0), medium (1) and high (2) signal quality. This creates real-world signal class perception.

Selected Input features:

1. SSS-SINR (Secondary Synchronization Signal – Signal-to-Interference-plus-Noise Ratio)
2. SSS-RSRQ (Reference Signal Received Quality)
3. DM_RS-SINR (Demodulation Reference Signal – SINR).

4. DM_RS-RSRQ (Demodulation RSRQ)
5. PBCH-SINR (Physical Broadcast Channel – SINR)
   PBCH-RSRQ (Physical Broadcast Channel – RSRQ)
6. distance_w (Weighted Distance from Base Station)

Hypothesis Statement:
The quality of 5G signal reception (categorized into low, medium, and high signal strength based on SS_PBCH-RSRP quantiles) can be accurately predicted using a combination of signal reference quality metrics (SSS, DM_RS, and PBCH for SINR and RSRQ) along with spatial information such as distance. These features have meaningful relationships with signal strength and are expected to allow classification into the three quality levels with reliable accuracy.

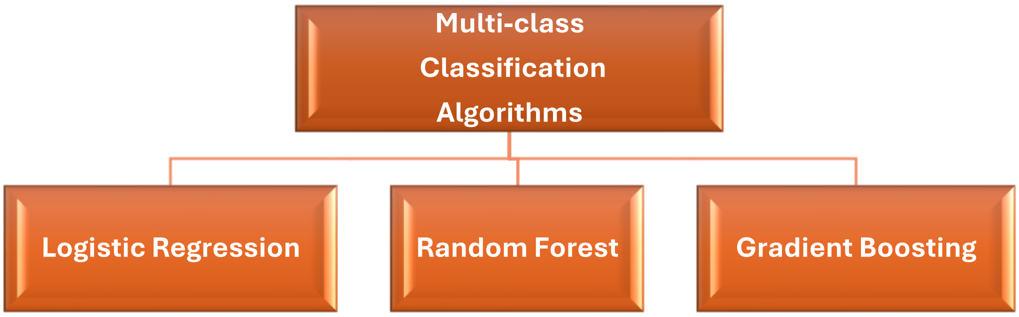


Figure 23 - Multi-Class Classification Algorithms

Random Forest Multi-class Classifier:
Random forest multi-classifier is an ensemble model that uses various decision trees which then integrates them to enhance model prediction and control overfitting (Chaudhary et al., 2016). This project selected this model due to its suitability for non-linear relationships. A random forest Classifier is used to classify signal quality into three performance categories using key signal features. It used 50 trees (estimators) and maximum tree depth of 10 and a random state of 42. This model was evaluated by a classification report which assessed the quality of signal classes.

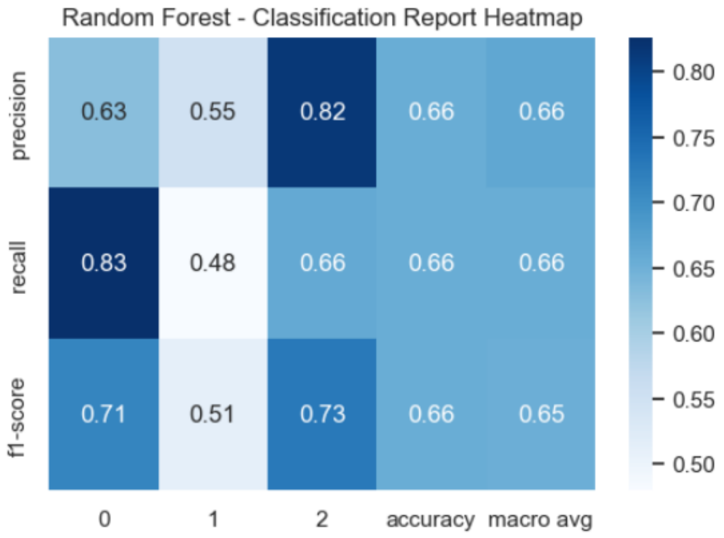


Figure 24 - Random Forest Multi-class Classification Report Heatmap

The above heat map was used to visualise a classification report of random forest, which achieved a moderate overall accuracy of 66% which was acceptable for a multi-class classification with overlapping classes and noisy measurements. The high recall (83%) for the 'Low' class indicated the model was very sensitive to weak signals, which was beneficial for network optimization efforts where identifying poor conditions is critical. The 'Medium' class performance is lower, likely due to signal measurements being less distinct from the other two categories. This might indicate the need for additional features or refined class boundaries. Overall, the F1 scores reflected balanced performance, especially between precision and recall for each class, which aligned with the hypothesis that signal measurements would predict quality tiers.

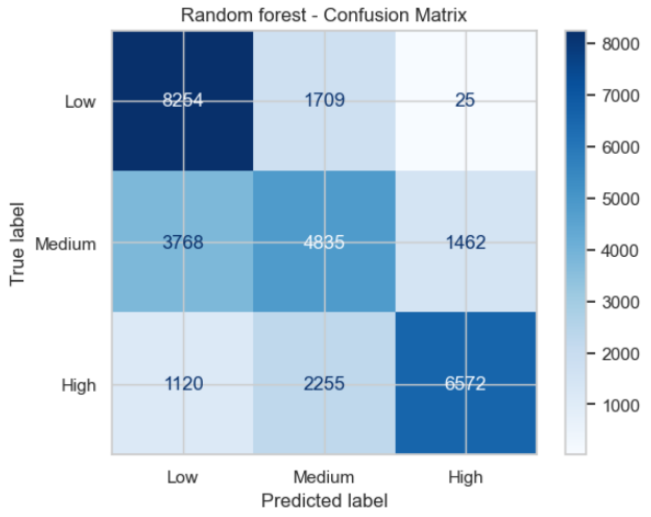


Figure 25 - Random Forest Binary Classifier Confusion Matrix

The above random forest confusion matrix indicates the diagonal dominance as correct predictors of classes ‘Low’ - 8254, ‘Medium’ -4835 and ‘High’- 6572. There were also some overlaps especially between Medium & Low: 3768 Mediums classified as Low and High & Medium: 2255 Highs classified as Medium. This reflected overlapping feature distributions in boundary cases (Trajdos and Kurzynski, 2016). Signal characteristics are not distinguished between Medium and adjacent classes.

Gradient Boosting Multi-class Classifier:

Gradient boosting builds trees sequentially where it corrects errors of its previous tree which is based on an ensemble model unlike random trees which build trees in parallel (Natekin and Knoll, 2013). This study selected this model as it corrects the error resulting in high accurate and robust algorithm.

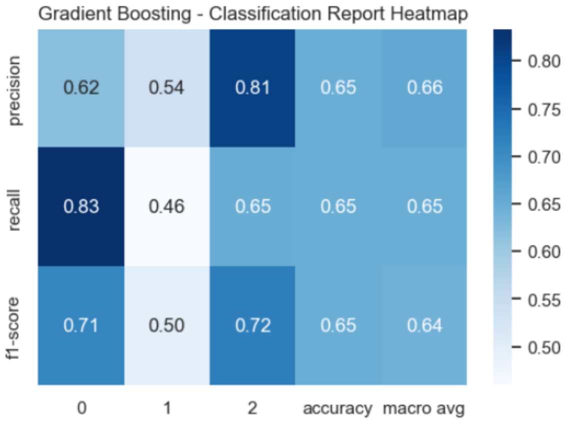


Figure 26 - Gradient Boosting Binary Classification Report Heatmap

The gradient boosting model predicted with an accuracy of 65% with a high recall of 83% for the ‘Low’ class which indicated that the model was good at detecting actual Lows. The outstanding precision of 81% for the ‘High’ signal helped the model to recognise strong signal scenarios. Medium class had low performance indicating confusion with Low or High signals.

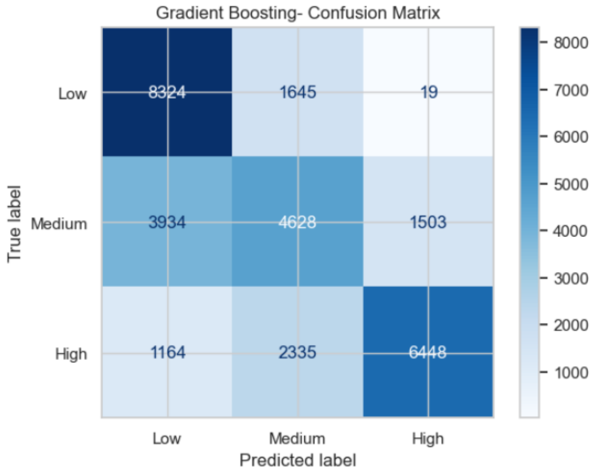


Figure 27 - Gradient Boosting Binary Classifier Confusion Matrix

The above gradient boosting heatmap correct predictions were Low – 8324, Medium – 4628 and High – 6448. Confusions had existed between neighbouring classes, especially between Medium which was misclassified as Low or High. 3934 Mediums misrepresented as Lows which struggled with boundary cases of both the classes.

Logistic Regression Multi-class Classifier:

Logistic Regression is a linear model which uses a sigmoid function to predict the class label. This model predicts the probability of data points belonging to a certain class which enables confidence-based decisions (Petkus *et al.*, 2017).

This model was predicted with an accuracy of 62% which was less than the other two models. It had 74% precision in the High class but the medium class was less performed which did not capture complex variance for the intermediate class.

3.7 Evaluation of Multi-class Classifiers

Model	Accuracy	Macro F1	Class 0 (Low) F1	Class 1(Med) F1	Class 2(High) F1
Random Forest	**0.66**	**0.65**	**0.71**	**0.51**	**0.73**
Gradient Boosting	**0.65**	**0.64**	**0.71**	**0.50**	**0.72**
Logistic Regression	**0.62**	**0.61**	**0.65**	**0.46**	**0.73**

Table 7 - Multi-class Classifiers Comparison

The best model for multi-class classification was **Random Forest** which had good scores on all metrics and achieved accuracy with balanced performances. This was very well aligned with the hypothesis. Second, **Gradient Boosting** offered similar results with high precision in detecting 'High' signals. Both models were good at predicting strong and weak signals but had low performance in detecting Medium signals. Lastly, **Logistic Regression** had an accuracy of 62% which was good but poor at handling non-linear relationships in complex feature spaced and was not suitable for this dataset.

The hypothesis that multi-source 5G signal features were used to classify signal quality was strongly supported by the performance of Random Forest and Gradient Boosting. These models successfully learned from complex, nonlinear relationships in the radio signal environment to differentiate between signal quality levels, especially "Low" and "High". But these models slightly struggled to predict Medium signal.

3.8 Feature Importance of Multi-Classification

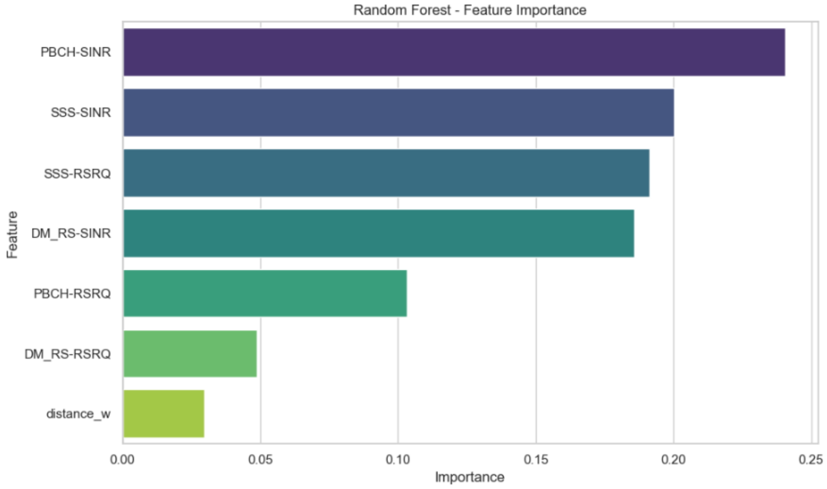


Figure 28 - Random Forest Multi-class Classifier Feature Importance

Random forest classifier spread importance more evenly across features where PBCH-SINR was the most influential for splitting decisions with ~0.23 importance. All three signal types (PBCH, SSS, DM-RS) contributed meaningfully, especially their SINR values. RSRQ (Reference Signal Received Quality) has a relatively lower impact than SINR metrics.

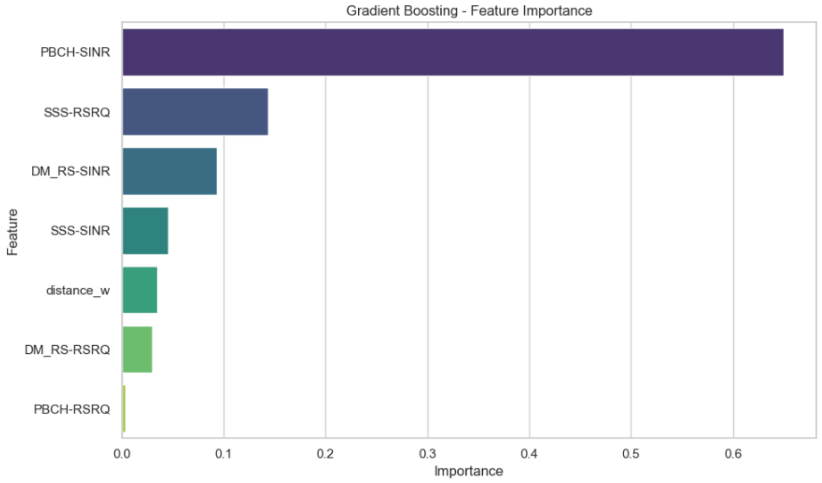


Figure 29 - Gradient Boosting Multi-class Classifier Feature Importance

Gradient Boosting relied heavily on PBCH-SINR and much less on others. It focused more on strong signal/noise indicators than average-quality ones (Adler and Painsky, 2022). Importance drops off quickly after the first feature, suggesting model behaviour was driven by dominant signals.

From the feature importance of these two models, **PBCH-SINR** was the most important feature which indicated that SINR at Physical Broadcast Channel was highly predictive and highlighted SINR-based features than RSRQ metrics due to redundancy or lower variance. distance_w was a consistently low-impact feature.

TASK 4- Optimization Using Genetic Algorithms

For task 4, this project aims to address the optimization of the K-Means clustering algorithm that is applied to the 4G dataset.

4.1 Optimization of K-Means Clustering

The performance of the K-means clustering algorithm heavily depends on proper optimization, which involves identifying the best solution to minimise errors and maximise the evaluation metrics. To enhance the performance of K-means, it depends on parameters like the number of clusters (k) and the initialisation method. Therefore, selecting optimal values for these parameters is crucial for achieving meaningful clustering results (Na *et al.*, 2010).

4.2 Optimization Problem Formulation

Objective: Optimize the parameters of the K-Means clustering algorithm to maximise the clustering quality measured by the silhouette score.
Parameters to Optimize:

1. **Number of Clusters (k)**: k is an integer value representing the number of clusters to form. The search space is defined as k ∈ [2,10].
2. **Initialization Method**: A categorical variable indicating the method for initializing the cluster centroids. The options are 'k-means++' (encoded as 0) and 'random' (encoded as 1).

Fitness Function: The silhouette score is used as the fitness function. It calculates an object's similarity to its own cluster with other clusters. The score ranges from -1 to 1, where a higher value would represent the distinct clusters.

4.3 Implementation of Genetic Algorithm

Priorly libraries needed for optimization were loaded and features selected 'SINR', 'RSRP', 'RSRQ', 'Speed', 'distance', and 'Hour' were the same which was used in task 2 clustering of the 4G dataset. A subset of 20,000 samples was randomly selected from the 4G dataset for computational efficiency. The features were scaled to ensure each attribute contributed equally to the clustering process.

GA configuration:
Genetic Algorithm configured using DEAP library which created fitness and individual setup. The creator.FitnessMax defined the fitness function. The weights=(1.0,) meant to maximize the objective silhouette score. The creator.Individual is defined as a custom individual where

a list of fitness assigned to it. Gene Definitions were registered which generate **n_cluster** from 2 to 10 and **init_method** was also created using 0='k-means++' and 1='random'. The **eval_kmeans** was the fitness function used in this GA to evaluate the quality of the given kmeans clustering configuration.

GA Execution function:
Genetic operators registered were **Selection**: tournament selection with a tournament size of 3, **Crossover**: uniform crossover with an independent probability of 0.5 for each gene, and **Mutation**: shuffle mutation with an independent probability of 0.3 for each gene. For each individual, the K-Means algorithm was applied with the specified number of clusters and initialization method. The resulting clustering was evaluated using the Silhouette Score, which serves as the fitness value.
eaSimpleWithElitism Function: This function defined a custom evolutionary loop for the Genetic Algorithm with elitism, meaning the top-performing individuals were preserved across generations.
For each generation (ngen):

- Variation: Applies crossover and mutation to create a new generation (offspring).
- Fitness Evaluation: Evaluates only the offspring that haven't been evaluated yet.
- Elitism: Selects the top elite_size individuals from the *previous population* to retain.
- Next Generation: Combines the elite with the best offspring to form the next generation.
- Fitness Check: Ensures all individuals have valid fitness (useful after mutation).
- Verbose Logging (optional): Prints the max and average fitness of the current generation.

The final population after evolution, were returned ideally containing highly fit solutions for good K-Means cluster configurations (Ahmed, Seraj and Islam, 2020).

In running GA optimization, it was initialised with a population of 30 individuals and the number of generations was set to 20 generations. result_population would store the final evolved population of GA. In the final evaluation, the GA retrieved the best individual from the final GA population which decoded it to get the best number of clusters (best_k) and initialisation method (best_init). It trained the kmeans model using optimized GA parameters and computed silhouette scores that displayed the best configuration found, both optimised and baseline silhouette scores. From the GA results, the best-performing individual was found by generation 7 with a silhouette score of 0.2241.

4.4 Results and Analysis

Evaluation Metric	Baseline (k=3)	Optimized (k=2)
Silhouette Score	0.1783	**0.2241**
Calinski-Harabasz Index	4818.47	**6541.96**
Davies-Bouldin Index	1.8015	**1.6718**
Initialization Method	k-means++	**random**

Table 8 - Optimized vs Baseline Clustering Comparison

The optimized K-Means configuration achieved improvements across all three clustering evaluation metrics. Silhouette score (higher is better) indicated cluster separation improvement and compactness. Calinski-Harabasz index (higher is better) higher value reflected better-defined clusters. In the Davies-Bouldin index, (lower score is better) lower values of GA suggested tighter and well-separated clusters.

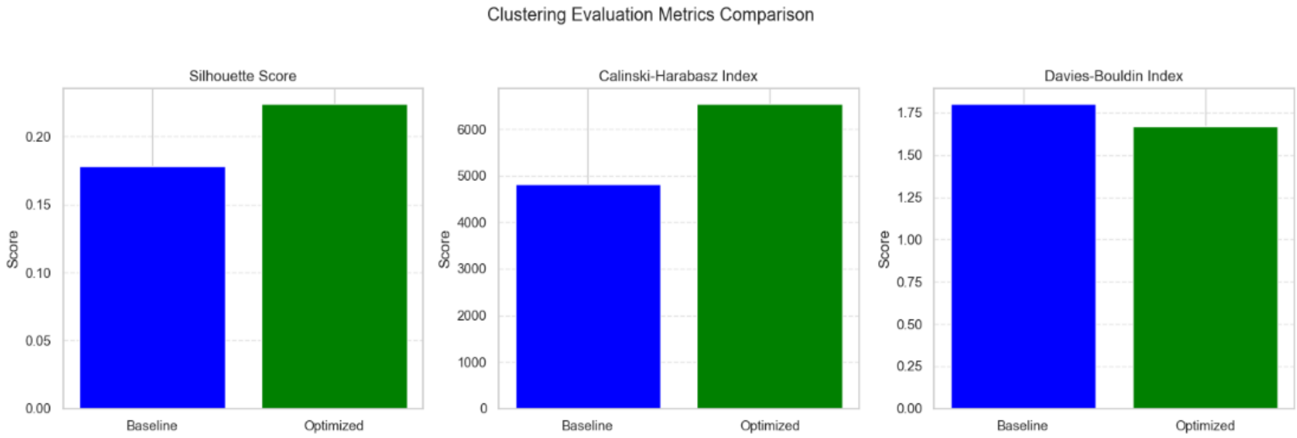


Figure 30 - Optimized K-Means Clustering Evaluation Metrics Comparison

Comparison Analysis with non-optimized approach: The baseline K-means model with a fixed number of clusters and initialization method served as a reference point whereas the genetic algorithm explored various configurations and led to the discovery of a better-performing model. The key differences were

- **Parameter Selection**: The baseline model used arbitrary parameter values, while the optimized model's parameters were the result of a systematic search.
- **Clustering Quality**: The higher Silhouette Score of the optimized model reflected clustering quality improvement.

- **Adaptability**: The Genetic Algorithm adapted to the data's inherent structure, selecting parameters that best captured the underlying patterns.

The application of a Genetic Algorithm for optimizing the K-Means clustering parameters on the 4G dataset has proven effective. By systematically exploring the parameter space, the Genetic Algorithm identified a configuration that outperforms the baseline model in terms of clustering quality (Farag *et al.*, 2015). This approach demonstrates the value of combining evolutionary algorithms with traditional clustering methods to enhance performance.

CONCLUSION

This project explored the application of machine learning techniques on telecommunication datasets to enhance understanding of user behaviour and network performance in both 4G and 5G environments. By leveraging unsupervised learning for the 4G dataset and supervised learning for the 5G dataset, the analysis provided a holistic view of mobile network dynamics.

In the 4G dataset, clustering techniques particularly K-Means were applied to group similar user patterns based on network quality metrics. The use of Genetic Algorithms to optimize clustering parameters led to improved performance, as reflected in key evaluation metrics like the Silhouette score, Calinski-Harabasz index, and Davies-Bouldin index. These clusters revealed insightful distinctions in user mobility, signal strength, and time-based variations, offering practical implications for network optimization and planning.

For the 5G dataset, classification models were trained to predict network experience categories using metrics such as SINR, RSRP, and speed. Both binary and multi-class classification approaches achieved strong performance, supported by detailed feature importance analysis that highlighted critical factors influencing network quality. These models demonstrated the potential for predictive analytics in maintaining and enhancing user experience in next-generation networks (Raca *et al.*, 2020).

Overall, the project emphasized the importance of thorough data preprocessing, thoughtful model selection, and evaluation-driven optimization. By combining exploratory analysis, clustering, classification, and evolutionary algorithms, the study delivered a well-rounded machine-learning pipeline applicable to real-world telecommunication challenges (Zanouda *et al.*, 2024). The insights gained not only validate the modelling approaches used but also provide a strong foundation for future advancements in intelligent network management and user-centric service delivery.

REFERENCE

Ackermann, M.R. *et al.* (2012) 'Analysis of agglomerative clustering,' *Algorithmica*, 69(1), pp. 184–215. https://doi.org/10.1007/s00453-012-9717-4.

Adler, A.I. and Painsky, A. (2022) 'Feature Importance in Gradient Boosting Trees with Cross-Validation Feature Selection,' *Entropy*, 24(5), p. 687. https://doi.org/10.3390/e24050687.

Ahmed, M., Seraj, R. and Islam, S.M.S. (2020) 'The k-means Algorithm: A Comprehensive Survey and Performance Evaluation,' *Electronics*, 9(8), p. 1295. https://doi.org/10.3390/electronics9081295.

Ahsan, M. *et al.* (2021) 'Effect of data scaling methods on machine learning algorithms and model performance,' *Technologies*, 9(3), p. 52. https://doi.org/10.3390/technologies9030052.

Ali, U. *et al.* (2022) 'Large-Scale dataset for the analysis of Outdoor-to-Indoor propagation for 5G Mid-Band operational networks,' *Data*, 7(3), p. 34. https://doi.org/10.3390/data7030034.

AlSagri, H. and Ykhlef, M. (2020) 'Quantifying Feature Importance for Detecting Depression using Random Forest,' *International Journal of Advanced Computer Science and Applications*, 11(5). https://doi.org/10.14569/ijacsa.2020.0110577.

Amato, A. and Di Lecce, V. (2023) 'Data preprocessing impact on machine learning algorithm performance,' *Open Computer Science*, 13(1). https://doi.org/10.1515/comp-2022-0278.

Boukerche, A., Zheng, L. and Alfandi, O. (2020) 'Outlier detection,' *ACM Computing Surveys*, 53(3), pp. 1–37. https://doi.org/10.1145/3381028.

Çetin, V. and Yıldız, O. (2022) 'A comprehensive review on data preprocessing techniques in data analysis,' *Pamukkale University Journal of Engineering Sciences*, 28(2), pp. 299–312. https://doi.org/10.5505/pajes.2021.62687.

Chaudhary, A., Kolhe, S. and Kamal, R. (2016) 'An improved random forest classifier for multi-class classification,' *Information Processing in Agriculture*, 3(4), pp. 215–222. https://doi.org/10.1016/j.inpa.2016.08.002.

Deng, D. (2020) 'DBSCAN Clustering Algorithm based on density,' *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pp. 949–953. https://doi.org/10.1109/ifeea51475.2020.00199.

Ding, C. and He, X. (2004) 'K-means clustering via principal component analysis,' *ACM DL*, p. 29. https://doi.org/10.1145/1015330.1015408.

Easaw, N. *et al.* (2023) 'Estimation of correlation matrices from limited time series data using machine learning,' *Journal of Computational Science*, 71, p. 102053. https://doi.org/10.1016/j.jocs.2023.102053.

Farag, M. *et al.* (2015) 'Genetic algorithm based on K-Means-Clustering technique for multi-objective resource allocation problems,' *British Journal of Applied Science & Technology*, 8(1), pp. 80–96. https://doi.org/10.9734/bjast/2015/16570.

FY, O. *et al.* (2017) 'Supervised Machine Learning Algorithms: Classification and comparison,' *International Journal of Computer Trends and Technology*, 48(3), pp. 128–138. https://doi.org/10.14445/22312803/ijctt-v48p126.

Gallaugher, M.P.B. *et al.* (2020) 'Skewed distributions or transformations? Modelling skewness for a cluster analysis,' *arXiv (Cornell University)* [Preprint]. https://doi.org/10.48550/arxiv.2011.09152.

Gogebakan, M. (2021) 'A novel approach for Gaussian mixture model clustering based on soft computing method,' *IEEE Access*, 9, pp. 159987–160003. https://doi.org/10.1109/access.2021.3130066.

Grandini, M., Bagli, E. and Visani, G. (2020) 'Metrics for Multi-Class Classification: an Overview,' *arXiv (Cornell University)* [Preprint]. https://doi.org/10.48550/arxiv.2008.05756.

Kousias, K. *et al.* (2023) 'A Large-Scale dataset of 4G, NB-IoT, and 5G Non-Standalone network measurements,' *IEEE Communications Magazine*, 62(5), pp. 44–49. https://doi.org/10.1109/mcom.011.2200707.

Ma, B., Guo, W. and Zhang, J. (2020) 'A survey of Online Data-Driven Proactive 5G Network Optimisation using Machine Learning,' *IEEE Access*, 8, pp. 35606–35637. https://doi.org/10.1109/access.2020.2975004.

Molin, S. (2021) *Hands-On Data Analysis with Pandas - Second Edition: A Python Data Science Handbook for Data Collection, Wrangling, Analysis, and Visualization*.

Na, S., Xumin, L. and Yong, G. (2010) 'Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm,' *IEEE Xplore*, pp. 63–67. https://doi.org/10.1109/iitsi.2010.74.

Natekin, A. and Knoll, A. (2013) 'Gradient boosting machines, a tutorial,' *Frontiers in Neurorobotics*, 7. https://doi.org/10.3389/fnbot.2013.00021.

Nazari, Z. *et al.* (2015) 'A new hierarchical clustering algorithm,' *IEEE Explore* [Preprint]. https://doi.org/10.1109/iciibms.2015.7439517.

Nohara, Y. *et al.* (2021) 'Explanation of machine learning models using shapley additive explanation and application for real data in hospital,' *Computer Methods and Programs in Biomedicine*, 214, p. 106584. https://doi.org/10.1016/j.cmpb.2021.106584.

Ontivero-Ortega, M. *et al.* (2017) 'Fast Gaussian Naïve Bayes for searchlight classification analysis,' *NeuroImage*, 163, pp. 471–479. https://doi.org/10.1016/j.neuroimage.2017.09.001.

Pal, M. (2005) 'Random forest classifier for remote sensing classification,' *International Journal of Remote Sensing*, 26(1), pp. 217–222. https://doi.org/10.1080/01431160412331269698.

Patro, V.M. and Patra, M.R. (2014) 'Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy,' *Transactions on Machine Learning and Artificial Intelligence*, 2(4). https://doi.org/10.14738/tmlai.24.328.

Petkus, T. *et al.* (2017) 'Quality Assessment of High-Resolution Images with Small Distortions after Compression,' *Baltic Journal of Modern Computing*, 5(2). https://doi.org/10.22364/bjmc.2017.5.2.04.

Raca, D. *et al.* (2020) 'Beyond throughput, the next generation,' *ACM DL* [Preprint]. https://doi.org/10.1145/3339825.3394938.

Rodriguez, M.Z. *et al.* (2019) 'Clustering algorithms: A comparative approach,' *PLoS ONE*, 14(1), p. e0210236. https://doi.org/10.1371/journal.pone.0210236.

Roy, A. (2024) 'Data visualisation to understand how data is structured using K-Means and hierarchical cluster analyses with interactive graphics,' *IOSR Journal of Applied Geology and Geophysic*, 12(6), pp. 01–05. https://doi.org/10.9790/0990-1206010105.

Saraswat, P. and Raj, S. (2022) 'DATA PRE-PROCESSING TECHNIQUES IN DATA MINING: a REVIEW,' *International Journal of Innovative Research in Computer Science & Technology*, pp. 122–125. https://doi.org/10.55524/ijircst.2022.10.1.22.

Saravanan, R. and Sujatha, P. (2018) 'A State of art Techniques on Machine Learning Algorithms: A perspective of Supervised learning Approaches in Data Classification,' *IEEE Explore*, pp. 945–949. https://doi.org/10.1109/iccons.2018.8663155.

Sinaga, K.P. and Yang, M.-S. (2020) 'Unsupervised K-Means clustering algorithm,' *IEEE Access*, 8, pp. 80716–80727. https://doi.org/10.1109/access.2020.2988796.

Starovoitov, V.V. and Golub, Yu.I. (2020) 'Comparative study of quality estimation of binary classification,' *Informatics*, 17(1), pp. 87–101. https://doi.org/10.37661/1816-0301-2020-17-1-87-101.

Trajdos, P. and Kurzynski, M. (2016) 'A dynamic model of classifier competence based on the local fuzzy confusion matrix and the random reference classifier,' *International Journal of Applied Mathematics and Computer Science*, 26(1), pp. 175–189. https://doi.org/10.1515/amcs-2016-0012.

Vujovic, Ž.Đ. (2021) 'Classification model evaluation metrics,' *International Journal of Advanced Computer Science and Applications*, 12(6). https://doi.org/10.14569/ijacsa.2021.0120670.

Zanouda, T. *et al.* (2024) 'Telecom Foundation Models: applications, challenges, and future trends,' *arXiv (Cornell University)* [Preprint]. https://doi.org/10.48550/arxiv.2408.03964.

Zhang, Y. *et al.* (2021) 'Gaussian Mixture Model Clustering with Incomplete Data,' *ACM Transactions on Multimedia Computing Communications and Applications*, 17(1s), pp. 1–14. https://doi.org/10.1145/3408318.