Machine Learning-Based Analysis of Radio Coverage Information and Network Scenarios for LTE Optimisation

By

Adepoju, Akinlolu

Table of Contents

List of Abbreviations 4

List of Figures 5

List of Tables 7

1.0 Introduction 8

1.1 Background 8

1.2 Dataset Description 9

2.0 Data Preprocessing and Exploration 11

2.1 Overview of the Preprocessing Process 12

2.2 Data Loading and Initial Inspection 12

2.3 Handling Duplicate and Missing Values 13

2.4 Outlier Detection 14

2.5 Exploratory Data Analysis (EDA) 20

2.5.1 Correlation Analysis 20

2.5.2 Pairplots and Scenario-Based Visualizations 21

Machine Learning Techniques 30

3.0 Clustering Analysis 31

3.3 K-Means Clustering Analysis 34

3.3.1 Determining the Optimal Number of Clusters 34

3.3.2 Implementation and Results 35

3.4 DBSCAN Clustering Analysis 38

3.4.1 Parameter Selection Using k-Distance Graph 38

3.4.2 Implementation and Results 39

3.5 Evaluation of Clustering Performance 41

4.0 Classification 42

4.1 Target Variable Selection 42

4.2 Data Preprocessing and Data Splitting 43

4.3 Binary Classification Analysis 48

4.3.1 Model Selection and Training 48

4.3.2 Evaluation Metrics and Results 48

4.4 Multi-Class Classification Analysis 54

4.4.1 Model Selection and Training 54

4.3.2 Evaluation Metrics and Results 55

5.0 Optimisation Using Genetic Algorithms 60

5.1 Optimisation for Classification 61

5.2 Implementation of Genetic Algorithms 61
5.3 Comparison of Optimised and Non-Optimised Models 62
Conclusion 66
References 67

List of Abbreviations

ANOVA - Analysis of Variance

AUC-ROC - Area Under the Receiver Operating Characteristic Curve

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

EDA - Exploratory Data Analysis

F-Statistic - Fisher's Statistic

FN - False Negative

FP - False Positive

GA - Genetic Algorithm

IQR - Interquartile Range

KDE - Kernel Density Estimate

KPI - Key Performance Indicator

LTE - Long-Term Evolution

ML - Machine Learning

MNC - Mobile Network Code

NR - New Radio

PCA - Principal Component Analysis

PCI - Physical Cell Identifier

RF - Random Forest

RSRP - Reference Signal Received Power

RSRQ - Reference Signal Received Quality

SINR - Signal-to-Interference-plus-Noise Ratio

SSE - Sum of Squared Errors

SVM - Support Vector Machine

TN - True Negative

TP - True Positive

XGBoost - Extreme Gradient Boosting

List of Figures

Figure 1.1: Machine Learning Workflow 9
Figure 2.1: Data Preprocessing and Exploration Workflow 11
Figure 2.2: Missing Data Percentage by Feature 13
Figure 2.3: Histograms and KDE Curves for Selected Numerical Features 14
Figure 2.4: Boxplots Illustrating Outlier Detection using the IQR Method 16
Figure 2.5: Boxplots Illustrating Outlier Detection using the Z-score Method 17
Figure 2.6: Correlation Matrix Heatmap Displaying Pearson Correlation Coefficients among Variables 21
Figure 2.7a – 2.7c: Pairplots of Selected Variables 22
Figure 2.8: RSRP Distribution Across Bands 24
Figure 2.9: RSRP Density Comparison Across Scenarios 24
Figure 2.10: RSRP Density Across Scenarios 25
Figure 2.11: RSRP vs Speed Across Scenarios 25
Figure 2.12: RSRP Distribution By Location 26
Figure 2.13: Radio Coverage Metrics Across Different Network Scenarios 27
Figure 2.14: SINR Distribution Across Scenarios 28
Figure 2.15: Time Series Plots for Signal Metrics 29
Figure 3.1: Clustering Workflow Using K-Means and DBSCAN 31
Figure 3.2: Elbow Plot of Inertia versus Number of Clusters for K-Means 34
Figure 3.3: K-Means Clustering of RSRP against other Variables 37
Figure 3.4: k-Distance Graph with Knee Point Indicating Optimal ε for DBSCAN 38
Figure 3.5: Pie Chart Representing DBSCAN Cluster Distribution and Noise Points 39
Figure 3.6: DBSCAN Clustering of RSRP against other Variables 40
Figure 4.1: Supervised Learning Workflow for Binary and Multi-Class LTE Classification 42
Figure 4.2: Correlation Matrix Heatmap Displaying Pearson Correlation Coefficients among Variables 44
Figure 4.3: Class Distribution of Target Variable 47
Figure 4.4: Model Performance Comparison for Binary Classification 50
Figure 4.5: Confusion Matrix for Binary Classification 51
Figure 4.6: ROC Curve for Binary Classification 52
Figure 4.7: Feature Importance in Logistic Regression for Binary Classification 53
Figure 4.8: Feature Importance in Random Forest for Binary Classification 54
Figure 4.9: Random Forest ROC-AUC for Multi-Classification 56

Figure 4.10: XGBoost ROC-AUC for Multi-Classification 56
Figure 4.11: Random Forest Confusion Matrix for Multi-Classification 57
Figure 4.12: XGBoost Confusion Matrix for Multi-Classification 57
Figure 4.13: Feature Importance in Random Forest for Multi-Classification 59
Figure 4.14: Feature Importance in XGBoost for Multi-Classification 59
Figure 5.1: Optimisation Workflow for Hyperparameter Tuning Using Genetic Algorithms 60
Figure 5.2: Evolution of Accuracy Over Generations (GA) 62
Figure 5.3: Performance Improvement After GA 63
Figure 5.4: Baseline Random Forest Confusion Matrix 64
Figure 5.5: Optimised Random Forest Confusion Matrix 64
Figure 5.6: Silhouette Scores for K-Means Clustering 65

List of Tables

Table 1.1: Passive dataset features with a short description 10
Table 2.1: Skewness and Kurtosis of Variables 15
Table 2.2: Number of outliers detected and their respective percentages in each variable 18
Table 2.3: Summary Statistics 19
Table 3.1: Information on the experimental environment 30
Table 3.2: Summary Statistics (Scaled) 33
Table 3.3: Summary Statistics (Unscaled) 33
Table 3.4: Cluster Centres in Scaled Format 35
Table 4.1: ANOVA Results 45
Table 4.2: Summary of Key Variables 46
Table 4.3: First six rows of RSRP and RSRP_CLASS columns 47
Table 4.4: Logistic Regression Classification Report 48
Table 4.5: Random Forest Classification Report 49

1.0 Introduction

This section outlines the background and purpose of applying machine learning (ML) to analyse radio coverage and network scenarios for LTE optimisation. It highlights the relevance of signal metrics in mobile networks and introduces the dataset used to extract performance insights.

1.1 Background

As mobile networks expand, managing 4G LTE and 5G NR performance has become increasingly complex. The surge in user demand and coverage requires data-driven solutions. ML enables the analysis of large-scale measurement data to detect patterns, predict performance degradation, and support proactive network management (Zheng *et al*., 2016; Boutaba *et al*., 2018). This study applies ML to optimise LTE performance through the analysis of RSRP and scenario data. The key objectives include data preprocessing for quality assurance, clustering with K-Means and DBSCAN to identify signal patterns, developing binary and multi-class classification models for performance prediction, and optimising model accuracy using Genetic Algorithms. These techniques collectively transform complex network data into actionable insights, as shown in Figure 1.1.

Clustering models will identify patterns of weak signals or interference, while classification models will categorise signal strength. Both will be evaluated using metrics like the silhouette score, accuracy, and F1 score. Interpretability tools, such as feature importance, enhance insights to support targeted, data-driven network performance optimisation.

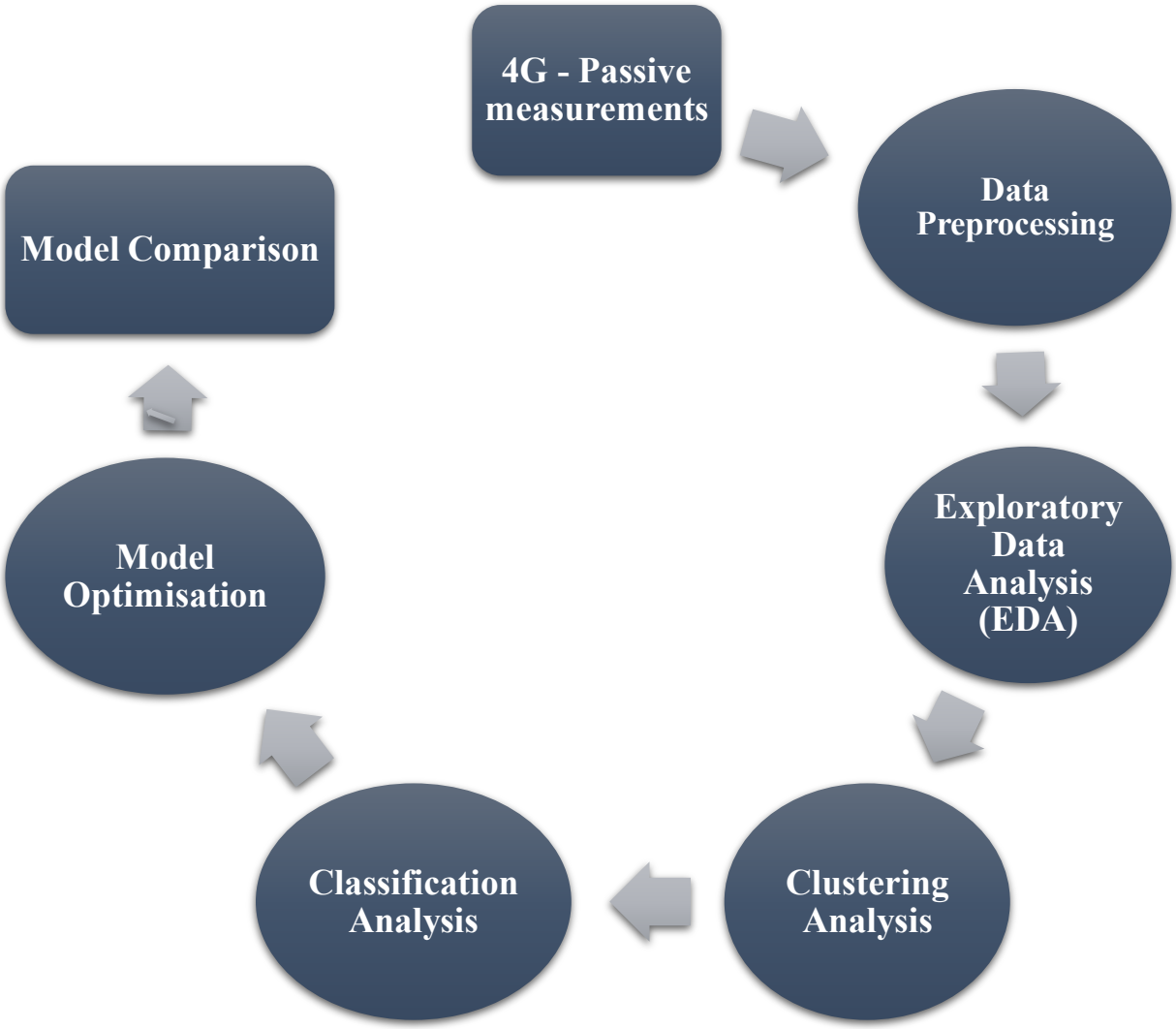


Figure 1.1: Machine Learning Workflow

1.2 Dataset Description

The 4G Passive Measurement dataset was selected and utilised for all tasks due to its inclusion of quantitative (e.g., Altitude, Speed, RSRP, SINR, Distance) and qualitative (e.g., PCI, MNC, scenarios, campaigns, bands) features, offering a comprehensive view of LTE network performance. These variables support the assessment of signal strength, mobility, and user experience, making the dataset ideal for applying machine learning techniques (Kousias *et al*., 2023; Wang *et al*., 2023; 5GWorldPro.com, 2022). Its detail enables robust analysis of RSRP variations across scenarios and supports clustering and classification for network optimisation. Table 1.1 outlines the dataset's key variables.

Table 1.1: Passive dataset features with a short description.

Feature	Description
Date, Time	Temporal fields indicating the timestamp of each measurement.
UTC	Coordinated Universal Time of the measurement.
Latitude, Longitude	Geolocation coordinates of the mobile device at the time of measurement.
Altitude	Elevation of the device above sea level (in meters).
Speed	The movement speed of the device (in km/h), useful for analysing mobility effects.
EARFCN	E-UTRA Absolute Radio Frequency Channel Number representing the LTE frequency.
Frequency	Carrier frequency of the mobile network (in MHz).
PCI	Physical Cell Identifier, used to differentiate cells on the same frequency.
MNC	Mobile Network Code, identifying the network operator.
CellIdentity	Unique identifier of the cell serving the device.
eNodeB.ID	Identifier for the eNodeB (base station) serving the device.
Power	Received signal power level (in dBm), indicating signal strength.
SINR	Signal-to-Interference-plus-Noise Ratio in dB, reflecting signal quality.
RSRP	Reference Signal Received Power in dBm, representing LTE signal strength.
RSRQ	Reference Signal Received Quality in dB, indicating the quality of the received LTE signal.
Scenario	The environment where measurements were taken (e.g., indoor, or outdoor).
cellLongitude, cellLatitude	Geolocation of the serving cell (base station).
cellPosErrorLambda1, cellPosErrorLambda2	Estimated positioning error margins for the cell location.
n_CellIdentities	The number of cells detected by the device.
Distance	Distance (in meters) between the device and the serving cell.
Band	The frequency band used for the connection.
Campaign	Identifier for different measurement campaigns, distinguishing between different data collection events.

The next sections will explore data preprocessing, exploratory data analysis, and the application of clustering and classification techniques for network optimisation.

2.0 Data Preprocessing and Exploration

This section presents an in-depth discussion of the procedures implemented to preprocess and explore the LTE network measurement dataset, as shown in the Figure below.

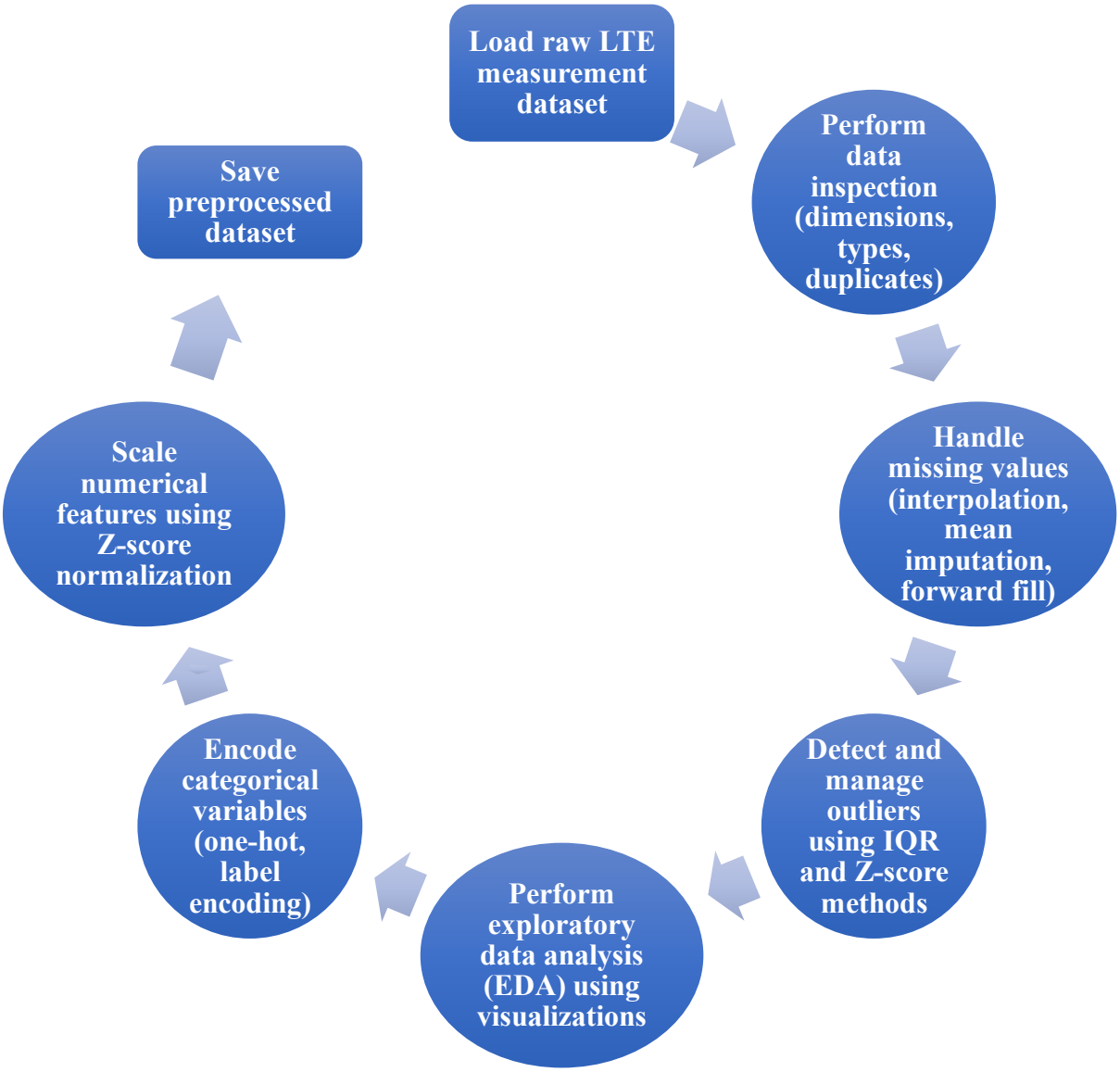


Figure 2.1: Data Preprocessing and Exploration Workflow.

2.1 Overview of the Preprocessing Process

Data preprocessing is an important preliminary step for transforming the LTE measurement data, which consists of over 527,000 entries and 27 features, into a structured format suitable for modelling (Fan *et al*., 2021). The process involves loading the data, addressing missing values and outliers, conducting exploratory analysis, and applying feature scaling. These steps improve data quality, reduce noise, and ensure that the dataset accurately reflects underlying patterns, supporting reliable machine learning performance and reducing the risk of misleading results (Fan *et al*., 2021).

2.2 Data Loading and Initial Inspection

The analysis was initiated by integrating a collection of libraries designed to support the entire machine learning workflow—from data preparation and visualisation to model training and optimisation. These libraries include NumPy and pandas which were used for efficient numerical computations and data handling. Matplotlib and seaborn were utilised to create informative data visualisations (Sundaram *et al.,* 2023). For preprocessing and model development, components from scikit-learn were extensively utilised, including StandardScaler for feature scaling, encoding techniques like LabelEncoder and OneHotEncoder, and a range of algorithms such as LogisticRegression, RandomForestClassifier, KMeans, and DBSCAN. The XGBoostClassifier was also implemented to leverage the benefits of gradient boosting. To optimise model performance, Genetic Algorithms were applied via the sklearn-genetic library, with the DEAP framework enabling flexible design of evolutionary strategies (Fortin *et al.,* 2012). Additional tools like kneed and SciPy enhanced analytical precision and model evaluation (Sundaram *et al.,* 2023).

The analysis commences by loading the dataset comprising 527,540 records and various features, including Date, Time, UTC, Latitude, Longitude, Altitude, Speed, and key performance indicators like RSRP, SINR, and RSRQ. The initial inspection involved checking the dataset's dimensions and data types, along with removing redundancy. The first column ‘Unnamed’ was dropped due to its lack of analytical value (DeCastro-García *et al*., 2018). The dataset contained 22 numerical and five categorical columns, encompassing temporal, geospatial, and signal metrics. Temporal variables (Date, Time, UTC) captured the timing of measurements, while Latitude, Longitude, and Altitude indicated the device's location.

RSRP and SINR represented signal strength and quality, whereas Speed and EARFCN provided mobility and frequency context. This heterogeneous structure necessitates specific preprocessing strategies to accommodate the unique nature of each feature (Sangeetha *et al.*, 2024).

2.3 Handling Duplicate and Missing Values

Duplicate and missing values can compromise data integrity and distort analysis outcomes. In this dataset, no duplicates were found; however, several features exhibited missing values (Kwak and Kim, 2017). Notably, UTC, Altitude, and Speed had missing rates of approximately 13.3%, while Power and SINR showed 4.1%, and cellPosErrorLambda1 had 1.5%. Figure 2.2 visualises this distribution, with missing values represented in red and complete data in green. For features with fewer than 5% missing values (Power, SINR, and cellPosErrorLambda1), listwise deletion was applied to maintain data quality with minimal loss (Kang, 2013). For features with higher missingness, imputation was employed. UTC was linearly interpolated to preserve temporal continuity (Lepot, Aubin, and Clemens, 2017). Altitude was imputed using the mean to retain distribution integrity (Lai *et al.*, 2024), and Speed was forward-filled to maintain consistency in mobility trends (Kamalov and Sulieman, 2021). Post-processing validation confirmed the complete removal of missing values, ensuring a consistent and reliable dataset for further analysis while reducing the risk of bias. This hybrid strategy balanced data retention with accuracy, supporting robust model development.

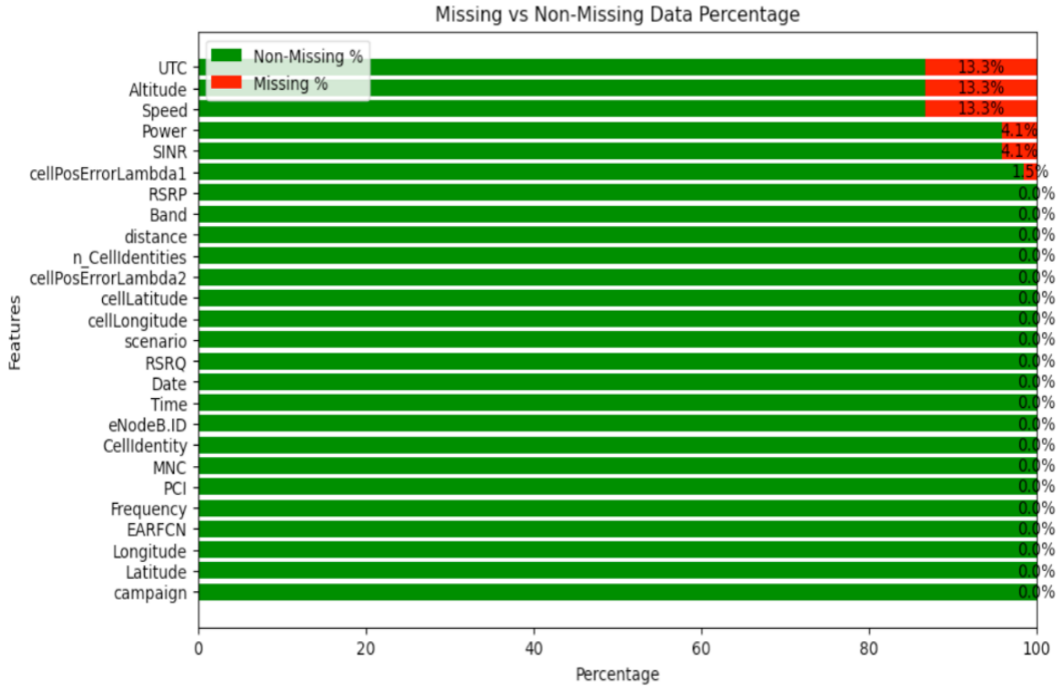


Figure 2.2: *Missing Data Percentage by Feature*

2.4 Outlier Detection

Outliers can distort statistical analyses and reduce model performance (Ododo and Addotey, 2025). This study utilised the Interquartile Range (IQR) and Z-score methods for detecting outliers based on feature distributions. IQR was applied to skewed features, while Z-score was employed for those approximating a normal distribution (Dastjerdy, Saeidi, and Heidarzadeh, 2023). The distribution characteristics of key features such as UTC, Latitude, Altitude, Speed, EARFCN, and Frequency were examined using histograms (Figure 2.3) to inform the choice of the detection method.

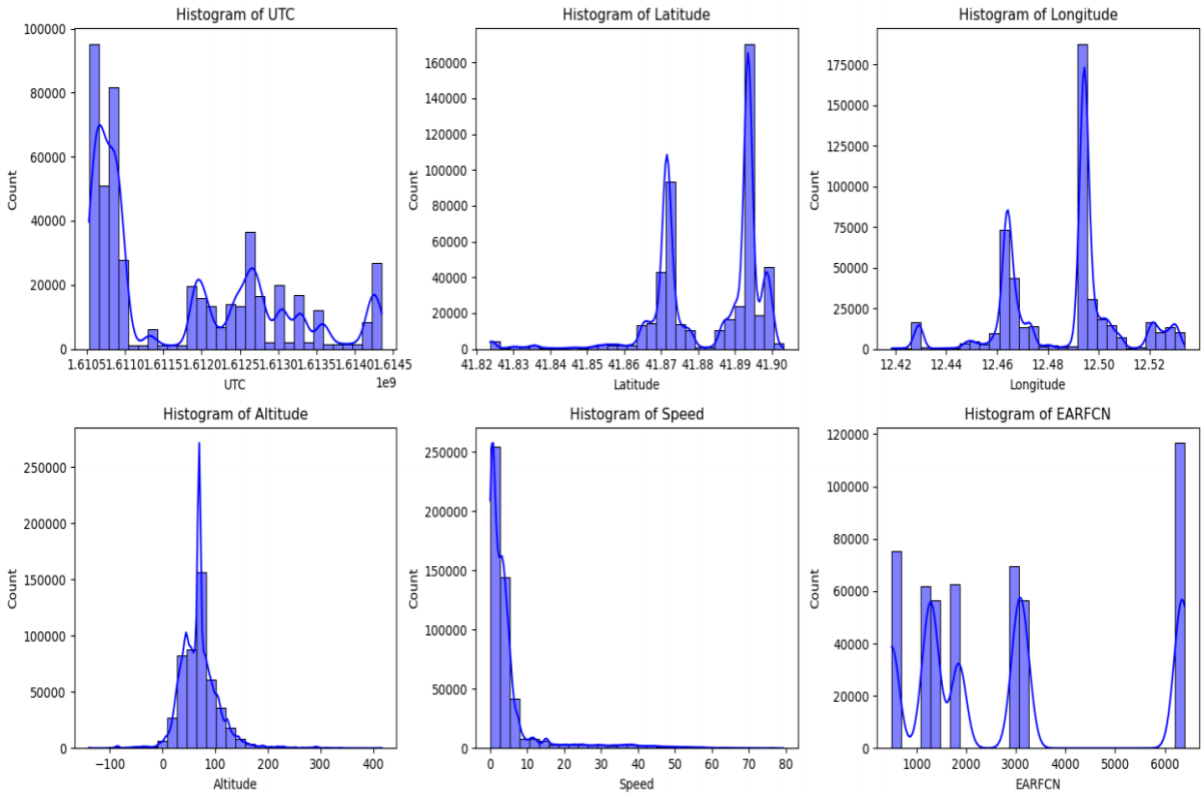


Figure 2.3: Histograms and KDE Curves for Selected Numerical Features.

Each histogram was augmented with a Kernel Density Estimate (KDE) curve, providing a continuous approximation of the underlying probability density beyond the bin-based representation. The histograms revealed substantial skewness in certain features, Speed and Altitude had skewness values exceeding 3.0 and 1.0, respectively. This visual analysis informs decisions regarding necessary transformations to approximate a normal distribution, such as log or Box-Cox transformations. The overall feature skewness was also analysed to guide the selection of an appropriate outlier detection approach. Table 2.1 summarises their skewness, kurtosis values and distribution description.

Table 2.1: Skewness and Kurtosis of Variables

Variable	Skewness	Kurtosis	Distribution Description
UTC	0.68	-0.84	Moderately skewed
Latitude	-1.22	2.12	Highly skewed
Longitude	-0.31	0.05	Approximately normal
Altitude	1.39	8.62	Highly skewed
Speed	3.66	14.71	Highly skewed
EARFCN	0.72	-0.93	Moderately skewed
Frequency	-0.45	-0.95	Approximately normal
PCI	0.29	-1.02	Approximately normal
CellIdentity	0.18	-1.95	Approximately normal
eNodeB.ID	0.18	-1.95	Approximately normal
Power	-0.02	-0.21	Approximately normal
SINR	0.05	-0.34	Approximately normal
RSRP	0.01	-0.19	Approximately normal
RSRQ	-0.48	-0.61	Approximately normal
cellLongitude	-0.39	0.18	Approximately normal
cellLatitude	-1.14	1.40	Highly skewed
cellPosErrorLambda1	1.47	3.80	Highly skewed
cellPosErrorLambda2	1.46	3.73	Highly skewed
n_CellIdentities	-0.17	-0.93	Approximately normal
distance	3.72	18.23	Highly skewed
Band	0.99	-0.68	Moderately skewed

Due to its effectiveness with non-normal distribution, the Interquartile Range (IQR) method was used to detect outliers in skewed features, applying thresholds of 1.5 and 3.0 to identify moderate and extreme values (Dastjerdy, Saeidi, and Heidarzadeh, 2023).

This classified 23.37% of EARFCN and Band entries, along with 7.09% for Speed, 7.01% for Distance, 1.79% for Altitude, and 1.59% each for cellPosErrorLambda1 and cellPosErrorLambda2 as extreme values. For features with approximately normal distributions, the Z-score method was applied, identifying minimal outliers: 0.02% for Longitude, 0.07% for Power, 0.20% for SINR, 0.06% for RSRP, 0.14% for RSRQ, and 0% for PCI, Frequency, and n_CellIdentities. Boxplots (Figures 2.4 and 2.5) were used to visualise data distribution, highlighting extreme values and potential outliers through interquartile ranges (Hu, 2020).

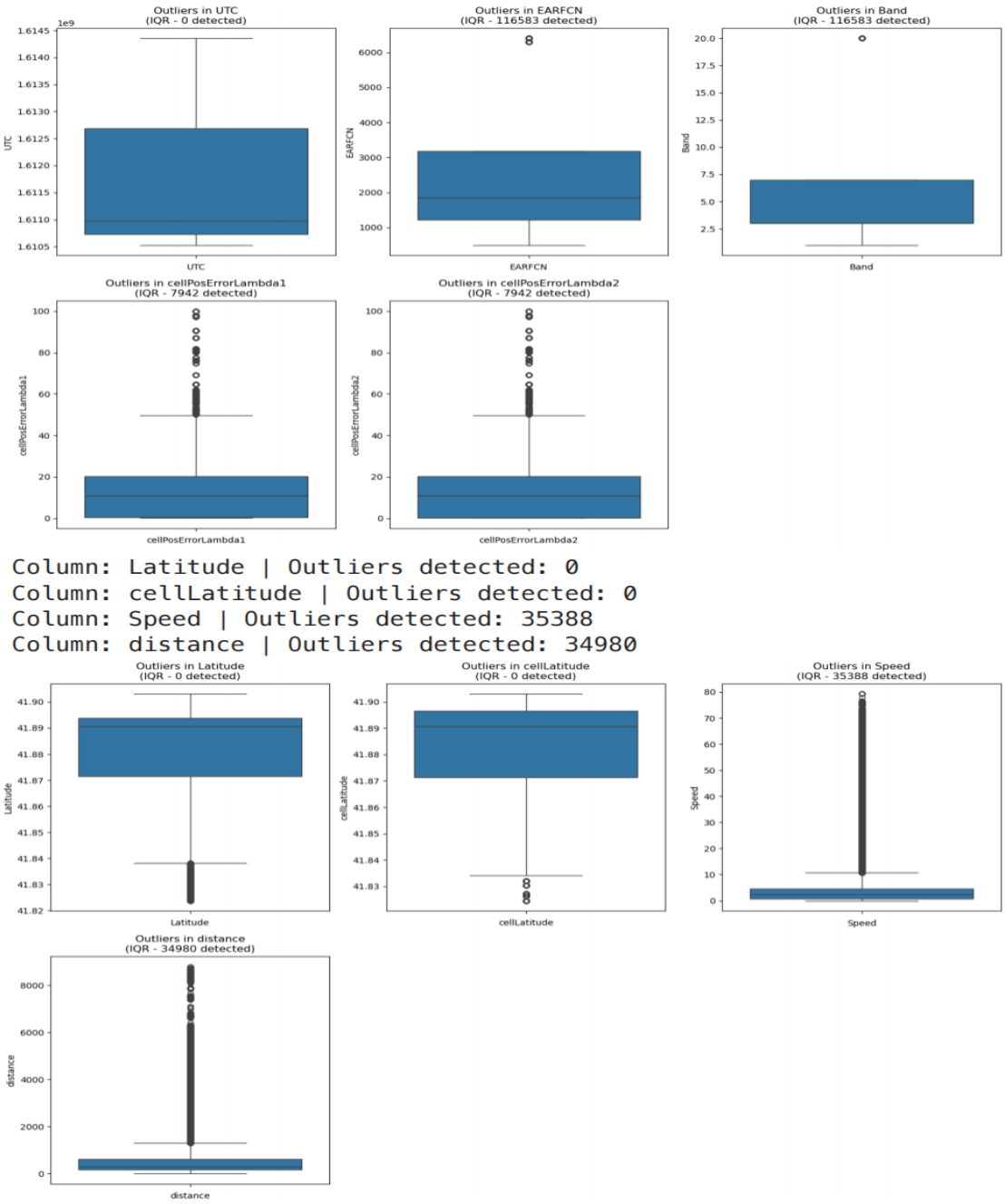


Figure 2.4: Boxplots Illustrating Outlier Detection using the IQR Method.

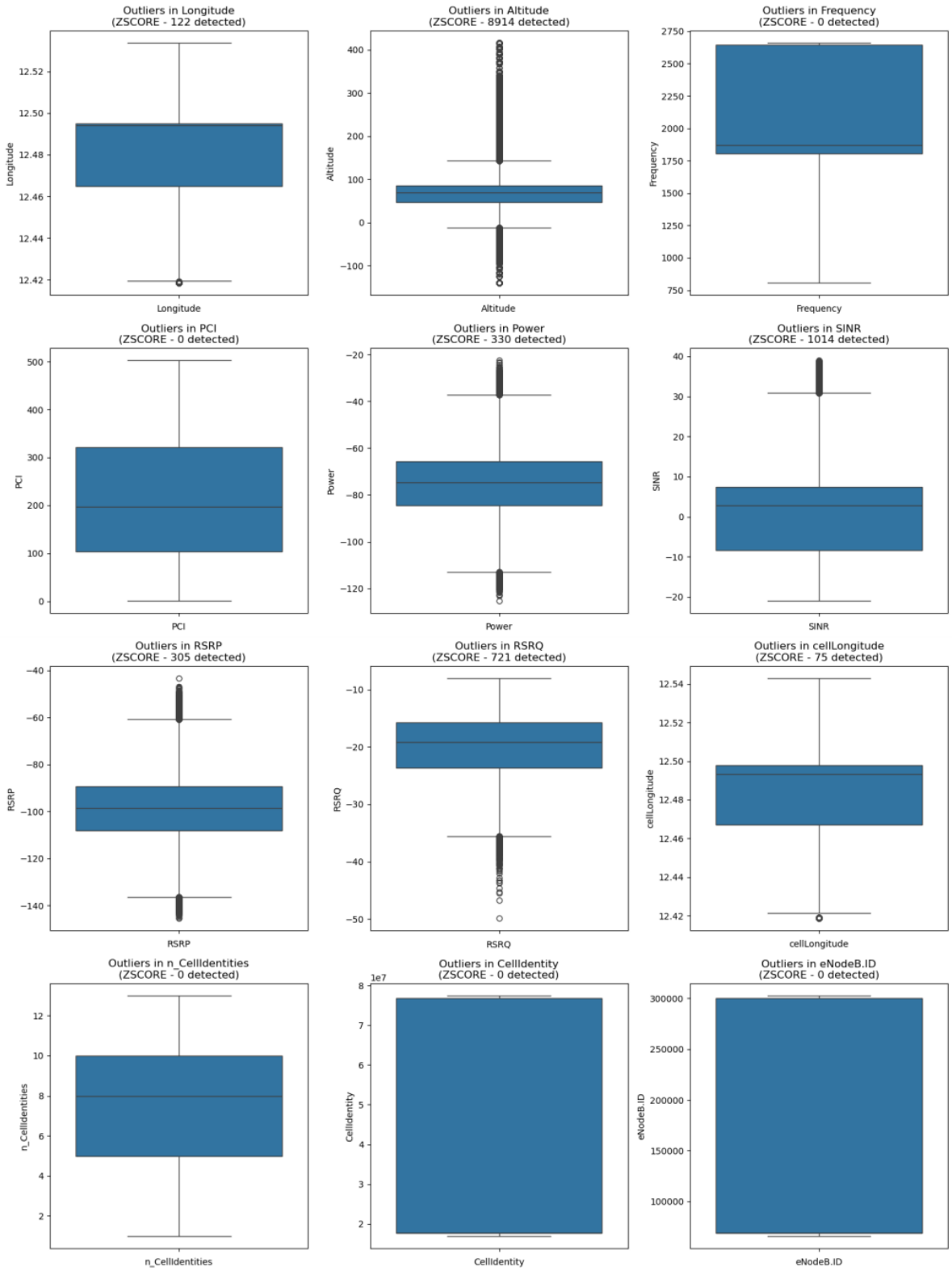


Figure 2.5: Boxplots Illustrating Outlier Detection using the Z-score Method.

Table 2.2: Number of outliers detected and their respective percentages in each variable.

Variable	**Extreme values Count**	**Extreme values Percentage**
UTC	0	0.00 %
Latitude	0	0.00 %
Longitude	122	0.02 %
Altitude	8,914	1.79 %
Speed	35,388	7.09 %
EARFCN	116,583	23.37 %
Frequency	0	0.00 %
PCI	0	0.00 %
CellIdentity	0	0.00 %
eNodeB.ID	0	0.00 %
Power	330	0.07 %
SINR	1,014	0.20 %
RSRP	305	0.06 %
RSRQ	721	0.14 %
cellLongitude	75	0.02 %
cellLatitude	0	0.00 %
cellPosErrorLambda1	7,942	1.59 %
cellPosErrorLambda2	7,942	1.59 %
n_CellIdentities	0	0.00 %
distance	34,980	7.01 %
Band	116,583	23.37 %

Figure 2.4 presents boxplot visualisations that further illustrate the presence of potential outliers in the EARFCN and Band features. However, upon closer inspection, these values were found to reflect the distinct and limited number of unique values within each column. Specifically, the Band feature comprised only four unique values: 1, 3, 7, and 20, while EARFCN contained eight unique values: 501, 1225, 1350, 1850, 3025, 3175, 6300, and 6400. These values were confirmed to be valid and representative of the underlying domain. Therefore, no outlier removal was applied to these features. Following preprocessing, the dataset size decreased to 498,948 observations, reflecting a 5.4% reduction from the initial 527,540 records. To further assess data quality and distribution, summary statistics, including mean, median, standard deviation, and quartiles, were computed for all numerical features (Cooksey, 2020). Table 2.3 presents selected feature statistics. For instance, RSRP, a key metric in LTE performance evaluation, exhibited a mean of –98.6 dBm with a standard deviation of 14.1 dBm. These statistics provide an initial quantitative understanding of central tendency and variability, informing subsequent analytical decisions.

Table 2.3: Summary Statistics

Statistic	Speed	RSRP	Band	Distance
Count	498948	498948	498948	498948
Mean	5.1186	-98.6737	7.6813	553.2473
Std	9.3063	14.1311	7.0896	785.2996
Min	0.0000	-145.5200	1.0000	0.0000
25%	0.6800	-108.1400	3.0000	158.5780
50%	2.5600	-98.7200	3.0000	288.1674
75%	4.6800	-89.2200	7.0000	617.1653
Max	79.2700	-43.1700	20.0000	8788.7241

2.5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) helps uncover relationships and patterns within the dataset through visualisations like pair plots, boxplots, and correlation matrices (Komorowski *et al.,* 2016). A duplicate of the preprocessed dataset was retained to preserve its integrity during transformation.

2.5.1 Correlation Analysis

A correlation matrix was computed using Pearson's correlation coefficient to assess the strength and direction of linear relationships between numerical features (Kuzudisli *et al*., 2023). According to Dudáš (2024), this method is widely used to assess inter-feature relationships and predictive potential, with coefficients ranging from −1 to 1 to indicate positive (variables increase together), negative (one increases as the other decreases), or no linear correlation (0). In addition to evaluating associations with the target variable, this approach reveals redundant features—those highly correlated with each other—potentially contributing little unique value. Identifying such features supports dimensionality reduction and enhances model interpretability and efficiency (Kuzudisli *et al*., 2023).

The correlation heatmap (Figure 2.6) revealed strong inter-feature relationships, notably between RSRP and Power ($r > 0.9$), and among Latitude-cellLatitude, Longitude-cellLongitude, and EARFCN-Band pairs, indicating redundancy (Pfaehler *et al*., 2021). To mitigate multicollinearity, one feature from each highly correlated pair was considered for removal. This step enhances model robustness and interpretability by ensuring non-redundant inputs and preserving unique information essential for accurate machine learning predictions (Basu and Maji, 2022).

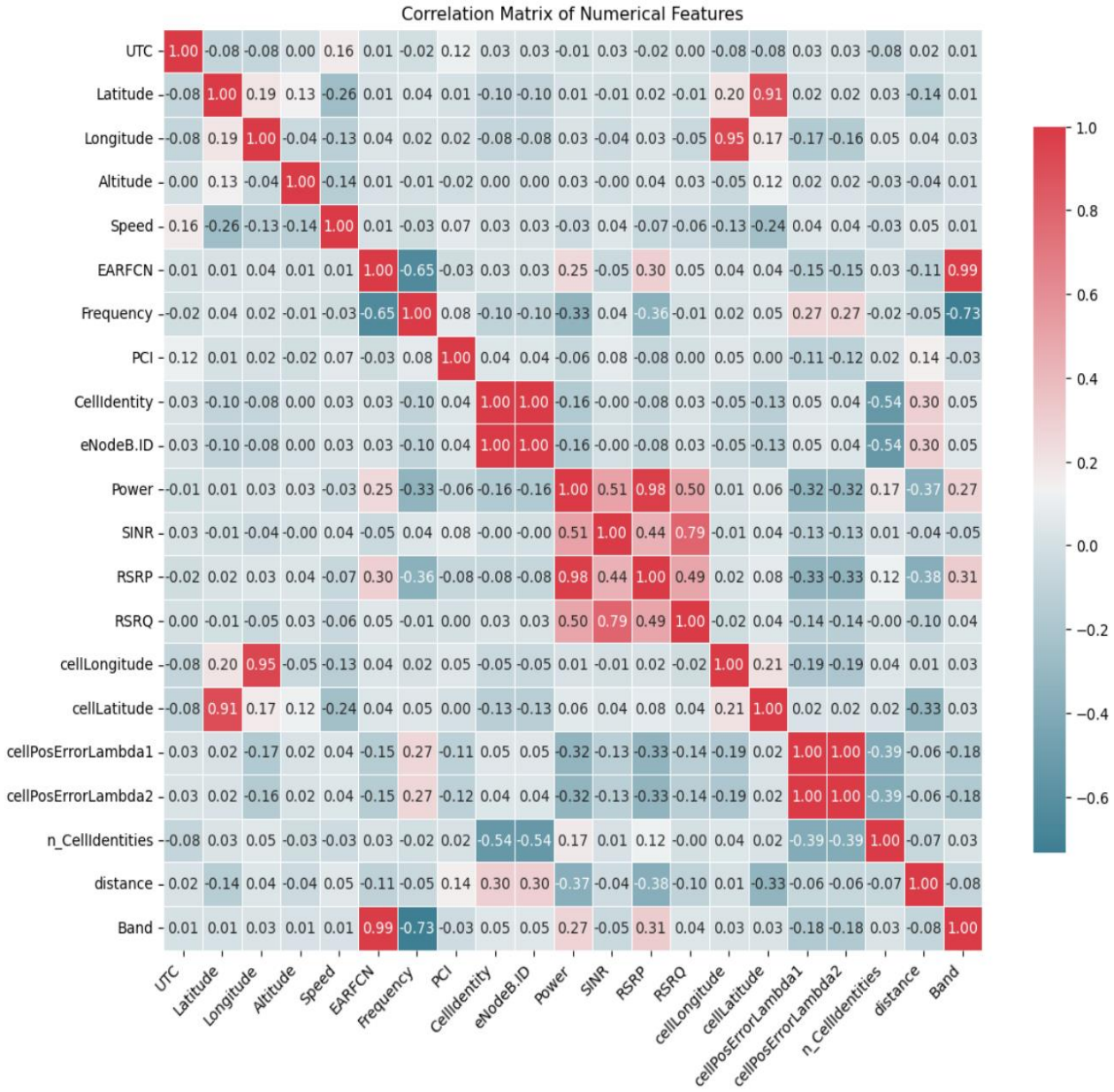


Figure 2.6: Correlation Matrix Heatmap Displaying Pearson Correlation Coefficients among Variables.

2.5.2 Pairplots and Scenario-Based Visualisations

Pair plots were generated to explore relationships between key features, providing a multidimensional view through scatterplots and Kernel Density Estimate (KDE) curves along the diagonals (Fatima, 2024). Figures 2.7a-c illustrate these plots, revealing linear, nonlinear, and null associations among the dataset's features.

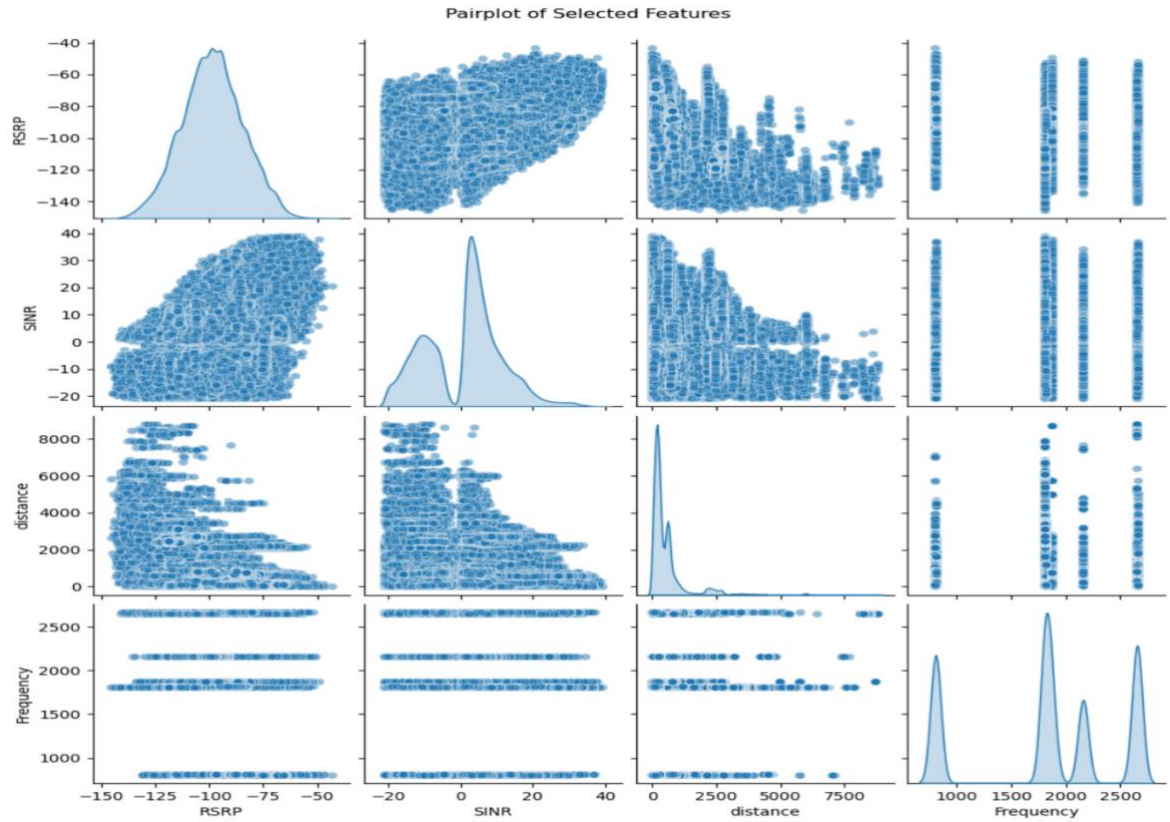


Figure 2.7a: Pairplots of Selected Variables.

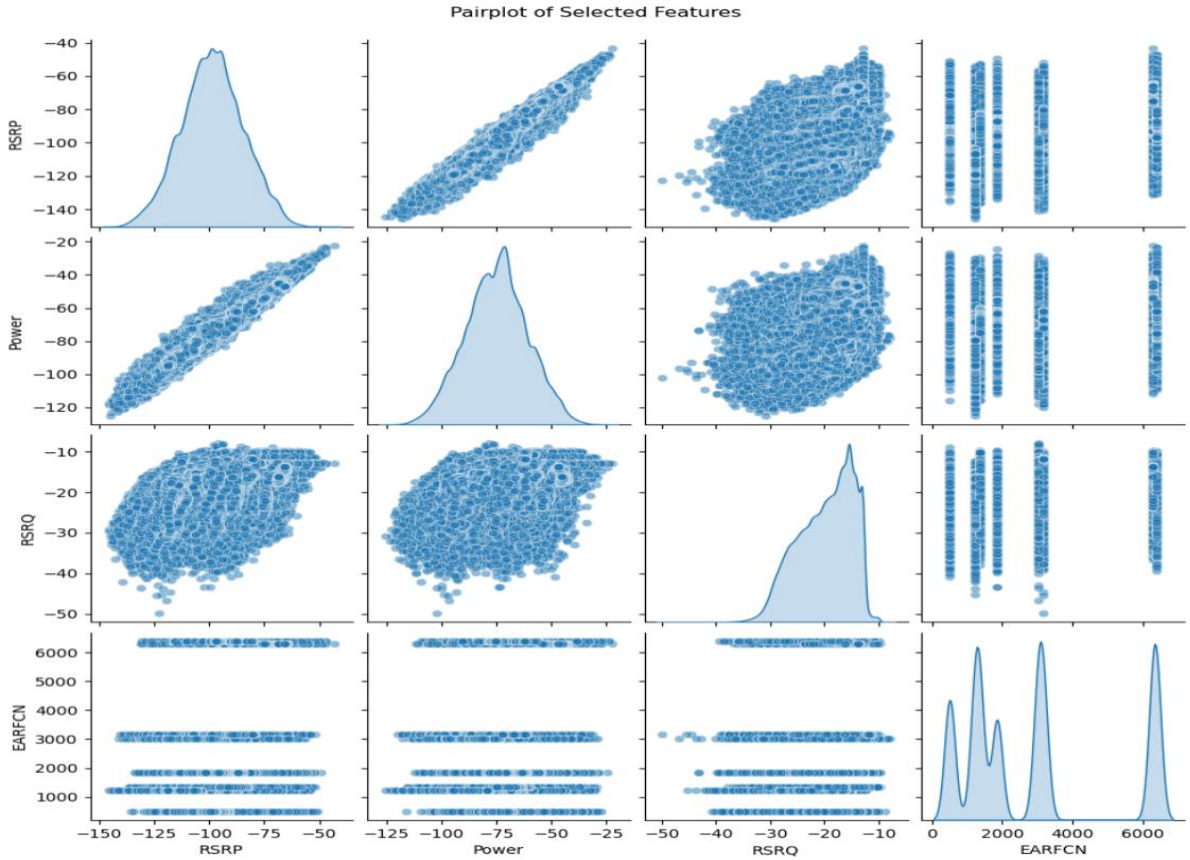


Figure 2.7b: Pairplots of Selected Variables.

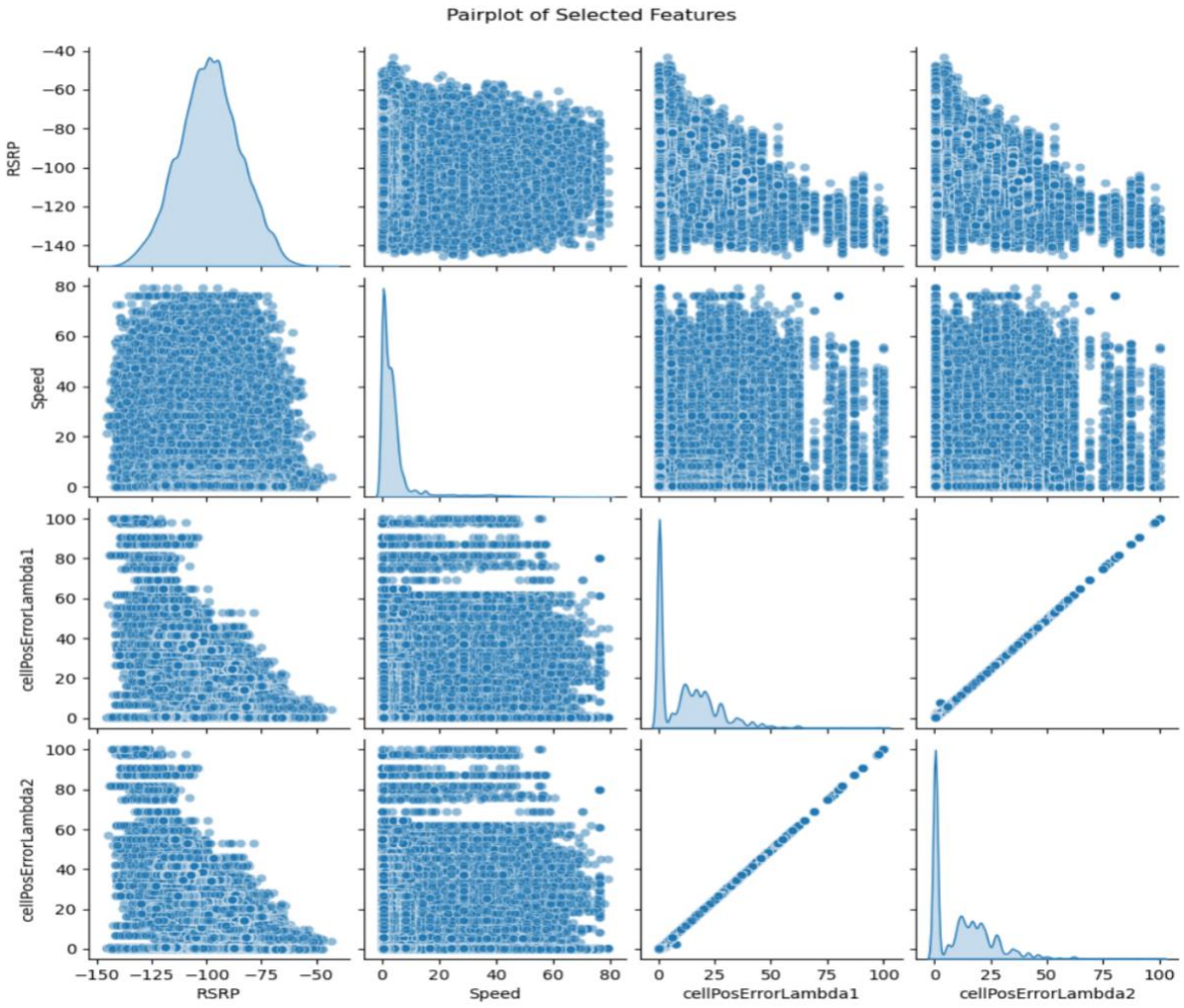


Figure 2.7c: Pairplots of Selected Variables.

Boxplots were used to visualise the distribution of RSRP values across different bands (Figure 2.8), revealing that Band 1 had a narrower interquartile range, indicating more stable signal strength. The dataset also included categorical variables representing network scenarios such as indoor static (IS), outdoor walking (OW), and outdoor driving (OD). Figures 2.9 and 2.10 display density and violin plots (Dougba-Noel *et al*., 2021), which highlight that indoor static scenarios show higher and more consistent RSRP values, while outdoor scenarios exhibit greater variability. These patterns align with findings by Noh and Choi (2019) and Kim-Geok *et al*. (2020), who reported higher signal stability in indoor environments due to reduced interference. Additionally, Figure 2.11 presents a scatterplot showing that OD scenarios are more sparsely distributed across speed, suggesting that mobility may be linked to increased RSRP variability, likely due to signal fluctuation and frequent transitions between network cells in dynamic outdoor environments.

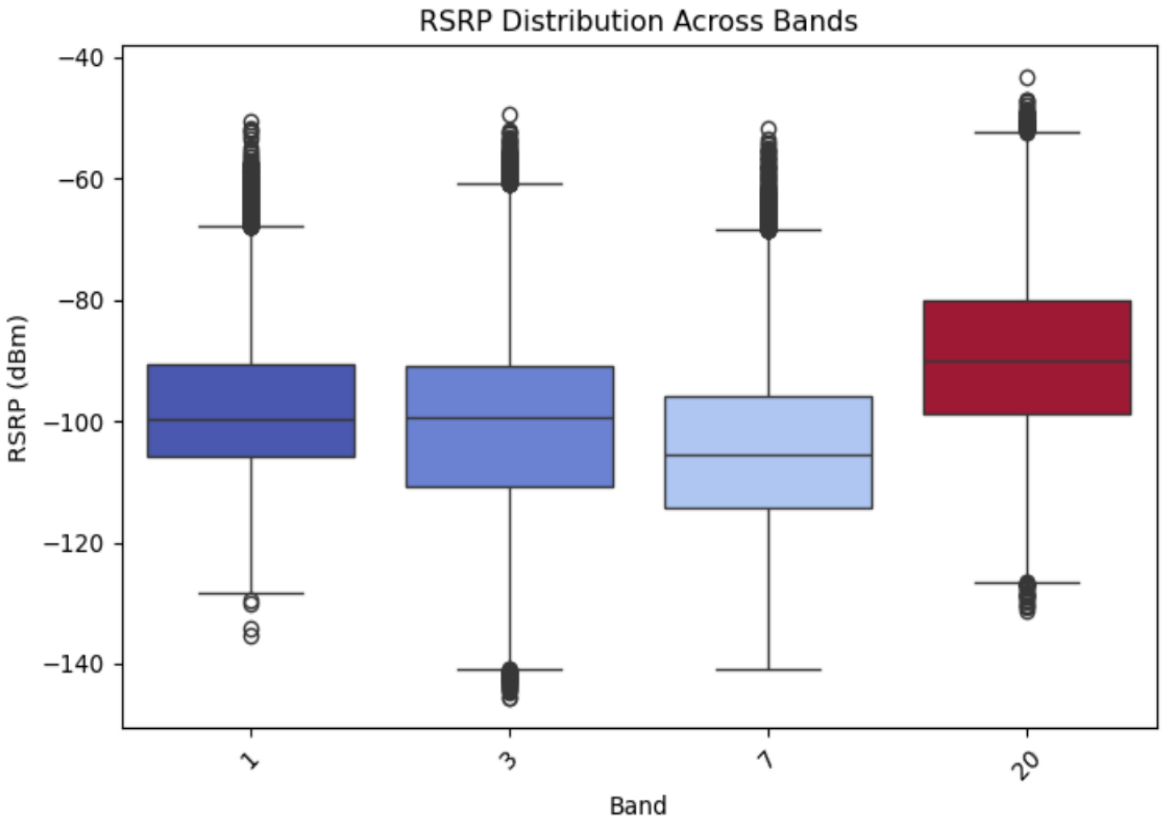


Figure 2.8: RSRP Distribution Across Bands.

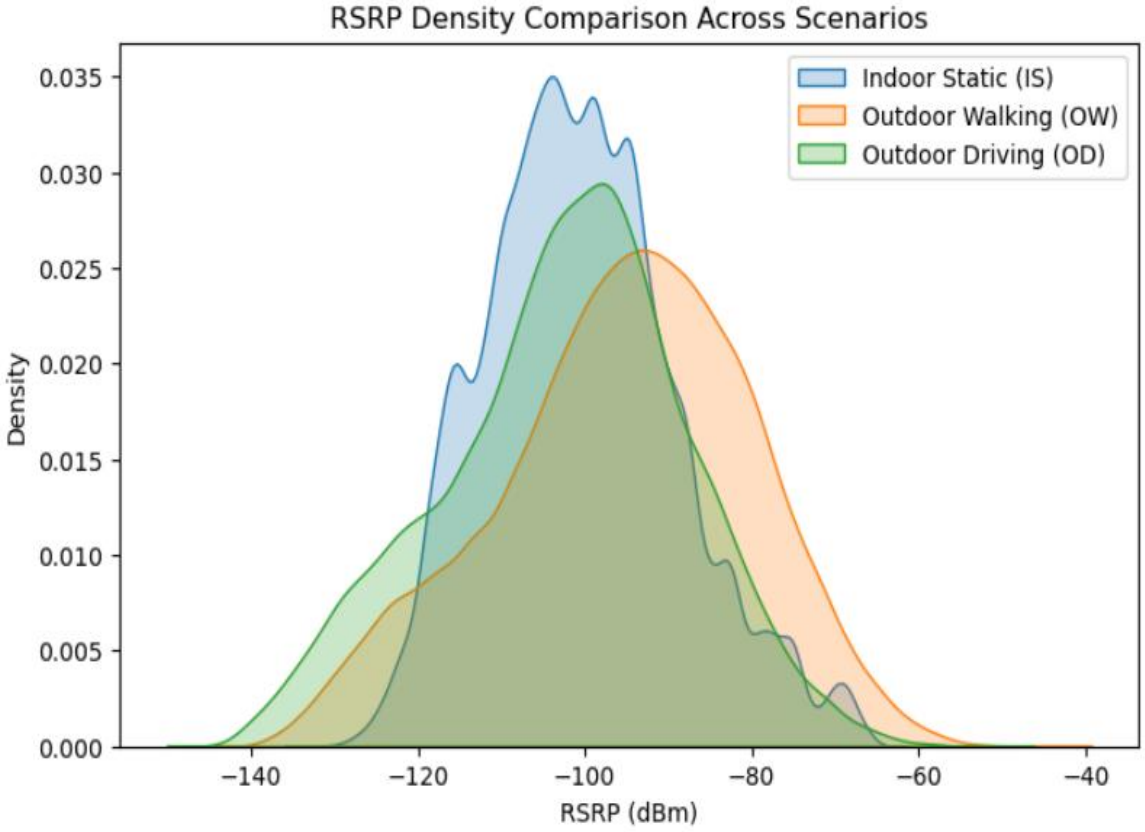


Figure 2.9: RSRP Density Comparison Across Scenarios.

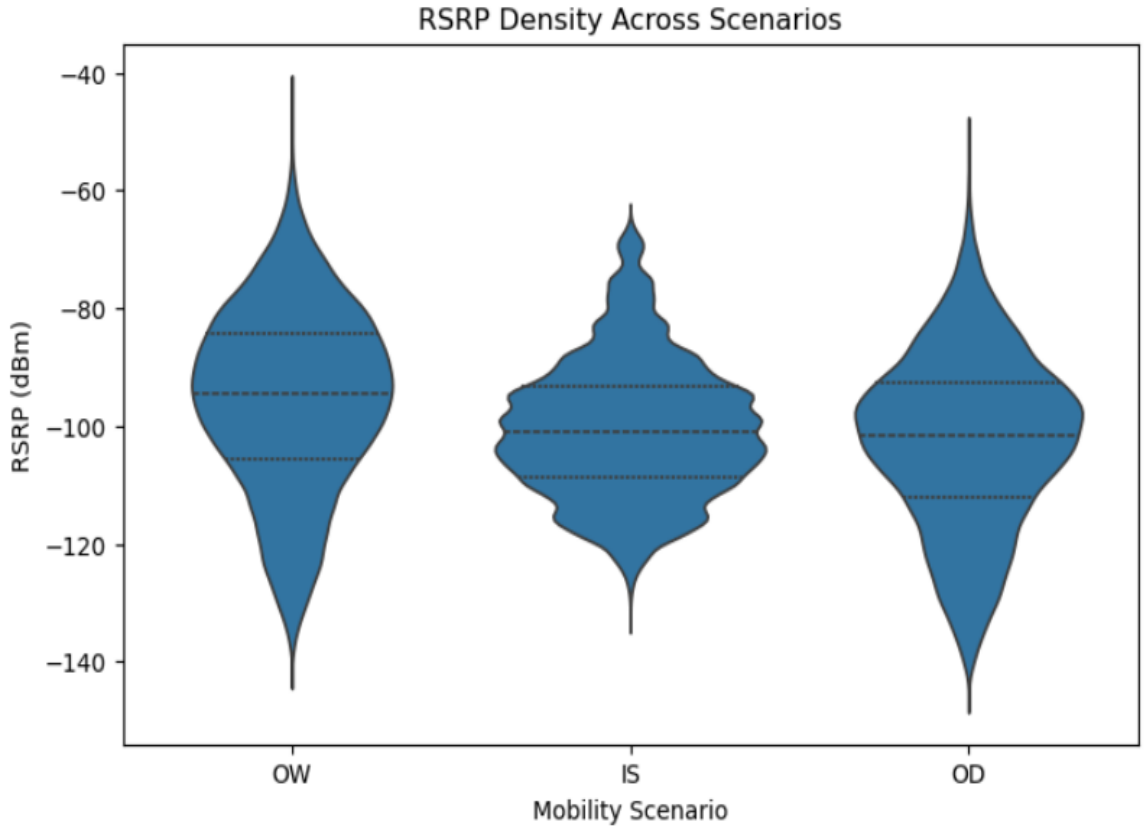


Figure 2.10: RSRP Density Across Scenarios.

Figure 2.11: RSRP vs Speed Across Scenarios.

Figure 2.12 illustrates the spatial distribution of RSRP variations across different geographic locations, providing insight into the geographic variability of signal strength. Figures 2.13 and 2.14 show box plots of key radio coverage metrics across various scenarios, highlighting central tendencies, interquartile ranges, and extreme values. Furthermore, Figure 2.15 displays time series plots for four essential signal strength and quality indicators, RSRP, RSRQ, Power, and SINR, capturing their temporal dynamics and facilitating the identification of underlying trends, periodic fluctuations, congested locations and potential anomalies over time.

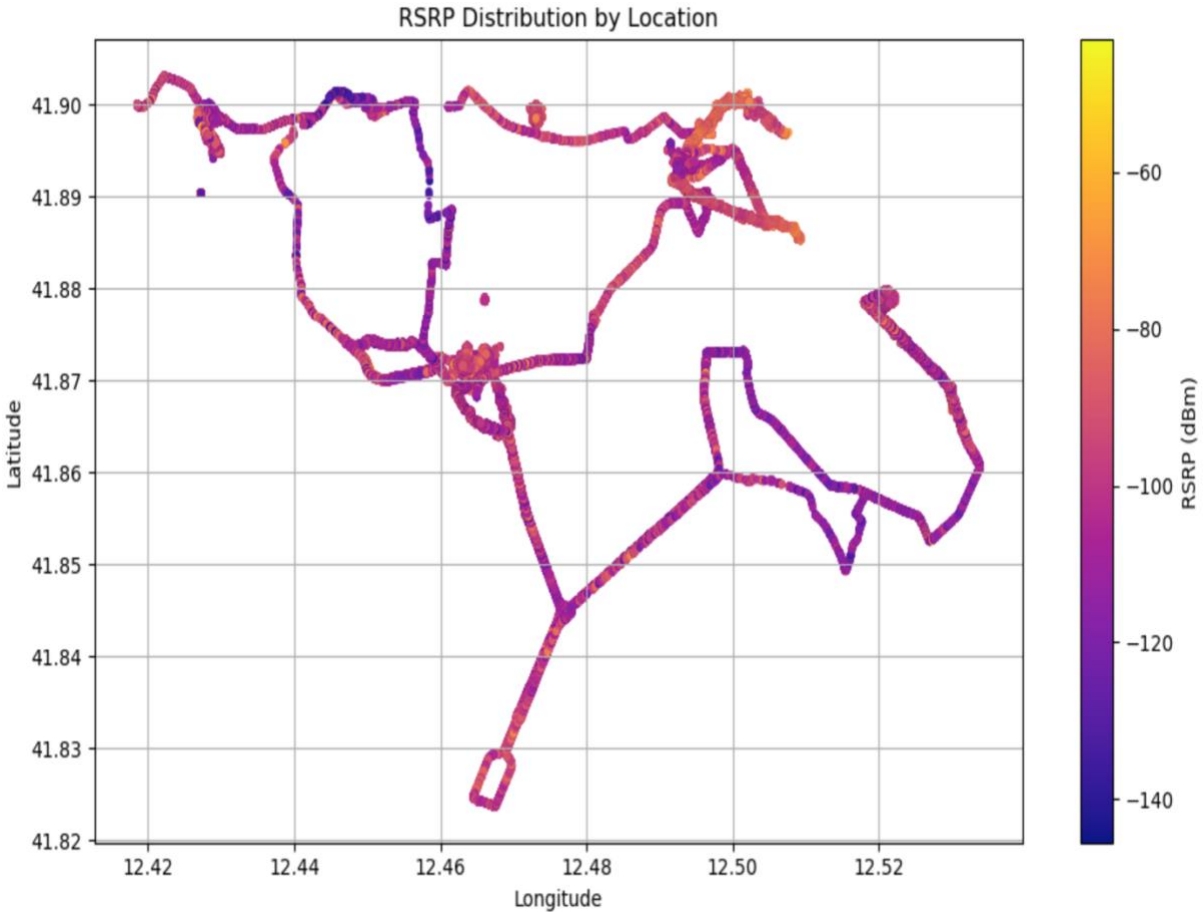


Figure 2.12: RSRP Distribution By Location.

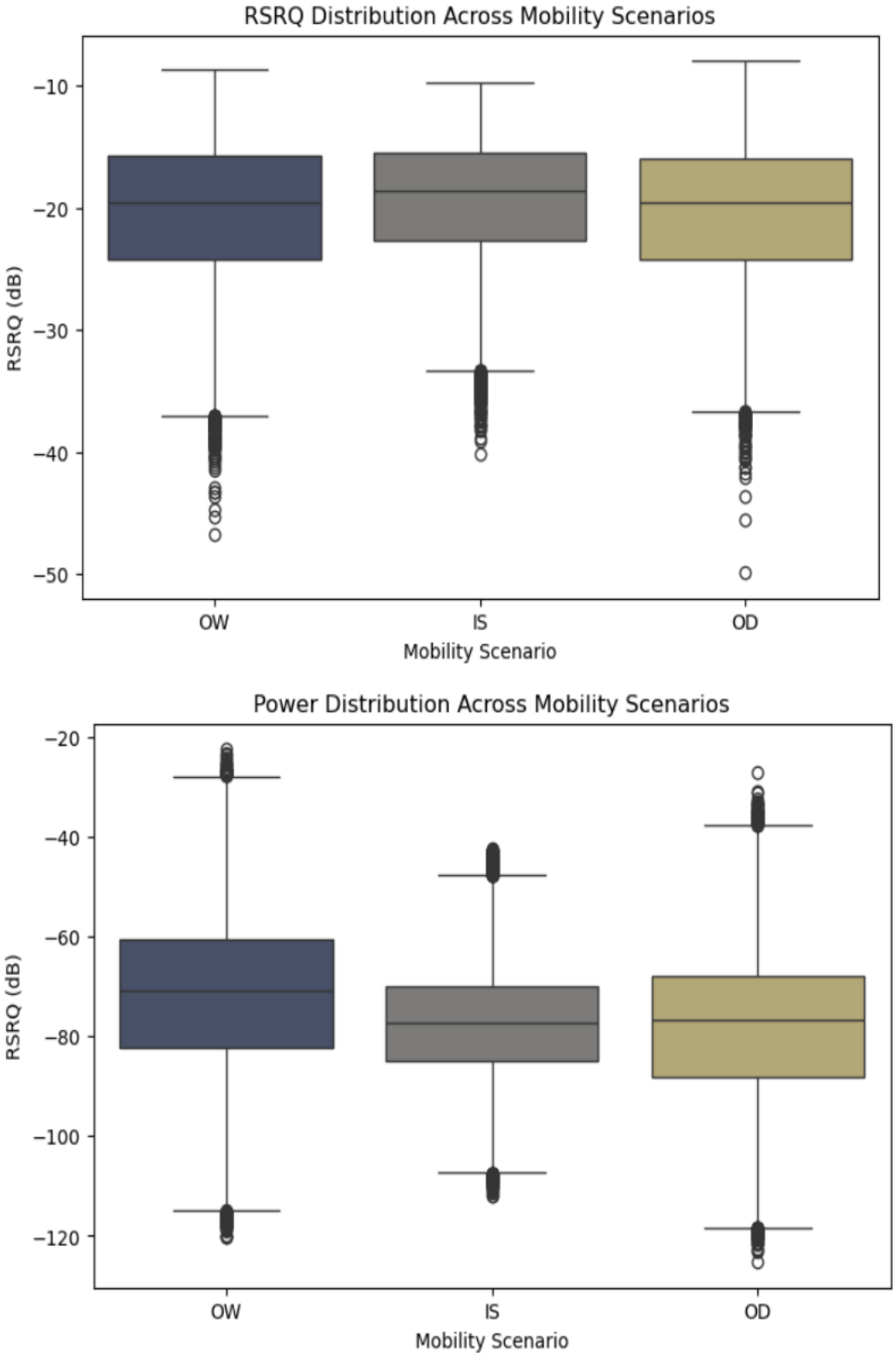


Figure 2.13: Radio Coverage Metrics Across Different Network Scenarios.

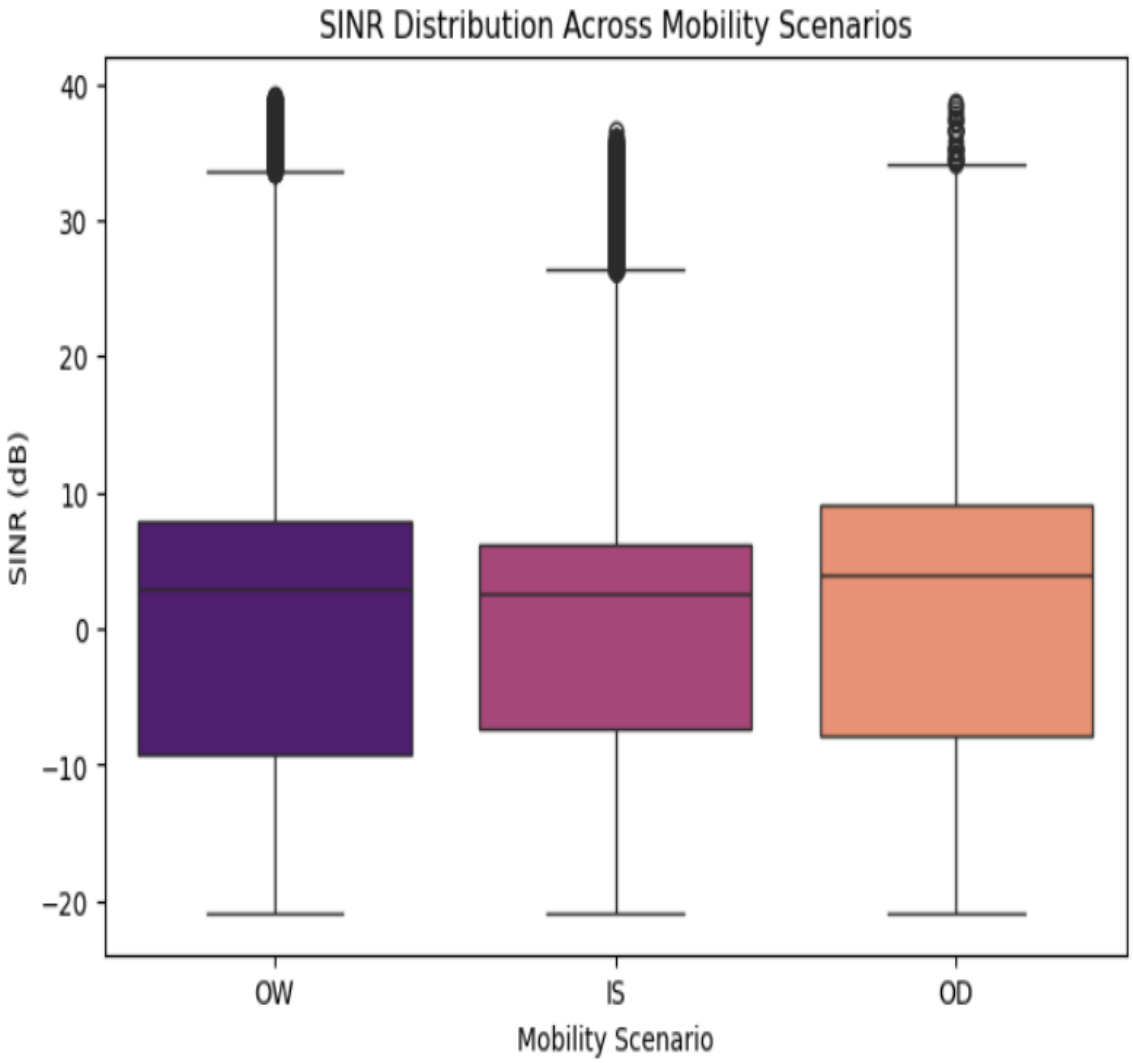


Figure 2.14: SINR Distribution Across Mobility Scenarios.

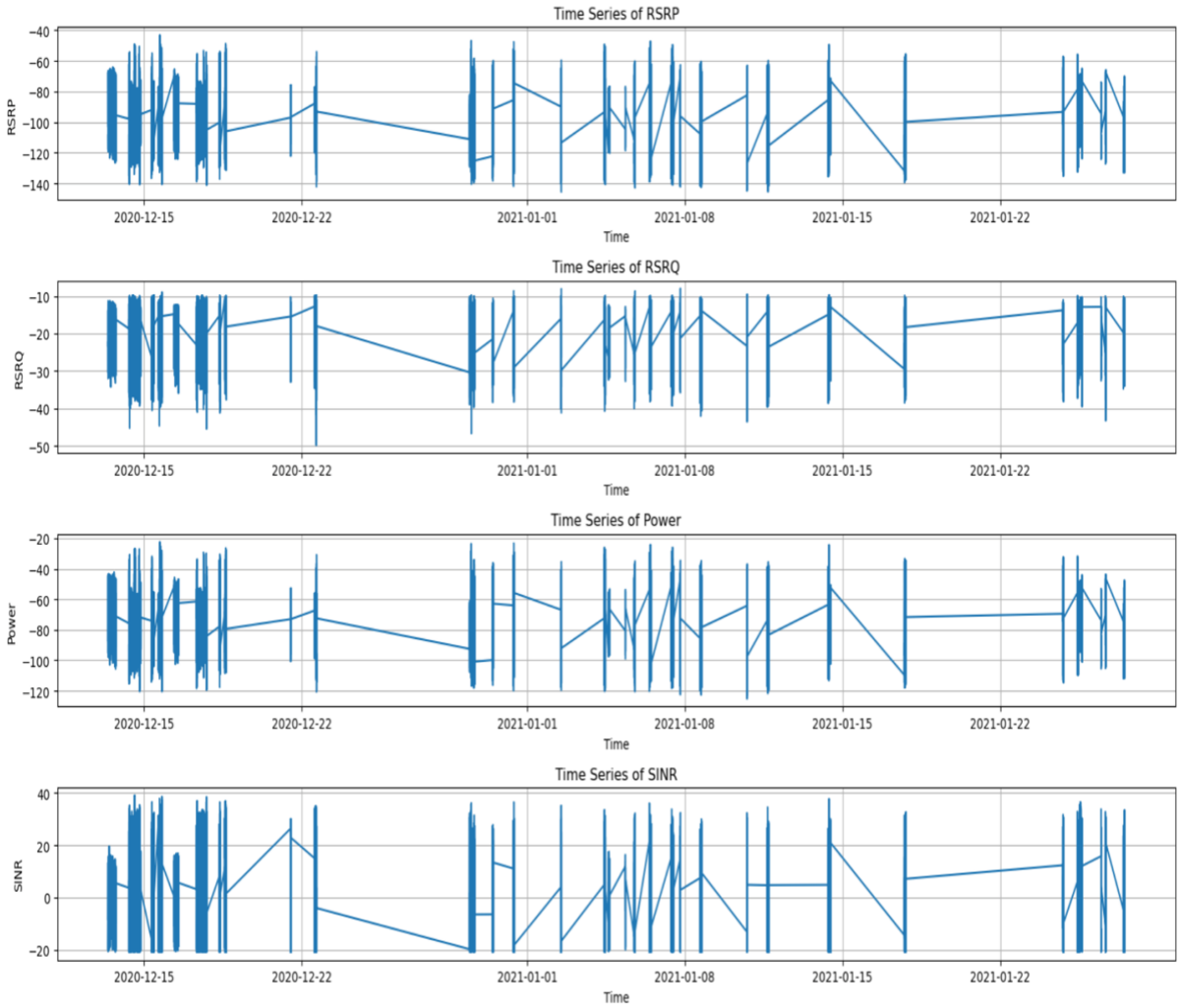


Figure 2.15: Time Series Plots for Signal Metrics.

2.6 Encoding and Feature Scaling

Encoding transforms categorical data into a structured format suitable for machine learning (Bolikulov *et al*., 2024). In this study, MNC, Scenario, and Campaign, which have 2, 3, and 193 unique values, respectively, were encoded to enhance model compatibility. MNC and Scenario were one-hot encoded due to their nominal nature, while Campaign, despite also being nominal, was label encoded to avoid high dimensionality and increased computational load from one-hot encoding (Bolikulov *et al*., 2024). This strategy efficiently transforms categorical features while maintaining computational efficiency. Additionally, date and time fields were standardised into recognised formats to support accurate temporal analysis (Binding and Tudhope, 2023). These preprocessing steps ensured the dataset remained clean, consistent, and ready for subsequent modelling tasks in LTE network performance analysis.

Machine Learning Techniques

Machine learning, a branch of artificial intelligence, enables systems to learn from data and make informed decisions without explicit programming (Kühl *et al*., 2019). It includes supervised, unsupervised, semi-supervised, and reinforcement learning techniques (Sarker, 2021). In this analysis, both supervised and unsupervised machine learning techniques were employed, with efficient management of computational resources being a key consideration. To manage processing demands, a random sample of 40,000 rows (8% of the cleaned dataset) was used. This approach balances performance and representativeness, ensuring reliable training and evaluation while reducing computational load (Mahmud *et al*., 2020). Table 3.1 outlines the environment used to implement the proposed machine learning models and optimisation procedures.

Table 3.1: Information on the experimental environment

No	**Name**	**Version / Specification**
1	Operating System	macOS
2	Processor	2.4 GHz Quad-Core Intel Core i5
3	Graphics	Intel Iris Plus Graphics 655 1536 MB
4	RAM	8 GB 2133 MHz LPDDR3
5	Python	3.12
6	Jupyter Notebook	7.0.8
7	Anaconda Navigator	2.6.5

In the following section, machine learning techniques, including clustering and classification, were applied to the sampled dataset. Clustering was used to uncover patterns without prior labels, aiding in the detection of anomalies. Classification models were used for predictive modelling and the extraction of actionable insights. These methods were chosen over other approaches, such as regression, as the objective was to segment data and categorise observations rather than predict continuous values.

3.0 Clustering Analysis

3.1 Introduction to Clustering Analysis

Clustering is an unsupervised learning technique used to group data based on inherent similarities without predefined labels (Rodriguez *et al*., 2019). In network analysis, it segments environments into regions with similar signal quality and operational conditions, aiding targeted optimisation (Bui *et al*., 2023). This study applied K-Means and DBSCAN due to their complementary strengths. K-Means is computationally efficient and suitable when clusters are compact and equally sized, offering a solid baseline when the number of clusters is known (Ikotun *et al*., 2022). DBSCAN, by contrast, identifies clusters of arbitrary shapes and is robust to noise, making it ideal for complex LTE datasets with irregularities (Amini, Wah and Saboohi, 2014). Using both algorithms enabled a comprehensive analysis of hidden patterns in the data, providing insights into user behaviour and signal distribution, as supported by recent studies (Mendes-Santos *et al*., 2021; Eckhardt *et al*., 2022). The workflow is shown in Figure 3.1.

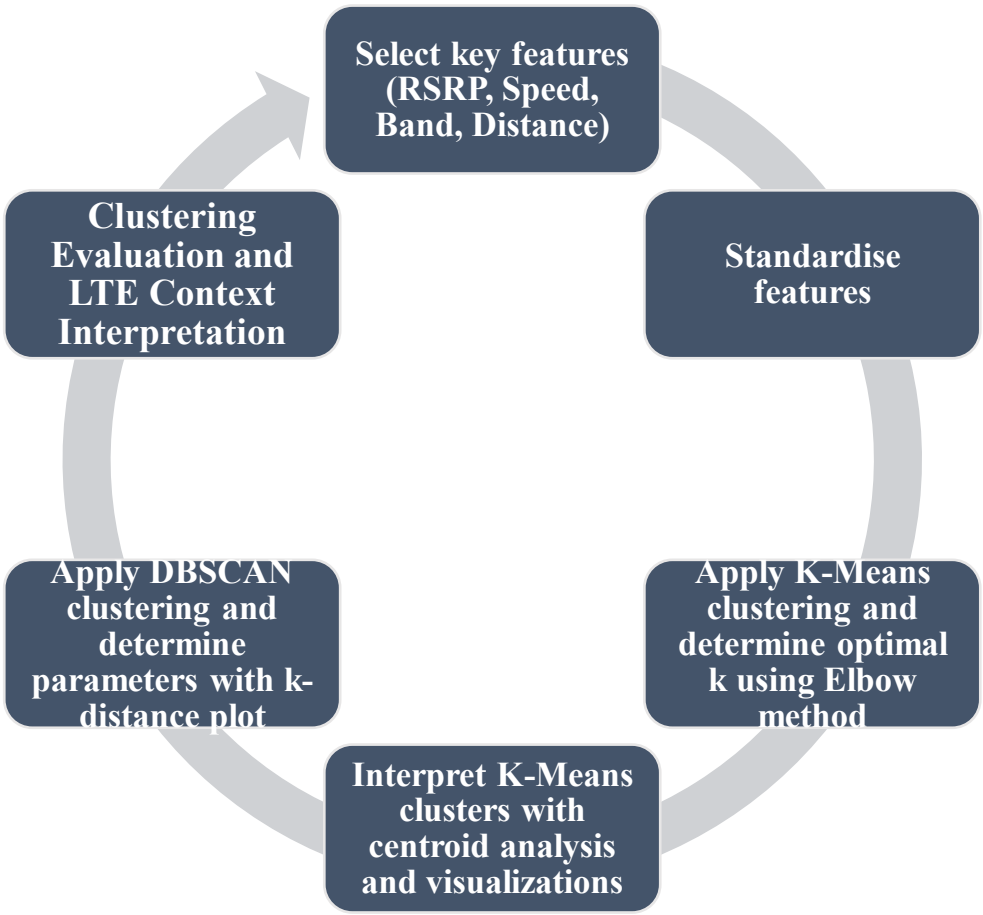


Figure 3.1: Clustering Workflow Using K-Means and DBSCAN

3.2 Feature Selection and Preparation for Clustering

The dataset contains many features; however, clustering requires selecting those that effectively reveal underlying network patterns (Eckhardt *et al*., 2022). Redundant or highly correlated variables, such as RSRP and Power ($r > 0.9$), can introduce multicollinearity and reduce computational efficiency (Pfaehler *et al*., 2021). Similarly, spatial features like Latitude and cellLatitude offer overlapping insights and may be replaced by a derived metric like distance. Including such variables increases dimensionality, distorts distance-based measures, and may hinder clustering performance. Removing redundant features helps reduce noise, improve computational efficiency, and enhance interpretability by ensuring that clusters reflect meaningful patterns within the data (Kuzudisli *et al*., 2023).

Based on this rationale, feature selection prioritised numerical attributes including Distance, RSRP, Band, and Speed, which effectively capture the spatial dynamics and performance characteristics relevant to LTE network conditions and mobility scenarios. Before clustering, Z-score normalisation was applied to standardise feature scales, ensuring equitable contribution during Euclidean distance-based analysis in algorithms like K-Means (Wongoutong, 2024). Standardisation adjusted the features to have a mean of zero and a standard deviation of one, thereby enhancing clustering accuracy by minimising scale-related bias (Wongoutong, 2024).

Table 3.2 shows the distribution of the scaled features, which are more symmetric compared to their original forms (Table 3.3). This preprocessing step is essential for reducing noise and distortion in the data, improving clustering coherence and reliability, and ensuring the algorithm interprets patterns based on relationships rather than raw magnitude differences.

Table 3.2: Summary Statistics (Scaled)

Statistic	Distance	RSRP	Band	Speed
Count	40,000	40,000	40,000	40,000
Mean	-6.89e-17	-1.05e-15	-1.71e-17	-2.13E-17
Std Dev	1.00	1.00	1.00	1.00
Min	-0.70	-3.08	-0.94	-0.55
25th Percentile	-0.50	-0.67	-0.66	-0.48
50th percentile (Median)	-0.34	0.01	-0.66	-0.28
75th Pctl	0.08	0.67	-0.10	-0.05
Max	10.28	3.40	1.74	7.53

Table 3.3: Summary Statistics (Unscaled)

Statistic	Distance	RSRP	Band	Speed
Count	40,000	40,000	40,000	40,000
Mean	552.55	- 98.56	7.69	5.18
Std Dev	784.24	14.15	7.09	9.43
Min	0.22	- 142.18	1.00	0.00
25th Percentile	158.58	- 108.05	3.00	0.68
50th percentile (Median)	286.57	- 98.50	3.00	2.59
75th Pctl	618.05	- 89.15	7.00	4.68
Max	8617.74	- 50.43	20.00	76.21

3.3 K-Means Clustering Analysis

K-Means is a widely used partitioning algorithm that divides the dataset into a predefined number of clusters, denoted by *k*. The algorithm iteratively assigns each data point to the nearest cluster centre (centroid) and updates the centroids based on the mean of the points assigned. The primary objective is to minimise the sum of squared errors (SSE) within each cluster. The simplicity, scalability, and interpretability of K-Means make it an appropriate choice for large datasets such as the 4G LTE dataset being analysed (Zubair *et al*., 2022).

3.3.1 Determining the Optimal Number of Clusters

Selecting an appropriate number of clusters (k) is a crucial step when applying the K-Means clustering algorithm. In this analysis, the Elbow Method was employed to determine the optimal value of k, as it is a widely accepted and effective heuristic for estimating the number of clusters in unsupervised learning tasks (Yuan and Yang, 2019). This method involves plotting the variation (often measured by the sum of squared errors or inertia) against a range of k values. The optimal number of clusters is indicated at the point where adding another cluster yields only a marginal reduction in the sum of squared errors, forming a characteristic "elbow" shape in the plot. To identify this elbow point, K-Means was applied iteratively with k ranging from 2 to 9. The resulting inertia values were plotted against k, as shown in Figure 3.2.

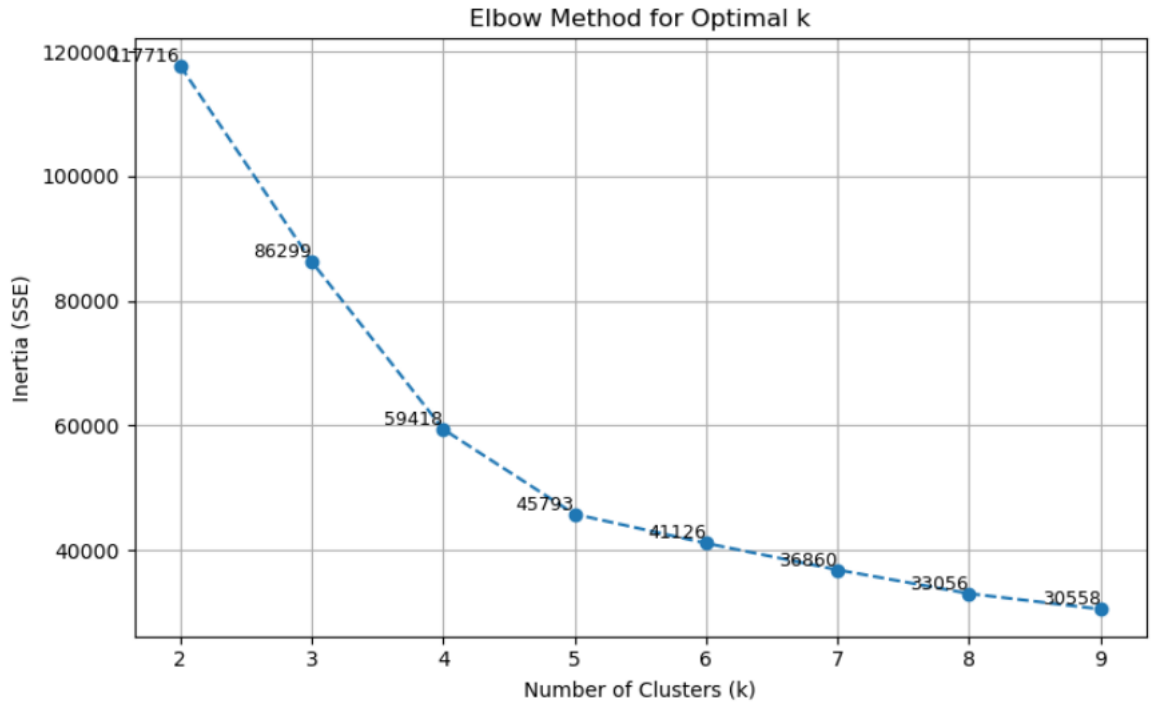


Figure 3.2: Elbow Method for Optimal K.

In this analysis, the Elbow Method was applied to determine the optimal number of clusters by evaluating the sum of squared errors (SSE) for k values ranging from 2 to 9. A distinct elbow was observed at k = 4, suggesting that beyond this point, improvements in cluster compactness diminish significantly. While the elbow in Figure 3.1 initially suggested k = 4 and k = 5 as potential candidates, a detailed analysis of the percentage drop in the Sum of Squared Errors (SSE) between consecutive k values provided further clarity. Notable reductions were observed between k = 2 to 3 (26.70%) and k = 3 to 4 (31.35%), followed by diminishing returns: 22.91% (k = 4 to 5), 10.20%, 10.36%, 10.31%, and 7.55% for subsequent values. These results indicate that the most significant improvement in cluster compactness occurred up to k = 4, after which the marginal benefit of adding additional clusters declined. Therefore, k = 4 was selected as the most appropriate number of clusters, balancing model simplicity with performance.

3.3.2 Implementation and Results

Following the selection of $k = 4$, the K-Means algorithm was executed on the standardised dataset. The algorithm converged after a finite number of iterations, and each record was assigned to one of the four clusters. The resulting centroids, computed in the standardised feature space, were then stored in a data frame for better readability, as shown in Table 3.4. Principal Component Analysis (PCA) was not applied in this process, as the clustering was performed on a small set of four features. Given the low dimensionality and the need to maintain the interpretability of the cluster characteristics, dimensionality reduction was deemed unnecessary (Jia *et al*., 2022).

Table 3.4: Cluster Centers in Scaled Format

Cluster	Distance	RSRP	Band	Speed
0	- 0.26	- 0.07	- 0.52	- 0.24
1	- 0.14	0.68	1.74	- 0.19
2	0.11	- 0.37	- 0.02	3.58
3	3.13	- 1.35	- 0.61	- 0.09

The clustering centre analysis of the LTE network data reveals distinct user groups with specific characteristics:

- **Cluster 0**: Users are close to the serving cell, have low mobility, and use lower frequency bands. They experience slightly below-average signal strength, likely representing indoor-static or outdoor-walking users in stable signal areas.
- **Cluster 1**: Users are slightly closer than average to the serving cell, with low mobility and strong signal strength. They use higher frequency bands, suggesting static users in high-density areas.
- **Cluster 2**: Users exhibit high mobility, are at a moderate distance from the serving cell, and have below-average signal strength. This group likely represents outdoor driving users experiencing fluctuating signal conditions.
- **Cluster 3**: Users are far from the serving cell, have low mobility, and use lower frequency bands. They experience the poorest signal strength, possibly due to being at the edge of coverage or in obstructed areas.

The clustering analysis provided valuable insights into user behaviour and signal performance within the LTE network, highlighting the need for targeted optimisation strategies. Each cluster revealed a distinct user profile based on key factors such as signal strength (RSRP), user mobility, frequency band usage, and proximity to the serving cell. This approach enabled a detailed understanding of network usage patterns and performance differences across user segments. These findings support the implementation of a segment-aware optimisation strategy, where network parameters such as radio resource management, cell planning, and spectrum allocation are aligned with user-specific characteristics.

These tailored strategies can significantly enhance the quality of service (Qos), improve user experience, and increase overall network efficiency. Additionally, a series of scatter plots mapping RSRP against speed, frequency band, and distance to the serving cell (Figure 3.3) reveals four defined overlapping clusters. Each cluster is represented by a different colour, with centroids visually marked. The centroids demonstrate a negative linear relationship between RSRP and distance, confirming that signal strength tends to be higher at shorter distances from the serving cell. Additionally, a negative linear relationship is observed between the centroids of RSRP and speed, suggesting that higher user mobility is associated with lower signal strength. In contrast, a positive relationship is evident between RSRP and band centroids, indicating that higher signal strength is more commonly associated with the use of higher frequency bands.

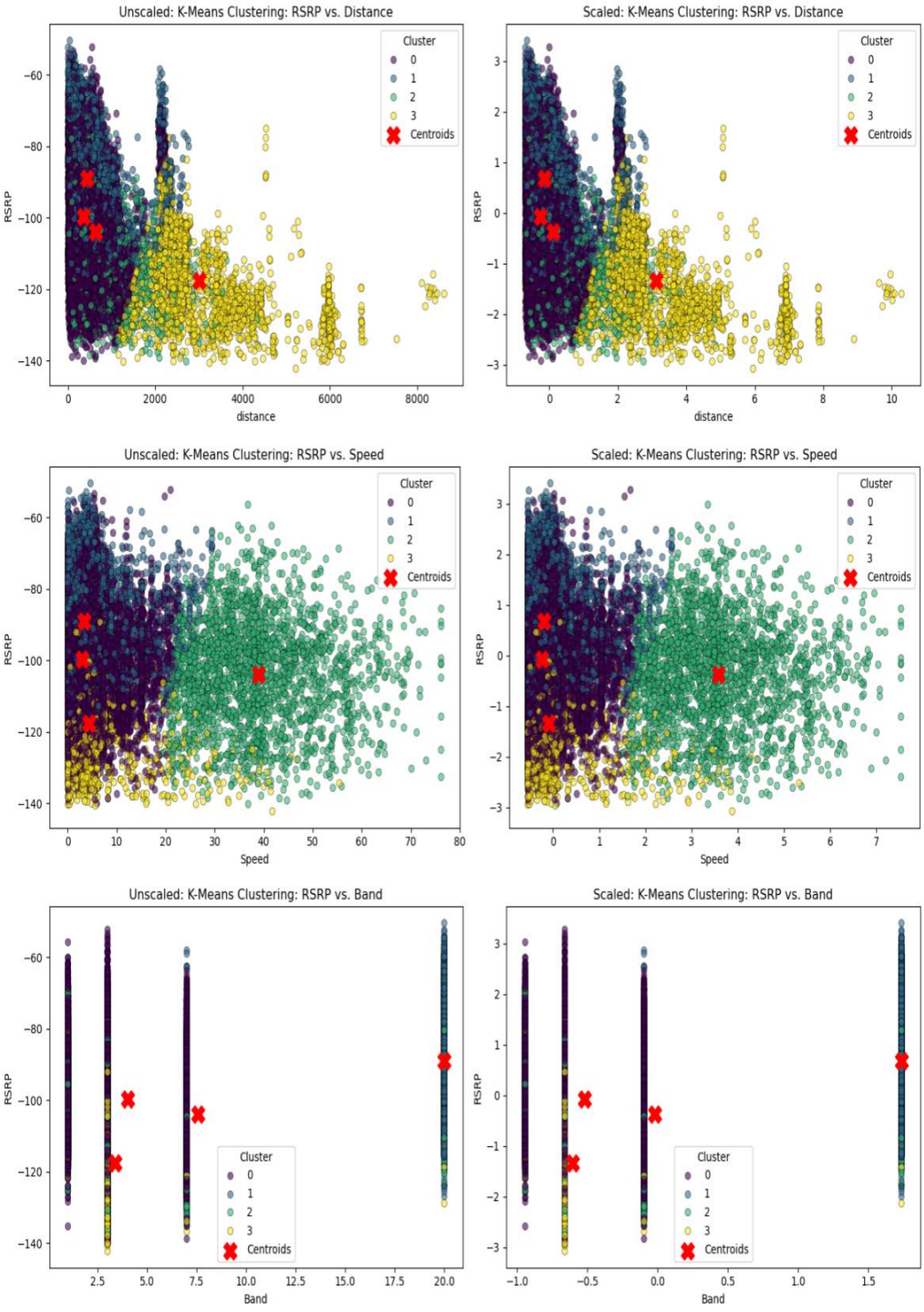


Figure 3.3: K-Means Clustering of RSRP against other Variables.

3.4 DBSCAN Clustering Analysis

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that does not require predefining the number of clusters. It uses epsilon (ε) to define the neighbourhood radius and a minimum number of samples to identify dense regions. DBSCAN effectively detects clusters of arbitrary shape and classifies outliers as noise (Deng, 2020).

3.4.1 Parameter Selection Using k-Distance Graph

To determine the optimal value for ε in DBSCAN, the k-distance graph was utilised. This method involves computing the distance to each point's min_samples-th nearest neighbour and sorting these distances in ascending order. The optimal ε corresponds to the "knee point," which is the location on the curve where a sharp change in slope is observed. As illustrated in Figure 3.4, this knee point was identified at approximately 0.797, which was selected as the optimal ε. According to Schubert *et al*. (2017), the min_samples parameter was set to 8, calculated as twice the number of features being clustered.

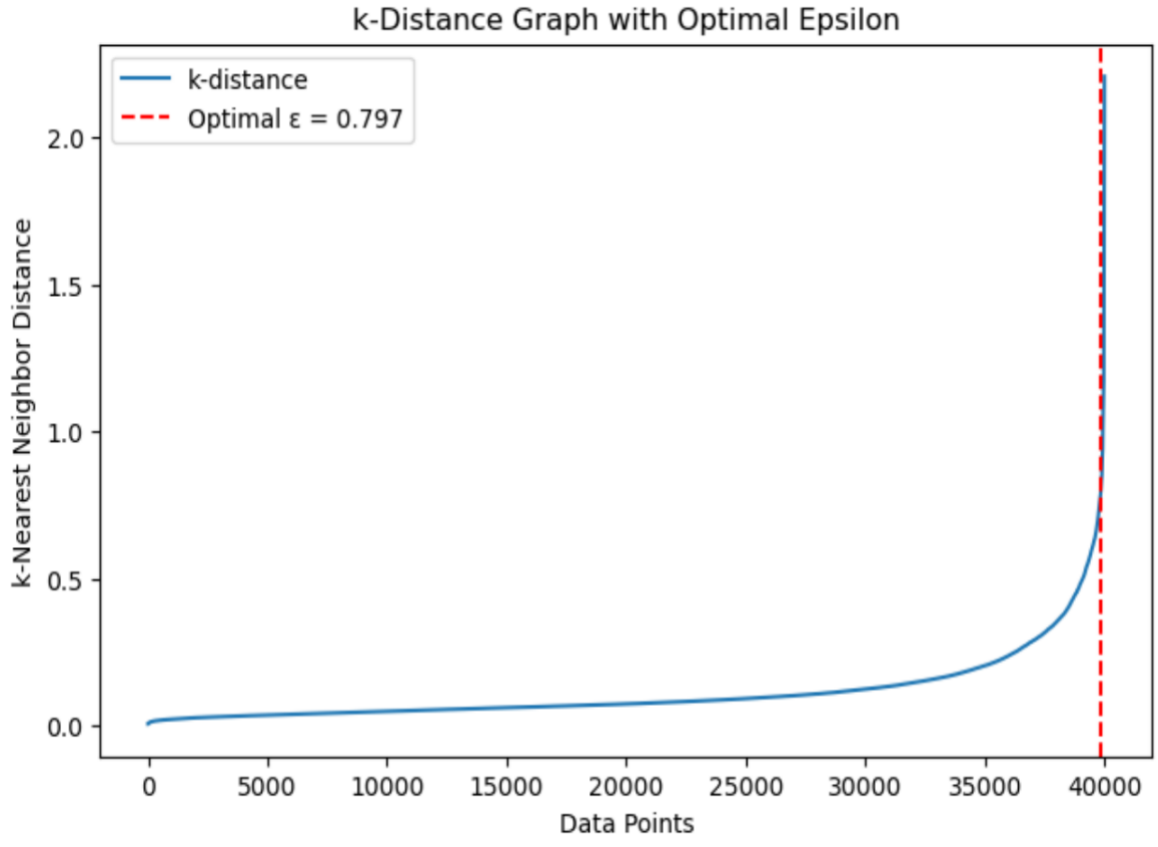


Figure 3.4: k-Distance Graph with Knee Point Indicating Optimal ε for DBSCAN.

3.4.2 Implementation and Results

DBSCAN was applied to the standardised dataset using ε = 0.797 and min_samples = 8, and this algorithm identified three main clusters and a small proportion as noise. Cluster 0 contained the majority (76.5%) of data points, suggesting regions of strong signal strength and low interference. Cluster 1 included 23.3%, while Cluster 2 comprised a minority (0.05%) of the data. Approximately 0.15% of points were labelled as noise (-1). Figure 3.5 presents a pie chart illustrating this distribution. DBSCAN's capacity to detect and isolate outliers is effective in mitigating the influence of anomalous data on subsequent analyses (Deng, 2020).

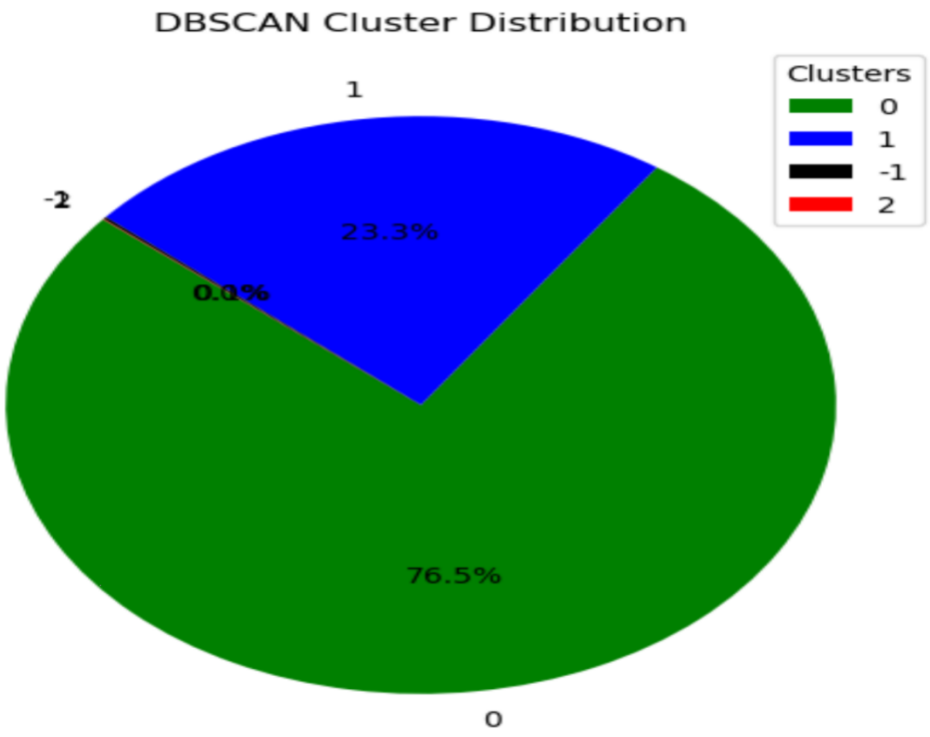


Figure 3.5: Pie Chart Representing DBSCAN Cluster Distribution and Noise Points.

Scatterplots comparing clusters based on RSRP and other variables (Figure 3.6) demonstrate that DBSCAN, like K-Means, segments the data into meaningful groupings. However, DBSCAN's density-based approach also identifies sparse regions as noise, whereas K-Means might assign them to clusters. This distinction uncovers the data's natural structure, where low-density regions may represent network variability or transient conditions. The plot highlights noise points sparsely positioned above the overlapping area of clusters 0 and 1, while cluster 2 is isolated at a point of high distance and low RSRP. Overall, substantial overlap among clusters suggests interconnected relationships between RSRP, speed, and band.

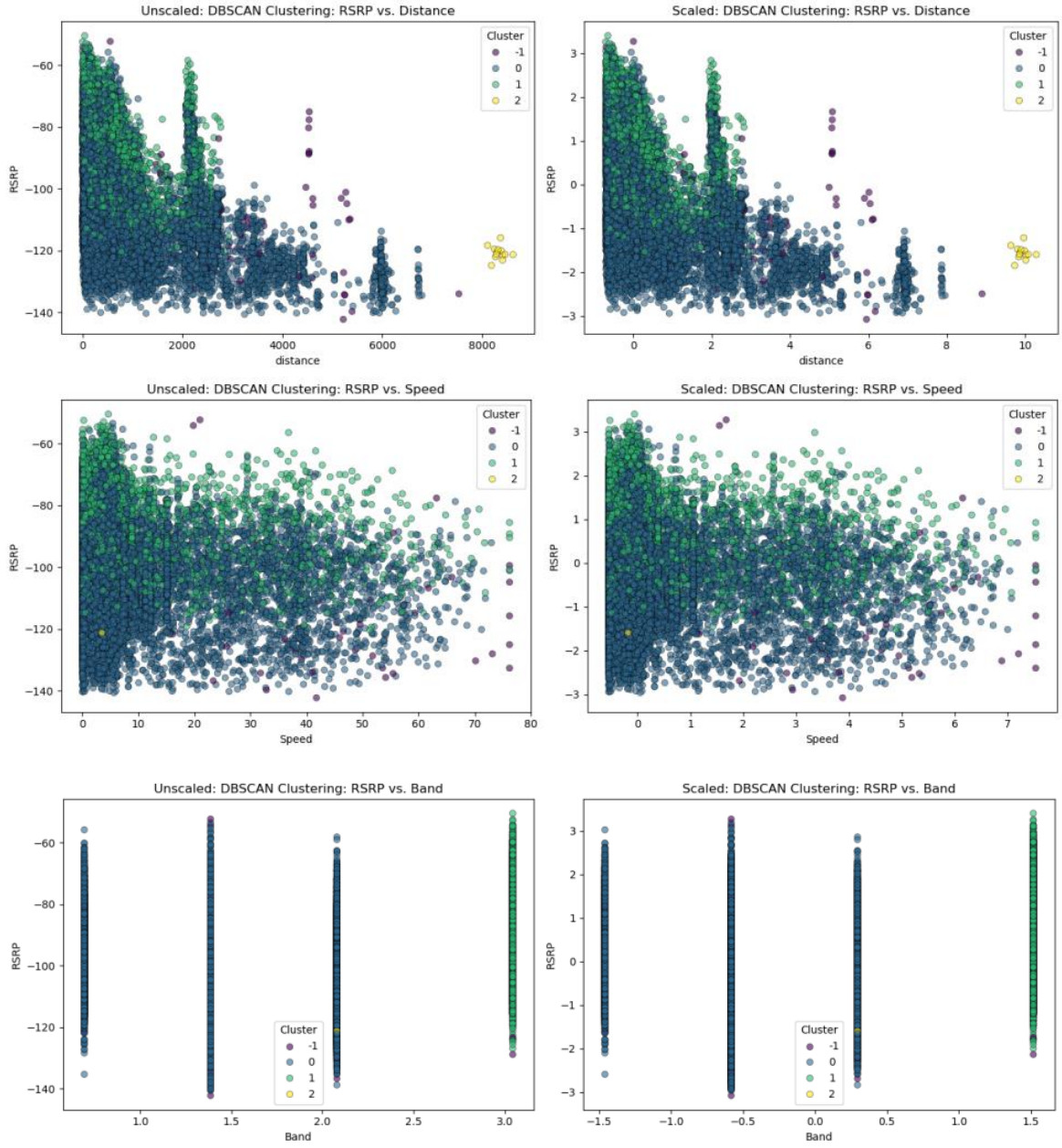


Figure 3.6: DBSCAN Clustering of RSRP against other Variables.

3.5 Evaluation of Clustering Performance

A silhouette analysis was performed to assess the quality of the clustering outcomes. The silhouette coefficient, ranging from -1 to 1, measures how well each data point fits within its assigned cluster compared to others, with higher values indicating better cohesion and separation (Shahapure and Nicholas, 2020). K-Means clustering achieved a silhouette score of 0.49, while DBSCAN achieved approximately 0.40. These results indicate that both algorithms produced meaningful groupings; however, K-Means provided moderately better-defined clusters, with an estimated 22.5% improvement in clustering quality.

A comparative evaluation of the two clustering algorithms reveals notable distinctions. K-Means, configured with four predefined clusters, produced relatively balanced and interpretable groupings, as illustrated in the visual scatterplots. However, DBSCAN, which follows a density-based approach, identified three main clusters along with a small proportion of noise points. While DBSCAN excels at detecting outliers and accommodating arbitrarily shaped clusters, its lower silhouette score suggests reduced overall separation and cohesion between clusters. The selection of an appropriate clustering method ultimately depends on the objectives of the analysis. In applications where anomaly detection and handling of non-spherical clusters are priorities, DBSCAN may be more suitable. Conversely, when the objective is to achieve balanced, clearly defined clusters that support straightforward interpretation, K-Means presents a more favourable option. In practice, adopting a hybrid or comparative clustering strategy can enhance analytical outcomes, particularly when dealing with complex or high-dimensional datasets (Rathore *et al*., 2019).

4.0 Classification

This section outlines a data-driven approach for LTE network optimisation using predictive classification models, as shown in Figure 4.1. The objective is to assess network performance states from a preprocessed LTE dataset. Classification, a supervised learning approach, leverages labelled historical data to predict categorical outcomes for new instances (Ramesh, Karuppasamy, & Veerappapillai, 2020). Both binary and multi-class classification were implemented to distinguish between favourable (e.g., good signal strength) and unfavourable (e.g., no signal strength) conditions and to categorise performance into multiple scenarios. This enhances anomaly detection, facilitates proactive network management, and guides optimisation strategies. The methodology includes target variable selection, model training, and performance evaluation. Additionally, model interpretability tools, such as feature importance analysis, were applied to identify key predictors influencing network performance and support informed decision-making.

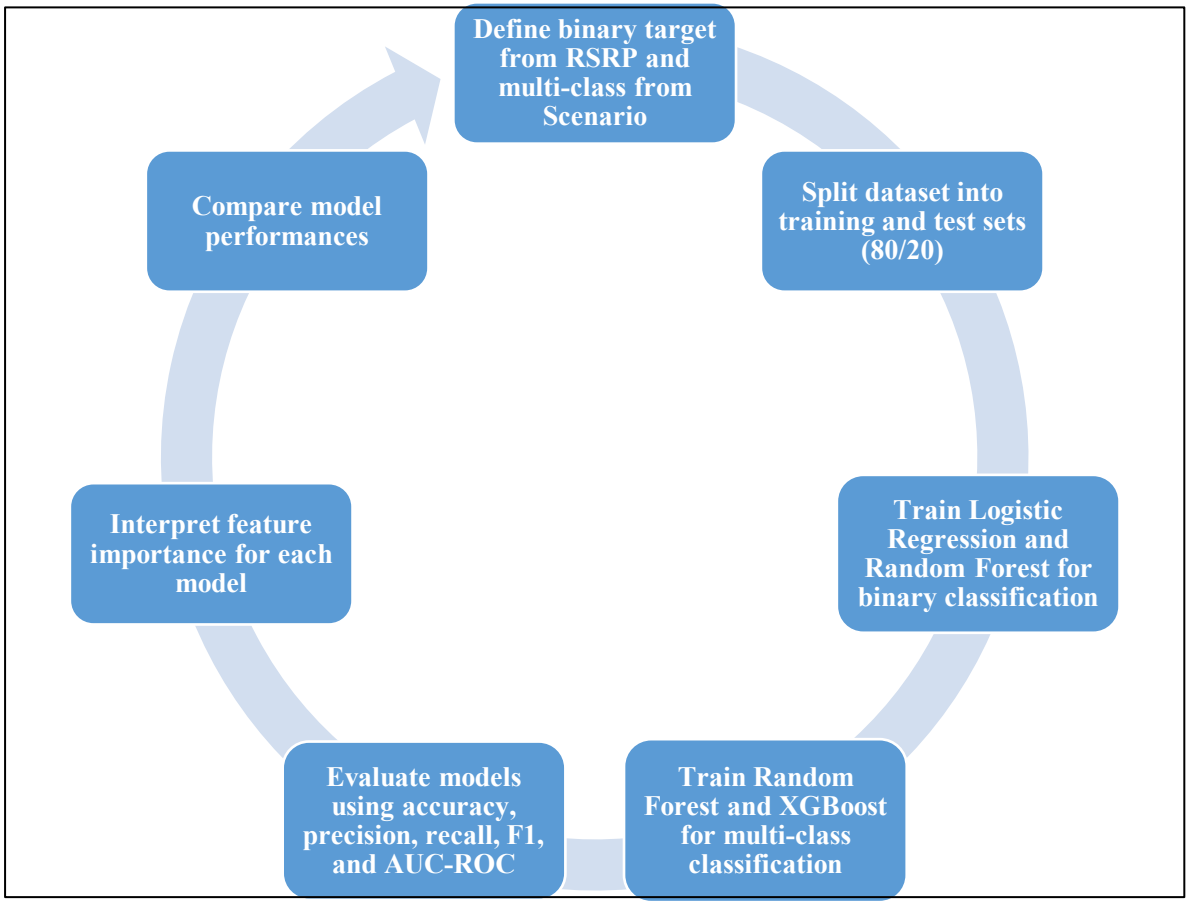


Figure 4.1: Supervised Learning Workflow for Binary and Multi-Class LTE Classification

4.1 Target Variable Selection

The analysis starts with a problem definition and the selection of an appropriate target variable. In this analysis, the selected target variable is the Reference Signal Received Power (RSRP), a fundamental LTE metric indicative of signal strength and overall network performance (Simpson and Sun, 2018). For the binary classification task, RSRP values were converted into categorical labels using a threshold informed by industry standards. Specifically, values above –100 dBm were classified as "Signal" (label 1), while those below were labelled as "No signal" (label 0) (Teltonika, 2024; Aggarwal, Teja and Mittal, 2024; Akpaneno, Akinbolati and Ekundayo, 2024). This transformation enabled effective binary classification. For the multi-class classification task, the Scenario variable was selected, comprising three classes: outdoor driving, outdoor walking, and indoor static. This categorisation facilitates a better understanding of performance variation under different mobility conditions, offering insights applicable to real-world optimisation strategies and targeted interventions in LTE network management.

4.2 Data Preprocessing and Data Splitting

A preprocessed sample dataset was used for the analysis, and a correlation analysis similar to that shown in Figure 2.6 was performed. Irrelevant features and those with absolute correlation values less than approximately 0.10 with the target variable (RSRP) were excluded from the analysis due to their weak association. However, speed was retained despite its low correlation, as illustrated in Figure 4.2, since Pedapolu *et al*. (2016) highlighted its influence on signal strength.

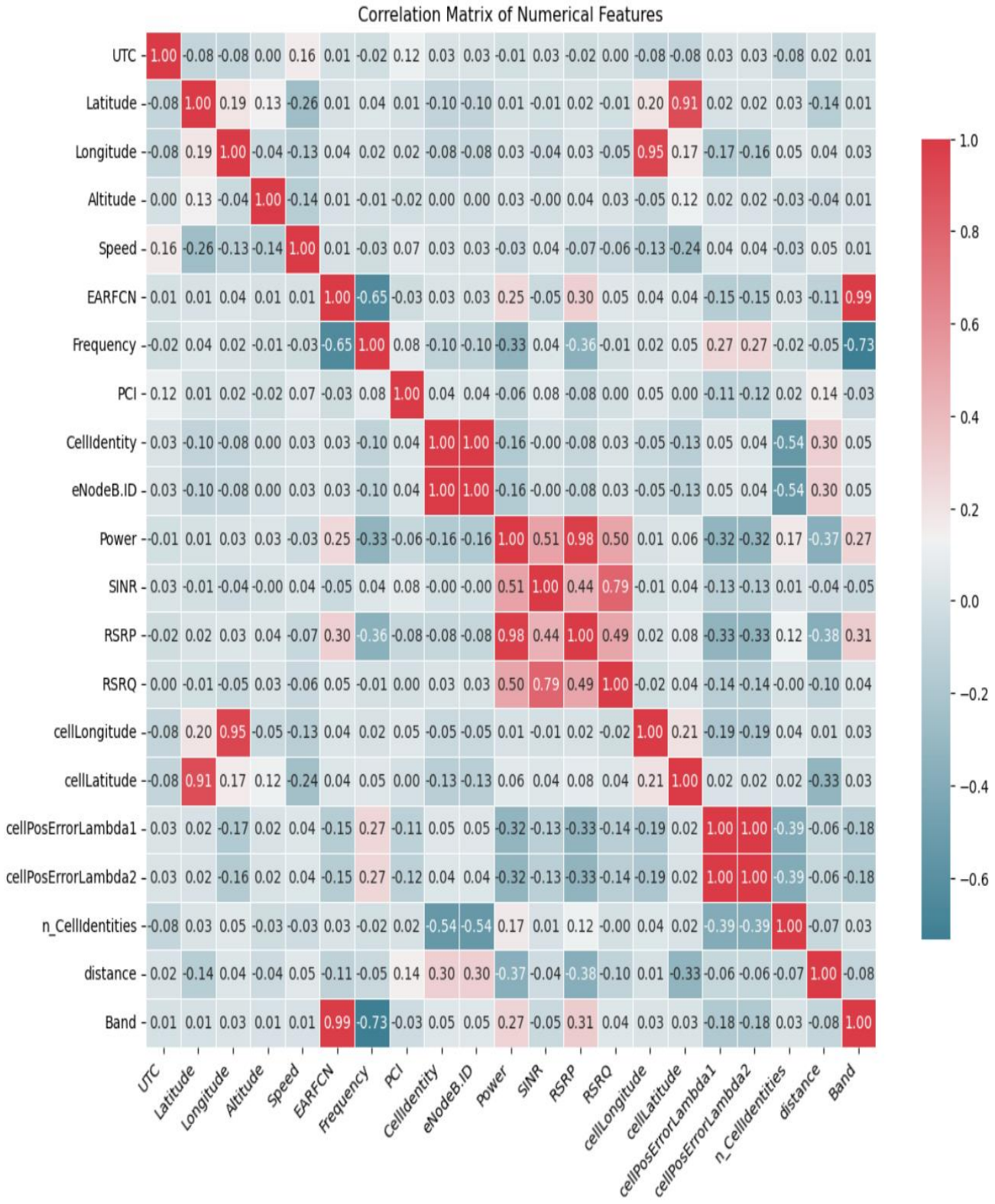


Figure 4.2: Correlation Matrix Heatmap Displaying Pearson Correlation Coefficients among Variables.

Also, to evaluate if categorical features, including MNC and Scenario, have a statistically significant impact on RSRP, an Analysis of Variance (ANOVA) test was performed, analysing the F-statistic and p-value (Tenny and Abdelgawad, 2023). The F-statistic measures the ratio of variance between group means and the variance within groups; higher values indicate a stronger effect of the grouping variable. The corresponding p-value represents the probability of observing such an effect by chance, assuming the null hypothesis is true. When the p-value falls below a predefined threshold (commonly 0.05), corresponding to a 95% confidence level, the result is considered statistically significant. This implies that the null hypothesis, which states there is no relationship between the feature and RSRP, can be rejected with acceptable confidence (Tenny and Abdelgawad, 2023). Table 4.1 presents the ANOVA results, revealing that both MNC and Scenario exhibit statistically significant associations with RSRP, as their p-values fall below the 0.05 threshold. Moreover, the Scenario variable produced a substantially higher F-statistic than MNC, indicating a stronger contribution to the variation in RSRP. This finding supports the relevance of Scenario as a key factor in network performance analysis.

Table 4.1: ANOVA Results

Factor	F-Statistic	p-Value	Significance
MNC vs. RSRP	221.90	4.76e-50	Significant
Scenario vs. RSRP	891.97	0.00	Significant

Features that exhibited low correlation and lacked statistical significance with the target variable were removed to improve model efficiency and reduce computational complexity. The resulting preprocessed dataset comprises 15 key variables and 40,000 observations, including encoded categorical features essential for classification tasks. This structured preprocessing approach ensures that only the most relevant attributes contribute to the predictive modelling process. Table 4.2 presents a detailed summary of the final dataset composition.

Table 4.2: Summary of Key Variables

Column	Non-Null Count	Data Type
Speed	40,000	float64
EARFCN	40,000	float64
Frequency	40,000	float64
Power	40,000	float64
SINR	40,000	float64
RSRP	40,000	float64
RSRQ	40,000	float64
cellPosErrorLambda1	40,000	float64
cellPosErrorLambda2	40,000	float64
n_CellIdentities	40,000	int64
Distance	40,000	float64
Band	40,000	float64
MNC_"Op"[2]	40,000	int64
Scenario_OD	40,000	int64
Scenario_OW	40,000	int64

The target variable (**y**) in the dataset was defined by categorising the Reference Signal Received Power (RSRP) values into two distinct classes: **1** for signal (RSRP > -100 dBm) and **0** for poor or no signal (RSRP ≤ -100 dBm) (Aggarwal, Teja and Mittal, 2024), and a new column, RSRP_Class**,** was created. The feature variables (**X**) were then defined by excluding both the original RSRP and RSRP_Class columns, resulting in a dataset with 14 predictor features and 40,000 observations. A class distribution analysis was conducted to check for biases and class imbalance in the target variable. The results were visualised using a pie chart, as shown in Figure 4.3 below, ensuring that both signal presence and absence were appropriately represented without bias in the RSRP_Class.

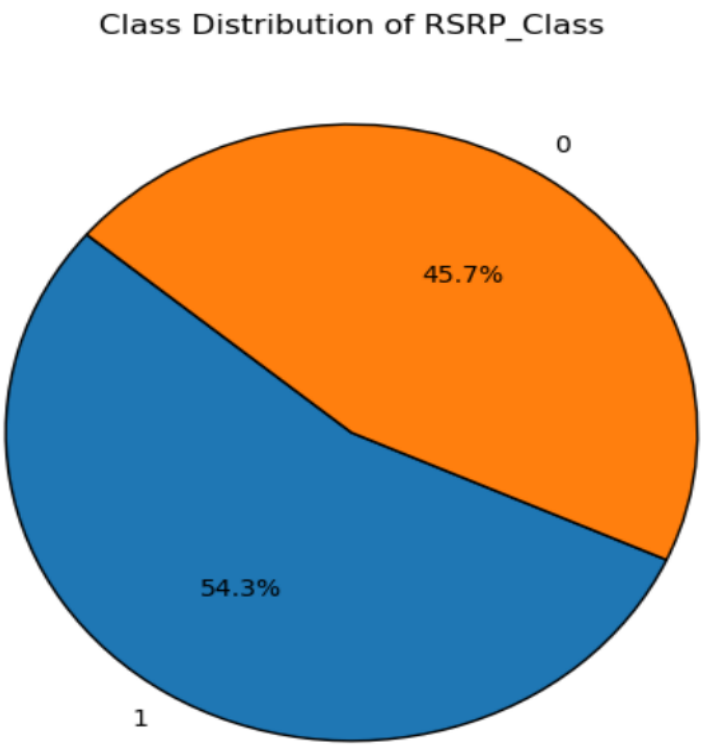


Figure 4.3: Class Distribution of Target Variable.

The final dataset was verified by checking the shapes of X (40,000, 14) and y (40,000). The first six rows for the RSRP and RSRP_Class were displayed (Table 4.3), confirming the correct transformation and suitability for further model training and evaluation.

Table 4.3: First six rows of RSRP and RSRP_CLASS columns

RSRP	RSRP_CLASS
-96.38	1
-114.08	0
-103.54	0
-80.70	1
-100.73	0

The dataset was divided into training and testing subsets using an 80-20 split ratio (Joseph, 2022), with 80% (32,000 rows) of the data allocated for model training and the remaining 20% (8,000 rows) set aside for performance evaluation. The training set facilitated the development of predictive models by allowing algorithms to learn the underlying patterns and relationships within the data. The testing set also provided an independent assessment of the model's generalisation capability, ensuring that performance metrics accurately reflect the model's ability to handle unseen data reliably.

4.3 Binary Classification Analysis

4.3.1 Model Selection and Training

For binary classification, two models are selected to capture different algorithmic perspectives: Logistic Regression and Random Forest Classifier. Logistic Regression, as a linear statistical model, serves as a foundational approach in classification tasks by estimating the probability that an observation belongs to a particular class through the logistic (sigmoid) function. It is effective when the outcome variable is binary in nature and is valued for its interpretability and computational efficiency, making it well-suited for this classification task (Das, 2023). The Random Forest classifier is a robust ensemble learning technique that builds multiple decision trees during training and combines their outputs through majority voting to make final predictions. This architecture enables it to effectively capture non-linear relationships and complex feature interactions while reducing the risk of overfitting through the use of bootstrapped samples and random feature selection (Fawagreh, Gaber and Elyan, 2014). During model development, each algorithm was fitted to the training data to learn the underlying patterns and evaluate their predictive performance.

4.3.2 Evaluation Metrics and Results

Model performance was assessed using a set of well-established evaluation metrics: accuracy, precision, recall (also known as sensitivity), F1-score, and the area under the Receiver Operating Characteristic curve (AUC-ROC). These metrics provide a comprehensive evaluation of the model's performance (Dritsas and Trigka, 2022) and are calculated using the core components of classification outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

1. **Accuracy**: Reflects the proportion of all correctly classified instances in the dataset.

$$Accuracy = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{FN} + \mathrm{TP} + \mathrm{FP}}$$

2. **Precision**: Measures how many of the predicted positive cases were actually positive.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

3. **Recall (Sensitivity)**: Captures the model's ability to identify actual positive cases

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. **F1-Score**: Provides a balance between precision and recall by calculating their harmonic mean

$$F1 - Score = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5. **AUC-ROC**: Evaluates the model's ability to distinguish between classes. A score closer to 1 indicates excellent discrimination performance.

As summarised in Tables 4.4 and 4.5 and illustrated in Figure 4.4, the Random Forest model achieved an overall accuracy of approximately 97%, with precision, recall, and F1-scores consistently around 0.97 for both classes. It significantly outperformed the Logistic Regression model, which attained an overall accuracy of around 95% with slightly lower performance metrics.

Table 4.4: Logistic Regression Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.96	0.94	0.95	3653
1	0.95	0.96	0.96	4347
Accuracy	–	–	0.95	8000
Macro Average	0.95	0.95	0.95	8000
Weighted Avg	0.95	0.95	0.96	8000

Table 4.5: Random Forest Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.97	0.97	0.97	3653
1	0.97	0.97	0.97	4347
Accuracy	–	–	0.97	8000
Macro Average	0.97	0.97	0.97	8000
Weighted Avg	0.97	0.97	0.97	8000

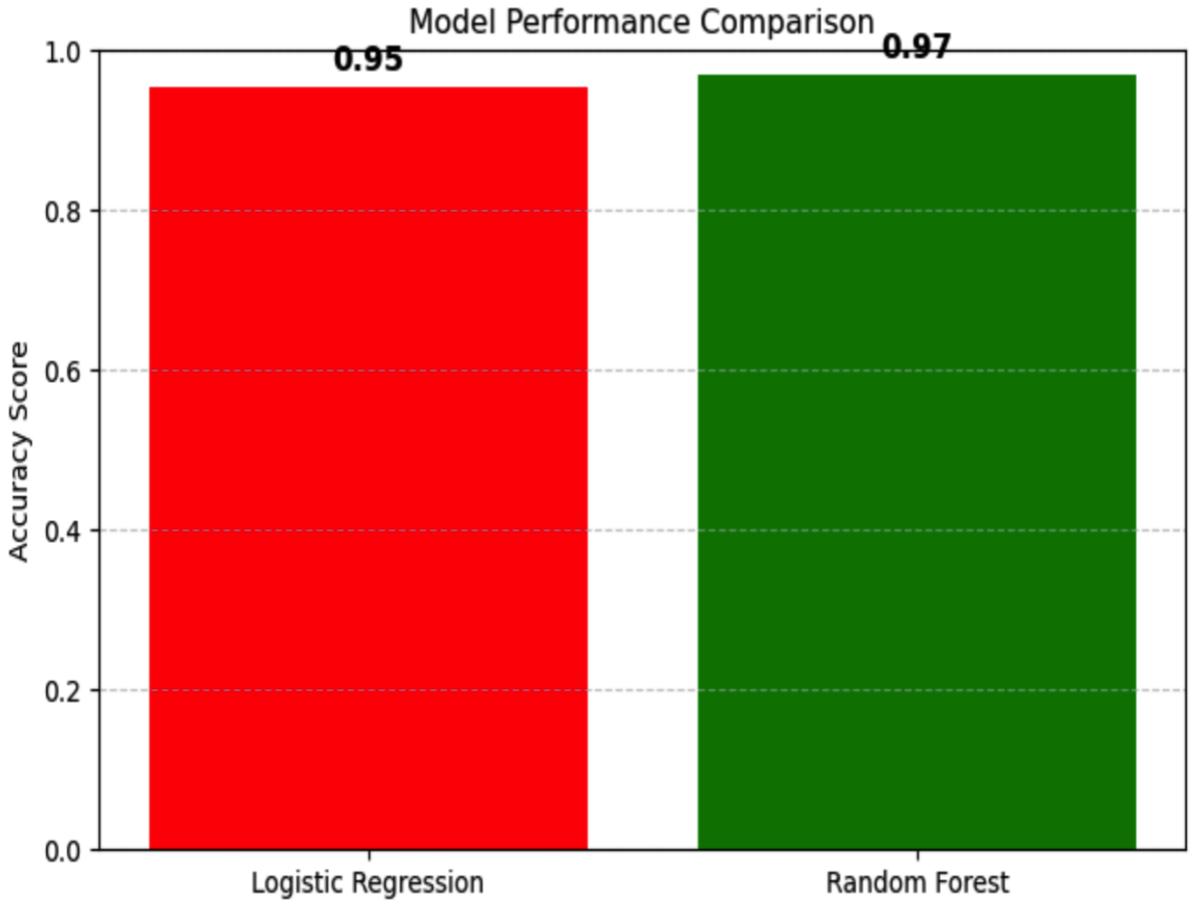


Figure 4.4: Model Performance Comparison for Binary Classification.

This performance was further validated by the confusion matrices, which are useful tools for understanding model performance in terms of TP, TN, FP, and FN (Sathyanarayanan, 2024). For the Logistic Regression model, the confusion matrix presented in Figure 4.5 below can be interpreted as follows: 3452 observations were correctly classified as negatives (TN), and 4185 observations were correctly classified as positives (TP). However, 201 negatives were misclassified as positives (FP), and 162 positives were misclassified as negatives (FN). Similarly, the confusion matrix for the Random Forest model indicates better performance, with 3531 TN and 4232 TP, while the number of misclassifications is lower, with only 122 FP and 115 FN. These results, combined with the ROC curves revealing AUC values of 0.9938 for Logistic Regression and 0.9967 for Random Forest (Figure 4.6), highlight the high discriminatory capabilities of both models, with Random Forest demonstrating slightly better performance.

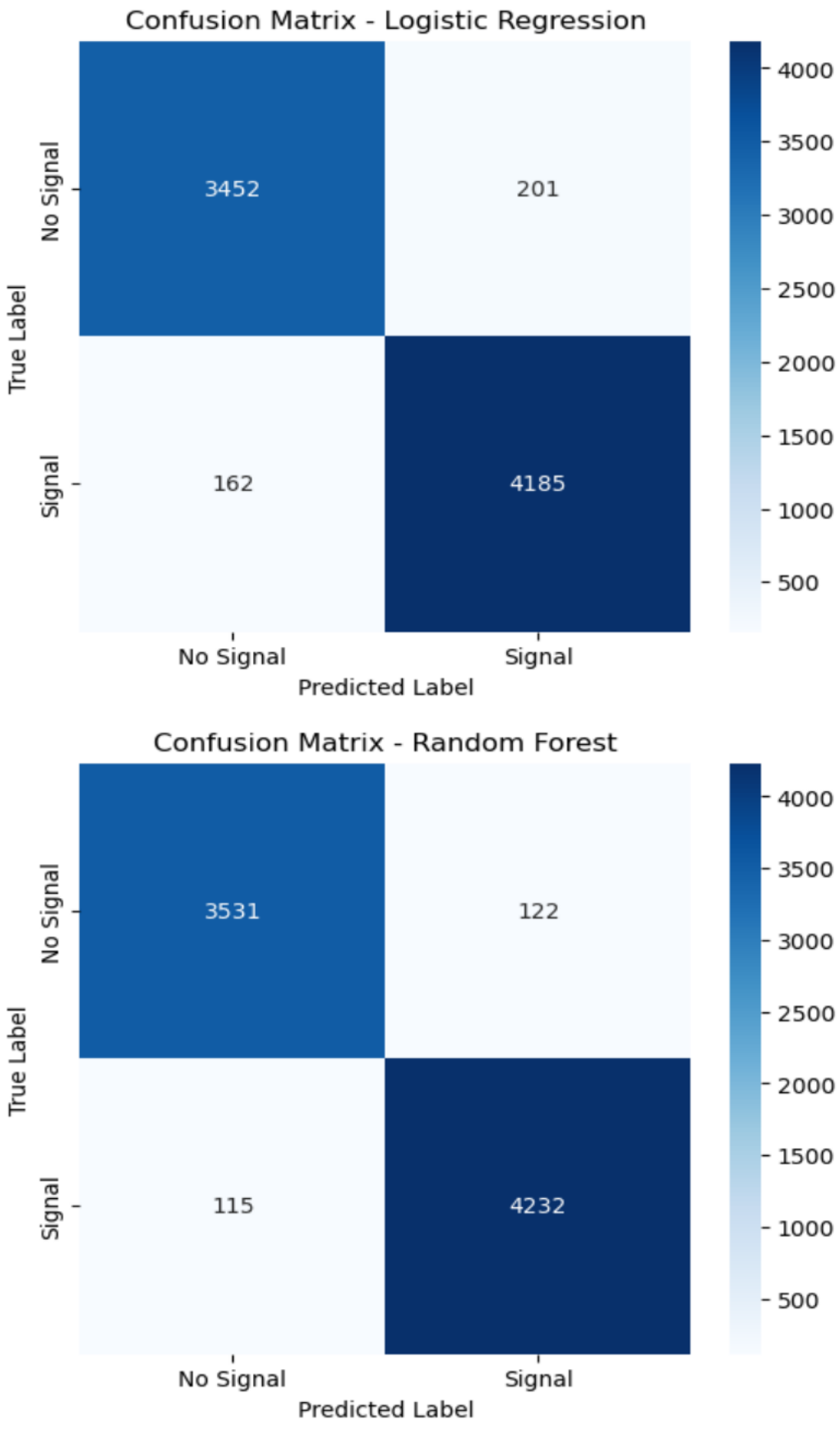
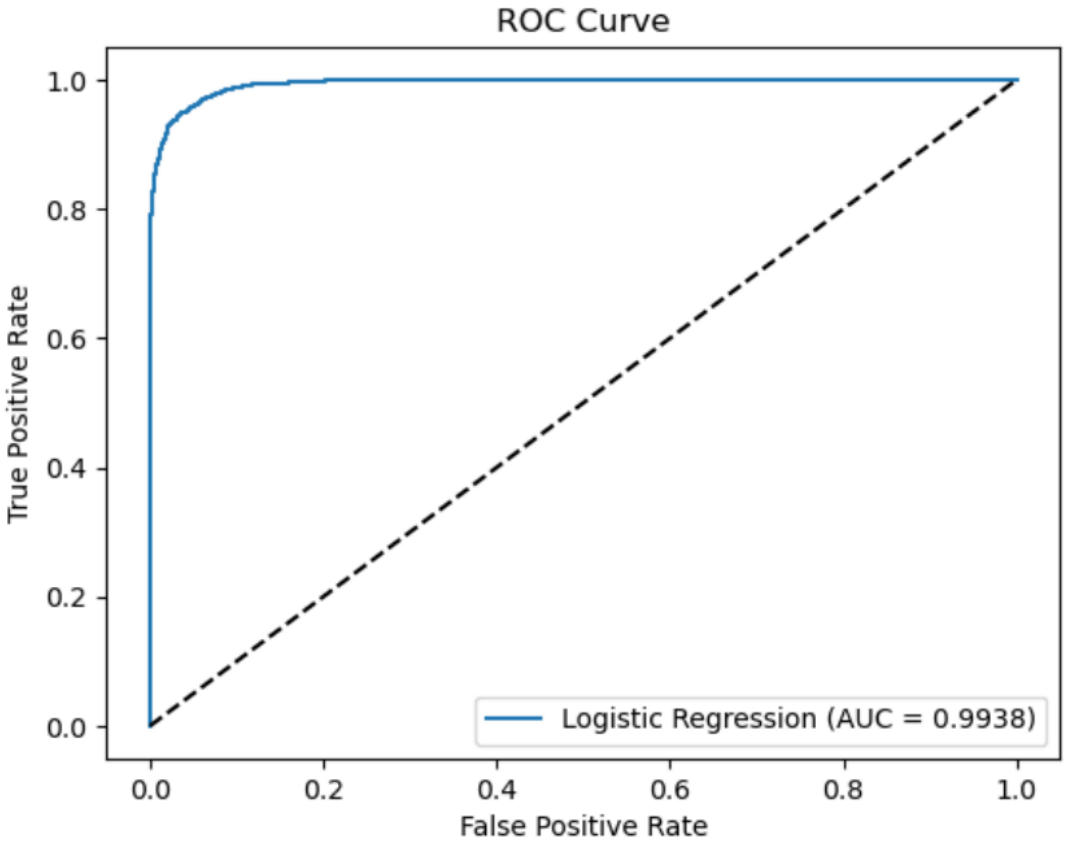


Figure 4.5: Confusion Matrix for Binary Classification.



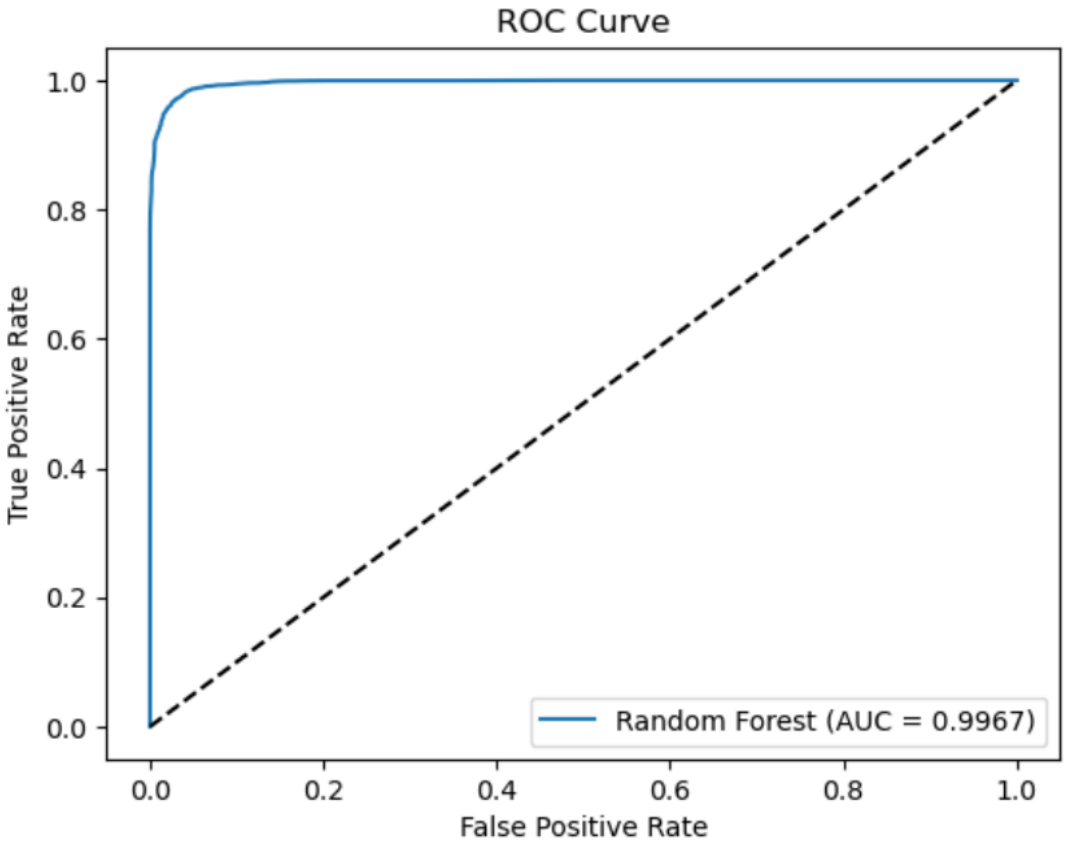


Figure 4.6: ROC Curve for Binary Classification.

Interpretability is crucial in network applications as it enables decision-makers to understand the underlying factors influencing model predictions and to take informed operational actions (Guo *et al.*, 2020). As depicted in Figure 4.7, the Logistic Regression feature importance plot using the values of each variable's absolute coefficients highlights that variables such as Power, SINR, RSRQ, and EARFCN significantly impact the model's predictions of network strength. Similarly, Figure 4.8 emphasises that Power, RSRQ, SINR, and distance are vital predictors in the Random Forest model. Collectively, these visual analyses lead to the conclusion that Power, SINR, and RSRQ are consistently the most influential features in predicting RSRP, underscoring their central role in assessing network performance.

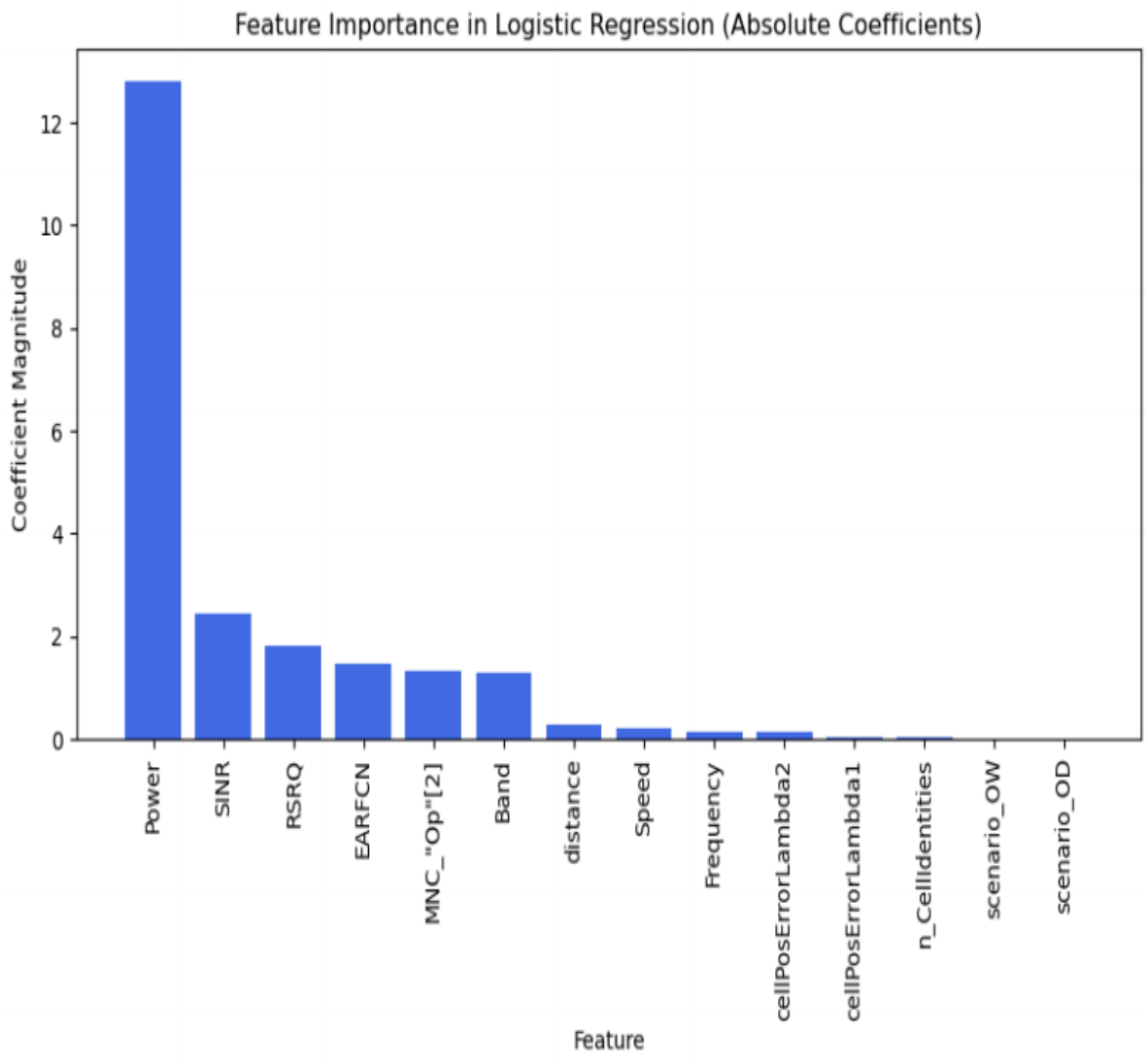


Figure 4.7: Feature Importance in Logistic Regression for Binary Classification.

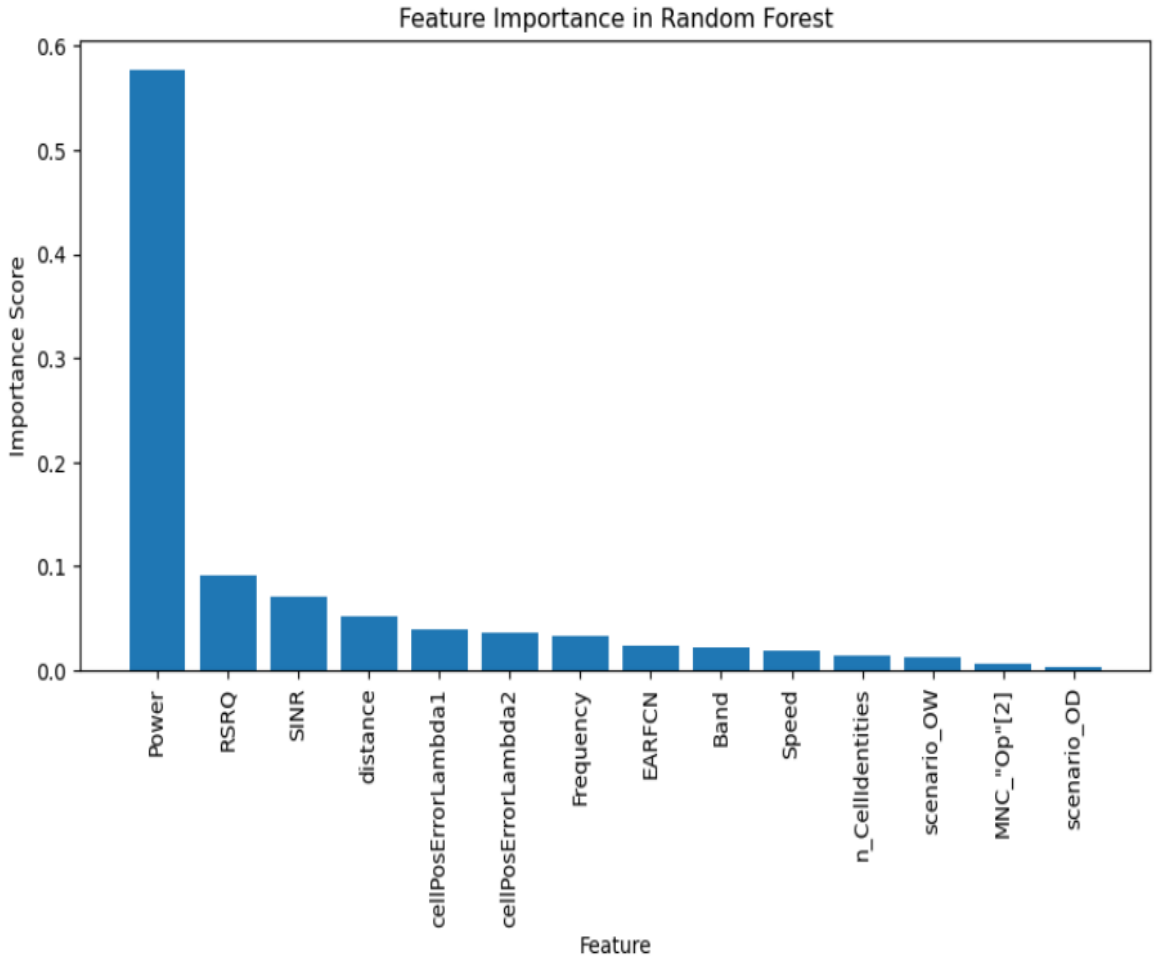


Figure 4.8: Feature Importance in Random Forest for Binary Classification.

4.4 Multi-Class Classification Analysis

4.4.1 Model Selection and Training

In this multi-class classification analysis, the Scenario variable with multiple unique values was selected as the target variable. Feature variables were selected based on statistical significance, retaining only those with p-values below 0.05 to ensure meaningful predictors for classifying individuals as outdoor driving, outdoor walking, or indoor static. In addition, the F-statistic was used to prioritise features with stronger predictive capabilities (Tenny and Abdelgawad, 2023). Consequently, a subset of features—namely UTC, Latitude, Longitude, Speed, PCI, Power, RSRP, cellLongitude, cellLatitude, Frequency, Altitude, distance and MNC—was extracted from the dataset. The target variable (scenario) was then encoded into three distinct classes—0 for Indoor Static, 1 for Outdoor Walking, and 2 for Outdoor Driving—to reduce dimensionality and enhance the model's predictive focus.

Following feature selection, the dataset was split into training and testing sets using an 80/20 ratio, with stratified sampling employed to preserve the original class distribution (Joseph, 2022). To address the multi-class classification task, two machine learning models were implemented: XGBoost Classifier and Random Forest Classifier. XGBoost, a highly efficient and scalable implementation of gradient boosting, optimises a multi-class loss function while incorporating regularisation to mitigate overfitting, making it effective for multi-class classifications (Chen and Guestrin, 2016). The Random Forest Classifier, also effectively applied in binary classification, was utilised for its strength in handling complex interactions between features (Fawagreh, Gaber and Elyan, 2014). Prior to model training, all selected features were standardised using Z-score normalisation and encoded where necessary, ensuring that algorithms sensitive to feature scale, such as XGBoost, could operate efficiently and produce reliable results (Chen and Guestrin, 2016).

4.3.2 Evaluation Metrics and Results

Both the Random Forest and XGBoost classifiers demonstrated high performance, with Random Forest achieving 99.72% accuracy and XGBoost reaching approximately 100%. These results, supported by high precision, recall, and F1 scores across all classes, highlight the effectiveness of the selected features and preprocessing pipeline. In terms of efficiency, XGBoost also proved faster, completing model development in approximately 2.5 seconds compared to Random Forest's 9.8 seconds.

The ROC curves in Figures 4.9 and 4.10 and the confusion matrices presented in Figures 4.11 and 4.12 illustrate the classification performance of the Random Forest model across the three classes. For **Class 0**, the model correctly identified 3,308 instances (TP), with four false positives and two false negatives, resulting in 4,686 true negatives, which was calculated as the total number of instances (8,000) minus the sum of TP, FP, and FN. For **Class 1**, the model achieved 3,419 true positives, 17 false positives (including misclassifications from Class 0 and Class 2), and five false negatives, giving a true negative count of 4,559. Regarding **Class 2**, there were 1,251 TP, just one FP (from Class 1), and 15 FN, leading to 6,734 TN. These results demonstrate that the Random Forest model performs well in identifying Class 0 and Class 1 instances, with minimal misclassifications. The relatively high TN values across all classes indicate strong overall classification accuracy and robustness of the model across the multiclass scenario.

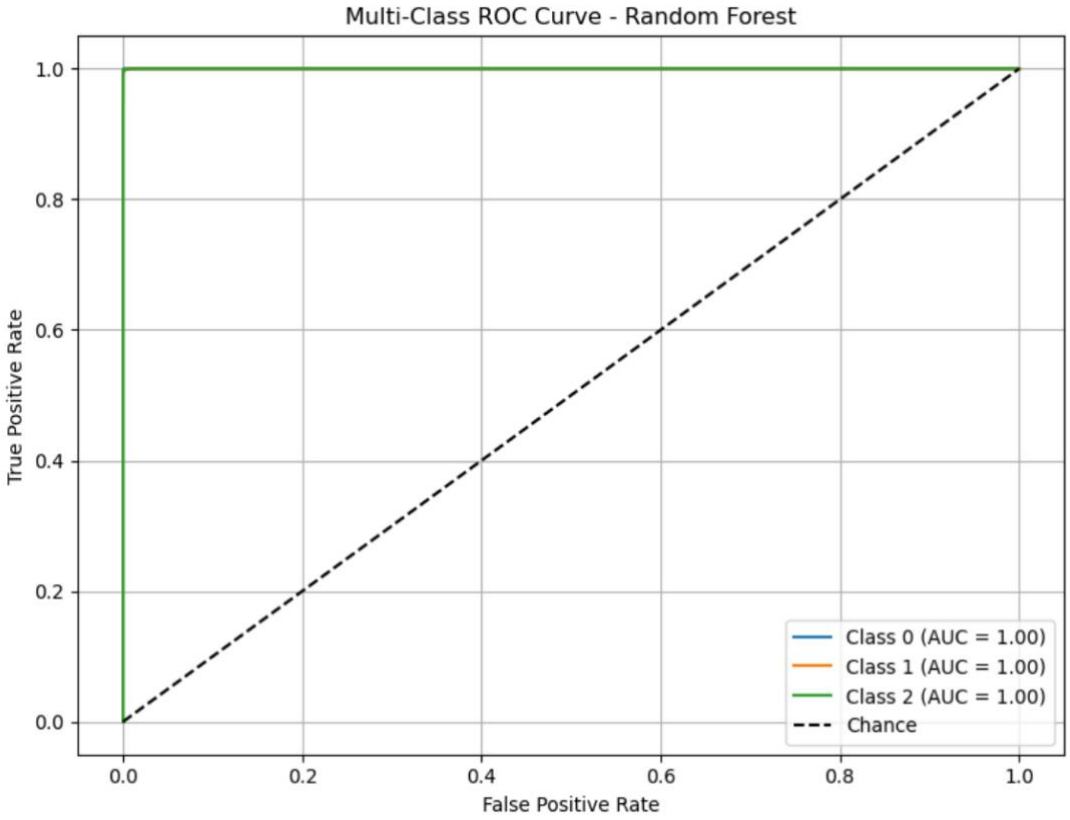


Figure 4.9: Random Forest ROC-AUC for Multi-Classification.

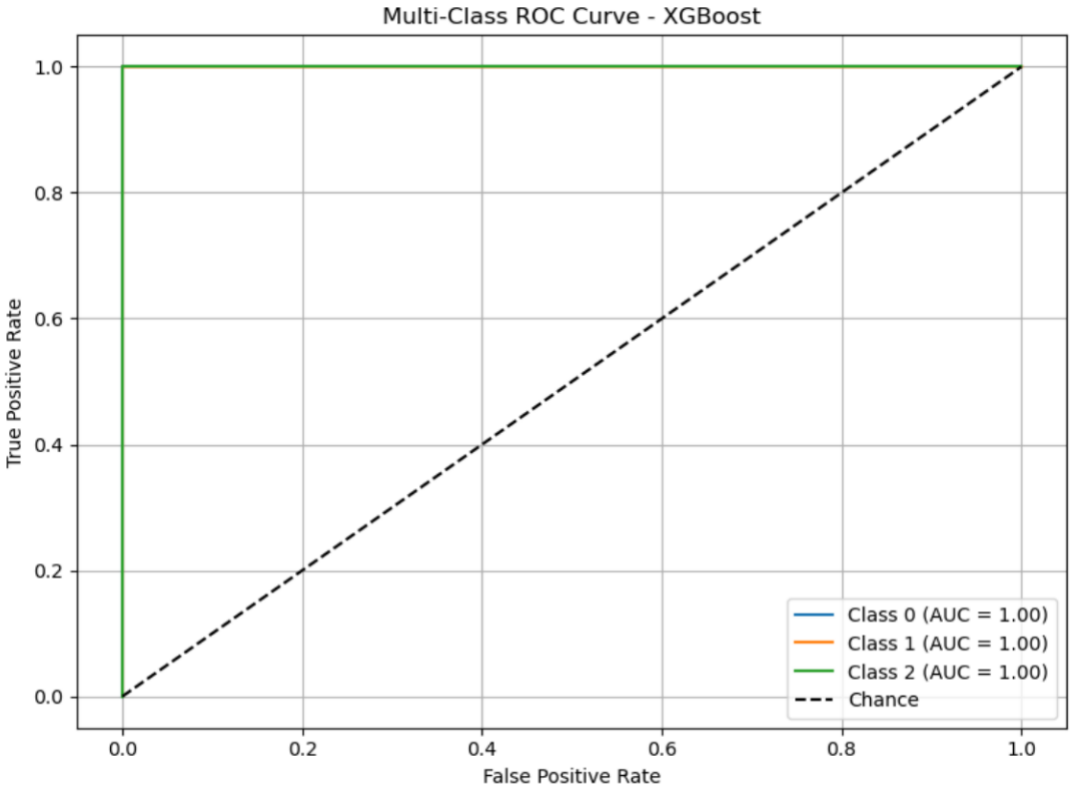


Figure 4.10: XGBoost ROC-AUC for Multi-Classification.

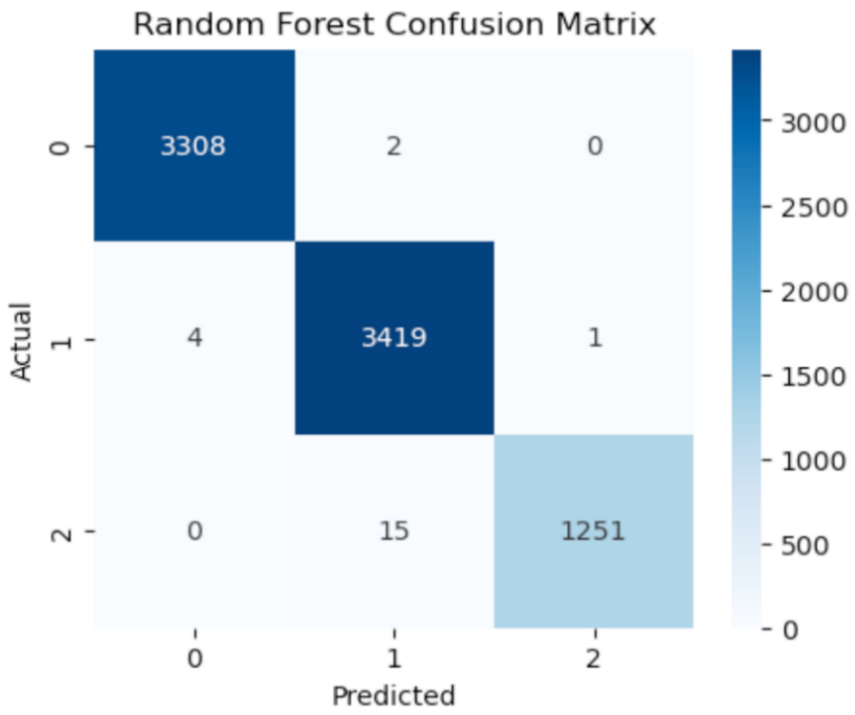


Figure 4.11: Random Forest Confusion Matrix for Multi-Classification.

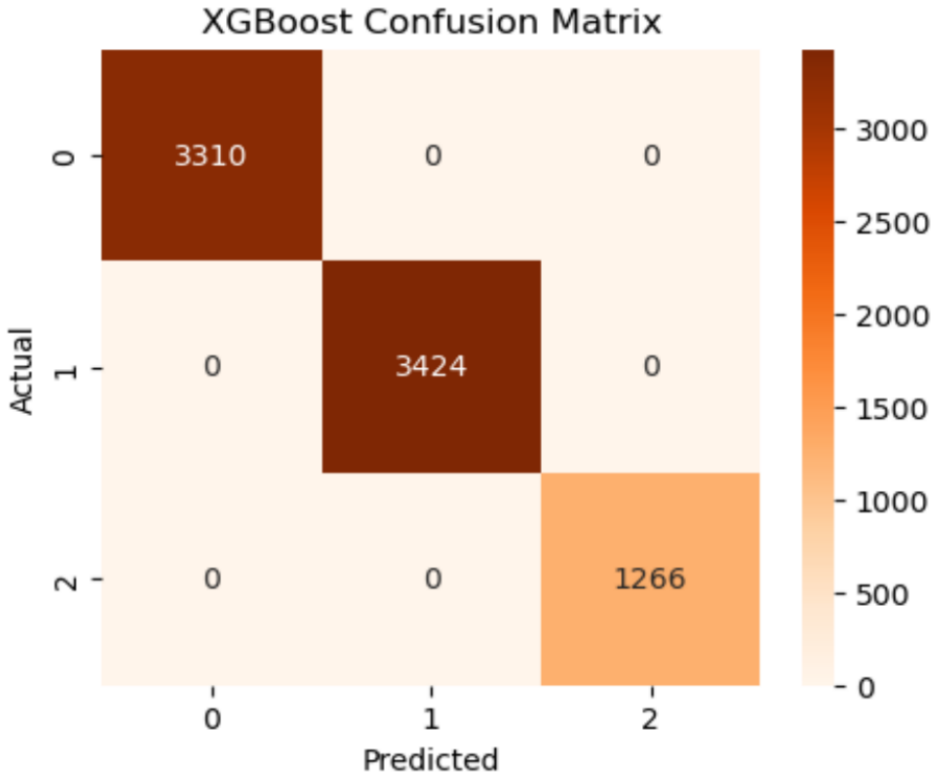


Figure 4.12: XGBoost Confusion Matrix for Multi-Classification.

In comparison, the XGBoost confusion matrix demonstrates better classification: for Class 0, TP is 3,310 with no FP or FN (thus TN = 4,690); for Class 1, TP is 3,424 with zero FP and FN (TN = 4,576); and for Class 2, TP is 1,266 with zero FP and FN, leading to TN = 6,734. This detailed breakdown shows how each model performs in correctly classifying each class, with XGBoost exhibiting marginally better performance by eliminating nearly all misclassifications.

To enhance interpretability, feature importance was evaluated for both models. The Random Forest feature importance plot showed that key predictors including, Longitude, UTC, Speed and Latitude, among others, played pivotal roles in determining the network scenario, as shown in Figure 4.13. The XGBoost model's feature importance analysis yielded a comparable ranking in Figure 4.14, underscoring the consistency and robustness of these variables in predicting network performance. These results highlight that the selected features, chosen based on statistical criteria, capture the essential characteristics of the network environment and are instrumental in accurately classifying the different operational scenarios. The high classification accuracies and feature importance outcomes reinforce the reliability of the methodology and provide valuable insights for operational decision-making in network management.

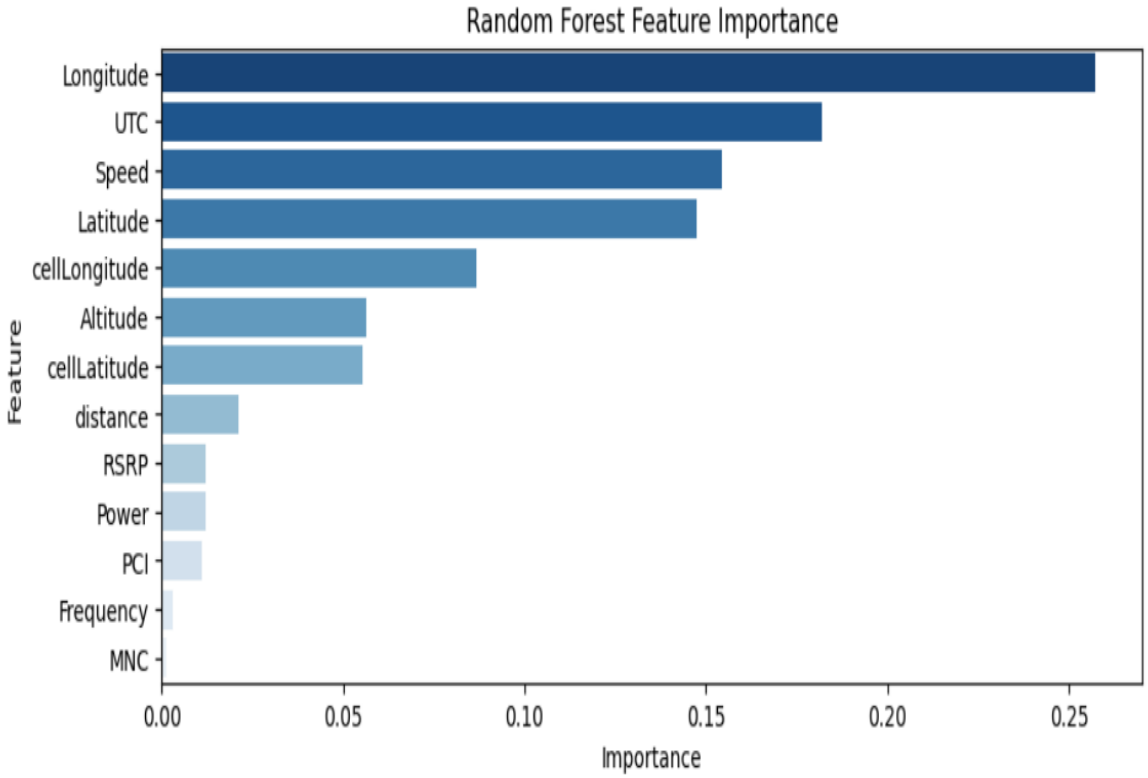


Figure 4.13: Feature Importance in Random Forest for Multi-Classification.

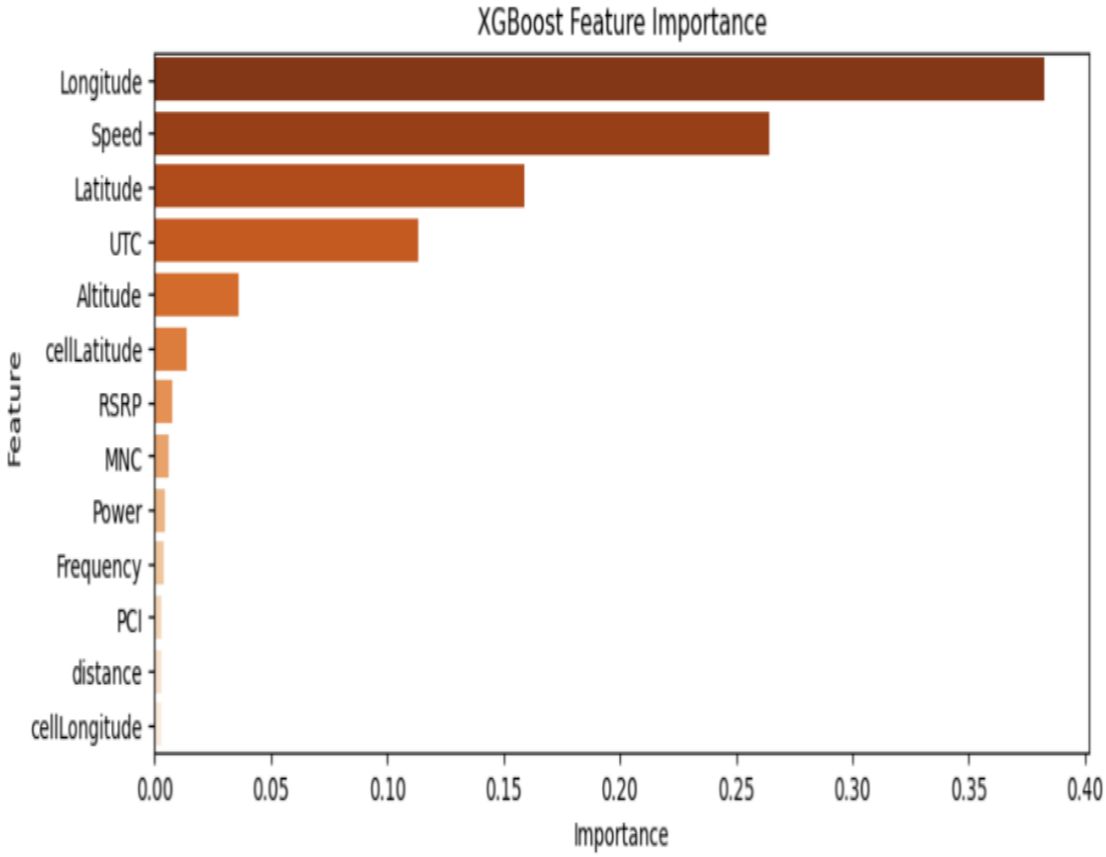


Figure 4.14: Feature Importance in XGBoost for Multi-Classification.

5.0 Optimisation Using Genetic Algorithms

This chapter details the formulation and implementation of optimisation techniques using Genetic Algorithms (GA) to enhance machine learning models (Figure 5.1). The optimisation process focuses on tuning hyperparameters for Random Forest classification by defining objective functions, decision variables, feasible bounds, and necessary constraints. The aim is to demonstrate the improvement in model performance when optimised hyperparameters are applied, as measured by metrics such as accuracy, along with other performance metrics for classification. Genetic Algorithms have been widely used for hyperparameter tuning in complex machine-learning problems due to their ability to navigate high-dimensional search spaces and identify near-optimal solutions without exhaustive grid search (Mehdary *et al.*, 2024).

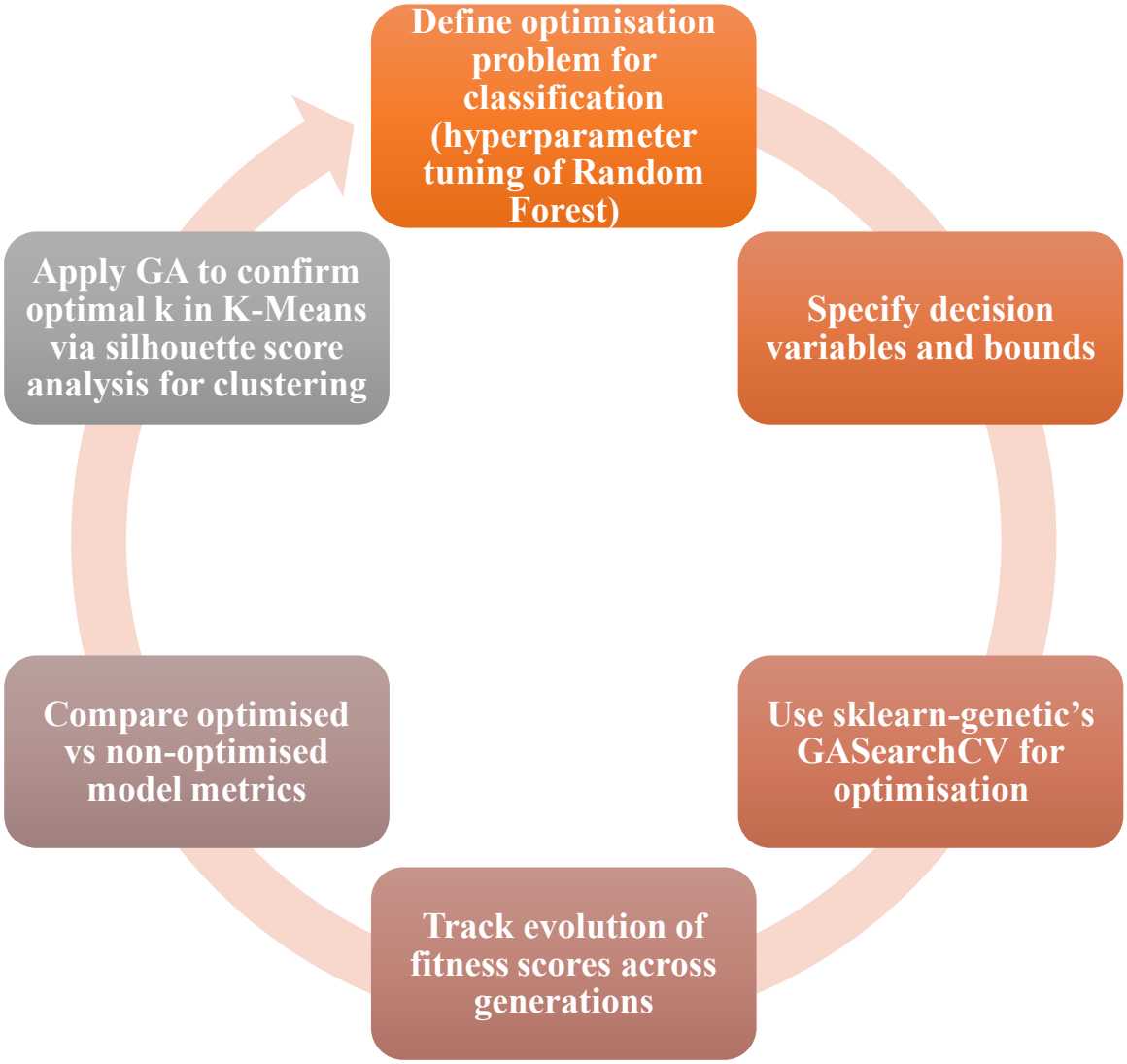


Figure 5.1: Optimisation Workflow for Hyperparameter Tuning Using Genetic Algorithms

5.1 Optimization for Classification

In this classification task, the objective was to enhance the predictive performance of a Random Forest classifier used to classify Reference Signal Received Power (RSRP) levels into binary categories. Random Forest models are known to be sensitive to hyperparameters such as the number of trees (n_estimators), the maximum depth of the trees (max_depth), the minimum number of samples required to split a node (min_samples_split), and the minimum samples per leaf (min_samples_leaf). Additionally, the choice of splitting criterion (gini or entropy) can also influence classification outcomes. The optimisation goal was to identify the best combination of these parameters to maximise classification accuracy on a validation set. To define the optimisation problem, an objective function was specified to maximise validation accuracy based on the selected hyperparameters. Bounds for each decision variable were determined based on practical considerations. The number of trees was set to vary between 50 and 500, the maximum depth between 3 and 50, min_samples_split between 2 and 10, and min_samples_leaf between 1 and 5. The splitting criterion was restricted to either "gini" or "entropy. " All variables were constrained to be integers, as tree-based models require discrete configuration values. The chosen bounds ensured that the model could be trained efficiently without excessive computational overhead.

5.2 Implementation of Genetic Algorithms

To optimise the Random Forest hyperparameters, a Genetic Algorithm (GA) was implemented using the GASearchCV module from the sklearn-genetic library. Genetic Algorithms simulate the process of natural evolution and are well-suited for exploring complex, non-linear search spaces. In this implementation, each individual in the population represented a different set of hyperparameters. The GA iteratively evolved this population through a series of biologically inspired operations, including selection, crossover, and mutation. The optimisation process began with the random initialisation of a population of 20 candidate solutions, each sampled from the defined search space. The fitness of each individual was evaluated by training a Random Forest model using 3-fold cross-validation and computing the average accuracy. Individuals with higher fitness were more likely to be selected for reproduction using evolutionary strategies. During the crossover phase, parent solutions exchanged segments of their hyperparameter configurations to generate new offspring. The algorithm used a mutation probability of 0.1 and a crossover probability of 0.8 to balance exploration and exploitation. This process, which introduced variability and maintained population diversity, was repeated over 10 generations, with the evolution of classification accuracy illustrated in Figure 5.2.

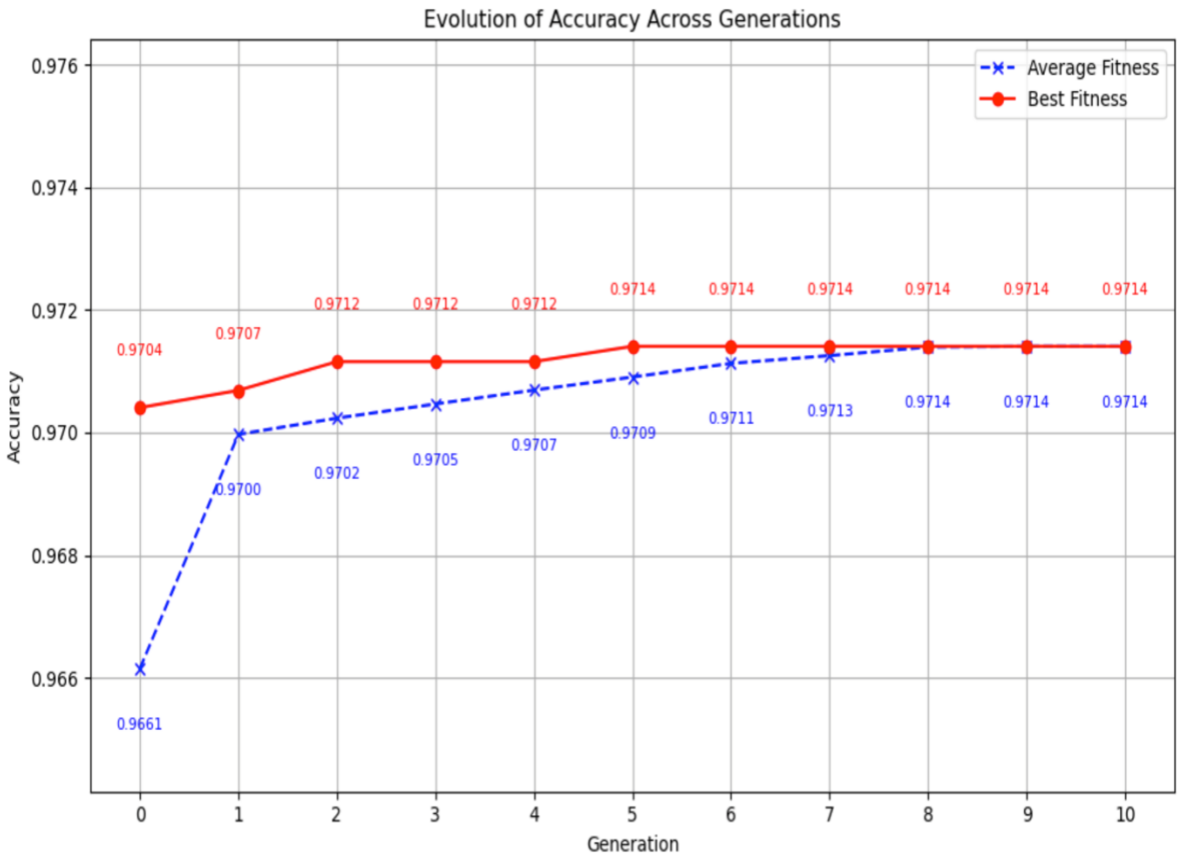


Figure 5.2: Evolution of Accuracy Over Generations (GA).

The implementation of GASearchCV allowed for seamless integration with scikit-learn pipelines and supported parallel execution, making it an efficient and practical approach for hyperparameter optimisation. Instead of relying on brute force grid search, the Genetic Algorithm efficiently explored the hyperparameter space, including n_estimators, max_depth, and related parameters, while balancing performance trade-offs (Mehdary *et al*., 2024). This approach helped prevent overfitting by selecting an optimal combination of tree quantity and depth, ultimately enhancing generalisation on unseen data. Upon completion, the best-performing configuration, consisting of n_estimators: 163, max_depth: 48, min_samples_split: 2, min_samples_leaf: 1, and criterion: 'entropy', was used to train the final Random Forest model.

5.3 Comparison of Optimised and Non-Optimised Models

To assess the impact of the Genetic Algorithm, the optimised model was compared to a baseline Random Forest with default settings. Evaluated on the same test set, the optimised model showed improvements of approximately 0.15%, 0.05%, 0.23%, and 0.14% in accuracy, precision, recall, and F1-score, respectively (Figure 5.3).

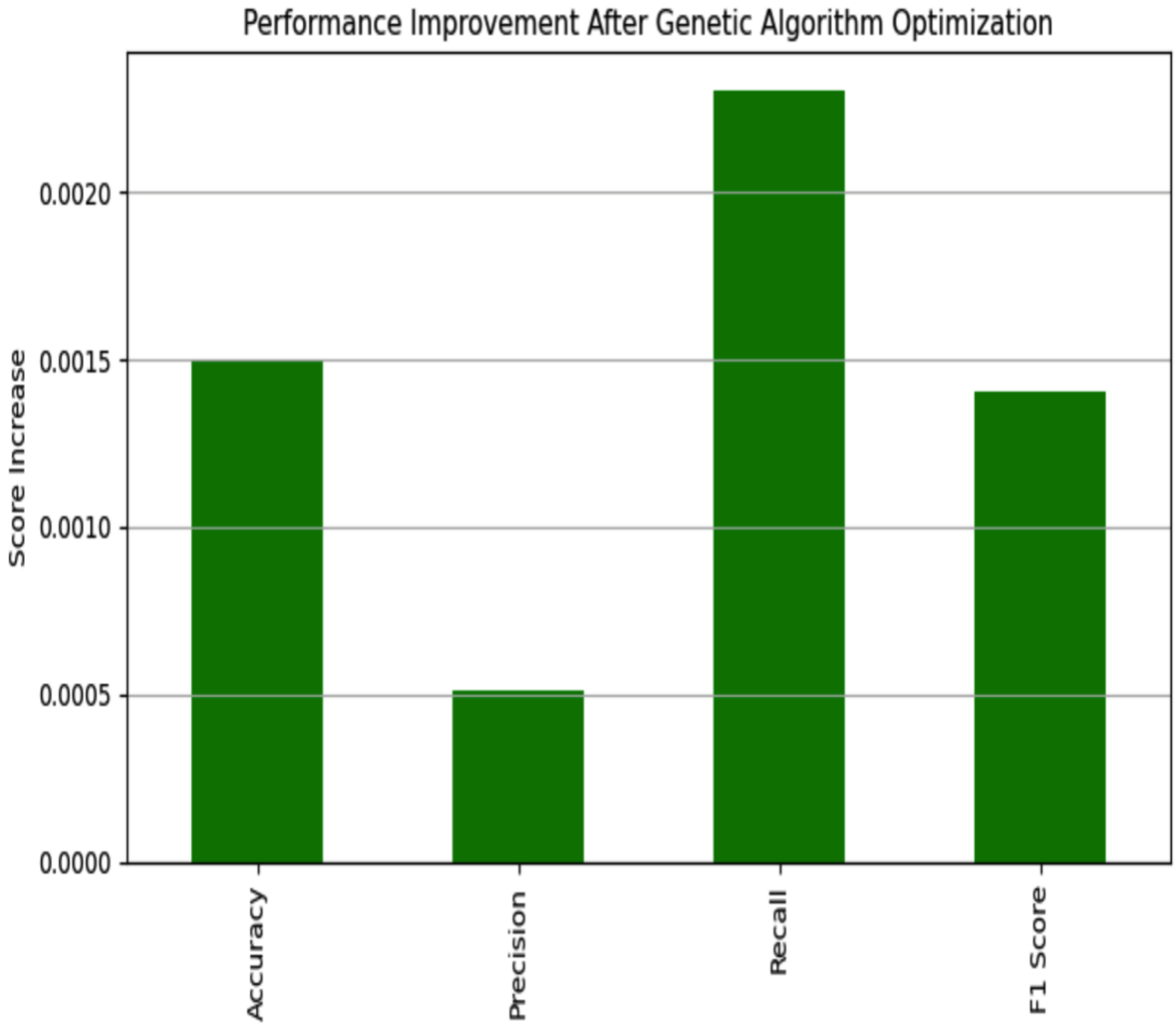


Figure 5.3: Performance Improvement after GA.

These improvements validate the effectiveness of the Genetic Algorithm in discovering a more optimal hyperparameter configuration. Additionally, the confusion matrix of the optimised model revealed reduced FP and FN, and increased TP and TN compared to the baseline, indicating better generalisation and classification balance, as shown in Figures 5.4 and 5.5. The results suggest that evolutionary optimisation methods like Genetic Algorithms offer a practical advantage when dealing with complex models and hyperparameter spaces. By automating the tuning process and systematically exploring the search space, the GA contributed to a measurable improvement in predictive performance, thereby fulfilling the objective of the optimisation task.

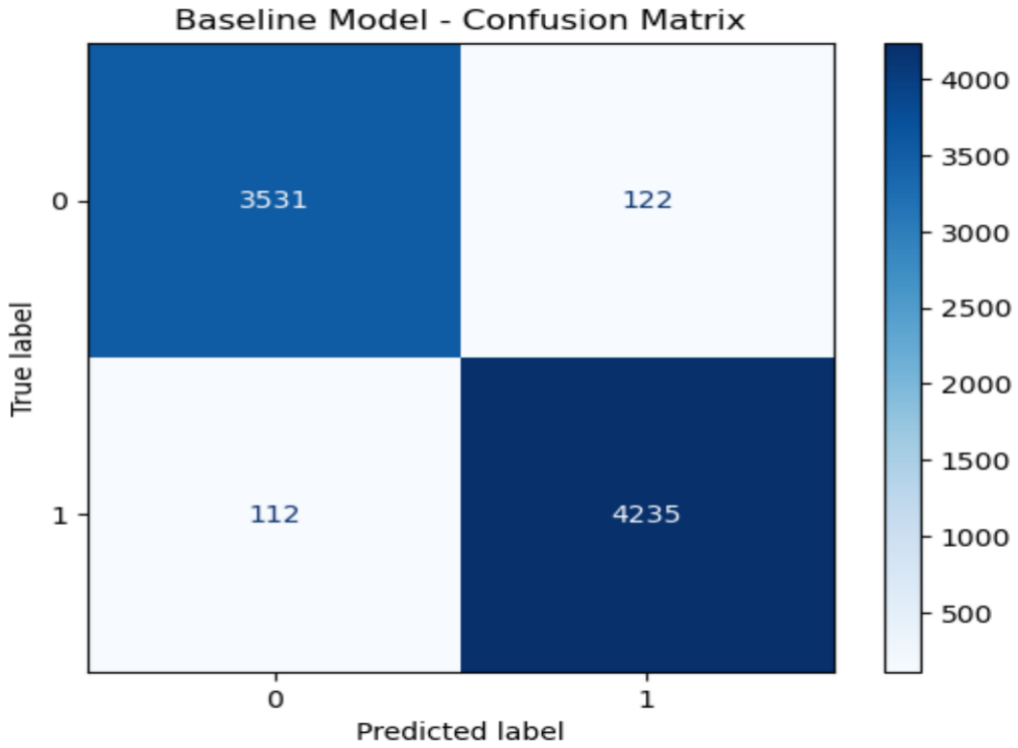


Figure 5.4: Baseline Model RF

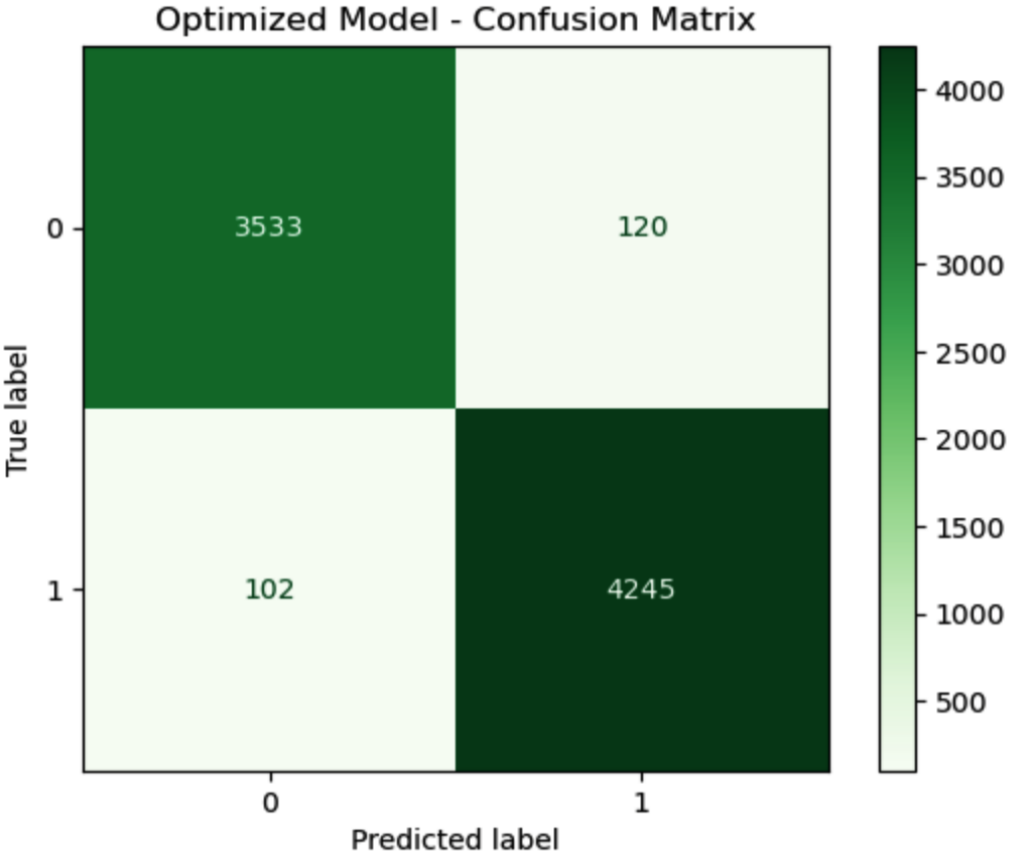


Figure 5.5: Optimised Model Random Forest

Additionally, to optimise the number of clusters (k) in the K-Means algorithm, a Genetic Algorithm (GA) was implemented using the silhouette score as the fitness function. Each individual in the GA population represented a candidate k value (ranging from 2 to 10), and fitness was evaluated by fitting K-Means and computing the silhouette score, as shown in Figure 5.6. Through iterative evolution involving crossover, mutation, and selection, the GA effectively explored the solution space and identified the optimal k as k = 4. This approach addressed the limitations of manual heuristics such as the Elbow Method, which initially indicated k = 4. However, the GA validated and reinforced the choice of 4 clusters through a more robust, data-driven process.

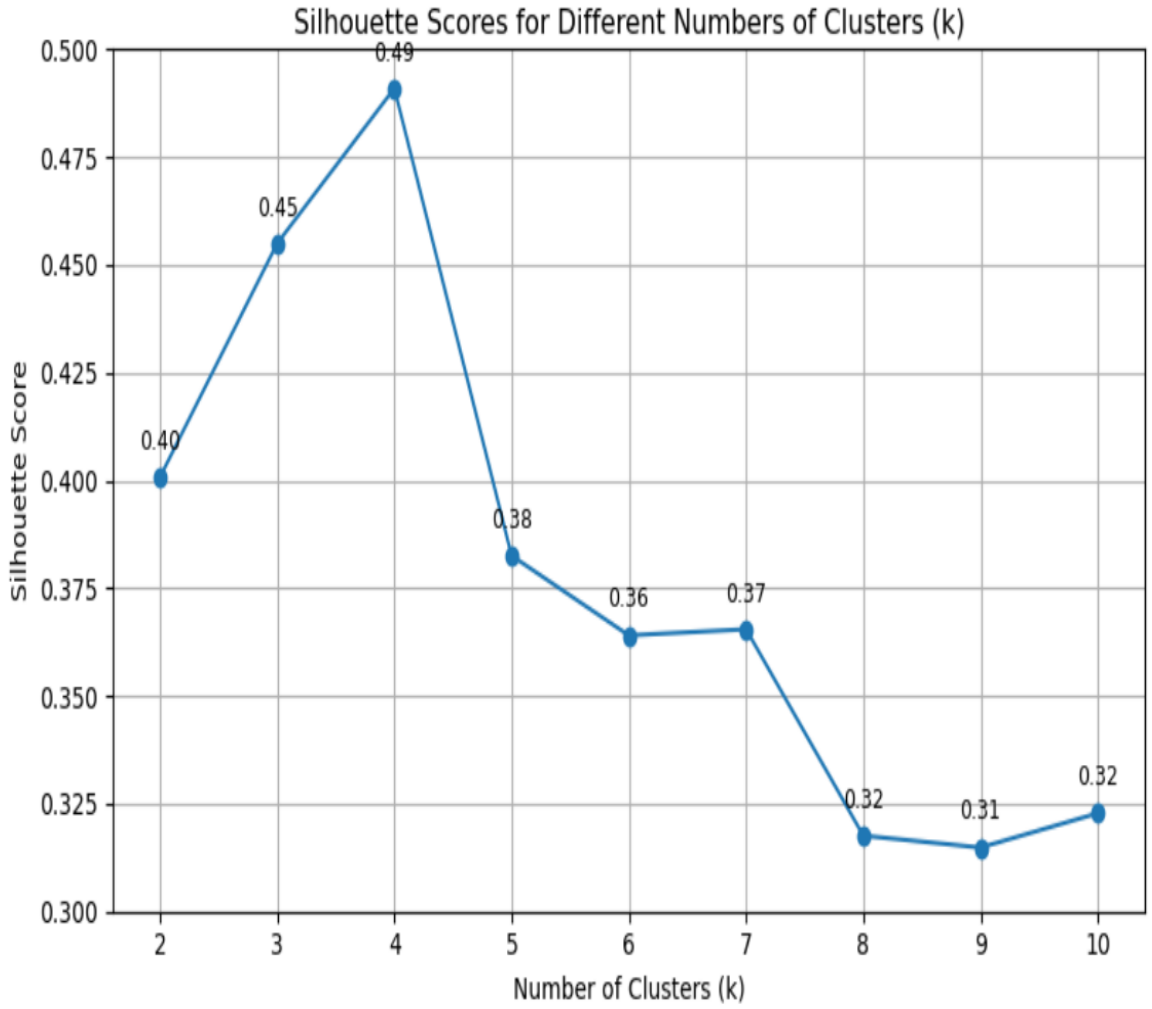


Figure 5.6: Silhouette Scores for K-means Clustering

Conclusion

This study demonstrated the effective application of machine learning techniques to optimise LTE network performance using a large-scale passive measurement dataset. A comprehensive pipeline comprising data preprocessing, clustering, classification, and genetic algorithm-based optimisation was developed to extract actionable insights into radio signal behaviour under diverse network conditions. Clustering analysis revealed user mobility and signal strength patterns by incorporating features such as distance to the serving cell, user speed, and frequency band. Among the classification models applied, XGBoost and Random Forest achieved high predictive accuracy in both scenario identification and signal strength prediction tasks. The use of Genetic Algorithms for hyperparameter tuning further enhanced model performance, highlighting their effectiveness in optimising complex machine learning workflows.

Despite these results, the study faced several limitations. The analysis was based on a single dataset, which limits the generalisability of findings to broader LTE or emerging 5G environments. Also, due to computational constraints, only 8% of the dataset was analysed, potentially omitting rare but meaningful patterns. Additionally, the focus on core metrics such as RSRP and speed could be expanded by incorporating contextual features such as device type, terrain, or weather conditions. Furthermore, the approximately 100% accuracy in the machine learning models may also indicate overfitting, where models memorise training data without generalising well to new instances. Although, this risk could be mitigated through regularisation techniques, cross-validation, or by increasing the size and diversity of the dataset.

All data were also anonymised, with no personally identifiable information processed, ensuring adherence to ethical standards. Based on these findings, it is recommended that mobile network operators implement adaptive machine learning pipelines for real-time network performance monitoring. Future research could explore integration with real-time, multi-source data, and the use of explainable AI methods such as SHAP or LIME, alongside deep learning and hybrid models, to further enhance LTE and 5G optimisation strategies.

References

5GWorldPro.com (2022). *What is difference between PCI in 4G LTE and PCI in 5G NR*. [online] 5G Training and 5G Certification. Available at: https://www.5gworldpro.com/blog/2020/11/11/what-is-difference-between-pci-in-4g-lte-and-pci-in-5g-nr/ [Accessed 30 Mar. 2025].

Aggarwal, D., Teja, S.C.R. and Mittal, S. (2024). A Stacking Ensemble Technique to Predict Speed and Distance in 4G and 5G Communication Datasets. *2024 IEEE International Symposium on Smart Electronic Systems (iSES)*, [online] pp.146–151. doi:https://doi.org/10.1109/ises63344.2024.00038.

Akpaneno, A.F., Akinbolati, A. and Ekundayo, R.K. (2024). Spatial Variation of the Received Signal Strength of Mobile Telephone Network (MTN) over Dutsin-Ma Town, Katsina State, Nigeria. *Journal of Basics and Applied Sciences Research (JOBASR)* , 2(1), pp.28–34. doi:https://doi.org/10.33003/jobasr-2024-v2i1-9.

Amini, A., Wah, T.Y. and Saboohi, H. (2014). On Density-Based Data Streams Clustering Algorithms: A Survey. *Journal of Computer Science and Technology*, [online] 29(1), pp.116–141. doi:https://doi.org/10.1007/s11390-013-1416-3.

Basu, I. and Maji, S. (2022). Multicollinearity Correction and Combined Feature Effect in Shapley Values. *Lecture notes in computer science*, pp.79–90. doi:https://doi.org/10.1007/978-3-030-97546-3_7.

Binding, C. and Tudhope, D. (2023). Automatic Normalization of Temporal Expressions. *Journal of Computer Applications in Archaeology*, 6(1), pp.24–39. doi:https://doi.org/10.5334/jcaa.105.

Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F. and Cho, Y.-I. (2024). Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. *Mathematics*, [online] 12(16), pp.2553–2553. doi:https://doi.org/10.3390/math12162553.

Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F. and Caicedo, O.M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1). doi:https://doi.org/10.1186/s13174-018-0087-2.

Bui, T.T., Nguyen, L.D., Kha, H.H., Vo, N.-S. and Duong, T.Q. (2023). Joint Clustering and Resource Allocation Optimization in Ultra-Dense Networks with Multiple Drones as Small Cells Using Game Theory. *Sensors*, [online] 23(8), pp.3899–3899. doi:https://doi.org/10.3390/s23083899.

Chen, T. and Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp.785–794. doi:https://doi.org/10.1145/2939672.2939785.

Cooksey, R.W. (2020). Descriptive Statistics for Summarising Data. *Illustrating Statistical Procedures: Finding Meaning in Quantitative Data*, [online] 1(1), pp.61–139. doi:https://doi.org/10.1007/978-981-15-2537-7_5.

Das, A. (2023). Logistic Regression. *Springer eBooks*, pp.3985–3986. doi:https://doi.org/10.1007/978-3-031-17299-1_1689.

Dastjerdy, B., Saeidi, A. and Heidarzadeh, S. (2023). Review of Applicable Outlier Detection Methods to Treat Geomechanical Data. *Geotechnics*, [online] 3(2), pp.375–396. doi:https://doi.org/10.3390/geotechnics3020022.

DeCastro-García, N., Muñoz, L., Rodríguez, M.F. and Carriegos, M.V. (2018). On Detecting and Removing Superficial Redundancy in Vector Databases. *Mathematical Problems in Engineering*, [online] 2018(c), pp.1–14. doi:https://doi.org/10.1155/2018/3702808.

Deng, D., 2020, September. DBSCAN clustering algorithm based on density. In *2020 7th international forum on electrical engineering and automation (IFEEA)* (pp. 949-953). IEEE.

Dougba-Noel, D., Gnoan, K., Justin, A., Konan Alphonse, A., Désiré, L., Dramane, D., Nafan, D. and Malerba, G. (2021). Biostat Biom Open Access J Normality Assessment of Several Quantitative Data Transformation Procedures. *Biostat Biom Open Access J*, [online] 10(3). Available at: https://juniperpublishers.com/bboaj/pdf/BBOAJ.MS.ID.555786.pdf [Accessed 30 Mar. 2025].

Dritsas, E. and Trigka, M. (2022). Lung Cancer Risk Prediction with Machine Learning Models. *Big Data and Cognitive Computing*, 6(4), p.139. doi: https://doi.org/10.3390/bdcc6040139.

Dudáš, A. (2024). Graphical representation of data prediction potential: correlation graphs and correlation chains. *The Visual Computer*, 40. doi:https://doi.org/10.1007/s00371-023-03240-y.

Eckhardt, C.M., Madjarova, S.J., Williams, R.J., Ollivier, M., Karlsson, J., Pareek, A. and Nwachukwu, B.U. (2022). Unsupervised machine learning methods and emerging applications in healthcare. *Knee Surgery, Sports Traumatology, Arthroscopy*, 31(2), pp.376–381. doi:https://doi.org/10.1007/s00167-022-07233-7.

Fan, C., Chen, M., Wang, X., Wang, J. and Huang, B. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*, [online] 9. doi:https://doi.org/10.3389/fenrg.2021.652801.

Fatima, N. (2024). *Multivariate Analysis: Exploring Relationships Between Variables*. [online] Medium. Available at: https://medium.com/%40noorfatimaafzalbutt/multivariate-analysis-exploring-relationships-between-variables-87a3363bd320 [Accessed 30 Mar. 2025].

Fawagreh, K., Gaber, M.M. and Elyan, E. (2014). Random forests: from early developments to recent advancements. *Systems Science & Control Engineering*, 2(1), pp.602–609. doi:https://doi.org/10.1080/21642583.2014.956265.

Fortin, F.-A., Rainville, F.-M.D., Gardner, M.A. and Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, [online] 13, pp.2171–2175. Available at: https://www.researchgate.net/publication/235707001_DEAP_Evolutionary_algorithms_made_easy.

Guo, M., Zhang, Q., Liao, X. and Zeng, D.D. (2020). An interpretable neural network model through piecewise linear approximation. *arXiv.org*. [online] Available at: https://arxiv.org/abs/2001.07119 [Accessed 5 Apr. 2025].

Hu, K. (2020). Become Competent within One Day in Generating Boxplots and Violin Plots for a Novice without Prior R Experience. *Methods and Protocols*, [online] 3(4). doi:https://doi.org/10.3390/mps3040064.

Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B. and Heming, J. (2022). K-means Clustering Algorithms: a Comprehensive Review, Variants Analysis, and Advances in the Era of Big Data. *Information Sciences*, 622(622). doi:https://doi.org/10.1016/j.ins.2022.11.139.

Jia, W., Sun, M., Lian, J. and Hou, S. (2022). Feature Dimensionality reduction: a Review. *Complex & Intelligent Systems*, [online] 8, pp.2663–2693. doi:https://doi.org/10.1007/s40747-021-00637-x.

Joseph, V.R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, [online] 15(4), pp.531–538. doi:https://doi.org/10.1002/sam.11583.

Kamalov, F. and Sulieman, H. (2021). *Time series signal recovery methods: comparative study*. [online] ResearchGate. doi:https://doi.org/10.48550/arXiv.2110.12631.

Kang, H. (2013). The Prevention and Handling of the Missing Data. *Korean Journal of Anesthesiology*, 64(5), pp.402–406. doi:https://doi.org/10.4097/kjae.2013.64.5.402.

Kim-Geok, T., Zar-Aung, K., Sandar-Aung, M., Thu-Soe, M., Abdaziz, A., Pao-Liew, C., Hossain, F., Tso, C.P. and Yong, W.H. (2020). Review of Indoor Positioning: Radio Wave Technology. *Applied Sciences*, 11(1), p.279. doi:https://doi.org/10.3390/app11010279.

Komorowski, M., Marshall, D.C., Salciccioli, J.D. and Crutain, Y. (2016). Exploratory Data Analysis. *Secondary Analysis of Electronic Health Records*, [online] pp.185–203. doi:https://doi.org/10.1007/978-3-319-43742-2_15.

Kousias, K., Rajiullah, M., Caso, G., Ali, U., Alay, O., Brunstrom, A., Nardis, L.D., Neri, M. and Benedetto, M.-G.D. (2023). A Large-Scale Dataset of 4G, NB-IoT, and 5G Non-Standalone Network Measurements. *IEEE Communications Magazine*, [online] 62(5), pp.44–49. doi:https://doi.org/10.1109/mcom.011.2200707.

Kühl, N., Goutier, M., Hirt, R. and Satzger, G. (2019). Machine Learning in Artificial Intelligence: Towards a Common Understanding. *Hawaii International Conference on System Sciences (HICSS-52)*. [online] Available at: https://www.researchgate.net/publication/327802544_Machine_Learning_in_Artificial_Intelligence_Towards_a_Common_Understanding [Accessed 2 Apr. 2025].

Kuzudisli, C., Bakir-Gungor, B., Bulut, N., Qaqish, B. and Yousef, M. (2023). Review of feature selection approaches based on grouping of features. *PeerJ*, [online] 11, pp.e15666–e15666. doi:https://doi.org/10.7717/peerj.15666.

Kwak, S.K. and Kim, J.H. (2017). Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, [online] 70(4), pp.407–411. doi:https://doi.org/10.4097/kjae.2017.70.4.407.

Lai, H., Gao, K., Li, M., Li, T., Zhou, X., Zhou, X., Guo, H. and Fu, B. (2024). Handling missing data and measurement error for early-onset myopia risk prediction models. *BMC Medical Research Methodology*, [online] 24(1). doi:https://doi.org/10.1186/s12874-024-02319-x.

Lepot, M., Aubin, J.-B. and Clemens, F. (2017). Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. *Water*, 9(10), p.796. doi:https://doi.org/10.3390/w9100796.

Mahmud, M.S., Huang, J.Z., Salloum, S., Emara, T.Z. and Sadatdiynov, K. (2020). A survey of data partitioning and sampling methods to support big data analysis. *Big Data Mining and Analytics*, 3(2), pp.85–101.

Mehdary, A., Chehri, A., Jakimi, A. and Saadane, R. (2024). Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection. *Sensors*, [online] 24(4), p.1230. doi:https://doi.org/10.3390/s24041230.

Mendes-Santos, T., Turkeshi, X., Dalmonte, M. and Rodriguez, A. (2021). Unsupervised Learning Universal Critical Behavior via the Intrinsic Dimension. *Physical Review X*, 11(1). doi:https://doi.org/10.1103/physrevx.11.011040.

Noh, S.-K. and Choi, D. (2019). Propagation Model in Indoor and Outdoor for the LTE Communications. *International Journal of Antennas and Propagation*, 2019, pp.1–6. doi:https://doi.org/10.1155/2019/3134613.

Ododo, F. and Addotey, N., (2025). Understanding the influence of outliers on machine learning model interpretability. *International Journal of African Sustainable Development Research*, 7(2). https://doi.org/10.70382/tijasdr.v07i2.019

Pedapolu, P.K., Kumar, P., Harish, V., Venturi, S., Bharti, S.K., Kumar, V. and Kumar, S. (2016). Significance of Mobility on Received Signal Strength: An Experimental Investigation. [online] *arXiv preprint*. Available at: https://arxiv.org/abs/1611.06682 [Accessed 4 Apr. 2025].

Pedapolu, P.K., Kumar, P., Harish, V., Venturi, S., Bharti, S.K., Kumar, V. and Kumar, S. (2016). *Significance of Mobility on Received Signal Strength: An Experimental Investigation*. [online] arXiv.org. Available at: https://arxiv.org/abs/1611.06682 [Accessed 4 Apr. 2025].

Pfaehler, E., Mesotten, L., Zhovannik, I., Pieplenbosch, S., Thomeer, M., Vanhove, K., Adriaensens, P. and Boellaard, R. (2021). Plausibility and redundancy analysis to select FDG-PET textural features in non-small cell lung cancer. *Medical Physics*, 48(3), pp.1226–1238. doi:https://doi.org/10.1002/mp.14684.

Ramesh, P., Karuppasamy, R. and Veerappapillai, S. (2020). A Review on Recent Advancements in Diagnosis and Classification of Cancers Using Artificial Intelligence. *BioMedicine*, 10(3), pp.5–17. doi: https://doi.org/10.37796/2211-8039.1012.

Rathore, P., Kumar, D., Bezdek, J.C., Rajasegarar, S. and Palaniswami, M. (2019). A Rapid Hybrid Clustering Algorithm for Large Volumes of High Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, 31(4), pp.641–654. doi:https://doi.org/10.1109/tkde.2018.2842191.

Rodriguez, M.Z., Comin, C.H., Casanova, D., Bruno, O.M., Amancio, D.R., Costa, L. da F. and Rodrigues, F.A. (2019). Clustering algorithms: A comparative approach. *PLOS ONE*, [online] 14(1), p.e0210236. doi:https://doi.org/10.1371/journal.pone.0210236.

Sangeetha, S.K.B., Mathivanan, S.K., Rajadurai, H., Cho, J. and Easwaramoorthy, S.V. (2024). A multi-modal geospatial–temporal LSTM based deep learning framework for predictive modeling of urban mobility patterns. *Scientific Reports*, [online] 14(1). doi:https://doi.org/10.1038/s41598-024-74237-3.

Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, [online] 2(3), pp.1–21. doi:https://doi.org/10.1007/s42979-021-00592-x.

Sathyanarayanan, S. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, pp.4023–4031. doi:https://doi.org/10.53555/ajbr.v27i4s.4345.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (2017). DBSCAN Revisited, Revisited. *ACM Transactions on Database Systems*, [online] 42(3), pp.1–21. doi:https://doi.org/10.1145/3068335.

Shahapure, K.R. and Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. doi:https://doi.org/10.1109/dsaa49011.2020.00096.

Simpson, O. and Sun, Y. (2018). LTE RSRP, RSRQ, RSSNR and local topography profile data for RF propagation planning and network optimization in an urban propagation environment. *Data in Brief*, 21, pp.1724–1737. doi:https://doi.org/10.1016/j.dib.2018.08.137.

Sundaram, J., Gowri, K., Devaraju, S., Gokuldev, S., Jayaprakash, S., Anandaram, H., Manivasagan, C. and Thenmozhi, M. (2023). An Exploration of Python Libraries in Machine Learning Models for Data Science. *Advances in computational intelligence and robotics book series*, pp.1–31. doi:https://doi.org/10.4018/978-1-6684-8696-2.ch001.

Teltonika (2024). *RSRP and RSRQ - Teltonika Networks Wiki*. [online] wiki.teltonika-networks.com. Available at: https://wiki.teltonika-networks.com/view/RSRP_and_RSRQ.

Tenny, S. and Abdelgawad, I. (2023). *Statistical significance*. [online] *National Library of Medicine*. StatPearls Publishing. Available at: https://www.ncbi.nlm.nih.gov/books/NBK459346/ [Accessed 4 Apr. 2025].

Wang, Z., Isci, S., Kanza, Y., Kounev, V. and Shaqalle, Y. (2023). Cellular Network Optimization by Deep Reinforcement Learning and AI-Enhanced Ray Tracing. *In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Spatial Big Data and AI for Industrial Applications (GeoIndustry '23).*, pp.41–50. doi:https://doi.org/10.1145/3615888.3627814.

Wongoutong, C. (2024). The impact of neglecting feature scaling in k-means clustering. *PLOS ONE*, 19(12), p.e0310839. doi:https://doi.org/10.1371/journal.pone.0310839.

Yuan, C. and Yang, H. (2019). Research on K-Value Selection Method of K-Means Clustering Algorithm. *J*, 2(2), pp.226–235. doi:https://doi.org/10.3390/j2020016.

Zheng, K., Yang, Z., Zhang, K., Chatzimisios, P., Yang, K. and Xiang, W. (2016). Big data-driven optimization for mobile networks toward 5G. *IEEE Network*, 30(1), pp.44–51. doi:https://doi.org/10.1109/mnet.2016.7389830.

Zubair, Md., Iqbal, MD.A., Shil, A., Chowdhury, M.J.M., Moni, M.A. and Sarker, I.H. (2022). An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling. *Annals of Data Science*. doi:https://doi.org/10.1007/s40745-022-00428-2.