PROJECT REPORT (REC-852)

On

MODEL AIDED ARTIFICIAL INTELLIGENCE FOR PATH LOSS PREDICTION IN MOBILE COMMUNICATION

Submitted for partial fulfillment of award of the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Submitted By

ARPIT JAIN (1719231024)
JATIN SHARMA (1719231048)
AMAN HASAN KHAN (1719231014)
HIMANSHU RAWAT (1719231047)

Under the Guidance of

Dr. Piyush Yadav
(Associate Professor)

Deptt. of Electronics and Communication Engineering

G. L. BAJAJ INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Plot no. 2, Knowledge Park III, Gr. Noida

Session: 2020-21 (EVEN SEM)

DECLARATION

We certify that

1. The work contained in this Project Report is original and has been done by us under the guidance of my supervisor.
2. The work has not been submitted to any other University or Institute for the award of any other degree or diploma.
3. We have followed the guidelines provided by the University in preparing the Report.
4. We have confirmed to the norms and guidelines in the Ethical Code of Conduct of the University.
5. Whenever we used materials (data, theoretical analysis, figures, and texts) from other sources we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission form the copyright owners of the sources, whenever necessary.

Date:

Place:

Arpit Jain (1719231024)

Jatin Sharma (1719231024)

Aman Hasan Khan (1719231024)

Himanshu Rawat (1719231024)

Deptt. of Electronics and Communication Engineering
G. L. BAJAJ INSTITUTE OF TECHNOLOGY AND MANAGEMENT
[Approved by AICTE, Govt. of India & Affiliated to A.K.T.U (Formerly U.P.T.U), Lucknow]

CERTIFICATE

Certified that **Arpit Jain, Himanshu Rawat, Jatin Sharma, Aman Hasan Khan** have carried out the project work **(Project -II, REC-852)** presented in this report entitled **"MODEL AIDED ARTIFICIAL INTELLIGENCE FOR PATH LOSS PREDICTION IN MOBILE COMMUNICATION"** for the award of **Bachelor of Technology** in **Electronics and Communication Engineering** during the Academic session 2020-21 from **Dr. A.P.J. Abdul Kalam Technical University (Formerly U.P.T.U), Lucknow**. The project embodies result of the work and studies carried out by Student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

(Dr. Piyush Yadav)
(Project Coordinator)
(Associate Professor)
Deptt.of ECE

(Dr. Piyush Yadav)
(Project Coordinator)
(Associate Professor)
Deptt.of ECE

(Dr. Satyendra Sharma)
HOD, Deptt.of ECE

Date

ACKNOWLEDGEMENT

We would like to thank our Project Guide, Dr. Piyush Yadav, for his continuous support and encouragement. He has always provided motivation and direction to this project and constantly pushed us to work harder on our ideas. We are highly indebted for his guidance and constant supervision as well as for providing necessary information regarding the project.

We would like to also express our gratitude towards our parents and members of G. L. Bajaj Institute of Technology and Management for their co-operation and encouragement which helped us in the completion of this project. Our thanks and appreciations also go to our colleagues for support and encouragement and people who have willingly helped us out with their abilities. We have learned a lot during this time while developing our project from all the people involved directly or indirectly with our work.

ARPIT JAIN (1719231024)

HIMANSHU RAWAT (1719231047)

JATIN SHARMA (1719231048)

AMAN HASAN KHAN (1719231014)

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATION

RSS	Received Signal Strength
BS	Base Station
IOT	Internet of Things
W/C	Which
ML	Machine Learning
RD	Raw Data
ANN	Artificial Neural Network
SVM	Support Vector Machine
BPNN	Back Propagation Neural Network
SVR	Support Vector Regression
NCIL	Network Cell Info Lite
LTE	Long Term Evolution
CDMA	Code Division Multiple Access
WCDMA	Wideband Code Division Multiple Access
GSM	Global System for Mobile Communication
4G	4^{th} Generation
DMRC	Delhi Metro Rail Corporation

ABSTRACT

There square measure varied linear path loss model that are created for wireless communication, advanced model square measure won't to additional accurately and flexibly predict the trail loss for varied things. This project uses a machine learning, computing framework for predicting path loss at completely different railway system station in capital of India victimization three key models that square measure regression toward the mean, non-linear regression and call tree regression model and principal part analysis. Basically, the trail loss dataset may be a real time dataset that are collected at 3 completely different railway system stations, at completely different height contains of multiple options like distance, RSS (received signal strength), height etc.

The data we tend to measure in capital of India railway system stations square measure used. we tend to discovered that the projected call tree regression model and mix path loss is additional correct and versatile compare to the linear and non-linear regression models for path loss prediction.

For the improvement of wireless network, the trail predicting of path loss is of great importance. With the event of fourth generation mobile communication varied path loss prediction strategies with varied accuracy and complexness ought to be projected during this paper, the principal and procedure of machine learning models and AI based mostly path loss prediction is bestowed. The real time dataset we tend to measured, square measure won't to observe the performance of assorted models like regression toward the mean model, non-linear regression and call tree model. As we tend to discovered that these

machine learning based mostly models outperform the varied on the market models. seeable of the very fact that the amount of measured dataset determines the accuracy of machine learning models and algorithmic program, to forestall this as a reason for having the low accuracy in varied models and algorithms, we tend to create a true time dataset of getting around quite three hundred readings of various parameters of path loss like RSS, distance, height of antenna etc. whereas victimization this real time dataset, we tend to project 2 ways to expand the training dataset. On one hand, the previous measured information will be reused in new situations or at completely different RSS. On the opposite hand, the models will be used to get the quantity of coaching samples supported the previous info obtained from our measured results. Finally, some problems for future analysis square measure mentioned.

CHAPTER – 1
INTRODUCTION

The attenuation of Associate in Nursing radiation as it propagates over space is explained by path loss. For link budgeting, system performance optimization, coverage prediction and base station (BS) location selection, a correct, simple, and general trail loss model is critical.

As a result, academics and researchers have made significant efforts to discover low-cost models for trail loss prediction in a variety of scenarios and at various frequencies. Several measurement campaigns are being carried out around the world to obtain information that will be helpful in creation, adjustment, and value these models. The fourth-Gen wireless n/w are designed to provide enhanced spectral potency, increased output, broad coverage, enhanced association density, and reduced radio latency. The support of IOT applications can include vast coverage regions and a variety of terrains.

Initially, path loss prediction models were built using empirical or well-established methodologies. Empirical models primarily allow measuring a very particular frequency range and condition of affairs. They provide mathematical representations of the relation between route propagation and loss characteristics like as frequency, antenna separation distance, and antenna height, among others. The regression toward the mean model, for example, the trail loss exponent, which is determined from empirical observation, used to describe how the received power decreases as the antenna separation distance increases.

Empirical models are simple to understand because only some of the parameter are essential and the model equations are short. The parameters of empirical models, on the other hand, are excluded from given knowledge in a very specific situation. As a result, when these models are applied to more generic situations, their accuracy is similarly unsatisfactory. Empirical models, on the other hand, will only describe the stats of trail loss at a given distance, but they will not be able to provide the specific received power at a given position. The main objective of this type of supervised learning with labelled information is to be informed a generic and proper operation between ins and outs, w/c makes it more suitable for detecting specified and regression errors.

For modelling process physics, deterministic models use radio-wave propagation methods and numerical analytic approaches, such as those supported by ray tracing and finite difference time domain. They will gain great accuracy & give the path loss value at a given location. On the other hand, include a lack of process potency and, as a result, preventative computation time in real-world contexts.

It is necessary to have site-specific pure mathematics information as well as material qualities. Furthermore, once the surrounding for propagation has been altered, we must repeat the extensive computing method. Machine learning could be a strategy for making predictions that is backed by a huge test or training dataset and model architecture. Machine-learning-based approach has recently used in auto-drive vehicles and data processing, laptop vision, voice recognition, and a variety of different domains. These types of exercises will be divided into two categories: supervised and unattended learning. The main objective of this type of supervised learning with labelled information is to be informed a generic and proper operation between ins and outs, w/c makes it more suitable for detecting specified and regression errors.

Unattended learning algorithms, on the other hand, should characterize the hidden pattern from unknown knowledge. In virtue of predicting path loss is supervised regression problem, which may be handled using supervised machine learning methods such as regression toward the mean, non-linear regression, and call trees. It's been suggested that instead of empirical models we can use machine-learning based models because they are more enhanced than established methods. When the problem did occur – it was usually in case of cross industry data collection project since consistent denominator was difficult to establish. Otherwise, even in the projects where multiple data collection sources were required, the denominators rarely varied. As a result of this simplicity of this illustration, linear regression is a excellent model.

The example may be an eqn that combines the set of input values x; the response is that the expected output for that set of input values y. As a result, each of the input values (x) and the output value are both numeric. Every input value or column is assigned one multiplier, referred to as a constant and represented by the Greek letter Beta in the equation (B). A constant is value-added to the route, giving it an extra degree of freedom (for example, going up and down on a 2-D map), and is known as the intercept or bias constant. Non-linear regression model is a type of model in which data is fitted to a model that is given as a mathematical relationship. Nonlinear model links the variables (X and Y) in a highly nonlinear (curved) connection, whereas plain regression toward the mean relates two variables (X and Y) where a line ($y = Mx + b$). The purpose of model is to decrease the concentration of squares as much as possible. The sum of squares can be a live that measures how far the observations differ from the nonlinear function that is used to predict the line.

This paper's significant contributions are summarized as follows.

(1) The basic building block and procedure for predicting path loss and support M.L is conferred. Few important problems like knowledge preprocessing, knowledge assortment, algorithmic rule choice model, performance analysis and hyperparameter settings square measure mentioned.

(2) To acquire enough knowledge for machine-learning-based models, 2 ways square measure thought-about to enlarge the coaching knowledge set by absolutely victimization the present values and therefore the classical models. knowledge given is taken into account in each the frequency domain and state of affairs domain.

(3) Different ML algorithms square measure used to validate the projected ways supported the measured knowledge. each outside and indoor situations square measure taken at totally different height levels, distance square measure taken under consideration and therefore the measured knowledge is employed to verify the practicableness of the ML-based prediction models.

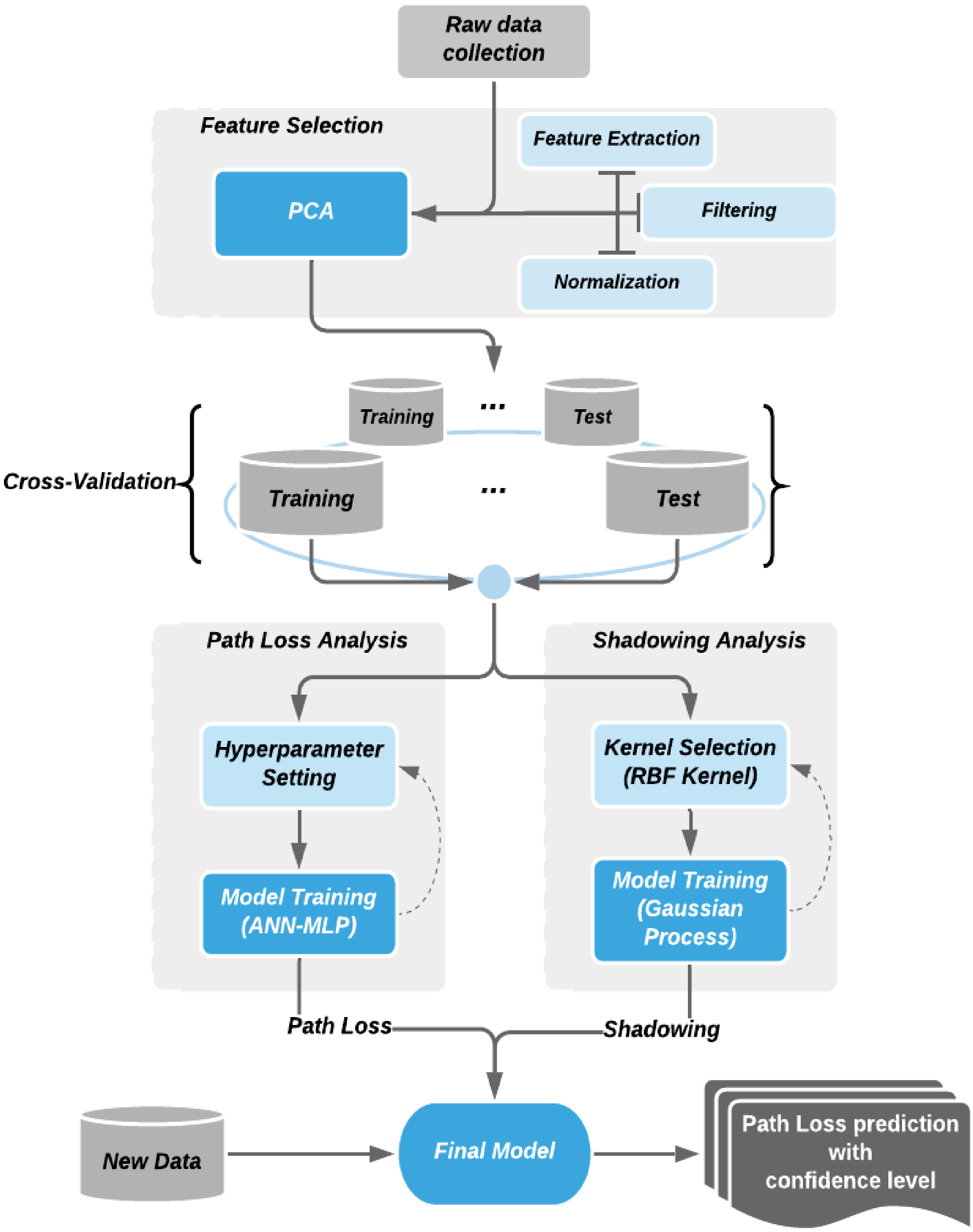

Fig 1.1: Path Loss analysis based on ML

1.1 PATH LOSS ANALYSIS BASED ON ML

1.1.1 Raw Data Collection

The collecting of raw data (also known as primary data) is the beginning point for any data analysis. Once the RD (Raw Data) has been acquired, it is processed into Information, which can then be transformed into Knowledge later in the analysis process. The goal of this White Paper is to describe the fundamental RD collecting concepts and problems, as well as how these principles and issues relate to each other.

• Determining what information/data needs to be collected

• Establishing a collection timeline

• Determining a collection method

• Collecting the Data

• Sorting the Data

For many years, Steps 1–4 were thought to be simple to complete. There were standard collecting processes for each of the data kinds that could be implemented without much difficulty. Many sectors had established industry-wide data gathering and presentation standards. Banking and finance, travel, resource sectors, and international shipping and trade are some of the businesses where data collecting has been standardised to a large extent. Data scientists may create so-called "templates," which they could then use as needed. In the rare case where no template appeared to be suitable for the purpose of data gathering due to project-specific constraints, new templates could usually be created by re-engineering existing templates. It was rarely necessary to create templates from scratch. When the data sources were standardized, quantitative data gathering was a breeze.

Based on unifying aspects of these sources, the data collecting and processing technologies were easy to link to the data sources. Finding common denominators across data sources/sets was necessary for identifying such traits. When problems did arise, it was mainly in cross-industry data collection efforts due to the difficulty in establishing consistent denominators. The denominators rarely varied in other projects, even when various data gathering sources were required.

1.1.2 Feature Selection

Feature selection, also known as variable selection, attribute selection, or variable subset selection in machine learning and statistics, is the process of choosing a subset of relevant features (variables, predictors) for use in model creation. Techniques for feature selection are employed for a variety of reasons: To avoid the curse of dimensionality, improve data compatibility with a learning model class, and encode inherent symmetries present in the input space, simplify models to make them easier to interpret by researchers/users, shorten training times, and encode inherent symmetries present in the input space. When utilising a feature selection technique, the key concept is that the data contains some features that are either redundant or irrelevant, and hence may be deleted without causing significant information loss. Because one relevant feature may be redundant in the presence of another relevant trait with which it is closely associated, the terms redundant and irrelevant are used interchangeably.

It's important to distinguish feature selection strategies from feature extraction techniques. Feature extraction generates new features based on the functions of the original features, whereas feature selection only returns a subset of the features. Feature selection approaches are frequently utilized in domains with a large number of characteristics and a small number of samples (or data points). The analysis of written texts and DNA microarray data, both of which have thousands of features and a few tens to hundreds of samples, are classic examples of feature selection in action.

1.1.3 Cross Validation

Cross-validation is a technique in which we train our model using a subset of the data set and then evaluate it using the other subset.

The following are the three steps involved in cross-validation:

1. Set aside a piece of the sample data set.
2. Train the model with the remaining data set.
3. Use the data set's reserve component to test the model.
4. Cross Validation Techniques

Validation: In this strategy, we train on half of the data set and use the remaining half for testing. The biggest disadvantage of this strategy is that we only train on half of the dataset; it's likely that the remaining 50% of the data contains critical information that we're overlooking while training our model, resulting in larger bias.

LOOCV (Leave One Out Cross Validation): In this strategy, we train on the entire data set but only use one data point from the available data set, iterating for each data point. It has several benefits as well as some drawbacks. This strategy has the advantage of making use of all data points, resulting in reduced bias. Because we are testing against a single data point, this strategy has the primary disadvantage of causing more variation in the testing model. If a data point is an outlier, the variation will be higher. Another disadvantage is that it takes a long time to execute because it iterates over the number of data points.

1.1.4 Path Loss Analysis

The decline in power density of an electromagnetic wave as it propagates over space is known as path loss or path attenuation. Path loss is an important factor to consider while analyzing and designing a telecommunication system's link budget. We plotted several readings in graphical format as we altered path loss with regard to distance in our procedural attempt to assess path loss and the models we incorporated to execute and materialize.

1.1.5 Shadowing Analysis

Shadowing is a small-scale qualitative research approach in which the researcher acts as an observer. Researchers shadow a research subject or participant in real-life events for a predetermined amount of time. The researcher does not interfere with the participant for this time period in order to prevent the research subject from deviating from their natural behaviour in the scenario or condition. It might be aggravating for researchers because they often need to dig deeper into what they learn during shadowing exercises. But, at the same time, shadowing can help validate the participant’s journey. Shadowing also gives a quick insight into the design context for an audience, quickly and accurately.

Depending on what the design researcher wants to learn from the practice, shadowing can last anywhere from 30 minutes to several weeks or months. Shadowing is a behavioral observation technique that allows users to be watched in their natural context. Researchers should not, however, make inferences based on the results of a few isolated observations.

CHAPTER 2

LITERATURE SURVEY

2.1 PATH LOSS

Yan Zhang, Jinxiao Wen, Guanshu Yang, Zunwen He and Jing Wang in their research paper "**Path Loss Prediction Based on Machine Learning: Principle, Method, and Data Expansion**" published in April 2019, they have researched about the path loss and how to predict it with the help of Machine Learning and they have researched about the path loss and said that path loss is a change in the power of a radio wave as it travels through the house between the transmitter and receiver [1]. Prediction of path loss is significant in wireless communication n/w design and development, such as link budget, coverage analysis, and locating base station, because receivers require a specified minimum power to copy data correctly. Several existing route loss models uses a regression model, w/c is produced by trial and error by forwarding a straight proportion between the distance and the path loss, and by comparing the graphical representation of data the issue has been countered [1].

Although regression models are very simple and straightforward, it does not guarantee accurate path loss prediction in all radio propagation settings. Enhanced modelling techniques were used for more correctly and adaptability simulate trail loss in a difficult and widely spread ecosystems [2]. Path loss is used to give the attenuation of nonparticulate radiation as it travels through a house. For link budgeting, system performance enhancement, coverage prediction and an associate degree correct, simple and general model for trail loss is required.

As a result, academics and engineers have worked hard to develop cost-effective algorithms for trail loss prediction in a variety of situations and frequencies [3]. Several activity programs are being held around the world to collect data that will be used to create, adjust, and evaluate these models.

Faris A. Almalki, Marios C. Angelides in their research paper "**A machine learning approach to evolving an optimal propagation model for last mile connectivity using**

low altitude platforms" published in 2019 As propagation models anticipate signal attenuation or path loss as a measure of the power density of an electromagnetic wave as it propagates over space from a transmitter, they explain path loss in their study. Path loss can be used to track system performance, network planning, and coverage in order to achieve optimal reception. When a signal is propagating to a maximum distance, many elements can influence it, including topography, frequency, and transmitter and reception antenna heights [5]. Radio signals flow in free space until they reach the complex ground pervasive environment, where shadowing, scattering, and other effects occur naturally, according to aerial platform propagation route loss models.. Thus, it is essential to identify the different type of environments that have been categorized by International Telecommunication Union (ITU) namely: Urban, Suburban, and Rural [4,6].

2.2 MACHINE LEARNING BASED PATH LOSS PREDICTION

Yan Zhang, Jinxiao Wen, Guanshu Yang, Zunwen He and Jing Wang in their research paper "**Path Loss Prediction Based on Machine Learning: Principle, Method, and Data Expansion**" published in April 2019, they have researched about the path loss and how to predict it with the help of Machine Learning and they have researched about the path loss and describe the basic principle of path loss predictors based on ML. We can use machine learning approaches to build a decent estimation function for path loss prediction once we know the output (path loss observation) and the relevant input features such as antenna-separation distance and frequency. This function maps input features to path loss values, and it can be either a white box (in decision-tree-based models) or a black box (in non-decision-tree-based models) (within SVR-based or ANN-based models). The approach for path loss predictors based on machine learning [1].

The collected data refers to measurement samples, with each sample containing the path loss value as well as the accompanying input attributes. System-dependent parameters and environment-dependent parameters are the two groups of input features. System-dependent parameters are those that are not affected by the propagation environment, such as carrier frequency, transmitter and receiver heights and positions, and so on. More system-dependent properties, such as the antenna separation distance and the angle between the line-of-sight path and the horizontal plane, can be obtained using the above parameters [1].

The factors that are determined by the geographic environment and meteorological conditions are known as environment-dependent parameters. Terrain, building conditions, and vegetation conditions are all factors that affect the geographic environment. The majority of them are available through three-dimensional (3D) digital maps, topographic databases, and land cover databases. Temperature, humidity, precipitation rate, and other weather characteristics are included [1].

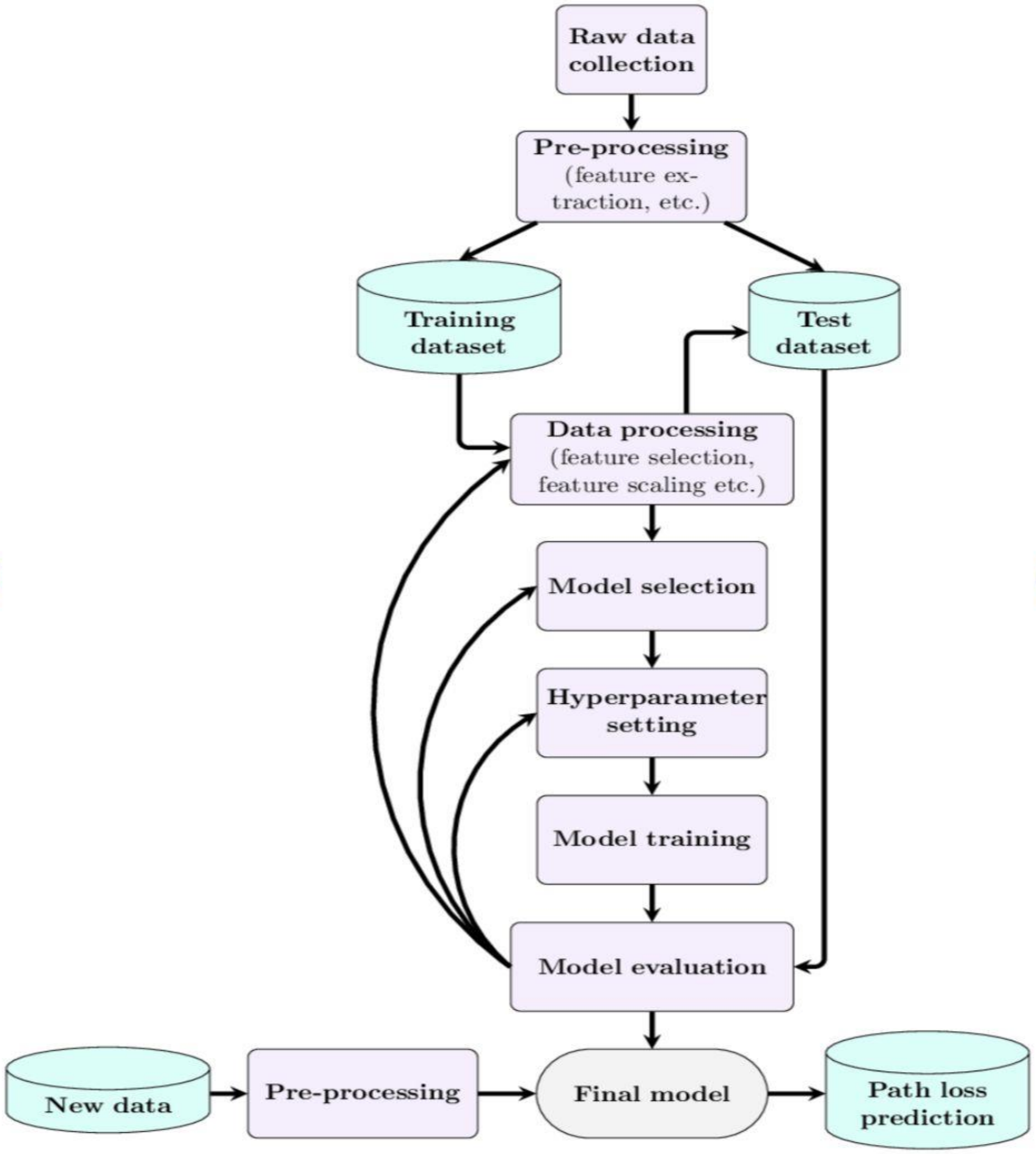

Figure 2.1: Procedure of machine-learning-based path loss prediction [1].

2.3 MODEL TRAINING

Aymen Ben Zineb, Mohamed Ayadi in their research paper "**A Multi-wall and Multi-frequency Indoor Path Loss Prediction Model Using Artificial Neural Networks**" published in 2015 talked about how they can train new models for predicting path loss inside the buildings or indoor. They claimed that the most significant and critical task in the modelling challenge is training. In fact, a well-trained model must be able to anticipate the expected output with high accuracy utilising interpolation or extrapolation based on prior knowledge acquired during learning from a new input. There are several proposed methods for neural network learning in the literature, which can be divided into supervised and unsupervised learning [7]. For data clustering, an unsupervised learning approach is utilized, which involves dividing data into separate groupings depending on particular properties. The input parameters and output values are known in supervised learning, such as the gradient decent approach, and the neural network can provide an inferred function that can be used to map new samples. This can be accomplished by modifying the bias and weight of each neuron in order to reduce the mean squared error (MSE) between the neural model's input and desired output [7].

The maximum number of influent parameters must be considered when developing an accurate neural network model. The inputs for the model that we created and presented in this research come from the multi-wall model. The transmitter–receiver distance (d), frequency (f), wall attenuation (L w), and floor attenuation (L w) are the components (L f). There is only one hidden layer in the model. The number of neurons in this hidden layer is set to 75% of the number of neurons in the input layer [7]. This number can be changed, and the consequences of doing so will be described in the following section. Finally, there is only one output in the output layer, which represents the measured signal route loss.

CHAPTER-3

PATH LOSS MODELS

3.1 LINEAR REGRESSION

Linear regression was the first type of regression analysis to be thoroughly explored and widely employed in real-world applications. [8] This is because models with linearly related unknown parameters are easier to fit and the statistical features of the resulting estimators are easier to determine than models with non-linearly related unknown parameters.

Because the illustration is so straightforward, linear regression is a great model. The answer is that the predicted output for that collection of input values is an equation that combines a specified set of I/p values x. (y). As a result, both the input (x) and output (y) values are numeric. Every input value or column is given one multiplier factor, which is referred to as a constant and is represented in the equation by the capital Greek letter Beta (B). The path has an extra degree of freedom (going up and down on a two-dimensional map) thanks to another constant known as the intercept or bias constant [22].

The road is a plane or hyper-plane in higher dimensions when there is only one input x. As a result, the varieties of the equation and, as a result, specific values being utilized for coefficients are shown. The complexity of a regression model, such as regression toward the mean, is frequently discussed. The no. of coefficients we have used in the model is referred to as this. When a constant becomes zero, it effectively removes the input variable's influence on the model and thus the model's prediction (0 * x = 0). This is essential when looking at regularization strategies that change the training procedure to reduce the complexity of regression models by applying golf stroke pressure to all of the coefficients' sizes, driving some to zero.

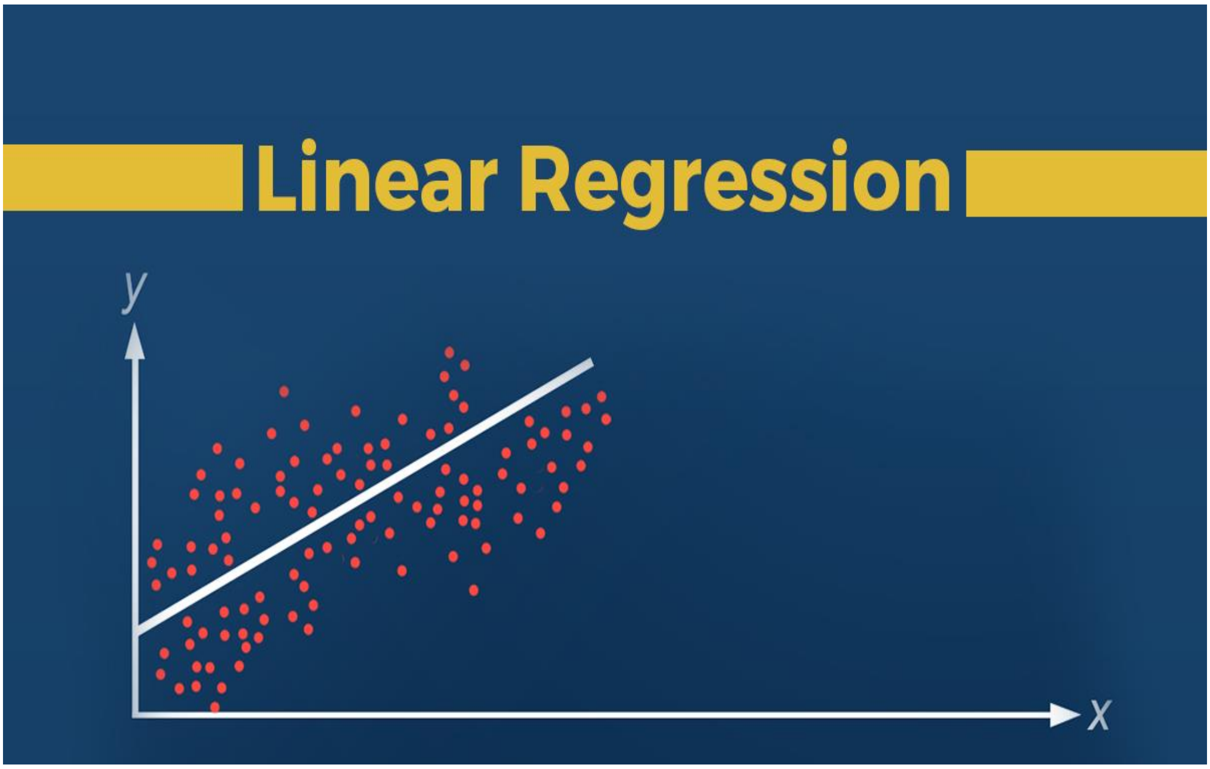

Fig 3.1: Linear Regression [21].

Linear regression has been extensively explored, and there is a wealth of literature on how to format your data to get the most out of the model. As a result, when discussing these criteria and objectives, there is a level of expertise that might be daunting. When utilizing Ordinary Least Squares Regression, the most common implementation of linear regression [2], you can use these rules more as guidelines.

Assumption of linearity. The relationship between your input and output is assumed to be linear in linear regression. It is incapable of supporting anything else. This may seem self-evident, but it's important to remember when you have a lot of them. To make the relationship linear, you may need to change the data (e.g., log transform for an exponential relationship) [2].

Remove the background noise. Your input and output variables are assumed to be noise-free in linear regression. Consider employing data cleaning processes to better expose and clarify your data's signal. This is especially crucial for the output variable, and you should try to eliminate outliers in the output variable (y) [21].
Collinearity must be eliminated. When your input variables are highly linked, linear regression will over-fit your results. Calculate pairwise correlations for your input data and exclude the ones that are the most correlated.

Gaussian Distributions are a type of distribution. If your input and output variables have a Gaussian distribution, linear regression will produce more accurate predictions. You could gain some advantage from applying transformations to your variables (e.g., log or BoxCox) to make their distribution seem more Gaussian.

3.2 NON-LINEAR REGRESSION

The Non-linear regression model is a type of multivariate analysis in which the data is used to create a model and then it is expressed as a mathematical relationship. Nonlinear regression model connects 2 variables (X and Y) with a curved line i.e. y = max + b, whereas straightforward rectilinear regression model connects them with straight line (y = max + b). The model's purpose is to reduce the total no. of squares as much as possible. The sum of squares is used to measure that how far the Y observation differ from the nonlinear function that is used to predict the Y variable. It is calculated by 1st determining the diff. between the fitted non-linear operation and each Y information purpose within the set. Each of these permutations is thus square. Finally, all of the square figures are offset from one another [21].

The higher the operate matches the data points within the set, the smaller the total of those square numbers. Exponent functions, pure mathematics functions, exponential functions, power functions, Konrad Zacharias Lorenz's curves, mathematician function, and different fitting method are all used in nonlinear regression. Non-linear regression modelling is a lot like the rectilinear regression modelling in which both ask for a diagrammatic representation of a given response from a particular set of factors. Nonlinear models are very difficult to create than the linear models since work is done through a succession of approx. that are derived from trial and error [21]. Many already devised methods, such as the Gauss Newton technique and the Levenberg Marquardt's approach, are used by mathematicians. Regression models that look nonlinear at first glance are most of the times linear.

Curve estimate process is frequently used to determine nature of meaningful relations at play in our data, allowing us to select an appropriate regression model be it linear or nonlinear. Rectilinear models can type curves in addition to lines, which depends upon the geometry of the rectilinear regression equations. Similarly, a nonlinear equation can be reworked using pure mathematics to mimic any linear equation; such a nonlinear equation is called "intrinsically linear. Nonlinear regression model is frequently used to forecast increase over time, for example [21]. The scatter-plot of data of the population over the time reveals that there is a relationship b/w time and increment, but that it's nonlinear, necessitating use of a non-linear regression. A model of logistical increment will help us to provide population estimates for time that was not measured as well as increment predictions in the future.

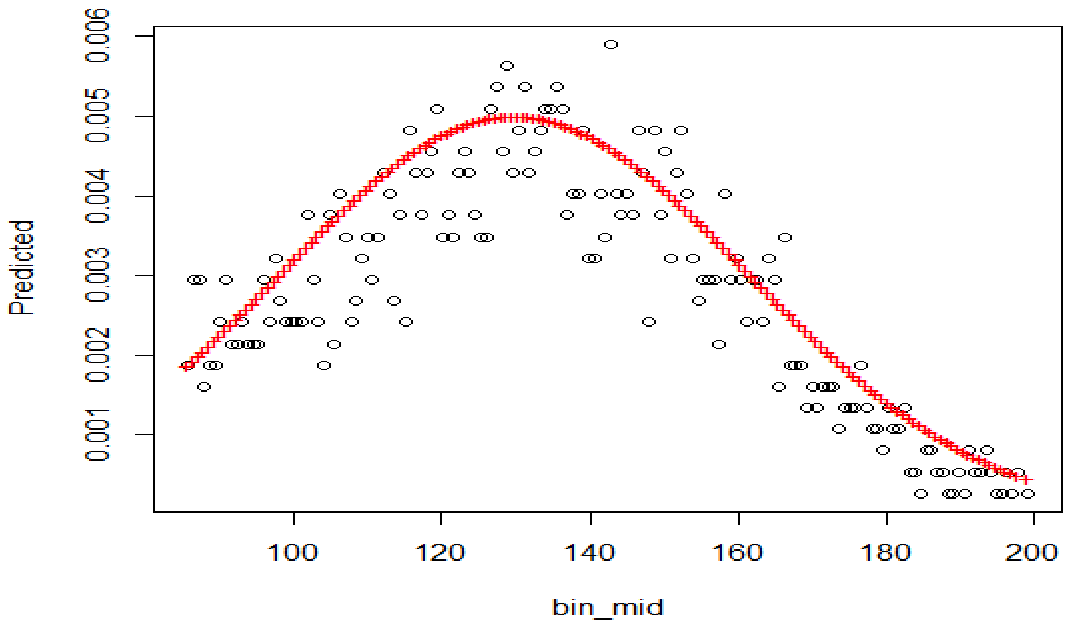

Fig 3.2: Non-Linear Regression

Even though it is less flexible than a linear regression model, a nonlinear regression model is employed to handle various mean functions. Predictability, parsimony, and interpretability are only a few of the benefits. A nonlinear regression may be used in a variety of ways, including financial forecasting [22].

A scatterplot of shifting financial prices over time reveals a link between price fluctuations and the passage of time. Because the connection is nonlinear, the appropriate model to utilise is a nonlinear regression model.

A logistic price change model can give estimates of unmeasured market prices as well as projections of future market price changes [22]. Depending on the condition of the economy, the bulk of financial and macroeconomic time series display distinct characteristics throughout time.

For example, recessions vs expansions, bull and bear stock markets, and low versus high volatility are some of the dual regimes in economic time series data that necessitate nonlinear models. Regime-switching, smooth, and threshold models are examples of nonlinear time series that adopt dual regimes and are referred to as state-dependent models.

A nonlinear regression can only produce reliable results if the connection between the dependent and independent variables is accurately specified and described. Furthermore, because bad starting values might lead to a non-convergent model, appropriate starting values are required [22]. Nonlinear regression frequently uses a quantitative dependent.

3.3 DECISION TREE

The purpose of this model is to learn how to use models in data processing, to create and predict the path loss at different metro stations based on the large number of input variables. A simple instance for classifying examples may be a decision tree. Assuming that each of the readings has a finite number of domains, and that there is only one target feature called "classification" in this section. Every part of the classification domain is referred to as a category. A call tree, also known as a classification tree, is a tree like structure where each inside node has an input feature of its own [19].

The arcs that return from a node labelled with Associate in Nursing input feature are labelled with each of the target's possible values, or the end results in a branched call node on a distinct feature [21]. Every node/leaf of the tree is labelled with a category or a probability distribution over the categories, indicating that the dataset created by us has comes under the specific class or a probability distribution. A tree is built by cacophonic the supply set, which serves as the tree's foundation node/leaf, into different subsets, which makes the tree's successor youngsters. The cacophonous is made of a set of cacophonous rules that support categorization options [19]. This method is applied to each derived set in an algorithmic manner known as algorithmic partitioning. The formula is finished when all constant values of the target variable are present in the set at a node, or when cacophonous does not add anything to the predictions.

This method is based on top-down approach of call trees is an example of a Greek formula, and it's the most efficient way of learning call trees from knowledge. In data processing, call trees can be defined as a combined method of mathematical and machine techniques that helps in the definition, categorization, and generalisation of data. All different paths from the foundation node to the leaf or branched node in a call tree take the approach of conjunction, or AND. It's possible to use ORs to join two additional ways along using minimum message length (MML) in an exceedingly call graph [20]. Call graphs are any extended to permit previously implicit and new attributes to be learned dynamically and used at various places throughout the graph. The additional general cryptography theme leads to higher prognostic accuracy and log-loss probability [21].

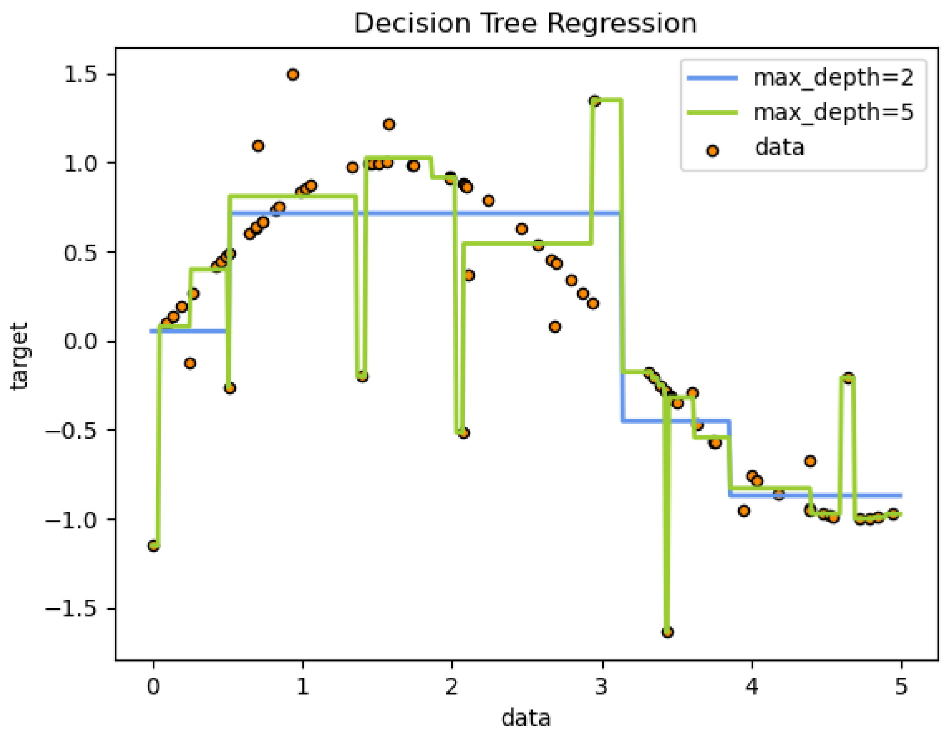

Fig 3.3: Decision Tree

A decision tree generates an estimate by asking the data a series of questions, each of which narrows the range of potential answers until the model is confident enough to offer a single forecast. The model determines the sequence of the questions as well as their substance. Furthermore, all of the questions are True/False in nature [20].

This is a bit difficult to comprehend since it is not how humans think naturally, and the simplest way to demonstrate this difference is to construct a genuine decision tree from scratch. x1 and x2 are two characteristics in the preceding issue that allow us to generate predictions for the target variable y by asking True/False questions.

There are different branches for each True and False response. We get to a forecast regardless of the answers to the questions (leaf node). Begin at the top, at the root node, and work your way down the tree, answering the questions as you go. As a result, any pair of X1 and X2 can be used.

I should explain one element of the decision tree: how it learns (how the "questions" are created and how the thresholds are set). In the training phase of model construction, a decision tree learns to map data to outputs as a supervised machine learning model [19].

During training, all past data relevant to the issue domain and the real value we want the model to learn to predict is fed into the model. Any associations between the data and the target variable are learned by the model.

Following the training phase, the decision tree generates a tree similar to the one above, determining the best questions to ask as well as the sequence in which they should be asked in order to create the most accurate estimations possible. When we want to create a forecast, we should provide the model the same data format in order to produce a prediction. The forecast will be a guess based on the train data on which it was trained.

3.4 ARTIFICIAL NEURAL NETWORK

When the sample size is big enough, ANN may be used to handle nonlinear regression problems and has low prediction errors, making it a popular approach for path loss prediction [9,10,11,12].

Interconnections between neurons produce ANNs, which are networks. The feed-forward ANN of multi-layer perceptron structure generally comprises an input layer, one or more hidden layers, and an output layer based on the neuron model [11]. Different weights connect neurons in the next layer, but there is no connection between neurons in the same layer, and there is no cross-layer connection.

The network size is determined by the number of hidden layers and neurons, and has a significant influence on model complexity and accuracy. Unfortunately, finding an appropriate ANN structure for route loss prediction remains a challenge. [9] shows that for a typical rural macro cell radio network planning scenario, a non-complex ANN, such as a feed-forward ANN with one hidden layer and only a few neurons, will probably give acceptable route loss prediction accuracy. When compared to non-complex architectures, ANNs with several hidden layers and numerous neurons may have poor generalisation characteristics [9,11]. This is most likely due to overtraining, in which the model performs well on data that is comparable to the training dataset but is not flexible enough to adapt effectively to data that is not similar to the training data.

The back propagation algorithm is a simple approach for training artificial neural networks. BPNN is a term used to describe this sort of network. This paper's study is based on a three-layer BPNN structure with complete connectivity between levels. Given a set of training samples (x1, y1), (x2, y2), etc., (xN, yN), where xi = x1 I x2 I etc., xL I R L is a feature vector and yi R 1 is the goal output, measured value of route loss. In the forward propagation phase, the predicted value of path loss yi 0 can be expressed as

$$y_i' = f_o\left(\omega_{om}\left(f_m\left(\omega_{ml}\mathbf{x}_i\right) + \theta_m\right)\right) + \theta_o$$

where ml denotes the connection weights between the hidden layer's neurons and inputs, om denotes the connection weights between the output layer's neurons and the hidden layer, and m and o denote the hidden layer's and output layer's thresholds, respectively.

The transfer functions fm () and fo () are for the neurons in the hidden layer and the neuron in the output layer, respectively.

The error originating at the output neuron propagates backward. The learning phase of the network proceeds by adaptively adjusting the weights based on the loss function, which is expressed as

$$E = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2$$

The mean squared error is denoted by E. The gradient descent technique is used in the back propagation algorithm. Slow convergence speed and local poles are two disadvantages of standard gradient descent. Other training techniques, such as the Levenberg-Marquardt method, the Fletcher-Reeves update method, and the Powell-Beale restart method, may also be considered. Because of its high convergence speed at the price of memory consumption, the Levenberg-Marquardt method is widely employed for route loss prediction [9,11].

3.5 SUPPORT VECTOR MACHINE

Support vector machine (SVM) is a statistical learning theory-based machine learning technique. SVM's core principle is to nonlinearly transfer a set of data in a finite-dimensional space to a high-dimensional space such that the dataset may be separated linearly. SVR is a regression problem-solving extension of SVM, therefore it may be utilised for path loss prediction [13].

The basic principle behind SVR is that the sample points should fall on a hyperplane that is too fine in the high-dimensional feature space. The following linear function can be used to describe the hyperplane in feature space.

$$f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b \qquad (1)$$

where w is the normal vector which determines the direction of the hyperplane, x is an input feature vector, φ (·) is the nonlinear mapping function, and b is the displacement item

The solution to the optimal hyperplane is a constrained optimization problem, which can be written as [14].

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i{}^*)$$
$$\text{s.t. } f(\mathbf{x_i}) - y_i \leq \varepsilon + \xi_i$$
$$y_i - f(\mathbf{x_i}) \leq \varepsilon + \xi_i{}^*$$
$$\xi_i, \xi_i{}^* \geq 0, i = 1, \ldots, N \quad (2)$$

where C is the regularization coefficient, I and I are the slack variables that allow the insensitivity range on both sides of the hyperplane to be slightly different, and is the insensitive loss, which means the predicted value can be considered accurate if the deviation between the predicted value and the actual value is less than.

Then, by introducing Lagrange multipliers and solving its dual problem, the approximate function can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{N}(-\alpha_i + \alpha_i{}^*)K(\mathbf{x}_i, \mathbf{x}) + b \quad (3)$$

K is a kernel function that is used to execute the nonlinear mapping from the low-dimensional space to the high-dimensional space, where I and I are Lagrange multipliers. The kernel function used is crucial to the SVR-based predictor's performance. The linear kernel, polynomial kernel, Gaussian radial basis function, sigmoid kernel, and their combinations are the most popular kernel functions today. The kernel function in this paper is a Gaussian kernel with a tunable parameter, which is specified by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \quad (4)$$

The Gaussian kernel [17–20] is a popular kernel function that is well suited to applications with modest feature dimensions and no prior information. In this work, the regularization coefficient, insensitive loss, and kernel function parameter were all explored using the same technique as in [13]. The linear kernel, polynomial kernel, Gaussian radial basis function, sigmoid kernel.

CHAPTER-4

EXPERIMENTAL SETUP

4.1 SPECTRUM ANALYZER

A spectrum analyzer plots the graph between amplitude of an input signal vs frequency range. The most typical use of this device is to predict the strength of coming or given signals irrespective of their nature. Most spectrum analyzers look at electrical signals as input, but they can also assess the spectral components of different signals including acoustic pressure waves. Other types of spectrum analyzers exist, such as optical spectrum analyzers that use direct optical techniques like a monochromator to make measurements [16].

Studying the levels of different electrical signals, which are not immediately visible in time domain waveforms, can reveal the power, distortion, dominant frequency, bandwidth, harmonics, and other spectrum of a signal [15]. Electronic equipment, such as wireless transmitters, can be described using these metrics.

By studying the spectra of electrical signals, you can see the majority frequency, harmonics, power, bandwidth, distortion and other spectrum components of a signal that aren't visible in time domain waveforms. These parameters aid in the classification of electrical devices such as wireless transmitters.

A traditional spectrum analyzer searches for signals within a spectral bandwidth and provides snapshots of the signal in the frequency or modulation domain. However, this is often not enough information to confidently describe the dynamic nature of modern RF signals [16]. The electronic instrument, used for analyzing waves in frequency domain is called spectrum analyzer. Basically, it displays the energy distribution of a signal on its CRT screen.

We used the mobile application Network Cell info lite as the medium which assisted us in accumulating the data, we required to execute the models we thought can be useful for us and eventually used a Spectrum analyzer to validate these readings as to cross check the accuracy of this dataset. After validating the dataset with the help of a spectrum analyzer we were able to confirm the accuracy of this dataset which eventually led us to predict the inconsistency in the models we proposed.

Fig 4.1: Spectrum Analyzer

4.2 NETWORK CELL INFO LITE

Network Cell Signal & Wi-Fi Info is a comprehensive cell network/Wi-Fi monitor and measurements/diagnostic history record tool that supports 4G+, LTE, CDMA, WCDMA, and GSM networks. Network Cell Info will help you fix your reception and connectivity issues while also keeping you informed about the radio frequency landscape in your area. A one-tap Wi-Fi/mobile internet performance speed test tool is also included in Network Cell Signal & Wi-Fi Info. To check your internet performance, run a speed test that includes download, upload, ping, and jitter test results.

Main Features of this app are as follows:

- Monitoring of cellular carrier and Wi-Fi signals in almost real-time (1 second) in Gauge/Raw Tabs (*) GSM, CDMA, UMTS (WCDMA), IWLAN, LTE, LTE+ support

- Wi-Fi/mobile internet performance speed test in a single touch (download, upload, ping, and jitter)

- Support for dual-SIM cards (*)

- 2/3 Signal-meter gauges (*) for both SIM cards and Wi-Fi

- Signal Plots (maximum of 2 cells)

- (*) Band number

- Other from the Gauge tab, there is a SIM# preference option.

- Map with cellular network information and signal metre gauges

- Measurements of cellular signals, history logs (in the Map tab)

- Cell locations (not carrier cell towers) on the Map, courtesy of Mozilla Location Service (MLS), except CDMA (*)

- The crowd-sourced Best Signal Finder displays your carrier's closest best signals.

- route colouring (on the map) based on signal intensity and map markers with location and signal information (minimum distance, minimum accuracy, motion sensor, etc.).

- Measurements of database export history in KML 2.2, MLS Geosubmit v.2, CLF v.3, Open Cell ID CSV, and CMWF

.

- database types in the Status Bar, you may see network information.
  Connection statistics (2G/3G/4G) SIM and device information Raw view of carrier network cellular information.

Information about the app is in the figure given below:

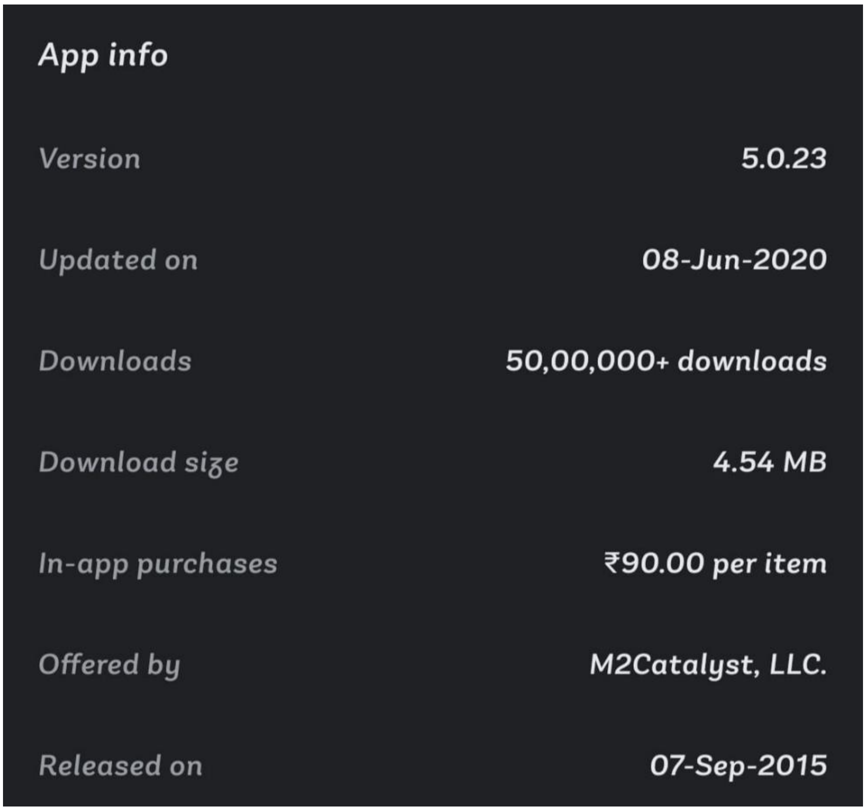

Fig 4.2: Application Information of NCIL

Signal power: This is the downlink signal power the phone is receiving from the nearby cellular tower.

Band Number: This is the frequency your phone is currently using. Band numbers are actually just frequencies and can be summarized as.

Band 20 (B20) = 800MHz
Band 8 (B8) = 900MHz

Band 3 (B3) = 1800MHz

Band 1 (B1) = 2100MHz

Band 7 (B7) = 2600MHz

Service: There are 3 services primarily at the moment: GSM, 3G and 4G.

If there is no symbol showing on your phone, then you are connected to GSM

If there is a H, H+ or 3G symbol showing, then you are on 3G

If there is a 4G or LTE symbol showing, then you are on 4G

If there is a 5G symbol showing, then you are on 5G

CID or TAC

The CID (in the case of a 3G connection) and TAC (in the case of a 4G connection), are the ID’s associated with the particular cell tower you are currently connected to.

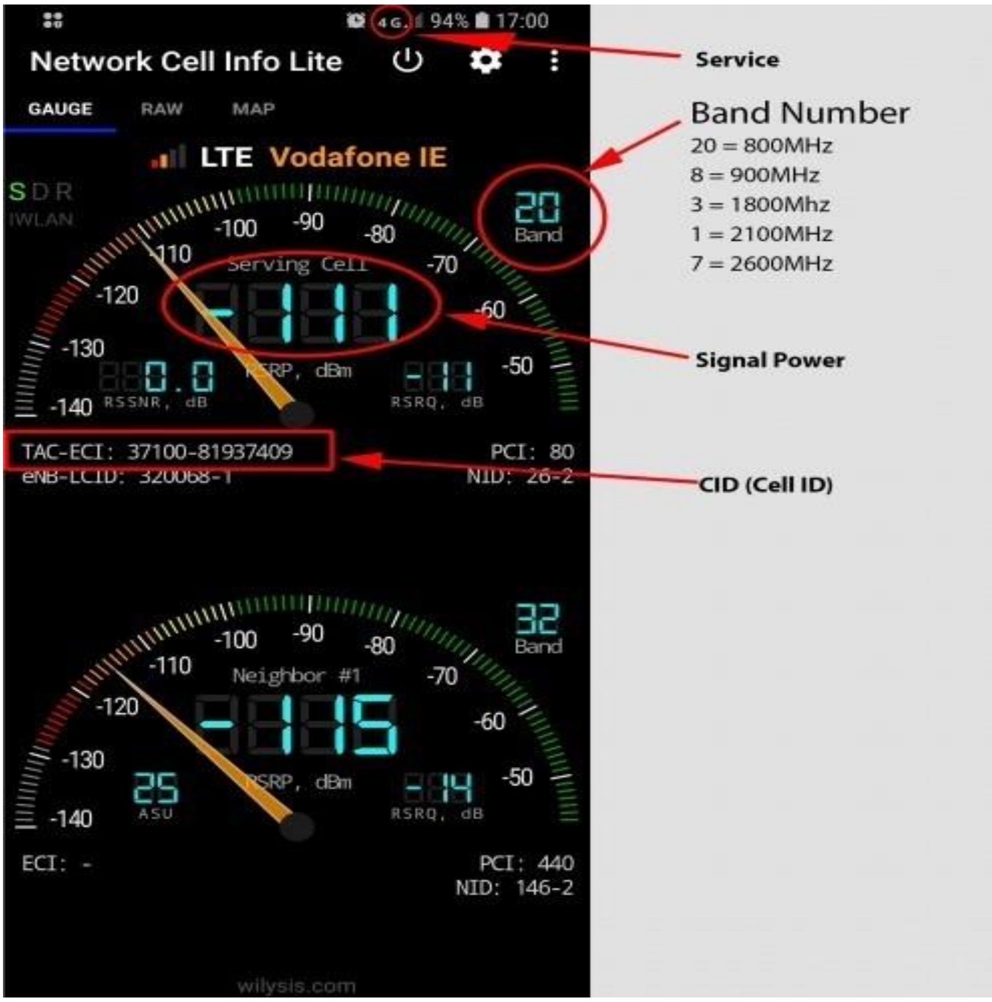

Fig 4.3: Screen Shot of GUI of NCIL

4G calling: 4G calling, or voice over LTE (VOLTE), means you can now make calls over 4G. Before, all calls had to use 3G (or GSM). A phone normally “rests” on 4G if it exists in your area. Every time you look at your phone, you will see the 4G symbol. When you use data, you will still see the 4G symbol. However, your phone may switch to 3G when a call is initiated or received, especially if the 4G signal is weak, and/or you do not have 4G calling enabled on your phone.

Therefore, 3G is very important with regards to phone calls. It makes your calls more reliable, is used for calls with the majority of handsets. Cell towers transmit both 3G and 4G. We would like to know the cell tower information for both of these services.

4.3 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web tool for sharing and creating documents containing live programmes, images, and equations [17].

The IPython project has been forked into the Jupyter Notebook project. Jupyter is named after the three programming languages Julia, Python, and R that it supports. Although there are already over 100 alternative kernels available, Jupyter offers the IPython kernel, which allows you to develop Python programmes [18]. Python is available in a number of different distributions. This tutorial will only address two of them for the sake of installing Jupyter Notebook [18].

The official Python version, CPython, is the most popular and can be downloaded from their website.

The Jupyter Notebook is a dynamic online notebook that allows professors and students to combine computational data (code, data, statistics) with narrative, multimedia, and graphing. It can be used by the faculty to create interactive textbooks with plenty of explanations and examples that students can try out right in their browsers [17]. It can be used by students to explain their reasoning, exhibit their work, and make links between their classroom work and the outside world. It can be used by scientists, journalists, and academics to disseminate their data and share the tales behind their calculations and enable future collaboration and innovation.

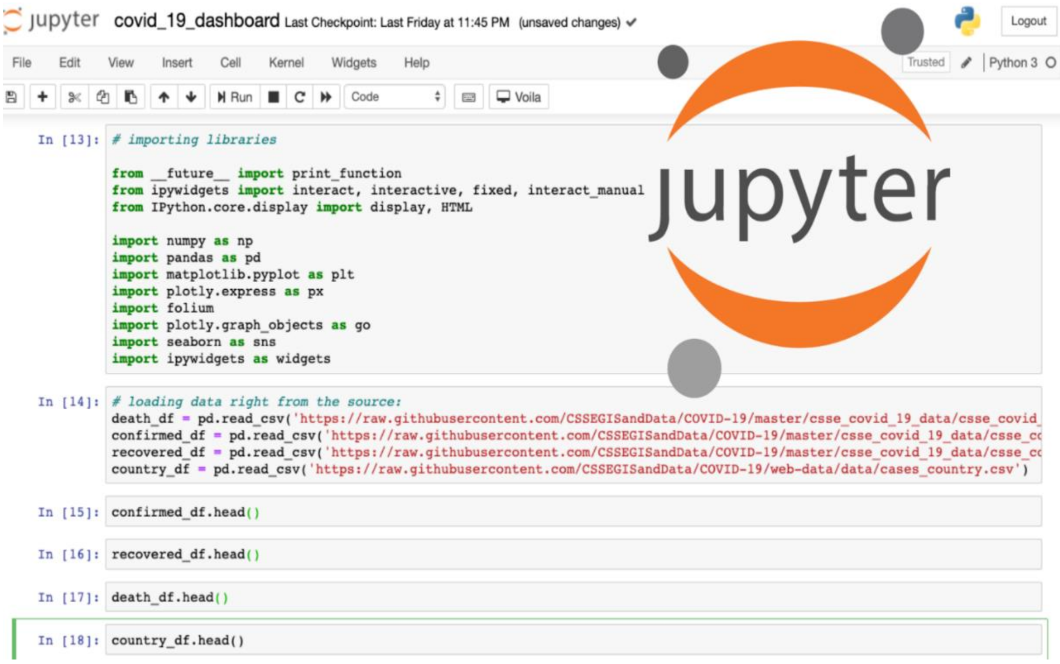

Fig 4.4: Jupyter Notebook [18].

Jupyter notebooks offer three distinct advantages:

1. They're excellent for displaying your work. Both the code and the results are visible. Kaggle's notebooks are a particularly good illustration of this [18].

2. Using other people's work as a starting point is simple. You may execute the code cell by cell to better understand what it does [18].

3. It's quite simple to host server-side, which is beneficial for security. A lot of data is sensitive and has to be secured, and one way to do it is to keep no data on local devices. This is free with a server-side Jupyter Notebook configuration [18].

When prototyping, the cell-based approach of Jupyter notebooks is great. But you quickly end up programming several steps — instead of looking at object-oriented programming.

CHAPTER 5

RESULTS AND DISCUSSION

As the models were decided which would eventually end up pretty beneficial in finding the solution we needed, we required a dataset which would help us with the accuracy we needed in the prediction of inconsistency among multiple metro stations. We opted for the busiest metro stations at different height levels in a metropolitan city like Delhi.

These locations are:

1. Mayur Vihar Phase-1
2. Whatever the fuck
3. Read point 2

5.1 MAYUR VIHAR PHASE – 1 METRO STATION

Mayur Vihar Phase 1 is an interchange metro station on the Delhi Metro's Blue and Pink lines. It is a Pink Line interchange station that opened on December 31, 2018. Pockets I and IV are both within walking distance of the station. For Pocket II, Trilok Puri, and Kalyan Puri, cycle-rickshaws and minibuses are available. The station's architecture is raised so that visitors can see the Commonwealth Sports Village skyline and the Yamuna River's flood plain.

We recorded multiple readings at platform in the metro station and we noted the RSS and created a dataset which we used to develop our graph and train our model to predict path loss. We maintained 5m gap for every reading and noted approx. 65 readings for this station.

We used two different SIM cards "**JIO 4G**" and "**VODAFONE 4G**" and noted the RSS for these SIM cards. A graph is drawn on the basis of its signal strength with distance

This station is between Akshardham Metro Station and Mayur Vihar Extension. There can be variation in the concentration of the people that gather at Mayur Vihar phase-1 metro station. The morning and evening rush hours can have a higher accumulation of people.

The morning rush hour timings are 9:00 AM – 11:00 AM

The evening rush hour timings are 6:00 PM – 8:00 PM

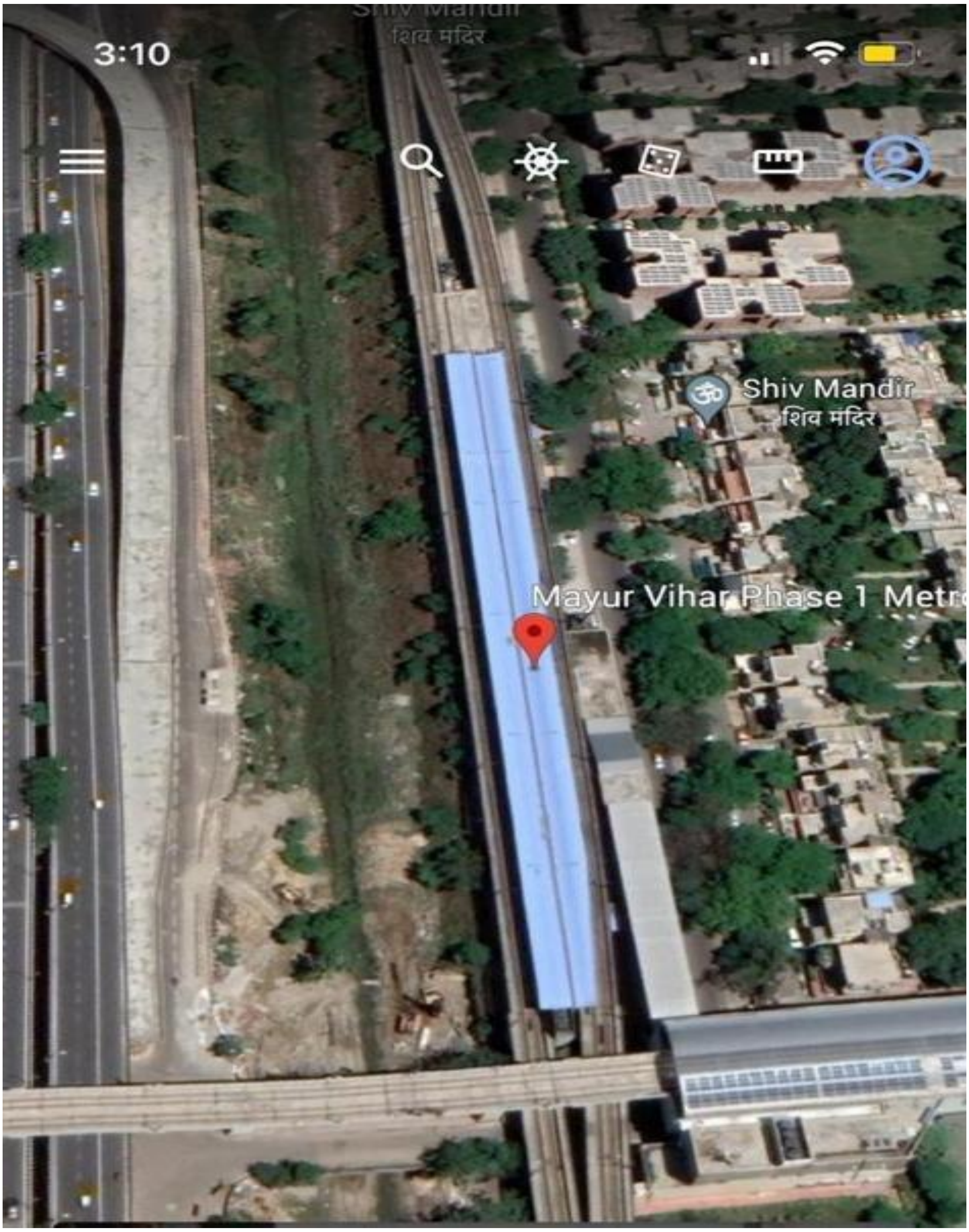

Fig 5.1: Satellite Image of Mayur Vihar phase-1 Metro Station

In the above figure the real time satellite image of Mayur Vihar phase-1 metro station has been shown. The image has been taken from google earth application. The image is showing the position of metro station and its surrounding.

Table 5.1: Dataset of Mayur Vihar Phase-1 Metro Station

Distance (in meter)	Jio4G	Vodafone	Jio Path Loss	Vodafone Path Loss
0	-86	-76	-17.2	-15.2
5	-87	-76	-17.4	-15.2
10	-84	-81	-16.8	-16.2
15	-84	-83	-16.8	-16.6
20	-85	-86	-17	-17.2
25	-79	-84	-15.8	-16.8
30	-81	-82	-16.2	-16.4
35	-83	-83	-16.6	-16.6
40	-79	-85	-15.8	-17
45	-90	-83	-18	-16.6
50	-89	-84	-17.8	-16.8
55	-83	-82	-16.6	-16.4
60	-88	-82	-17.6	-16.4
65	-78	-83	-15.6	-16.6
70	-74	-85	-14.8	-17
75	-73	-87	-14.6	-17.4
80	-86	-88	-17.2	-17.6
85	-89	-84	-17.8	-16.8
90	-89	-87	-17.8	-17.4
95	-86	-82	-17.2	-16.4
100	-87	-84	-17.4	-16.8
105	-85	-90	-17	-18
110	-84	-86	-16.8	-17.2
115	-75	-82	-15	-16.4
120	-82	-84	-16.4	-16.8
125	-87	-86	-17.4	-17.2
130	-92	-90	-18.4	-18
135	-89	-85	-17.8	-17
140	-84	-85	-16.8	-17
145	-90	-88	-18	-17.6
150	-91	-86	-18.2	-17.2
155	-84	-86	-16.8	-17.2
160	-87	-85	-17.4	-17

165	-84	-85	-16.8	-17
170	-79	-86	-15.8	-17.2
175	-90	-85	-18	-17
180	-87	-85	-17.4	-17
185	-87	-86	-17.4	-17.2
190	-76	-87	-15.2	-17.4
195	-83	-74	-16.6	-14.8
200	-82	-81	-16.2	-16.2
205	-83	-82	-16.6	-16.4
210	-85	-84	-17	-16.8
215	-86	-85	-17.2	-17
220	-86	-85	-17.2	-17
225	-82	-86	-16.2	-17.2
230	-78	-88	-15.6	-17.6
235	-79	-82	-15.8	-16.4
240	-81	-81	-16.2	-16.2
245	-83	-88	-16.6	-17.6
250	-85	-89	-17	-17.8
255	-88	-90	-17.6	-18
260	-90	-92	-18	-18.4
265	-90	-88	-18	-17.6
270	-88	-84	-17.6	-16.8
275	-87	-83	-17.4	-16.6
280	-84	-82	-16.8	-16.4
285	-86	-89	-17.2	-17.8
290	-79	-87	-15.8	-17.4
295	-81	-84	-16.2	-16.8
300	-81	-87	-16.2	-17.4
305	-86	-80	-17.2	-16
310	-89	-80	-17.8	-16
315	-83	-83	-16.6	-16.6

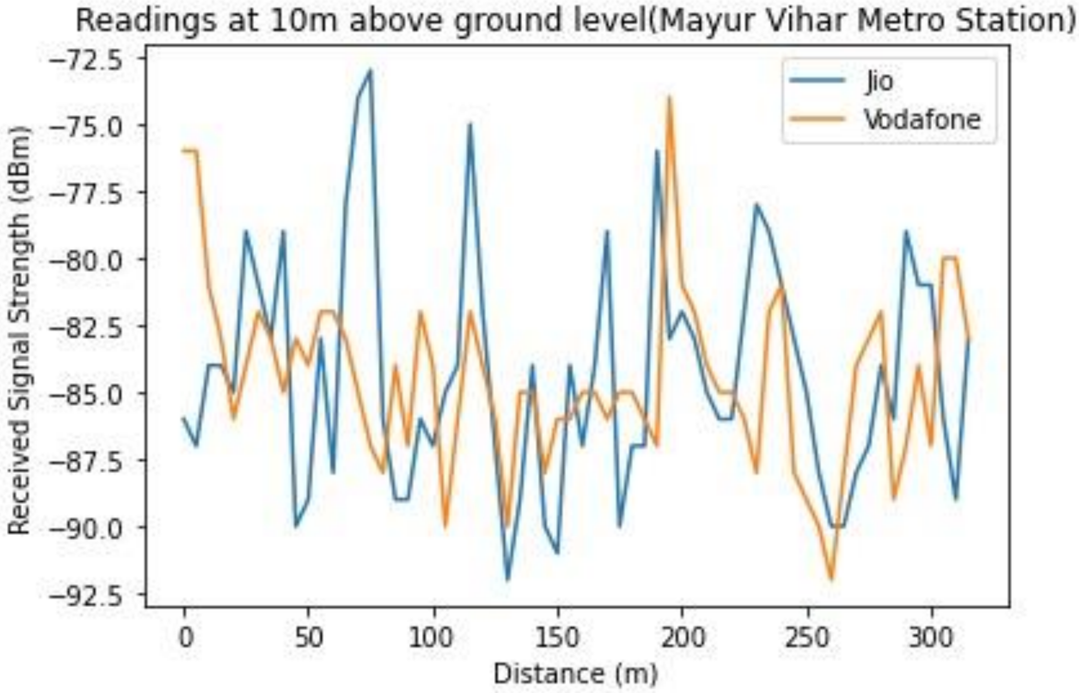

Fig 5.2: RSS Variation with respect to Distance

We measured the Received Signal Strength readings and created a statistic for the Received Signal Strength readings and trained our model and we found a graph with different points on it showing the variation of RSS with respect to Distance. The Graph shows the different points on axis and its between RSS and distance is shown in above figure.

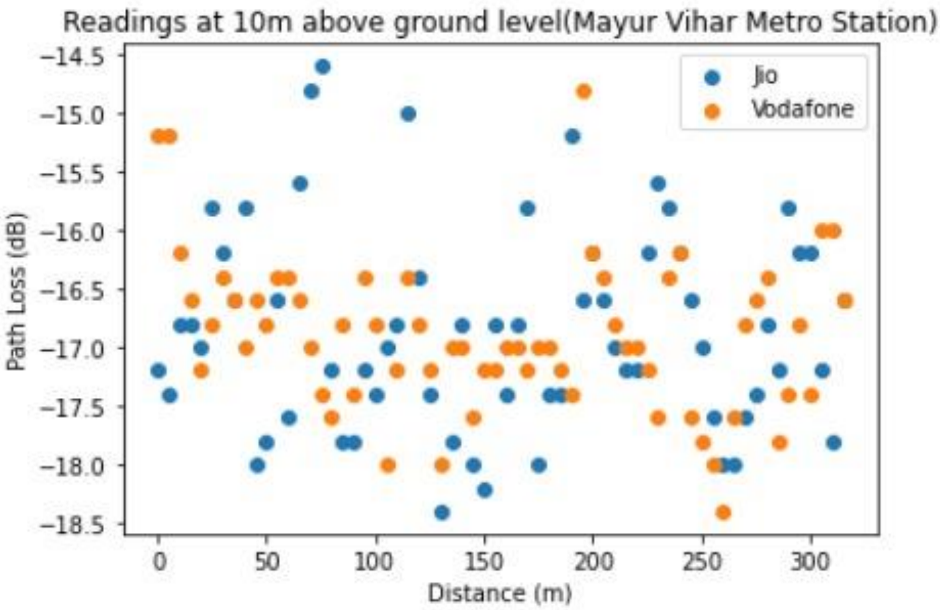

Fig 5.3: Variation of Path Loss with Distance at Mayur Vihar phase-1 Metro Station

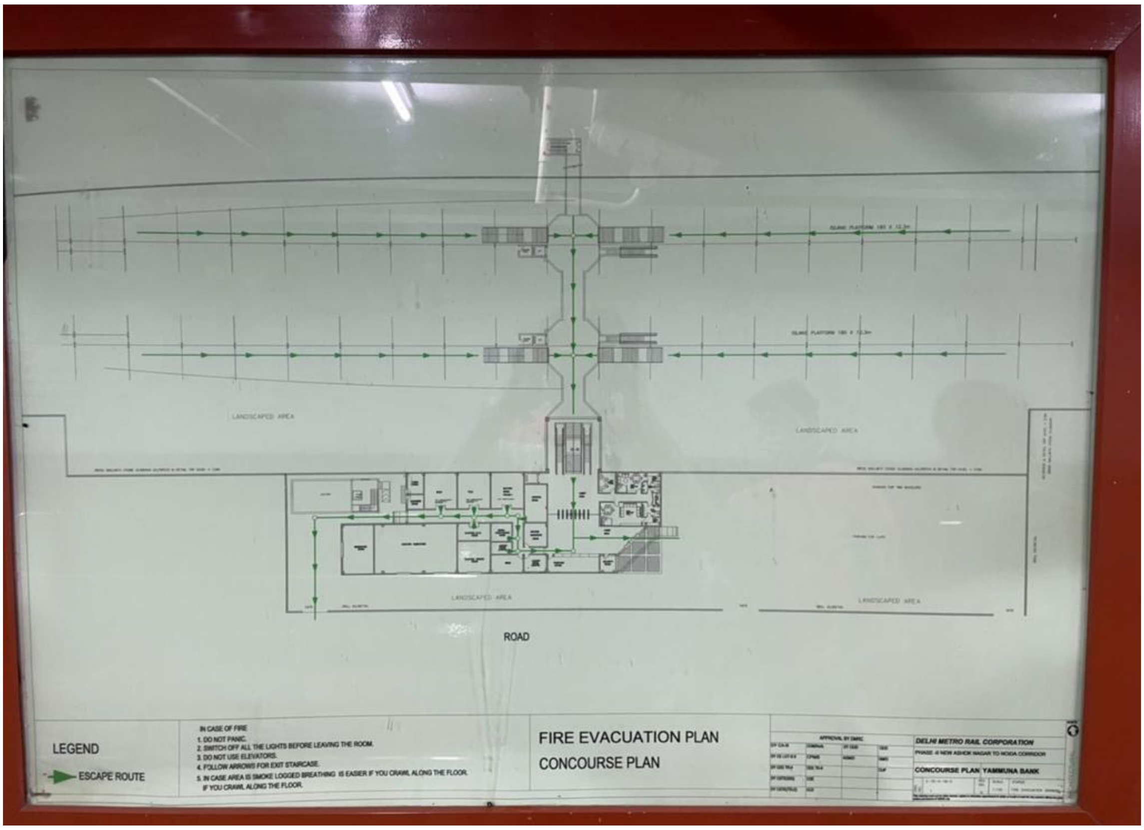

Fig 5.4: Layout of Mayur Vihar Phase-1 Metro Station

We calculated the path loss for these RSS readings and created a statistic for the path loss readings and trained our model with it and we found a graph with different points on it predicting the path loss. The Graph shows the different points on axis and its between path loss and distance.

Table shown below states the data with the value of parameters of statistical dataset of Mayur vihar-1 metro station. This data is calculated with the help of Jupyter Notebook using different machine learning models. With the help of this data we came across the statistical value of our readings.

Table 5.2: Propagation Parameters of Mayur Vihar Phase 1 Metro Station

Parameter	Distance (in meter)	JIO 4G	VODAFONE	JIO Path Loss	VODAFONE Path Loss
Count	64.000000	64.000000	64.000000	64.000000	64.000000
Mean	157.50000	-84.34375	-84.50000	-16.86250	-16.900000
Std	93.094934	4.310264	3.328568	0.866209	0.66571400
Min	0.0000000	-92.00000	-92.00000	-18.40000	-18.400000
25%	78.750000	-87.00000	-86.25000	-17.40000	-17.250000
50%	157.50000	-85.00000	-85.00000	-17.00000	-17.000000
75%	236.25000	-82.00000	-82.75000	-16.20000	-16.550000
max	315.00000	-73.00000	-74.00000	-14.60000	-14.800000

5.2 YAMUNA BANK METRO STATION

The Yamuna Bank Metro Station is on the Delhi Metro's Blue Line. This station acts as a hub for the Noida and Vaishali branches of the Blue Line. Cross-platform transfer in the same direction of travel is possible thanks to two island platforms. The phrase "free school" refers to a school that is located beneath the bridge near the station.

The Yamuna Bank depot is situated beside the at-grade station. We recorded multiple readings at platform in the metro station and we noted the RSS and created a dataset which we used to develop our graph and train our model to predict path loss. We maintained 5m gap for every reading and noted approx. 70 readings for this station.

We used two different SIM cards “**JIO 4G**” and “**VODAFONE 4G**” and noted the RSS for these SIM cards. A graph is drawn on the basis of its signal strength with distance

This station is between Akshardham Metro Station and Indraprastha. There can be variation in the concentration of the people that gather at Yamuna Bank metro station. The morning and evening rush hours can have a higher accumulation of people.

The morning rush hour timings are 9:00 AM – 11:00 AM

The evening rush hour timings are 6:00 PM – 8:00 PM

Fig 5.5: Satellite Image of Yamuna Bank Metro Station

We measured the Received Signal Strength readings and created a statistic for the Received Signal Strength readings and trained our model and we found a graph with different points on it showing the variation of RSS with respect to Distance. The Graph shows the different points on axis and its between RSS and distance.

Table 5.3: Dataset of Yamuna Bank Metro Station

Distance (In Meter)	Jio4G	Vodafone	Jio Path Loss	Vodafone Path Loss
0	-74	-75	-14.8	-15
5	-85	-78	-17	-15.6
10	-82	-73	-16.4	-14.6
15	-83	-74	-16.6	-14.8
20	-85	-71	-17	-14.2
25	-80	-73	-16	-14.6
30	-82	-75	-16.4	-15
35	-83	-80	-16.6	-16
40	-84	-76	-16.8	-15.2
45	-82	-78	-16.4	-15.6
50	-80	-83	-16	-16.6
55	-82	-77	-16.4	-15.4
60	-78	-75	-15.6	-15
65	-83	-81	-16.6	-16.2
70	-83	-78	-16.6	-15.6
75	-80	-83	-16	-16.6
80	-80	-85	-16	-17
85	-83	-84	-16.6	-16.8
90	-88	-81	-17.6	-16.2
95	-87	-82	-17.4	-16.4
100	-86	-83	-17.2	-16.6
105	-88	-82	-17.6	-16.4
110	-89	-85	-17.8	-17
115	-84	-82	-16.8	-16.4
120	-82	-84	-16.4	-16.8
125	-81	-81	-16.2	-16.2
130	-80	-74	-16	-14.8
135	-78	-82	-15.6	-16.4
140	-79	-80	-15.8	-16
145	-80	-81	-16	-16.2
150	-76	-75	-15.2	-15
155	-77	-84	-15.4	-16.8
160	-69	-81	-13.8	-16.2
165	-70	-80	-14	-16

170	-68	-86	-13.6	-17.2
175	-66	-87	-13.2	-17.4
180	-78	-82	-15.6	-16.4
185	-82	-82	-16.4	-16.4
190	-80	-81	-16	-16.2
195	-79	-80	-15.8	-16
200	-78	-82	-15.6	-16.4
205	-79	-80	-15.8	-16
210	-79	-75	-15.8	-15
215	-85	-78	-17	-15.6
220	-82	-73	-16.4	-14.6
225	-83	-74	-16.6	-14.8
230	-82	-75	-16.4	-15
235	-80	-83	-16	-16.6
240	-88	-81	-17.6	-16.2
245	-88	-82	-17.6	-16.4
250	-84	-82	-16.8	-16.4
255	-80	-84	-16	-16.8
260	-81	-81	-16.2	-16.2
265	-80	-74	-16	-14.8
270	-80	-77	-16	-15.2
275	-82	-78	-16.4	-15.3
280	-83	-79	-16.6	-15.8
285	-85	-85	-17	-17
290	-84	-84	-16.8	-16.8
295	-86	-83	-17.2	-16.6
300	-88	-82	-17.6	-16.4
305	-84	-87	-16.8	-17.4
310	-82	-80	-16.4	-16
315	-79	-81	-15.8	-16.2
320	-78	-79	-15.6	-15.8
325	-77	-80	-15.4	-16
330	-80	-82	-16	-16.4
335	-82	-84	-16.4	-16.8
340	-84	-88	-16.8	-17.6
345	-85	-89	-17	-17.8

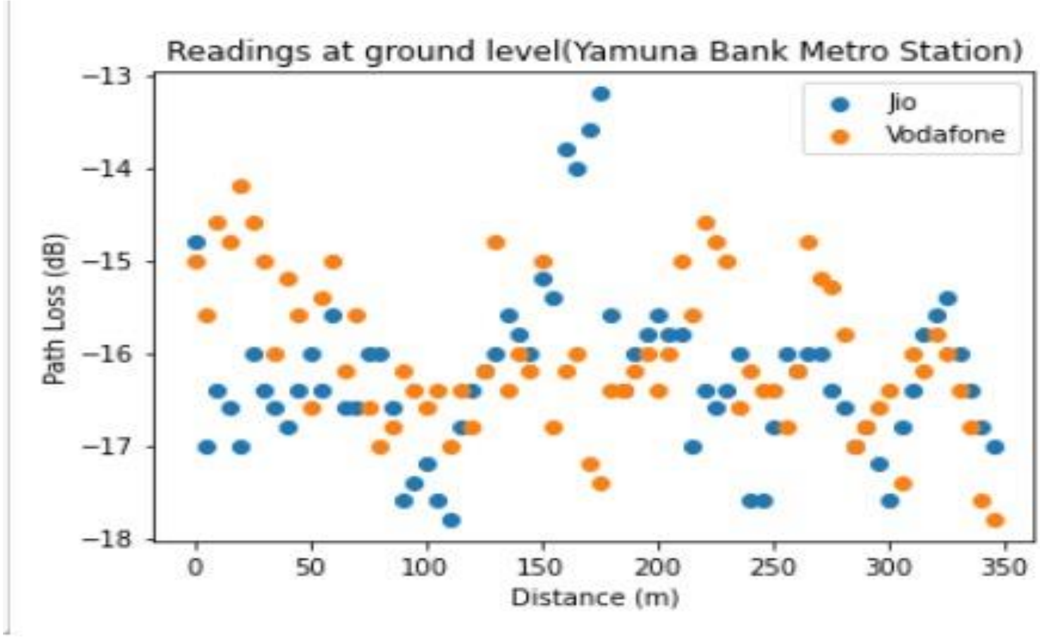

Fig 5.6: Variation of Path Loss with Distance at Yamuna Bank Metro Station

The graph between varying path loss with distance is shown above. This graph is based on the readings taken at Yamuna Bank metro station which is at ground level. We measured the Received Signal Strength readings and created a statistic for the Received Signal Strength readings and trained our model and we found a graph with different points on it showing the variation of RSS with respect to Distance. The Graph shows the different points on axis and its between RSS and distance.

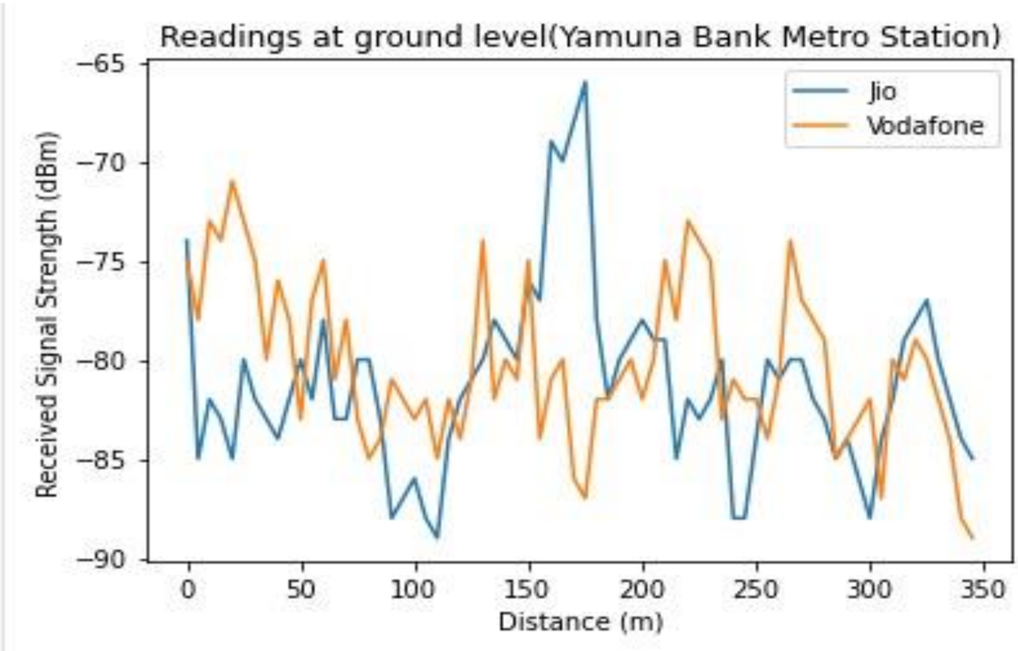

Fig 5.7: RSS Variation with respect to Distance

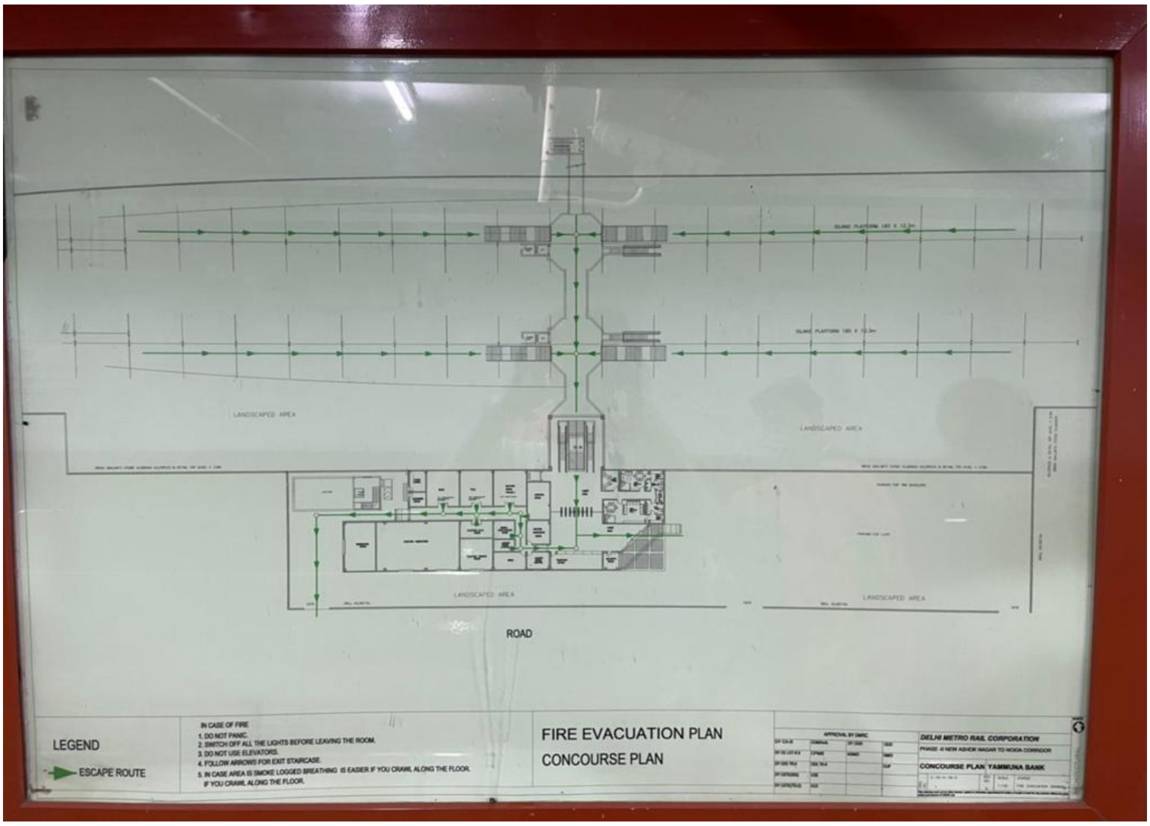

Fig 5.8: Layout of Yamuna Bank Metro Station

We calculated the path loss for these RSS readings and created a statistic for the path loss readings and trained our model with it and we found a graph with different points on it predicting the path loss. The Graph shows the different points on axis and its between path loss and distance.

Table 5.4: Propagation Parameters of Mayur Vihar Phase 1 Metro Station

Parameter	**Distance (in meter)**	**JIO 4G**	**VODAFONE**	**JIO Path Loss**	**VODAFONE Path Loss**
Count	70.000000	70.000000	70.000000	70.000000	70.000000
Mean	172.50000	-81.20000	-80.228571	-16.24000	-16.038571
Std	101.755426	4.515994	4.033093	0.903199	0.812456
Min	0.0000000	-89.00000	-89.00000	-17.80000	-17.800000
25%	86.250000	-84.00000	-83.25000	-16.80000	-16.60000
50%	157.50000	-82.00000	-81.00000	-16.40000	-16.200000
75%	258.75000	-79.25000	-78.00000	-15.85000	-15.450000
max	345.00000	-66.00000	-71.00000	-13.20000	-14.200000

5.3 RAJIV CHOWK METRO STATION

The Rajiv Chowk metro station is located on the Blue and Yellow Lines in New Delhi. On the top level, it serves as a transfer station for the Blue Line, while on the lower level, it serves as a transition station for the Yellow Line. It serves Rajiv Chowk in the centre of Delhi and is one of the busiest stations on the network. Every day, it transports 5 lakh passengers. The size of the Connaught Place metro station is about 39,503 square feet (3,669.9 m2).

Just outside the station are several companies, major buildings, restaurants, and movies. The station is built beneath Central Park. Rajiv Chowk, formerly known as Connaught Circus, is the official name of Connaught Place. The Home Ministry ordered in September 1995 that the 75-year-old Connaught Place (CP) would be renamed Rajiv Chowk and Connaught Circus would be renamed Indira Chowk.

We recorded multiple readings at platform in the metro station and we noted the RSS and created a dataset which we used to develop our graph and train our model to predict path loss. We maintained 5m gap for every reading and noted approx. 70 readings for this station. We used two different SIM cards “JIO 4G” and “VODAFONE 4G” and noted the RSS for these SIM cards [17]. A graph is drawn on the basis of its signal strength with distance

This station is between Barakhamba Road and Patel Chowk. There can be variation in the concentration of the people that gather at Rajeev Chowk metro station. The morning and evening rush hours can have a higher accumulation of people.

The morning rush hour timings are 9:00 AM – 11:00 AM
The evening rush hour timings are 6:00 PM – 8:00 PM

We have attached a satellite image of Rajiv chowk metro station with the help of an app called earth satellite. It is a real time image of this metro station situated at Connaught Place in New Delhi. It has 4 platforms. The commuters can check the facilities available at Rajiv Chowk Metro Station, Rajiv Chowk First and Last Metro Timings and Entry & Exit Gates information.

A number of selfie points have been set up by the Delhi Metro authorities at its prominent stations, including at Rajiv Chowk and JLN Stadium, to cheer up the athletes taking part in Tokyo Olympics, officials said on Friday. India is being represented by its largest ever contingent of 127 athletes at the Tokyo Games, which also includes the highest female representation of 56 sportspersons.

"Pledging support to Indian athletes at the Tokyo Olympics, DMRC has installed selfie points at several prominent stations. Many enthusiastic patrons joined the entire nation in cheering for our athletes. #Cheer4India," the DMRC tweeted.

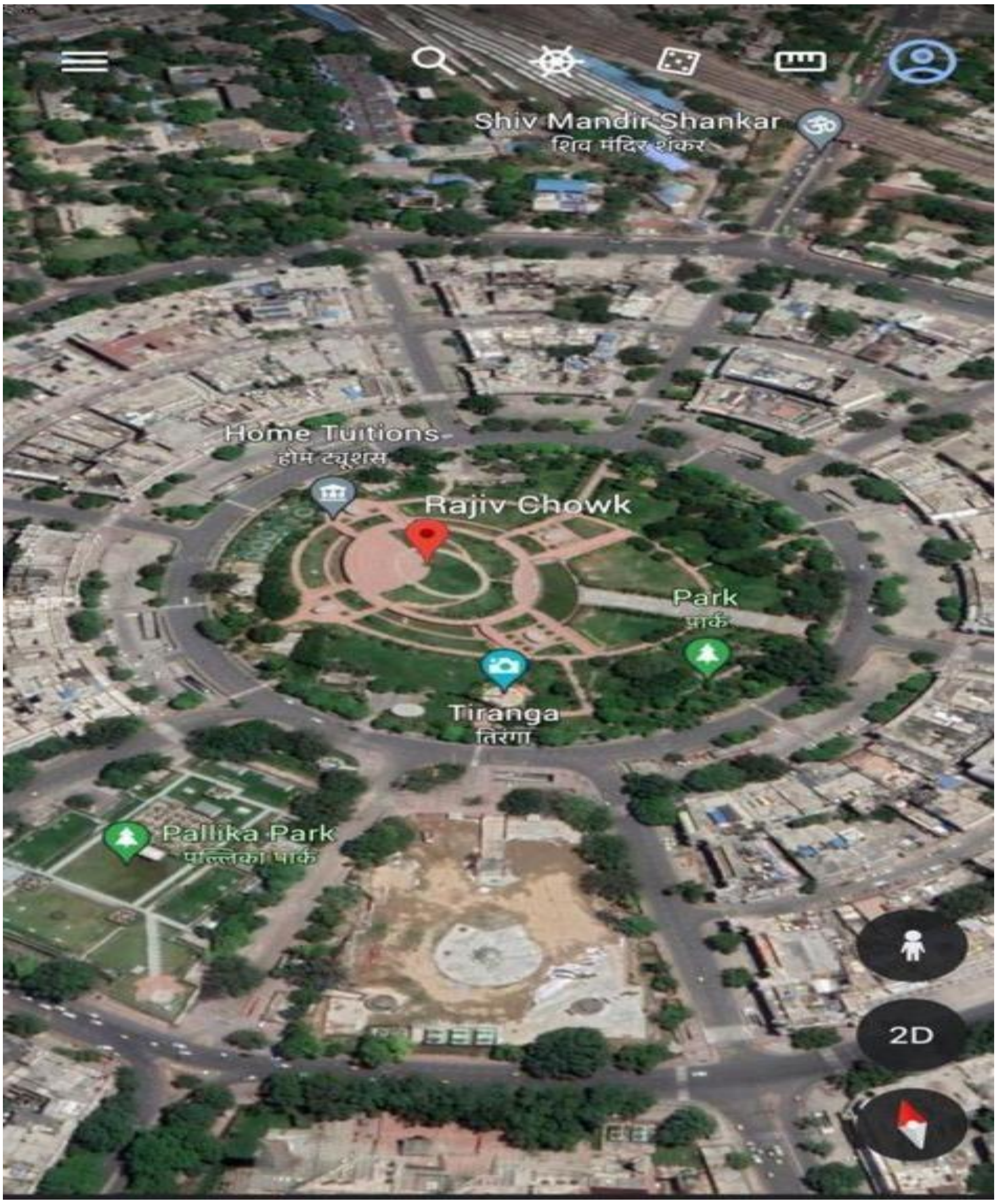

Fig 5.9: Satellite Image of Rajiv Chowk Metro Station

We measured the Received Signal Strength readings and created a statistic for the Received Signal Strength readings and trained our model and we found a graph with different points on it showing the variation of RSS with respect to Distance. The Graph shows the different points on axis and its between RSS and distance.

Table 5.5: Dataset of Rajiv Chowk Metro Station

Distance (In Meter)	**Jio4G**	**Vodafone**	**Jio Path Loss**	**Vodafone Path Loss**
0	-81	-110	-16.2	-22
5	-79	-109	-15.8	-21.8
10	-87	-108	-17.4	-21.6
15	-81	-107	-16.2	-21.4
20	-82	-100	-16.4	-20
25	-81	-109	-16.2	-21.8
30	-83	-104	-16.6	-20.8
35	-76	-102	-15.2	-20.4
40	-77	-108	-15.4	-21.6
45	-94	-108	-18.8	-21.6
50	-83	-103	-16.6	-20.6
55	-81	-112	-16.2	-22.4
60	-82	-108	-16.4	-21.6
65	-77	-104	-15.4	-20.8
70	-81	-102	-16.2	-20.4
75	-78	-95	-15.6	-19
80	-79	-94	-15.8	-18.8
85	-76	-89	-15.2	-17.8
90	-82	-93	-16.4	-18.6
95	-77	-94	-15.4	-18.8
100	-85	-87	-17	-17.4
105	-81	-79	-16.2	-15.8
110	-71	-77	-14.2	-15.4
115	-81	-81	-16.2	-16.2
120	-77	-83	-15.4	-16.6
125	-89	-75	-17.8	-15
130	-92	-74	-18.4	-14.8
135	-74	-79	-14.8	-15.8

140	-78	-78	-15.6	-15.6
145	-91	-74	-18.2	-14.8
150	-75	-72	-15	-14.4
155	-97	-74	-19.4	-14.8
160	-92	-72	-18.4	-14.4
165	-95	-71	-19	-14.2
170	-79	-74	-15.8	-14.8
175	-78	-71	-15.6	-14.2
180	-86	-69	-17.2	-13.8
185	-80	-69	-16	-13.8
190	-73	-69	-14.6	-13.8
195	-76	-70	-15.2	-14
200	-65	-71	-13	-14.2
205	-68	-70	-13.6	-14
210	-70	-72	-14	-14.4
215	-71	-72	-14.2	-14.4
220	-73	-75	-14.6	-15
225	-78	-79	-15.6	-15.8
230	-85	-81	-17	-16.2
235	-88	-90	-17.6	-18
240	-90	-91	-18	-18.2
245	-92	-95	-18.4	-19
250	-92	-97	-18.4	-19.4
255	-88	-94	-17.6	-18.8
260	-85	-90	-17	-18
265	-88	-88	-17.6	-17.6
270	-86	-89	-17.2	-17.8
275	-87	-90	-17.4	-18
280	-90	-95	-18	-19
285	-90	-96	-18	-19.2
290	-92	-100	-18.4	-20
295	-94	-99	-18.8	-19.8
300	-84	-103	-16.8	-20.6
305	-83	-104	-16.6	-20.8
310	-86	-108	-17.2	-21.6
315	-88	-108	-17.6	-21.6
320	-82	-109	-16.4	-21.8

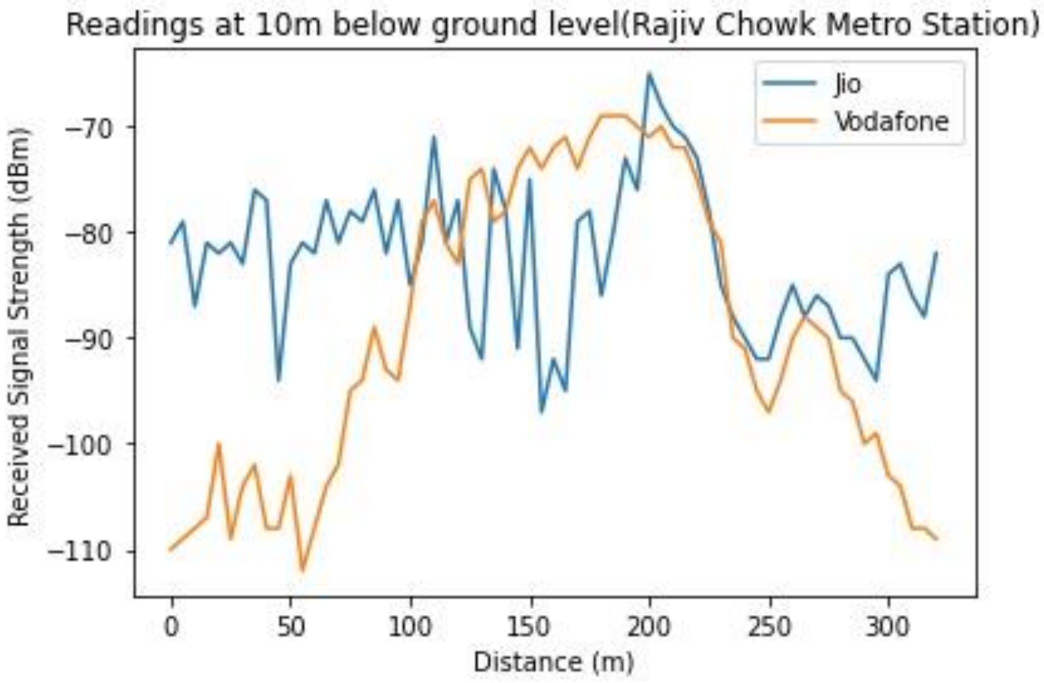

Fig 5.10: RSS Variation with respect to Distance

We calculated the path loss for these RSS readings and created a statistic for the path loss readings and trained our model with it and we found a graph with different points on it predicting the path loss. The Graph shows the different points on axis and its between path loss and distance.

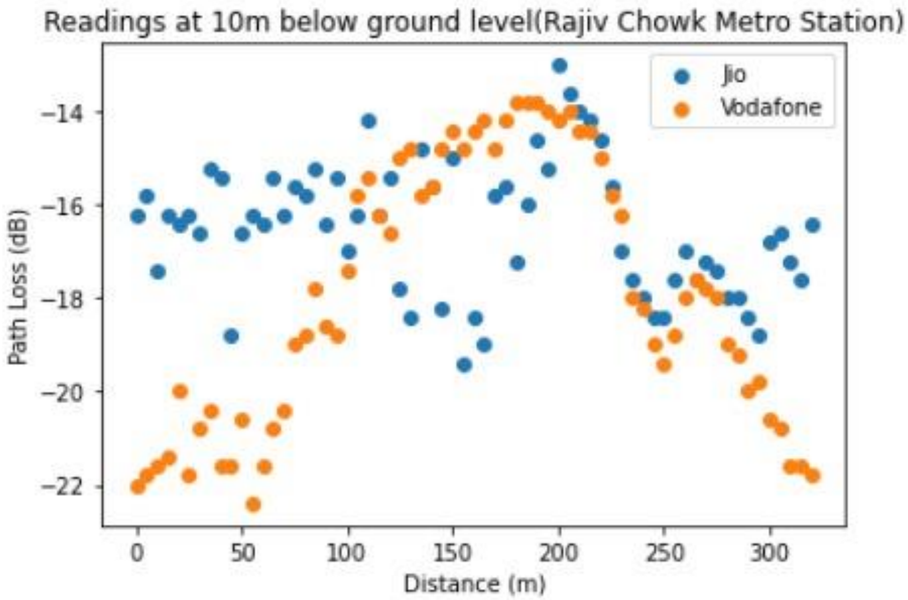

Fig 5.11: Variation of Path Loss with Distance at Rajiv Chowk Metro Station

Fig 5.12: Layout of Rajiv Chowk Metro Station

Table 5.6: Propagation Parameters of Rajiv Chowk Metro Station

Parameter	Distance (in meter)	JIO 4G	VODAFONE	JIO Path Loss	VODAFONE Path Loss
Count	65.000000	65.000000	65.000000	65.000000	65.000000
Mean	160.00000	-82.49230	-89.584615	-16.49846	-17.916923
Std	94.538352	7.093402	14.130964	1.418680	2.826193
Min	0.0000000	-97.00000	-112.00000	-19.40000	-22.400000
25%	80.000000	-88.00000	-103.00000	-17.80000	-20.60000
50%	160.00000	-82.00000	-90.000000	-16.40000	-18.200000
75%	240.00000	-78.25000	-75.000000	-15.60000	-15.000000
max	320.00000	-65.00000	-69.000000	-13.00000	-13.800000

CHAPTER – 6

IMPACT ON SOCIETY

- As we travelled pretty much every day to our institutions to gain knowledge, being day scholars, we had to travel through metros and came across an issue we couldn’t overlook.

- We realized the uneven and inconsistent signal strength that varied from one metro station to another we wondered what the reason might be?

- with curious minds we searched for ways to get the answers we were looking for and came across path loss and realized that it was the methodology we were actually looking for We took the readings at 3 of the busiest platforms in Delhi, a metropolitan city which would enable us get accurate real time readings and eventually let us understand the problem better.

- We took the readings at Mayur Vihar phase-1, Rajiv chowk and Yamuna bank metro stations and at each station we took readings on the platform at a distance of 5m each whereas another piece of information which ended up being beneficial for us was the height of the platform which was 10m below ground level, at ground level and 10m above ground level respectively.

- After the readings were taken which were namely around 60 observations. We used 3 models to accurately identify and predict which model is to be proposed which would eventually help us realize the level of inconsistency among these metro stations.

- These models were:
    - Linear regression model
    - Non-linear regression model
    - Decision tree regression model.

- After incorporating these readings into jupyter notebook and running the code we realized how accurate decision tree was for the problem we faced and how accurately it predicted the necessary answers. We also found linear regression model to be pretty accurate as well and was pretty close enough to decision tree in terms of accuracy. Since decision tree was way more accurate, decision tree was our proposed model and it helped us determine the inconsistency in the signal strength among these metro stations.

- This is the kind of model which will help us determine the inconsistency of the data signals among various locations and will eventually help the people and the concerned authorities to take counter measures for the proposed problem and find solutions for it. The model that ended up as the proposed model was the Decision tree regression model, we obtained 100% accuracy from it (+_2%).

- So, it'll be helpful in the application in finding solutions in the signal inconsistency issue at multiple locations.

CHAPTER-7

CONCLUSION

With the development and deployment of 4G networks, network planning puts forward higher requirements on the accuracy, complexity, and versatility of path loss prediction. Machine learning methods, can model hidden non-linear relationships and thus can be used for path loss prediction. Based on historical data, machine-learning-based models can build relationship between path loss and input features. By implementing the project, we have issued that due to loss of signal strength there's an issue in our network while traveling through different metro stations at different height. By this project we developed a model to find out the signal strength and the path losses during different time period with different concentrations of crowd. There is room for a number of improvements in this project as data and accuracy can vary with huge numbers of data.

7.1 LIMITATIONS

- One of the limitations that we came across and realized can be something that might vary the accuracy in the dataset was the concentration of people at each metro station.

- We also realized that the time of the day can also vary the signal strength to some extent and since we cannot obtain the readings at all times during a day this can also be counted as a limitation.

APPENDIX – 1

CODE FOR MAYUR VIHAR-1 METRO STATION

LINEAR REEGRESSION:

```
import pandas as pd
import seaborn as sns
import numpy as np

data = pd.read_csv(r'mayurvihardataset.csv')

data.head()

data.tail()

data.shape

data.describe()

# JioPathLoss

sns.relplot(x = 'DistanceInMeter', y = 'JioPathLoss', data = data)

print(data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone'], axis = 1)
test = data['JioPathLoss']
```

```
X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state
= 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred

regr.score(X_test, y_test)

# VodafonePathLoss

sns.relplot(x = 'DistanceInMeter', y = 'VodafonePathLoss', data = data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
test = data['VodafonePathLoss']

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state
= 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred

regr.score(X_test, y_test)
```

NON- LINEAR REGRESION:

```
import pandas as pd

x = pd.read_csv(r'mayurvihardataset.csv')

x.describe()

# JioPathLoss

import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)



g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])



plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')



#to use curve fit

from scipy.optimize import curve_fit
```

```
d = x['DistanceInMeter'].values
pl = x['JioPathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 5000) #maxfev is the max number of iterations
print(c)


for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])


plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')

# r squared value

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)

# VodafonePathLoss

import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)

g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])
```

```
plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')



from scipy.optimize import curve_fit
d = x['DistanceInMeter'].values
pl = x['VodafonePathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 5000) #maxfev is the max number of iterations
print(c)



for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])



plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)
```

DECISION TREE:

JIO:

```
import pandas as pd
df = pd.read_csv(r'mayurvihardataset.csv')
df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone', 'Jio4G'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'VodafonePathLoss'], axis = 1)

# target
inputs
```

```
from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Jio4G = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Jio4G_n'] = le_DistanceInMeter.fit_transform(inputs['Jio4G'])
inputs_n = inputs.drop(['DistanceInMeter', 'Jio4G'], axis = 1)


le_JioPathLoss = LabelEncoder()
target['JioPathLoss_n'] = le_JioPathLoss.fit_transform(target['JioPathLoss'])
target_n = target.drop('JioPathLoss', axis = 1)
target_n

inputs_n

from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 8]])
```

VODAFONE:

```
import pandas as pd
df = pd.read_csv(r'mayurvihardataset.csv')
df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'JioPathLoss'], axis = 1)
```

```
target

from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Vodafone = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Vodafone_n'] = le_DistanceInMeter.fit_transform(inputs['Vodafone'])
inputs_n = inputs.drop(['DistanceInMeter', 'Vodafone'], axis = 1)


le_VodafonePathLoss = LabelEncoder()
target['VodafonePathLoss_n'] =
le_VodafonePathLoss.fit_transform(target['VodafonePathLoss'])
target_n = target.drop('VodafonePathLoss', axis = 1)
target_n

inputs_n

from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 8]])
```

APPENDIX – 2

CODE FOR YAMUNA BANK METRO STATION

LINEAR REGRESSION:

```
import pandas as pd

import seaborn as sns
import numpy as np

data = pd.read_csv(r'yamunabankdataset.csv')

data.describe()

# JioPathLoss

sns.relplot(x = 'DistanceInMeter', y = 'JioPathLoss', data = data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone'], axis = 1)
test = data['JioPathLoss']

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state
= 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred
```

```
regr.score(X_test, y_test)

# VodafonePathLoss

sns.relplot(x = 'DistanceInMeter', y = 'VodafonePathLoss', data = data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
test = data['VodafonePathLoss']

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state
= 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred

regr.score(X_test, y_test)
```

NON- LINEAR REGRESSION:

```
import pandas as pd

x = pd.read_csv(r'yamunabankdataset.csv')

x.describe()
```

```
# JioPathLoss

import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)



g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])



plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')



from scipy.optimize import curve_fit
d = x['DistanceInMeter'].values
pl = x['JioPathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 9000) #maxfev is the max number of iterations
print(c)



for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])



plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')
```

```
# r squared value

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)

# VodafonePathLoss

import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)

g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])
plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')


from scipy.optimize import curve_fit
d = x['DistanceInMeter'].values
pl = x['VodafonePathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 5000) #maxfev is the max number of iterations
print(c)


for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])
```

```
plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)
```

DECISION TREE:

JIO:

```
import pandas as pd
df = pd.read_csv(r'yamunabankdataset.csv')

df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'VodafonePathLoss'], axis = 1)

target

from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Jio4G = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Jio4G_n'] = le_DistanceInMeter.fit_transform(inputs['Jio4G'])
inputs_n = inputs.drop(['DistanceInMeter', 'Jio4G'], axis = 1)



le_JioPathLoss = LabelEncoder()
target['JioPathLoss_n'] = le_JioPathLoss.fit_transform(target['JioPathLoss'])
target_n = target.drop('JioPathLoss', axis = 1)
target_n

inputs_n
```

```
from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 6]])
```

VODAFONE:

```
import pandas as pd

df = pd.read_csv(r'yamunabankdataset.csv')

df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'JioPathLoss'], axis = 1)

target

inputs

from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Vodafone = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Vodafone_n'] = le_DistanceInMeter.fit_transform(inputs['Vodafone'])
inputs_n = inputs.drop(['DistanceInMeter', 'Vodafone'], axis = 1)


le_VodafonePathLoss = LabelEncoder()
```

```
target['VodafonePathLoss_n'] =
le_VodafonePathLoss.fit_transform(target['VodafonePathLoss'])
target_n = target.drop('VodafonePathLoss', axis = 1)
target_n

inputs_n

from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 15]])
```

APPENDIX-3

CODE FOR RAJIV CHOWK METRO STATION

LINEAR REGRESSION:

```
import pandas as pd

import seaborn as sns

data = pd.read_csv(r'rajivchowkdataset.csv')

data.describe()

# JioPathLoss

sns.relplot(x = 'DistanceInMeter', y = 'JioPathLoss', data = data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone'], axis = 1)
test = data['JioPathLoss']

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state
= 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred
```

```
regr.score(X_test, y_test)

# VodafonePathLoss

sns.relplot(x = 'DistanceInMeter', y = 'VodafonePathLoss', data = data)

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train = data.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
test = data['VodafonePathLoss']

X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.3, random_state = 2)

regr = LinearRegression()

regr.fit(X_train, y_train)

pred = regr.predict(X_test)

pred

regr.score(X_test, y_test)
```

NON-LINEAR REGRESSION:

```
import pandas as pd

x = pd.read_csv(r'rajivchowkdataset.csv')

x.describe()

# JioPathLoss
```

```
import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)


g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])


plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')


from scipy.optimize import curve_fit
d = x['DistanceInMeter'].values
pl = x['JioPathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 10000) #maxfev is the max number of iterations
print(c)


for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])


plt.plot(x['DistanceInMeter'], x['JioPathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')
```

```
# r squared value

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)

# VodafonePathLoss

import numpy as np
def func(distance, c0, c1, c2, c3):
    return c0 + c1 * distance - c2 * np.exp(-c3 * distance)

g = [100, 0.01, 100, 0.01]

%matplotlib inline
import matplotlib.pyplot as plt

n = len(x['DistanceInMeter'])
y = np.empty(n)
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], g[0], g[1], g[2], g[3])


plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')


from scipy.optimize import curve_fit
d = x['DistanceInMeter'].values
pl = x['VodafonePathLoss'].values
c, cov = curve_fit(func, d, pl, g, maxfev = 20000) #maxfev is the max number of iterations
print(c)
```

```
for i in range(n):
    y[i] = func(x['DistanceInMeter'][i], c[0], c[1], c[2], c[3])



plt.plot(x['DistanceInMeter'], x['VodafonePathLoss'])
plt.plot(x['DistanceInMeter'], y, 'r.')

from sklearn.metrics import r2_score
print('R^2: ', r2_score(y, pl)) # (predicted, measured)
```

DECISION TREE:

JIO:

```
import pandas as pd
df = pd.read_csv(r'rajivchowkdataset.csv')
df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Vodafone'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'VodafonePathLoss'], axis = 1)

from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Jio4G = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Jio4G_n'] = le_DistanceInMeter.fit_transform(inputs['Jio4G'])
inputs_n = inputs.drop(['DistanceInMeter', 'Jio4G'], axis = 1)



le_JioPathLoss = LabelEncoder()
target['JioPathLoss_n'] = le_JioPathLoss.fit_transform(target['JioPathLoss'])
target_n = target.drop('JioPathLoss', axis = 1)
target_n
```

```
inputs_n

from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 14]])
```

VODAFONE:

```
import pandas as pd
df = pd.read_csv(r'rajivchowkdataset.csv')
df.head()

inputs = df.drop(['JioPathLoss', 'VodafonePathLoss', 'Jio4G'], axis = 1)
target = df.drop(['DistanceInMeter', 'Jio4G', 'Vodafone', 'JioPathLoss'], axis = 1)

from sklearn.preprocessing import LabelEncoder
le_DistanceInMeter = LabelEncoder()
le_Vodafone = LabelEncoder()
inputs['Distance_n'] = le_DistanceInMeter.fit_transform(inputs['DistanceInMeter'])
inputs['Vodafone_n'] = le_DistanceInMeter.fit_transform(inputs['Vodafone'])
inputs_n = inputs.drop(['DistanceInMeter', 'Vodafone'], axis = 1)


le_VodafonePathLoss = LabelEncoder()
target['VodafonePathLoss_n'] =
le_VodafonePathLoss.fit_transform(target['VodafonePathLoss'])
target_n = target.drop('VodafonePathLoss', axis = 1)
target_n
```

```
inputs_n

from sklearn import tree

model = tree.DecisionTreeClassifier()

model.fit(inputs_n, target_n)

model.score(inputs_n, target_n)

model.predict([[3, 4]])
```

REFERENCES

1. Rappaport, T.S. (2002) Wireless Communications: Principles and Practice, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA.

2. Phillips, C.; Sicker, D.; Grunwald, D. (2013) A survey of wireless path loss prediction and coverage mapping methods. IEEE Commun. Surv. Tutor. 15, 255–270. [CrossRef]

3. Östlin, E.; Zepernick, H.J.; Suzuki, H. Macrocell (2010) path-loss prediction using artificial neural networks. IEEE Trans. Veh. Technol. 59, 2735–2747. [CrossRef]

4. A. Al-Hourani, S. Kandeepan, S. Lardner (2014) Optimal LAP altitude for maximum coverage, IEEE Wireless Commun. Lett. 3 (6) 569–572.

5. S.H. Alsamhi, M.S. Ansari, O. Ma, F. Almalki, S.K. Gupta (2018) pp. 1–8.

6. A. Al-Hourani, S. Kandeepan,(2013) Cognitive Relay Nodes for airborne LTE emergency networks, in: 7th International Conference on Signal Processing and Communication Systems (ICSPCS), Carrara, Australia, pp. 1–9.

7. Pearlmutter, A.B.: (1995) Gradient calculations for dynamic recurrent neural networks: survey. IEEE Trans. Neural Netw. 6(5), 1212– 1228

8. Yan, Xin (2009), Linear Regression Analysis: Theory and Computing, World Scientific, pp. 1–2, ISBN 9789812834119,

9. Östlin, E.; Zepernick, H.J.; Suzuki, H. Macrocell path-loss prediction using artificial neural networks. IEEE Trans. Veh. Technol. 2010, 59, 2735–2747.

10. Ayadi, M.; Zineb, A.B.; Tabbane, S. (2017) A UHF path loss model using learning machine for heterogeneous networks. IEEE Trans. Antennas Propag., 65, 3675–3683.

11. Isabona, J.; Srivastava, V.M. (2016) Hybrid neural network approach for predicting signal propagation loss in urban microcells. In Proceedings of the 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 21–23 December; pp. 1–5.

12. Zineb, A.B.; Ayadi, M. A multi-wall and multi-frequency indoor path loss prediction model using artificial neural networks. Arabian J. Sci. Eng. 2016, 41, 987–996.

13. Uccellari, M.; Facchini, F.; Sola, M.; Sirignano, E.; Vitetta, G.M.; (2016) Barbieri, A.; Tondelli, Vietri sul Mare, Italy, 13–16 September; pp. 1–6.
14. Chang, C.C.; Lin, C.J. LIBSVM-A Library for Support Vector Machines. 2003. Available online: http: //www.csie.ntu.edu.tw/cjlin/libsvm/ (accessed on 9 May 2019).

15. Barnoski, M.K., Chen, B.U., Joseph, Thomas R., Lee, J." (1979) Integrated-optic spectrum analyzer" circuits and systems, IEEE Transactions. Vol. 26, Issue No.12 , Dec.

16. Matthew T. Hunter , Achilleas G. Kourtellis, Christopher D. Ziomek, Wasfy B. Mikhael (2010) "Fundamentals of Modern Spectral Analysis" IEEE, 978-1-4244-7959-7/10.

17. Fuentes, Montse. July 1, 2016. "Reproducible Research in JASA." AMSTAT NEWS.

18. "Jupyter." July 21, 2016. Available at http://jupyter.org.

19. Ms. Sorower a literature survey on algorithms for multi-label learning. Oregon State University, Corvallis. 2010 Dec;18.

20. Beel J, Langer S, Genzmehr M, Nürnberger A. (2013) Introducing Docear's research paper recommender system. In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries Jul 22 (pp. 459-460). ACM

21. Bard, Y. Nonlinear Parameter Estimation. New York: Academic Press. (1974).

22. Astrid Schbieder, Gerhard Hommel, Maria Blettner (2010), Linear Regression Analysis.