Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Statistics and Machine Learning
2023 | LIU-IDA/STAT-A--23/033--SE

Machine Learning Clustering and Classification of Network Deployment Scenarios in a Telecom Network setting

Maskininlärningsbaserad klustering och klassificering av nätverksscenarion i ett telekomnätverk

Chayan Shrang Raj

Supervisor : Filip Ekström Kelvinius
Examiner : Jonas Bjermo

External supervisor : Aron Gosch

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Abstract

Cellular network deployment scenarios refer to how cellular networks are implemented and deployed by network operators to provide wireless connectivity to end users. These scenarios can vary based on capacity requirements, type of geographical area, population density, and specific use cases. Radio Access Networks of different generations, such as 4G and 5G, may also have different deployments. Network deployment scenarios cover many aspects, but two major components are Configuration settings and Performance Measures which refer to the network nodes, hardware build-up and software settings, and the end user behavior and connectivity experience in the area covered by the wireless network.

In this master thesis, the aim is to understand how different area types - such as Rural, Suburban, and Urban – affect the cellular network deployment in such areas. A novel framework was developed to label each node (base station) with the area type it is associated with. The framework utilizes spatial analytics on the dataset provided by Ericsson for the LTE nodes working with 4G technology in combination with open-source libraries and datasets such as GeoPy and H3 Kontur population dataset respectively, to create area type labels. The area types are labeled based on the calculated population density served by each node and are considered true labels based on manual sanity checks performed. A supervised machine learning model was used to predict the nodes based on the CM and PM data to understand the strength of the relationship between the features and true labels.

This thesis also includes analysis and insights about characteristic deployment scenarios under different area types. The main goal of this master thesis is to utilize machine learning to uncover the characteristic features of a variety of node groups inherent in a telecom network, which, in the long run, contributes to better service operation and optimization of existing cellular infrastructure. Nodes (base station) are labeled in the data to be able to distinguish their associated area-type. In addition to this clustering is performed to uncover the inherent characteristic behavior groups in the data and compare them against the output from the classification model. Lastly, the investigation was done on the potential impact of node placements such as indoor or outdoor, on the corresponding features.

In conclusion, the study's results showed us that a correlation exists between deployment scenarios and the different areas. There are a few prevalent common denominators between the node groups such as Pathloss and NR Cell Relations that drive the classification model to a better classification metric, F1 score. Clustering of CM and PM data uncovers inherent patterns in different node groups under different area types and provides information about characteristic features of the groups such as CM data displaying two configuration setting clusters, and PM data showing three different user behavior patterns.

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisors Aron Gosch, Georgios Almyras, and Rafal Piotrowski for their thorough support and continuous guidance while performing this master's thesis in collaboration with Ericsson. Also, I would like to thank my manager, Jonny Setzman for giving me this thesis opportunity and helping me with all the administrative work at Ericsson. Moreover, I also had the opportunity to work with Jonas Eriksson A and Roman Zhohov from the Traffic Modelling Analytics team and all the team members, who were also very supportive and gave valuable feedback on my thesis report. It was an edifying experience with such a friendly and supportive team at Ericsson.

I would like to mention and sincerely thank Filip Ekström Kelvinius, my supervisor at Linköping University for helping me with the thesis report and continuous academic guidance in this thesis opportunity. I would like to thank my examiner Jonas Bjermo for providing valuable critical feedback and a direction toward a good thesis work. Also, Thank you Theodor Emanuelsson for opposing my thesis work and offering a different perspective, while also helping me improve the quality and comprehensiveness of the thesis.

Lastly, I am deeply thankful to my family and friends in India and in Sweden for their unfettered support and unconditional love during the thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

BS	Base Station 8
CDMA	Code Division Multiple Access 7
CM	Configuration Management 27
ECDF	Empirical Cumulative Distribution Function 43
eNB	evolved node B 8
Gbps	Gigabytes Per Second 7
GHz	Gigahertz 7
GSM	Global System for Mobile communication 7
LTE	Long Term Evolution 3
ML	Machine Learning 12
MLlib	Machine Learning Library 4
MLP	Multilayer Perceptron 17
NR	New Radio 3
OFDMA	Orthogonal Frequency Division Multiple Access 7
PCA	Principal Component Analysis 38
PM	Performance Measures 28
RAN	Radio Access Network 6
RBS	Radio base station 2
RF	Radio Frequency 7
TDMA	Time Division Multiple Access 7
UE	User Equipment 5

1 Introduction

Cellular networks are a critical part of the telecommunications industry, enabling millions of people to communicate with each other using mobile phones, tablets, and other connected devices. Modern cellular telecom networks consist of both communication hardware apparatus and software tools that create an ecosystem of networks that allows cell phones, laptops, and other mobile devices to connect and perform various operations such as receiving a call or sending a text message. It is a complex system of interconnected Radio base stations and network infrastructure that works together to provide wireless coverage to a specific geographical area [7].

A cellular telecom network is built up by a number of geographically dispersed Radio Base Stations RBS - also referred to as nodes - which in their most classical form consist of a set of directional antennas mounted on a tower with an electronics cabinet at its base. The cabinets are housing compute resources to run a large set of different Software (SW) and Hardware (HW) components with their specific roles to play. The tower sites are typically interconnected by cables to their neighbors and to a core network facilitating connectivity towards the normal internet [55]. The wireless connectivity offered by this system is characterized by its build-up of the coverage "cells" formed by the geographical area into/from which an antenna can radiate and collect radio signals on a specific carrier frequency.

Mobile networks have a finite capacity which means the ability to cater to simultaneous connections. The more people using mobile phones in a certain area, the more capacity is required and this usually means there is a need to better understand user behavior in order to provide appropriate quality of network coverage [11]. Mobile networks should be designed according to the local population requirements and user behavior. With increasing demands and the introduction of 5G, it has become even more important to understand the underlying patterns in cellular network traffic scenarios and customer needs.

In the context of cellular networks, network patterns may refer to network deployment scenarios and user behavior which may or may not be independent of each other. Network deployment patterns may include the placement of base stations, antennas, software settings, and other features. User behavior depends on the characteristic patterns of the users connected to that network which naturally increases the complexity to study those behaviors [52]. Several different network patterns are commonly used in cellular networks, each with its advantages and disadvantages.

The thesis is done in collaboration with Ericsson. The main goal of the master thesis is to utilize machine learning to uncover the characteristic features of a variety of node groups inherent in a telecom network. Spatial information gathered from the nodes is utilized to implement the area-type classification algorithm that tags each node and cell to a specific geographical area such as Rural, Suburban, and Urban. Next, Classification models are created for the automatic assignment of nodes to a specific area type according to their characteristic behavior. The classification model is built using a standard machine learning pipeline that includes data gathering, data cleaning, feature extraction, model training, and evaluation steps. Lastly, Descriptive statistical profiles of prevalent and important features are created and clustering is used to uncover the underlying patterns in the data and their behavior in different network deployment scenarios.

1.1 Motivation

Cellular network providers need to better understand network deployment scenarios and user behaviors to provide reliable and efficient communication services to their customers. Effective deployment of cellular networks involves the strategic placement of base stations and other network infrastructure to provide the best possible coverage to the population and make the system energy efficient while increasing the capacity for the area being served. This includes factors such as population density, topography, and potential sources of interference.

To meet the growing network complexity there is a need to employ more powerful methods such as machine learning to discover characteristic traffic behaviors in a variety of demographic areas. For example, traffic and configurations may differ due to population density in Rural, Suburban, and Urban areas. In addition to this, localized deployment variations, in the vicinity of amenities such as stadiums or universities, may present special scenarios to be explicitly studied.

1.2 Aim

This thesis aims to provide an analytical framework to enable understanding and provide insights into traffic characteristics in a telecom network setting under various geographical features per region. And then utilize machine learning to understand inherent patterns in telecom networks.

1.3 Research questions

The following research questions will be covered in this master thesis:

1. How can spatial information be used to tag nodes and cells in area types?
2. How can machine learning be used to identify groups of nodes with similar characteristic behaviors in a telecom network?
3. What are the most prevalent denominators between the groups in terms of model features and statistics?

1.4 Delimitations

In this thesis, the focus is on LTE/4G (Long Term Evolution) LTE technology because of its mature market instead of NR/5G (New Radio) NR since it is relatively new and the number of observations is both scarce and rapidly evolving in terms of technology and characteristic behavior. Each observation is aggregated over six days and normalized over a thousand users which means it will represent average user behavior. Big Data techniques and tools

have been used in this analysis because of the volume of data at hand, and machine learning is performed using PySpark MLlib (Machine Learning Library) MLlib. The machine learning part is restricted by the models implemented in this library.

2 Background

The growth and development of cellular network connectivity have greatly impacted the way people communicate and access information, making it easier and more convenient for people to stay connected and informed while on the go. Due to ever-increasing and complex customer demands, the telecommunications industries are constantly on a path to developing advanced and reliable mobile network services. The advancements in device types, new network functionalities, and new applications require a broader range of traffic and mobility scenarios to be taken into consideration in the design and testing of networks. In this section, the discussion is mainly about the background knowledge of the concepts pertinent to this thesis work and research questions.

2.1 Cellular Network

A cellular network is a radio network spanning over a certain geographical area where a number of radio antennas are used to emit radio signals forming coverage cells for connectivity of a small part of the area. Each such cell emits radio signals with each cell connected to a fixed location transceiver known as Radio base station [3]. Covering a wide area allows User Equipment (UE) UE such as mobile phones, laptops and I-Pads to communicate even if the equipment is in motion during transmission.

The design approach for early mobile radio systems was to use a single, high-powered transmitter mounted on a tall tower to cover as much area as possible and points of interest (hot-spots) with high user density like stadiums, hospitals, office spaces, and more. But since then the improvements in both hardware infrastructure and software tools have improved the capability of cellular network systems to serve thousands of users simultaneously in a relatively small area [56]. A simplified diagram for understanding cellular network functioning is shown in Figure 2.1

Components of a cellular network

In this section, a study about the background of various components of cellular networks is given to understand the most critical concepts of the thesis and build a foundation for later chapters. Leading forward with the latest developments, today, the base stations use cellular technology where a base station covers geographical areas by dividing them into

regions/sectors. A high-power transmitter is split into different regions to increase the subscriber capacity and cover a large geographical area effectively.

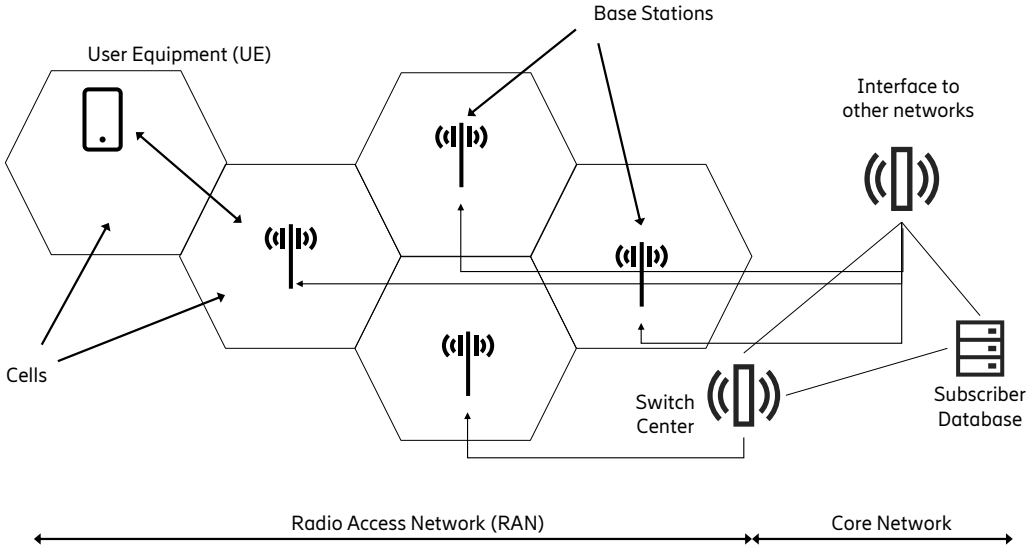


Figure 2.1: Simplified cellular network architecture. This architecture shows how different telecommunications components interact with each other and create an ecosystem that provides wireless connectivity to the users.

Radio Access Network (RAN)

Radio Access Network (RAN) RAN is the fundamental method for devices in a mobile network to connect to each other and exchange information. It acts as a carrier between the core network and user equipment through radio connectivity. RAN manages all the radio resources required for a successful connection between devices. It consists of several base stations that provide over-the-air connectivity to the nearby area [46]. There exist different kinds of RAN architectures based on various telecom technologies and application areas. Today, cellular network systems are in the process of transitioning from LTE with 4G towards NR 5G. Three important RAN technologies are discussed below.

3rd Generation Partnership Project: 3GPP

The inception of 3GPP was realized to provide a complete standardized system description for mobile technologies [40]. The initial scope of 3GPP was to overlook all the technical specifications and reports for a 3G mobile system but the scope was amended to include the development of more evolved technologies beyond 3G such as 4G, 5G, 4G/5G, and in the future 6G. The technical specifications group in 3GPP includes Radio Access Network (RAN), Services & Systems Aspects (SA), and Core Network & Terminals (CT). This encompasses all the steps for the development and implementation of new mobile technologies like 4G and 5G.

LTE (Long Term Evolution): 4G

Long Term Evolution was the next step from 3G towards 4th generation radio technologies in the last decade to enhance the capacity and speed of mobile communications. It was built

on top of the legacy offering of the 3GPP project where LTE uses a different air interface and packet structure than the previous 3G systems. The air interface is based on Orthogonal Frequency Division Multiple Access (OFDMA) OFDMA and is different from previous technologies such as Time Division Multiple Access (TDMA) TDMA used in Global System for Mobile communication (GSM) GSM and Code Division Multiple Access (CDMA) [33] CDMA.

A key concept in LTE technology is the application of the multiple antenna techniques-also referred to as MIMO and it is an advantageous element because of the use of spatial diversity of radio channels. MIMO requires multiple transmitters and receivers to exchange information through different channels simultaneously [20]. In LTE, Downlink (Download rate) and Uplink (Upload rate) are based on multiple access technologies: specifically, OFDMA for the downlink, and single-carrier frequency division multiple access (SC-FDMA) for the uplink. These technologies are based on transmitting data packets in parallel rather than transmitting a high-rate stream of data with a single carrier.

NR (New Radio): 5G

There is a huge leap from the "business-as-usual" structure of 4G to targeting new services and business models in 5G with higher spectral efficiencies, improved energy efficiency, and the introduction of millimeter wave (mm-wave) technology. Devices in the 5G networks should be capable of operating in multiple spectrum bands, ranging from radio frequency (RF) RF to mm-wave [2]. Departing from conventional 3G/4G infrastructure where radios work at less than or equal 5GHz, mm-wave 5G radios communicate at much higher (28 GHz and 38 GHz) GHz spectrum bands.

FD (Full Duplex) transmission is used in 5G to significantly enhance data rates. FD transceiver is capable of transmitting and receiving on the same frequency at the same time which was not possible earlier [38]. One important change from previous technologies is the use of a multi-tier dense heterogeneous network consisting of macro cells combined with a large number of low-power nodes. It has been shown in experiments that in urban settings, the 5G peak data rate could reach as much as 20 Gbps Gbps -almost 25 times improvement from 4G LTE network [27]. 4G and 5G systems used in the RAN ecosystem are shown in figure 2.2

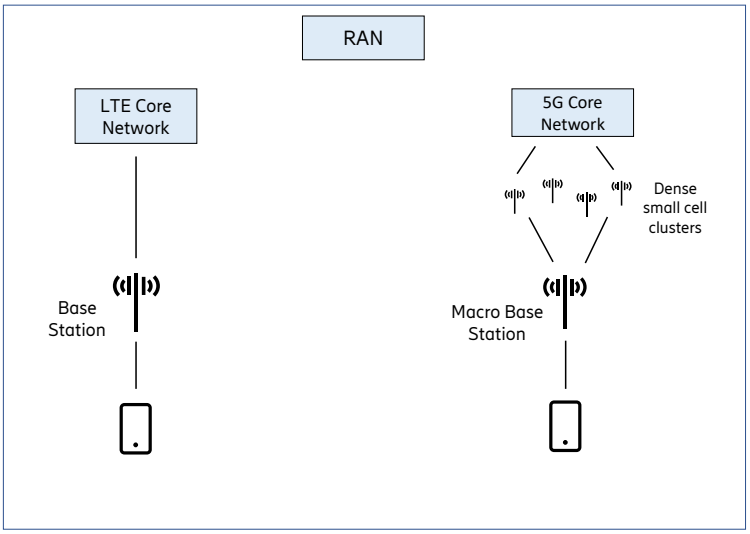


Figure 2.2: RAN ecosystem. In this ecosystem, 4G and 5G are displayed. As it can be seen, 5G contains a more heterogeneous design whereas 4G contains a simplified RBS and link to the users.

User Equipment (UE)

Any device equipped with a radio transceiver that can transmit and receive low-powered radio signals to receive information over the air using the functionalities offered by a RAN is called UE (Mobile Phones, laptops, I-Pads, etc). A beam is formed while communicating with a base station and there should be a plain line of sight for uninterrupted mobile service.

Base Station

The base station is a hardware system that is situated at a specific geographical location to provide the surrounding area with physical wireless network connectivity to mobile devices. It serves as both transmitter and receiver that uses radio signals to transfer information. It is composed of several antennas mounted on a tower with system components installed at the base. The software component of the base station contains the configuration management settings provided by the mobile service provider and is responsible for optimizing call connections at high speeds. Base station BS is also referred to as node B in 3G networks. For LTE standard, it is often called eNB eNB for evolved node B, and gNodeB for 5G [37]. These days, there exist indoor base stations for better coverage in the area where outdoor stations lack clear LoS (Line of Sight). The base station antenna is mounted on tall towers because, from this high point, it is easier to stay in communication with cellphone users, who are often near the ground [19].

Cells

A cell is the logical concept/entity that offers connectivity service to a UE and hosts it while it stays connected. The cell is in effect a geographical area combined with a radio carrier frequency. The geographical area is formed as the area onto which a specific base station antenna is directed and can transmit and receive radio signals from, with enough signal strength. The radio carrier frequency is the base frequency used to transmit and receive the radio signals and is what the UE tunes its transmitter and receiver chains to use [18]. A certain geographical area can thus be served by more than one cell if they employ the same antenna but different radio carrier frequencies. I.e., the cells are overlapping.

The performance of an LTE eNB depends on its radio resource management algorithm and its implementation. Cell coverage is dependent on many factors such as localization of nodes and antennas, transmitting power, signal throughput, etc. Geographical cell areas as formed by base stations are shown in figure 2.3.

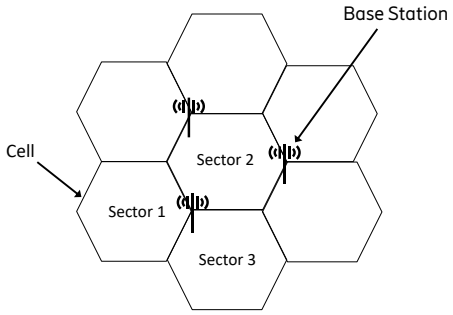


Figure 2.3: Simplified Cell Structure. Cell regions are created by antennas installed on BSs. One BS creates three sectors, each creating an angle of 120 degrees, totaling 360 degrees.

Pathloss refers to the decrease in signal strength as it propagates through space. Cell range is based on the average received power at the cell boundary, where the average is computed based on the pathloss of the signal. This produces a power rating called the transmit power at the base station which is important while considering call handover to the User Equipment. Other non-user-defined parameters such as hills, tunnels, foliage, and buildings greatly affect the signal strength and hence the overall coverage of these cells. Service providers can use different antenna settings for different environments such as universities, stadiums, or motorways [34]. There could be various cell types used under various network deployment scenarios and are described briefly in figure 2.4

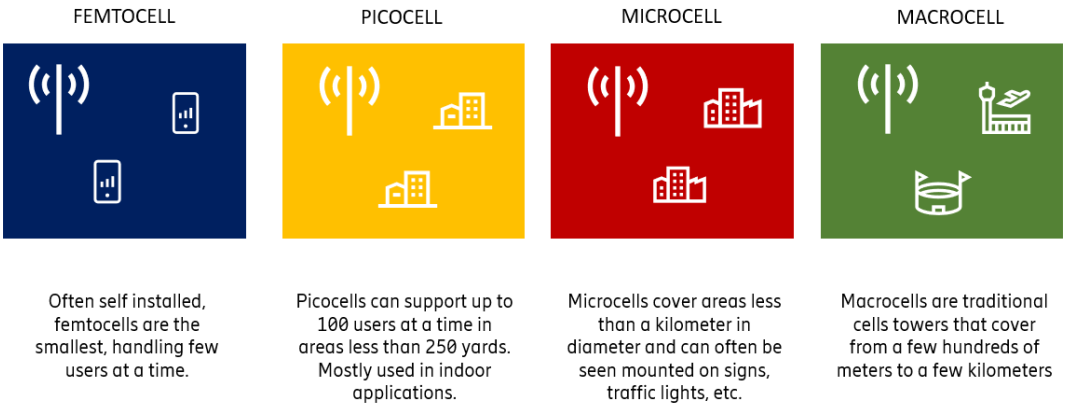


Figure 2.4: Types of cells configured by network service providers. Each cell serves a unique purpose in different deployment scenarios.

Core Network

The core network sometimes also referred to as a backbone network is responsible for routing telephone calls and providing internet connectivity to public devices in the area. It allows high-speed connection capability between the interlinked nodes and public traffic with the help of switches and routers, with more attention to the former component. Furthermore, in 4G (LTE), core networks are referred to as evolved packet cores (EPC). The core network provides a number of functionalities for a smooth interconnection between networks such as Aggregation, Authentication, Call control/switching, and charging. which is outside the scope of this master thesis [12].

2.2 H3 Hierarchical Spatial Index

The antennas are powered by the electronics component at the base station, and these antennas then emit radio signals in certain physical geographical cells covering some area. This area is important in cellular network analysis to accurately define field parameters or other network settings. Various studies have been done to choose an appropriate representation for cells from which hexagons are chosen as the de-facto shape of the cells since it represents the radio signals in an efficient manner and greatly simplify geographic information analysis [45].

Unlike circles, Hexagons do not overlap and cover the area entirely which allows the network to process frequency reuse efficiently, meaning, the frequency band is fully propagated to the other cell without any void, an important concept in cellular networks. Compared with other shapes such as rectangles or triangles, hexagons have only one distance between a hexagon center point and its neighbors, compared to two distances for squares or three distances for triangles [6]. As depicted in Figure 2.5, this property simplifies performing analysis.

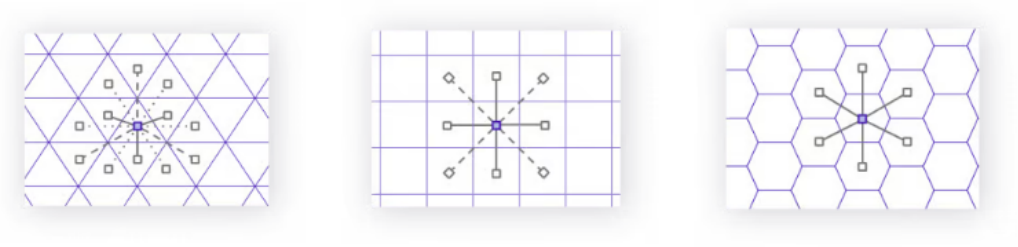


Figure 2.5: Choice of Hexagons. The difference in distance between triangles and squares.

In this master thesis, there is no 3 Dimensional spatial analysis so, the Figure 2.6 visualizes the globe segmented into hexagon shapes, and then in Figure 2.7, the 3D space is converted to 2D geographical regions.

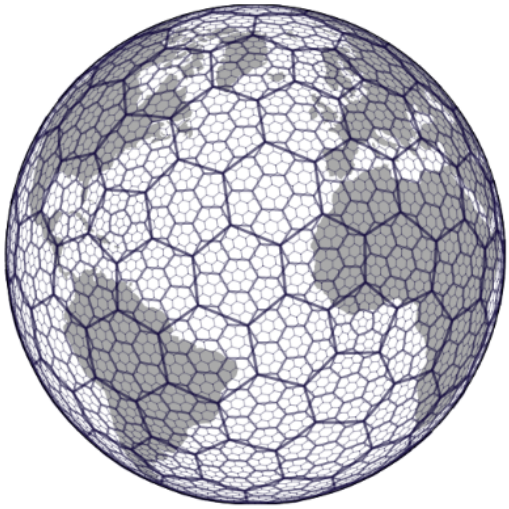


Figure 2.6: H3 divides the globe into hexagons

To represent these hexagons in the real world, there is a need for global grid systems to project cells on a geographical area. Two key components of a global grid system are: a map projection and a grid laid on top of the map [6]. A map projection allows us to visualize a three-dimensional location on Earth to a two-dimensional point on a map. A grid is then overlaid on the map, forming a global grid system. A visual representation of a hexagonal globe is shown in figure 2.7.

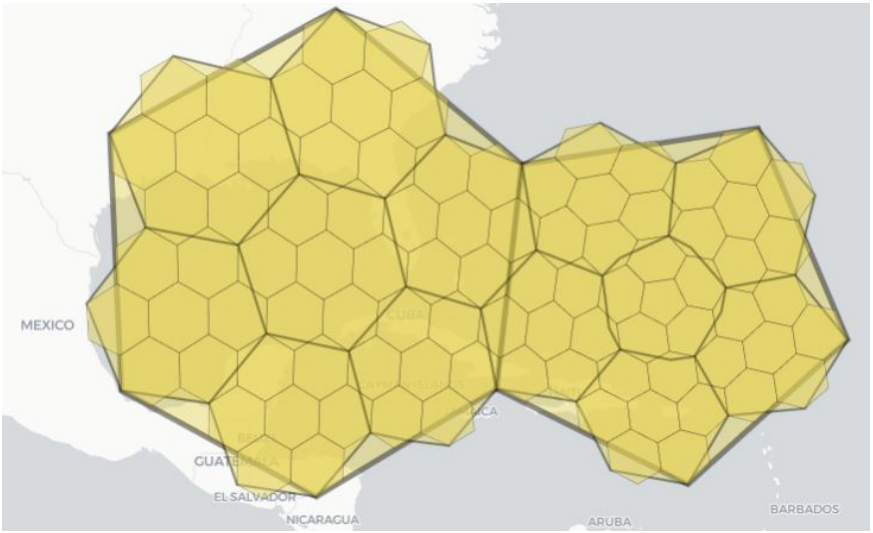


Figure 2.7: H3 hexagonal grid system. The 3-dimensional space is overlaid as 2-dimensional space over geographical regions.

H3 uses a hierarchical approach to divide the Earth's surface into hexagons of varying sizes and resolutions, with each hexagon identified by a unique index value. Resolutions can be used to obtain different levels of granularity, allowing users to efficiently perform spatial analysis at various scales [6]. H3 was developed by Uber Engineering and is available as an open-source library for use in a variety of applications, including mapping, routing, and

geospatial analysis. Some statistics for the sizes of the hexagon cells at different resolution levels are given in Table 2.1

Resolution	Average Hexagon Area (km^2)
0	4,357,449.416078383
1	609,788.441794133
2	86,801.780398997
3	12,393.434655088
4	1,770.347654491
5	252.903858182
6	36.129062164
7	5.161293360
8	0.737327598
9	0.105332513
10	0.015047502
11	0.002149643
12	0.000307092
13	0.000043870
14	0.000006267
15	0.000000895

Table 2.1: Hexagon Areas by H3 Resolution Level. According to the problem statement, it may be convenient to use different resolution levels. In this master thesis, resolution 8 is chosen.

Resolution 8 was chosen for the analysis because it conforms to the default resolution 8 provided in kontur population data described in section 3.1. The area of 0.7 km^2 is the average area of each hexagon and is used in the analysis to calculate the population density of a cell per km^2, which is easier for analysis and density calculations.

2.3 Machine Learning Algorithms

Machine learning (ML) ML is a field within computer science that encompasses mathematics and statistics. It allows real-world phenomena to be modeled as statistical models. The purpose of a model in machine learning is to simulate real-world processes as accurately as possible and solve complex problems. In machine learning there exists a number of techniques such as supervised, semi-supervised, and unsupervised. For this project, the use of supervised and unsupervised learning was done.

Input vector, referred to as a set of features or attributes that are used as input to a machine learning model. The true label refers to the known or correct output or target value associated with each input instance in the training dataset. It represents the ground truth or the desired outcome that the model aims to learn and predict.

In supervised learning, the input vector is mapped along with its true label and is fed into the machine learning model, and unsupervised learning can be used to discover patterns from input data without any labels [31]. The algorithms described in this section are based on the Python implementation available in the ML framework for PySpark called MLlib (Machine Learning Library), which is a library that can be used for various processes such as data pre-processing, train a number of available machine learning models, model evaluation and also deal with big data for Machine Learning.

Supervised learning

Supervised learning is the most widely used method in machine learning for classification, prediction, and regression problems. It takes a set of input features X and learns a function

that maps input predictors to true labels which is a finite set of potential outcomes Y. There are specific algorithms that are suited for supervised learning that use labeled data and "supervise" models into classifying or predicting outcomes. Regression encapsulates several different methods where the output is continuous, while the output of the classification model is discrete.

Decision Tree

Even though the Decision Tree is not used in this thesis project but in order to understand the Random Forests algorithm described in section 2.3, it is important to grasp the underlying concept of the Decision Tree because Random Forests, in its simplest form, is basically an ensemble of many Decision Trees. A Decision Tree is a recursive and hierarchical model composed of decision rules that are applied to partition the feature space of dataset into pure, single subspace. For the following discussion an m x n feature matrix $\mathbf{X}$ contains m samples described by n features. The m x 1 vector $\mathbf{y}$ contains the corresponding categorical true labels for the samples in X.

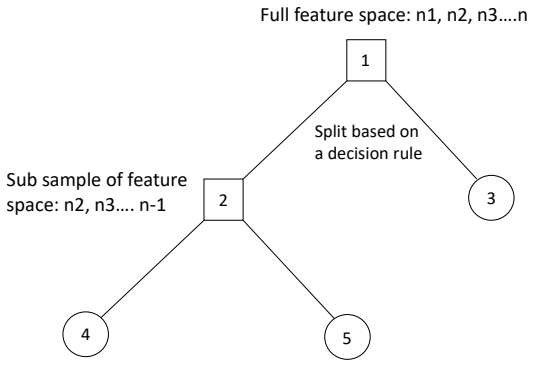


Figure 2.8: Decision Trees structure. Here nodes are shown as squares and leaves are denoted by circles. The branch structure is the line connecting nodes and leaves.

Decision rules are discriminant functions that are applied recursively at the nodes that divide the feature space into subspaces and aim to minimize classification errors while creating class decision boundaries. Decision trees have a few types of nodes such as root nodes, branch nodes, and leaf nodes. In Figure 2.8, the root node is represented by square 1 which represents the entire feature space, i.e., the best split of the node is based on any one feature, which is calculated using a decision rule. Node 2 represents the branch node and Nodes 3,4 and 5 are known as leaf nodes. Each branch node acts as a parent to two children (leaves) nodes. A decision pathway is represented by a line connecting a parent with their children.

Several feature selection criteria have been developed to evaluate decision tree partitions [30]. In PySpark, the default splitting criteria is the Gini Index, which is a measure of impurity or heterogeneity in a node. It indicated how well a particular feature splits the data based on the class labels. Firstly, the probability (p) of each class label C_i in the current node is calculated by dividing the instances with that class label by the total number of instances in the node as shown in 2.1:

$$p_i = \frac{\text{count of instances with class } C_i}{\text{total number of instances}} \tag{2.1}$$

Lastly, Gini Index is calculated that was employed by Breiman et al. [9] and the formula is described in equation 2.2:

$$Gini(\text{node}) = 1 - \sum_{i=1}^{C} p_i^2 \tag{2.2}$$

Gini Index ranges from 0 to 1, where 0 represents the pure node and 1 represents a completely impure node. The Gini index for a split is calculated as a weighted sum of the Gini indices of the resulting child nodes. The lower the Gini index after the split, the better the split is considered to be in terms of reducing impurity and separating the classes. The decision tree algorithm aims to find the feature and split point (feature value) that minimize the Gini index, leading to the best separation of classes and the most homogeneous child nodes.

Random Forests

Random Forests is a type of supervised machine learning algorithm that uses a tree-based classification technique, which was developed by Breiman in 2001 [8]. A Random Forest is a collection of tree predictors $h(x;\theta_t), t = 1,\ldots,T$; where θ_t... i.i.d. are random vectors, T is the number of Decision Trees, x is the feature space. Randomness is injected into the training process by selecting a random subset of the features to be split at each node, and selection of the random subset of observations, meaning that each resulting tree will be somewhat different from one another. The fundamental concept behind the success of random forests is the Law of Large Numbers. The individual variance is reduced by averaging over the randomness-injected ensemble. As $t \to \infty$, the Law of Large Numbers ensures

$$E(X,Y) = [(Y - \bar{h}(X))^2] \leqslant E(X,Y)[(Y - E(\theta)h(X;\theta))^2] \tag{2.3}$$

where, Y is the true label, $\bar{h}(X))^2$ is the average prediction of the ensemble of Decision Trees and $E(X,Y)$ is the average prediction error of Random Forests.

The individual trees are grown to large/maximal depth, which makes a major departure from the traditional CART (Classification and Regression Trees) paradigm. The growth of individual trees also somewhat affects the bias-variance trade-off where maximal tree depth minimizes bias but potentially increases prediction variance [17]. The averaging over the ensemble tapers off the bias-variance trade-off and provides better inference. It can work with both categorical and numerical variables so there is no need to one-hot encode categorical variables unlike in other ML algorithms. For regression analysis, it takes the average of all the votes from individual trees and predicts the value. For the classification problem, Random Forest receives a class vote from each tree and then makes a final decision based on the majority vote. The algorithm has the following steps:

1. Randomly select k attributes from total m attributes where k < m, the default value of k is usually taken as $\sqrt{m}$.
2. Randomly select subset p number of observations out of the total number of n observations.
3. Construct individual decision trees for each sample.
4. Every decision tree will produce an output, a class if it's a classification problem or a number in case of regression.
5. Final output is calculated based on Majority voting or Averaging, depending on if it's a classification or regression problem, respectively.

Hyper-parameter tuning is an important step when it comes to selecting the best-performing model and Random Forests contains a lot of hyper-parameters that could be tuned to optimize the model's performance. These parameters are set by the user before training and are model specific. The random forest offers a number of hyper-parameters. Some of them are the number of decision trees in the forest, The maximum depth of each decision tree in the forest, The number of features to consider when looking for the best split, and splitting criteria, among others. Cross-validation can be used to configure a good set of hyper-parameters for a specific data set and problem. Philipp Probst et. al. [42] suggest using a large number of trees that can increase the performance along with parameter search for maximum depth of trees, subset selection, and random samples that affect the randomness in the Random Forest (bias-variance trade-off).

Random Forests also provide a feature (variable) importance tool, which can rank the features according to their importance in driving the predictions. The importance measure for variable X_j is determined by the sum of all instances of node impurity reduction where variable X_j is used to partition the data. Let t_L and t_R denote the two children nodes when partitioning the data at node t, and let N_t denote the number of samples making it to node t, the decrease in impurity is defined as:

$$\Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R) \tag{2.4}$$

where i(·) is the Gini impurity measure described in equation 2.2, and $p_L = \frac{N_{tL}}{N_t}, \quad p_R = \frac{N_{tR}}{N_R}$

Lastly, the children nodes are obtained when the data is partitioned by the parent node at s = (X_m < c). Lastly, the total measure for a feature X is defined by averaging over all trees T and all nodes t,

$$VI(X_m) = \frac{1}{n_{\text{tree}}} \sum_{X_t \in T : v(s_t) = X_m} p(t) \Delta i(s_t, t) \tag{2.5}$$

where VI denotes the variable importance, p(t) denotes the proportion $\frac{Nt}{N}$ of samples reaching node t and v(s_t) denotes the variable used to split node t.

Multinomial logistic regression

Logistic Regression is a machine learning algorithm that is used for classification problems. The name could be confusing as it works on regression outcome but represents it as a classification using either the sigmoid or softmax function. The model was developed to answer the need of modeling posterior probabilities of the K classes via linear functions in x, while at the same time making sure the function output sums to one and remain in [0,1]. The model has the form:

$$\log\left(\frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)}\right) = \beta_{10} + \beta_{T1}x \tag{2.6}$$

$$\log\left(\frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)}\right) = \beta_{20} + \beta_{T2}x$$

$$\vdots$$

$$\log\left(\frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)}\right) = \beta_{(K-1)0} + \beta_{T(K-1)x} \tag{2.7}$$

The algorithm is modeled in terms of K-1 log-odds or logit transformations. Using algebra, a simple calculation shows that:

$$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + \beta_{Tk}x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{T\ell}x)}, \quad k = 1, \ldots, K-1 \tag{2.8}$$

$$\Pr(G = K-1|X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{T\ell}x)} \tag{2.9}$$

where G denotes a placeholder for any of the K classes and the equation 2.9 clearly sums to one. To point out the dependence on the entire parameter set $\theta = \{\beta_{10}, \beta_{T1}, \ldots, \beta_{(K-1)0}, \beta_{T(K-1)}\}$, the probabilities are denoted as $\Pr(G = k|X = x) = p_k(x; \theta)$. For binary logistic regression, the model is used for two dependent variables e.g. cat or dog, and for Multinomial logistic regression, it can have more than two dependent classes. In logistic regression, modeling of the probability of X belonging to a class, rather than the response Y itself is done [25].

The logistic regression model is fit by maximum likelihood, using the conditional likelihood of G given X. The log-likelihood of N observations can be written as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log p_k(x_i; \theta) \tag{2.10}$$

- $L(\theta)$ is the log-likelihood function.
- N is the number of observations in the dataset.
- K is the number of outcome categories.
- y_{ik} is an indicator variable that takes the value 1 if observation i belongs to category k and 0 otherwise.
- $p_k(x_i; \theta)$ is the predicted probability of observation i belonging to category k given the predictor variables x_i and model parameters θ.

To maximize the log-likelihood, the derivatives are set to zero:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i; \beta)) = 0 \tag{2.11}$$

which are $p + 1$ equations nonlinear in β and β is the concerned coefficient vector.

To solve the above equation 2.11, there are many optimization algorithms that could be utilized but a common method is Newton–Raphson algorithm [26], which requires the second-derivative or Hessian matrix:

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i; \beta)(1 - p(x_i; \beta)) \tag{2.12}$$

Starting with β^{old}, a single Newton update is:

$$\beta_{\text{new}} = \beta_{\text{old}} - \mu \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta} \tag{2.13}$$

where the derivatives are evaluated at β^{old}.

It is easier to write the update and Hessian in matrix notation. Let $\mathbf{y}$ denote the vector of y_i values, $\mathbf{X}$ the $N(p + 1)$ matrix of x_i values, $\mathbf{p}$ the vector of fitted probabilities with ith element $p(x_i; \beta^{\text{old}})$ and $\mathbf{W}$ a N x N diagonal matrix of weights with ith diagonal element $p(x_i; \beta^{\text{old}})(1p(x_i; \beta^{\text{old}}))$. Then

$$\frac{\partial \ell(\beta)}{\partial \beta} = X^T(y - p) \tag{2.14}$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -X^T W X \tag{2.15}$$

The Newton step is thus:

$$\beta_{\text{new}} = \beta_{\text{old}} + (X^T W X)^{-1} X^T (y - p)$$

$$= (X^T W X)^{-1} X^T W (X\beta_{\text{old}} + W^{-1}(y - p)) \tag{2.16}$$

$$= (X^T W X)^{-1} X^T W z \tag{2.17}$$

In the equation 2.16 and 2.17, the Newton step is expressed as a weighted least squares step, with the response

$$z = X\beta_{\text{old}} + W^{-1}(y - p) \tag{2.18}$$

sometimes known as an adjusted response. These equations get solved using iterative optimization since at each step **p**, **W** and **z** gets updated.

$$\beta_{\text{new}} \leftarrow \arg\min_{\beta} (z - X\beta)^T W (z - X\beta) \tag{2.19}$$

This way, the algorithm does converge and the updated coefficient vector is obtained.

Multilayer perceptron Classifier

Multilayer perceptron MLP or feedforward neural networks are the quintessential deep learning models that form the basis for all the advanced architecture in use today. The term neural in these networks is inspired by neuroscience. The goal of artificial neural networks is to approximate an arbitrary function f [21]. A multilayer perceptron (MLP) is a type of artificial neural network (ANN) that is commonly used for supervised regression/classification tasks. It consists of multiple layers of interconnected artificial neurons, where each neuron applies a nonlinear activation function to its input. The MLP is trained using a supervised learning algorithm, such as backpropagation, to optimize its weights and biases for accurate classification.

There are several steps involved in building a neural network architecture and training it on a dataset for a specific task. The general steps involved in MLP implementation are described as follows [17]:

1. Architecture
    - The MLP consists of different layers such as an input layer, one or more hidden layers, and an output layer.
    - The input layer receives the feature vectors or input data.
    - Each hidden layer contains multiple neurons that perform computations on the inputs.
    - The output layer produces the predicted class probabilities or labels.
2. Forward Propagation
    - The input data is passed through the network using forward propagation.

- For the l-th layer, the weighted sum of inputs is calculated as:

$$z_l = W_l \cdot a_{l-1} + b_l \tag{2.20}$$

where W denotes the input weight matrix initialized, a is the activated input after application of the activation function, b represents the bias vector and l is the layer number.

- The activation function introduces non-linearity into the network, allowing it to learn complex patterns. The activation function applied to the weighted sum is denoted as:

$$a_l = g(z_l) \tag{2.21}$$

where $g()$ is the activation function for layer l.

3. Activation Functions

- The activation function transforms the neuron's input into its output. Common activation functions used in MLPs include the sigmoid function, ReLU (Rectified Linear Unit), and softmax function.
- Sigmoid function (commonly used in hidden layers):

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2.22}$$

where is the exponential function and z represents the weighted sum of inputs.

- ReLU (Rectified Linear Unit) function (commonly used in hidden layers):

$$g(z) = max(0, z) \tag{2.23}$$

where max is the maximum function.

- Softmax function (used in the output layer for multi-class classification):

$$g(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.24}$$

where z_j refers to the weighted sum of inputs for the j-th neuron in the output layer, z_k refers to the weighted sum of inputs for the k-th neuron in the output layer and K represents the total number of neurons (classes) in the output layer.

4. Loss function

- The loss function in MLP is used to quantify the discrepancy between the predicted output of the model and the true output (labels) for a given set of input data [1]. Common loss functions for classification tasks include cross-entropy loss.
- Cross-Entropy Loss (for multi-class classification):

$$L = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik}) \tag{2.25}$$

where N is the number of samples, K is the number of classes, y_{ik} is the true label (one-hot encoded) for sample i and class k, and $\hat{y}_{ik}$ is the predicted probability for sample i and class k.

5. Backpropagation

- It involves computing the gradients of the loss function with respect to the network's weights and biases and then updating these parameters to minimize the loss.
- At each layer, the gradients are computed by multiplying the gradients from the subsequent layer with the derivatives of the activation function and the weights connecting the layers:

$$\delta_l = ((W_{l+1}^{\top}\delta_{l+1}) \odot g'(z_l) \tag{2.26}$$

where δ represents the error (or gradient) at the l-th layer, $W_l + 1$ is the weight matrix connecting the (l+1)-th layer to the l-th layer, δ_{l+1} denotes the error at the (l+1)-th layer, $\odot$ represents element-wise multiplication, $g'(z_l)$ is the derivative of the activation function applied to the weighted sum z_l at the l-th layer.

- After forward propagation, the predicted output is compared to the true output (labels) to calculate the error. The error is then propagated back through the network to update the weights and biases using the backpropagation algorithm. The error (or gradient) at the output layer is given by:

$$\delta_L = \nabla_a L \odot g'(z_L) \tag{2.27}$$

where $\nabla_a L$ is the gradient of the loss with respect to the activation of the output layer, and $\odot$ denotes element-wise multiplication.

- The gradients of the biases and weights are computed as:

$$\frac{\partial b_l}{\partial L} = \delta_l$$

$$\frac{\partial W_l}{\partial L} = \delta_l \cdot (a_{l-1})^{\top}$$

6. Weight update

- The MLP is trained by iteratively adjusting the weights and biases using gradient descent optimization. The process continues until the model converges or a stopping criterion is met.
- Gradient Descent (update the weights and biases in the opposite direction of the gradients):

$$W_l \leftarrow W_l - \eta \cdot \frac{\partial W_l}{\partial L}$$

$$b_l \leftarrow b_l - \eta \cdot \frac{\partial b_l}{\partial L}$$

7. Prediction - Once the MLP is trained, it can be used to make predictions on new, unseen data. Forward propagation is performed on the input data, and the output layer provides the predicted class probabilities or labels.

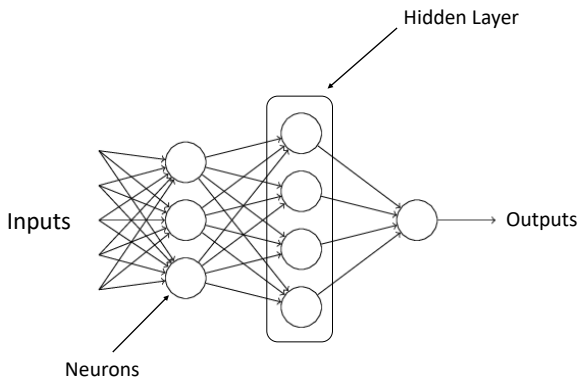


Figure 2.9: Multilayer Perceptron: Skeleton showing the major parts of the neural network architecture

Performance Metrics

The purpose of performance metrics in the context of machine learning is to quantify the ability of an algorithm to learn from the data provided for a specific problem. Different problems call for different performance metrics such as classification tasks that employ metrics like F1-score, Precision, and recall whereas regression tasks make use of MSE, RMSE, etc. Performance metrics is an important tool in assessing the quality of predictions made by machine learning models and comparing different algorithms [28].

The confusion matrix aids in evaluating performance metrics for classification tasks which is a table that contains the model's True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) predictions. In this master thesis, 4 major performance metrics are used to evaluate the classification models: Accuracy, F1-score, Precision, and Recall.

- Accuracy: It determines the model's overall correctness, keeping all the measures into account.
$$\frac{TP}{TP + FP + FN + TN} \tag{2.28}$$
- Precision: It calculates the proportion of correctly predicted positive observations out of all positive predicted observations.
$$\frac{TP}{TP + FN} \tag{2.29}$$
- Recall: It calculates the proportion of correctly predicted positive observations out of all actual positive observations
$$\frac{TP}{TP + TN} \tag{2.30}$$
- F1-score: It calculates a balanced measure that combines both Precision and Recall into a single value. It is useful when there is an unbalanced distribution of classes.
$$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.31}$$

Unsupervised learning

Unsupervised learning, as the name suggests is a type of learning that is not supervised by true labels and is commonly used in pattern mining, clustering, and data groupings. The goal is to directly infer the properties of the probability densities of the features without any supervision. For the purpose of this master thesis, it is appropriate to stick to a clustering problem where the algorithm groups similar data points based on a specific criterion or cost function [16].

K-means

K-means belongs to the family of unsupervised learning techniques where p data points are automatically assigned to k different clusters based on a distance measure. There are various distance measures such as Manning distance, Minkowski Distance and Hamming Distance that are used for different machine learning problems and specific algorithms [41]. A cluster comprises a group of observations (points) created by choosing an arbitrary center point as one of the observations and evaluating the Sum of Square Error (SSE) from the Euclidean distances to all other observations within the cluster. Assigning each point x_i to the nearest centroid is done based on the formula 2.32:

$$\arg\min_{c_j \in C} ||x_i - c_j||^2 \tag{2.32}$$

where | | x - y | | denotes the Euclidean distance between vectors x and y.

Since distances are sensitive to noise (outliers) present in the data or having features in different scales (meters or kilometers), this makes it important for all features to be in the same range (scaling) [53]. It is interesting to note that there is no previous knowledge of the optimum number of clusters that should be created for any specific problem or data set. In order to find the optimal number of clusters, there are several methods that could be used to perform.

One of them is the elbow method, where the clusters are optimized using the distance between the points in a cluster. Closer the observations are to each other within the cluster, the better the cluster. The most commonly used distance metric used is the within-cluster sum of squares which denotes the sum of squared distances between each observation and its centroid within a cluster [49]. The results are then plotted on a graph, and the plot is in the shape of an elbow, where the choice of the number of clusters may be subjectively driven because oftentimes, it could be difficult to discern a clear pattern in the graph. In such cases, additional techniques such as silhouette score may be used.

The silhouette score is another technique used to evaluate the quality of clustering in unsupervised machine learning. It is more robust than the elbow method discussed because while calculating within-cluster quality, it also takes into account inter-cluster distances which denotes how well a cluster compares to the other clusters. This method is used in this project to find the quality of the resulting clusters. A higher silhouette score indicates that the points in a cluster are tightly packed together and well-separated from other clusters and vice-versa [17].

2.4 Equivalence test of feature distributions

Sometimes, there is a need to understand the intrinsic distributions of features used in the analysis to comment on the probable outcomes. It is entirely possible to make heuristic judgments about the population and in order to validate those comments, hypothesis testing is an important tool in statistics to evaluate the validity of a claim or hypothesis. Hypothesis testing is a statistical method that is used to discover if there is enough evidence shown by observed data to support or reject a hypothesis about a population based on a sample of data. In

hypothesis testing, two hypotheses are created namely; null and alternate hypothesis, which are basically statements about a population, and then the sample data is used to ascertain the credibility of those statements about the population [43]. Then collection of a sample of data is done and use it to test the null hypothesis.

Start by stating the null hypothesis, which is a statement that there is no significant difference or relationship between two variables or groups in the population, and an alternative hypothesis which is the opposite of the null hypothesis. A significance level is chosen, which then, is the probability of rejecting the null hypothesis when it is actually true. The most common significance level is 0.05, which means that the null hypothesis will be rejected if the probability of obtaining the observed result by chance is less than 5%. This sets up the design for different hypothesis experiments.

A test is performed by collecting a sample of data to calculate a test statistic that measures the deviation between observed and theoretical values. p-value is defined as the probability of obtaining a test statistic as extreme as or more extreme than the one observed, assuming that the null hypothesis is true. If the p-value is less than the significance level, the null hypothesis is rejected and the conclusion is made that there is enough evidence to support the alternative hypothesis. If the p-value is greater than the significance level, the null hypothesis is failed to be rejected and the conclusion is made that there is not enough evidence to support the alternative hypothesis. Hypothesis testing is a powerful tool for making decisions based on data and is widely used in scientific research, business, and other fields.

2.5 Related Work

In this section, relevant previous work or publications are covered.

Geo-spatial analysis of node groups

One of the most important aspects of this thesis depends on using spatial location information of the cells and nodes to label them according to the cell population density coverage. Node placements have a huge impact on wireless network servicing and the effectiveness of its operations. Younis et al. [57] discuss different deployment scenarios where area coverage optimization strategy plays a big role in the determination of the placement of nodes and corresponding analysis.

Big data analysis of telecom network

With the burgeoning demand for the telecom network capacity, data associated with telecommunications network architecture has '4-V' characteristics, namely 'Volume', 'Variety', 'Velocity', and 'Veracity' [54]. The '4 Vs' abbreviation indicates the presence of 4 pillars, which is commonly known as, big data. They are characters of extremely large data volume, different data formats, high-speed data streaming, and high data value. Telecom service providers can use telecom big data to carry out internal operations and maintenance, as well as external applications in relevant industries [50].

Machine learning on user behavior data in telecom network

Machine learning is widely used in modeling technical problems of next-generation mobile network systems, such as large-scale MIMOs, device-to-device (D2D) networks, heterogeneous networks built by femtocells and small cells, base station analysis, and so on [32]. Donohoo et al. [15] performed similar experiments using five real user profiles, including the user locations and energy consumption, but their data is not accessible to the public. The experiment showed that up to 90 percent successful energy demand prediction is possible with the aid of supervised learning algorithms.

Data mining using clustering in telecom network data

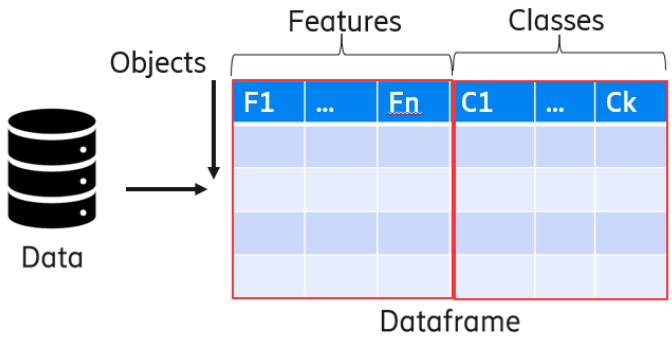
Paying heed to the increasing amount of big data generated in different industries, it has become increasingly important to make sense of the raw database entries. Data mining techniques are applied on very large datasets for finding hidden correlations, patterns, and unexpected trends [4]. The clustering (unsupervised learning) model is a more fitting data mining approach to find and understand characteristic natural groupings within mobile customers, based on their mobile usage behavior [48]. Bernaille et al. [5] applied the k-means clustering algorithm to user traffic clustering and labeled the clusters to applications using a payload analysis tool which created new insights regarding behavior complexity and separation.

3 Method

This chapter describes the method that was used during the thesis work to answer the research questions. Data analysis of LTE eNodeBs is done along with configuration and performance data for all the node groups. Spatial information gathered from the nodes allows the implementation of the area-type classification algorithm that tags each node and cell to a specific geographical area such as Rural, Suburban, and Urban. Classification models are trained for finding relationships between nodes and area types with a certain degree of confidence. Lastly, statistical profiles are created and analyzed for understanding traffic behaviors and use clustering to uncover inherent patterns in the data and their connection to certain network deployment scenarios.

3.1 Data Description

Data sets are in tabular format, containing both categorical and numerical values. Since dealing with Big Data, HDFS is used to store data sets and are loaded in spark using Python implementation as PySpark. The raw data is stored in both *csv* and *parquet* file formats and is supported by a data processing framework.



Node name	Node location	Area Type	PM	CM	Amenties	...
Node_A	(56.723,44.765)	Urban	[PM1,...]	[CM1,...]	[School, Hospital, Stadium]	...
...						

Figure 3.1: Example data skeleton. It contains both categorical and numerical features. Each dataset is in tabular format.

In mobile wireless networking, there are plenty of network measurements and configuration settings that are continuously handled between, and gathered from the user equipment (UE), nodes in the radio access network, and core network (CN) of the telecom infrastructure. There are several data sets used in this master thesis to answer the research questions. Sections 3.1 3.2 3.3 3.4 describe the nature of these data sets. Data flow and corresponding steps taken in this analysis have been depicted in Figure 3.2:

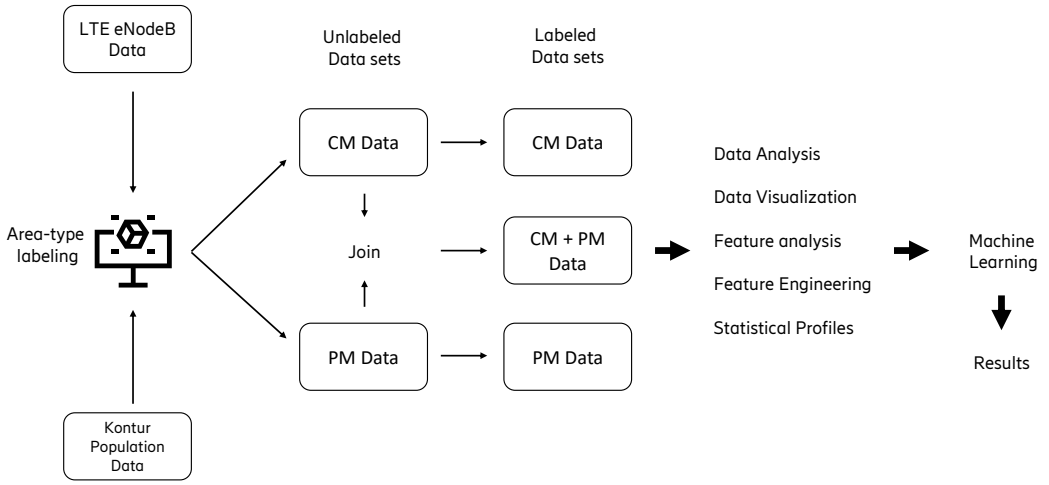


Figure 3.2: Methodology flow chart. LTE and kontur population data are used to obtain area labels. CM and PM datasets are labeled by joining on primary columns. Afterward, Data exploration and machine learning pipeline are formed to answer research questions.

LTE eNodeB data

This is a mobile network provider data set describing provider setting based on different network deployment scenarios. It is loaded in pyspark in a "csv" format. The original data set has around 1 million observations and 11 features. Each node is a base station with at least one cell in that base station. Nodes and cells are uniquely defined for every observation in this data set.

Since, only the North American market is present in the dataset, given the latitude and longitude information, it has been filtered to contain nodes located only in that region. Data description of LTE eNodeB data is given below 3.1 that shows parameter name, feature description, and data type:

Feature	Description	Data Type
eNodeB	Unique identifier of the base station	String
Cell	Cell (antenna) ID attached to the base station	String
Latitude	Latitude information of the cell	Float
Longitude	Longitude information of the cell	Float
Azimuth	Angle made from true North (0 degree)	Float
IndoorOutdoor	Position of the cell; either Indoor or Outdoor	String
operationalState	If the cell is Enabled or Disabled	String
Radio ProductName	Product name of the radio equipment used	String

Table 3.1: LTE eNodeB Data Description

Kontur Population data

H3, a 16-level hexagonal global grid system developed by Uber as described in section 3.1, maps the entire earth with two-dimensional hexagons and assigns indexes to all cells based

on the spatial hierarchy. This creates a medium between the geographical cell concept and performing spatial analysis as shown in figure 3.3. Each level, also known as resolution, denotes the area of the cells ranging from average 0.000000895 km^2 at resolution 15 and 4,357,449.4 km^2 at resolution 0 as shown in table 2.1. USA Kontur population data set consists of unique hexagon ids and the corresponding population density for the USA region [6]. The default resolution for this open-source data set is 400m which translates to resolution 8 in Uber's H3 library. This is the reason, the H3 index with resolution 8 is the optimal choice for this master thesis as used for area-type labeling.

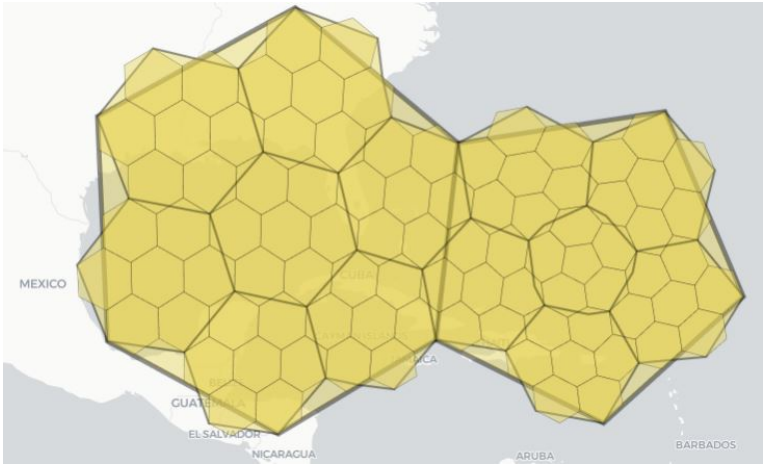


Figure 3.3: Example hexagon layer on real world geography [35]

The data set is loaded in pyspark in csv format and has around 4 million observations and two features representing unique hexagon id and population density in each hexagon as shown in table 3.2.

Feature	Description	Data Type
h3	Unique hexagon id for resolution 8	String
population	Population density inside that cell	Float

Table 3.2: Kontur Population Data Description

Configuration Management (CM) data

Configuration management (CM) CM data describes configuration settings currently in use for those cells/nodes under different network deployment scenarios. The data set has values as numeric categories, meaning, each feature describes the observation in discrete numeric values. The data set is loaded in pyspark in paraquet format and has around 1 million observations and 16 features.

Features	Description	Data Type
eNodeB	Unique identifier of the base station	String
Cell	Cell (antenna) ID attached to the base station	String
Frequencies	Frequency operation of the node	Float
Sector Carriers	Parameter related to sector formed by cells	Float
External eNodeB MOs	External nodes connected to current MO	Float
External LTE Cell MOs	Cells defined in eNB but belong to another eNB	Float
LTE Frequency Relations	Frequency relation to neighbor cells	Float
External LTE Cell Relations	Relations to cells on another node	Float
Internal LTE Cell Relations	Relations to cells on same node	Float
Active X2 to eNodeB	No. of active X2 links per node	Float
External gNodeB MOs	No. of 5g nodes (gNB) connected to LTE nodes	Float
External NR Cell MOs	No. of external 5g (NR Cell) per node	Float
NR Frequency Relations	New Radio frequency relations to gNodeB	Float
NR Cell Relations	Total cell relations in New Radio gNodeBs	Float
Active X2 to gNodeB	No. of active X2 links to gNb per Lte node	Float
IAT	Inter arrival time	Float

Table 3.3: Configuration Management Data Description

Performance Measurment (PM) data

Performance Measurement (PM) PM data describes performance measures of users connected to nodes and the mobile network provider under different network deployment scenarios. The data set is numerical in nature, meaning, each element has numeric values. The data set is loaded in pyspark in paraquet format and has around 0.6 million observations and 19 features.

Features	Description	Data Type
eNodeB	Unique identifier of the base station	String
Cell	Cell ID attached to the base station	String
Paging	Signals sent to users to establish connection	Float
RRCUsersperCell	Average RRC connected users per cell	Float
RRCConnectedUsers	Average RRC connected users per node	Float
EN-DCoverLTE(%)	Percentage of EN-Dc over LTE RRC Users	Float
ActiveRRC(%)	Number of active users connected	Float
ConnectionEstablishments	Established connections to cells	Float
RRCDLThroughput(kbits)	DL Volume over the number of RRC Users	Float
RRCULThroughput(kbits)	UL Volume over the number of RRC Users	Float
UEActive-DLThroughput	Active UEs connected Downlink speed	Float
UEActive-ULThroughput	Active UEs connected Uplink speed	Float
Handovers	Mobility between cells	Float
MeasurementReports	Reports relationship with neighboring cells	Float
CarrierAggregation (%)	Percentage of Volume when cell is used as SCell	Float
Pathloss	Attenuation of radio signal over a distance	Float
SINR	Signal to Interference and Noise Ratio	Float
CQI-SE-64	Channel quality at Modulation 64	Float
CQI-SE-256	Channel quality at Modulation 256	Float

Table 3.4: Performance Measures Data Description

3.2 Area Type Labeling

The first step in this master thesis is to label each cell/node with its area type i.e. Rural, Suburban, or Urban. The aim is to understand the characteristic behavior of different deployment scenarios under different area types. A framework has been built that uses spatial information of the nodes to tag them in the above area types and these labels are assumed to be true labels for the machine learning models.

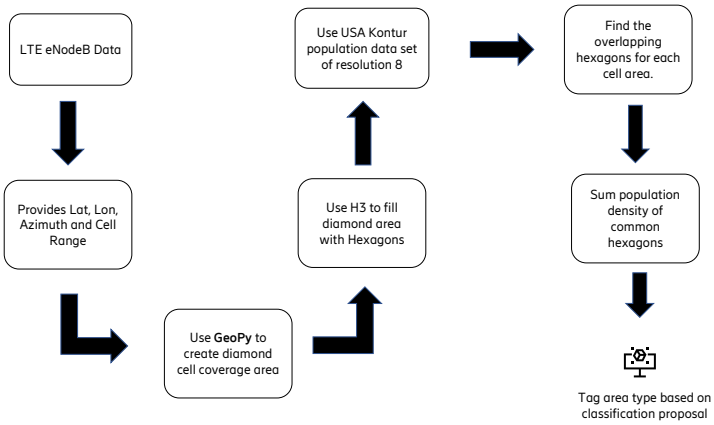


Figure 3.4: Area Type Labeling Flowchart

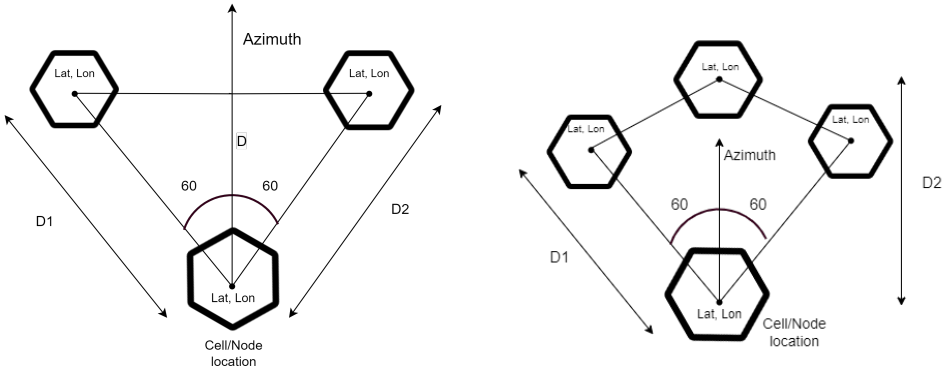
Steps taken to label each node with an area type are given below:

1. LTE eNodeB data is provided that contains the Latitude, Longitude, Azimuth angle, and cell range (described below) of each cell/node.
2. Cell range is calculated using the 'pmTaInit2Distr' parameter which describes the initial distance distribution of UEs camping on the cell. This is the primary feature commonly used to estimate the theoretical cell range. Normalization of this measure is done in percentage and has taken into consideration the area with 80% traffic
3. Using all the above features, a cell population coverage area is formed using the GeoPy library that allows us to do spatial analysis. A diamond shape cell coverage area is formed keeping the node at one of its vertex.
4. Using H3 Hierarchical spatial index, fill the diamond shape with hexagons of resolution 8. As discussed before, hexagons are chosen because cellular cells are represented as hexagons.
5. Using an open USA Kontur population data set, find all the hexagons in each of the Shapes and sum the population density for each cell. Each hexagon has an area of 0.7 Km^2 (average and could vary).
6. Area type classification proposal is provided by Mobile Experts and ETSI and each observation is tagged based on the population density rules given. After sanity checks and expert knowledge at Ericsson, these tags act as true labels.
7. The area type label has been obtained and spatial information are no longer needed so LTE eNodeB and Kontur population data set are removed from further analysis hereafter.

Cell Coverage area calculation

A simpler assumption, the circular-shaped cell could also be chosen and is rather common in literature as a reasonable approximation for an omnidirectional base station. But in this analysis, multiple antennas are forming different cell positions and angles. Two methods were tried to calculate the area covered by each cell/antenna.

Mathematical calculations are performed to find the distance and the angle coverage of each cell which has been described through equations 3.1 to 3.2. GeoPy library has been used to find the distance between any two coordinates given a bearing (angle) between the two in the metric system, i.e., kilometers. Cell range is calculated using a distribution parameter that describes the initial distances of UEs connected to the cell. Normalization of this measure is done in percentage and has taken into consideration the area with 80% traffic. The approach has been visualized below and the motivation for trying the two methods was the exaggerated results (more dense, dense urban area-types) shown by the triangle method in Figure 3.5:



(a) Triangle Approach (b) Rhombus Approach

Figure 3.5: Comparison between triangle and rhombus approach

Triangle area calculation - Since, there are observations with different cell ranges, an example cell range of 5 km is taken to experiment.

Pythagoras' theorem is used to find the distances between points if the angle between the triangle sides is known. A standard 120-degree angle is typically made by antennas that create cell area for mobile networks and this divides the angle in exactly half i.e. 60 degrees.

Let, for example, cell range = 5 km = D

$$\cos 60 = \frac{5}{D1} \tag{3.1}$$

$$D1 = \frac{5}{\cos 60} = \frac{5}{1/2} = 10km \tag{3.2}$$

Similarly,

$$D2 = 10km$$

Rhombus area calculation - In Rhombus, a key difference is that the values of D1 and D2 are equal to the cell range and have been simply plugged into the distance API of the GeoPy library.

H3: Uber's Hexagonal Hierarchical Spatial Index

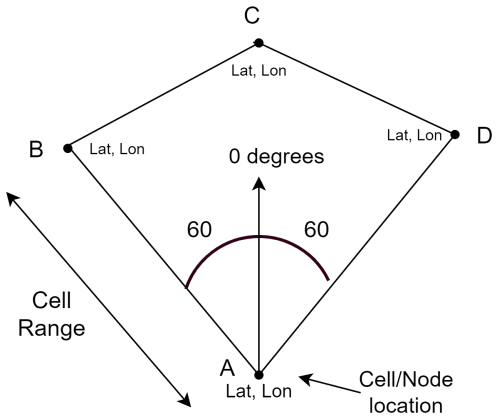


Figure 3.6: Cell coverage area methodology

Steps to implement the area coverage in Python using the GeoPy library:

1. Considering point A as the cell location, additional given parameters are Azimuth angle (angle made by the antenna from True north; zero degrees) and cell range (signal range).
2. Since a cell creates a sector that ranges over an angle of 120 degrees, it can be seen that triangles BAC and CAD are both angled 60 degrees at A.
3. To find the bounding box coordinates of each point, point A is given, the bearing of points B, C, and D from A, and the cell range.
4. GeoPy geodesic distance method gives the latitude/longitude from a certain point, given the bearing and distance between two points. Hence, three API calls are made for points B, C, and D. In the original sense, a geodesic is a path having the shortest route between two points in a curved space, in this case, Earth's surface. Geodesic distance produces more accurate results than Euclidean distance since, the ISOMAP algorithm allows geodesic distance to capture all the views of the manifold structure, unlike Euclidean distance [47].
5. Afterwards, a geojson file is created for each cell containing the geometry formed by the above bounding box having four coordinates. The H3 polyfill method fills a geometry given in geojson format with H3 hexagons at a specified resolution.

The final result is shown below:

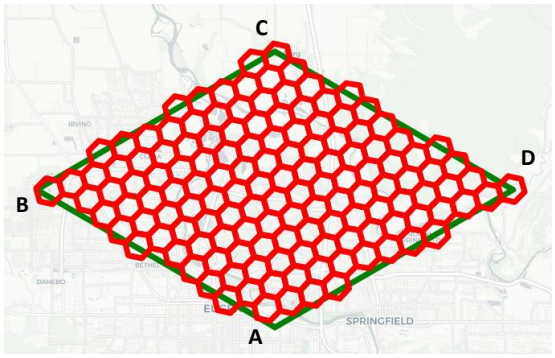


Figure 3.7: A cell coverage area filled with H3 hexagons

Calculation of the cell coverage area and the number of hexagons contained in that area using the H3 library is done. The next step is to find the common intersecting hexagons from the Kontur population data set. The above cell area regions filled with hexagons are then overlapped onto Kontur population hexagons shown below, each containing population density:

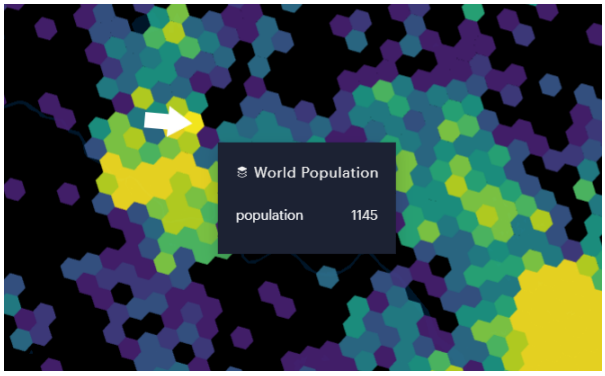


Figure 3.8: Hexagon grid laid on the world map

Area-type classification is based on the following proposal and these classifications are assumed as true labels for the machine learning analysis. Here Subs is the subscriber density.

Area Type Classification Proposal*	
Subs	Area Type
< 5	Dwelling
>= 5 and < 200	Rural
>= 200 and < 1500	Suburban
>= 1500 and < 3000	Dense Suburban
>= 3000 and < 10000	Urban
>= 10000	Dense Urban

Figure 3.9: Area Type Classification Proposal. *Inspired by Mobile Experts and ETSI [29]

3.3 Equivalence test of feature distributions

The 6 area classes are based on an ETSI standard, however, it is not necessarily the case that all these 6 classes will be relevant in this project as was shown in data exploration and analysis that it looks like 3 classes seem more in line with significant differences in terms of configuration Management CM data and Subscriber behavior PM data. The effort is to examine statistically significant evidence provided by the data at hand between closely created groups such as Urban and Dense Urban, Suburban and Dense Suburban and Rural and Dwelling as defined in figure 3.9.

It can be inferred from this test that the underlying distribution between the six classes is similar and thus can be merged. Since the data sets included in this master thesis do not strictly follow a Gaussian distribution, a non-parametric test known as Mann-Whitney U Test is commonly used as an equivalence test between two independent groups [36]. A general formulation of hypothesis testing user Mann-Whitney U Test is to assume that:

1. All the observations from both groups are independent of each other.
2. The distributions of both populations are identical under the null hypothesis H_0.
3. The distributions of both populations are not identical under the alternative hypothesis H_1.

The step-by-step algorithm for implementing the Mann-Whitney U test hypothesis testing is given as:

1. The first step is to state the null hypothesis, which is a statement that there is no significant difference between the two groups being compared. In this case, the hypothesis states that there is no significant difference between Urban and Dense Urban, Suburban, and Dense Suburban, and Rural and Dwelling classes.
2. The alternative hypothesis is the opposite of the null hypothesis, and states that there is a significant difference between the two groups being compared. In this case, the hypothesis states that there is no significant difference between Urban and Dense Urban, Suburban, and Dense Suburban, and Rural and Dwelling classes.
3. The next step is to collect the data from the two groups being compared. For this experiment design, 300 random observations from both CM + PM data, were sampled from each class population. This is done to remove sampling bias because there are only 459 observations that belong to the Dense Urban class.
4. The Mann-Whitney U test requires that the data be ranked, rather than using the actual values. This is because the Mann-Whitney U test is non-parametric, meaning that it does not assume that the data is normally distributed. Combine the data from both groups and rank them from smallest to largest, ignoring ties. If there are ties, assign each tied observation the average of their ranks.
5. The U statistic is the sum of the ranks of the observations in one group, minus the expected value of that sum under the null hypothesis [36]:

$$U_1 = R_1 - \frac{n_1(n_1+1)}{2} \tag{3.3}$$

where n1 is the sample size for sample 1, and R1 is the sum of the ranks in sample 1

$$U_2 = R_2 - \frac{n_2(n_2+1)}{2} \tag{3.4}$$

6. Choose the smaller of the two U statistics as the test statistic: U = min(U1, U2). The U statistic is a non-parametric alternative to the t-test, which is used when the assumption of normality is violated.

7. For samples more than 30, the value of U approaches a normal distribution, and so the null hypothesis can be tested by a Z-test [24]. This does not indicate the use of parametric distribution but simply using the standard normal distribution to estimate the chance-level probability of a result as extreme or more extreme than what would be obtained with the Mann-Whitney U test.

8. So the standard deviation of U is given as

$$\text{Std dev} = \frac{\sqrt{(n_a \cdot n_b) \cdot (n_a + n_b + 1)}}{\sqrt{12}} \tag{3.5}$$

   And the corresponding z value is given by:

$$z = \frac{U - \frac{(n_a \cdot n_b)}{2}}{\text{Std dev}} \tag{3.6}$$

9. Now the comparison is done between the observed z value and the critical z value to determine whether to retain or reject the null hypothesis:

   - if the absolute value of the obtained z is less than 1.96 for significance level α = 0.05, then retain H_0
   - if the absolute value of the obtained z is more than 1.96 for significance level α = 0.05, then retain H_1

3.4 Machine Learning System Design

True labels have been derived in the section 3.2 using area-type classification. Supervised machine learning is performed in this master thesis to predict the area types based on CM and PM features because there is a lot of unknown/missing information about the node placement/geographical location and sometimes only CM or PM features are available either through network service providers or Ericsson data extraction methods. The results obtained from supervised machine learning shall be used to predict the area types of different nodes in the field. The unsupervised machine learning algorithm is used to uncover inherent patterns of the nodes placed in different area types. An assumption in the analysis is that these classes (area types) may represent certain geographical conditions that could be useful in identifying patterns in different network deployment scenarios. To validate this hypothesis, machine learning is a modern technique to detect and understand a function that connects the input with the output.

Validation of the process is done by applying three different machine learning models; Random Forest, Multinominal Logistic Regression, and a Multilayer Perceptron on individual datasets i.e CM, PM, CM+PM to understand if the features listed above can represent and classify cells/nodes to different area-types. The machine learning pipeline is built using Machine Learning Library (MLLib) in PySpark which is a distributed computing data analysis tool. Benchmarking of good results is done by Experts at Ericsson. Figure 3.10 highlights the steps used for machine learning analysis.

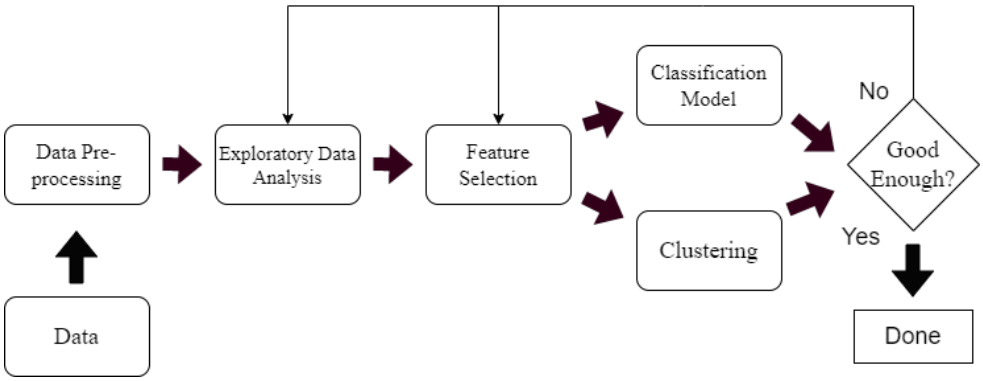


Figure 3.10: Machine Learning system design

Data Pre-processing

Cellular Network traffic is highly complex and is highly varying in nature. Data pre-processing is a major step in this case for a well-performed data analysis and machine learning pipeline. The pre-processing has 5 steps:

1. Unique Identifier - Since different mobile service providers use their methods to tag each cell in their data set. It is important to create a unique identifier for cells that can be used as a standard across all data sets. In this method, each node is tagged 'eNodeB', and cells are identified as 'cell' keywords respectively.

2. Removing Null Values - Considering the technical error while generating the data, it is possible to have incomplete information about various features. Null values were filled with zero since an absence of value denotes no information about the feature and can be labeled as zero. In the case of categorical features, the observations are dropped.

3. Dropping Duplicates - Data collected by mobile service providers could have traces of duplicate values introduced by technical issues or faulty system reporting, hence, the duplicate observations have been dropped. Duplicate observations constituted only around 0 - 1%.

4. Labeling data set by joining - True labels are obtained using LTE eNodeB and Kontur population data sets, that contain all the cells/nodes used in this master thesis, but field data such as CM, and PM do not have labels tagged to them. So, CM and PM data were inner joined on a set of unique primary keys of 'eNodeB' and 'cell' to tag each observation contained in CM and PM data with area type. In an inner join, only the rows that have matching values in both tables are included in the result set. This means that if there is a row in one table that does not have a matching row in the other table, it will not be included in the result set.

5. Scaling Features - As discussed, cellular traffic data could be highly varying and hence certain transformations are required to distribute the importance of features equally. Min-Max normalization was used in this step. It greatly increases the training speed of the models used and prevents numerical difficulties during the prediction [23].

Exploratory Data Analysis

One of the first objectives of a data scientist, after data cleaning, is to understand the features of a dataset and to uncover some knowledge about the responses and their overall interactions. Many visualization techniques are used to plot different features such as numerical,

categorical, text, etc. The visualizations provide insights that can be used to make informed decisions about corresponding real-world settings [10].

Different kinds of plots were used such as Histograms, Cumulative distribution plots to represent numerical values, Box plots, and pie-chart to represent categorical values. Both CM and PM data were visualized and analyzed. Exploratory data analysis is important to uncover and provide information about predictors and their relation with the response variable which could lead to an increase in the predictive ability of the machine learning model [14]. It is a way to establish a pathway between heuristic expectations in the real world and what the data conveys about the real world.

Feature Engineering

It is often the case that not all the original features that came with the data set contain predictive information related to response and hence need to be engineered or dropped. It is counterproductive to include uninformative features while predictive modeling [10]. There are two types of features used in this analysis. One is related to the configuration management (CM) of the nodes and the other provides performance measures (PM). These features have been chosen based on modeling the typical user behavior in a telecom network.

Classification Model

The six area-type classes referred to in section 3.9 were inspired by studies done at ETSI (European Telecommunications Standards Institute) [29], which is an independent, not-for-profit, standardization organization in the field of information and communications and Mobile Experts which is a mobile network consultancy. To consider them as the ground truth, machine learning is utilized for the computer to learn the defining function between inputs and outputs and test the hypothesis that there is indeed an underlying pattern in the data that can clearly distinguish between the classes discussed above. Since there are two data sets with different sets of features, Three different classification models are trained for CM (Configuration Management), PM (Performance Measures), and CM + PM data sets at the node level.

The models are trained in the PySpark framework with appropriate API method calls for respective algorithms. The datasets are available in tabular format with different predictors for each of the CM, and PM data and the response is the area type labels obtained in section 4.3. The dataset is split into training and testing sets for fitting the model with 80% observations and 20% observations in each category. The training set is further divided into training and validation sets to perform cross-validation and 3 folds are performed for each classifier. Each classifier contains several hyper-parameters which were tuned using a cross-validation technique. The hyper-parameters are important to select the best model for a particular dataset.

To evaluate model performance, several evaluation metrics have been used such as Accuracy, F1-score, precision, and recall. F1-score is chosen as the preferred metric since it is the harmonic mean of precision and recall, and this creates a balance of over-optimistic results and imbalanced datasets [22].

Cross Validation - Hyper-parameter tuning

CrossValidator class is available in PySpark framework that allows big data cross-validation technique to tune the hyper-parameter. The procedure begins by splitting the data set into a set of folds (sample of data) which are used as separate training and test datasets. E.g., with k=3 folds, CrossValidator will generate 3 (training, test) dataset pairs, each of which uses 2/3 of the data for training and 1/3 for testing.

CrossValidator computes the average evaluation metric for the 3 Models produced by fitting the Estimator on the 3 different (training, test) dataset pairs. Here are the steps to perform Cross-Validation:

1. Load the data into a PySpark data frame using the read function. CSV, JSON, or Parquet file formats are supported.
2. Preprocessing of the data is done if required. This may involve cleaning, feature engineering, and scaling. PySpark's built-in functions or user-defined functions (UDFs) can be used for this purpose.
3. Define the machine learning algorithms that are discussed in section 3.4 and set its hyper-parameters. This is called the estimator. LogisticRegression, RandomForestClassifier, and MultilayerPerceptron estimators are used in this master thesis. Each has its own set of hyper-parameter.
4. Define the metric that is used to evaluate the performance of the model. This is called the evaluator. MulticlassClassificationEvaluator is used in this thesis since the project deals with more than two classes.
5. Define the cross-validator by instantiating the CrossValidator class. Set the estimator, evaluator, and number of folds to use for cross-validation.
6. Train (fit) the cross-validator models created in step 5 on the data and evaluate with the evaluation metric defined in step 4.
7. If the performance of the model is not satisfactory, hyper-parameter tuning is done by repeating steps 3 to 6 with different hyper-parameters. PySpark's built-in ParamGridBuilder is recommended to create a grid of hyper-parameters to search.

Clustering

K-Means clustering is used in this master thesis project to cluster and find inherent network patterns in the dataset as discussed in section 2.3. In this master thesis, silhouette score has been used to find the best clusters which is a more robust and comprehensive measure of clustering quality that takes into account both the compactness of the clusters and the separation between clusters, and is generally considered to be a better method than the elbow method for evaluating the optimal number of clusters in k-means clustering [44]. The general steps to calculate the silhouette score and find the optimal number of clusters are:

1. For each point x_i, calculate the following:

   a. The average distance between x_i and all other points in the same cluster. This is denoted as a_i.

   b. The average distance between x_i and all other points in the nearest cluster that x_i is not a part of. This is denoted as b_i.

2. Calculation of the silhouette score for each point x_i is done using the equation:

$$s(i) = \frac{(b_i - a_i)}{max(a_i, b_i)} \tag{3.7}$$

   The silhouette score s(i) ranges from -1 to 1, where a score of 1 indicates that the point is very well-clustered and a score of -1 indicates that the point is misclassified.

3. Calculate the average silhouette score for all points in the dataset to get an overall measure of the quality of the clustering solution.

4. Repeat steps 1 to 4 for different values of k and choose the value of k that gives the highest average silhouette score.

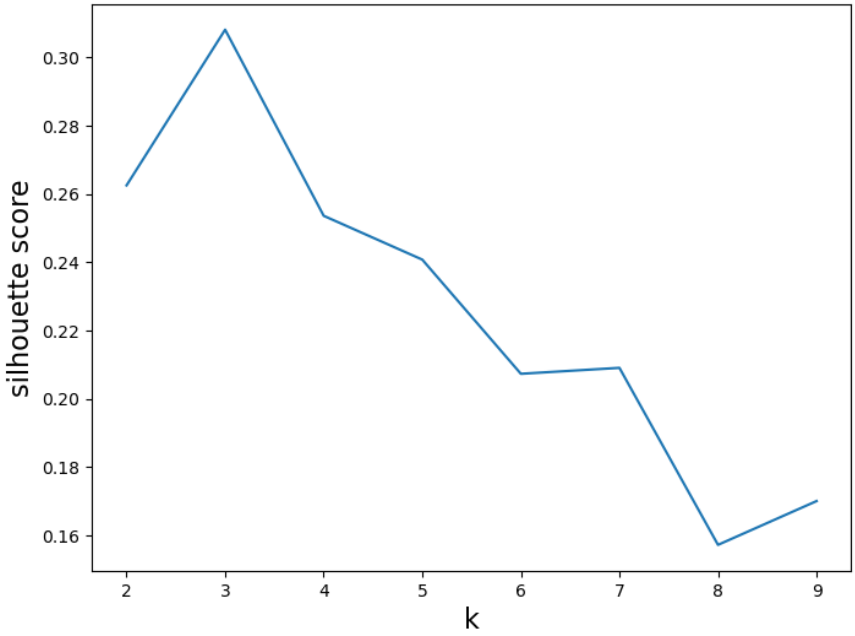


Figure 3.11: This example line plot describes how silhouette score varies with different numbers of clusters. Y-axis denotes the silhouette score and X-axis denotes the number of clusters. It can be clearly seen that for cluster 3, the score is maximum, and thus is the best choice for number of clusters

After finding the optimal number of clusters, the K-Means model is fitted on the training data without true labels and specifies the optimal number of clusters identified using the silhouette score. This divides the dataset having groups of observations assigned to different clusters and visualizing the clusters can render the decision-making process easy. In order to visualize clustering space into two dimensions, Principal Component Analysis (PCA) PCA technique can be used to visualize the clusters. Principal components include determining eigenvectors and eigenvalues from the dataset and top vectors are chosen as the components [13]. Most modern statistical frameworks are equipped with methods to perform PCA so the underlying working of PCA is not discussed in this project as it is outside the scope of the thesis.

Since visualizing clusters in a plot requires only two components, so it is assumed that PCA is performed after K-means and two principal components are plotted on the graph. An example of K-means clusters are shown in Figure 3.12:

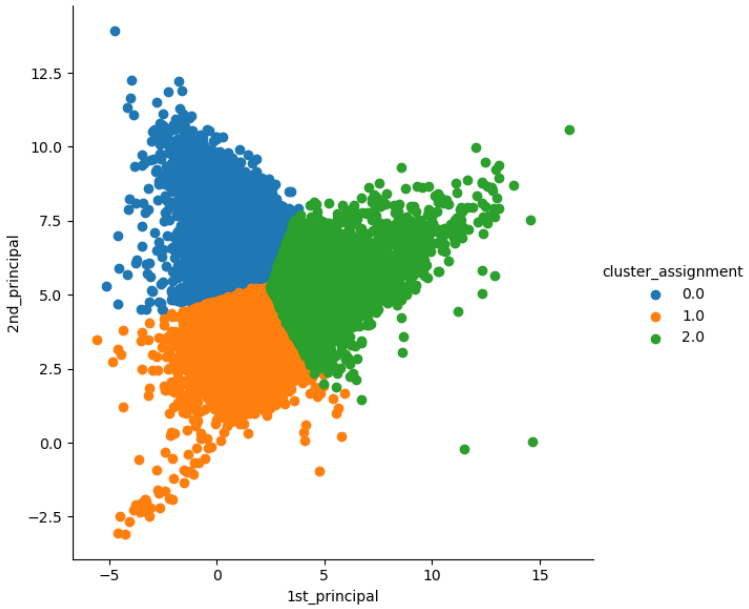


Figure 3.12: This example clustering shows three optimum clusters plotted on a two-dimensional graph. Here X-axis denotes 1st principal component with corresponding eigenvalues and Y-axis denotes 2nd principal component and corresponding eigenvalues. The clusters are assigned values from 0 to 2 and are color coded. This plot understanding clusters easier than comparing numerical values.

4 Results

This chapter presents the results generated by the techniques used in the method chapter during the thesis and provides answers to the research questions. The chapter includes several sections that start with area-type labeling and finish with Indoor/Outdoor Cell Placement analysis.

4.1 Equivalence test of feature distributions

The null hypothesis tested in this study was that the distributions of two independent groups are identical. The groups are a pairwise set of classes i.e, a) Dense Urban and Urban b) Dense Suburban and Suburban, and c) Rural and Dwelling. While the alternative hypothesis was that there is a significant difference between the distribution of the two groups. The test statistic and the corresponding z-value are shown in table 4.1:

Groups	Test Statistic	z-value
Dense Urban and Urban	44566.266	0.61
Dense Suburban and Suburban	45401.816	0.12
Rural and Dwelling	50219.416	1.56

Table 4.1: Hypothesis Testing results describing the U statistic and the corresponding z-value for each group.

In table 4.1, the z-values obtained from the experiment are all less than the critical z value of 1.96 as mentioned in section 3.3, which means there is not sufficient evidence to reject the null hypothesis, which is, the two groups are actually coming from the same distribution. This enables us to use a reduced set of classes, resulting from the combination of sets of two groups of classes namely:

- Urban (Dense Urban + Urban)
- Suburban (Dense Suburban + Suburban)
- Rural (Rural + Dwelling)

4.2 Data Exploration

Area Type labeling

The diamond approach to calculate the population density covered by each cell was chosen to identify the area type of the cells/nodes as discussed in section 3.2. The resulting distribution of classes as per ETSI standards is shown in figure 4.1:

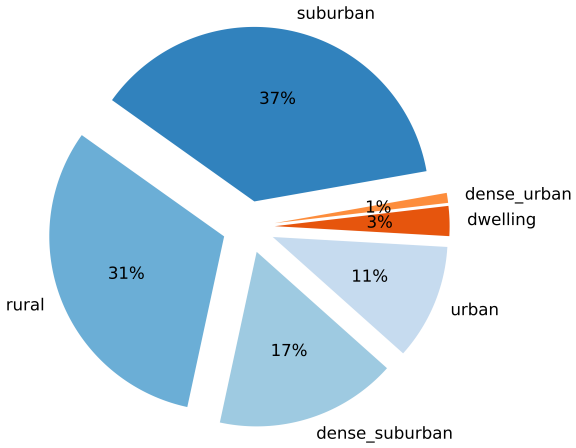


Figure 4.1: Cell level area type classification with six classes which were obtained from ETSI and Mobile Experts classification proposal.

As given in section 4.1, it can be concluded that there are three representative classes that a cell can be associated with. As seen in figure 4.2 how the distribution of cells in different area types is visualized, with the Suburban class being the most prominent followed by Rural and then Urban. All further analysis is done with the resulting three classes.

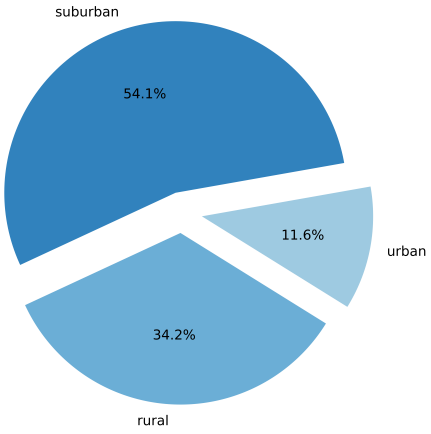


Figure 4.2: Cell level area type classification with three classes after merging the groups of classes where the class distribution was similar to each other.

Since a node consists of cells (antennas) that provide different cellular coverage to different sectors of the region. Every cell makes a sector of 120 degrees at some Azimuth angle and the number of cells installed at a particular node could range from at least 3 to almost 24 cells. This makes it an interesting topic to study area types associated with nodes rather than cells. As discussed in section 3.7, all the hexagons that lie within the cell coverage area are accumulated. Each hexagon id gives us the population density of the real geographic region that surrounds that hexagon. Hexagons are combined (population density) together to calculate area-type of the cell. For calculating node population density, the combination is only done for distinct hex ids covered by aggregating hexagons of all the cells and sum the population together neglecting the overlap effect. This gives the area type for the corresponding node as shown in figure 4.3:

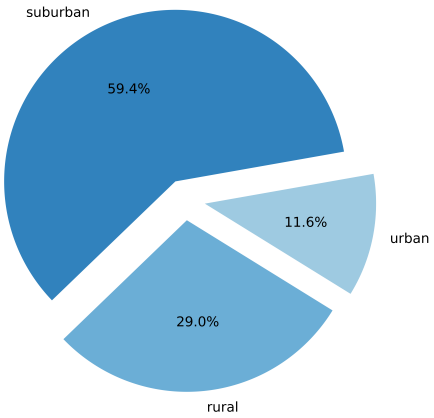


Figure 4.3: Node level area type classification with three classes where nodes are created by aggregating all the distinct hexagon ids for each cell.

The figure below shows real-world nodes located in the region of Florida and their corresponding area type evaluated using the method described in section 3.7. The figure is just for demonstration purposes and also serves as a sanity check of our methodology used in tagging area types. There are only three area types shown and as discussed before in section 3.3, only these labels will be used instead of six labels for further analysis. It can be seen that nodes classified as Urban are placed close to the city center while suburban and rural nodes are placed more outside the densely populated areas.

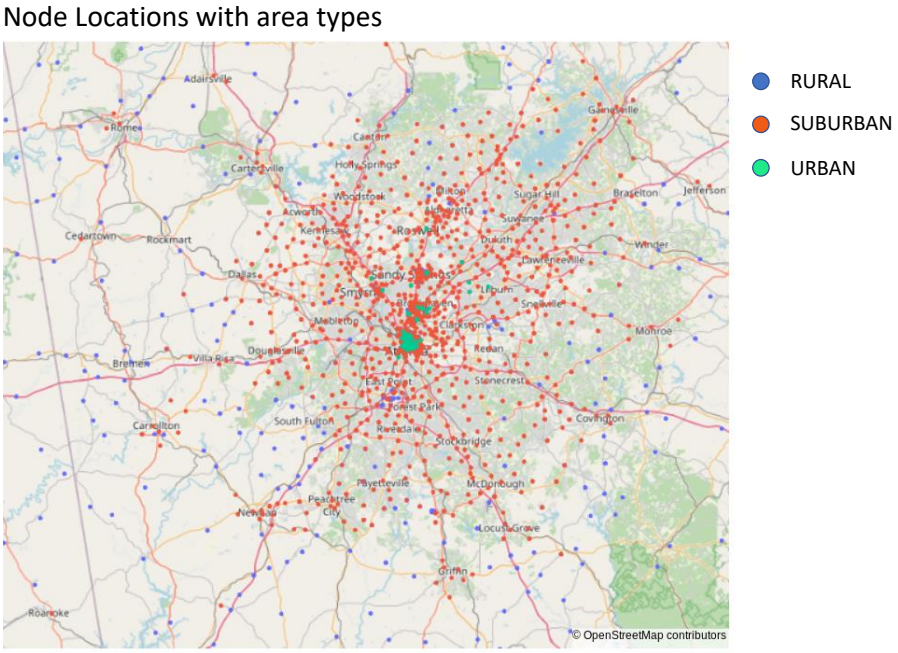


Figure 4.4: Nodes locations and area types in Miami in Florida Region, USA. Here it can be seen that the nodes that cover less population like Rural nodes, marked by blue are much spread out like towns, and highways, whereas Suburban nodes, marked by red, are moderately packed together where there is more population, and Urban nodes, most commonly reside in city centers, offices, and downtown areas.

Exploratory Data Analysis

Configuration Management Data

Empirical Cumulative Distribution Function (ECDF) ECDF plot of Configuration Management features per area type. Since these are empirical plots, no assumption has been made about the distribution of features. Some observations could be made based on the ECDF graphs shown below in the Figure 4.5:

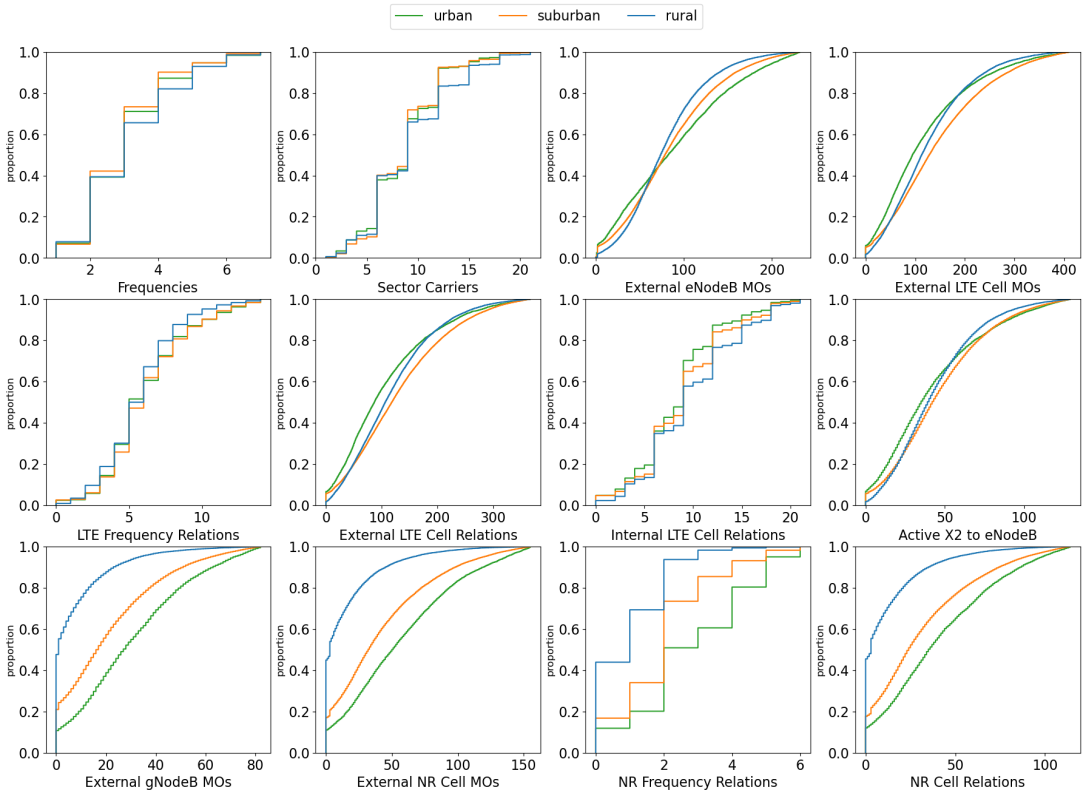


Figure 4.5: ECDF of CM features per area type. The X-axis represents the features and Y-axis denotes the proportion of values.

- In features like Frequencies and Sector Carriers, it seems some values are stepwise, meaning some proportion of observations have the same feature value.
- The biggest difference is in the coexistence of 4g (LTE) and 5g (NR). In an urban area with more 5g deployment, it can be seen that more NR cell relation, gNodeBs, etc.
- There are some parameters that stand out in this analysis such as is seen that higher External gNodeB MOs in Urban areas than in rural.
- Prominent features from this analysis would be 'External gNodeB Mos', 'NR Frequency Relations', and 'NR Cell Relations', 'External NR Cell MOs'.

Box-plot of Configuration Management features per area type. Box-plot shows the distribution of a variable over its whole range (percentiles). Specifically, first 25%, then the bulk 50% (25-75), and lastly more than 75%. The whiskers are created using the inter-quantile range. The IQR is a measure of statistical dispersion, and it represents the difference between the first quartile (Q1) and the third quartile (Q3).

$$\text{IQR} = Q3 - Q1 \tag{4.1}$$

$$\text{Upper whisker} = Q3 + 1.5 * IQR \tag{4.2}$$

$$\text{Lower whisker} = Q1 - 1.5 * IQR \tag{4.3}$$

Some observations could be made for features on the box-plot graphs shown below in Figure 4.6 :

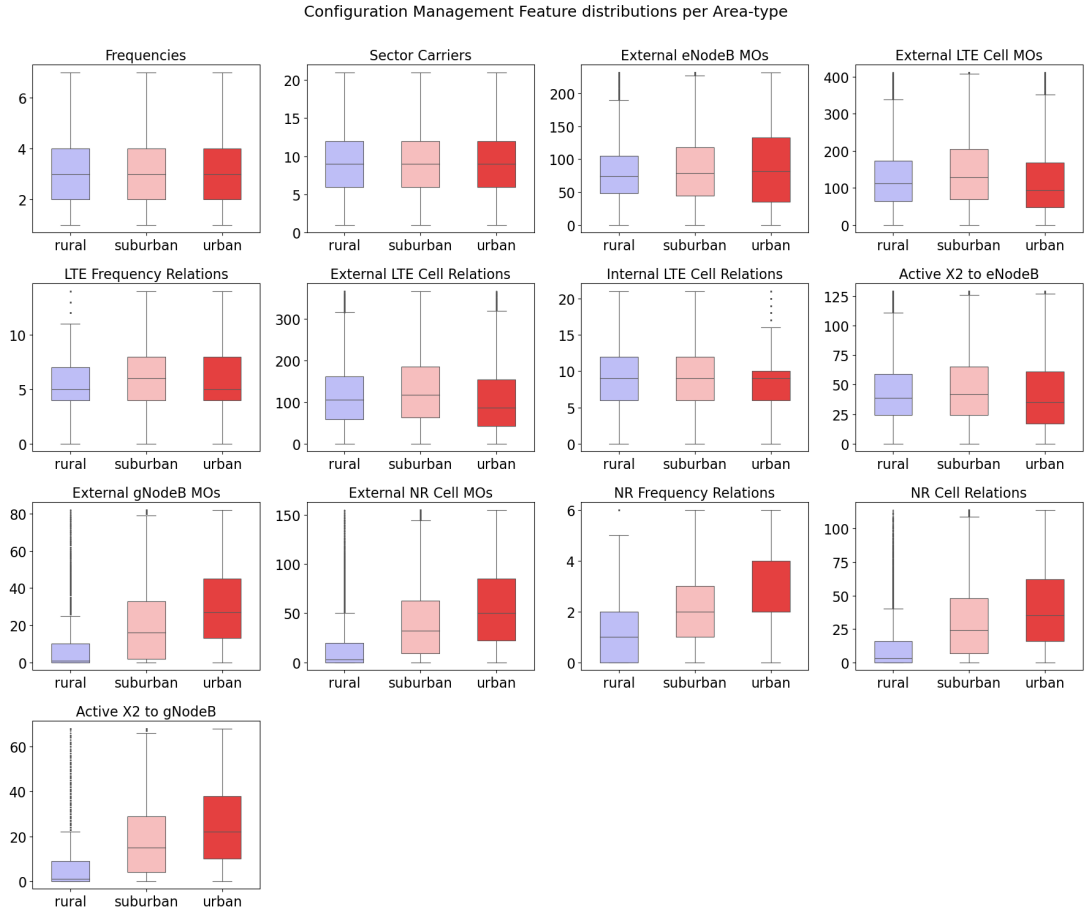


Figure 4.6: Distribution of CM features per node per area type. Box plots show the distribution of features in each category along with the inter-quantile range. The x-axis represents each area type and Y-axis represents the value range for each feature.

- NR Cell Relations is higher in urban areas with most of the values lying between 20 and 60, and relations can be found to go rarely higher than 20 in Rural Areas.
- External gNodeB MOs have too many outlying values in rural areas. Whereas, nodes in urban areas have 50% of values between 20 and 40.

Performance Measures Data

Empirical Cumulative Distribution Function plot of Performance measures features per node per area type. The following observations could be made based on the ecdf graphs shown below in figure 4.7:

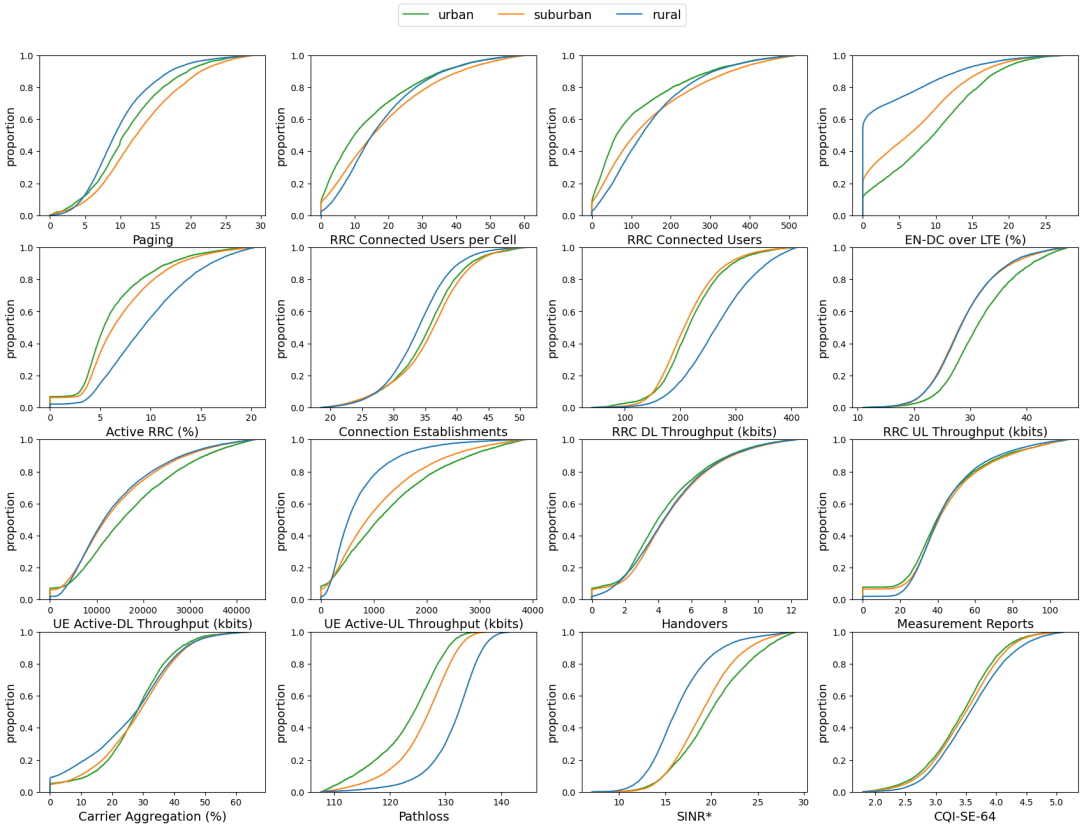


Figure 4.7: ECDF of PM features per area type. The X-axis represents the features and Y-axis denotes the proportion of values.

- In the dense urban area there are more 5g nodes so more LTE nodes are able to set up EN-DC connections.
- UE-Active DL-Throughput is highest in the urban area - it is connected with more EN-DC over LTE in the Urban area and could be with more probability to setup carrier aggregation.
- It can be seen that the highest RRC DL throughput is in the Rural area, please note that these parameters contain IDLE periods so it looks that ues in the Rural area has fewer idle periods and it could be in relation to connection establishment.
- UL-throughput is also highest in urban areas and this time RRC UL throughput is also higher in urban areas as was expected.
- Handovers and Measurement reports are on the same level

Box-plot of Performance measures features per area type. Some observations could be made for features on the boxplot graphs shown below in figure 4.8:

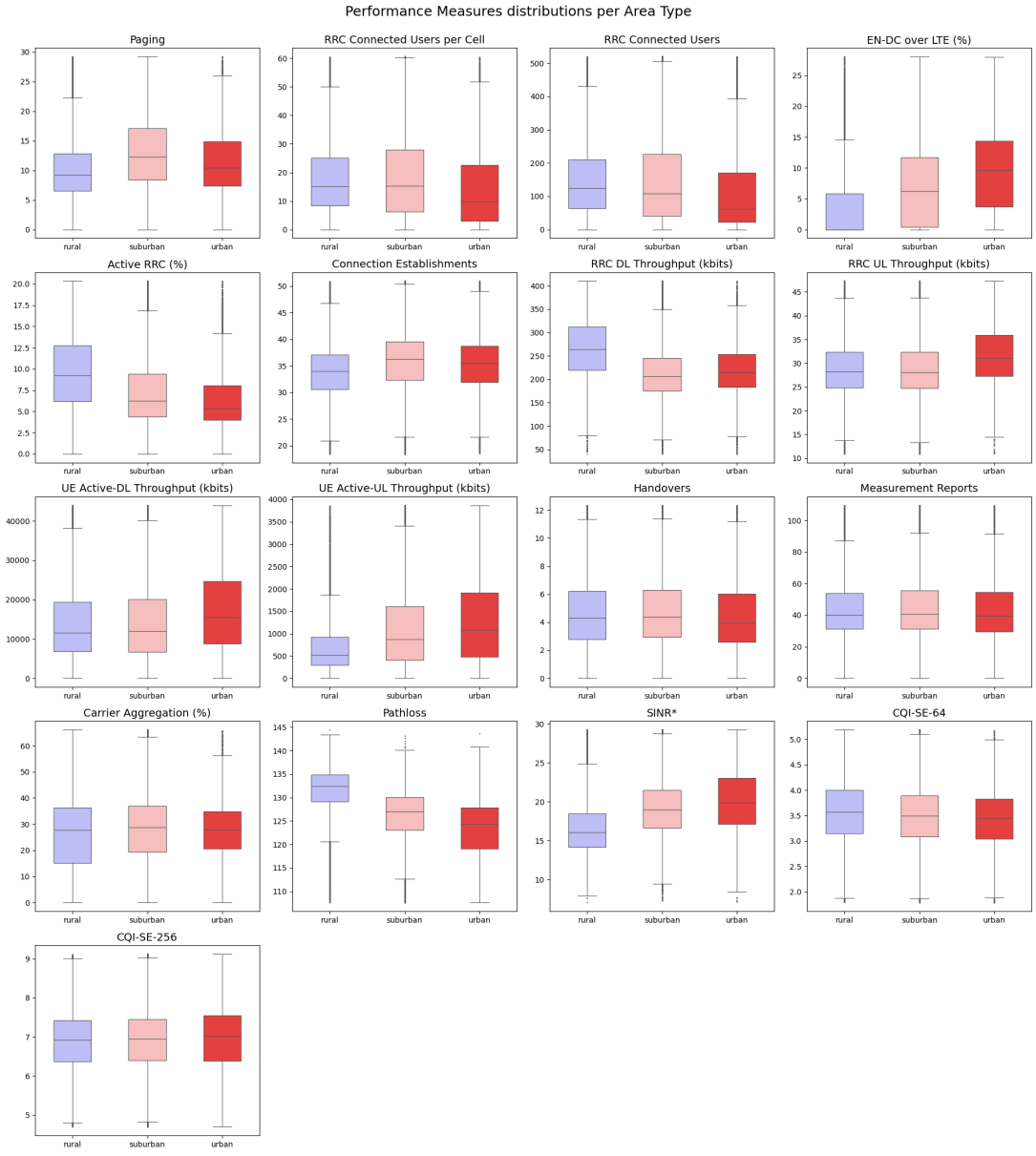


Figure 4.8: Distribution of PM features per node per area type. Box plots show the distribution of features in each category along with the inter-quantile range. The x-axis represents each area type and Y-axis represents the value range for each feature

- It is evident that there is Higher Active RRC (%) in Rural areas with an average of 10 higher than the average of 5 in urban settings.
- Nodes located in Urban areas have Much higher UE Active-UL Throughput than in rural which could mean that users are more connected to the network in Urban areas.
- Low pathloss in urban areas but much higher in rural and suburban.

4.3 Feature Correlation Analysis

This analysis is performed for a deeper understanding of the features present in our dataset. Since, in this preliminary study, there are a lot of features present, it may not be the case that every feature contributes to feature selection. There are some observations that can be made and understand better the correlation between the features.

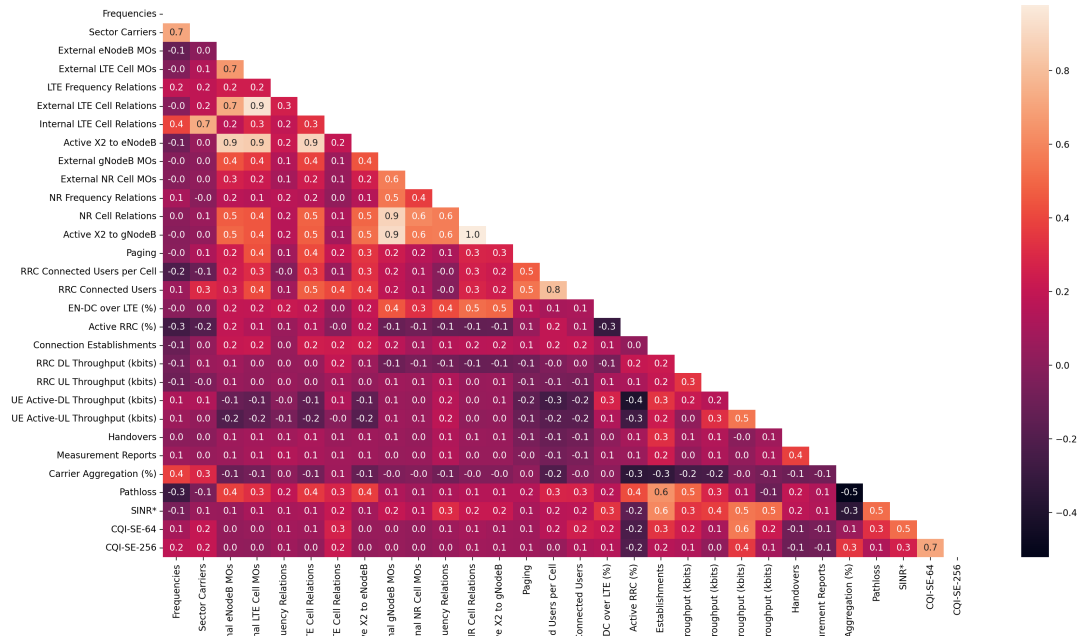


Figure 4.9: Feature correlation heatmap. Heatmap shows the correlation coefficient for each feature pair.

1. These are node-based features that are being analyzed for the final machine learning pipeline and clustering. It consists of CM + PM features.
2. This figure tells us the correlation between features which is defined between -1 and 1. 1 represents complete correlation (same information) -1 represents anti-correlation (opposite direction). High correlation should be analyzed further.
3. 0.9 measure between 'External LTE Cell Relations' and 'External LTE Cell MOs' means both features convey the same information and one of them can be dropped.
4. Similarly, for correlation between 'Active X2 to eNodeB' and 'External eNodaB MOs', 'External LTE Cell Mos, 'External LTE Cell Relations'.

4.4 Machine Learning

Assuming true labels were obtained as described in section 4.2, three supervised machine learning algorithms are trained on both CM and PM data using cross-validation for hyper-parameter tuning to support our hypothesis that features chosen above have a relationship with our classes. The models are compared quantitatively using the F1-score evaluation metric and several others to understand how well the Telecom features (CM + PM) can distinguish between different area types.

Classification - Random Forest

In this section, the Random Forests algorithm described in section 2.3 is modeled using the PySpark framework. *RandomForestClassifier* method is used to train the model in PySpark. In a Spark model, transform is a universal method that is used to get predictions from the models on test data. *MulticlassClassificationEvaluator* is used to set the evaluation metric, which, in our case is a multi-class classification problem and 'F1' as the metric choice.

The model is trained on both CM and PM data. The split for the dataset is 80% train and 20% test with Cross-Validation performed on training data that is further divided into train and validation datasets. The parameter grid search technique was used to evaluate 3 folds cross-validation with the following grid search values:

1. Max Depth: 5, 10, 20 leaves
2. Number of trees: 50, 100, 200 trees
3. maximum features subset: 3, 5, 9, 14 features

And the best classifier model with hyper-parameter is as shown below and was then fit on the whole training set and F1-score was evaluated:

- Max Depth - 20 leaves
- Number of Trees - 200 trees
- Maximum feature subset - 9 features

F1-Score on evaluation of testing data was 0.77.

Classification - Multinomial Logistic Regression

Multi-class classification is also supported via multinomial logistic (softmax) regression in PySpark with Python implementation of the algorithm. The model is fit using the *LogisticRegression* method in PySpark. It is also possible to regularize the coefficients using the combination of L1 and L2 penalties. Alpha is the argument used in PySpark, for alpha = 0, the penalty is an L2 penalty. For alpha = 1, it is an L1 penalty.

The split for the dataset is 80% train and 20% test Train-Test split for the CM+PM data set. Cross-validation was performed using 3 folds with the following grid search values:

1. Max Iteration: 10, 20, 50, 100
2. Regularization Parameter: Range between 0 and 1

And the best classifier model with hyper-parameters are as shown below and was then fit on the whole training set and F1-score was evaluated on testing data:

- Max Iteration - 50
- Regularization Parameter - 0.0

F1-Score after evaluating on testing data was 0.73

Classification - Multilayer Perceptron

The classifier can be trained using the $MultilayerPerceptronClassifier$ method. Data is stacked within partitions for the purpose of distributed computing. If the block size is more than the remaining data in a partition then it is adjusted to the size of this data and *seed* to control the randomness of the model.

The split for the dataset is 80% train and 20% test Train-Test split for the CM+PM data set. Cross-validation was performed using 3 folds with the following grid search values:

1. Max Iteration: 20,50,100
2. stepSize: 0.01, 0.1

And the best classifier model with hyper-parameters are as shown below and was then fit on the whole training set and F1-score was evaluated on testing data:

- Max Iteration - 50
- Regularization Parameter - 0.01

F1-Score after evaluating on testing data was 0.75

Random Forests classifier can be concluded as the best classifier for this dataset and this specific problem. All the performance metrics are higher in the RF case, followed by MLP and lastly Logistic regression. The classification metrics for all the classifiers can be summarized in Table 4.2:

	Accuracy	**F1-Score**	**Precision**	**Recall**
Random Forests	0.78	0.78	0.80	0.79
Logistic Regression	0.76	0.73	0.74	0.76
Multi-layer Perceptron	0.77	0.75	0.76	0.77

Table 4.2: Results for ML classification Models. The table describes a few performance metrics discussed in sec 2.3. Since F1-score is the harmonic mean of Precision and Recall, it is considered to be the preferred metric for this master thesis.

Clustering

Clustering is an unsupervised learning method (without true labels) where it can capture the patterns in data that are similar to each other. This will give us insights into the characteristic behavior of users, configuration, settings, etc. under different area types and deployment scenarios. Preliminary results include clustering nodes based on both CM and PM features.

Configuration Management Data

The silhouette score method was used to determine the optimal number of clusters. The range of clusters was set from 2 to 10. The higher the silhouette score, the better the cluster. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). The silhouette graph is given in figure 4.10:

Two clusters are the best choice for CM data and it differentiates the data points very clearly.

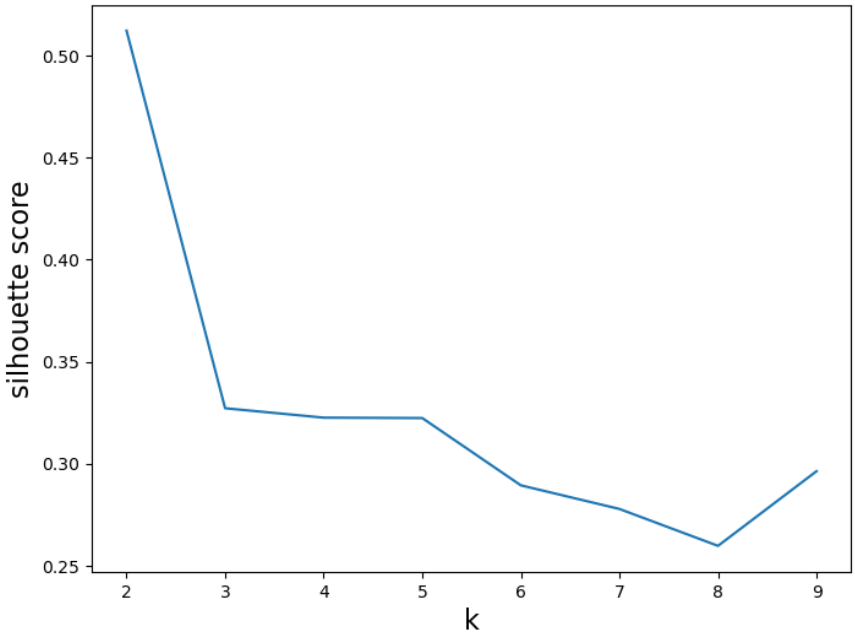


Figure 4.10: Silhouette Score for CM features clustering. X-axis denotes the number of clusters and Y-axis denotes the silhouette score. For CM features, two clusters can be seen as the optimal choice for clustering.

And the corresponding clusters are shown 4.11:

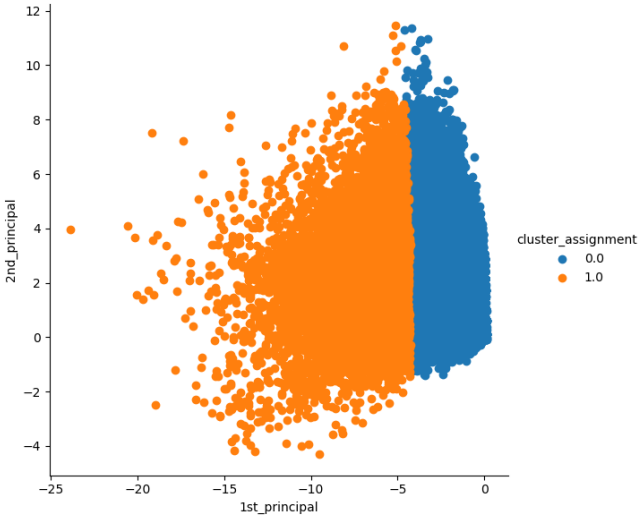


Figure 4.11: CM nodes getting clustered into two clusters. There are some extreme examples present in both the clusters in the right top corner. Cluster 1 seems to have lower values than Cluster 0. X-axis denotes 1st principal component and Y-axis denotes 2nd principal component with corresponding eigenvalues.

Cluster analysis was performed using descriptive statistics of the CM features for the corresponding clusters 0 and 1. By going through the table, it can be inferred that cluster 0 has higher values and could be denoted as 'heavily loaded' nodes. Mean values are higher than

Median and this could indicate the presence of outlying observations. The Table is shown as follows 4.3:

CM Features	Cluster 0			Cluster 1		
	Mean	Median	Max	Mean	Median	Max
Frequencies	3.4	3	16	3.3	3	16
Sector Carriers	9.5	9	28	8.6	9	28
External eNodeB MOs	156.2	144	512	68.9	65	388
External LTE Cell MOs	282.1	252	1891	108.2	98	610
LTE Frequency Relations	7.4	6	22	6	5	21
External LTE Cell Relations	240.1	224	1217	96.4	89	429
Internal LTE Cell Relations	11.5	9	45	9.2	9	48
Active X2 to eNodeB	88.4	82	394	35.4	34	179
External gNodeB MOs	58.7	49	256	11.2	7	172
External NR Cell MOs	132.5	96	3186	22.5	14	1058
NR Frequency Relations	4	3	18	1.7	1	18
NR Cell Relations	83.3	72	496	15.7	10	120
Active X2 to gNodeB	48.7	42	251	9.3	6	71

Table 4.3: Comparison of statistical values for Clusters 0 and 1 for CM features.

Performance Measures Data

PM features are used in this clustering step. PM features mostly resemble real-world user behavior connected to the nodes and the algorithm generates three clusters as evidenced by the silhouette score shown in Figure 4.12:

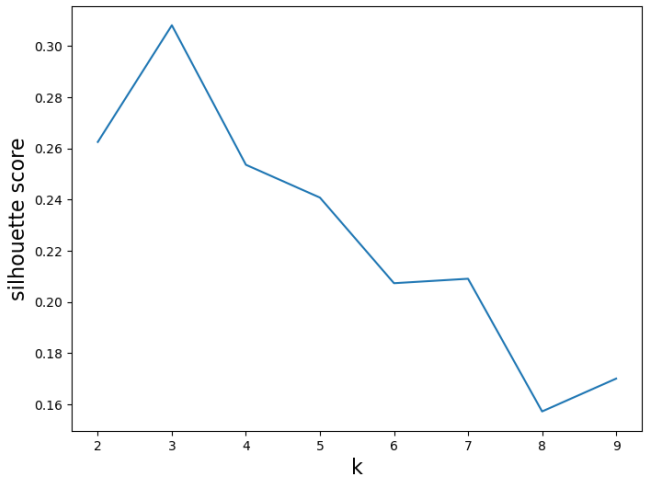


Figure 4.12: Silhouette Score for PM features clustering. X-axis denotes the number of clusters and Y-axis denotes the silhouette score. For PM features, three clusters can be seen as the optimal choice for clustering.

And the corresponding clusters are shown in figure 4.13:

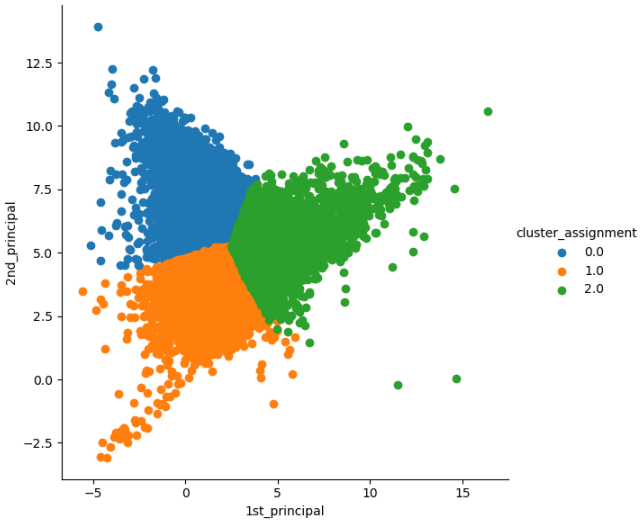


Figure 4.13: CM nodes getting clustered into two clusters. There are some extreme examples present in both the clusters in the right top corner. Cluster 2 seems to have higher values than Cluster 0 and 1 with nodes in the top right corner. Cluster 1 has subdued values with nodes in the left bottom corner. X-axis denotes 1st principal component and Y-axis denotes 2nd principal component with corresponding eigenvalues.

Looking at the tables below, it can be concluded that the user behavior among different clusters is a bit different with Cluster 2 having the maximum UE Active-DL Throughput which could mean more users connected and exchanging data. Cluster 1 has the least CQI-SE-64 which denotes the channel quality and hence, the correspondingly lowest DL Throughput. Cluster analysis was performed using descriptive statistics of the PM features for the corresponding clusters 0, 1 and 2, which is shown in Tables 4.4 and 4.5:

PM Features	**Cluster 0**			**Cluster 1**		
	Mean	**Median**	**Max**	**Mean**	**Median**	**Max**
Paging	20	18.6	62	11	10	41.6
RRC Connected Users	397.2	368.8	1703.7	106.7	89.3	532.9
RRC Users per Cell	44.6	40.4	282.3	16.1	13.8	103.5
Active RRC (%)	8.4	7.2	60.6	12.7	11.1	94.3
Connection Establishments	37.1	36.9	71.8	32.2	32.2	138.9
RRC DL Throughput	210.1	204.3	530.2	252.8	236.4	1326.2
RRC UL Throughput	27.9	27	104.7	32.4	29.7	680.8
UE Active-DL Throughput	11148.4	10175.3	40902.9	9836.3	9160.5	51743.3
UE Active-UL Throughput	1113.4	948.9	13127.1	622.7	458.5	16548.3
Handovers	4.7	4.3	21.1	6	5	106
Measurement Reports	51.7	41.2	776.4	62.5	44.4	1984.5
Carrier Aggregation (%)	31.2	32.6	73.9	22.8	22.2	90.2
Pathloss	126.9	127.1	139.5	130.3	131.2	144.5
SINR*	19	19	28.9	15.9	15.7	28.4
CQI-SE-64	3.7	3.7	6.1	3.1	3.1	6.5
CQI-SE-256	7	7.1	9.5	6.4	6.4	10.2

Table 4.4: Comparison of statistical values for Cluster 0 and 1 for PM features.

PM Features	Cluster 2		
	Mean	**Median**	**Max**
Paging	10.9	10.1	39.2
RRC Connected Users	115.9	101.4	464.8
RRC Users per Cell	13.3	11.6	81.4
Active RRC (%)	5	4.7	33.7
Connection Establishments	37.3	37.1	231.3
RRC DL Throughput	245.8	229	4068.7
RRC UL Throughput	32.1	29.3	1278.6
UE Active-DL Throughput	29346.7	26591.3	170832.9
UE Active-UL Throughput	3278.7	2033.1	61586
Handovers	5.2	4.4	82.7
Measurement Reports	57.9	43.7	1479.7
Carrier Aggregation (%)	31.7	31.8	92.4
Pathloss	121.9	123.1	139.5
SINR*	21.3	21.1	31.4
CQI-SE-64	4	3.9	7.9
CQI-SE-256	7.5	7.4	14.3

Table 4.5: Comparison of statistical values for Cluster 2 for PM features.

Indoor/Outdoor Cell Placement

The results for indoor/outdoor classification are obtained by the combination of the Ericsson radio equipment lookup table and the network service provider specification. Each cell is marked whether it is placed indoors or outdoors, this dataset was joined with the CM and PM datasets, which already include area types for each cell, to get the area types along with cell placement.

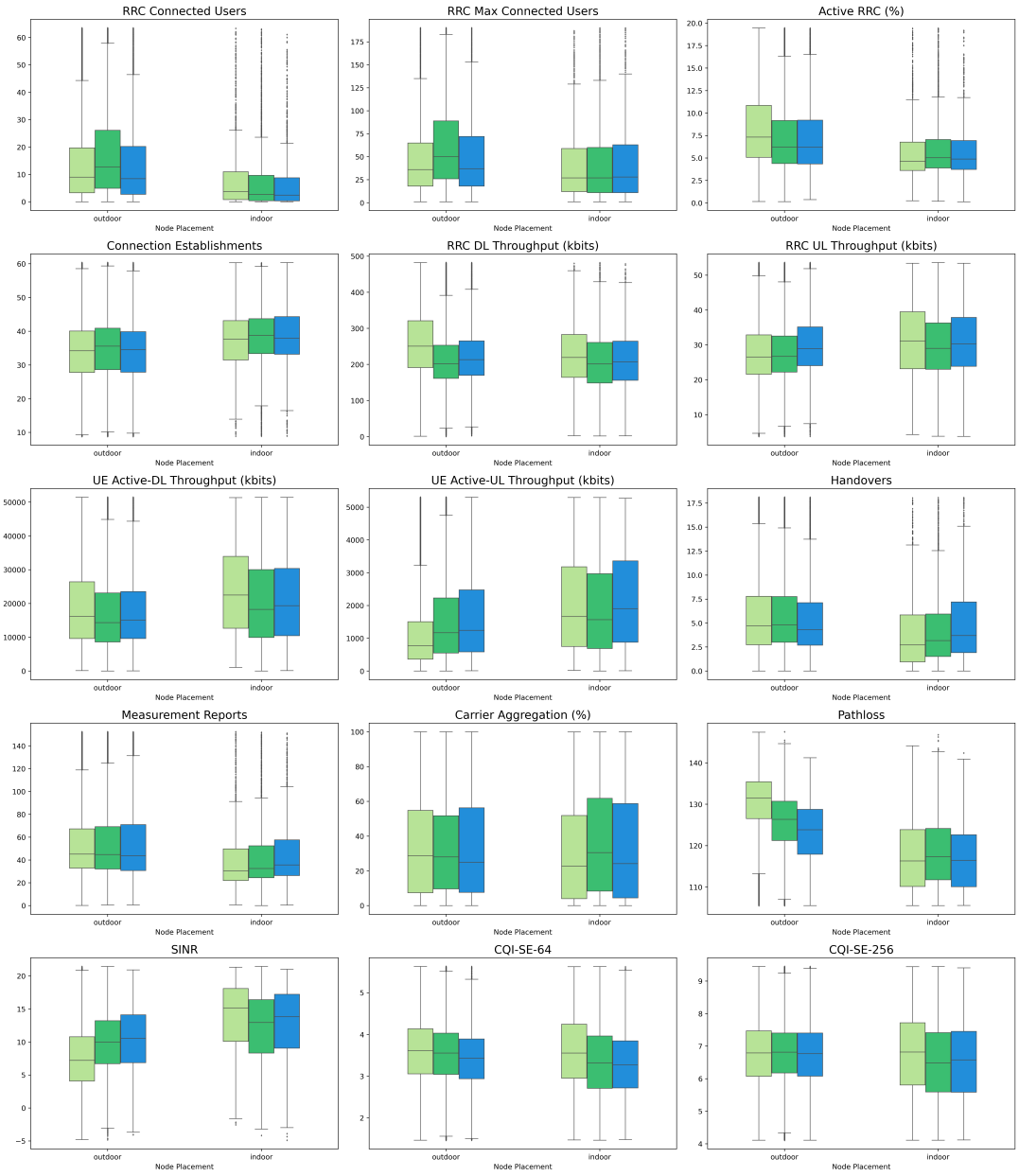


Figure 4.14: Feature analysis per area type in indoor/outdoor cell placement. It is evident that the average RRC connected users is low in indoor cells may be due to the fact that there are building, that could be closed on weekends or nights. Also, connection establishments per second are more in indoor cells, which could indicate a recurrent connection with the network. Also, UE active throughput is higher in Urban areas which makes sense, as there are more connections.

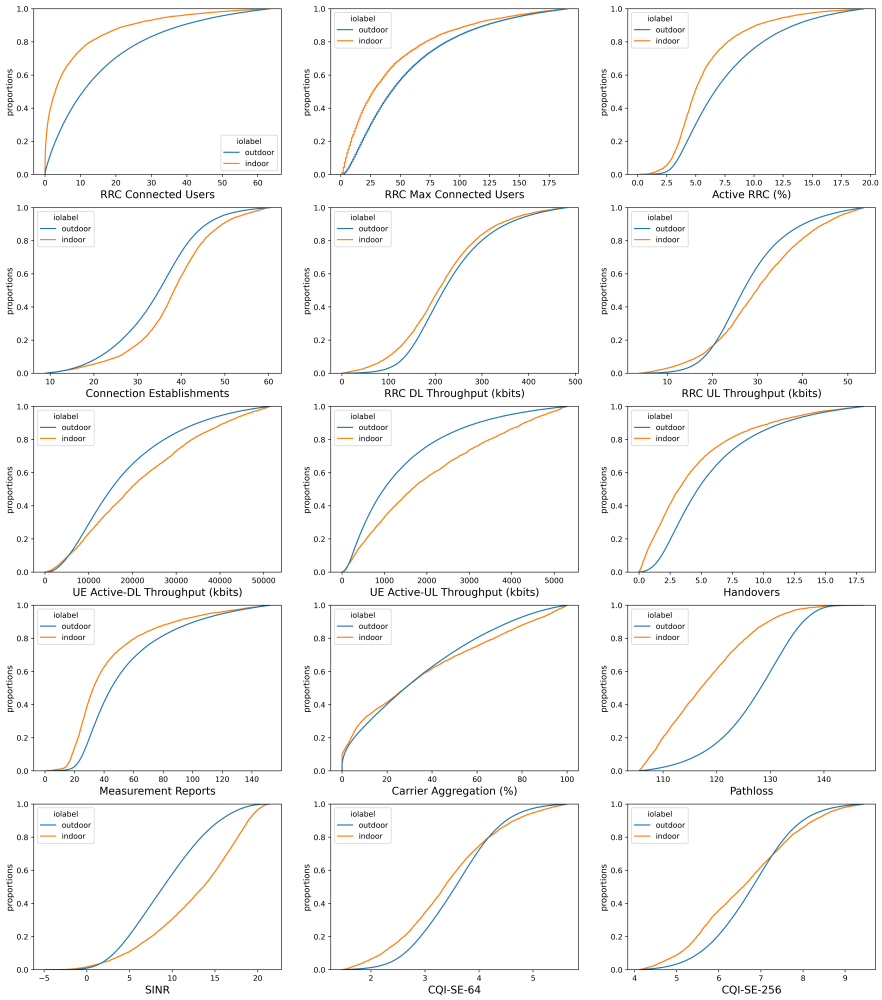


Figure 4.15: Feature analysis in indoor/outdoor cell placement. There is less Pathloss in indoor cells which makes sense as the cells are closely situated and the cell range is much higher. RRC throughput more in indoor. so users are more connected and send more data.

5 Discussion

This chapter aims at discussing the thesis results, the methodology used in the analysis, and the work in terms of a wider context.

5.1 Results

In this section, the results from the thesis are discussed. For each method, the results are generated and its contribution to the study is explained.

Data Labeling

Using ETSI standards, Six classes were defined according to the population density covered by each cell. After careful observation of the cdf of both CM and PM data, it seems that some of the class definitions had overlapping groups and it called for the need to check the hypothesis about the combination of classes. The hypothesis results in section 4.1 indeed confirm our hypothesis that there are groups of classes in this project that have similar distribution and we can consider them as the ground truth i.e., Rural, Suburban, and Urban. It can also be confirmed visually by referring to the geographical plot in figure 4.4 as Rural nodes are more separated and cover outskirt areas with scarce population density. Suburban nodes are closely situated and look at what appears to be more populated areas. Lastly, Urban nodes are highly dense most probably serving areas like multiplexes, city centers, University. As shown in figure 4.3, more than half of the nodes belong to the Suburban class and only 11% belong to Urban settings, which makes sense as there are less densely packed areas such as city centers and more suburban areas such as supermarket stores. This labeling could be valuable for Ericsson as it gives a tool to filter the nodes spread around the world based on area types and monitor the configuration and performance features.

Data Exploration

As seen in figure 4.7, there are higher External gNodeB MOs in Urban areas than in rural settings which were expected as it is safe to say that Urban settings have more advanced/"built-out" coverage capabilities. It can be intuitively reasoned that areas close to the city center have more users, and probably more modern equipment e.g. more 5G/NR coverage. This

is very much in line with what was expected to see. Referring to figure 4.5, there is also a "stepwise" incremental difference in intensities between the areas. Urban is the most intense, suburban in the middle, and rural the lowest. So there is an incremental change as you move from city-center to less populated areas.

There are many CM-values (configuration KPIs) that are very similar regardless of area-type e.g. Sector Carriers, External LTE MOs, active X2 to eNodeB, frequencies with not such large differences. Prominent features from this analysis would be 'External gNodeB Mos', 'NR Frequency Relations', and 'NR Cell Relations', 'External NR Cell MOs'. NR Cell Relations is higher in urban areas with most of the values lying between 20 and 60, and Relations are found to go rarely higher than 20 in Rural Areas. This behavior sounds reasonable according to area types. It is known that at least the level of "NR-maturity" can be used to differentiate the area types from one another.

Pathloss and SINR are anti-correlated within the area types and have more pathloss in rural areas than others. Pathloss is the average loss in signal strength over a distance. The urban setting has the worst pathloss and it could be due to various reasons such as Obstruction from tall buildings, Reflection, and diffraction from architectures, and in general high population density leading to more noise. Many subscriber behavior KPIs seem very similar: CQI-SE, Measurement-reports, and Number of Handovers. Data is aggregated on the node level and it could be possible to lose some information when aggregating the cells.

PM KPIs are calculated based on average over 6 days, this might lead to a value that is robust but loses observability on the low granularity differences between the areas. Perhaps there is a need to revisit these formulas in future works. Maybe, there are some features that activate to "alleviate" differences and "smooth out" any differences so a user experiences similar Quality of Experience (QoE) regardless of which area-type they are in. Being technologically developed networks, there is a high probability of self-adjustment phenomena. So maybe some of these KPIs from the subscriber point-of-view are not very good to differentiate areas from one another. (some speculation)

Machine learning - Classification

Based on RF-feature importance it can be said that IAT does not contribute anything due to it being a static constant of 10s for the entire network. There is a pretty even split of contributions from CM-values. One point of discussion is that it is not sure that it adds any value to have both CM, PM, and PM+CM studies. Maybe, it's a good idea to just keep the best ones. The other ones can be kept as an appendix instead. For PM+CM, it can be seen that Pathloss has a huge impact on the classification. This observation is quite reasonable based on what was seen in data exploration. F1-score of 0.77, which is good, it shows that the suggested methodology works but there is still room for improvement. Perhaps there are e.g. some outliers or KPIs that affect the results or maybe this is just a difficult classification problem that it should not be expected 100% accuracy. More elaboration on FP, FN, TP, TN can apprise us of the common misclassification/mistakes of the classifier.

Machine learning - Clustering

Clustering clearly indicates that there is an inherent behavior groups in the data that are split/mixed across the different area types:

- CM:
    1. Silhouette score determines two clearly separated clusters and it can be seen that result in figure 4.11.
    2. It can indicate two "configuration types" that are commonly adopted by mobile service providers in this USA region and with a particular service partner.

3. Referring to the box plot analysis in figure **??**, it can be deduced that one type of configuration is 'lightly' loaded and the other is 'heavily' loaded. Loaded may indicate, higher values for CM, and PM features.
4. This result can be further addressed as to what optimal classes (area types) could be used to classify node groups.

- PM:

1. Silhouette score determines three clearly separated clusters and results can be seen in figure 4.13.
2. It can indicate three "subscriber types" in the area types and is open to studying what behaviors can be inferred from these clusters.
3. This result can be further addressed as to what optimal classes (area types) could be used to classify node groups under different user behaviors.

5.2 Methodology

In this section, the methods used during the thesis work are discussed.

Development

Besides obtaining insight into node groups and their behavior under different deployment scenarios, it is also important to verify whether the method functions adequately. This chapter evaluates the implementation of the proposed methods and discusses whether the method produces plausible results and is useable. Access to hardware, data, and software was readily available in this project. It allowed us to perform analytical iterations, improving on previous work. Some data limitations were noticed that limited the scope of the study such as only USA regions were considered in this study, also data was handled only for one service provider.

Different service providers have different methods to handle, store and analyze data and it is probable that this could have also infused some technical error (wrong data entry) in data sets provided by the service provider. Some limitations related to PySpark (Big data analysis framework) can also be mentioned such as there is limited support for machine learning algorithms as multi-processing is a key part involved in big data analysis and hence are not currently supported with Python backend. Exploratory data analysis also poses an issue while considering large data amounts as processing high numerical volume data could slow down the PySpark tool. Also, PySpark data frames do not support the latest data visualization libraries such as Seaborn, Plotly and hence need to be converted to Pandas data frames for using the above-mentioned libraries.

Area labeling framework

The area labeling framework presented in section 3.2 is used to produce true classes and is the key part of this master thesis. A lot of moving parts are involved in this labeling algorithm including open source libraries, publicly available open data sets, and methodical procedures involved. The most time-consuming part was to research a method to visualize the polygon represented by a cell on a geographical plane and to decide on a way to estimate population density inside that region. Finally, using a novel technique of juxtaposing innate hexagonal perception of cell structures and H3 hierarchical indexing, also it was possible to visualize and calculate cell coverage area population density as shown in figure 3.7.

It is worthwhile to notice that while dealing with spatial analyses, the method will always be prone to some measurement errors leading to a drop in confidence in spatial accuracy.

Nonetheless, Computer vision techniques might be able to reduce the errors even more and produce highly accurate results, which could be future work. The veracity of open data sets and libraries is also dependent on version changes and sometimes things may break/change accordingly. There may exist many different methodologies to achieve this task.

Machine learning system design

Standard data pre-processing steps were performed to create the data sets that are fit for machine learning analysis. Dropping duplicates, imputing null values, removing outliers, and feature scaling. Since telecom data is dynamic in nature, it is also important to pay attention to extreme values. Machine learning was performed as described in section 3.4 to infer if there does exist a relationship between area types and the features. It is important for the objective to also measure and evaluate the strength of the relationship between the classes. Model selection was based on a cross-validation technique that returns the best hyper-parameters for each algorithm and for each data set.

In this research, clustering is used to find patterns in user subscriber behavior and network service provider configurations in each area type. It is interesting to see that the results are as expected from the LTE study based on assumptions about node locations and subscriber traffic behavior. The features are used as input to an unsupervised learning model to create clusters with similar network characteristics. K-Means clustering for big data was performed which is described in detail in section 2.3 by determining the optimal number of clusters using silhouette score.

Replicability, Reliability, and Validity

The restricted size and the fact that verification of labeling quality of the data set from the service provider could not be verified may lead to reliability concerns. The data set is confidential and not publicly available which may further impact replicability. The ambition has always been that the method described in the thesis should, in theory, be applicable to any kind of cellular network provider. Unsupervised learning techniques have been used to find patterns in network traffic and create groups with similar behavior, which may be referred to as user subscriber behavior.

A study with a high degree of reliability has a large probability of leading to similar results if repeated and it has been the highest priority in this study to use the robust methodology and utilize open-source tools, techniques that are adaptable to different problems. In certain contexts, it may be the case that the most relevant information for the study is not to be found in the scientific literature but rather with individual software developers and open-source projects. The cited references, thus, play an important role in ascertaining the source.

Big data has its own properties signified by the four V's; Volume, Variety, Velocity, and Value, which makes many traditional analysis methods unsuitable. To be able to adapt to the changes in data analysis, it may need new methods and frameworks. One major concern for validity is that the proposed labeling framework and user traffic behavior may not be accurate enough to line up with real network behavior. A key question here is whether or not these results are applicable only to this data or network behavior in general. Another networking issue is the large uncertainty connected to network data. It is possible that radio conditions such as noise, latency, and connection issues may affect the results

5.3 Ethical Considerations

Data analytics can sometimes favor a specific group that might marginalize or leads to conflicts with other stakeholders. An example of this is from the book *Weapons of math destruction*, where the author Cathy O'Neil describes how over-reliance on mathematical models might upset the social structure [39]. Moving decision-making and data interpretation more

from humans towards machines may result in a system that tries to validate its own choices. The shift towards targeted and personalized data collection has been shown to produce discriminatory outcomes with disparate impact and material consequences [51]. The potential benefits of this research could include helping stakeholders with insights about different deployment scenario configurations and helping hardware production with better-optimizing units, while the potential risks could include violating the privacy and confidentiality of participants. Data anonymization has been done to protect the original location and specifics of node groups used by mobile service providers.

6 Conclusion

In this thesis, we have presented a framework for discovering user traffic patterns under different deployment scenarios in different area types such as Rural, Suburban, and Urban. It was discovered that the characteristic features of a variety of node groups inherent in a telecom network which are groups with similar characteristics may be used as building blocks for various types of network traffic. Below each research question has been attempted to answer using our methodology and results:

1. **How can spatial information be used to tag nodes and cells in area types?**

   A framework has been created using spatial information of node groups such as Latitude, Longitude, Azimuth angle, and cell range. The above parameters are combined mathematically to create real-world cell coverage area and using H3 spatial indexing along with USA Kontur Population data, area types were formulated. The decision to choose between six and three area-type classes was also confirmed using hypothesis testing between two groups. We have also performed sanity checks by visualizing ares type labels in a real-world geographical map which aligns superbly with the expectations.

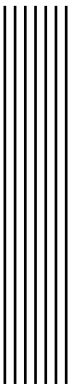
2. **How can machine learning be used to identify groups of nodes with similar characteristic behaviors in a telecom network?**

   By applying unsupervised learning algorithms to aggregated (data already available from Ericsson) LTE CM and PM data which are clearly defined in respective clusters. By applying cluster analysis to these groups we can identify essential features that are important as building blocks for subscriber traffic modeling. Cluster analysis provides us with actionable insights into user traffic behavior and configuration settings of node groups. It shows how each model behaves in different area types. Given this insight, we have the option to adapt to new kinds of traffic and subscriber modeling.

3. **What are the most prevalent common denominators between the groups in terms of model features and statistical profiles?**

   We have achieved this research topic by performing exhaustive data analysis and data exploration which clearly provides us with stand-outs in the behavior of configuration

and performance user traffic. We understand the feature importance by using the Random Forest Feature importance tool, which gives us information about how pathloss is one of the key elements in cellular networks.



Bibliography

[1] Michael A and Nielson. *Neural Networks and deep learning*. Determination Press, 2015.

[2] P. Marsch et al. "5G Radio Access Network Architecture: Design Guidelines and Key Considerations". In: *IEEE Communications Magazine* 54.11 (Nov. 2016), pp. 24–32.

[3] Saad Asif. *5g mobile communications: Concepts and technologies*. CRC Press, 2018.

[4] A. Bascacov, C. Cernazanu, and M. Marcu. "Using data mining for mobile communication clustering and characterization". In: *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2013, pp. 41–46. DOI: 10.1109/SACI.2013.6609004.

[5] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. "Traffic classification on the fly". In: *ACM SIGCOMM Computer Communication Review* 36.2 (2006), pp. 23–26.

[6] Uber Engineering Blog. *H3: Uber's Hexagonal Hierarchical Spatial Index*. https://www.uber.com/en-SE/blog/h3/. Accessed: 2023–03-02. 2018.

[7] R. Borralho, A. Mohamed, A. U. Quddusand P. Vieira, and R. Tafazolli. "A Survey on Coverage Enhancement in Cellular Networks: Challenges and Solutions for Future Deployments". In: *IEEE Communications Surveys & Tutorials* 23.2 (Jan. 2021).

[8] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[9] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. "Cart". In: *Classification and regression trees* (1984).

[10] Brandon Butcher, Smith, and Brian J. *Feature Engineering and Selection: A Practical Approach for Predictive Models: by Max Kuhn and Kjell Johnson. Boca Raton, FL Chapman & Hall/CRC Press, 2019, xv 297 pp., $79.95 (H), ISBN 978-1-13-807922-9*. 2020.

[11] Rafael Saraiva Campos. "Evolution of Positioning Techniques in Cellular Networks, from 2G to 4G". In: *Wireless Communications and Mobile Computing* . (Jan. 2017).

[12] Tata Communications. *What is a core network and how does it work?* https://www.tatacommunications.com/knowledge-base/network-core-network-explained/. Accessed: 2023–04-03. 2023.

[13] Chris Ding and Xiaofeng He. "K-Means Clustering via Principal Component Analysis". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 29. ISBN: 1581138385. DOI: 10.1145/1015330.1015408. URL: https://doi.org/10.1145/1015330.1015408.

[14] Guozhu Dong and Huan Liu. *Feature engineering for machine learning and data analytics*. CRC Press, 2018.

[15] Brad K. Donohoo, Chris Ohlsen, Sudeep Pasricha, Yi Xiang, and Charles Anderson. "Context-Aware Energy Enhancements for Smart Mobile Devices". In: *IEEE Transactions on Mobile Computing* 13.8 (2014), pp. 1720–1732. DOI: 10.1109/TMC.2013.94.

[16] Salim Dridi. "Unsupervised Learning - A Systematic Literature Review". In: (Dec. 2021).

[17] Margaret H Dunham. *Data mining: Introductory and advanced topics*. Pearson Education India, 2006.

[18] Rizanne Elbakly and Moustafa Youssef. "Crescendo: An Infrastructure-free Ubiquitous Cellular Network-based Localization System". In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 2019, pp. 1–6. DOI: 10.1109/WCNC.2019.8885420.

[19] Engineering and Technology History Wiki. *Cellular Base Stations*. https://ethw.org/Cellular_Base_Stations. Accessed: 2023–03-02. 2015.

[20] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas. "LTE-advanced: next-generation wireless broadband technology [Invited Paper]". In: *IEEE Wireless Communications* 17.3 (June 2010), pp. 10–12.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[22] Cyril Goutte and Eric Gaussier. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation". In: *Advances in Information Retrieval*. Ed. by David E. Losada and Juan M. Fernández-Luna. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 345–359. ISBN: 978-3-540-31865-1.

[23] Joel Grus. *Data Science from Scratch, 2nd Edition*. USA: O'Reilly Media, Inc., 2019, pp. 99–100. ISBN: 9781492041139.

[24] Martin Happ, Arne C. Bathke, and Edgar Brunner. "Optimal sample size planning for the Wilcoxon-Mann-Whitney test". In: *National Library of Medicine* (Feb. 2019).

[25] Trevor Hastie, Tibshirani Robert, and Friedman Jerome. *The Elements of Statistical Learning*. Springer New York, 2009.

[26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[27] E. Hossain and M. Hasan. "5G cellular: key enabling technologies and research challenges". In: *IEEE Instrumentation & Measurement Magazine* 18.3 (June 2015), pp. 11–21.

[28] Mohammad Hossin and MN Sulaiman. "A review on evaluation metrics for data classification evaluations". In: *International Journal of Data Mining & Knowledge Management Process* 5.2 (2015), p. 1.

[29] European Telecommunications Standards Institute. *Area type proposal*. https://www.etsi.org/standards/Networks. Accessed: 2023–24-05. 1988.

[30] Mingers J. "An empirical comparison of selection measures for decision-tree induction". In: Springer, Mar. 1989.

[31] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

[32] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. "Machine learning paradigms for next-generation wireless networks". In: *IEEE Wireless Communications* 24.2 (2016), pp. 98–105.

[33] S. Jimaa, Kok Keong Chai, Yue Chen, and Y. Alfadhl. "LTE-A an overview and future research areas". In: *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (Nov. 2011), pp. 395–399.

[34] Pradnya Kamble, Dixit Jain, Mehul Jain, Vrushab Jain, and Sohil Mehta. "LTE Network Coverage Area". In: *International Journal of Recent Technology and Engineering* 3 (2015).

[35] H3 library. *Cell statistics across resolutions*. https://h3geo.org/docs/core-library/restable. Accessed: 2023–03-02. 2023.

[36] Henry B Mann and Donald R Whitney. "On a test of whether one of two random variables is stochastically larger than the other". In: *The annals of mathematical statistics* (1947), pp. 50–60.

[37] Satoshi Maruyama, Katsuhiko Tanahashi, and Takehiko Higuchi. "Base Transceiver Station for W-CDMA System". In: 2003.

[38] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. "A First Look at Commercial 5G Performance on Smartphones". In: *Proceedings of The Web Conference 2020*. WWW '20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 894–905. ISBN: 9781450370233. DOI: 10.1145/3366423.3380169. URL: https://doi.org/10.1145/3366423.3380169.

[39] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. USA: Crown Publishing Group, 2016. ISBN: 0553418815.

[40] 3GPP Organization. *About 3GPP*. https://www.3gpp.org/about-3gpp. Accessed: 2023–23-03. 1998.

[41] Shraddha Pandit and Suchita Gupta. "A Comparative Study on Distance Measuring Approaches for Clustering". In: *International Journal of Research in Computer Science* 2 (Dec. 2011), p. 29. DOI: 10.7815/ijorcs.21.2011.011.

[42] Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. "Hyperparameters and tuning strategies for random forest". In: *WIREs Data Mining and Knowledge Discovery* 9.3 (Jan. 2019). DOI: 10.1002/widm.1301. URL: https://doi.org/10.1002%5C%2Fwidm.1301.

[43] Philip Sedgwick. "Non-parametric statistical tests for two independent groups: numerical data". In: *BMJ* 348 (2014). DOI: 10.1136/bmj.g2907. eprint: https://www.bmj.com/content/348/bmj.g2907.full.pdf. URL: https://www.bmj.com/content/348/bmj.g2907.

[44] Ketan Rajshekhar Shahapure and Charles Nicholas. "Cluster Quality Analysis Using Silhouette Score". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 2020, pp. 747–748. DOI: 10.1109/DSAA49011.2020.00096.

[45] Rahul Kumar Sharma and Ashish Dewangan. "Coverage and Rate Probability in Hexagonal Cell Structure". In: *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)* 02 (11 Nov. 2013).

[46] S. K. Singh, R. Singh, and B. Kumbhani. "The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities". In: *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (June 2020).

[47] Sergios Theodoridis and Konstantinos Koutroumbas. “Chapter 6 - Feature Generation I: Data Transformation and Dimensionality Reduction”. In: *Pattern Recognition (Fourth Edition)*. Ed. by Sergios Theodoridis and Konstantinos Koutroumbas. Fourth Edition. Boston: Academic Press, 2009, pp. 323–409. ISBN: 978-1-59749-272-0. DOI: https://doi.org/10.1016/B978-1-59749-272-0.50008-6. URL: https://www.sciencedirect.com/science/article/pii/B9781597492720500086.

[48] K Tulankar, M Kshirsagar, and R Wajgi. “Clustering telecom customers using emergent self organizing maps for business profitability”. In: *International Journal of Computer Science and Technology* 3.1 (2012), pp. 256–259.

[49] Edy Umargono, Jatmiko Suseno, and S.K Gunawan. “K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula”. In: Jan. 2020. DOI: 10.2991/assehr.k.201010.019.

[50] Yongfeng Wang, Xinzhou Cheng, Lexi Xu, Jian Guan, Tao Zhang, and Mingjun Mu. “A novel complaint calls handle scheme using big data analytics in mobile networks”. In: *Signal and Information Processing, Networking and Computers: Proceedings of the 1st International Congress on Signal and Information Processing, Networking and Computers (ICSINC 2015), October 17-18, 2015 Beijing, China*. CRC Press. 2016, p. 347.

[51] Madisson Whitman, Chien-yi Hsiang, and Kendall Roark. “Potential for participatory big data ethics and algorithm design: a scoping mapping review”. In: *Proceedings of the Participatory Design Conference: Short Papers, Situated Actions, Workshops and Tutorial-Volume 2*. 2018, pp. 1–6.

[52] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. “Primary user behavior in cellular networks and implications for dynamic spectrum access”. In: *IEEE Communications Magazine* 47.3 (Mar. 2009), pp. 88–95.

[53] Ian H. Witten, Eibe Frank, and Mark A. Hall. “Chapter 7 - Data Transformations”. In: *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*. Ed. by Ian H. Witten, Eibe Frank, and Mark A. Hall. Third Edition. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann, 2011, pp. 305–349. ISBN: 978-0-12-374856-0. DOI: https://doi.org/10.1016/B978-0-12-374856-0.00007-9. URL: https://www.sciencedirect.com/science/article/pii/B9780123748560000079.

[54] Lexi Xu, Yue Chen, KoK Keong Chai, Tiankui Zhang, John Schormans, and Laurie Cuthbert. “Cooperative load balancing for OFDMA cellular networks”. In: *European Wireless 2012; 18th European Wireless Conference 2012*. 2012, pp. 1–7.

[55] Qianlan Ying, Zhifeng Zhao, Yifan Zhou, Rongpeng Li, Xuan Zhou, and Honggang Zhang. “Characterizing spatial patterns of base stations in cellular networks”. In: *2014 IEEE/CIC International Conference on Communications in China (ICCC)*. 2014, pp. 490–495. DOI: 10.1109/ICCChina.2014.7008327.

[56] W. R. YOUNG. “Advanced Mobile Phone Service”. In: *THE BELL SYSTEM TECHNICAL JOURNAL* (1979).

[57] Mohamed Younis and Kemal Akkaya. “Strategies and techniques for node placement in wireless sensor networks: A survey”. In: *Ad Hoc Networks* 6.4 (2008), pp. 621–655. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2007.05.003. URL: https://www.sciencedirect.com/science/article/pii/S1570870507000984.