18MAT333

GRAPH ANALYSIS AND ALGORITHM

CASE STUDY

DATASET: 5G NETWORK DEPLOYMENTS DATA

Mathangi Sriram – CB.SC.I5DAS21039

R Arivanandam – CB.SC.I5DAS21114

ABSTRACT:

The dataset provides insight into the geographical distribution of deployments and status of telecom operators, primarily Airtel and Reliance Jio, across various cities in India. It includes information such as the operator's name, city presence, status (commercial availability or pre-release), and geographical coordinates. This dataset offers opportunities for analysing network coverage, market penetration, and performance comparison among operators in different regions. Geospatial analysis could unveil patterns of deployment and potential areas for expansion. Such analysis could inform strategic decisions for network optimization and market expansion strategies within the Indian telecommunications landscape.

DATASET INFORMATION:

The dataset provides a comprehensive overview of the presence and status of deployments of different telecom operators which provided 5G network service by mid 2023, primarily Airtel and Reliance Jio, across multiple cities in India. Each entry includes details such as the operator's name, the city where their services are available, and the status of these services, whether they are commercially available or in a pre-release phase. Additionally, geographical coordinates in the form of longitude and latitude are provided for each city, enabling geospatial analysis.

ABOUT THE DATA:

The source data was from ookla speed test site which was in the form of a geo-json file. It was converted to csv file for better understanding of data. It comprises of columns namely unique_id, operator, city_name, status, latitude and longitude. Where operator tells us about operator present in the specific deployment.

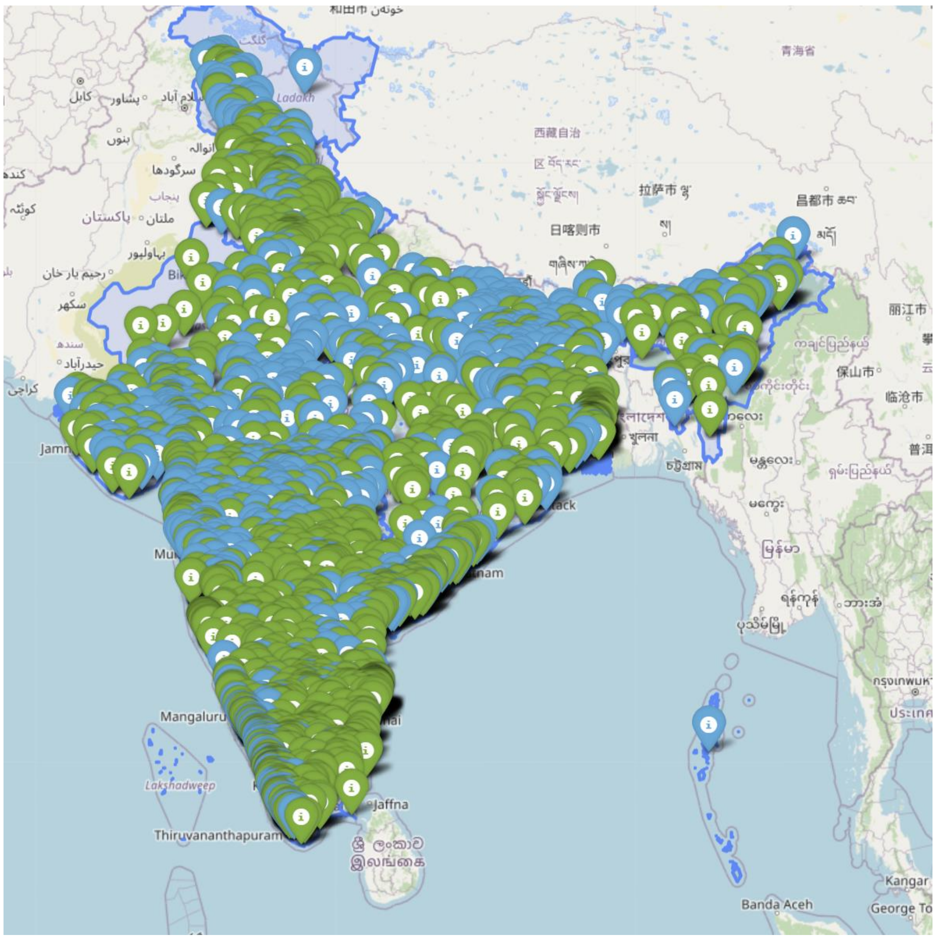
SOURCE: https://www.speedtest.net/ookla-5g-map

OBJECTIVES:

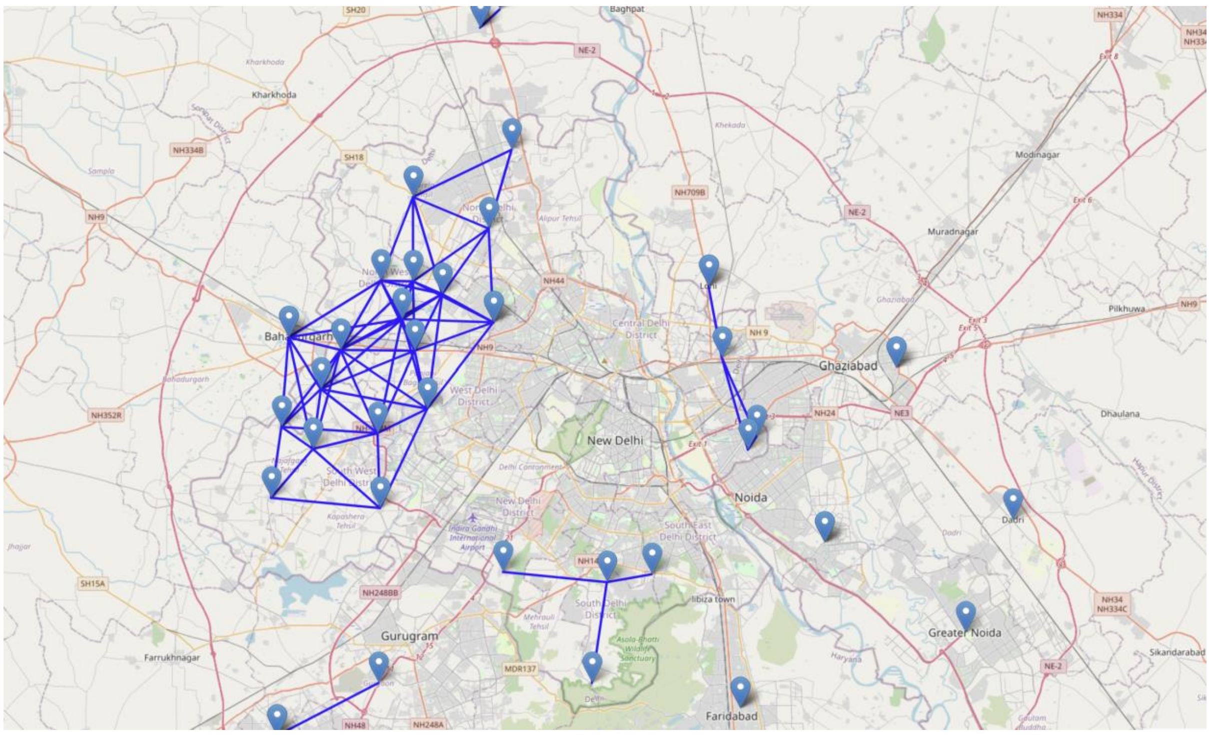
The objective of this analysis is to examine the geographical distribution of 5G deployments by major telecom operators in India, specifically Airtel and Reliance Jio. The focus is on predicting the operator based on geographical coordinates using machine learning models like K-Nearest Neighbors and Graph Convolutional Networks. The aim is to provide insights into optimal operator recommendations for users based on their location and to identify deployment patterns for strategic decision-making.

GENERAL OBSERVATION:

We can generally see that the 5G deployments are clustered primarily in urban areas where 5G access is in demand than in rural areas.



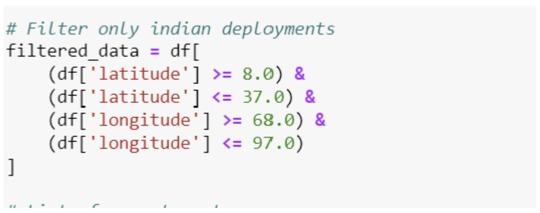
AFTER ZOOMING:



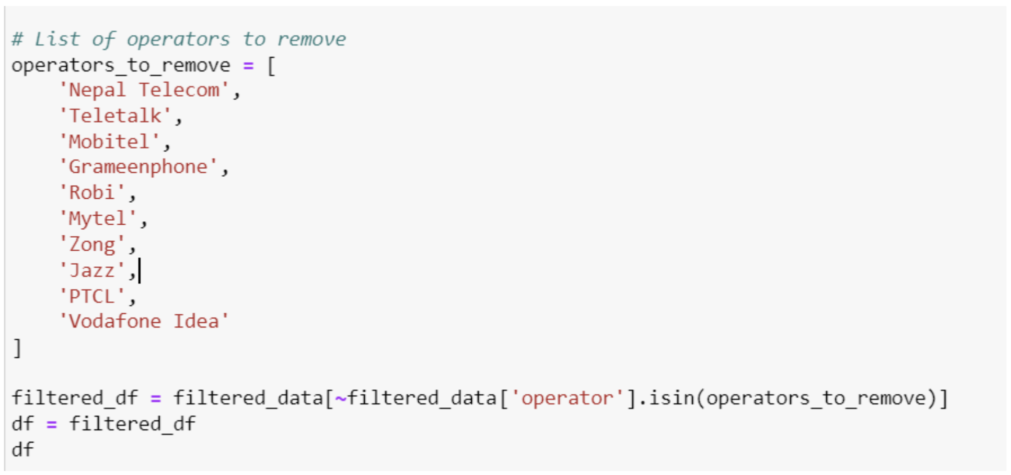
PRE-PROCESSING:

The data from ookla was for the deployment of 5G towers present all over the world we cleaned the data in such that only deployments which are found in India is analysed. This is done by several methods as follows:

Getting the points inside the boundaries of India.

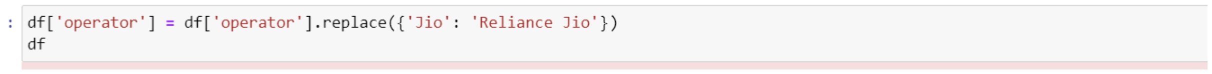

```
# Filter only indian deployments
filtered_data = df[
    (df['latitude'] >= 8.0) &
    (df['latitude'] <= 37.0) &
    (df['longitude'] >= 68.0) &
    (df['longitude'] <= 97.0)
]
```

Removing operator which are not present in India. Note-even though Vodaphone Idea present in India still date they don’t have 5G deployments.



```
# List of operators to remove
operators_to_remove = [
    'Nepal Telecom',
    'Teletalk',
    'Mobitel',
    'Grameenphone',
    'Robi',
    'Mytel',
    'Zong',
    'Jazz',
    'PTCL',
    'Vodafone Idea'
]

filtered_df = filtered_data[~filtered_data['operator'].isin(operators_to_remove)]
df = filtered_df
df
```

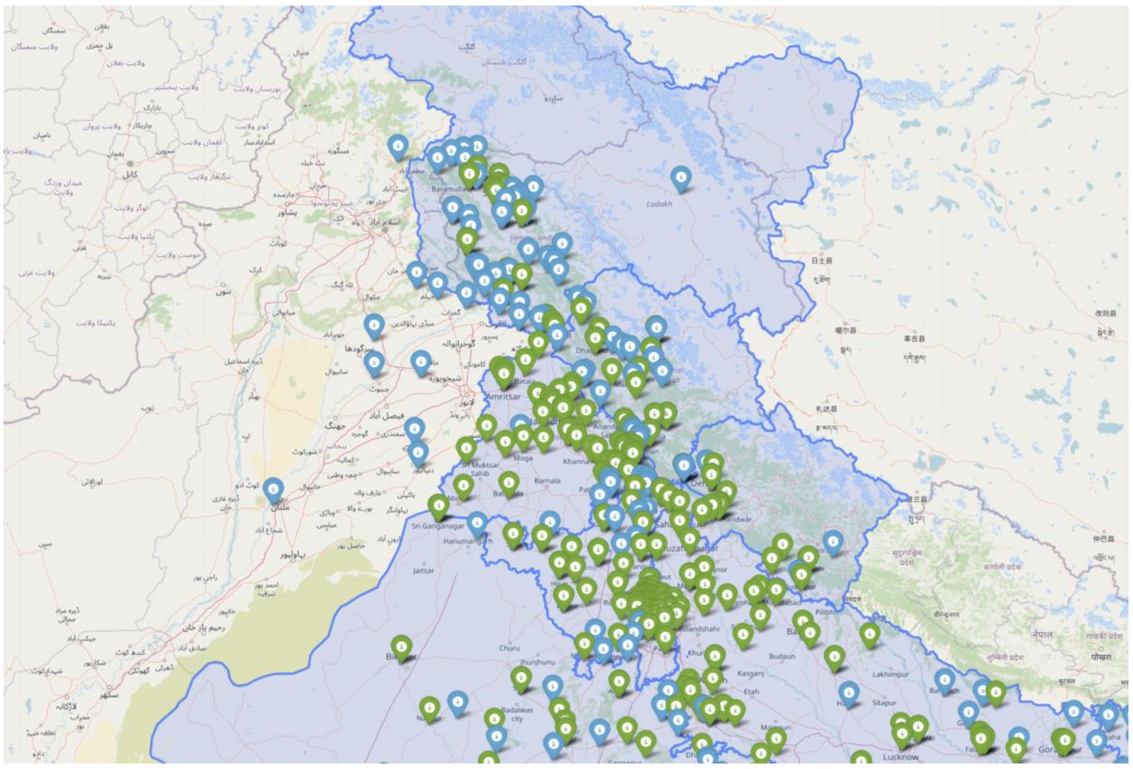
Renaming operators which will help while grouping.



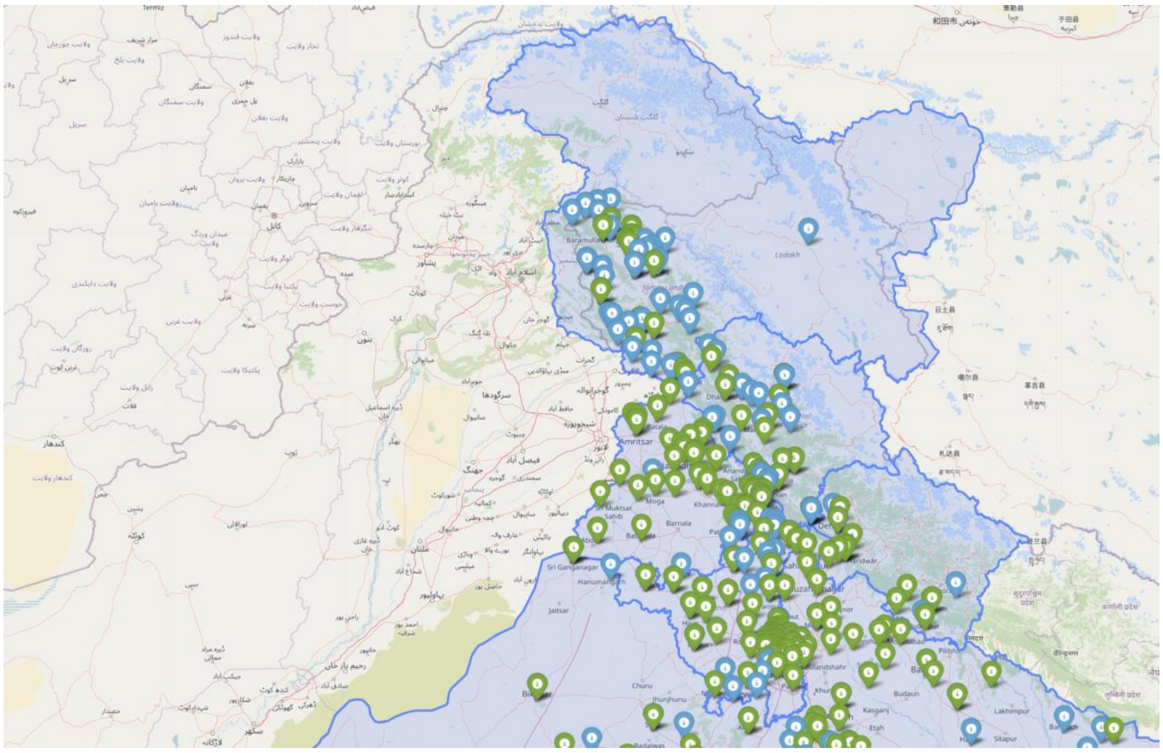
```
: df['operator'] = df['operator'].replace({'Jio': 'Reliance Jio'})
  df
```

Manually removing the data points by visualising and using states Geo-Json file of India

Before:



After:



Thus acquired the desired data with 2613 rows × 5 columns which comprises of Deployments present only in India.

CONSTRUCTING EDGES:

The theory behind constructing edges is that each data tower will be able to communicate with the other at 10 kilometres in an average. Thus, the data towers of same operators will be able to communicate at an average distance of 10 km. Thus, an edge is constructed between them.

CENTRALITIES:

Centrality measures are metrics used to quantify the relative importance or influence of nodes within a network.

These measures provide insights into the structural significance of nodes based on various criteria.

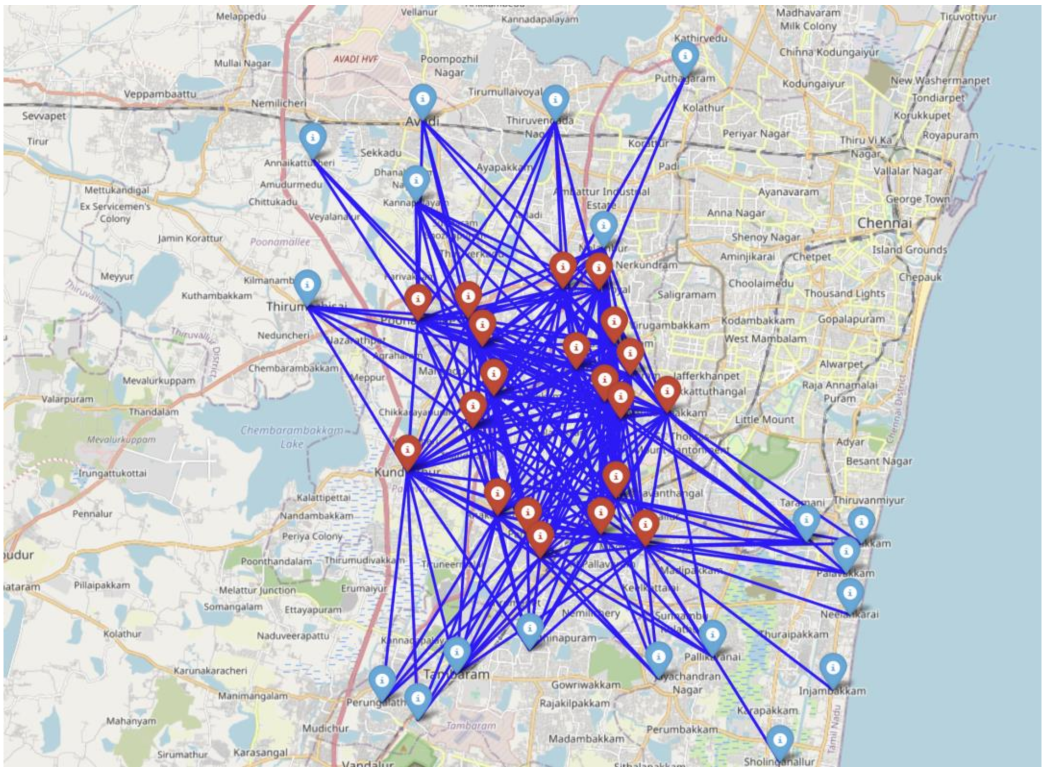
They are of many types, here we are discussing about

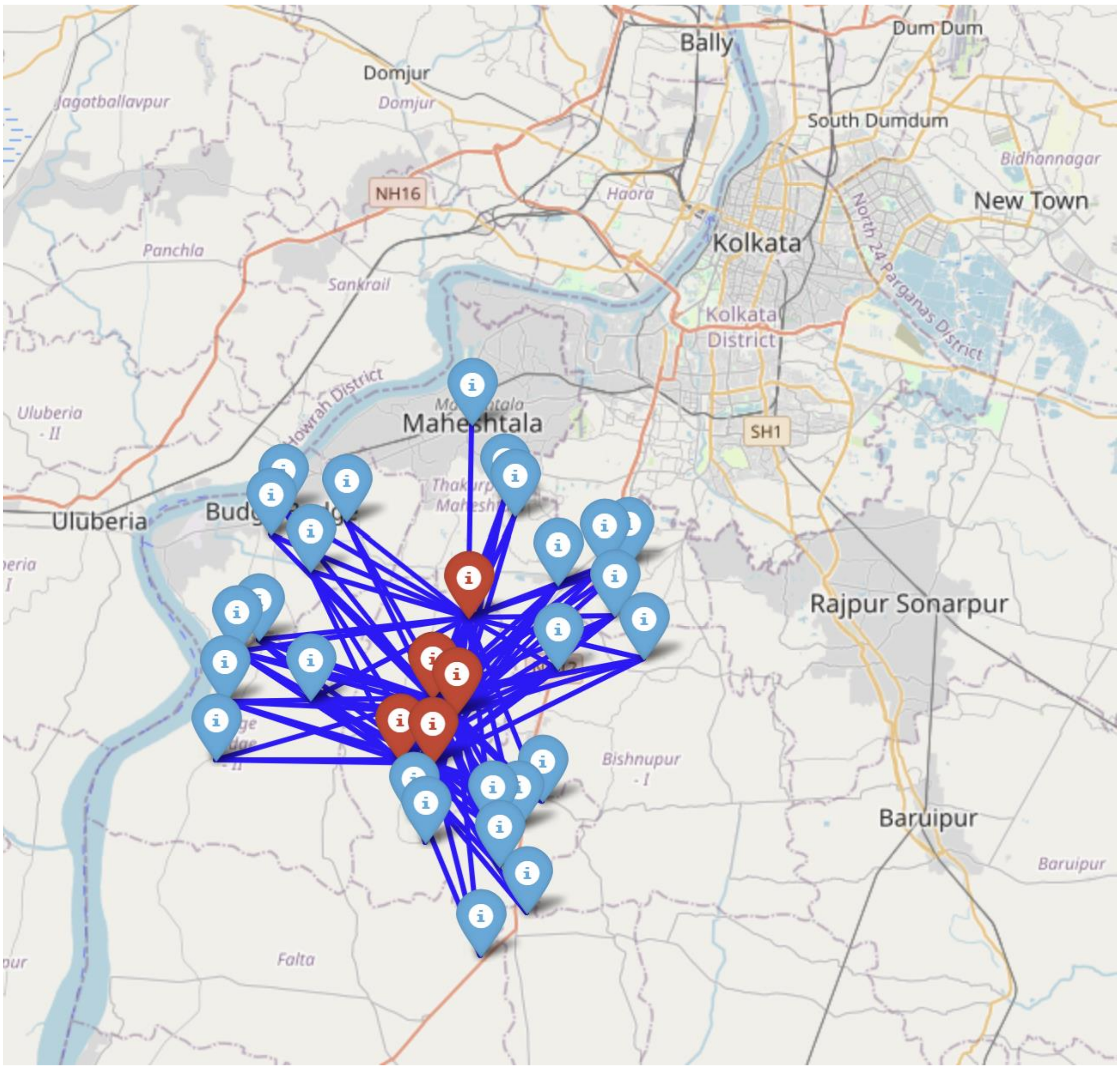
- Degree centrality
- Closeness centrality
- Betweenness centrality
- Eigen-vector centrality

Degree Centrality:

Degree centrality of a node is the number of edges connected to it. Higher degree centrality nodes are more connected and often considered more influential within the network.

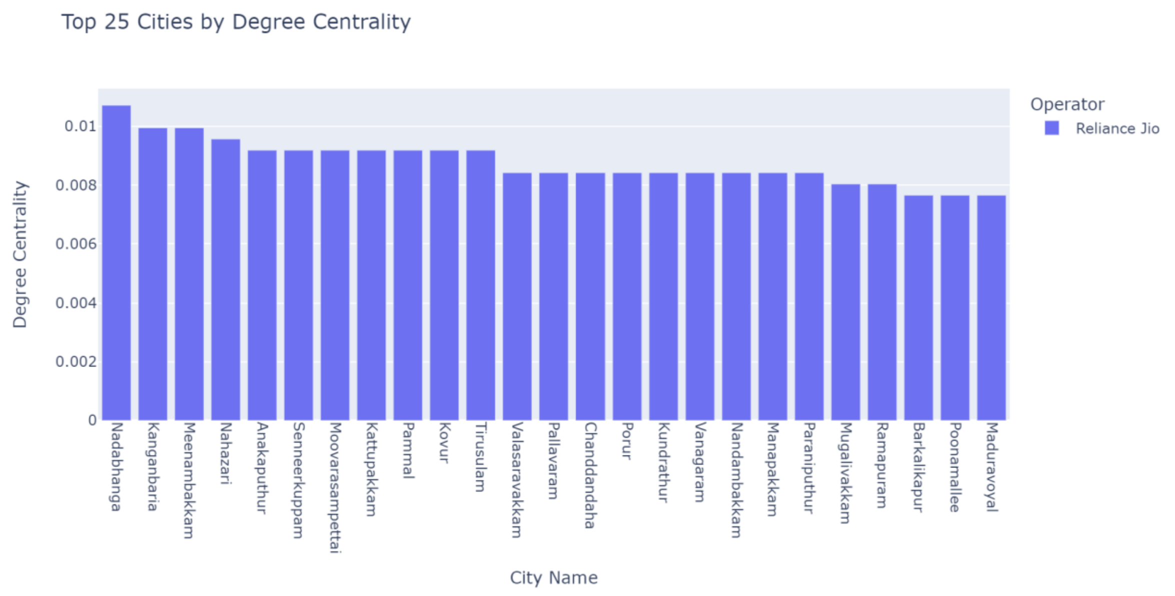
In our data we have detected the top 25 influential nodes with respect to their degree centrality. These node's neighbouring nodes and edges are plotted on top of a map to give us an idea about the 5g network connectivity.





We can infer from graph that the top 25 nodes are present in the outskirts of Chennai and Kolkata. They are both metropolitan cities and most of whose IT industry is present in the outskirts. This could be a major reason for the higher degree centrality nodes present in these regions.

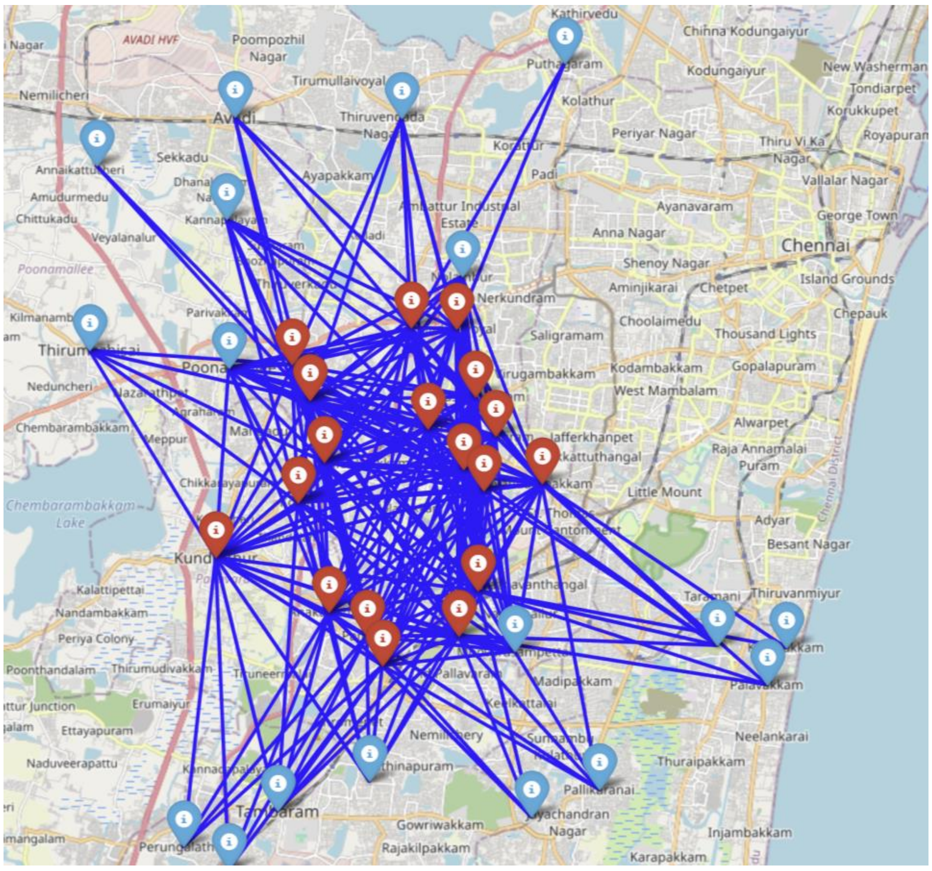
The below bar graph showcases the degree centrality for each.

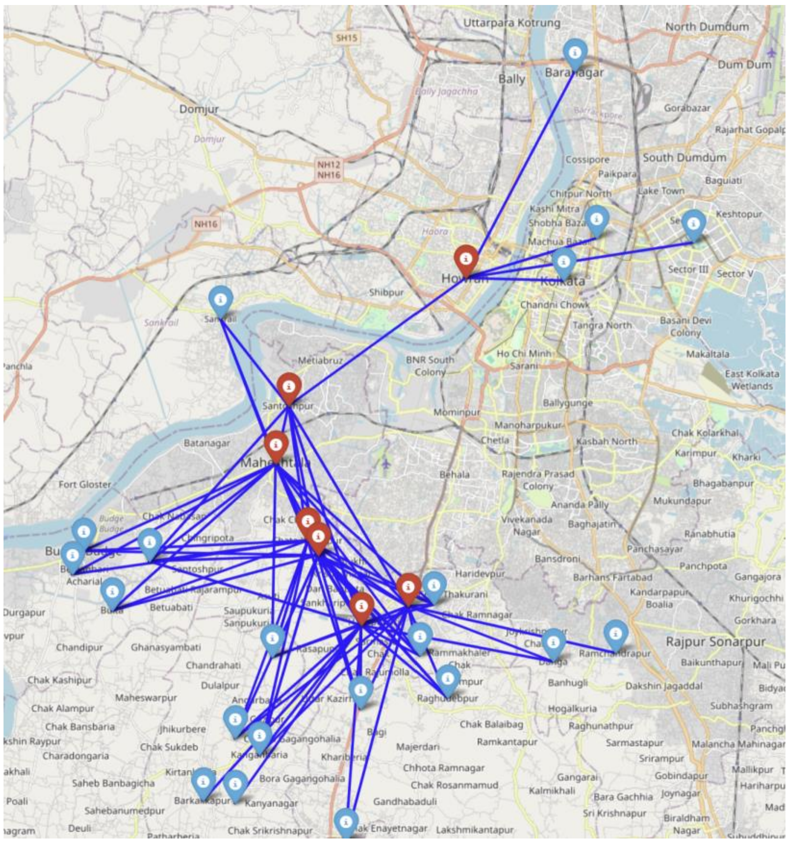


Closeness Centrality:

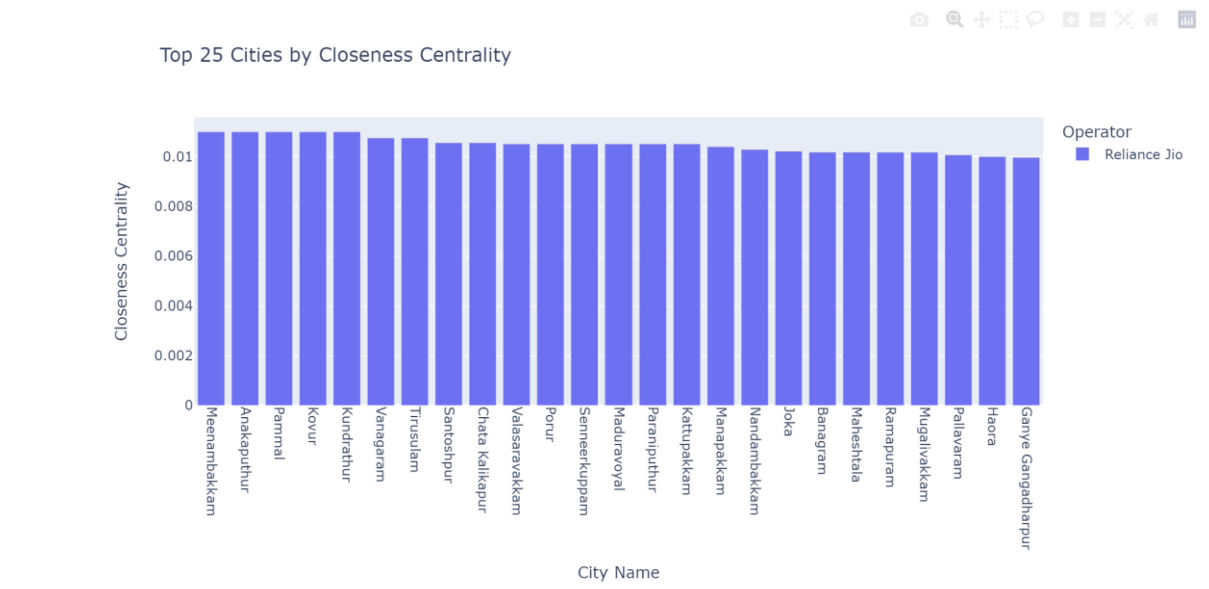
Closeness centrality quantifies how close a node is to all other nodes in the network. Nodes with higher closeness centrality are considered more central and have shorter average distances to all other nodes, indicating their greater accessibility within the network.

In our dataset, we have identified the top 25 influential nodes based on their closeness centrality. These nodes serve as important points within the network, as they are strategically positioned to efficiently transfer information or resources to other nodes. Visualizing the neighbouring nodes and edges of these influential nodes on a map provides insights into the spatial distribution and connectivity of the 5G network infrastructure.





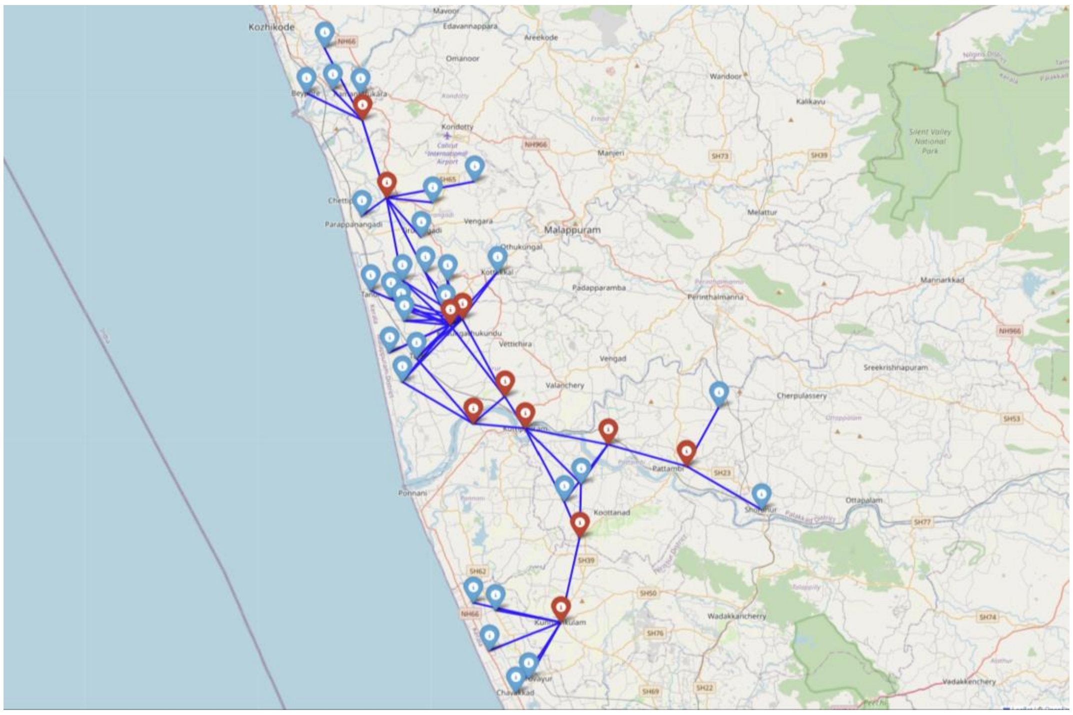
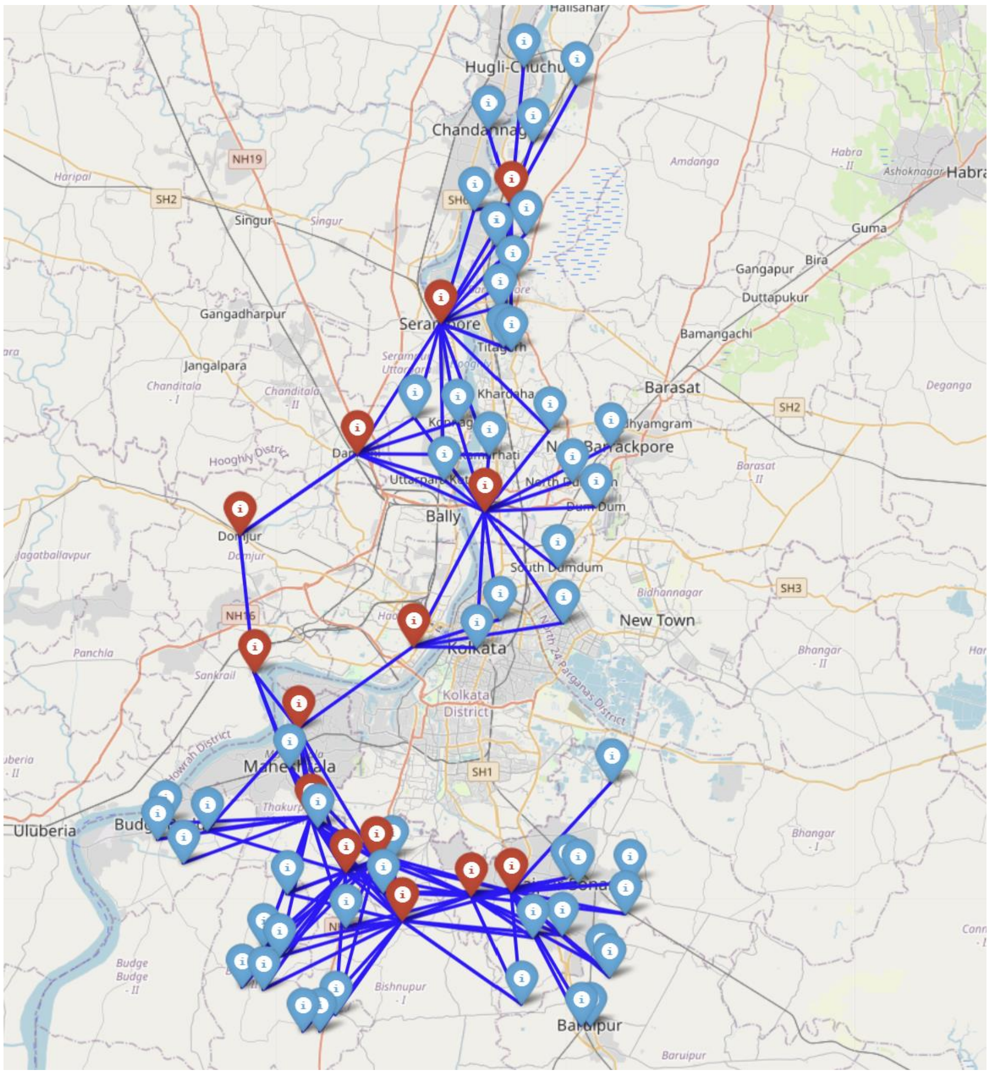
A noticeable trend in closeness centrality mirrors that observed in degree centrality within the outskirts of Kolkata and Chennai.



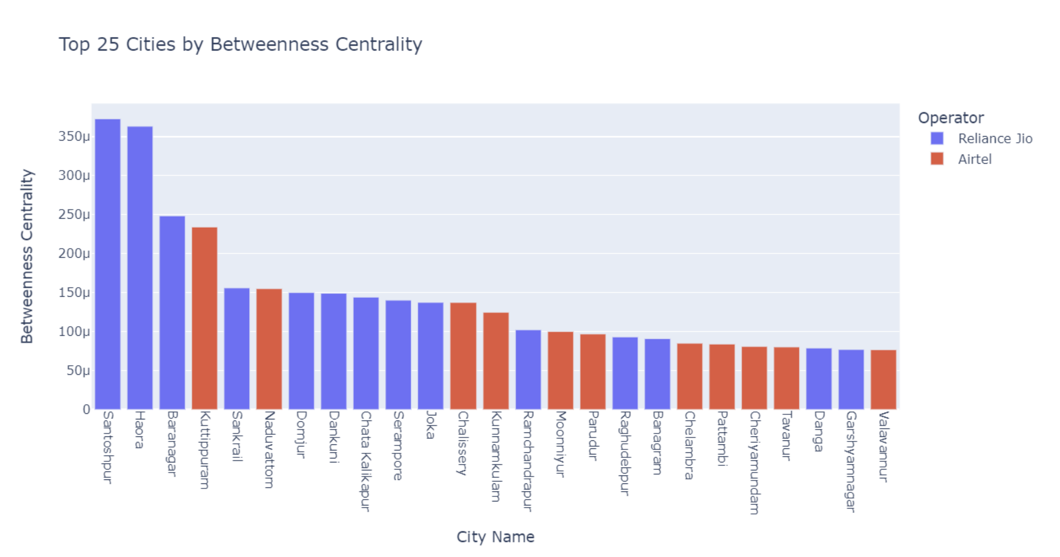
Betweenness centrality:

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network. Nodes with higher betweenness centrality play a significant role in facilitating communication and information flow across the network, acting as bridges.

In our dataset, we have identified the top 25 nodes with the highest betweenness centrality. These nodes are crucial in regulating the flow of information within the 5G network, serving as key connectors between various locations.


Kozhikode
Malappuram
Ponnani
Pattambi
Shoranur
Kunnamkulam
Chavakkad

Hugli-Chuchura
Chandannagar
Serampore
Titagarh
Barasat
New Barrackpore
Dum Dum
Bally
Domjur
Kolkata
New Town
Maheshtala
Budge Budge
Uluberia
Baruipur

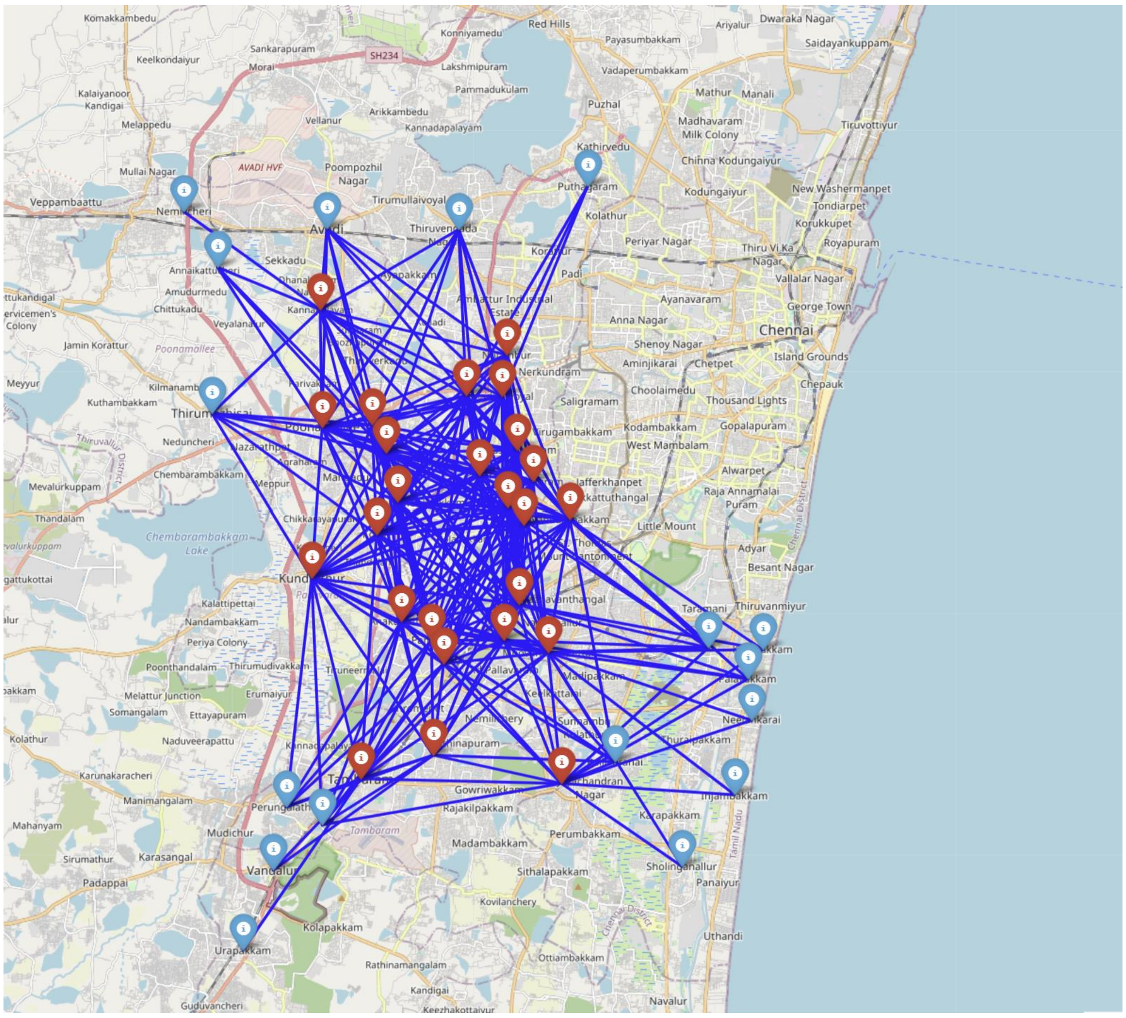
We can see that both in Kolkata and Kerala there is a good linkage connecting various towers. We can also see from the below Airtel has a good connectivity in Kerala whereas Reliance Jio has a good connectivity in Kolkata.



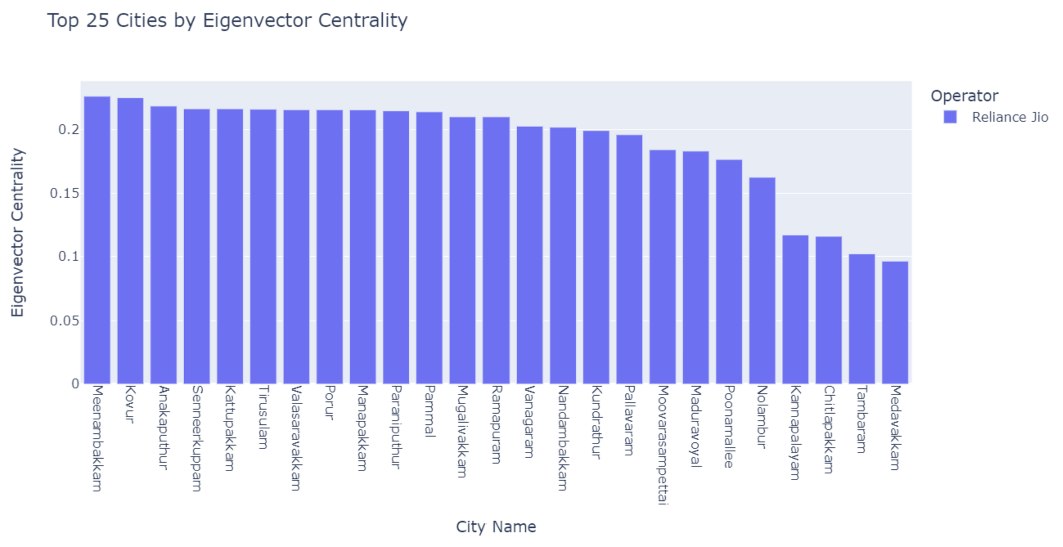
Eigen-vector Centrality:

Eigen vector centrality measures the importance of a node in a network by considering both its direct connections and the connections of its neighbours. Nodes with higher eigen vector centrality are not only well-connected themselves but are also connected to other nodes that are themselves important in the network. This centrality metric identifies nodes that have a significant influence over the network due to their strategic positioning within the network structure.

We have identified the top 25 nodes with the highest eigen vector centrality. These nodes play a crucial role in shaping the flow of information and communication within the 5G network. Visualizing these nodes and their connections on a map provides valuable insights into the key players driving network dynamics and connectivity patterns.



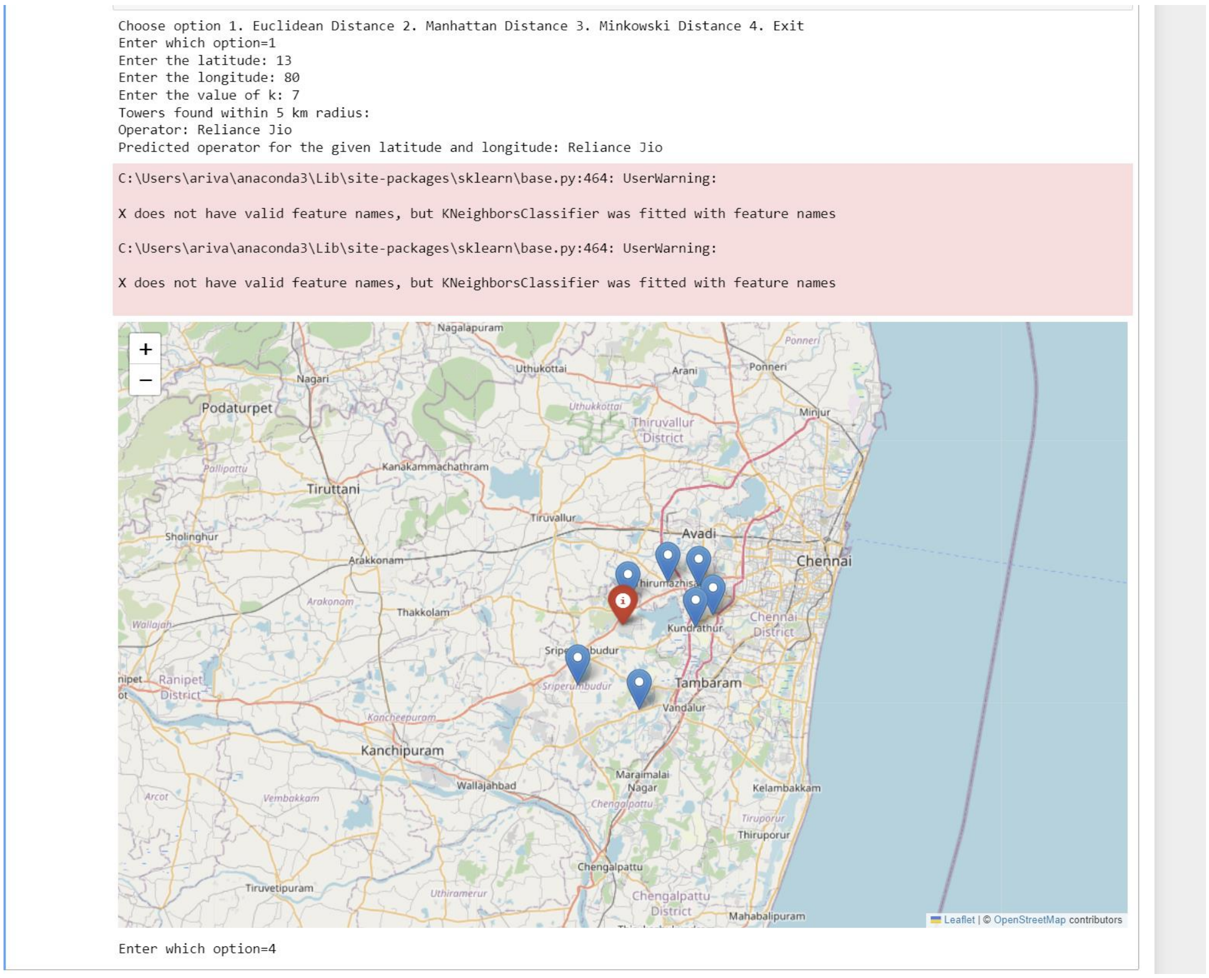
We can infer that in Chennai it is not only well-connected themselves but are also connected to other nodes that are themselves important in the network.



MACHINE LEARNING MODELS:

K-NEAREST NEIGHBOURS:

With the help of the location of the user we will be able to identify the best operator for the user using K-nearest neighbour ml model. We will first initialize the minimum range to be 5km since it is the maximum range at which 5G can work then find the k nearest operators for the location and suggest the best operator for the user



```
Choose option 1. Euclidean Distance 2. Manhattan Distance 3. Minkowski Distance 4. Exit
Enter which option=1
Enter the latitude: 13
Enter the longitude: 80
Enter the value of k: 7
Towers found within 5 km radius:
Operator: Reliance Jio
Predicted operator for the given latitude and longitude: Reliance Jio
C:\Users\ariva\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning:

X does not have valid feature names, but KNeighborsClassifier was fitted with feature names

C:\Users\ariva\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning:

X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
```

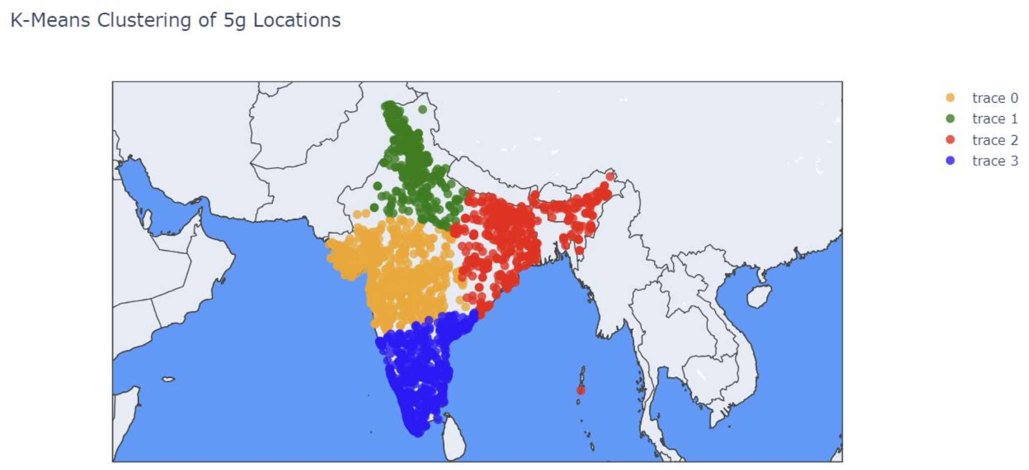
```
Enter which option=4
```

Thus, for the given location Reliance Jio is the best operator to use 5G.

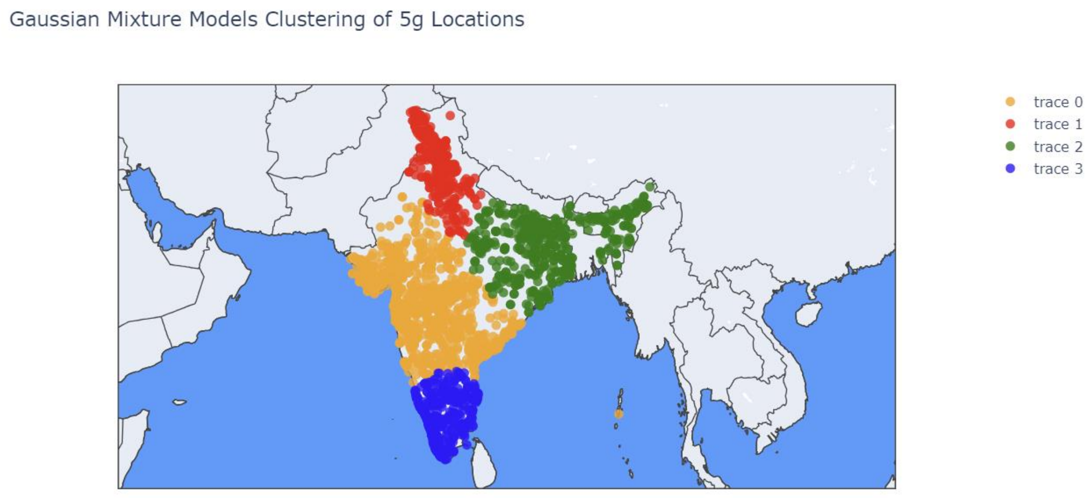
COMPARISION OF UNSUPERVISED ML MODELS

When considering the deployment's data points, grouping them strategically could enhance administration and service practices. We will see what differences are seen in each model.

1) K-means clustering



2) Gaussian mixture model

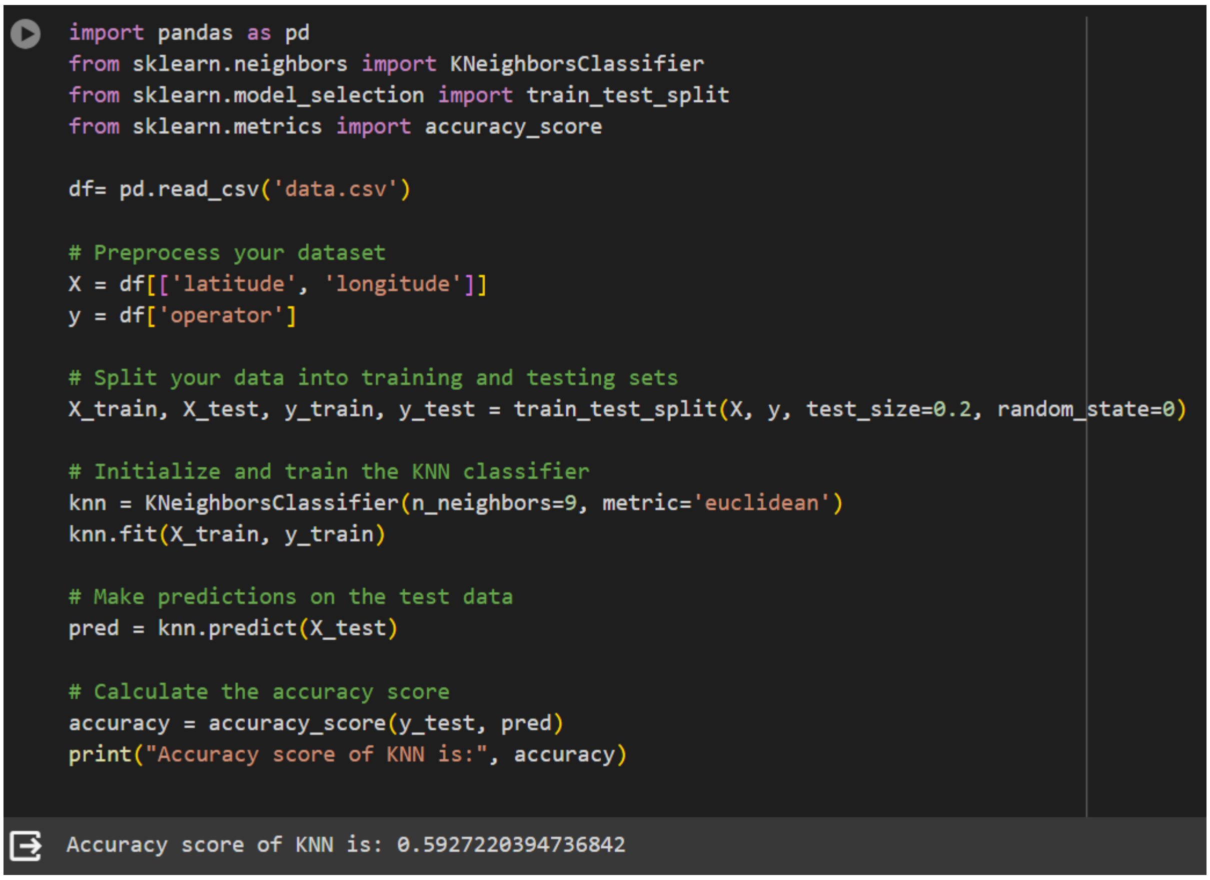


From comparing both the models we can see that the central region takes up more area in gaussian model that in k-means clustering this potentially due to the more overlapping of data points in the southern and central region of India.

Graph Convolutional Network:

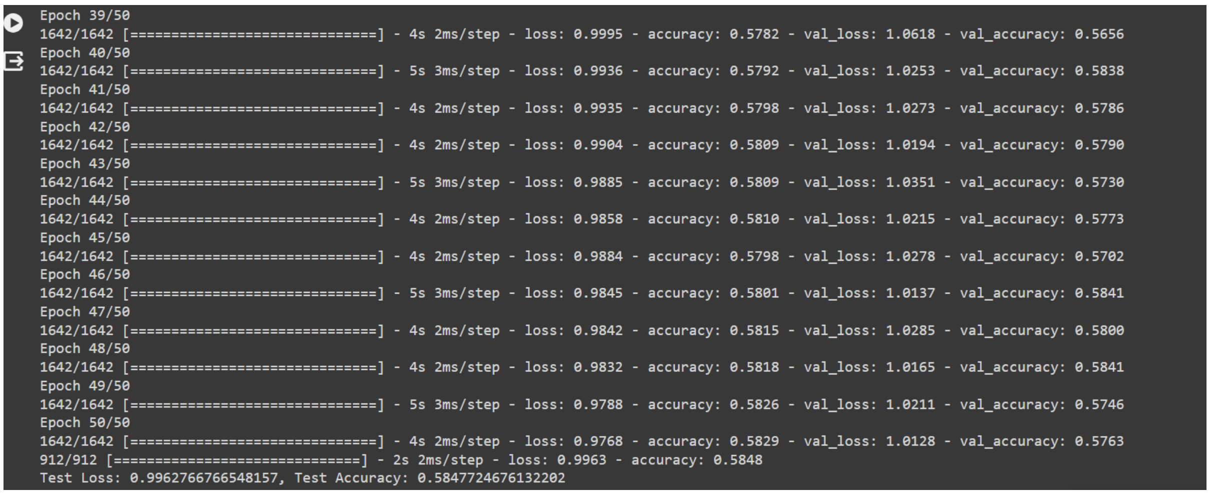
GCN, or Graph Convolutional Network, is a type of neural network tailored for data represented as graphs. In a graph, nodes represent entities, and edges signify relationships between them. GCNs operate by applying convolutional operations on these graph structures, allowing them to gather information not only from individual nodes but also from their neighbouring nodes.

Considering our dataset, the Graph Convolutional Network (GCN) will predict the operator (Airtel or Reliance Jio) based on latitude and longitude coordinates. GCNs are ideal for graph-structured data, where nodes represent entities (like geographic points) and edges show relationships between them. By leveraging spatial connections between nearby points, the GCN could learn to associate operators with specific locations. Latitude and longitude coordinates serve as features, while the operator is the target. The model learns spatial patterns to predict operators for new locations.

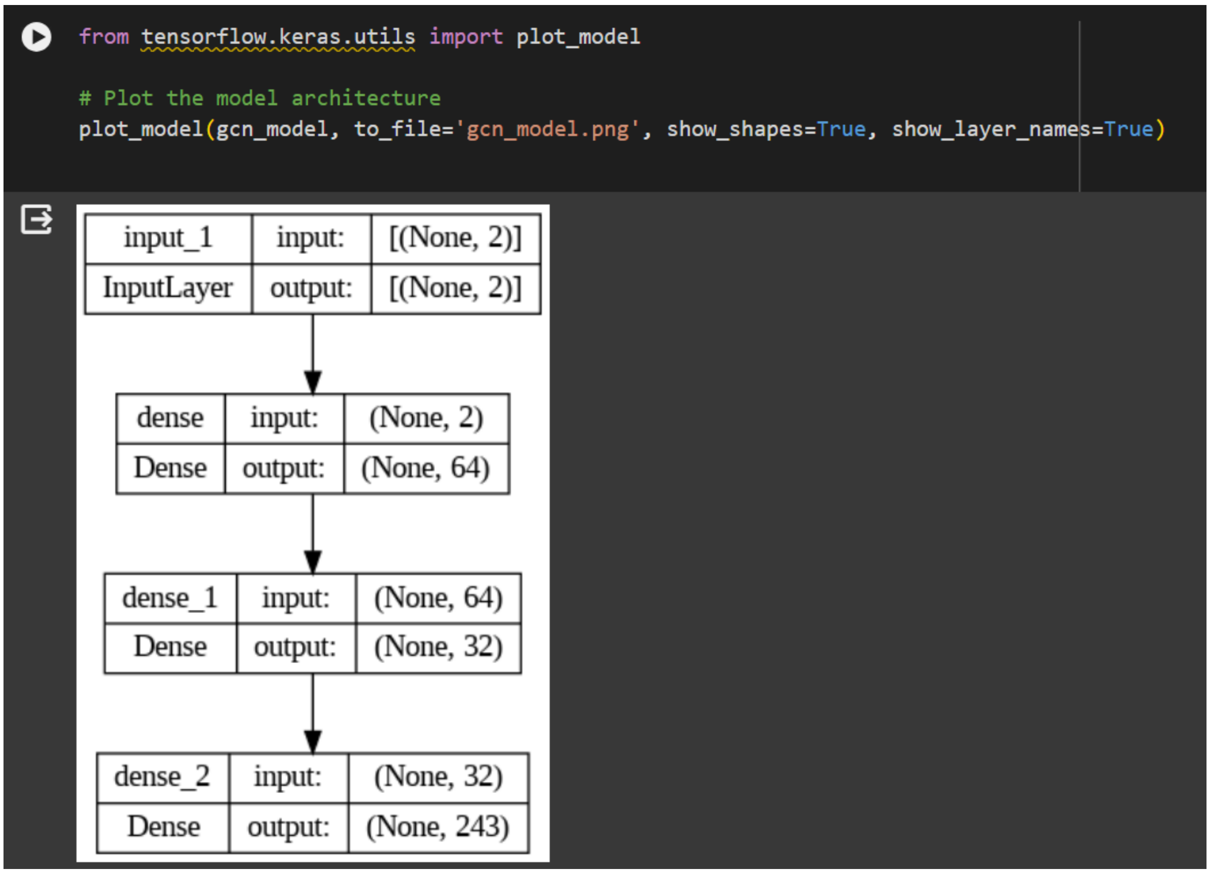


```python
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df= pd.read_csv('data.csv')

# Preprocess your dataset
X = df[['latitude', 'longitude']]
y = df['operator']

# Split your data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Initialize and train the KNN classifier
knn = KNeighborsClassifier(n_neighbors=9, metric='euclidean')
knn.fit(X_train, y_train)

# Make predictions on the test data
pred = knn.predict(X_test)

# Calculate the accuracy score
accuracy = accuracy_score(y_test, pred)
print("Accuracy score of KNN is:", accuracy)
```

```
Accuracy score of KNN is: 0.5927220394736842
```

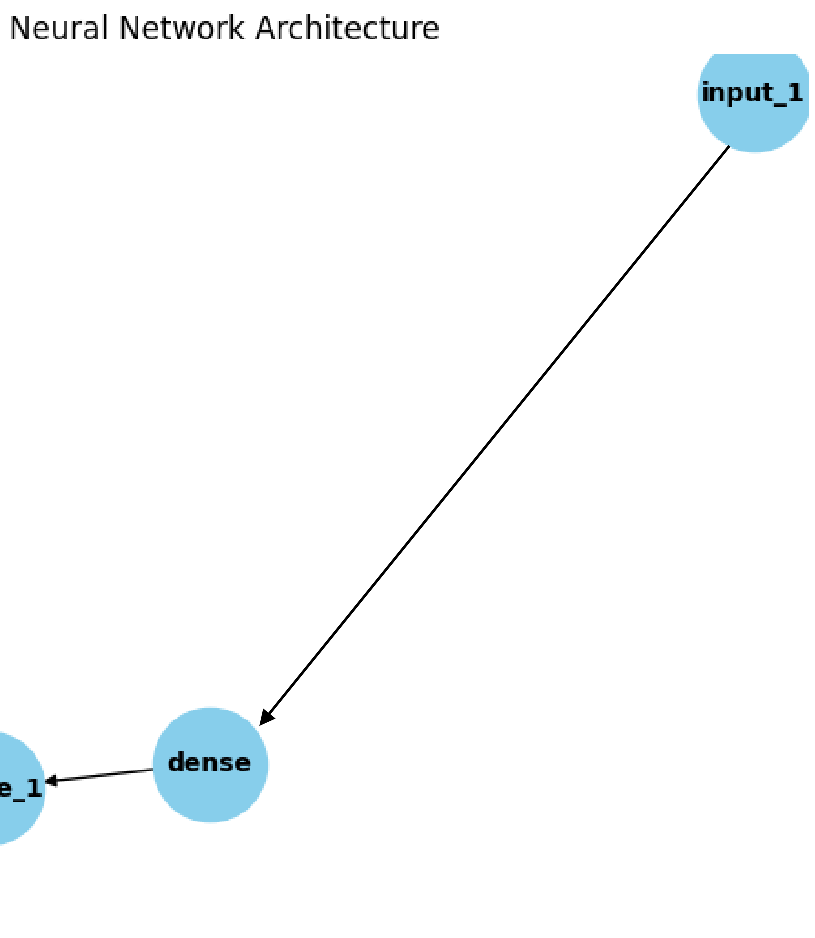


```
Epoch 39/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9995 - accuracy: 0.5782 - val_loss: 1.0618 - val_accuracy: 0.5656
Epoch 40/50
1642/1642 [==============================] - 5s 3ms/step - loss: 0.9936 - accuracy: 0.5792 - val_loss: 1.0253 - val_accuracy: 0.5838
Epoch 41/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9935 - accuracy: 0.5798 - val_loss: 1.0273 - val_accuracy: 0.5786
Epoch 42/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9904 - accuracy: 0.5809 - val_loss: 1.0194 - val_accuracy: 0.5790
Epoch 43/50
1642/1642 [==============================] - 5s 3ms/step - loss: 0.9885 - accuracy: 0.5809 - val_loss: 1.0351 - val_accuracy: 0.5730
Epoch 44/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9858 - accuracy: 0.5810 - val_loss: 1.0215 - val_accuracy: 0.5773
Epoch 45/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9884 - accuracy: 0.5798 - val_loss: 1.0278 - val_accuracy: 0.5702
Epoch 46/50
1642/1642 [==============================] - 5s 3ms/step - loss: 0.9845 - accuracy: 0.5801 - val_loss: 1.0137 - val_accuracy: 0.5841
Epoch 47/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9842 - accuracy: 0.5815 - val_loss: 1.0285 - val_accuracy: 0.5800
Epoch 48/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9832 - accuracy: 0.5818 - val_loss: 1.0165 - val_accuracy: 0.5841
Epoch 49/50
1642/1642 [==============================] - 5s 3ms/step - loss: 0.9788 - accuracy: 0.5826 - val_loss: 1.0211 - val_accuracy: 0.5746
Epoch 50/50
1642/1642 [==============================] - 4s 2ms/step - loss: 0.9768 - accuracy: 0.5829 - val_loss: 1.0128 - val_accuracy: 0.5763
912/912 [==============================] - 2s 2ms/step - loss: 0.9963 - accuracy: 0.5848
Test Loss: 0.9962766766548157, Test Accuracy: 0.5847724676132202
```



```python
from tensorflow.keras.utils import plot_model

# Plot the model architecture
plot_model(gcn_model, to_file='gcn_model.png', show_shapes=True, show_layer_names=True)
```

input_1 | input: | [(None, 2)]
InputLayer | output: | [(None, 2)]
dense | input: | (None, 2)
Dense | output: | (None, 64)
dense_1 | input: | (None, 64)
Dense | output: | (None, 32)
dense_2 | input: | (None, 32)
Dense | output: | (None, 243)



As the test model that we have taken involves a lot of criteria it will be difficult for the GCN thus the accuracy level of KNN is greater than that of GNC.

CONCLUSION

With the help of these ML models and centralities we were able to predict the desired network provider for each user. We were able to discuss about the clustering of the data points. Also, we were able to find out the centralities and their properties for certain networks.

TOOLS AND SOFTWARE USED:

Language-Python,

Platform- Jupyter Notebook, Google Colab

LIBRARIES USED

1. Pandas

2. Matplotlib
3. Plotly
4. Folium
5. Networkx
6.Tensorflow
7.Sklearn
8. json