5G NR CONFIGURATION APP REPORT

Brief Introduction

The 5G NR Setting Application is a web-based tool designed to dynamically estimate optimal 5G network configurations based on real-world geographic data. By extracting information from **OpenStreetMap** which provides road speeds, building counts, and building heights.The application estimates key factors like user mobility and population density. These insights are then used to determine the ideal **subcarrier spacing**, **frequency band**, and **cyclic prefix mode** for a given area. Containerized with Docker, this tool ensures a consistent deployment environment, making it easy to test and evaluate without extensive setup.

The application can also we cloned directly from github:https://github.com/UENDIMUCA/5G_NR_Setting_App/

List of technologies and programming languages use:

- **Python 3.11**
- **Flask**
- **Docker**
- **HTML, CSS, and JavaScript**
- **OpenStreetMap (OSM) & Overpass API:** A free, community-driven map database and its query service used to retrieve geographic data for analysis.
- **osm2geojson:** A Python library that converts raw OSM JSON data into the GeoJSON format, which is widely supported by mapping tools .
- **Requests:** A Python library for making HTTP requests, essential for fetching data from external APIs like Overpass.
- **Math Module:** A built-in Python module used for performing mathematical calculations, such as computing areas for population density estimation.Since we had problem fetching density from OSM we decided to better do a function that calculates the density population.
- **Leaflet:** A JavaScript library for interactive maps used to display geographic data on the web and handle user interactions such as map clicks.

extract_map.py is Data Extraction Script which is used for retrieving data offline or pre-processing.Briefly this script when it runs it has Overpass queries to retrieve road network and building footprint data like their height ,number for a specific are.Then it saves the raw JSON output into a file names roade_network.json and buildings.json.Then it converts these JSON files into GeoJSON format using the `osm2geojson` library, which is a standard format for geographic data and is used for map

app.py is Flask Application:This is the main back-end file that creates a Flask web server with an API endpoint **(`/api/5g_config`)** and a home route that renders your front-end .Here the

script receives firstly longitude and latitude parameters from front end when user clicks on map.It queries the Overpass API that fetch road and building data.When we get the data we decided to not fetch directly density but to calculate it based on a function that we created in this script.We calculate population density by multiplying building count, average floors, and an assumed occupants-per-floor value, then dividing by the area of a 5km radius circle.

Conditions:

- Area classification:
    1. An area is considered a “Big Capital” if it has population density greater than 50,000
    2. An area is Considered “Urban” if it has population density between 10,000 and 50,000
    3. An area is Considered “Rural” if it has population density is less than 10,000

- Frequency Band (Frequency strength depends on three main properties) :
    1. Area classification: Large dense areas frequencies should vary between 3.5 and 24GHZ, however smaller ones vary between 3.5 GHZ and 700 MHZ.
    2. Building count: When we have more buildings in an area the Frequency strength should be reduced to be able to reach wider ranges.
    3. Building floor: If we have an area with high buildings(more than 7 floors) we should consider lower frequencies too, because buildings reduce the signal reachability.

- Cyclic prefix:
    1. Normal: For Capital/Urban areas that usually have a lot of cells.
    2. Extended: For Rural areas that usually don’t have much cells for coverage.

- Subcarrier Spacing:
    1. If Mobility speed is greater than 70 KM/H, we will need 120 KHZ to encounter the doppler effect.
    2. If Mobility speed is between 45 and 70 KM/H, we will need 30 KHZ .
    3. If Mobility speed is 45 KM/H or less, we will need 15 KHZ, because in that speed there is no to minimal doppler effect, so no need for high subcarrier size .

Finally, it returns all this information as JSON for the front end to display.

index.html is for Front-End User Interface .This HTML file provides the interactive interface for the application by leveraging HTML, CSS, and JavaScript along with the Leaflet library to display an interactive map; when a user clicks on the map, the click coordinates are captured by JavaScript and sent to the **`/api/5g_config`** API, which returns configuration details (**including road count, building count, average speed, population density, and 5G network parameters),** and the interface then displays these details in a styled results box while drawing a circle with a 5km radius on the map to visually represent the queried area.

requirements.txt:This file lists all the Python dependencies needed for project.

Docker file.We used Docker to containerize the entire application.

- It starts from an official Python 3.11 image.
- It sets up the working directory, installs dependencies from `requirements.txt`, and copies the project files into the image.
- It exposes the necessary port (5000) and specifies the command to run both the data extraction script first and thenFlask application.

We preferred to use docker so the app runs constantly on any machine with Docker, eliminating environment configuration issues. Alternatively, you can run the application without Docker by manually installing Python 3.11and the required dependencies, then executing the scripts directly from the command line.

Steps how to run the project by using Docker

1.Download and install Docker Desktop from docker.com

2.You can clone from github or directly use the zip file sent

3.Build Docker image:**docker build -t 5g-nr-app** .

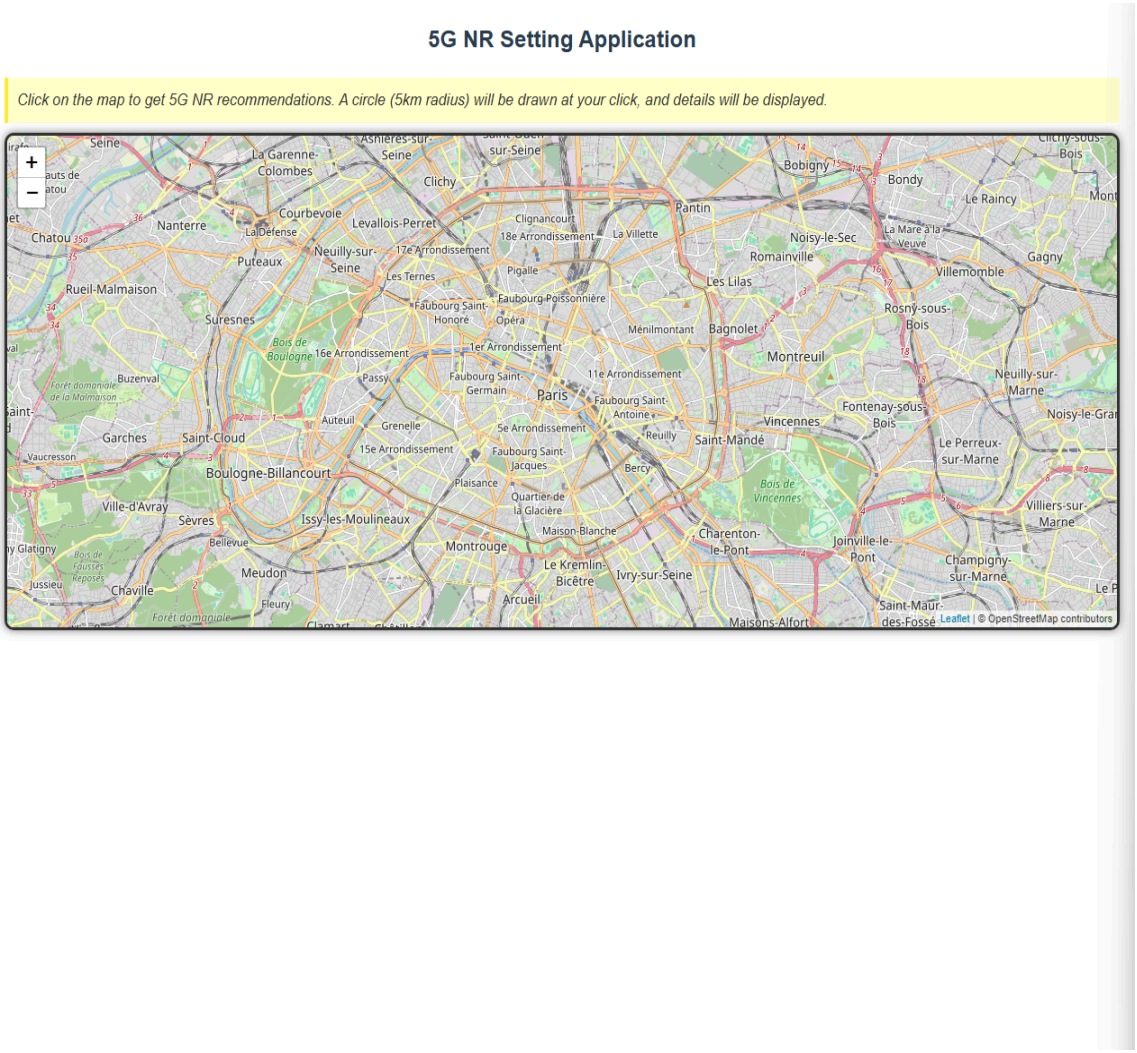
4.Run the Docker Container:**docker run -p 5000:5000 5g-nr-app**

5.Access the Application:**http://localhost:5000 or http://127.0.0.1:5000/**

PS.When you click on map it can take some seconds to show the circle an data as due to DOCKER .And blue circle has radius 5 km

Results and Screenshots:

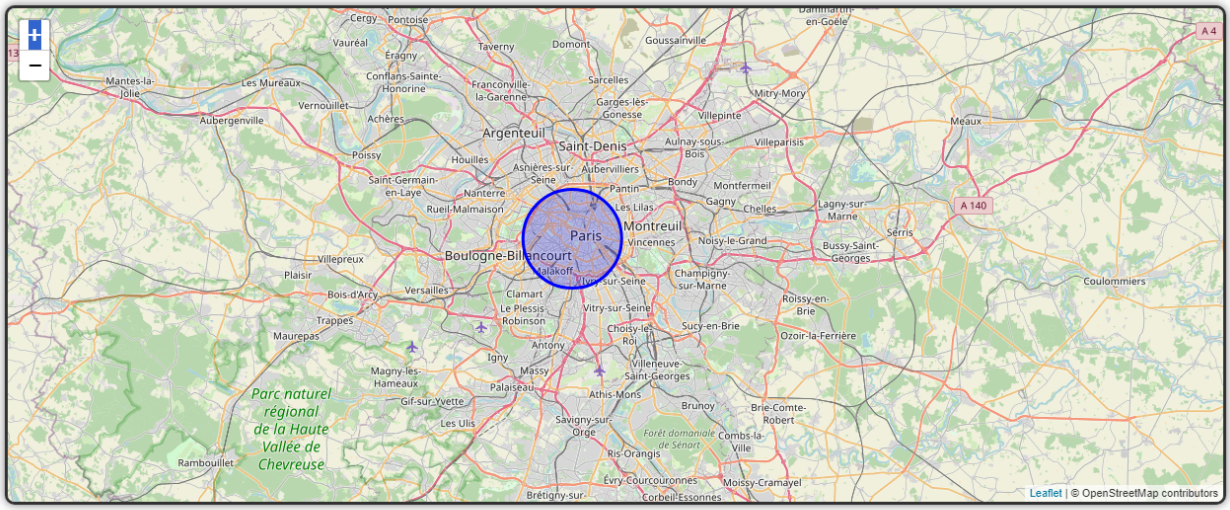
- When you first run the application it should take time then it will show:



- For example here we chose a Capital:

5G NR Setting Application

Click on the map to get 5G NR recommendations. A circle (5km radius) will be drawn at your click, and details will be displayed.



5G NR Configuration

Location: (48.85070650057199, 2.3294448852539067)

Road Count: 0

Building Count: 94951

Average Building Floors: 5.290682088008885

Average Road Speed: 50 km/h

Population Density: 191885.6874195348

Area Type: Big Capital

Subcarrier Spacing: 30 kHz

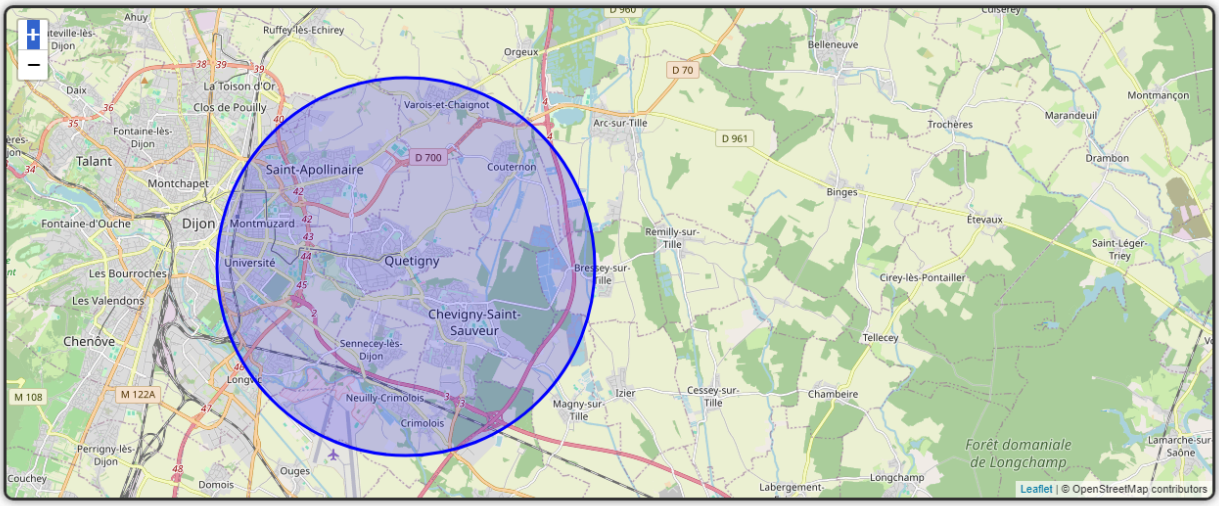
Frequency: 3.5 GHz

Cyclic Prefix Mode: Normal

- For example here we chose an Urban:

5G NR Setting Application

Click on the map to get 5G NR recommendations. A circle (5km radius) will be drawn at your click, and details will be displayed.



5G NR Configuration

Location: (47.310867420342184, 5.114135742187501)

Road Count: 8133

Building Count: 27779

Average Building Floors: 2.6043956043956045

Average Road Speed: 47.22569641841956 km/h

Population Density: 27634.711487564662

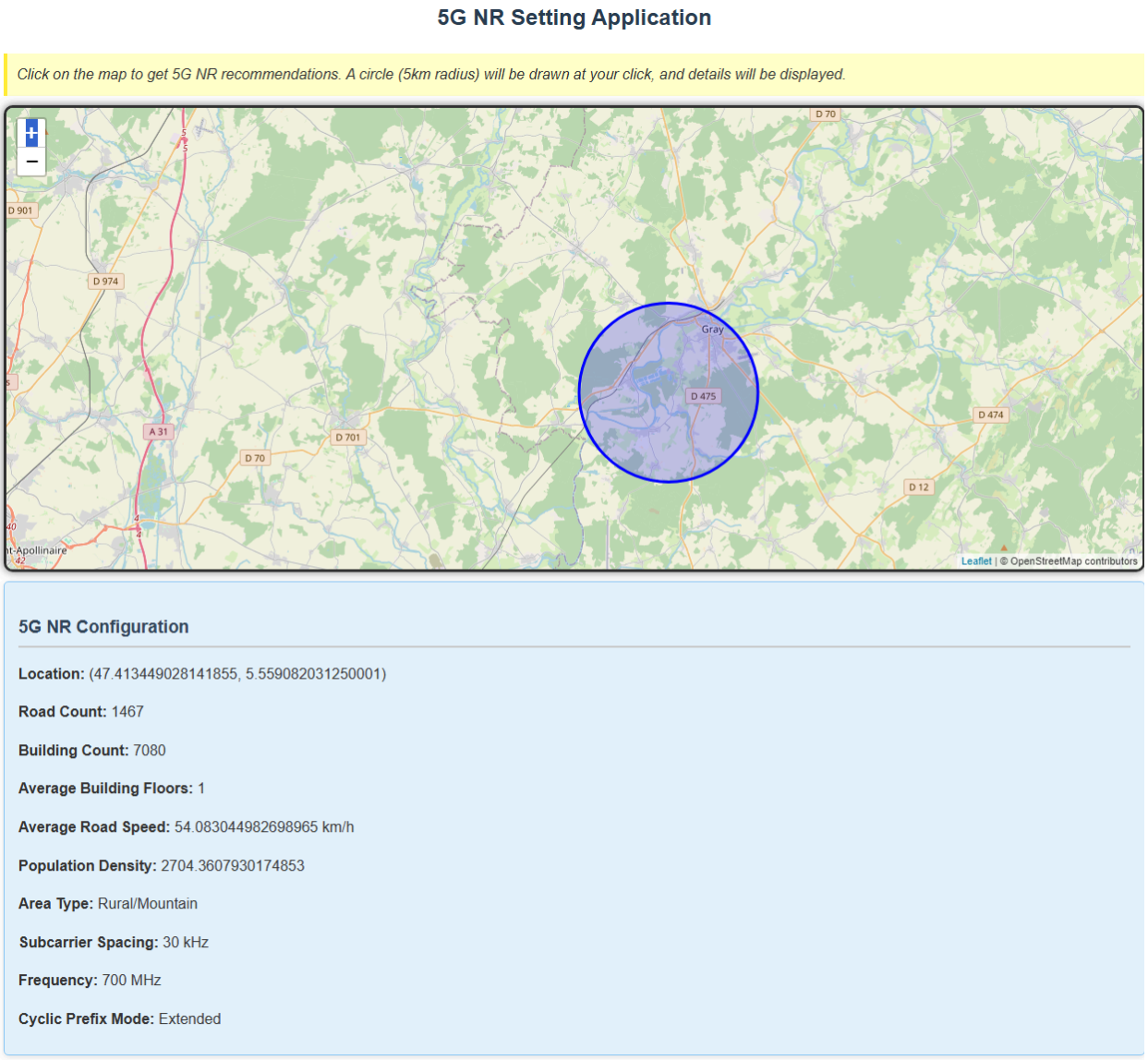
Area Type: Urban

Subcarrier Spacing: 15 kHz

Frequency: 700 MHz

Cyclic Prefix Mode: Normal

- For example here we chose a Rural area:



To be Noted:

1. The circle radius is 5 km which represents the area being chosen by the user to retrieve data for.
2. The data shown below the map represent the data retrieved from the OverPass api, also with some data that we estimated and deduced using the retrieved data itself.
3. If there is some data showing 0, it means that the api doesn't provide it for this certain chosen area.