Boston University
Electrical and Computer Engineering

EC 463 Senior Design Project

Second Prototype Testing Plan

3/4/2021

Aerial 5G Network Modeling

Team 17

Team Members
Taehyon Paik ttpaik@bu.edu
Minseok Sakong msakong@bu.edu
Talbot Taylor talbotct@bu.edu
Stanley Zhang zhangs5@bu.edu
Yuan Chi Chung ycc88@bu.edu

1.0 Prototype Required Materials

2.1 Hardware:

- Android phone equipped with 5G AT&T network
- DJI Flame Wheel Drone
    - F450 Frame Kit
    - 2312E 800KV Motors
    - 30A ESC Brushless Speed Controllers
    - 10 inch FPV 1045 Propeller
    - 2x 3000mAh LiPo battery
    - Flysky FS-i6X Transmitter

Figure 1: DJI Flame Wheel Drone

2.2 Software:

- Data collection
    - Our speed testing application
    - Ookla SDK for our app (Kotlin)

- Data display and management
    - Python3
        - Pandas
        - Plotly
        - Numpy
        - Scipy
        - Sklearn
        - Matplotlib

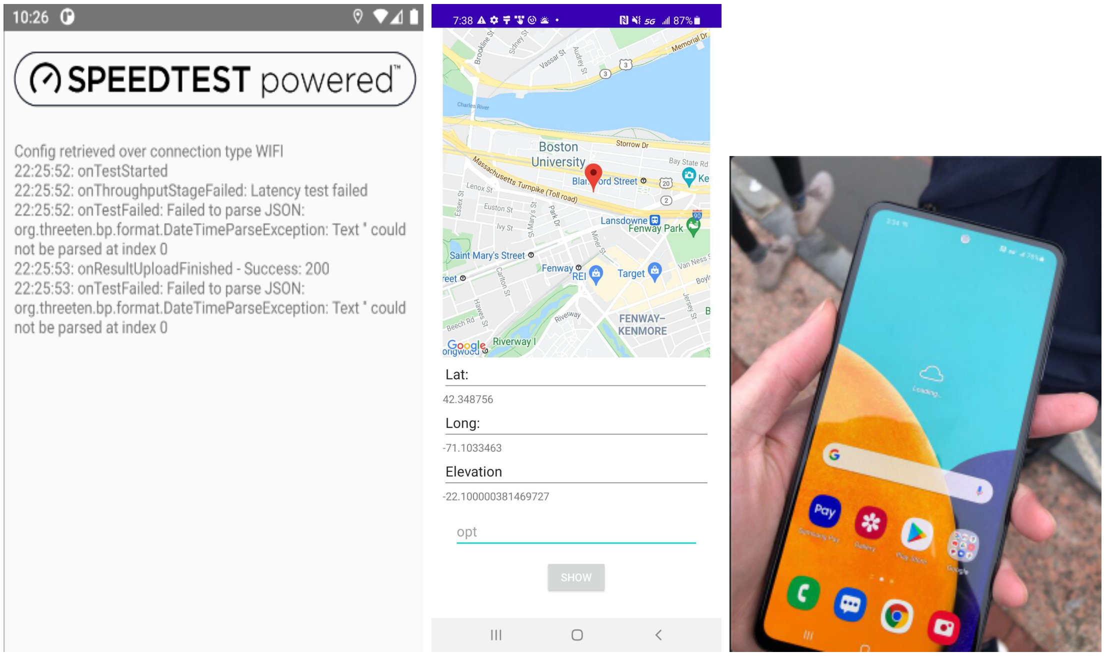


Figure 2: Our application based on the Ookla SDK

<u>3.0 First Prototype Testing Outline</u>

We will take you through the three sections of our demonstration, data collection, visualization and Machine Learning model creation.

3.1 *Data collection* is first and is where we will run a speedtest on our application to collect data to be used in the visualization and ML steps. The Android phone is equipped and connected to the 5G AT&T network. After the speed test is complete, the results will be saved as a .json file for use.

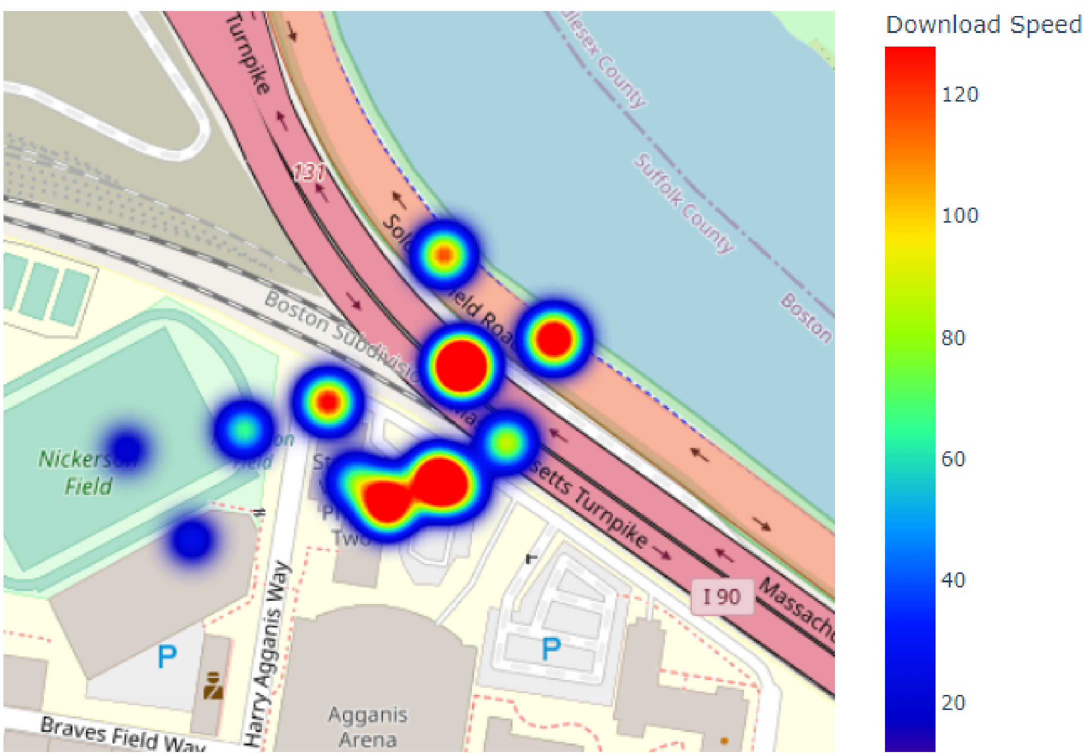
3.2 *Data visualization* is where our python script will take this data and create a visual representation of the including testing locations and speed. You will be able to see the newly collected data as well as previous tests displayed. These will be shown with various colors representing the speed of tests at each location. We also have heatmaps to display the speeds over a wider, more gradual area. Different maps show upload speeds, download speeds and latency.

3.3 *Machine Learning Model* here we will take the data collected previously and construct a model based on clustered linear regression to be predictive for speeds at different altitudes and locations. We will also apply normalization to the results to improve our performance. We will display the predicted results as plots and print the error rates as well as the accuracy of the model.

```python
import pandas
import plotly.express as px
data = pandas.read_csv("SpeedTestExport_20211116 (1).csv")

fig = px.scatter_mapbox(data, lat = "Lat", lon = "Lon", hover_name = "Date", hover_data = ["Download", "DownloadBytes", "Upload", "UploadBytes", "Latency", "ServerName", "ConnType", "ConnDetail
fig.update_layout(mapbox_style = "open-street-map")
fig.update_layout(margin = {"r":0,"t":0,"l":0,"b":0})
fig.show()
```

Figure 3. Code for mapping



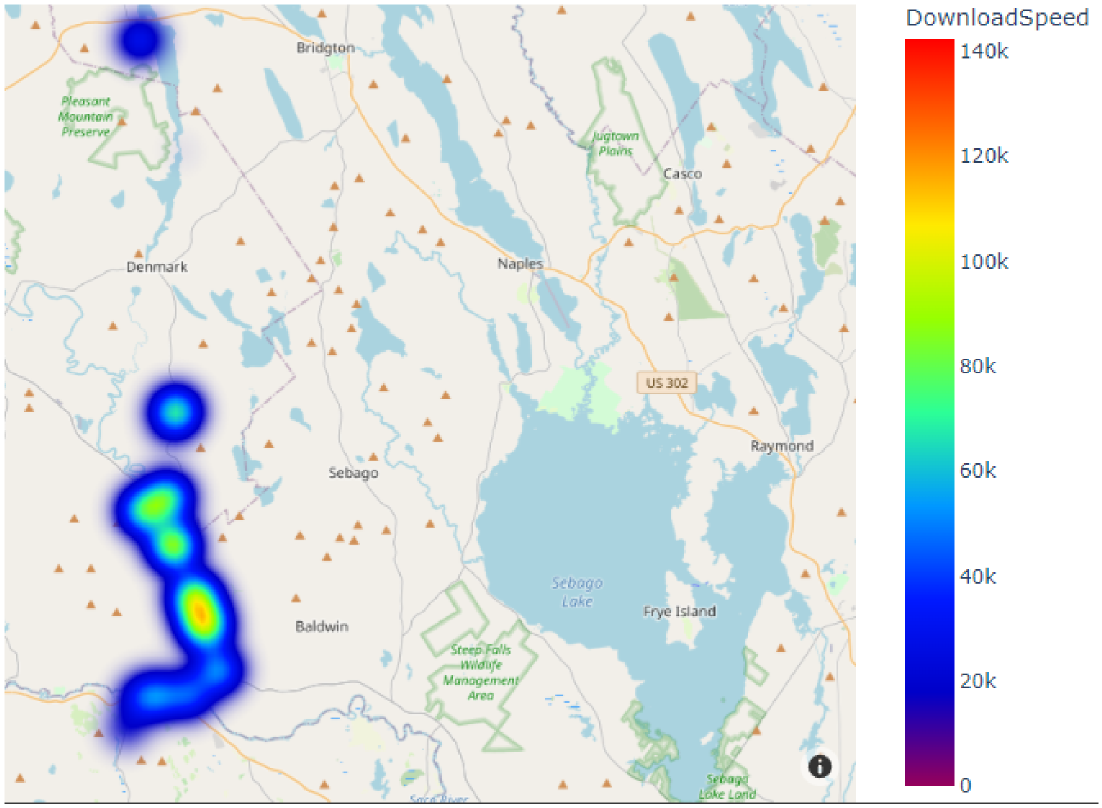
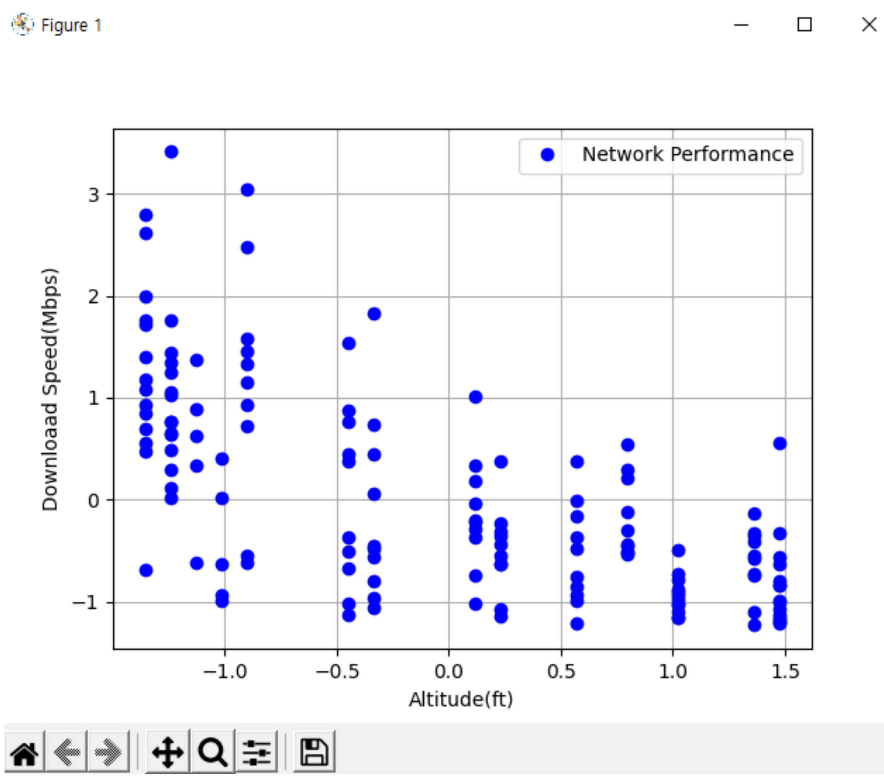


Figure 4. Example of heat mapping and data points

Pre-CLR Data



CLR applied Data

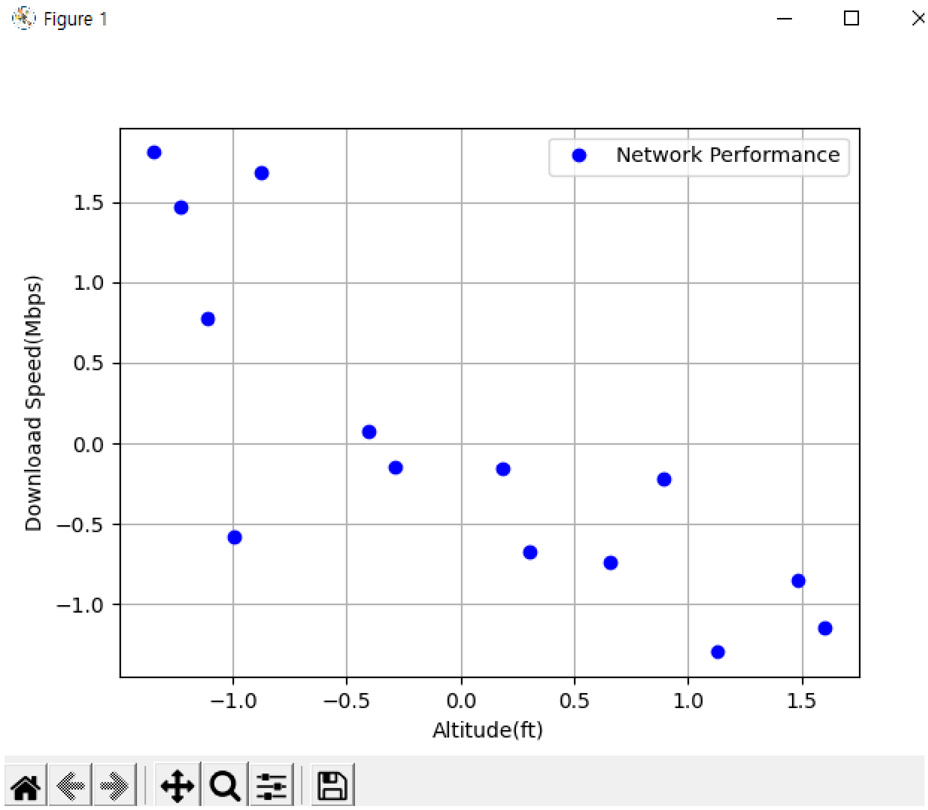


Figure 5. Clustered Linear Regression data points

```
Coefficients: [-0.83909022]
Prediction for test set: [-0.92936574 -1.22670394  0.35909977]
Mean Absolute Error: 0.34425058292909333
Mean Square Error: 0.12036275347902585
Root Mean Square Error: 0.34693335596195685
Accuracy:  63.07686325388332
```

Figure 6. Normalized Clustered Linear Regression results

4.0 Pre-testing Procedure

Drone:

I. Attach phone with 5G connection to the drone
II. Open open our testing application
III. Turn on drone
IV. Make sure drone is paired with controller

5.0 Testing Procedure

5.1 Data collection:

I. Fly drone to desired altitude and location
II. Run our application’s speed test to record data

III. Download data as .json file

5.2 Data visualization:

I. Input .json file for use in python script
II. Display mapped data

5.3 Model Usage

1. Train model using clustered linear regression
2. Apply normalization
3. Display data points and results

6.0 Measurable Criteria

6.1 Criteria for successful running and displaying:

Data is collected from application on phone

- Collection is repeatable and consistent

Data is exported to a .json file

- File can be sent to team for use
- Data is properly formatted and labeled for python script

.json file is opened and displayed for visual clarity of the user

- Map is scalable as amount of data increases
- There are multiple maps corresponding to different aspects of connection speed

Machine Learning results properly display

- Error rates and accuracy are correct based on results
- Plots contain correct data points based on model

Machine Learning results reach current desired accuracy

- Error rates are below 0.5
- Accuracy is above 60