Boston University
Electrical and Computer Engineering

EC 463 Senior Design Project

Final Prototype Testing Plan

4/1/2021

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 Required Materials

1.1 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 1.1: DJI Flame Wheel Drone

1.2 Software:

- Data collection
  - Created speed testing application (Kotlin)
  - Ookla SDK for created application
  - Google Firebase Firestore

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

- Web Application
  - ASP.NET/C#
  - HTML/CSS
  - Plotly Chart Studio
  - Flask

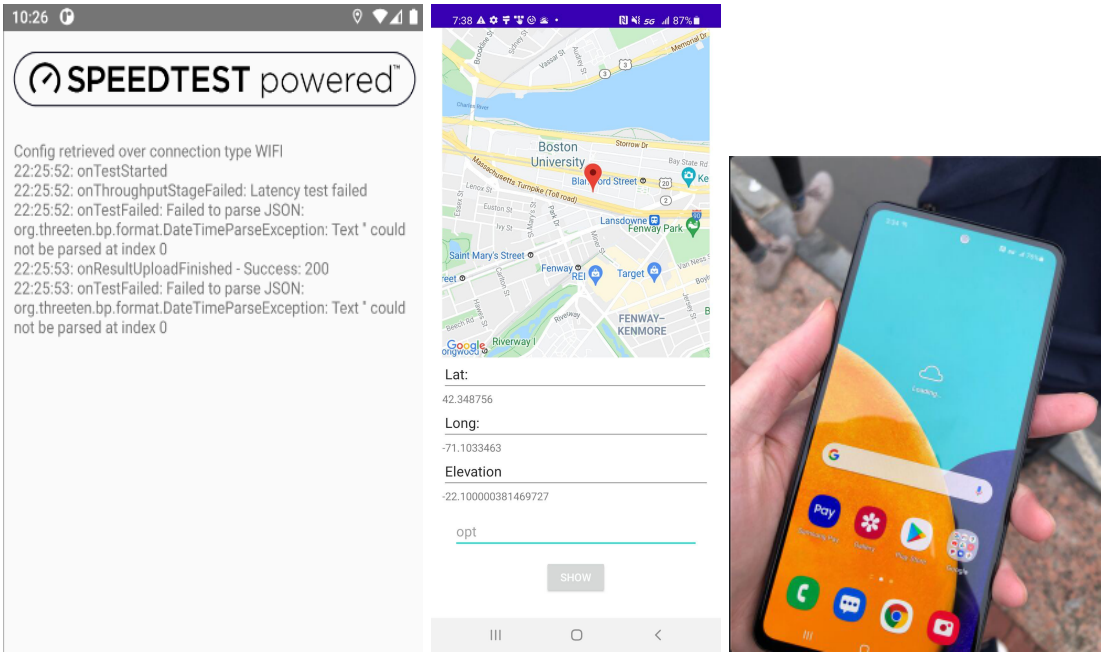


Figure 1.2: Our application based on the Ookla SDK

2.0 Final Testing Outline

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

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.

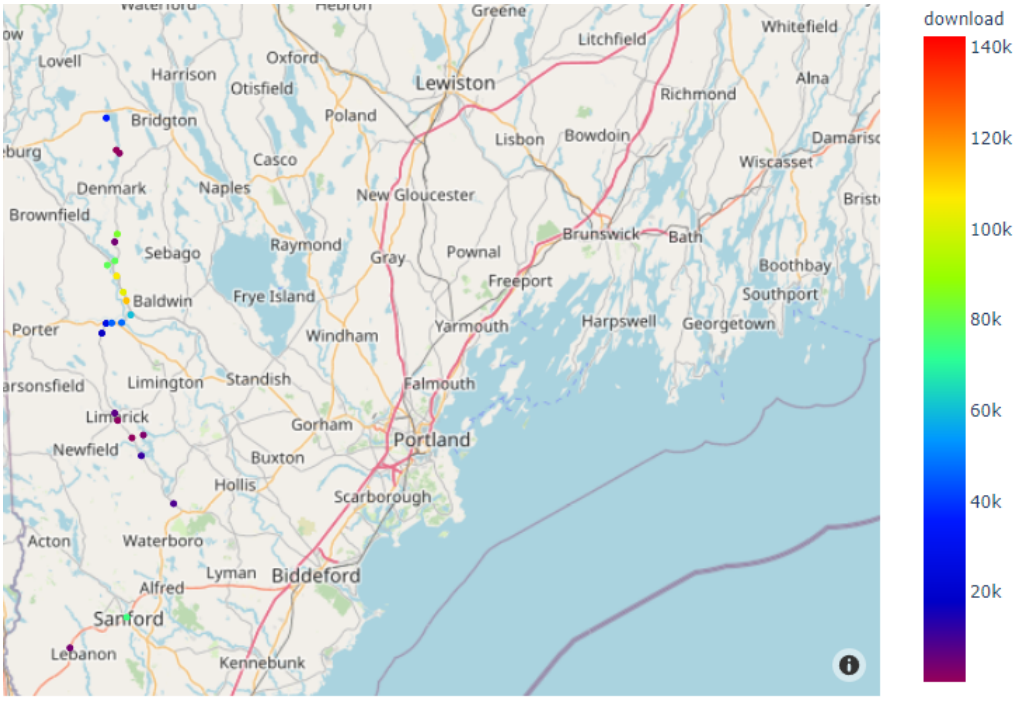
3.3 *Machine Learning Model* here we will take the data collected previously and construct a model based on clustered KNeighbors regression to be predictive for upload speeds and download speeds based upon a given input of latitude, longitude and altitude. The model was also evaluated by displaying the error rates and R squared score.

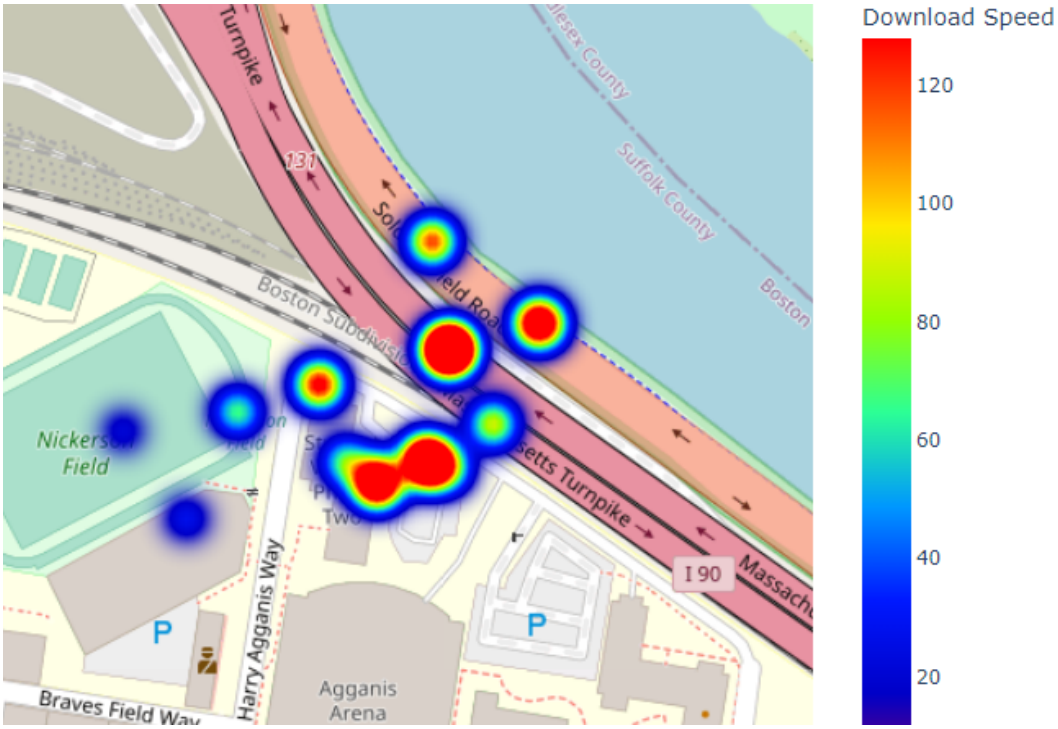
3.4 *Web Application* is where we take all of our material from data, visualization, and modeling, and show it in one user framework. Our web application is currently in ASP.NET and C#, connected to the firebase database, and uses the ML model functionality.

```
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





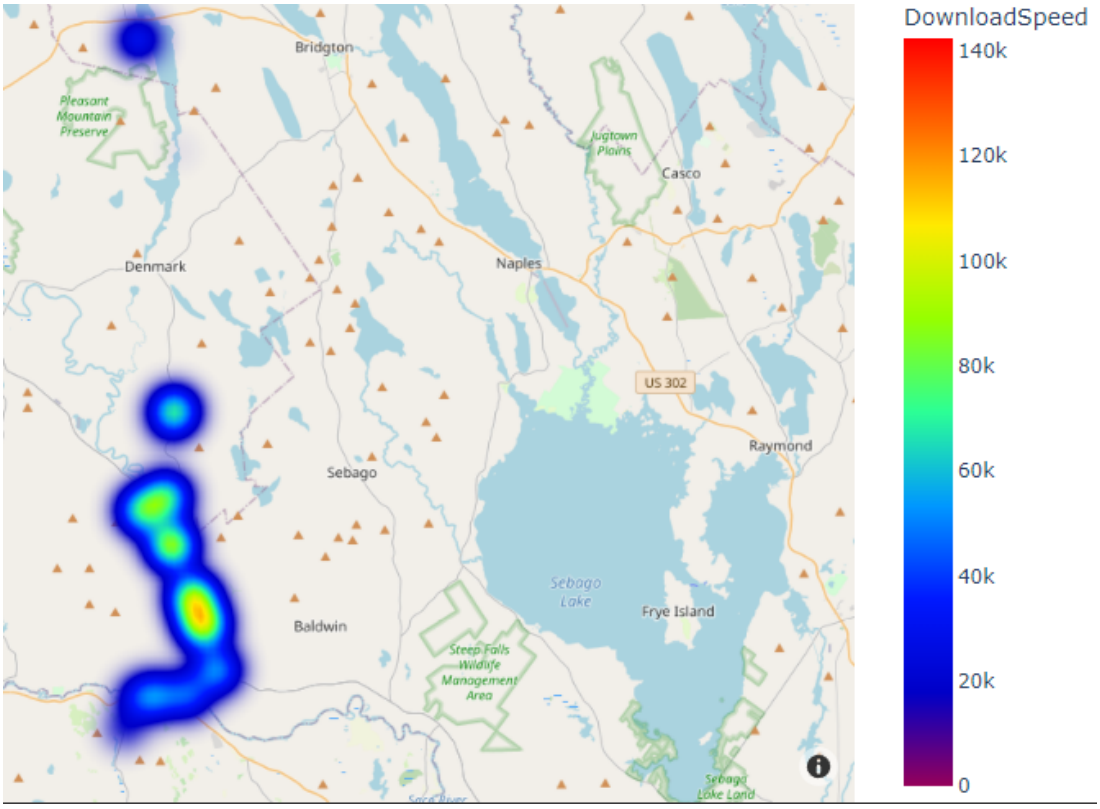
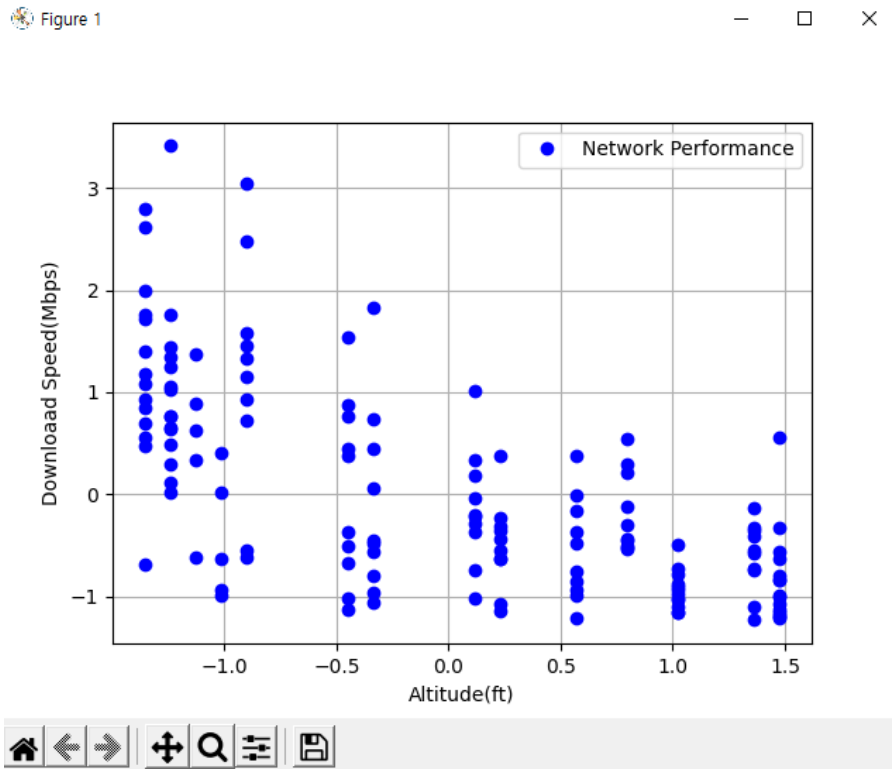


Figure 4. Example of heat mapping and data points

Pre-CLR Data



CLR applied Data

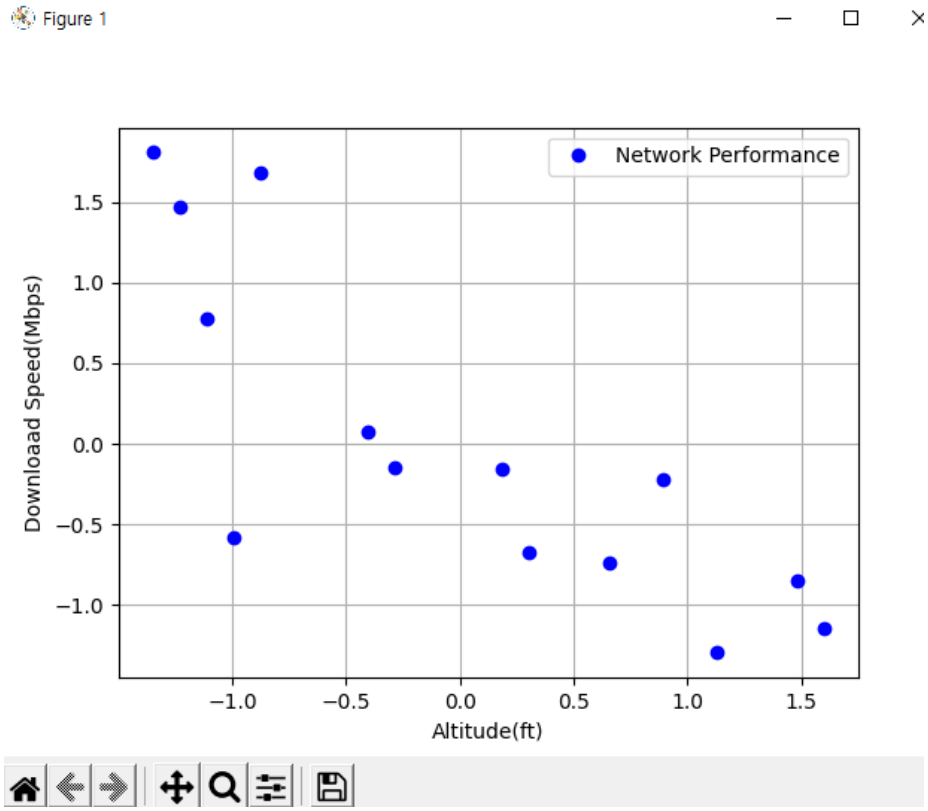


Figure 5. Clustered data points

```
Prediction for test set: [[21.0251396  14.67906233]
 [21.0251396  14.67906233]
 [38.112365   23.5667678 ]]
Mean Absolute Error: 3.2735914666666637
Mean Square Error: 24.892781082381553
Root Mean Square Error: 4.989266587624032
r2Score:  76.9683002727003
```

Figure 6. Improved results using KNeighbors Regression

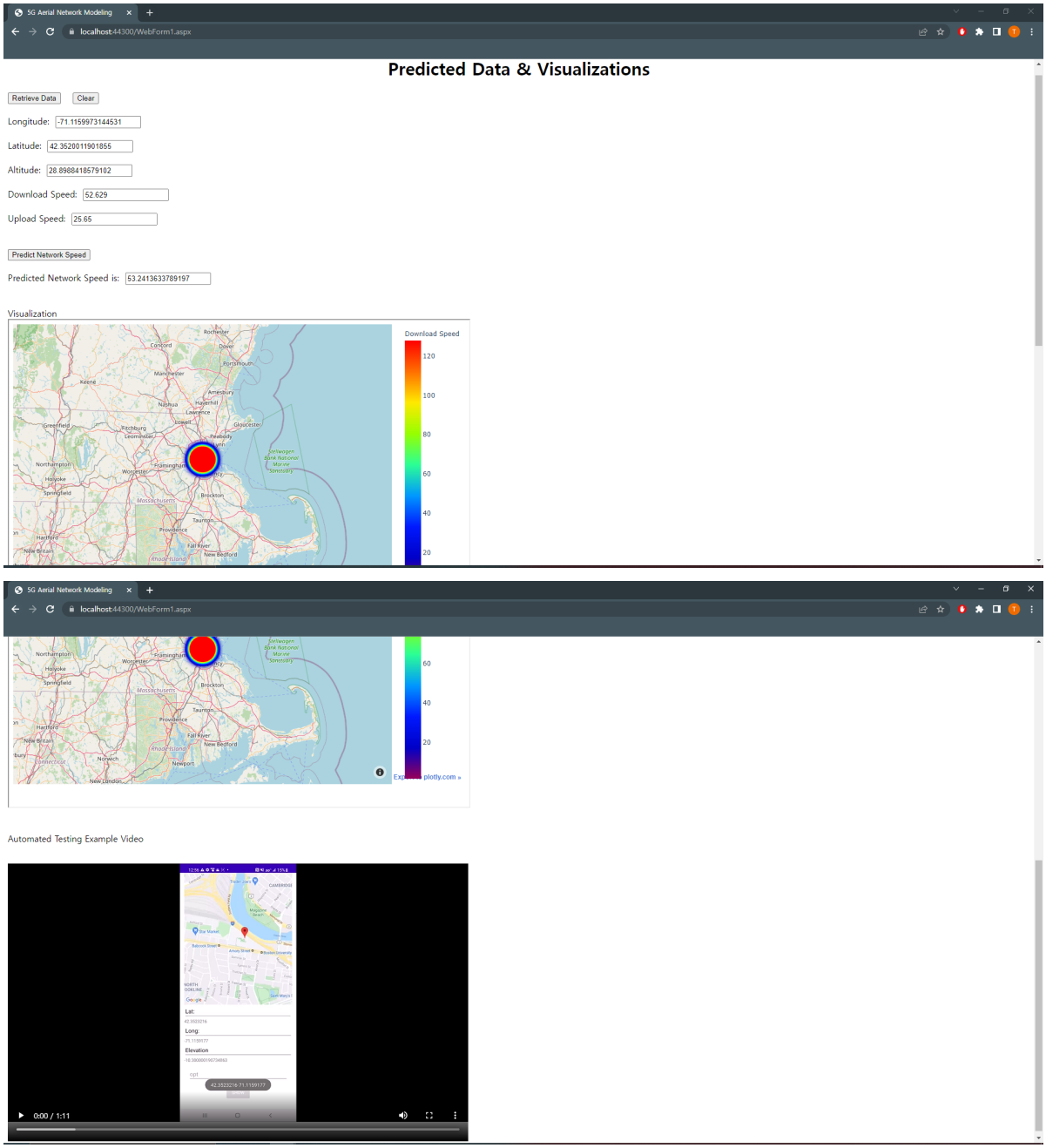


Figure 7. Web Application

3.0 Pre-Testing Procedure

Drone:

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

4.0 Testing Procedure

5.1 Data Collection:

1. Fly drone to desired altitude and location
2. Run our application's speed test to record data

5.2 Data Visualization:

1. Query collect data from database
2. Input data file for use in python script
3. Display mapped data

5.3 Machine Learning Model Usage

1. Train model using clustered KNeighbors regression
2. Make predictions
3. Display evaluation metrics

5.4 Web Application

1. Turn on the web application
2. Press the "Retrieve Data" button to fetch recently collected data
3. Press the "Predict Network Speed" button to predict speeds with fetched data
4. View visualizations, automated testing videos

5.0 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 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 5 (MAE, RMSE) and below 30 (MSE)
- Accuracy is above 70

Web application displays relevant information and tools for project

- Web application prediction results accurate compared to network speed
- Allows for pulling of data from database
- Displays data visualization tools and maps