Boston University
Electrical and Computer Engineering

EC 464 Senior Design Project

Second Prototype Report

3/18/2021

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 Setup Summary

1.1 Required Materials

Hardware:

- Android phone with 5G cellular connection on 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

Software:

- Data collection
    - Our Android mobile application
        - Kotlin
        - Java
        - Google Maps API
        - Ookla Speedtest SDK
    - Data Storage and Management
        - Speedtest-cli
        - Firebase_admin

- Data Visualization
    - Python3
        - Pandas
        - Plotly
        - Numpy

- Machine Learning Tools
    - Python3
        - Numpy
        - Scipy
        - Matplotlib
        - Sklearn

1.2 Data Collection

Over the past few weeks before the prototype demonstration, our drone had been having issues with broken parts, but we were able to resolve that before presenting. Therefore, the first action we took during the demo was to show our drone was functioning and able to operate as needed.

We then displayed the improved mobile application we had built upon from last semester. It is now able to automatically and continuously collect data while in flight. This is an improvement from our last prototype because it uses the Ookla SDK and runs without requiring human interaction. The results from the speed tests were automatically uploaded to a Firebase database and then able to be extracted into a .json file, as we demonstrated. This data is then pipelined into our data visualization and machine learning sections.

Automatically recording the altitude during our tests required additional work. The client requested the altitude in relation to the height off of the ground. Unfortunately, our Google Maps API only provides the WGS84 ellipsoid altitude. In order to determine the Geoid height, the application must request two different altitudes simultaneously: 1. ellipsoid height (h) and 2. geoid height (N). The orthometric height is then determined using the equation: H = h - N to find mean sea level (orthometric height), the altitude that the client requested.

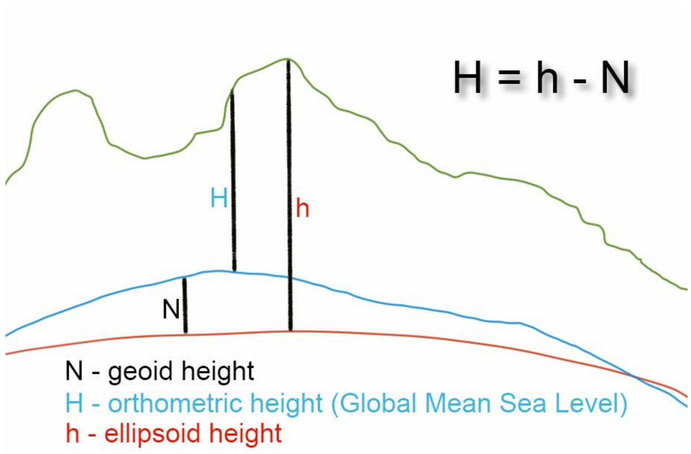


Figure 1. Example chart for finding altitude

1.3 Data Visualization

We put this collected data into our Python script that displays the information in a geographical heatmap to indicate how fast the network performed at the locations of interest. We were able to display both the newly collected data as well as previous tests conducted. The data was collected from various places on campus, including the Student Village dorms and the Photonics Building.

We also presented other maps to show upload speeds, download speeds and latency to properly evaluate network performance across available factors. This format aids in visualizing the range of data. Our current methods create various 2D maps based on clusters of similar altitudes. We are currently creating a similar method to display our predictive model's results, based on our 5G data. On ECE day we will provide an application demonstration that allows users to search for predicted results at input locations.

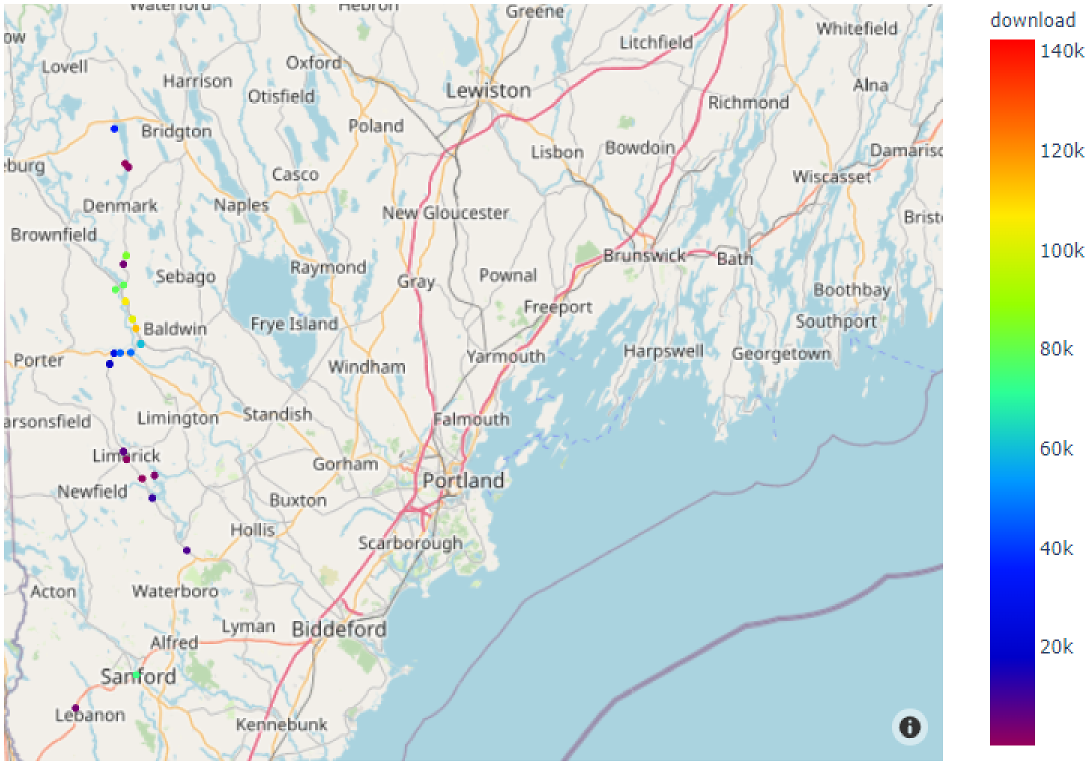


Figure 2. Mapped discrete data points

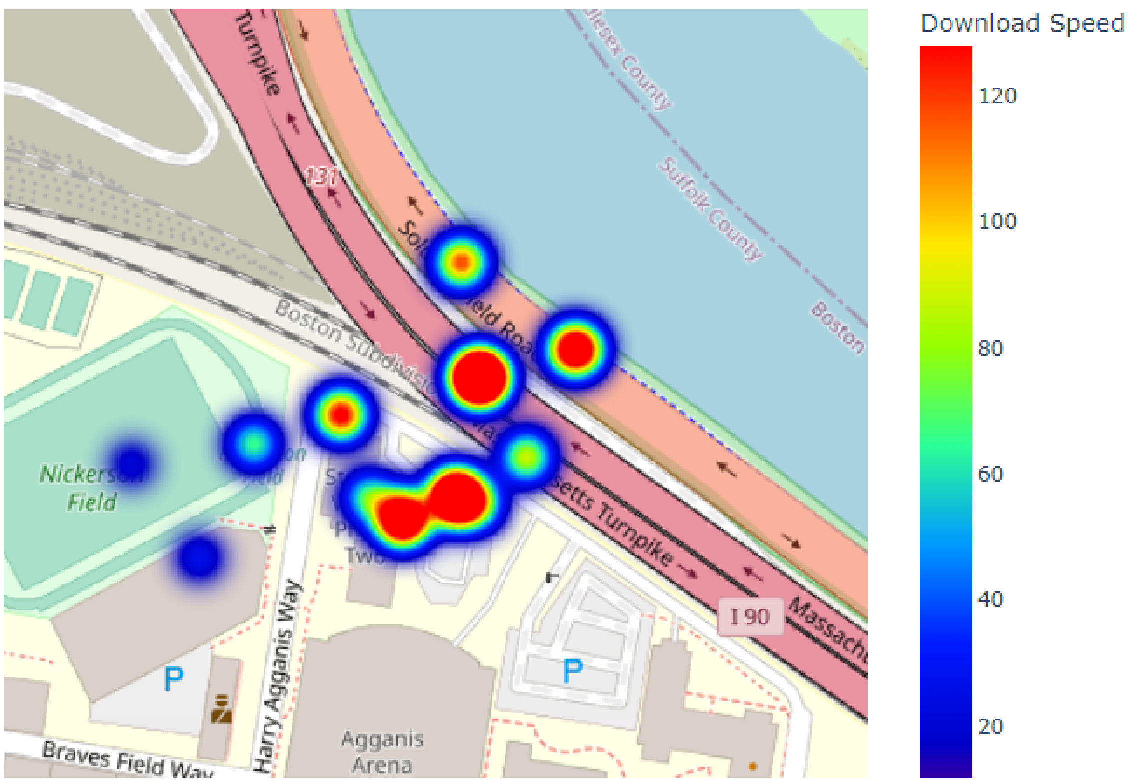


Figure 3. Heat mapping of collected data on BU campus

1.4 Machine Learning Modeling

We displayed our basic linear regression model for the data that we have collected so far, using libraries such as pandas, numpy, matplotlib, and sklearn.

The processes for this model include:

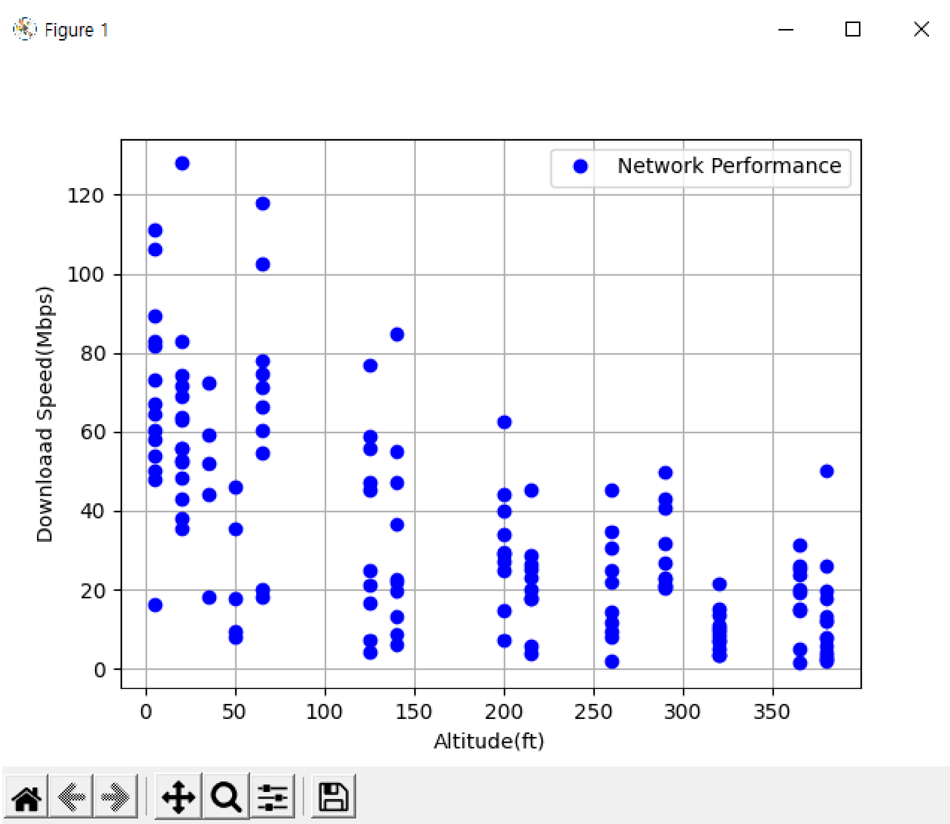
- Setting/splitting datasets
- Clustering datasets with respect to similar altitudes
- Fitting the multiple variables into linear regression model
- Finding the intercept/coefficient
- Visual comparison between predicted and actual values
- Model evaluation using different error rate calculations.

The trend of the predicted values versus actual values have a linear relationship so far, and calculated error rates using:

- R-squared
- Mean Squared Error
- Root Mean Squared Error

These rates are below 10%, which gives us an accurate linear regression model for predicting values with our current dataset.

Pre-CLR Data



CLR applied Data

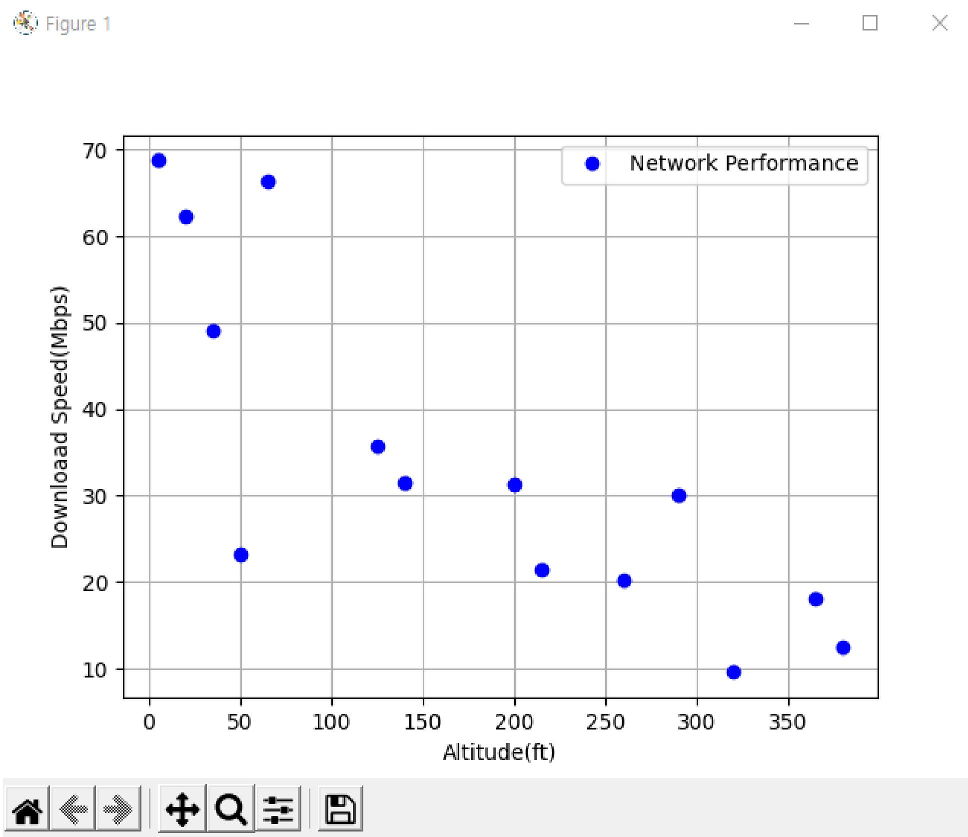


Figure 4. Clustered Linear Regression data points

```
Intercept:  56.88062314393973
Coefficients: [-0.12593099]
Prediction for test set: [16.58270621 10.91581165 41.13924934]
Mean Absolute Error: 6.560986059454322
Mean Square Error: 43.7200827851124
Root Mean Square Error: 6.6121163620366215
Accuracy:  63.076863253883296
```

Figure 5. Normalized Clustered Linear Regression results

2.0 Measurements

2.1 Collected Data

Our data pool has significantly increased in size from the last prototype demonstration. During this demo, we collected additional data related to the network speeds within the Photonics Building and added this to our growing test dataset. This data appears consistent with previous tests in the same geographic area. Additionally, the data mapped well into our data visualization system, which proved useful during the demonstration of the prototype. We also used this data in our machine learning model, which has significantly improved from our previous iterations. The few data points we added during the demonstration did not have a large impact on our predictive model's results.

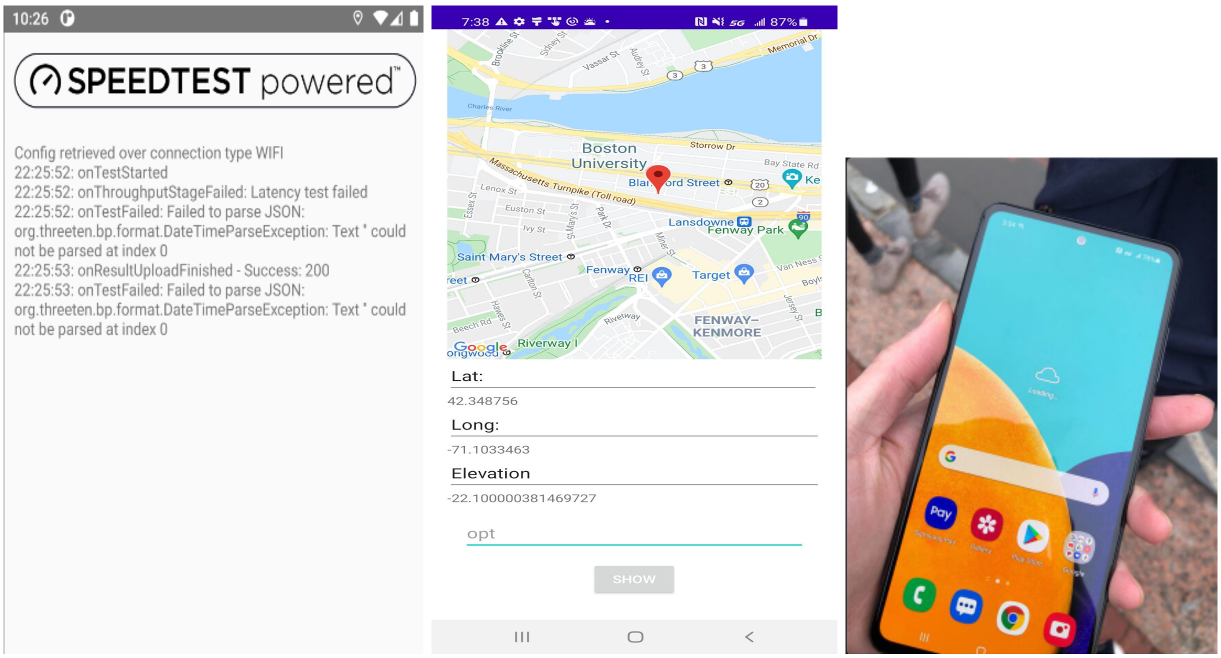


Figure 6: Our data collection application based on the Ookla SDK

3.0 Conclusions

3.1 Results

Our prototype performed as expected in the data collection, data visualization and predictive modeling portions of our testing. Each team member took part in the presentation and shared the aspects they were able to contribute to since the last demonstration. While we have made large strides in the accuracy of the results of our model, there are still refinements to be made. Future improvements will allow us to expand our visualization tools which will better represent our data. It is promising that our clustered linear regression model has improved and enables us to focus on refining other aspects of the project.

3.2 Future Plans

We are pleased with the suggestions provided from this demonstration and we plan on making the necessary improvements before the end of our project. Next steps include using our drone to collect more information to expand our dataset, fine tuning the machine model and finishing our application for user interaction with predicted data. The data collection will be simple and only require some time devoted to it. The model improvements may be reached through further experimentation with factors such as normalization, adjusting our altitude clusters, factoring additional dimensions like latitude and longitude or testing some different models on our data. The application for our final demonstration will involve migrating our current visualization tools to work with our prediction model and wrapping this in an application for use.