Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

Aerial 5G Network Modeling

Submitted to
Tim Geraghty

by
Team #17
Aerial 5G Network Modeling Blue

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

Submitted 4/18/2022

Aerial 5G Network Modeling

Table of Contents

Executive Summary

Mobile network operators are in the process of building out their 5G networks to support an ever-growing range of applications. The impact of these emerging capabilities is wide ranging and will have major impacts on many aspects of AT&T's business and infrastructure. Using our Aerial 5G android application based off of the Ookla Speedtest SDK, we have collected 5G connection data and built a machine learning model that is predictive of 5G speeds in different locations and conditions. For clarity, the collected data and predicted results are visualized using geolocation maps. This allows users of the application to further examine 5G networks and predict connection speeds in urban environments. The current machine learning model accounts for geographic location, altitude, and connection source location. It also predicts latency and upload/download speeds for these conditions.

1 Introduction

5G, a fifth generation mobile network and successor to 4G, is currently in the process of becoming the new global wireless standard. 5G technology is designed to be able to virtually connect more devices together and deliver higher data speeds, lower latencies, better reliability and greater network capacity. In building these 5G networks, mobile network operating companies like AT&T require the utilization of drones to complete various tasks such as delivering packages, analyzing structures, and surveying natural disasters. One area of which there is very little understanding is the performance of telecommunications networks in airspace. Our project provides insight on this by collecting data on 5G network performance across different heights and environmental conditions in order to perform data visualization and predictions. AT&T and other 5G companies can use these results to revamp their networks and evaluate the potential performance of their drone usage, which relies on 5G for communication.

This project uses an Android mobile application that automatically collects network speed tests from the operating device. The user can run the application and send more data to the current database. The project is integrated separately with a machine learning component to take the user collected data and create models for predictions. The user is able to directly interact with the collected data and predictions through a web application that displays a geographical map that shows the user the upload and download speeds at any given location.

2 System Overview and Installation

2.1 Overview Block Diagram

Our 5G Network modeling project consists of 4 main parts: a drone (QwinOut Quadcopter), an app for automated network data collection, a machine learning model, and a platform for data visualization. The drone is built from a QwinOut Quadcopter assembly kit and is capable of flying up to 250ft. The data collection application uses Ookla SDK APIs and conducts automated network data collection, including latency, download/upload speed, latitude, longitude, and altitude. The machine learning model is programmed using Python with the Sci-kit Learn package. Lastly, the data visualization platform is built in Python, using the Plotly package to host our maps in our web application.

Figure 2.1 shows the data flow model of the project. An android phone with the data collection app is mounted on the drone, and the drone flies up to 250ft. While flying, the data collection app is executed and collects network data from different altitudes. Collected data is directly uploaded to the Google Firebase real-time DB. Then, the collected data are extracted and parsed to a CSV file for the machine learning model and the visualization. The machine learning model is trained with the collected data and the visualization model shows data points with a heatmap.

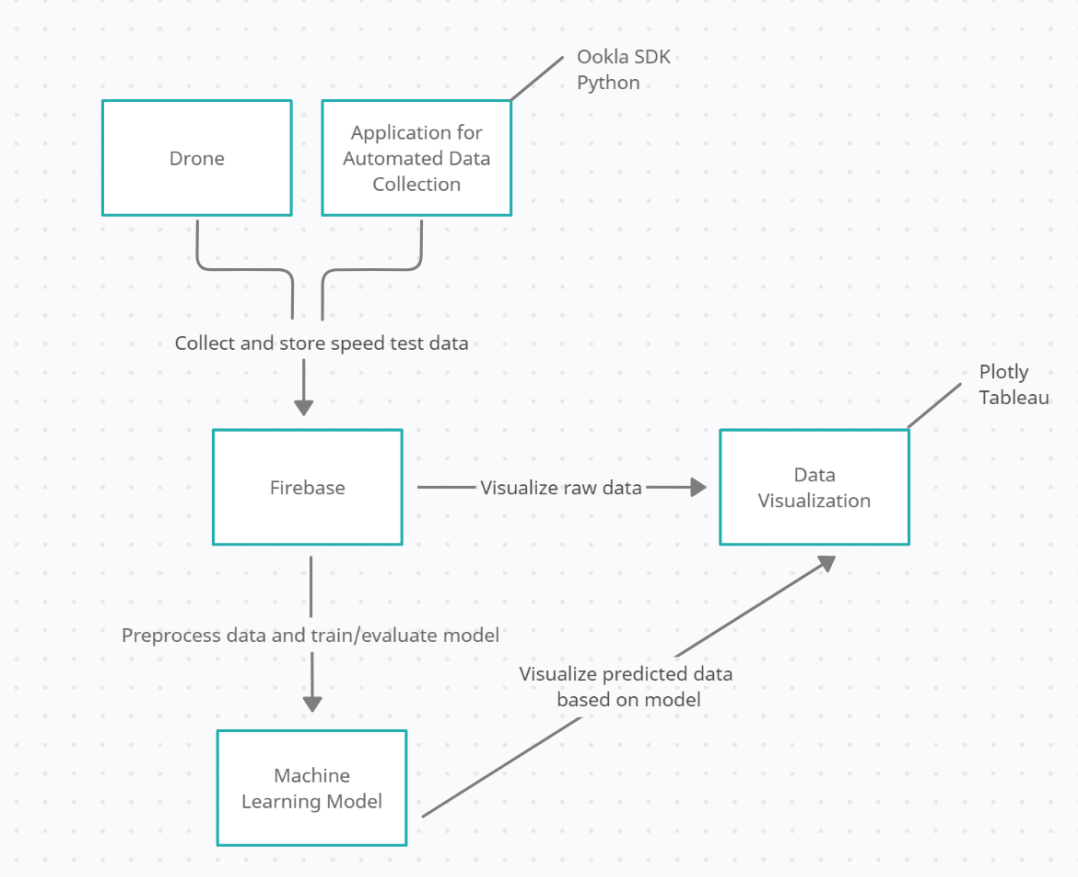

Figure 2.1 System Block Diagram

2.2 User Interface.

Figure 2.2.1 demonstrates the user interface of the data collection application. The data collection application shows the progress of data collecting and database uploading by console prints. In addition, the application shows the location information from Google Maps API to users.

Figure 2.2.2 demonstrates the data visualization platform. The platform visualizes each data point with a heatmap.

Figure 2.2.3 demonstrates the web application platform. This platform has features for retrieving test data, predicting network speed, and visualization as a whole.

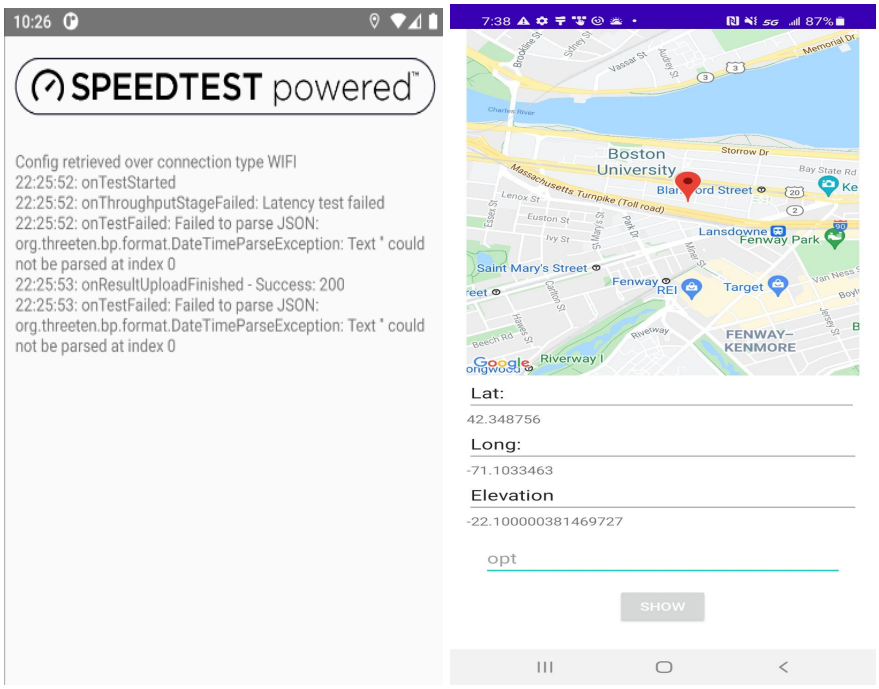

Figure 2.2.1 Data collection application

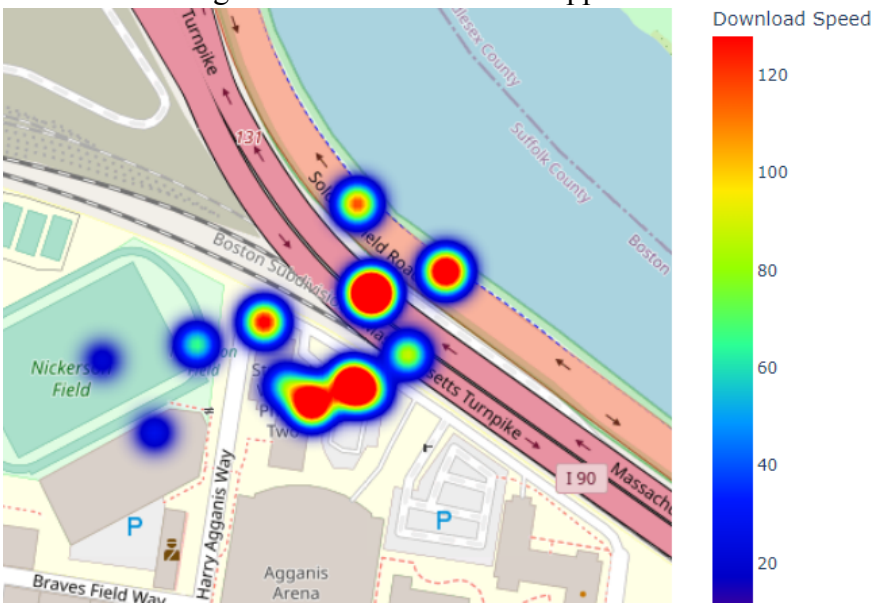

Figure 2.2.2 Data visualization

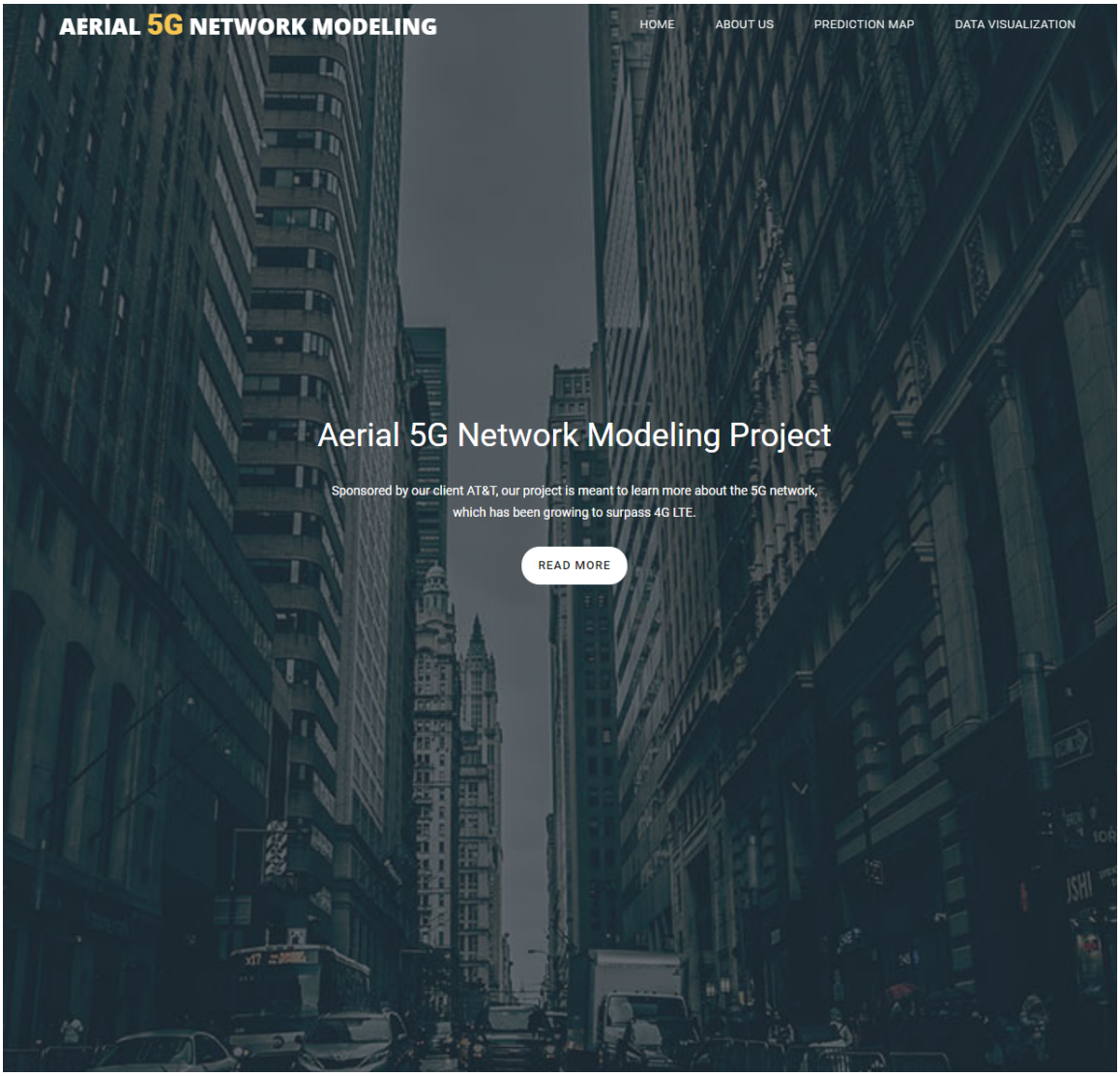

Figure 2.2.3 Web application

2.3 Physical Description.

Figures 2.3.1 and 2.3.2 show the main physical components of the QwinOut Quadcopter drone. Figure 2.3.1 demonstrates the overall design of the drone. The drone contains 2 body frames and 4 wings with a motor attached to the end of each wing. The android phone is mounted on one of the body frames. Figure 2.3.2 demonstrates a detailed sketch of a wing.

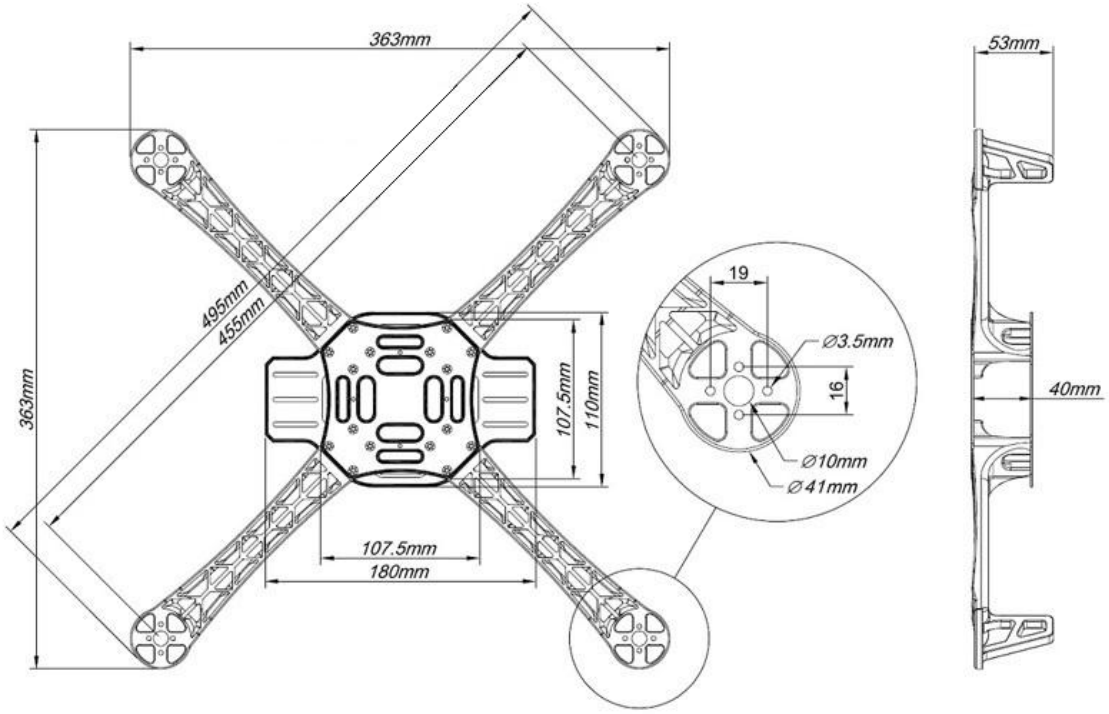

Figure 2.3.1 Overview of the drone

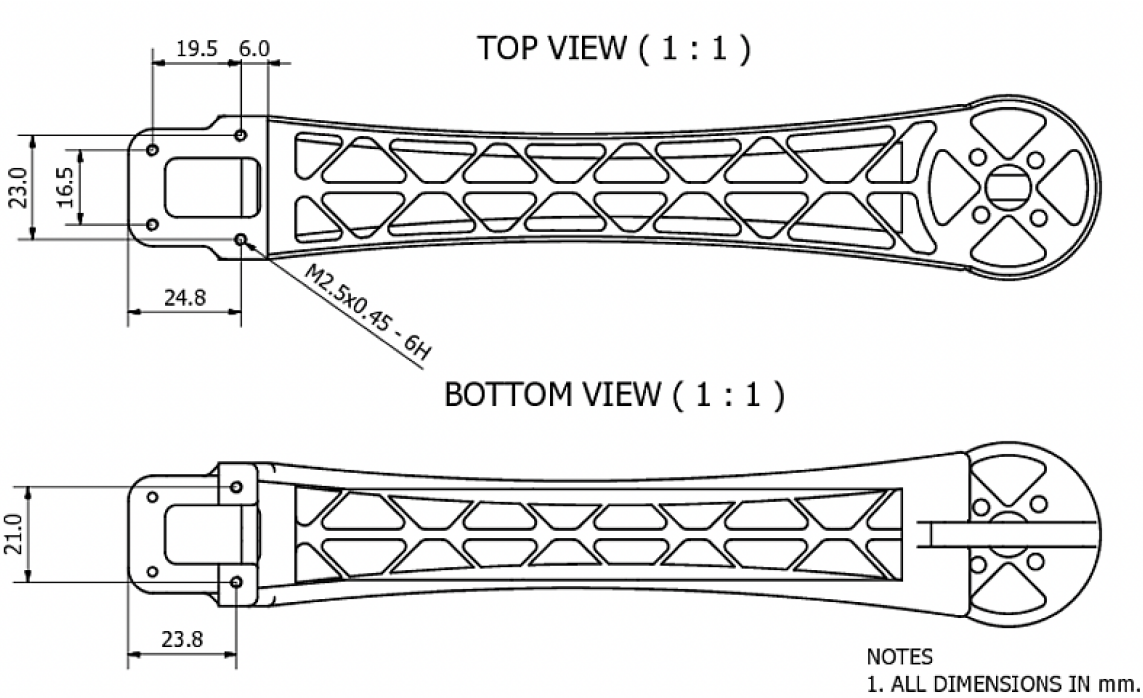

Figure 2.3.2 Detailed sketch of the wing

2.4 Manual to use the prediction & visualization platform

Figure 2.2.3 shows the GUI of network data prediction and visualization for users. Users can enter their current location information (longitude, latitude, and altitude) and press the “predict network speed” button. Then, the predicted network speed is displayed in the platform as well as the data visualization on the map.

3 Operation of the Project

3.1 Operating Mode 1: Normal Operation

Data Collection Mobile Application

Our data collection mobile application is written in Kotlin and uses the Ookla Speedtest SDK to automate the speed testing process for data collection. When a test is run, its results are automatically sent to our Google Firebase database for use in our predictive model and data visualization. There are options to continuously run tests every 10-20 seconds, even when the application is minimized. This process greatly reduces the time it takes to assemble data and allows for use even when separated from the mobile device, such as when it is being flown by a drone in the air.

To run speed tests:

1. Install application on Android mobile device
2. Open application
3. Activate GPS tracking slider
4. Select type of test
5. "Single Test" for one test, "Background Scan" for continuous testing
6. To end testing, select "Reinitialize SDK" and close application

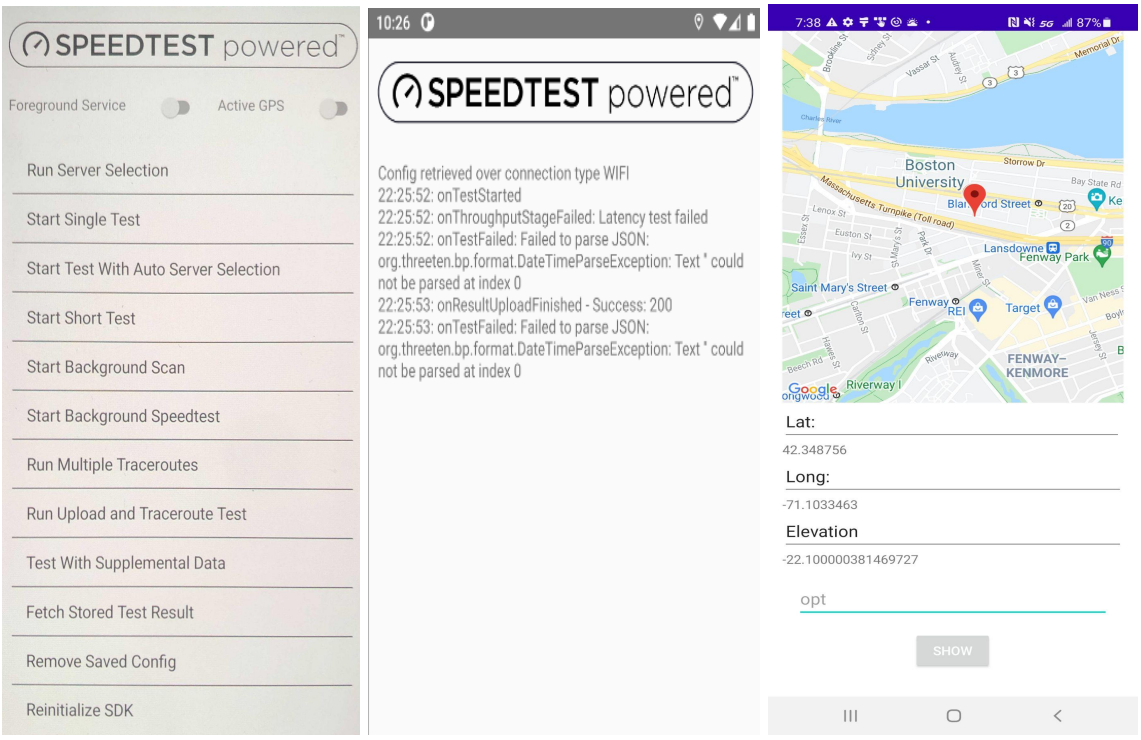

Figure 3.1.1 Left to right: Application home screen, active test screen, GPS display screen

Prediction and Visualization Flask Web Application

Our web application is made for use by the end user of our predictive model. The user is able to run a single speed test like that of our mobile application and see the displayed results.

To run speed tests:

1. Open web application
2. Press “Retrieve Data”
3. View results

Additionally, the user is able to select any location within proximity of collected data to display the predicted network speeds based on our machine learning model.

To predict network speed:

1. Open web application
2. Select Prediction Map tab
3. Scroll down to map
4. Click anywhere on map

The user is also able to navigate a heat map of collected data and mouse over the hotspots to view network speeds and altitudes at those locations.

To navigate heatmap:

1. Open web application
2. Scroll down to map
3. Use scroll wheel to zoom in and out
4. Use mouse to left click and drag and navigate around map

To reset the application or be refreshed in the web browser, press the Clear button.

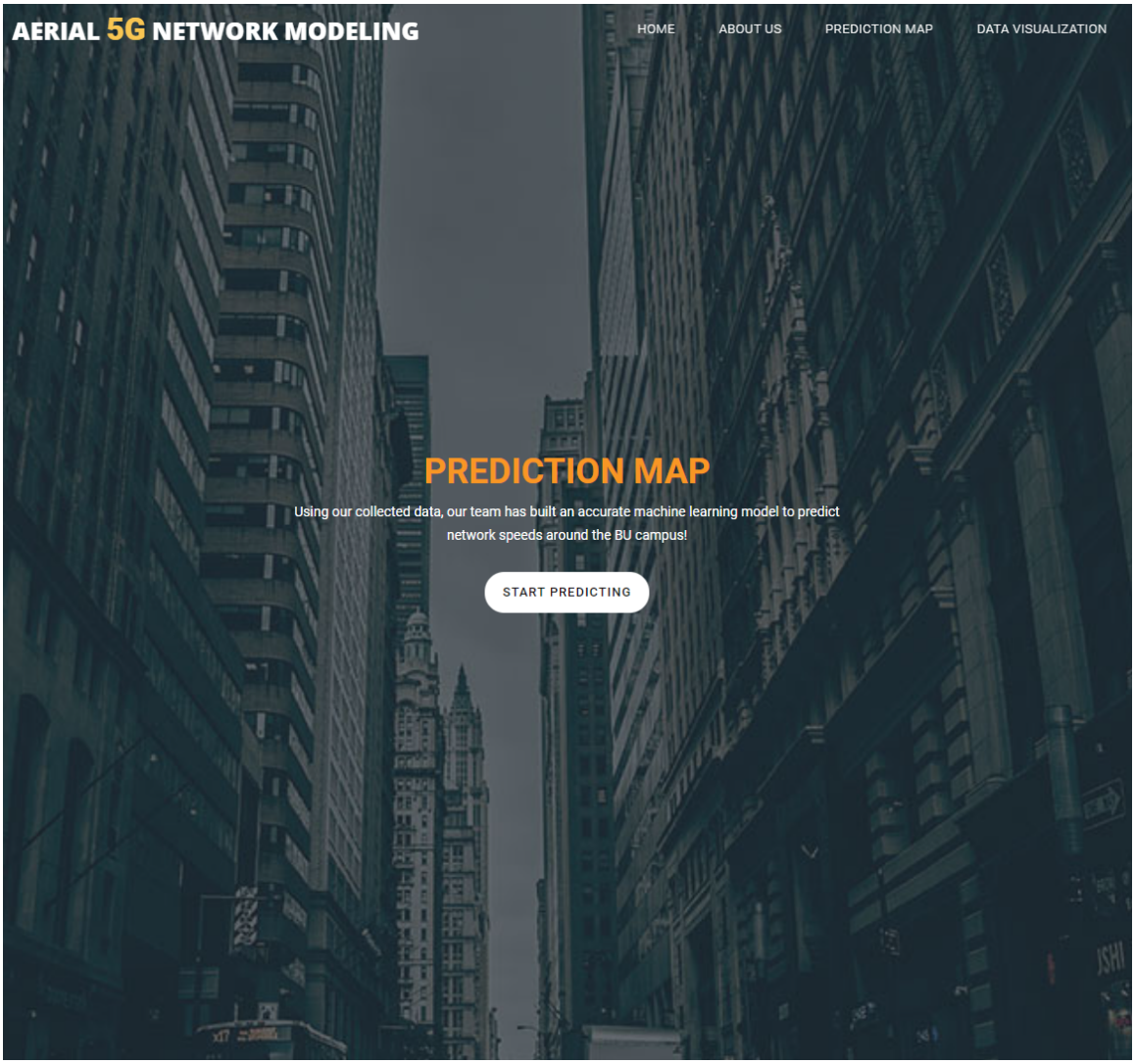

Figure 1.3 Web Application using Flask Framework

Figure 1.4 Prediction map of web application

3.2 Operating Mode 2: Abnormal Operations

Our applications are made to be streamlined for easy use in issues such as being unable to halt a process in the application. The application can be closed, which will reset the process. Additionally, the mapped geolocation display can be reset by refreshing the web application if there are issues with the application.

3.3 Safety Issues

The only safety concerns one may have when interacting with this project would be through the data collection portion. Drone operators should research flight restrictions in their local area as well as follow safe flight operation practices.

There is no risk of injury or harm from the use of our applications, nor is there risk of privacy or security breaches, as there is no collection or storage of user information or data.

4 Technical Background

4.1. Hardware Component

The DJI Flame Wheel Drone and the Android phone are the only necessary hardware components. The DJI Flame Wheel Drone kit contains a F450 Frame Kit, 2312E 800KV Motors, 30A ESC Brushless Speed Controllers, 10 inch FPV 1045 Propeller, 2x 3000mAh LiPo battery, and Flysky FS-i6X Transmitter.

4.2. Software Component

4.2.1 Data collection

Our data collection application can automatically and continuously run a speedtest on our application to collect data to be used in the visualization and ML steps while in flight. Without requiring human interaction, the results from the speed tests are automatically uploaded to a Firebase database where they can be extracted into a JSON file, as we demonstrated. This JSON file 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.

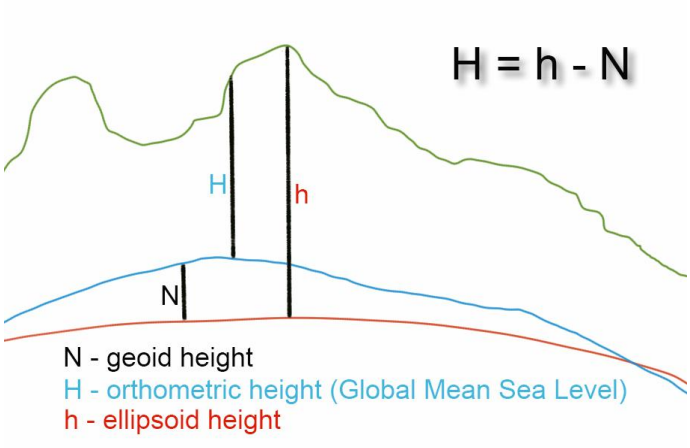

Figure 4.2.1

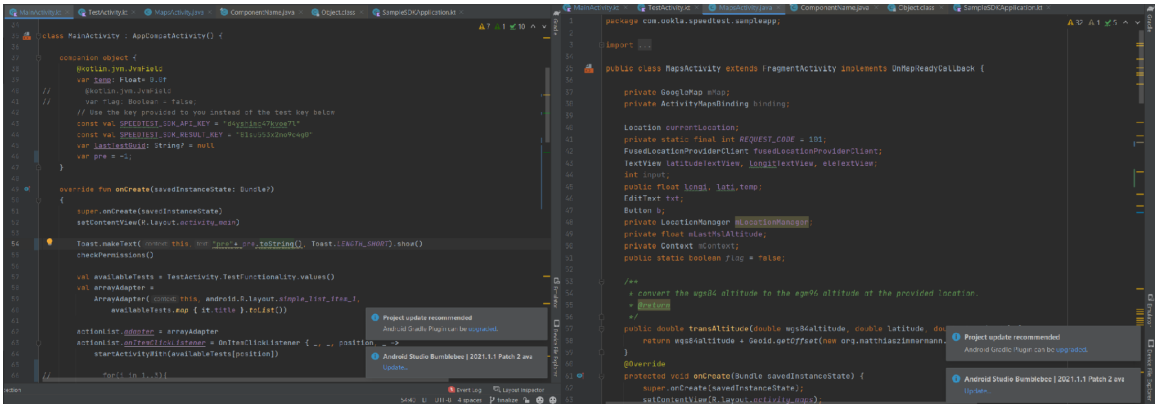

Figure 4.2.2 Automation/Compatibility Solution

4.2.2 Data Visualization

Data visualization is where our python script will take collected data to create a visual representation of testing locations and network speed. Users 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.

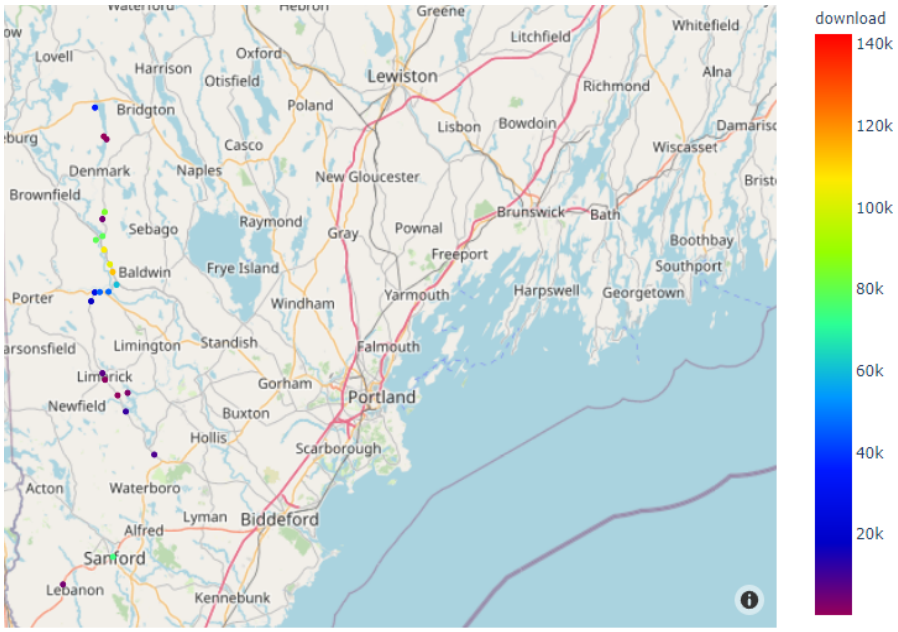

Figure 4.2.3 Mapped discrete data points

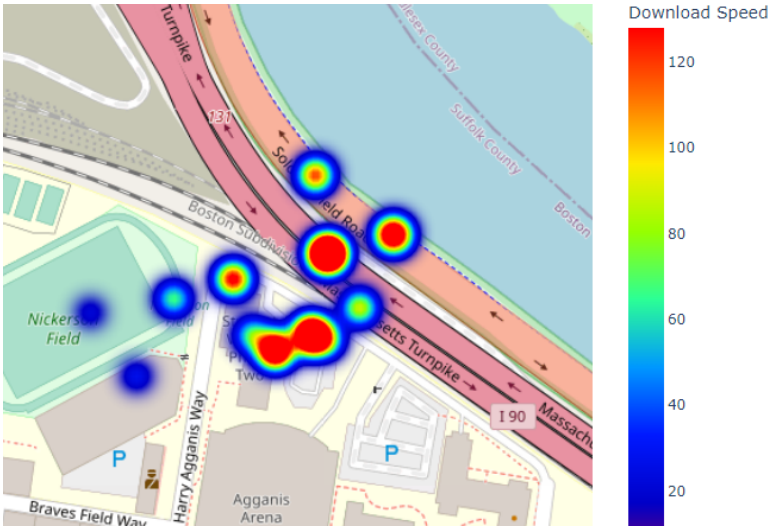

Figure 4.2.4 Heat mapping of collected data on BU campus

4.2.3 Machine Learning Model

We will take the data collected previously and construct a model based on clustered K Neighbors regression to be predictive for speeds at different altitudes and locations. Normalization such as z-score normalization is applied to improve performance. The model produces two outputs to the regressor model: the download and upload speed. This is predicted based upon the relevant inputs which are the longitude, latitude and altitude. In evaluating the model, we use the mean absolute error, mean square error, and root mean squared error along with the R squared score ("Accuracy").

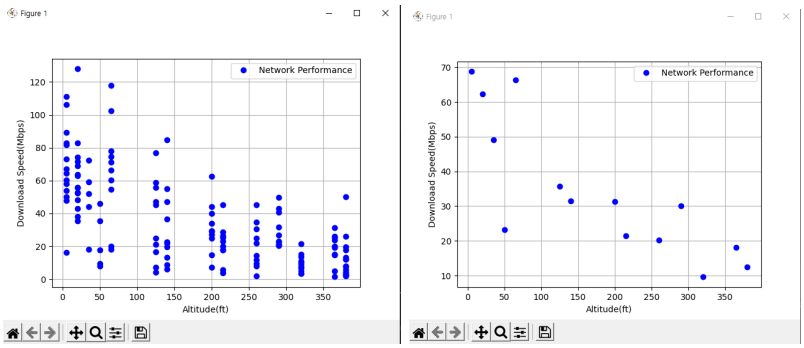

Figure 4.2.5 Translation of data from Pre-Clustered to Clustered.

```
Prediction for test set: [[23.07746048 16.16737515]
 [23.07746048 16.16737515]
 [34.76584544 21.87268896]]
Mean Absolute Error: 4.49937623111111
Mean Square Error: 38.10726126913231
Root Mean Square Error: 6.173107910050844
Accuracy:  58.37824889078716
```

Figure 4.2.6 Results

5 Relevant Engineering Standards

5.1 speedtest by Ookla(SDK)

5.2 Google Map API

5.3 HTML

5.4 Pandas

5.5 Numpy

5.6 Sklearn

5.7 matplotlib

5.8 scipy

6 Cost Breakdown

The budget of this project was spent on the hardware required for building and operating the drone for data collection. Our client was kind enough to provide our team with the required 5G cellular phone for the duration of our project. This was quite helpful in keeping our budget reasonable throughout the project. The other components of our project are software based which allows us to avoid spending in that area.

There were no specifications for price from the client, so our budget was based on the department's budget of $500 for completion of the project. Our project was able to stay within these budget limits, even with the need to buy a few replacement parts for broken and faulty parts of the drone.

The drone used for the data collection process was within the budget requirements for the project but not specifically made for the payload we equipped it with for this project. If one wished to purchase a more specialized drone for the task of carrying a phone or other cellular device, this could be a possible improvement for ease of use at increased cost.

Project Costs for Production			
Item	Description	Quantity	Unit Cost
1	QWinOut DIY F450 Quadcopter Dronekit	1	$235.88
2	ShareGoo Universal Tall Landing Gear	1	$7.89
3	25C 2200mAh 11.1V Lipo Battery 2 Pack	1	$25.19
4	Radiolink R8EF RC Receiver	1	$15.99
5	QWinOut V2.3 KK Flight Control Unit	1	$19.88
		Total Cost	$304.83

Figure 6.1 Production Cost Estimate Table

7 Appendices

7.1 Appendix A - Specifications

Team 17
Team Name: Aerial 5G Network Modeling
Project Name: Aerial 5G Network Modeling

Requirements	**Value, range, tolerance, units**
Power Supply	11.1V, 2200mAh for Drone
Drone Dimensions	Weight: 4.15lb Load-Weight: < 500g Dimensions: 10.39 x 10.28 x 5.28 in
Practical	Overall cost should be < $300 (excluding extra drone batteries)
Firebase Storage	10GB storage
Wireless Connection	5G connection is mandatory for the purpose of project
User Data	User's Network Data is stored for an unnoticed amount of time in firebase storage for continued data collection/model training purposes

7.2 Appendix B – Team Information

Team 17, 5G Drone Blue, consists of five computer engineering students: Yuan Chi Chung, Taehyon Paik, Minseok Sakong, Talbot Taylor and Stanley Zhang. The team found this project important as it offers an opportunity to improve understanding of 5G network performance at different altitudes and conditions. All took an interest in this project because of its focus on machine learning and for the opportunity to learn more about cellular and wireless communication networks.

Yuan Chi Chung is a graduating computer engineering student from Taichung Taiwan. His interests include mobile/web development and AWS. He is currently working for Snapbrillia,"an AI hiring platform that accelerates the quality of hires and DEI by enabling teams to interview better and hire the best people faster."

Taehyon Paik is a graduating computer engineering student from Arlington, VA. His interests are in automated testing, machine learning, and web development. He will be working full time as a

software engineer at Innovative Defense Technologies starting this summer to continue his career in the software engineering field.

Minseok Sakong is a computer engineering student at Boston University. His academic concentration is on machine learning. He is interested in machine learning, deep learning, and data engineering. He is currently working as a research assistant at Boston University of School of Medicine.

Talbot Taylor is a graduating computer engineering student from Washington, DC. He has an interest in machine learning, cloud computing, and mobile/web development. He is currently working part time with the Halvik Corp as a Robotic Process Automation engineer and plans on continuing his career and building his skillset.

Stanley Zhang is a graduating computer engineering student from New York, NY. He is interested in mobile/web development along with machine learning. He will be working full time as an associate software developer starting this summer.