Boston University
Electrical & Computer Engineering
EC463 Capstone Senior Design Project

First Prototype Testing Report

5G Network Performance Testing - Sky Seer

Team #16
5G Drone Team Red

Team Members:

Ryan Mondonedo, mondor28@bu.edu
Peter Wallace, peterjw@bu.edu
Jong Whee Jeon, crm04140@bu.edu
Dohun Ji, kaahsh38@bu.edu
Jungin Chang, changju@bu.edu

Setup and Materials

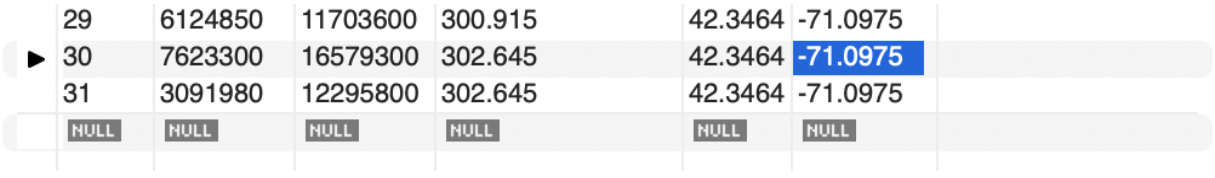
The setup that was carried out was consistent with our test plan. An Android phone was connected to a network and a script was run on it to collect network speed data. As mentioned in the test plan, the prototype test did not use an actual 5G network since the main purpose was testing our system's capabilities and we do not yet have the 5G SIM card from our client.

For our prototyping, the only physical materials we needed were a phone with an internet connection and a laptop to view the database as well as to run the machine learning model. As far as software materials, we used an AWS RDS SQL table for our data storage, PyCharm for running our python machine learning code, and Termux linux emulator for Android to run our data collection script on our test phone. As far as specific data collection materials go, we limited our prototype to a proof of concept gathering network speed data via the open source speedtest-cli for python, and got location data from the server that the phone connected to to simulate sending geolocation data from the phone to the database. We also used the pymysql library to integrate our SQL queries into our python script. For future prototyping we will be fetching and sending accurate gps data (coordinates + altitude) using either Android native hardware or the ipinfo API.

Measurements Taken

The measurements we took are as follows:

- Can the network speed be measured by the phone?
    - Results: yes, upload and download speeds could be collected. We measured the upload and download speed at 12295800 bits/sec and 3091980 bits/sec, respectively
- Can the data be pushed to the database?
    - Results: yes, data was pushed into our database (number 31) {'id','download_speed','upload_speed','distance to the best server','lat','lon'}



	29	6124850	11703600	300.915	42.3464	-71.0975
▶	30	7623300	16579300	302.645	42.3464	-71.0975
	31	3091980	12295800	302.645	42.3464	-71.0975
	NULL	NULL	NULL	NULL	NULL	NULL

- Can the data be imported into the machine learning model?
    - Results: yes, we were able to pull sample data taken from Kaggle from a sample SQL database to run the machine learning model locally on one of our team member's laptop
- Can the machine learning model calculate predictions based on the data?
    - Results: yes, we used a multi-out regression model from SciKit Learn on python, with a measured accuracy against test labels of 88%

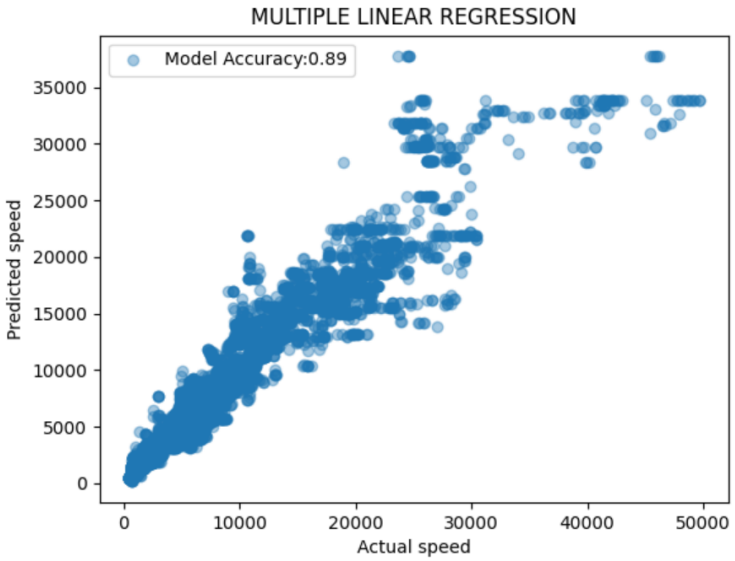


Fig 1. Plot produced by model, predicted speed vs actual speed through linear regression

Conclusions

Data collection:

We were able to obtain reasonable and consistent test results including upload speed, download speed, and distance to the best server. Since the network speed tests were taken in the same place several times, the collected data of distance to the best server, upload speed, download speed had a low variation over several Ookla network speed tests.

Every collected data for each network speed test including gps coordinate (latitude, longitude), upload speed, download speed, and distance to the best server was successfully pushed into the database and synchronously updated.

Machine Learning:

A sample speed dataset from Kaggle was used to simulate the machine learning model that will be produced in this project, which will have a similar format. Country code, total tests, and distance formatted as training variables will be replaced by gps coordinates and altitude once our data collection system is established. Targets of the model are download and upload speed

The result predicted through the sample test dataset has an error range of 100kbps that is less than the standard deviation of the original dataset. We achieved model accuracy over 88% and produced a graph showing linear regression between predicted and actual labels, as seen in Fig 1. Going forward in this project, we can work to optimize accuracy of the machine learning model even more.