

EE/CprE/SE 4920 STATUS REPORT 3

09/19/24 - 10/03/24

Group number: 22

Project title: CyRide Visualization

Client: Mohammed Soliman

Advisor: Mohamed Selim

Team Members & Role:

Braden Buckalew: Programmer

Endi Odobasic: Programmer

Evan Schlarmann: Programmer

Andrew McMahon: Programmer

Chiran Subedi: Programmer

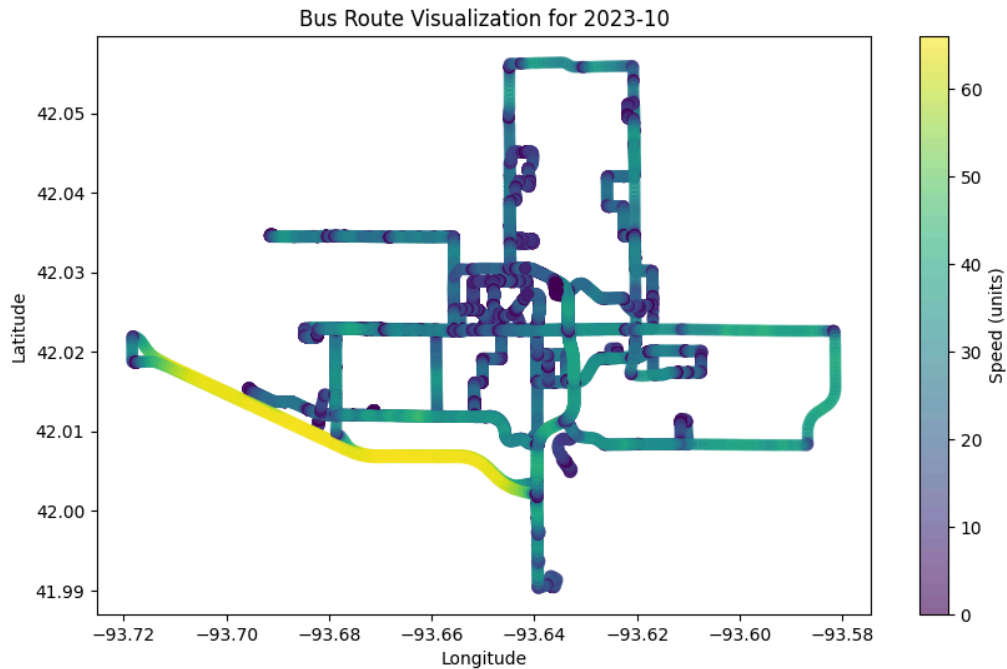
Week Summary

This week our team aimed towards building on the historical data and how the data will be used in the application. We also started the groundwork for how we will get live data. Lastly, we found a new map API instead of Google Maps to reduce costs.

Accomplishments

Created a new web socket that gets the most recent location of a bus. This uses an HTTP request to mimic calling the UE and sends the data to its connected users. When the request timeouts or fails, it will instead use machine learning to get the data. The HTTP URL can be replaced when access to the live data is given.

Created monthly graphs of historical data to pinpoint and understand what our data is giving us. We discovered One bus for an entire year has done multiple routes instead of just one route. We also discovered sometimes the bus went off route or was out of service and gave us GPS data not on a road.



Pending Issues

The bus UE is non-functional and has no connections for live bus data.

Individual Contributions

<u>NAME</u>	<u>Individual Contributions</u>	<u>Description</u>	<u>Week Hours</u>	<u>Cum. Hours</u>
Evan Schlarman	<ol style="list-style-type: none"> Created web socket to get the newest location Updated documentation for new dependencies 	<ol style="list-style-type: none"> The web socket pulls all the bus IDs and calls our HTTP endpoint to get locations. This will be changed later to call the UE for each bus ID. The web socket has a runtime of 4 seconds to match the historical data given and will call the machine learning algorithm if the requests can't reach the UE. The documentation needed to be updated to match new dependencies for Python HTTP requests 	7	83

Braden Buckalew	1. Analyze Historical Data	1. Wrote a Python script to analyze our year long data csv file into separate monthly files. Each file has a graph with longitude, latitude, and speed.	7	76
Endi Odobasic	1. Interactive Maps 2. SARIMA Machine Learning Model	1. Converted and implemented our old map to use our new interactive map to fit the project. No more using Google Cloud services for their maps, now we use Leaflet with OpenStreetMap(OSM). 2. More research into the subset of the ARIMA, SARIMA model and see how it fits with our project. Proving to be difficult to research and find things out. No usage yet.	7	79
Andrew McMahon	1. Prepare frontend for historical data 2. Develop script for data cleaning	1. Learned frontend syntax and prepared all endpoints for historical data display 2. Began to format script for data cleaning with Braden. Since the historical data is uncleaned and contains multiple routes, we need to develop this script to map individual route data for display	7	75
Chiran Subedi	1. Reviewed project documents from last semester. 2. Explored GitLab to look at implementations.	1. Examined key documents from last semester, including final design reports, task breakdowns, and app documentation, to better understand the project's scope and status. 2. Investigated the code repository on GitLab, reviewing the structure and implementation details.	7	7

Plans For the Upcoming Weeks

- Extrapolate the historical data sets into their respective routes and filter out any outlying data.
 - Decide on a machine learning model and create sample implementations
-

Weekly Client/Advisor Meeting Summary

During the meeting, we discussed the data analytics and saw that the bus data was following different routes through the months. This data must be parsed to separate the routes and remove outlying data. Since we will soon have the data parsed and ready, we also discussed a plan to find a machine-learning algorithm and its implementation.