

## **EE/CprE/SE 491 WEEKLY REPORT 5**

**Feb 28 - Mar 5**

**Group number: 20**

**Project title: GridGPT**

**Client &/Advisor: Gelli Ravikumar**

**Team Members/Role:**

- **Tin Ngo -> AI Integration Specialist**
- **Jackson Phillips -> AI Integration Specialist**
- **Emma Heithoff -> Power Systems Specialist**
- **Eddy Andrade -> Frontend Lead**
- **Nick Doty -> Backend Lead**

### ○ **Weekly Summary**

- This past week, we met with the graduate students assisting our project within the context of their broader work. We discussed that the master dss file data will be extracted to the Neo4j database using scripts that allow the database to utilize the information as well as requirements for generating scripts from user input. Additionally, they mentioned our access to the VM and computing environment will be available shortly. Our learning in and outside of meetings will be applied in cohesive practice with this access.

### ○ **Past week accomplishments**

- Tin Ngo: Attempted to create an application that would generate code given a prompt. Tested prompts that are similar to “generate a Python function that would find the sum of 2 numbers”. Using the pipeline, a high-level functionality of the AI model, took too long. This problem stems from not having enough computational resources. I realized that I was using a model with too large of parameters. I moved to a smaller model that had 137 million parameters, and I could get a response from a prompt, but it was incoherent. I also found a smaller dataset to train this smaller AI model with. Hugging Face estimates a cost of \$103,729 to train a 7-billion parameter text model. I selected the cheapest options for this estimate.
- Jackson Phillips: Last week, I focused more on what libraries to use for training, and this week I focused more on how to use PyTorch to train. It seems relatively simple at a surface level, with a script inputting the model, dataset, and any other parameters. I'm sure it will become more complicated as we get deeper into it.
- Emma Heithoff: I met with the team and graduate student's on the grid simulations done by previous design groups. The presentation connected my machine learning knowledge to the DSS research I did during the last period. I researched areas from the other members of the project after benefiting from seeing the power specialist role needed in the context of the graduate student's work. I did not work on the project as much as usual due to a non-academic constraint.
- Eddy Andrade: I got to get a better view of the project on the Frontend side and I had a much better

understanding of how our implementation of GridGPT will work on this project. I also got to see the different types of directories and libraries that are being used in the project, and what they do within. Getting to know how our implementation will come into play helps a lot with how I can approach the implementation of GridGPT.

- Nick Doty: After talking with the grad students working on the project, I got a better understanding of what I need to do to fully understand Docker, InfluxDB and Neo4jDB. One of these clarifications was to use a linux machine instead of windows, which I previously started with. After a small learning curve, I was able to get Docker and InfluxDB successfully running on my personal linux computer.

○ **Pending issues**

- Tin Ngo: We don't have access to the current infrastructure (HPC) to start training or using models. Creating an application that leverages a hugging face model was difficult because it would take so much computational power that I don't have.
- Jackson Phillips: We still need access to our VMs from our advisor so we can start getting hands-on with these models.
- Emma Heithoff: I also would like to work within the VMs with the AI specialists to apply my research and answer some questions through experience.
- Eddy Andrade: No pending issues currently. Everything is good on my end.
- Nick Doty: N/A

○ **Individual contributions**

<b><u>NAME</u></b>	<b><u>Individual Contributions</u></b> <i>(Quick list of contributions. This should be short.)</i>	<b><u>Hours this week</u></b>	<b><u>HOURS cumulative</u></b>
Tin Ngo	Application experiment and research	6	33
Jackson Phillips	Training research	6	32
Emma Heithoff	DSS application for project goals	4	28
Eddy Andrade	Next.js documentation research	6	32
Nick Doty	InfluxDB and Docker setup and research	6	32

○ **Plans for the upcoming week**

- Tin Ngo: Continue Research on how to optimize training AI models on Hugging Face. Dive deep into the current infrastructure (we will have access to it soon). Draft out a plan on how to train AI model and utilize it within existing infrastructure.
- Jackson Phillips: Ideally, we will get our VMs ASAP so I can start running training and other tasks to learn more by getting hands-on experience with this technology.
- Emma Heithoff: Primarily, I want to integrate my .dss research with each member's progress as well as the grad student's work so far.
- Eddy Andrade: Continue researching on Next.js framework to develop a better understanding on the implementations.

- Nick Doty: This week I will continue my research in Docker, Flask, and InfluxDB. Moreso, I will research deeper into Neo4jDB (Since last week was mainly focused on InfluxDB) to get a better understanding of it and hopefully get it running on my personal linux computer alongside InfluxDB.

○ **Summary of weekly advisor meeting**

1. Handling DSS Files

- DSS Files: These are typically associated with the Hydrologic Engineering Center's Data Storage System (HEC-DSS), used in water resources applications. Each DSS file contains datasets, which might include time series data, spatial data, etc.
- Unzipping and Parsing: You'll need to write or utilize existing code to unzip the file containing all the DSS files and then parse each DSS file to extract the relevant data.

2. Building the Project on the Master DSS File

- Master DSS File: This file likely serves as the central hub for your data, possibly containing references to or aggregations of data from the other DSS files.
- Data Extraction and Transformation: Code will be required to extract and transform the data from the master DSS file into a format that can be ingested by Neo4j.

3. Connecting Data to Neo4j

- Neo4j Database: Neo4j is a graph database management system, ideal for storing and querying connected data.
- Data Ingestion: You'll need to develop scripts or use existing tools to ingest the transformed data from the DSS files into Neo4j, creating nodes and relationships as defined by your project's schema.

4. Front-End Development

- Displaying Neo4j Nodes on a Map: The front end will require a mapping library (like Leaflet or Google Maps) to display the spatial data. You'll need to query the Neo4j database to retrieve the nodes and their parameters, then visualize them on the map.
- Pop-Up in Text Editing Window: For editing or displaying additional information about the nodes, implement a pop-up or modal in the UI that interacts with the text editing window.

5. Code Generation

- Automatic Code Generation: Implementing a feature for automatic code generation based on user input (e.g., comments or specific commands) will require a combination of parsing the input, understanding the intent, and generating the corresponding code snippets. This targets our first use case