

# EE / CprE / SE 492 – sdmay21-23

## Grid AI

### Week 2 Report

2/9/21 – 2/22/21

Client: Dr. Ravikumar Gelli

Advisor: Dr. Ravikumar Gelli

#### Team Members:

Justin Merkel — *ML Developer, Backend Developer*

Patrick Wenzel — *Frontend Developer*

Abhilash Tripathy — *Frontend Developer*

Karthik Prakash — *Backend Developer*

Abir Mojumder — *Backend Developer*

#### Weekly Summary

One of the main objectives for the frontend these past couple weeks were to Dockerize ReactJS and Flask and then be able to run the material-ui-dashboard. We ran into an issue where the path finding in the Docker container didn't seem to be working correctly which neither us nor Dr. Gelli could figure out. However, Dr. Gelli was able to help us find a possible solution that we believe will work. In the upcoming weeks, we will figure this issue out, and get the material-ui-dashboard up and running.

While we were trying to Dockerize ReactJS, there was also an effort to set up an end-to-end connection between the frontend and the backend. A new plugin for ReactJS called "use-neo4j" was used to make a connection between React and neo4j, while this approach means that some queries could be made from the database directly in the frontend, it had fairly restrictive methods for use, for example where exactly in the Context a query can be made, etc. So it is better to do the queries in Flask (flask-neo4j already done), and simply calling the api.routes within React which receives data in Json format.

#### Past Week Accomplishments

- Although we haven't been able to successfully Dockerize ReactJS for the frontend, Dr. Gelli has helped us find some promising resources on how to set up our container correctly to be able to get ReactJS in a Docker container.
- Successfully initialized Neo4j Docker and imported test data
- Implemented a new model that predicts the future kWh output of a node (rather than look at anomalies) in order to produce a simpler full stack product that allows the frontend to request new predictions that are more easily represented than the anomalies to focus on a minimum viable product. - Justin

#### Pending Issues

- material-ui-dashboard not compiling on the frontend
- Expand the data preparation scripts to include all 240 nodes from just feeder A. -Justin
- Figure out required query methods for Flask, to be used by the webapp.
- Efficiently format data inside Neo4j

- Potentially set up a different database for better data configuration

### Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Patrick Wenzel	Tried to set up a Docker container for ReactJS. Also tried to set up a ReactJS/Flask container but decided to do those separately and to set up the ReactJS container first.	6	13
Justin Merkel	Created a new ML model to predict the future output of a given node. Figured out how to save/load models into python.	6	16
Abir Mojumder	Experimented React-neo4j endpoints using a new plugin. Decided to use a more traditional method of neo4j->flask->react instead.	5	12
Karthik Prakash	Initialized Neo4j Docker and successfully imported sample test data	7	12
Abhilash Tripathy	Worked with Patrick to manage various project dependencies on the frontend as well as docker configurations	6	12

### Plans for Coming Week

- Patrick - Get ReactJS in a Docker container as well as get Flask in a Docker container and then be able to load the material-ui-dashboard
- Full 240 scripts and potentially hyperparameter tuning and supporting the backend integration of the model-Justin
- Abir - Start writing api.route methods in flask to send query data to frontend as required.
- Karthik - Import more data in a reasonable format to Neo4j and work on backend endpoints
- Abhilash - Making homepage layouts, basic react components and endpoints