EE / CprE / SE 492 – sdmay21-23
Grid AI
Week 4 Report

3/2/21 – 3/15/21
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 — Frontend/Backend Developer

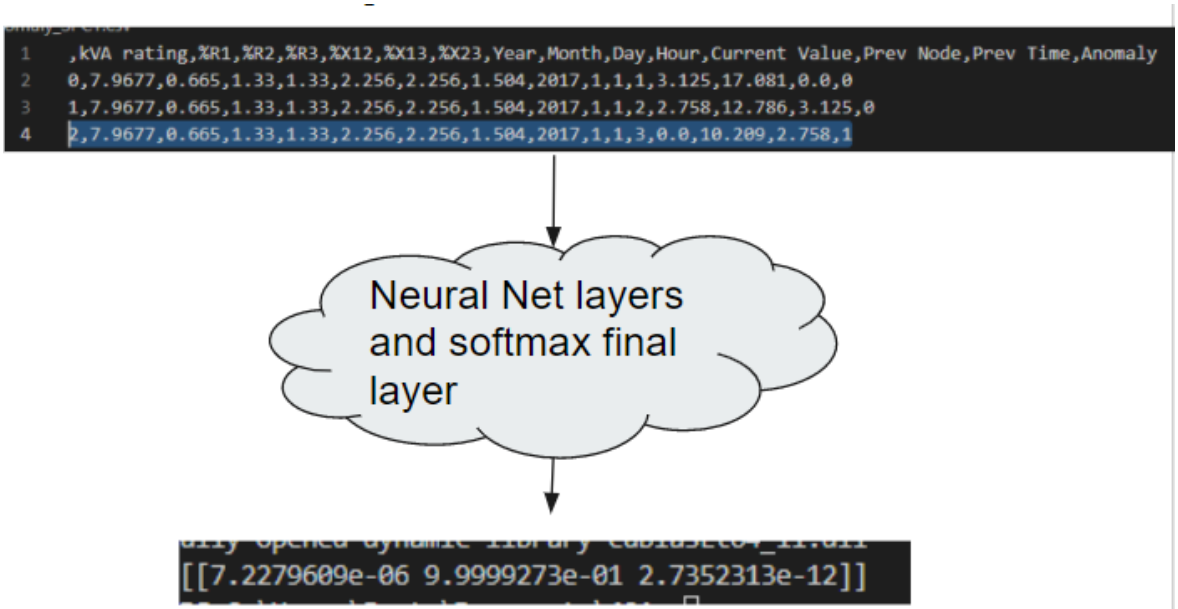
Weekly Summary

For the past two weeks, one of the main objectives for the frontend was to be able to route the different pages of the dashboard with Flask. There has been no breakthrough with this so far so it will be discussed for time reasons if using the already existing NodeJS routing system will work fine for this project and just use the Flask routes for making database calls to send and receive data.

The work for the Machine Learning models is wrapping up now that we were able to create a model for the purposes of anomaly classification and detection. The models can still be improved by adding in more parameter tuning but is not necessary to meet our goals or requirements. The only thing left in this aspect is to get anomaly models for the other types of transformers.

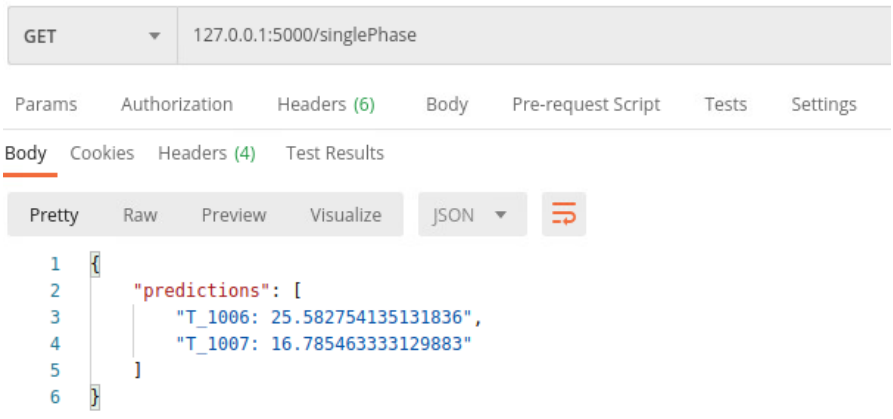
Past Week Accomplishments

- Justin-
    - Investigated the logistic regression for anomaly detection. Determined 3 classes of data that could be used for classification.
        - Normal/no anomaly
        - Power Spike
        - Power Failure
    - Created a script to add in the two anomaly types into data so that it could be used to train a logistic regression model.
    - Used keras to create a logistic regression model using softmax function n=3. The n=3 means that there will be 3 values that sum to 1 (represents probability)



-
    - This is showing the model in progress, The input is line 4 which has an anomaly class of 1 (power spike)
    - Then that data goes through the ML model layers and outputs 3 values corresponding to the probability that the data input is each class. [normal, power spike, power failure]
    - This example shows we correctly predict 99.999273% chance this is a power spike

- Karthik -
    - Added sample of time-series data into Neo4j database for testing/proof-of-concept purposes
    - Integrated prediction models of each transformer type for kWh output into Flask API
        - Flask and Neo4j Docker containers are connected for easy data retrieval and processing
        - Code is pushed to master branch for whole team to access



GET 127.0.0.1:5000/threePhase

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "predictions": [
        "T_1003: 19.32233428955078",
        "T_1004: 6.887632369995117",
        "T_1005: 5.431519508361816",
        "T_1008: 14.86520767211914",
        "T_1009: 8.719930648803711",
        "T_1010: 6.841784954071045",
        "T_1013: 3.5457944869995117",
        "T_2002: 47.757904052734375",
        "T_2003: 13.345588684082031",
        "T_2005: 8.898749351501465",
        "T_2010: 2.4334750175476074",
        "T_2011: 6.058200359344482",
        "T_2029: 10.419930458068848",
        "T_2030: 6.6460490226745605",
        "T_2032: 1.416085958480835",
        "T_2034: 4.1469011306762695",
        "T_2035: 5.82332706451416",
        "T_2037: 5.3579182624816895",
        "T_2040: 104.3635025024414",
        "T_2041: 3.4584693908691406",
```


GET 127.0.0.1:5000/singlePhaseCT

Params Authorization Headers (6) Body Pre-request

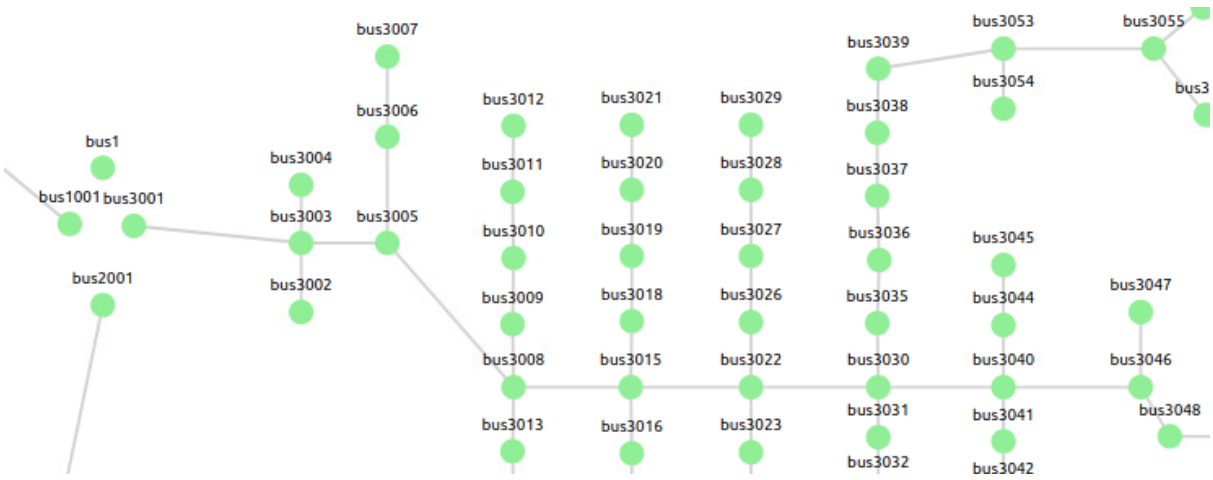
Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "predictions": [
        "T_1011: 3.5967936515808105",
        "T_1012: 1.3217524290084839",
        "T_1014: 0.6054900288581848",
        "T_1015 : 0.7233574390411377",
        "T_1016: 1.5135319232940674",
        "T_1017: 4.150126934051514",
        "T_2008: 2.1413440704345703",
        "T_2009: 1.1424473524093628",
        "T_2014: 3.4769840240478516",
        "T_2015: 1.3328561782836914",
        "T_2016: 2.4273345470428467",
        "T_2017: 3.80916166305542",
        "T_2018: 3.131518602371216",
        "T_2020: 3.6322054862976074",
        "T_2022: 2.11201810836792",
```


Abir -

- Use link properties from database to create corresponding links between nodes



Pending Issues

- Being able to route the different pages with Flask
- Fully end-to-end transactions running on master vm
- Implement real-time data updates for kWh outputs with new MySQL instance and task scheduler
- Integrate logistic regression models into relevant endpoints
- Import node relationships for Neo4j
- Allow for new data to be uploaded into databases

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Patrick Wenzel	Trying to route pages using Flask	4	28
Justin Merkel	Logistic regression investigation and implementation for one of the transformer types.	10	33
Abir Mojumder	Node linking in the graph	4	24
Karthik Prakash	Configured endpoints for transformer prediction models that uses values from Neo4j	5	25
Abhilash Tripathy	Refactoring table displays	4	24

Plans for Coming Week

- Patrick - Compare Flask routing vs NodeJS routing and pick which option is best for us
- Justin - Complete the logistic regression for the other two transformer types using the same process.
- Abir - Make pop-up tables to show current and predicted value for the node that is clicked on from the graph diagram.
- Karthik - Implement real-time updates to kWh outputs stored in Neo4j and help with end-to-end integration
- Abhilash - Restructure frontend architecture, make UIs for the new APIs in backend.