

-EE / CprE / SE 492 – sdmay21-23

Grid AI

Week 6 Report

3/29/21 – 4/12/21

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*

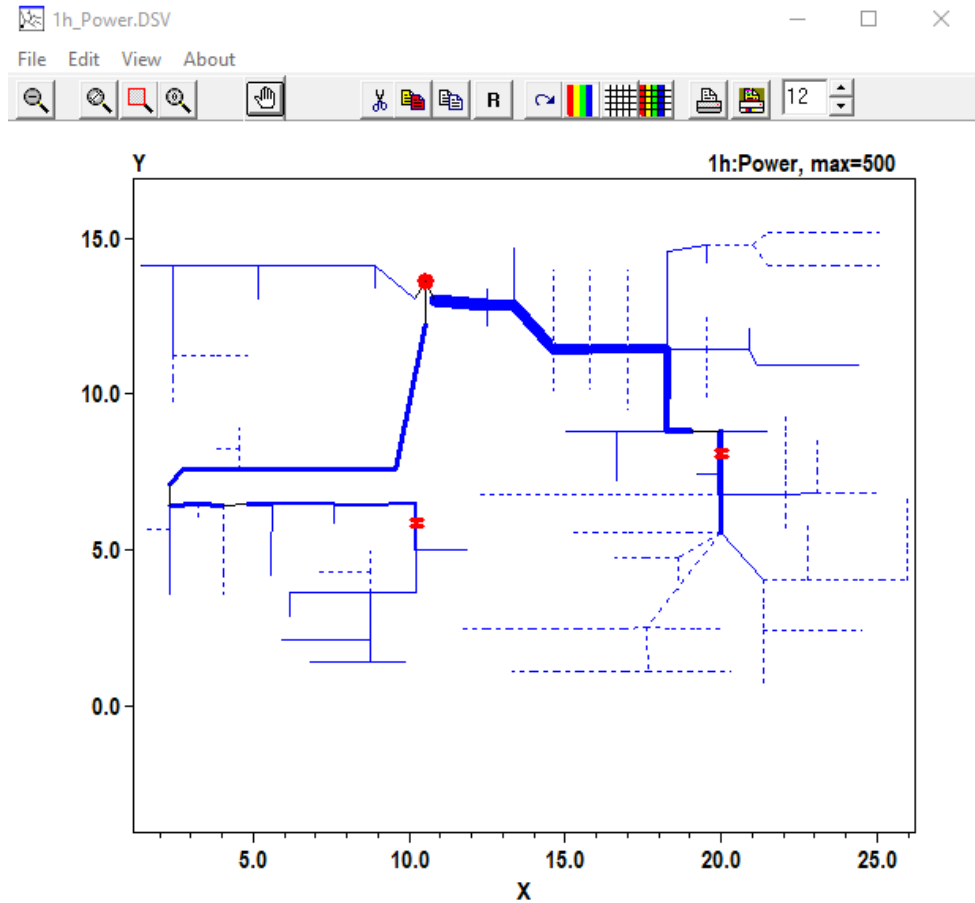
Abir Mojumder — *Frontend/Backend Developer*

Weekly Summary

During this past work period, the frontend has created pages for each of the individual components and they also have their own custom routes. Along with that, the backend has implemented the dynamic time-series data updates. The Flask API has a scheduled background task that updates the necessary properties in the Neo4j nodes using the smart meter data from the MySQL instance.

Past Week Accomplishments

- Justin-
 - Started simulating with the openDSS model and working on the scripting to run simulations to generate new data for demonstrating the final product.



- - Showing the powers of the OpenDSS grid

- Karthik -
 - All of the smart meter data has been stored in the MySQL updates
 - Created endpoint for getting the past 24 hours worth of data for a specified bus

```
@app.route('/history/<bus>')
def get_table(bus):
    b = datetime.now(timezone(timedelta(hours=-5)))
    query=f"select date,`Bus {bus}` from smart_meter where date between date_sub('2017-{b:%m}-')
    result = mysql_conn.execute(query)
    output = []
    for row in result:
        output.append([str(row[0]) + ':' + str(row[1])])
    return {'result': output}
```

GET 127.0.0.1:5000/history/1017

Params Authorization Headers (6) Body Pre-reqs

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": [
3     "2017-04-11 21:00:12: 6.773",
4     "2017-04-11 22:00:12: 3.064",
5     "2017-04-11 23:00:12: 3.146",
6     "2017-04-12 00:00:12: 3.05",
7     "2017-04-12 01:00:12: 1.536",
8     "2017-04-12 02:00:12: 1.747",
9     "2017-04-12 03:00:12: 2.406",
10    "2017-04-12 04:00:12: 2.896",
11    "2017-04-12 05:00:12: 2.118",
12    "2017-04-12 06:00:12: 3.01",
13    "2017-04-12 07:00:12: 5.827",
14    "2017-04-12 08:00:12: 7.654",
15    "2017-04-12 09:00:12: 9.125",
16    "2017-04-12 10:00:12: 3.391",
17    "2017-04-12 11:00:12: 1.623",
18    "2017-04-12 12:00:12: 1.809",
19    "2017-04-12 13:00:12: 1.741",
20    "2017-04-12 14:00:12: 2.216",
21    "2017-04-12 15:00:12: 6.256",
22    "2017-04-12 16:00:12: 5.378",
23    "2017-04-12 17:00:12: 6.388",
24    "2017-04-12 18:00:12: 4.213",
25    "2017-04-12 19:00:12: 6.462",
26    "2017-04-12 20:00:12: 9.407",
27    "2017-04-12 21:00:12: 8.942"
28  ]
29 }
```

- Neo4j nodes are updated every hour since that is the rate at which data is provided

```
2021-04-12 10:00:00.007346-05:00
2021-04-12 11:00:00.009797-05:00
2021-04-12 12:00:00.008967-05:00
2021-04-12 13:00:00.003335-05:00
2021-04-12 14:00:00.006844-05:00
2021-04-12 15:00:00.005467-05:00
2021-04-12 16:00:00.008966-05:00
2021-04-12 17:00:00.003906-05:00
2021-04-12 18:00:00.004901-05:00
2021-04-12 19:00:00.011704-05:00
2021-04-12 20:00:00.003146-05:00
2021-04-12 21:00:00.002017-05:00
(base) ubuntu@ubuntu-vm:~/Desktop/sdmay21-23/Hello_World_Projects/pwenzel/api$
```

9:56 PM 4/12/2021

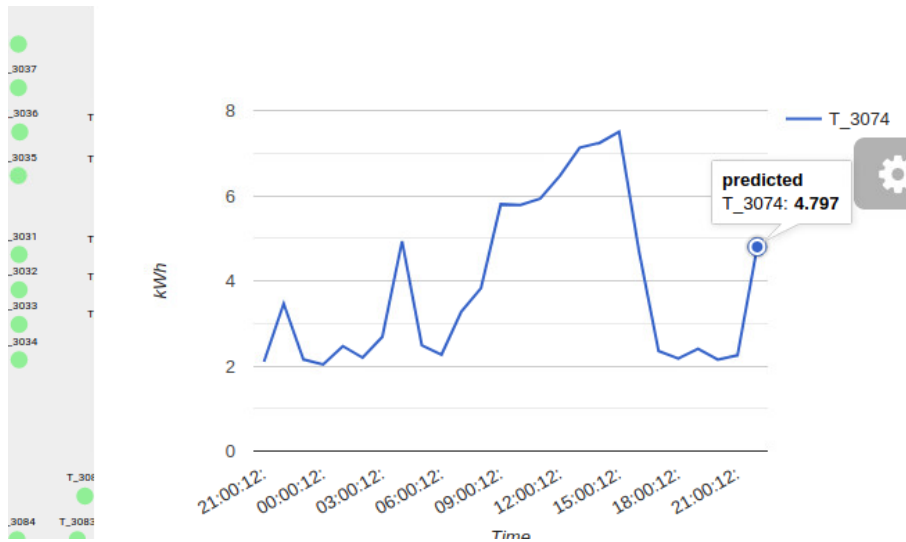
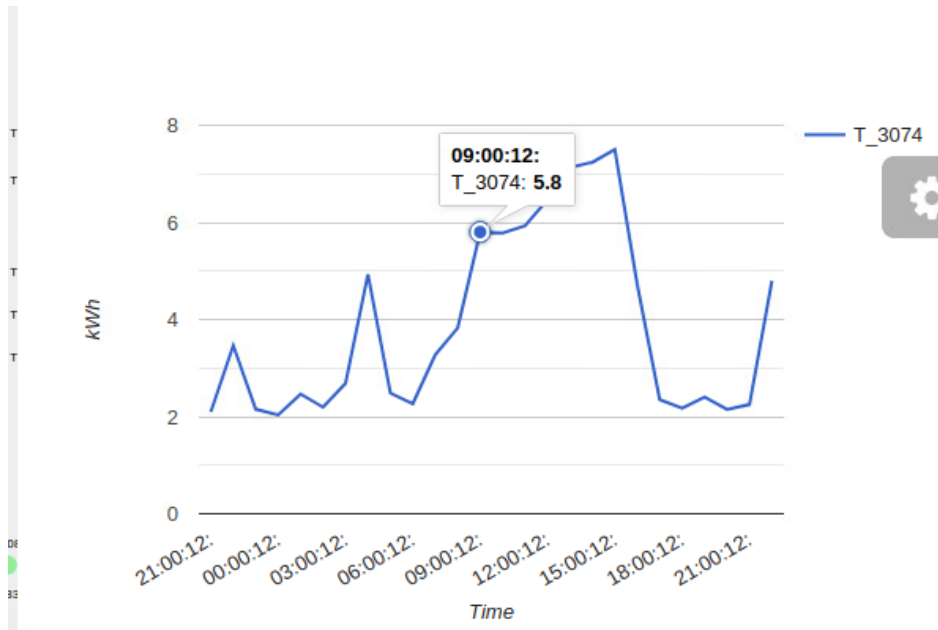
```
n                                prev

{                                {
  "identity": 553,                "identity": 552,
  "labels": [                     "labels": [
    "Single Phase CT"             "Single Phase CT"
  ],
  "properties": {                 "properties": {
    "%X13": 2.076,                "%X13": 2.16,
    "%X12": 2.076,                "%X12": 2.16,
    "%X23": 1.384,                "%X23": 1.44,
    "BusID": "T_1017",            "BusID": "T_1016",
    "PrevNode Val": 6.844,        "PrevNode Val": 0.936,
    "Hour": 21,                   "Hour": 21,
    "%R3": 1.19,                  "%R3": 1.1,
    "%R2": 1.19,                  "%R2": 1.1,
    "%R1": 0.595,                 "%R1": 0.55,
    "CurrVal": 8.942,              "CurrVal": 6.844,
    "Previous Bus": 1016,          "Previous Bus": 1013,
    "PrevVal": 9.407,             "PrevVal": 5.334,
    "Month": 4,                   "Month": 4,
    "Year": 2017,                 "Year": 2017,
    "Voltage rating of Winding 1 (kV)": 7.9677,
    "X": 2.4,                     "X": 2.4,
    "Y": 9.73,                    "Y": 10.48,
    "Day": 12                      "Day": 12
  }
}                                  }
```

Abhilash -

- Clean up the homepage by removing redundant default-UI material ui components.
- Discussed new features to implement on the frontend.

Abir - Line data displaying past 24hr values and predicted data:



Note: Previous 24hr data was not live, so predicted data doesn't look like it matches any patterns.

Patrick:

- Put individual components into their own pages and routed them. (Cannot show picture due to issue below)

Pending Issues

- Docker container not showing changes even after running docker-compose down and restarting the container and is showing work prior to creating custom routes and pages for individual components.

- Add nodes that are not listed in distribution transformer sheet within system description
- Import node relationships for Neo4j
- Allow for new data to be uploaded into databases
- Setting up scripts to allow the master VM to use a stable or latest version of the application. Add tag system in github repo.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Patrick Wenzel	Determining best setup for routing	3	42
Justin Merkel	Experimenting with OpenDSS and simulation. Started the scripting for it.	6	47
Abir Mojumder	Line chart now displays 24hours history data and predicted value per node. Research on modifying node properties (color, zoom, select node) through external components.	4	35
Karthik Prakash	Configured MySQL instance with all smart meter data. Integrated MySQL to update transformer nodes in Neo4j instance. Added some endpoints to access extra time-series information.	8	40
Abhilash Tripathy	Clean up UI on homepage	3	35

Plans for Coming Week

- Patrick - Get the container to run correct code again.
Clean up interface
- Justin - Finish the openDSS scripts and integrate into backend
- Abir - Updating GUI with new features based on client feedback:
 - Add page display anomaly status of all nodes.
 - Line chart allows comparison with other nodes by overlapping the line data
 - Adding static power grid information for display to the user on the front page.
- Karthik - Create more endpoints for getting time-series data from MySQL and transformer information and look into scripts for uploading data from files to database instances.

- Abhilash - Make new components that can be used on the homepage. Constrain the line chart and node graph, so that they don't overlap.