GTFS Real-Time Public Transport (PT) towards Digital Twins (DT): PTDT

Dicky Sentosa Gozali

School of Civil and Environmental Engineering, Queensland University of Technology, Brisbane, Australia

1 GTFS Real-Time (RT) Public Transport (PT) Monitoring towards Digital Twins (DT) Project

A real-time public transport monitoring from web crawling of publicly available and universally accepted dataset, General Transit Feed Specification (GTFS), allowing one to monitor the movement of the public transport across various modes (e.g., Bus, Trains, Ferry). The project is a WebApp that is run under Flask Server.

Author and Creator: Dicky Sentosa Gozali

2 RL-PTDT with HFS Live Smoothening.

3 Input Data

1. GTFS Static
2. GTFS-Realtime Vehicle Position
3. GTFS-Realtime Trip Updates
4. High Frequency Bus Shapefile

There are three main components of this webApp: - Data Renderer: this is to fetch data from web source, produce the static preprocessed data for PT trip details and run the data streamer. - Data Server: the 'backbone' of the system that accepts the processed and rendered data to run the static information into the visualizer. - Data Visualizer: the UI of the webApp allowing the real-time display of the public transport at different timestamp.

3.1 Notes

1. Download RabbitMQ with admin access on your local computer.
    1. Follow the RabbitMQ Installation Setup Documentation available.
2. Use the package manage pip to install necessary packages/

```
pip install -r requirements.txt
```

3. Your device is connected to the hetrogenmodel drive to download the appropriate static files.
    1. Alternatively, ensure that you have fetched all the necessary static files locally.

3.2 Processes

1. Run the python vehicles information fetcher to compute the daily appropriate static files.
    1. Execute/ run the VP_Fetcher
        1. HFS Route Stop Details
        2. Shapes4 Combine
    2. Restart Kernel and Run all codes again to fetch the realtime data. `python` `python ./data/gtfs_vp_3Nov23.py`
2. Run the Flask Server (iff the static files are generated properly)
    1. Open any Web Browser or Click on the address link shown in the Flask Server
        1. `python` `python app.py`
3. Run the python realtime trip information fetcher to compute the daily trip updates.

```
python ./data/gtfs_tu_3Nov23.py
```

4 Expected Output

1. Daily HFS Route Stop file
    - This file is a static file of planned scheduled generated everyday providing details of trips, stops, vehicle paths of the day.
2. Daily shapes4 File
    - This file is a static file of planned path/ trajectory of available scheduled generated everyday providing details of the path each scheduled trip would have taken within the day
3. Web Browser `ip-address:port-number`
    - A visualizer to display the position of public transport vehicles throughout the day.

5 Future Directions

5.1 Low Frequency Modes of Public Transport

In the future endeavors, one should concentrate on expanding to an even broader range of other modes for allowing better visualization. This could be initiated by computing the low frequency public transport.

5.2 Advanced Graph Interpolation Computation

Applying advanced graph network interpolation to remove inaccurate data computation on the interpolation process due to available data infidelity.

5.3 Expansion of the LiveMap with private vehicle analysis across large-scale road network

Incorporating other data sources of private vehicles to train the AI Model. This could include private vehicles from Bluetooth MAC Scanners integration or satellite vehicles data over time. Such diverse datasets would significantly improve the model's accuracy and applicability.

5.4 Spatio-temporal Transit On-time Prediction Modelling

The application of spatio-temporal prediction modelling that would feed in from the measurements computed from the visualizer. This would allow a more readable and understandable data processed from the physical entity.

In summary, these future directions encompass the integration of diverse data sources, improvement in computational algorithm and introduction of graph prediction modelling to predict the future traffic states.