Konwert India Motors Internship Report

By: Shreyashri Biswas
Position: AI/ML Intern

Note: This document entails the details for the following:

- LTE to server data pipeline
- LSTM model for predictive maintenance
- Parameters for anomaly detection

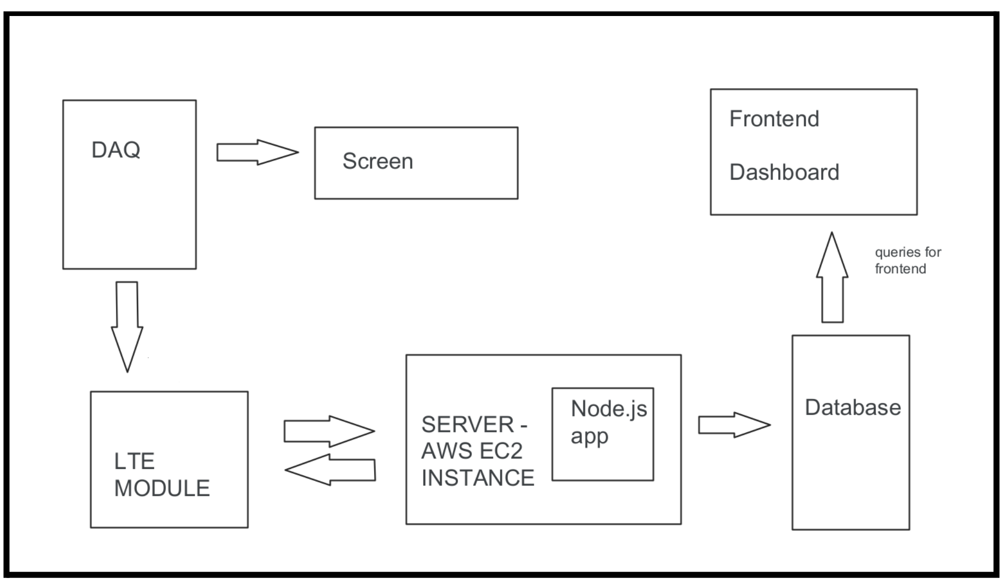
---

1. LTE to server data pipeline

Components/Tech stack used

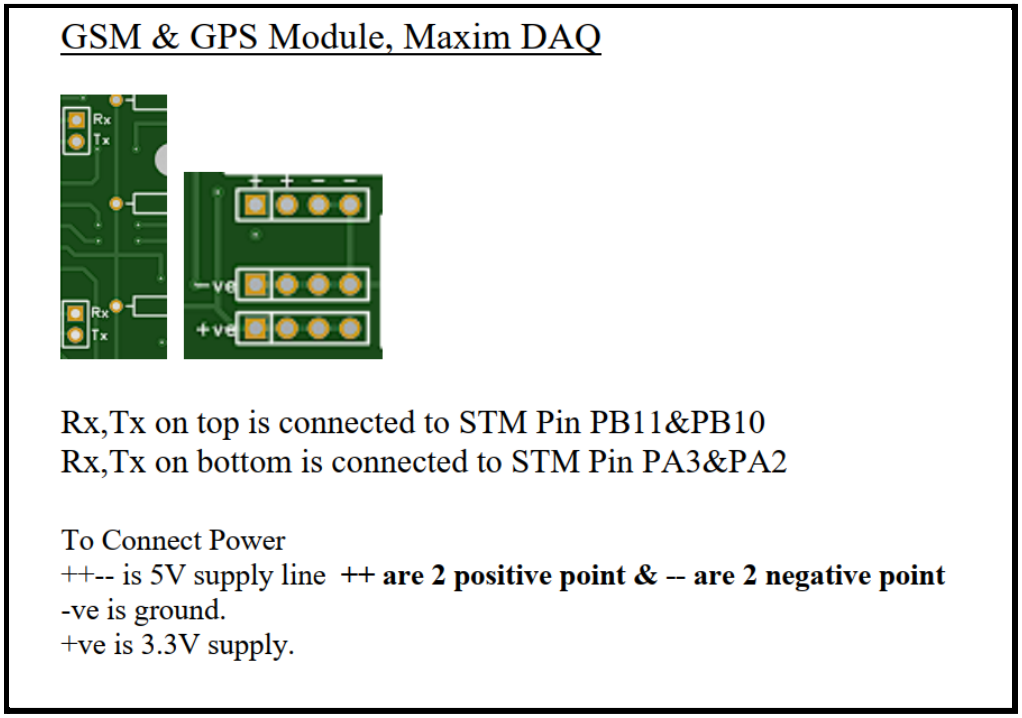
- SIM7600 LTE module
  - https://www.rhydolabz.com/wiki/?p=18634
- AWS Web services
  - Used EC2 instance to host the node.js application
- For Backend: Node.js, Express.js, MongoDB
- For Frontend: React.js

System Architecture



The figure here shows the data flow from the DAQ device(Data Acquisition Device) to the LTE module through the server which in our tech stack is AWS EC2 instance.

Key points for setting the pipeline

- Connecting the DAQ with the LTE module
    - The DAQ is connected to the LTE module via UART ports.
    - Connect the LTE module to the DAQ in the following manner
    - 

      GSM & GPS Module, Maxim DAQ

      Rx,Tx on top is connected to STM Pin PB11&PB10
      Rx,Tx on bottom is connected to STM Pin PA3&PA2

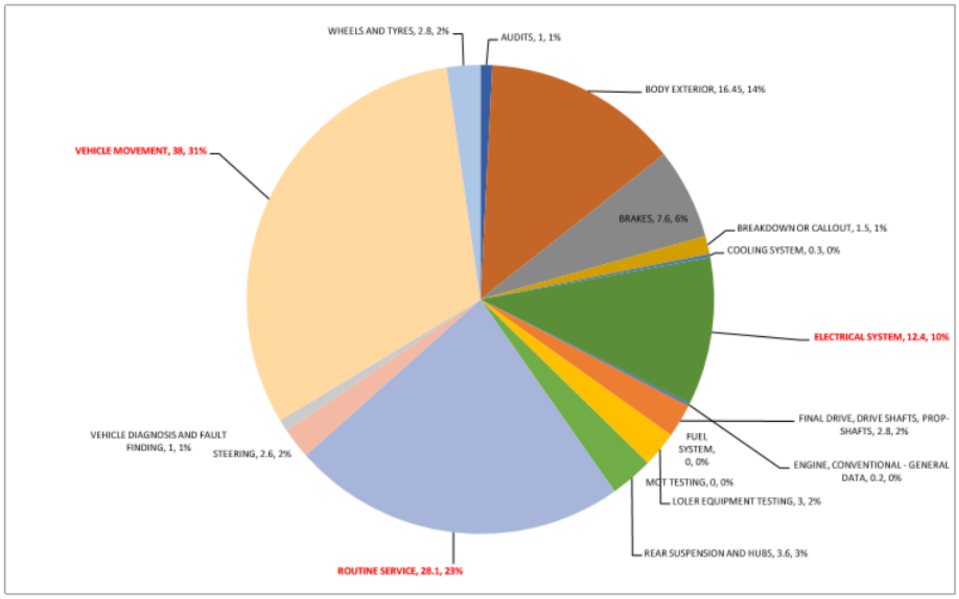
      To Connect Power
      ++-- is 5V supply line **++ are 2 positive point & -- are 2 negative point**
      -ve is ground.
      +ve is 3.3V supply.
    - After the DAQ and LTE modules are connected, we need to deploy the node.js app
- Deploying the node app
    - The entire node app is delivered on the same github repo this document is hosted on.
    - Follow Amazon's documentation on how to create an instance on an EC2 service
    - Once the instance is created, connect with the instance.
    - Once connected, using "git clone", clone this repository
    - Go to the folder with app.js
    - Install node modules along with all the dependencies with this command
        - npm install node http-errors express path cookie-parser morgan
    - Once installed, deploy the app with
        - Npm start
        - Install nodemon if in active development
    - Once it's deployed, you can test the app by going to the open DNS link given on the AWS instance window.

    - Use the DNS link or the Public ip and run in the address bar of your browser
        - Make sure not to put http or https before the ip or the dns link in the address bar
    - A html web page saying “Welcome ”will open up if everything is deployed properly
    - There are 3 http endpoint setup which can be used for testing
        - / - GET
        - /test - GET
        - /post - POST
        - Use POSTMAN for testing the endpoints if in active development
- Testing the pipeline
    - .ino files in this repo were written for Arduino UNO. Same can be used for STM32
    - There are two .ino files
        - one uses the library “tinygsm” for sending the http request
        - The other file uses AT commands for finding the GPS location
    - Change the server name and the port number (3000) in the tinyGSM.ino file
    - Run the tinyGSM.ino file on the microcontroller
    - Open the serial monitor to check the working of the code
    - If the pipeline is properly set, the http endpoint will be show up on the aws instance cmd prompt

2. Parameters for anomaly detection

Parameter vehicle

a. Health Factor: The results are achieved as probability percentage which translates to a time in the future when vehicles have to come up for servicing.
    i. Service required (boolean) Y?N?
    ii. which service: show the reason
    iii. classify as: immediate risk(60%>), short term risk(40-60%), long term risk(<40%S)
    iv. Pie chart for all the services



v.

b. Fuel Factor: how is the fuel consumption
c. The base variables: vehicle registration number, registration date, number of garage visits of the vehicle, vehicle repair history, odometer reading, repair type of the vehicle and vehicle repair types count.
d. The derived variables: age of the vehicle, count of occurrence of each service type, average labor time and average cost of spare parts.
e. The base and derived variables that were inputs to the model to arrive at predictions constituted the predictor variables

Parameters Battery - with respect to the MAXIM IC

f. Remaining Capacity (05h)
g. SOC (06h)
    i. graph req
h. Full capacity(10h)
i. SOH (Age)
    i. regression
    ii. graph req
j. RUL
k. Voltage
    i. VCell Register (09h) In multi-cell application, VCell register reports the 2.5X the voltage measured at the Cellx pin. This represents the per cell voltage of the battery pack. In single-cell application, VCell register reports the voltage measured between BATT and GND
l. Temperature
    i. Temp Register (08h)
m. Current
    i. Current Register (0Ah)
n. Critical Condition

    i. manual thresholds
    ii. give interrupt on ALRT pin.
    iii. An external pullup is required to generate a logic-high signal. Alerts can be triggered by any of the following conditions:
        1. • Battery removal: (V TH > V BATT – V DET) and battery removal detection enabled (Ber = 1).
        2. • Battery insertion: (V TH < V BATT – V DET-HYS) and battery insertion detection enabled (Bei = 1).
        3. • Over/undervoltage: VAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).
        4. • Over/undertemperature: TAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).
        5. • Over/undercurrent: IAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).
        6. • Over/under SOC: SAlrtTr register threshold violation (upper or lower) and alerts enabled (Aen = 1).
        7. • 1% SOC change: RepSOC register bit d8 (1% bit) changed (dSOCen = 1)

o. time to empty (11h)
p. time to full (20h)
q. cycle of charge and discharge (17h)
r. Status (00h)
    i. all flags related to alert thresholds and battery insertion or removal.
    ii.

Table 7. Status (00h) Format

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Br	Smx	Tmx	Vmx	Bi	Smn	Tmn	Vmn	dSOCi	Imx	X	X	Bst	Imn	POR	X

    iii. All flags
        1. POR (Power-On Reset)
        2. Imn and Imx (Minimum/Maximum Current-Alert Threshold Exceeded)
        3. Vmn and Vmx (Minimum/Maximum Voltage-Alert Threshold Exceeded)
        4. Tmn and Tmx (Minimum/Maximum Temperature-Alert Threshold Exceeded)
        5. Smn and Smx (Minimum/Maximum SOC-Alert Threshold Exceeded)
        6. Bst (Battery Status): Useful when the IC is used in a host-side application. This bit is set to 0 when a battery is present in the system, and set to 1 when the battery is absent. Bst is set to 0 at power-up.
        7. dSOCi (State-of-Charge 1% Change Alert): This is set to 1 whenever the RepSOC register crosses an integer percentage boundary such as 50.0%, 51.0%, etc. Must be cleared by host software. dSOCi is set to 1 at power-up.
        8. Bi (Battery Insertion)

9. Br (Battery Removal)
10. X (Don't Care): basically undefined error

Predictive Maintenance Prediction Hypothesis

s. The higher number of visits to the garage indicates higher likelihood of repeat visits
t. The higher the age of the vehicle indicates greater probability of needing maintenance
u. The higher odometer reading in vehicle indicates greater wear and tear probability and hence need for maintenance
v. The greater number of occurrences of repair type (work area description) for specific vehicle indicates higher probability of the recurrence
w. The higher average labour cost over 1 year suggests vehicle involves high cost of maintenance and will need frequent servicing
x. If the number of tasks performed in the last job is high the vehicle potentially has many issues and is more likely to have faults
y. The higher repair parts cost indicate there is a higher wear and tear on the vehicle and hence likelihood of other parts failing
z. If there is high labor time in the last Job there could be larger problem in the vehicle and hence probability of vehicle needing further service is higher

3. LSTM model for predictive maintenance

- Long Short Term Memory (LSTM) networks are especially appealing to the predictive maintenance domain due to the fact that they are very good at learning from sequences.
- This LSTM network predicts the remaining useful life of aircraft engines.
- The database was taken from the nasa website.
    - https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository
    - Dataset no. 17 : Turbofan Engine Degradation Simulation Data Set-2
- The same outline for model training could be used once the dataset is created from the DAQ sensor
- **IMPORTANT: Run anomaly detection using the 2nd section of this document before running the ML model in order to make a better dataset**
- The trained ML model is in the Repo as well.