ARAI Journal of Mobility Technology
Volume 2 • Issue 2 • April–June, 2022, pp. 192-197
https://doi.org/10.37285/ajmt.1.2.4

Research Paper

Deep Learning Model for Prediction of Air Mass Deviation Faults

Karthik Chinnapolamada

RDI/CEP, Mercedes Benz Research and Development Private Limited, Embassy Crest Phase-2, Whitefield Road, Bangalore-560066, India

ABSTRACT

Major Systems of an internal combustion Engine are Air System, Fuel system, Exhaust system. Any malfunction in these systems increases emissions. OBD legislation mandates to monitor these systems for any faults and appropriate action should be taken in case of the any faults which increases vehicle emissions.

The idea of the paper is to find the Air mass flow deviation faults using datamining and machine learning based approach. Detection of fault is classifying whether system is faulty or not. Objective is to create a deep learning model using the available vehicle data to classify the system for a fault.

Three main inputs for the Air Mass flow in an internal combustion Engine are

1) Fresh Air which measure using Mass Air Flow sensor
2) Low Pressure EGR
3) High Pressure EGR

During vehicle lifetime, due to different real vehicle operating conditions and environmental conditions, deviation in the set point of air mass flow and actual mass flow are possible to an extent, which can affect vehicle emissions. Deviation in the Air Mass flow can be caused by intake Air mass, LP-EGR, HP-EGR. The Aim of the project is to create the deep learning model for Air Mass Flow Hi and Low faults using the available data, and associate the fault to the component in the Intake Air System.

KEYWORDS: OBD, LPEGR, HPEGR, Machine Learning, Emissions, Deep Learning, Air Mass Deviation, Internal Combustion Engine, Air Mass Flow, Hi and Low faults, Intake Air System.

Introduction

OBD refers to the On Board Diagnostics, which refers to Vehicles self-diagnosis, and reporting capability. Fulfilment of OBD-regulation is required worldwide for type approval of any vehicle and the regulations discussed in paper are based on California regulations, where the regulations are first introduced. Section 1968.2 of California code of regulations defines the OBD requirements for California. The purpose of this regulation is to reduce Motor Vehicle emissions by establishing emission standards and requirements for OBD.

- Legislative Requirements Mandates detection and reporting of Faults in Engine which increases the Emissions
- Malfunction of any components in the Intake Air System will have adverse effect on the emissions; hence detecting Air System Faults is a Mandatory requirement.
- Any deviation in the Intake Air Mass flow set point can increase the Emissions in Vehicle so OBD system shall able to detect and report any such deviation in Air Mass flow.
- Current approaches dependent on Empirical formulas for detection of Air Mass flow deviation faults.
- Idea is to apply AI and ML techniques for the prediction of Fault.

Purpose

The Objective of the Paper is to create a deep learning model, which can predict Intake Air System Mass flow deviation faults by making use of the available vehicle data

Scope of Project

- To predict Air Mass Flow deviation in the Internal combustion by making use of the existing vehicle data available
- Model shall be able to predict the cause of deviation in the Air Mass flow.

ABBREVIATIONS: LP - Low Pressure; HP - High Pressure; EGR - Exhaust Gas recirculation Valvel; OBD - On Board Diagnostics; AI - Artificial Intelligence; ML - Machine Learning

- Here Cause of Deviation refers to fault in Air Mass flow Sensor, Low pressure Exhaust gas recirculation valve and High pressure Exhaust gas recirculation valve.

Motivation

Lots of Vehicle measurements are available in the organization, the idea is to make use of the existing data to create Model which predicts the Air system Faults. This helps in making the better engine control and fault prediction and pinpointing of the Air Mass Faults more reliable.

Existing System

Current Systems depends on different physical parameters like Mass flow, Pressure, Temperatures at different points in the Engine to decide on the deviation in actual Mass flow and set point, classify whether the system has fault or not.

Proposed System

The new system should learn fault patterns from existing data and should predict the faults by analysing the input sample.

Features of the Proposed System

The System should be able to predict the below faults by analysing the input sample data

1. Air Mass Sensor High Flow
2. Air Mass Sensor Low Flow
3. Low pressure Exhaust Gas Recirculation High Flow
4. Low pressure Exhaust Gas Recirculation Low Flow
5. High pressure Exhaust Gas Recirculation High Flow
6. High pressure Exhaust Gas Recirculation Low Flow
7. No Fault

Process

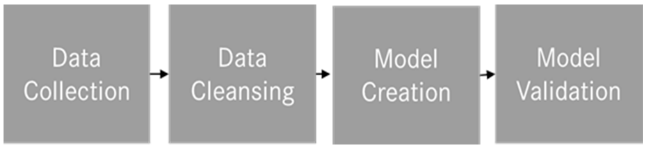


Fig. 1. Process flow.
(Created by Author to represent the Process flow in executing the project)

The whole process divided into four steps

1. Data collection – First step where all the required data for creating the model will be collected
2. Data cleansing – Raw data collected has to be processed to extract the required data and to the required format.
3. Model Creation- The processed Data has to be fed to the Machine Learning Algorithms for creating Data based Model
4. Model Testing – Testing of the Model created in Step 3

Data Collection

- Total Data collected :38.5 GB
- Data format: .mdf , .dat
- Data Source: Engine Control Unit
- Type of Data: Numerical
- Data contains recording of different Engine control signals like Engine Speed, Injection Quantity, Intake Air Mass flow, EGR rate etc.
- All these Signals are used by the Engine control unit software to control different Engine Actuators for Efficient Engine operation and to keep the Emissions in control
- Aim of the project is to identify signals which impact the Air system faults, analyze the behavior of these signals and implement the ML algorithm which gives the best results

Data Cleansing

Data cleansing or **data cleaning** is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data (Source: Wikipedia).

A training and testing dataset created for training and testing the model.

The below steps are performed as a part of Data cleansing

- **Identification of the required variables(Features) –** Feature selection is one of the key steps in the whole process of datamining and it will have a huge impact on the performance of the model. Irrelevant features can negatively affect the performance of the model. Hence feature selection has to be done carefully to achieve the better prediction results. A total of 18 features selected, features are selected based on the expert opinion in the Air system, independent of current inputs for fault detection and effect of the feature on the fault itself.

TABLE 1
Features considered for the Model creation

Feature 1	Accelerator Pedal Position
Feature 2	Charge Air Cooler Down Stream Pressure
Feature 3	Charge Air Cooler Down Stream Temperature
Feature 4	Air Mass Per Stroke
Feature 5	Actual EGR Percentage
Feature 6	Boost Pressure
Feature 7	Set point EGR Percentage
Feature 8	Exhaust Flap Position
Feature 9	EGR Position
Feature 10	Engine Speed
Feature 11	Injection Quantity
Feature 12	EGR Differential Pressure
Feature 13	Swirl Valve position Demand
Feature 14	Turbine Input Temperature
Feature 15	Turbine Input Pressure
Feature 16	VTGA Position Demand
Feature 17	Throttle Position
Feature 18	Engine Torque Request

(Created by Author to represent the features selected for Machine Learning Model)

- **Labeling each Vehicle measurement with Fault class for each fault from 0 to 6**

  Each file has been recoded with a particular fault created. Labeling of each fault with corresponding fault class is primary step. This fault class serves as target class for training the model. Labeling of each training file has to be done carefully as wrong labeling of file affects the performance of the model. All the files used for training and testing are labeled with fault class before start of the analysis. All labeled files are further separated into different train set as the processing of whole data takes huge time for the analysis.

  After labeling files, each file needs to process for the features. Vehicle recordings, which only have all the required features, considered for the training and testing dataset creation. Each file is analyzed for the required features and files which has the required features are separated and used for training the model.

Fault Class	Fault
0	No Fault
1	Air Mass Sensor Low Flow
2	Air Mass Sensor High Flow
3	Low Pressure EGR Low Flow
4	Low Pressure EGR High Flow
5	High Pressure EGR Low Flow
6	High Pressure EGR High Flow

- **Filtering Samples**

  Not every sample in the files with required inputs considered for the training as not all required enable conditions met. Hence, each data sample added to the dataset only when the fault enable conditions met.

```
Not faulty Sample (class 0):
    IF (ANY of one Fault criteria is satisfied)
        Add Sample to the Dataset;
    Else
        Discard Sample;
    End
Faulty Sample (Other than class 0):
    IF (Particular Fault criteria is satisfied)
        Add Sample to the Dataset;
    Else
        Discard Sample;
    End
```

- **Identification of steady state conditions**

  Identified the Steady state condition by evaluating the different engine parameters and sample added to data set only if the steady state conditions met. Filtered all transients out of the data set. Sample space of 30 samples are taken for calculating the Moving average of Engine Speed and Injection quantity to verify the steady state

$$n_{avg} = ((n_1 + n_2 + n_3 + \cdots \ldots + n_{30})/30)$$

  Ignore sample, which has a deviation of over 10% from moving average

```
IF $|n_{avg} - n_1| \le 10\%$ :
        Add Sample
ELSE:
        Discard Sample
```

  'n' represents Engine Speed and Injection Quantity

- **Normalization of data**

  Normalization of data is one of the important steps in the data mining. Different features are recorded on a different scale and it is very important to bring all the feature to the same scale. Normalization is required to achieve this. Min-Max normalization applied to normalize data.

  Formula:

$$x_{norm} = \left(\frac{(x) - (x_{min})}{(x_{max}) - (x_{min})}\right) . \left((x_{maxnew}) - (x_{minnew})\right) + \left((x_{minnew})\right)$$

$$x_{normperc} = x_{norm} * 100$$

  x = *sample value*

  x_{norm} = *Normalized sample value in the range* [0 1]

  $x_{normperc}$ = *Normalized sample value in the range* [0 100]

  x_{max} = *maximum value of* x *in the dataset*

  x_{min} = *minimum value of* x *in the dataset*

  x_{maxnew} = *new maximum value of* $x = 1$

  x_{minnew} = *new minimum value of* $x = 0$

  All the different features, which are in different, scale converted to the range 0 to 1 or 0 to 100% for training the model.

- **Outlier Detection**

  BOX plot drawn to detect the outlier. No sample removed from the dataset as all the data recorded in the vehicle and all scenarios represent the driving behavior.

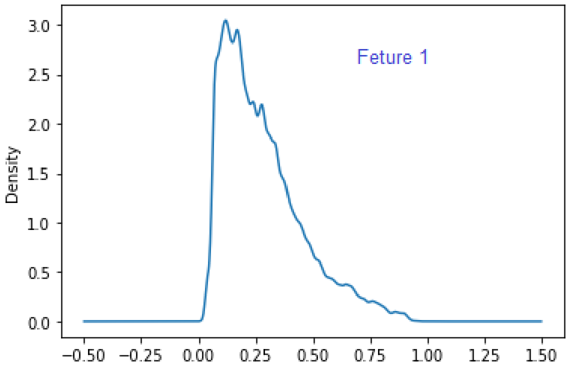
- **Density plots**

  Density plots drawn to understand the distribution of data

  Transformations Techniques are used to correct the Skewness in the plots

- **Files with double faults**

  Files which more than one fault are discarded for the creation of dataset



Before Transformation

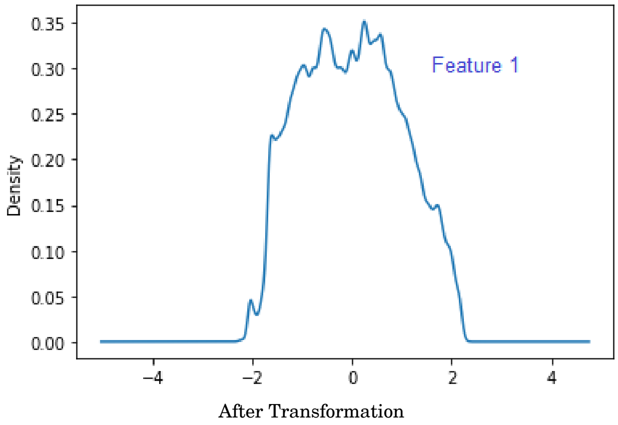


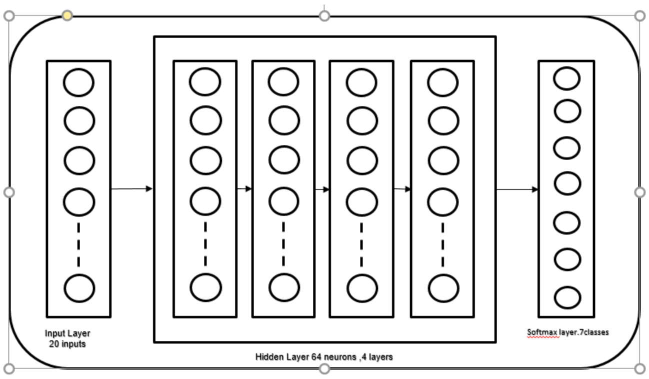
Fig. 2. Feature Transformation.

Model Creation

Neural network model

A multilayer perceptron comprises of multiple layers of logistic regression. Multi-layer perceptron is a feed forward neural network; Multi-layer perceptron model built fort classification of Air Mass deviation faults.

Neural Network Architecture



Network created is fully connected, for simplicity all connections are not shown

Fig. 3. Neural Network Architecture.

The above figure explains the architecture of the neural network.

Number of Inputs: twenty

Number Hidden layers: four

Number of neurons in each layer: sixty-four

Output Layer: Softmax

Number of output classes: seven

Note: In The architecture shown above, not all the connections shown for the sake of simplicity, but neural network is fully connected.

Soft Max function:

Softmax function calculates the probabilities distribution of the event over 'n' different events. This function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

$$\partial(z_i) = e^{z_i} / \sum_{j=1}^{K} e^{z_i}$$

Soft max function used at the output layer for the Multi class classification.

Neural Network Model –Step 1

- Created Fully connected Neural Network
- 32 neurons in each layer and 3 layers
- Tanh function is used as the activation function
- Number of neurons and layers decided based on trial and error basis
- Soft max layer at the output
- Trained network with 300 epochs with a batch size of 30000
- Trained the Model with the same dataset as the one used for the Decision Tree.
- No sampling techniques has been employed to capture the time dependence between the samples.
- A separate a set of files from the data are considered for the testing the model. The same data cleansing steps are performed on the test data files to create the test data set.

Results:

- Efficiency with the Test data is close to ~50%

Reason:

- Data is highly imbalanced and hence there is chance over fitting. So the model did not really performed well on the test data

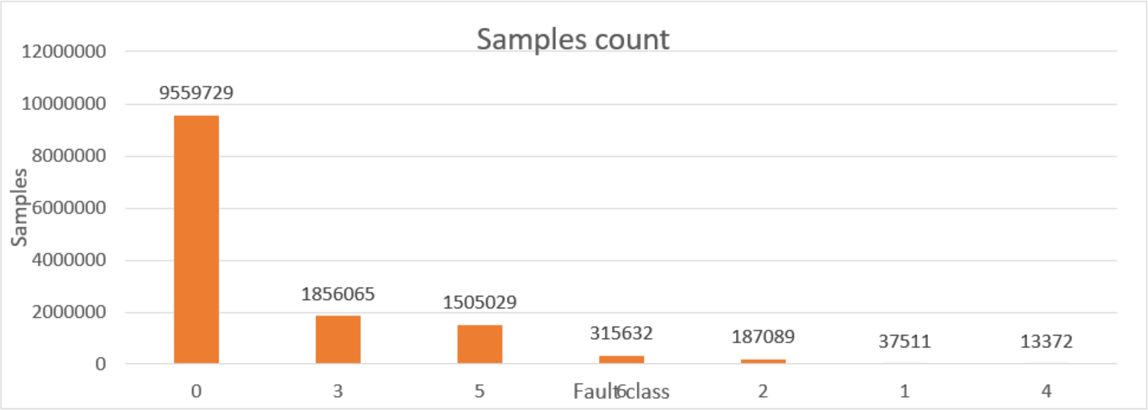


Fig. 4. Graph Representing Imbalance Dataset.

Neural Network Model –Step 2

Handling of Imbalance data

Class weights: It is clear from the figure 7, there is huge imbalance in the data. Class one and class 4-sample count are very less in comparison to the class 0 sample count. This can lead to over fitting; one of ways to increase the prominence for minority samples is adjust class weights of the minority classes. Class weights adjusted as per the sample count, more the number of samples less weight for the class.

TABLE 2

class Weights

Class	Weight
0	0.02
1	0.25
2	0.15
3	0.06
4	0.3
5	0.1
6	0.12

Drop out of Neurons: After each layer of the neural network, 15% of the neurons are dropped out to avoid the over fitting of the model.

Number 15% is arrived by trial and error basis. Tried with different values between 5% and 25% and achieved better efficiency with 15% drop out of neurons.

Model Creation

Below three are the major changes performed in step 2 of neural network model creation

1. Adjusting class weights to handle imbalance data
2. Considering Fault criteria for the creation of data set. Followed same steps as mentioned in Data Cleansing
3. Drop out of neurons after the each layer to avoid over fitting of the Model

Test Data details

A separate a set of files from the original data considered for the testing the model. The same data cleansing steps performed on the test data files to create the test data set.

Neural Network Model details:

- Checked for the Fault Enable condition for each sample of the dataset
- For No Fault condition, sample is considered if at least one fault is enabled
- For Faulty sample, sample is added to dataset only when the specific fault is enabled
- Created Fully connected Neural Network
- 64 neurons in each layer and 4 layers
- 15% neurons dropped out after each layer to avoid overfitting
- Number of neurons and layers decided based on trial and error
- Soft max layer at the output
- Trained network with 300 epochs with a batch size of 30000
- Results:
- Efficiency with the Test data is close to ~61%
- Reason:
- Multiple files for the same driving pattern is considered.

With neural network model is step 2, efficiency of the model is not good and still model needs improvement. While analyzing the reason, found that multiple files for same scenarios are considered and tests covering different driving patterns needs to consider for the creation of data set. Hence decided further to analyze the impact different files considered for the dataset on the model creation.

Neural Network Model –Step 3

The whole data considered for creation of model predominantly has three kinds of files

1. Low ambient temperature recording
2. Recorded with Normal drive behavior
3. Standard driving cycles

Model Creation

Major change in step 3 is to consider the Standard drive cycle data for the creation of dataset.

Size of Data: 15 GB

Size of Data set after data cleansing 1.05GB

Test Data Details

A separate a set of file from the data are considered for the testing the model. Performed the same data cleansing steps test data files.

Created a test data of 500MB for testing the model

Model Details

Most of details are same

- Created Fully connected Neural Network
- 64 neurons in each layer and 4 layers
- Number of neurons and layers decided based on trial and error
- Soft max layer at the output
- Trained network with 300 epochs with a batch size of 30000
- Created a test data of 500mb
- Total number of test Samples:1906324

Results:

- Efficiency with the Test data is to ~92.55%

Plot accuracy versus Loss:

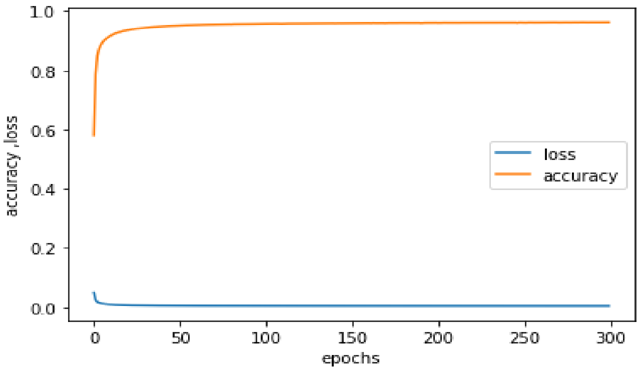
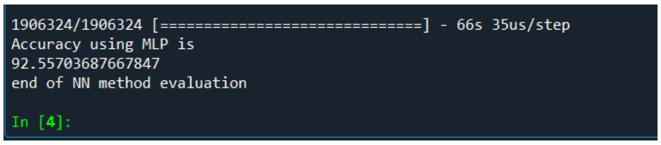


Fig. 5. Accuracy Loss versus Epochs.

Results on Test Data:



```
1906324/1906324 [==============================] - 66s 35us/step
Accuracy using MLP is
92.55703687667847
end of NN method evaluation

In [4]:
```

Fig. 6. Output.

Acknowledgements

My sincere thanks and grateful acknowledgement to **Mr. Tobias Erbes Manager, System & Technology HV Batteries, Mercedes-Benz AG** for his support in data collection, valuable suggestions and constant encouragement.

I take this opportunity to express my sincere thanks with deep indebtedness to my supervisor **Mr. Atul Gupta, Senior Program Manager, Mercedes Benz Research and Development India Private Ltd.** for his Valuable suggestions

Conclusion

Multi-Layer Perceptron Model for classifying Air Mass flow deviation in the Air System created successfully by making use of the available vehicle teste data and validated. An efficiency of 92.55% achieved with the current model. Further, Multi-Layer Perceptron Model can be improved by analyzing the reasons for the incorrect predictions by the model and model needs to be validated by collecting the test date from different vehicle operating conditions, different intensities of the fault level and across different engines.

References

[1] Tom M. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc. International Edition. 1997. http://www.cs.cmu.edu/~tom/mlbook.html

[2] Christopher M. Bhisop, Pattern Recognition & Machine Learning, Springer, 2006. ISBN: 978-1-4939-3843-8. https://link.springer.com/book/9780387310732

[3] Julie Main, Tharam Dillon and Simon Shiu: A Tutorial on Case-Based Reasoning. https://dl.acm.org/doi/10.5555/357410.357787

Address correspondence to: *Mr. Karthik Chinnapolamada, Sr. Tech Lead, RDI/CEP, Mercedes Benz Research and Development Private Limited, Embassy Crest Phase-2, Whitefield Road, Bangalore*-560066, *India.*
Phone: 09901028983
E-mail: karthik.chinnapolamada@daimler.com