**Detection And Classification Of Incipient Faults In Automotive Using Machine Learning Approach**

Machine Learning Course Project

****

**By :Peculiar\_Panch**

**Binny Mathew**

**Jay Thakkar**

**Md Fraz Ansari**

**Bhushan Naware**

**Contents:**

1. **Abstract**
2. **Problem Statement**
3. **Methodology**
4. **Our Contribution**
5. **Results**
6. **Conclusions**
7. **References**

**Abstract:**

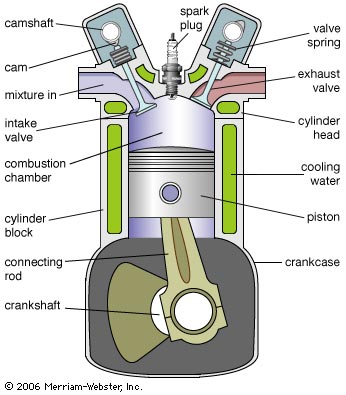
Integrated vehicle health management (IVHM) or integrated system health management (ISHM) is the unified capability of systems to assess the current or future state of the member system health and integrate that picture of system health within a framework of available resources and operational demand. IVHM helps in better management of vehicle and vehicle fleet health. This is achieved through correct use of reliable sensing and prognosis systems to monitor part health and also using usage data to assist in understanding the load experienced and likely future vehicle load. Detection of early and low magnitude faults in automotive engines requires dynamic engine models which can capture within-cycle continuous switching dynamics of engine under different input and state conditions. In this project we tried to predict the state of engine error using machine learning concepts. We are targeting here the training of data and interpretation of faults on the earlier work done and trying to explore the machine learning opportunities in the context of fault detection.

**Problem Statement:**

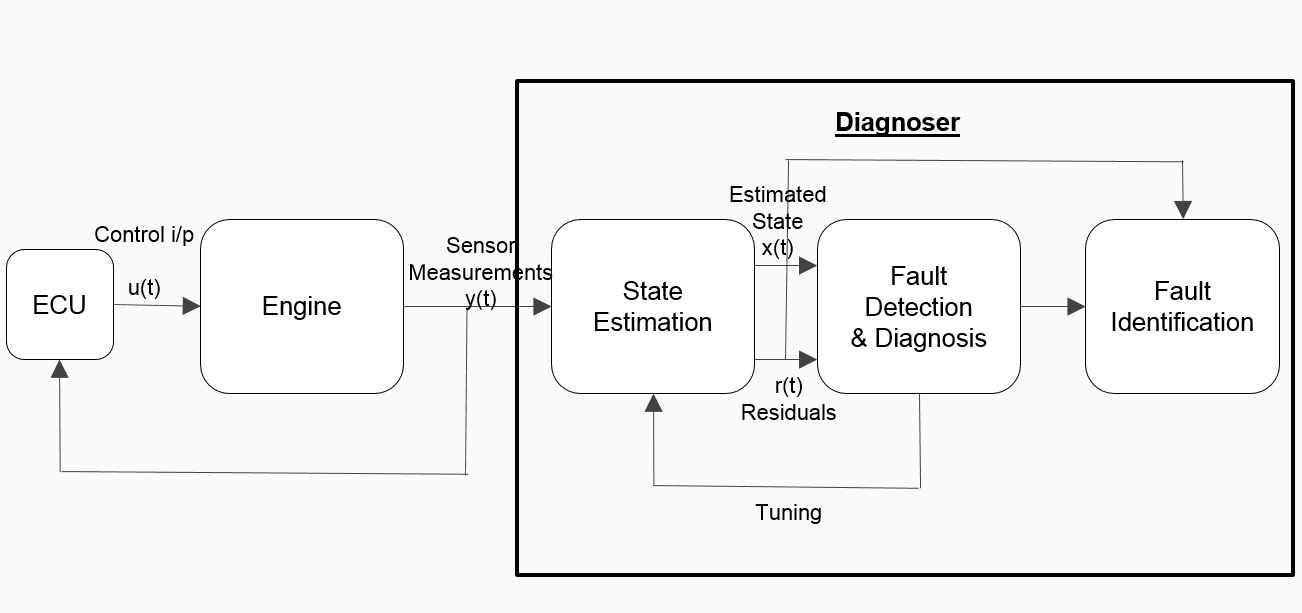
Online fault diagnosis of automotive engines is a crucial part of integrated vehicle  
health management (IVHM) and associated prognosis strategies, which are being employed  
in increasingly many automotive vehicles at present times. Detection of early and low  
magnitude faults in automotive engines requires dynamic engine models which can capture  
within-cycle continuous switching dynamics of engine under different input and state  
conditions. However, because implementations of such accurate model based diagnosis  
schemes require huge computational capacity usually not found on engine electronic control  
units (ECUs), a trade-off is sought between model complexity and performance of diagnosis  
schemes. We are targeting here the training of data and interpretation of faults on the earlier work done and trying to explore the machine learning opportunities in the context of fault detection.

**Methodology:**

**Understanding the existing model.**



**Figure: Spark Ignition 4-stroke Engine Components**



**Figure: Introduction to the work, block diagram**

Figure shows an online fault diagnosis scheme in which WCCM could be employed.  
It could be used inside nonlinear estimators/observers for accurate estimation of engine  
variables that are usually not measured, such as the pressures, temperatures and masses of  
individual gas species in cylinders. These variables contain fault signatures needed for fault  
detection and isolation. Being a diagnosis-oriented model, the WCCM has the advantage of  
accessing control signals coming out of ECU, and can improve the diagnosis functionality.  
Further, such a model when implemented online could help the ECU in implementing fault  
tolerant control schemes and integrated vehicle health management (IVHM). Note that even  
for fault tolerant control, unlike control-oriented models, diagnosis models need not have to  
be hard real time, allowing for more complexity to be added to the model for better  
diagnosability.

**Identifying our contribution:**

Existing work of the NP-MASS lab takes the measurement residuals as an input and computes likelihood function for each non faulty and faulty states; and identifies the state of the system. We try to enhance the work, by replacing identification logic with Machine learning algorithm. The input is measurement residual for ML algorithm and the output is classification of state. We trained the system using naïve Bayes classifier, Random Forest and J48.

Since number of features were much more than the observations available we used random forest algorithm for the classification task.

**Results:**

Results of training data using Random Forest, Naïve Bayes, J48 algorithm:

***Random Forest:***

Instances: 1400

Attributes: 8524

=== Classifier model (full training set) ===

=== Summary ===

Correctly Classified Instances 581 96.8333 %

Incorrectly Classified Instances 19 3.1667 %

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.980 0.013 0.961 0.980 0.970 0.960 0.999 0.998 0

0.967 0.013 0.960 0.967 0.963 0.951 0.995 0.989 1

0.947 0.013 0.959 0.947 0.953 0.938 0.995 0.988 2

0.980 0.002 0.993 0.980 0.987 0.982 1.000 0.999 3

Weighted Avg. 0.968 0.011 0.968 0.968 0.968 0.958 0.997 0.993

***Naive Bayes Classifier:***

=== Summary ===

Correctly Classified Instances 1608 80.4 %

Incorrectly Classified Instances 392 19.6 %

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.834 0.063 0.814 0.834 0.824 0.765 0.903 0.744 0

0.816 0.083 0.767 0.816 0.791 0.719 0.896 0.709 1

0.738 0.086 0.741 0.738 0.739 0.653 0.849 0.652 2

0.828 0.029 0.904 0.828 0.864 0.823 0.907 0.806 3

Weighted Avg. 0.804 0.065 0.807 0.804 0.805 0.740 0.889 0.727

***J48 TREE:***

=== Summary ===

Correctly Classified Instances 523 87.1667 %

Incorrectly Classified Instances 77 12.8333 %

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.940 0.018 0.946 0.940 0.943 0.924 0.965 0.908 0

0.720 0.027 0.900 0.720 0.800 0.751 0.929 0.810 1

0.927 0.113 0.732 0.927 0.818 0.757 0.926 0.718 2

0.900 0.013 0.957 0.900 0.928 0.906 0.955 0.917 3

Weighted Avg. 0.872 0.043 0.884 0.872 0.872 0.834 0.944 0.838

**Sample Dataset**

-6.0264586e-02 2.3597935e+03 -2.3887552e+04

-1.2360102e-02 2.3467388e+03 -2.3843801e+04

-1.0681965e-02 2.0565110e+03 -2.4058397e+04

-6.8862581e-02 1.7233474e+03 -2.2834164e+04

-3.6298998e-02 2.0853097e+03 -2.1739735e+04

**Conclusions:**

We are able to apply various algorithm to train our system that predicts and classifies faults with very high accuracy. In this project we have used residual data to train the system, this work can further be extended to use raw data directly from the various sensors.

**References:**

[1] Nadeer E P, Amit Patra, and Siddhartha Mukhopadhyay, Hybrid State Space Modelling of an SI Engine for Online Fault Diagnosis.