Relation Discovery for Diagnostics Using Machine Learning Technique

Karthika P [1], Dr.P. Prabhusundhar [2]
[1] Mrs.*Karthika P, Research Scholar, Gobi Arts & Science College, Gobichettipalayam*
[2] *Dr.P.Prabhusundhar, Assistant Professor, Gobi Arts & Science College, Gobichettipalayam*

***Abstract*—Warranty prediction is a most important task in reliability engineering. It needs to estimate the expected number of product failures in any given time period during the length of the warranty contract, Since the sensor capabilities and engineering effort available for diagnostic purposes is limited. It is, in practice, impossible to develop diagnostic algorithms capable of detecting many different kinds of faults that would be applicable to a wide range of product configurations and usage patterns. However, it is now becoming feasible to obtain and analyse on board data on products as they are being used. It makes automatic data-mining methods an attractive alternative, since they are capable of adapting themselves to specific product configurations and usage. In order to be though, useful, such methods need to be able to detect interesting relations between huge number of available signal. This method unsupervised for discovering useful relations between measured signals in a product, both during normal operations and when a fault has occurred. The interesting relationships are found in a two-step procedure. In the first step, identify a set of "good" models, by establishing a mean square error threshold over the complete data set. In the second step, estimate model parameters over time, in order to capture the dynamic behaviour of the system. use two different approaches here, the Least Absolute Shrinkage and Selection Operator method and the Recursive Least Squares filter. The usefulness of obtained relations is then evaluated using supervised learning to separate different classes of faults.**

***Index Terms*— Fault detection, diagnostics, Machine learning, Signal Processing, Algorithms and Reliability.**

I. INTRODUCTION

Nowadays, the manufacturing industry faces significant transformations. Due to the rapid growth of the digital world and the broad application of data science, different human activity fields are pursuing improvement. Modern manufacturing is also referred to as Industry, manufacturing under the conditions of the fourth industrial revolution that resulted in data robotization, automation, and widespread use. Every day, the amount of data to be stored and processed is increasing. Today's manufacturing companies, therefore, need to find new solutions and use cases for this knowledge. Data, of course, brings its advantages to manufacturing as it helps them automate large - scale processes and speed up implementation time. Mechatronic systems of today are typically associated with a high software and system complexity, making it a challenging task to both develop and, especially, maintain those systems. The maintenance strategy is typically reactive, meaning that a fault is fixed only after it has occurred. In the industry it is difficult to move towards predictive maintenance because of limited budget for on-board sensors and the amount of engineering time it takes to develop algorithms that can handle several different kinds of faults, but also work in a wide range of product configurations and for many different types of operation under varying environment conditions.

The current trend of increasing number of components in electronic continues, then the only solution will be to move towards automated data analysis to cope with increasing costs. At the same time, with the introduction of low-cost wireless communication, it is now possible to do data mining on-board real electronic products as they are being used. An approach that allows discovery of relations between various signals that available on the internal electronic network. It is based on the assumption that while it is difficult to detect faults by looking at signals in isolation, the interrelations of connected signals are most likely to be indicative of abnormal conditions. One requirement for our approach is to be able to perform relation discovery in a fully unsupervised

way. This is important since, while an engineer may be able to predict a large number of "good" relations between various signals. It is not feasible to develop specialized diagnostic algorithms for each of those cases, unless fault detection is done in an automatic way.

II RELATED RESEARCH

Automated data mining for electronic applications has previously been the topic of several papers.

Data Mining is a process of automatic searching large data set to discover patterns and trends; those are used for sophisticated analysis after applying statistical and mathematical algorithms to determine meaningful data and the prognostication of future events and trends. *(Ashish Kumar and Kavita Choudhary (2014)*.

Methods for equipment monitoring are traditionally constructed from specific sensors and/or knowledge collected prior to implementation on the equipment. *(Byttner, ¨ognvaldsson, and Svensson (2011)).*

The data-based fault detection and isolation (DBFDI) process becomes more potentially challenging if the faulty component of the system causes partial loss of data. *(Silva (2008).*

III. DESCRIPTION OF DATA

Analysed a total of fourteen driving runs, four of which were performed under normal operating conditions and the other ten with various faults introduced. The truck was equipped with a logging system which collected data from the internal product signal network as well as from a set of extra sensors. In total, twenty-one signals were recorded with a sampling frequency of 1 Hz. Each driving run was approximately four hours in length, under a variety of driving conditions. Specifically targeted the air - intake system, since it is prone to wear and needs regular maintenance during the lifetime of a product. Four different faults have been injected into the product. The first two were clogging of air filter and grill. air filter change and grill cleaning are routine maintenance operations that are performed regularly. The third fault was charge air cooler leak. Such leaks occur in the joints between charge air cooler and connecting air pipes and are considered faults that are hard to find manually. Finally, exhaust pipe was partially congested, which is a rather uncommon fault

IV. RELATION DISCOVERY

The method use for discovering relations consists of three steps. Start with data preprocessing & remove influence of ambient conditions. Then, proceed to choose the most interesting signals to model, as well as which signals should be used to model them. Finally, estimate model parameters. In this last step uses two types of approaches, the Least Absolute Shrinkage and Selection Operator method and Recursive Least Square Recursive Least Squares method. In this work using mean square error as the main criterion for determining which relations are interesting.

A. Preprocessing

First, all signals are performed normalization and removed obvious outliers. Then, available signals were divided into system and ambient signals. In order to increase the signal to noise ratio of signals, begin by filtering out the effects of ambient conditions on the system, using a procedure introduced fig 5. Namely, attempt to model each system signal y_k as a linear combination of all ambient signals:

$$\Theta_k = \underset{\Theta \in \mathbb{R}^a}{\arg\min} \left(\sum_{t=1}^{n} \left(y_k(t) - \Theta^\top \varphi_{amb}(t) \right)^2 \right) \qquad (1)$$

$$y_k^{norm}(t) = y_k(t) - \Theta_k^\top \varphi_{amb}(t) \qquad (2)$$

where α is number of ambient signals, $y_k(t)$ is the value of a system signal $_k$ at time t, θ_k is a vector of parameter estimates for the model of y_k and φ_k is the regressor for the model of y_k. Intuitively, y_k^{norm} is this part of signal y_k that cannot be explained by external conditions.

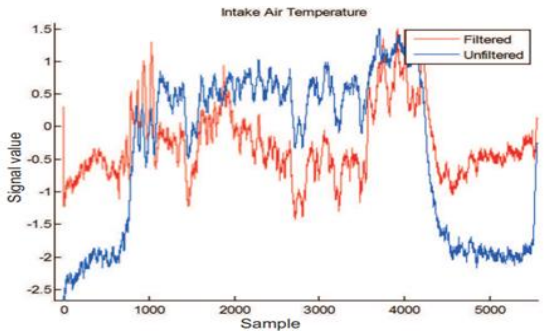


Fig 1: Signal normalization

Fig 1 above illustrates how the intake air temperature is affected by the ambient conditions. After ambient filtering, the signal has significantly less variance.

B. Signal selection

The next step is to find out which relations between signals are interesting. In the first stage attempt to model each system signal using all other systems signals as regressors:

$$\Psi_k = \arg\min_{\Psi \in \mathbb{R}^{s-1}} \left(\sum_{t=1}^{n} \left(y_k(t) - \Psi^{\top} \varphi_k(t) \right)^2 \right), \sum_{i=0}^{s-1} \|\Psi_{k,i}\| < C_k \quad (3)$$

where s is number of system signals, Ψ_k is a vector of parameter estimates for the model of y_k and ϕ_k is the regressor for model of y_k. The Least Absolute Shrinkage and Selection Operator constraint C_k provides an upper bound on the sum of absolute values of all parameters for $y_{k..}$ Gradually increase its value, performing a cross-validation test after each run. Initially, the mean squared error of the model keeps decreasing, but at some point, it begins to increase, as seen in fig 2. The Least Absolute Shrinkage and Selection Operator constraint forces small parameters representing insignificant relations to go to zero and significant relations to be prioritized, resulting in models with less non '0' parameters than standard least squares approaches.

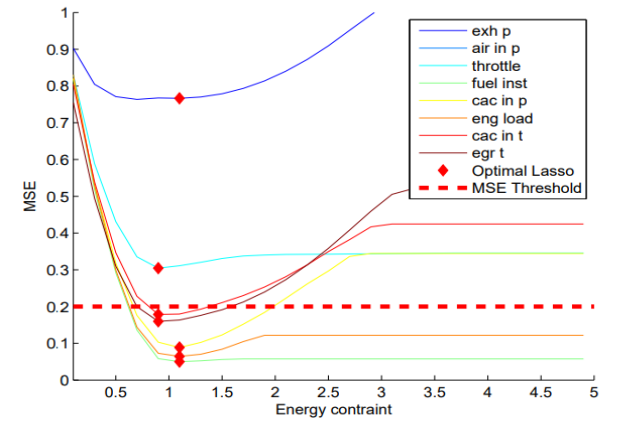


Fig 2: Least Absolute Shrinkage and Selection Operator parameter estimation

Not all system signals can be modelled in a good way. Some of them have very low correlation to any other system signal and are thus of no interest since any relationship with other system signals are insignificant. These signals are found by studying the mean square error of each model. When increase C_k, the mean square error initially decreases, until the model starts to overfit the training data and the mean square error goes up. This allows us to find a good C_k value for each signal, by repeating this procedure over a set of time slices and choosing C_k which results in the lowest average mean square error. Moreover, all system signals with the mean square error above a given threshold are disregarded.

The second stage consists of finding and removing insignificant model parameters, namely those which are unstable and with low values. To find the relations that are actually important, a sequence of estimates for each regressor within a model is collected over a series of time slices. The perform attest to find which of those estimates are significant, (example) which are non-zero. This allows us to remove artificial signal dependencies, leaving only strong relationships.

The end result of the signal selection is a set of system signals that are worthwhile to model & for each of them, a unique regression vector containing signals that should be used to model them.

B. Signal selection

To estimate parameters for the models have used two different approaches. The first one is the Least Absolute Shrinkage and Selection Operator method, as explained in previous sections. In split available data into a number of time slices, and, for each slice, calculated optimal model parameters, storing them in an array W.

This approach allows an estimator to easily adapt to changing models. On the other hand, when there are two different models that are similarly plausible, Least Absolute Shrinkage and Selection Operator estimator is likely to oscillate between them in a nearly random way.

A second method is a Recursive Least Square filter [3], which recursively calculates the estimates over a sliding window defined by the forgetting factor. It aims to minimize a weighted linear least squares cost function, thus exhibiting very fast convergence at the cost of poor tracking performance.

$$P(0) = \delta_{init}^{-1} I \qquad \Theta(0) = \Theta_{init} \tag{4}$$

$$e(n) = y(n) - \Theta^{\top}(n-1)\varphi(n) \tag{5}$$

$$g(n) = \frac{P(n-1)\varphi(n)}{\lambda + \varphi^{\top}(n)P(n-1)\varphi(n)} \tag{6}$$

$$P(n) = \lambda^{-1}P(n-1) - g(n)\varphi^{\top}(n)\lambda^{-1}P(n-1) \tag{7}$$

$$\Theta(n) = \Theta(n-1) + e(n)g(n) \tag{8}$$

The estimates from all Recursive Least Square estimators are collected into an array

$$W(t) = [\Theta_1(t) \cdot \cdot \Theta s(t)]$$

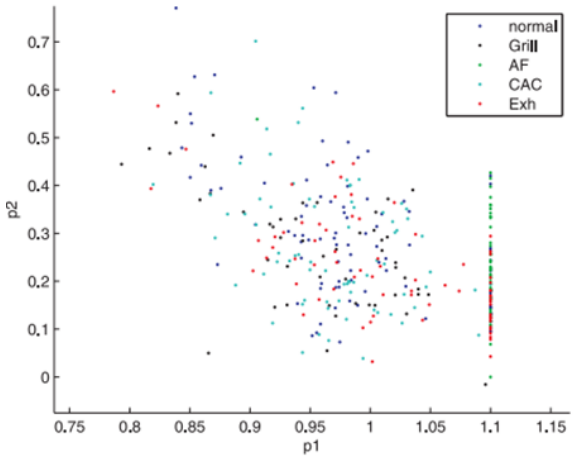
since each new sample from the system results in new, updated estimates.

Using the Least Absolute Shrinkage and Selection Operator method, obtain one set of model parameters for each time slice. With Recursive Least Square, get significantly more model parameters, but this data is more interdependent. While model parameters from Least Absolute Shrinkage and Selection Operator method are all calculated from different parts of input time series, Recursive Least Square models are all evolutions of the same predecessor. fig 3 and 4 [3] and [4] show parameters used for modelling the signal fuel inst. The X and Y axis each represents one of the model parameters (i.e. dimensions in the input space of the classifier, as explained in the following section). The dots in the figures each correspond to a single estimate from the Recursive Least Square estimator and from the Least Absolute Shrinkage and Selection Operator estimator, respectively.

As can be seen, our method has autonomously discovered that fuel inst can be approximated using cac in p and in manif (input manifold temperature). In other words, there exists a relation

"Fuel inst = p_1∗cac in p+p_2∗in manif t"

The actual values of p_1 and p_2 parameters, of course, depend on the exact values of the signals in question, but as can be seen in fig 3 and 4, they show an interesting regularities. There are some differences between the two methods, but the general pattern is the same. It appears that some of the faults can be quite easily differentiated from normal operation, but some faults are impossible to detect. This is mainly due to two reasons. Firstly, the injected charge air cooler leaks were very small in comparison to what is considered a serious problem, and secondly, there are very few sensors located sufficiently close to the fault area. In a similar fashion, fig 5 and 6 show the parameters corresponding to relations "eng load = $p_1 *$ fuel inst" and "fuel inst = $p_2 *$ cac in p". There is no direct physical correspondence between those two



parameters, but as can be seen, at least in the case of Recursive Least Square method, this relation can still be useful for detecting some faults.

Overall, though, it is rather difficult to evaluate the quality of discovered relations. Some of them make sense from the domain expert point of view in the general sense, but actual parameter values tend to vary greatly between different time slices.

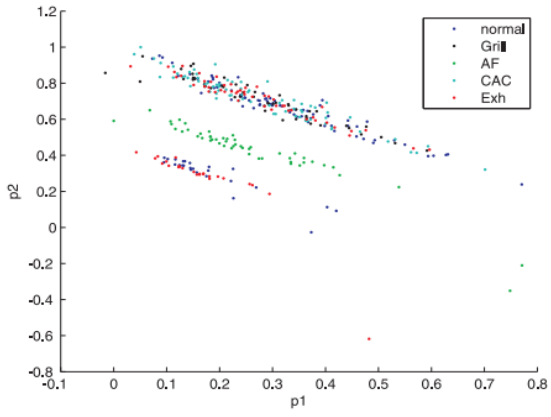


Fig 3: Model parameters

(Least Absolute Shrinkage and Selection Operator Method)

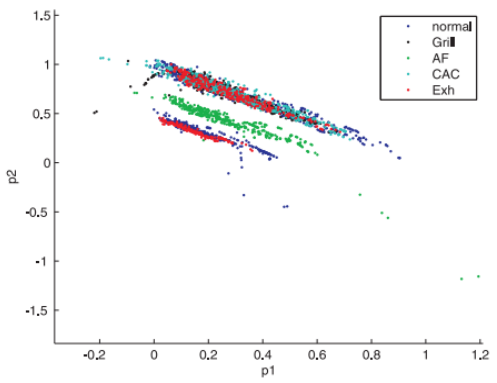


Fig 4: Model parameters

(Recursive Least Square Method)

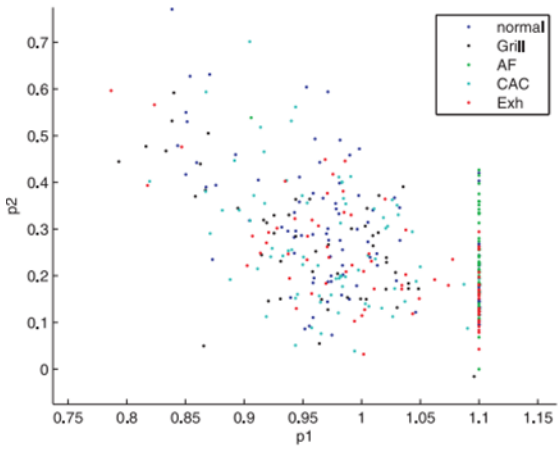


Fig 5: Model parameters

(Least Absolute Shrinkage and Selection Operator Method)

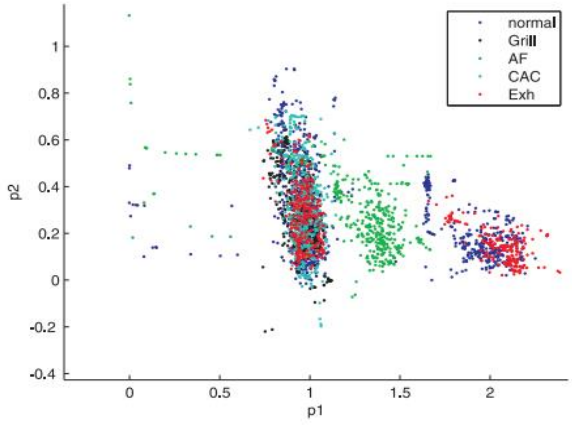


Fig 6: Model parameters

(Recursive Least Square Method)

Therefore, to use supervised learning in order to analyse how much useful information is there in those relations.

V. EVALUATION

Three different classifiers were used: linear regression [8], support vector machine [9] and random forest [7]. Each classifier has been used for multi class classification within the model parameter space generated during the system monitoring step, from either Least Absolute Shrinkage and Selection Operator or Recursive Least Square estimators.

In all cases, the input to the classifier is array W, which contains all the estimates for all models found over time. Both the forgetting factor (for Recursive Least Square) and the data slice size (for Least Absolute Shrinkage and Selection Operator) are parameters for tuning. Larger slices and forgetting factor gives better signal to noise ratio and a more robust solution. On the other hand, they are less sensitive to faults that only appear under certain conditions. In our case, a partially clogged air filter will only have a visible Figure 5: Model parameters (Least Absolute Shrinkage and Selection Operator method). Fig 6: Model parameters (Recursive Least Square method) effect if the engine is running at high power, since this is the only situation when a large air flow is required. In order to visualize the behaviour of our unsupervised relation discovery method, the classification task was run a number of times with different time slices and forgetting factors. Fig 7 and 8 present the result of that experiment. It is easily seen that choosing too small forgetting factor for Recursive Least Square is detrimental. On the other hand, the effect of choosing too small data slices is not visible, at least for reasonable window sizes.

In general, the random forest classifier outperforms both support vector machine and linear classifier by a pretty large margin. Besides that, Recursive Least Square estimator appears to be a little better than the Least Absolute Shrinkage and Selection Operator estimator, but the difference is not huge.

An interesting observation is that the number of data slices does not have a big impact on the classification accuracy, but there is a definite sweet point for the forgetting factor at 0.001. As a final comment, the resulting classification error appears to be rather high, but it is important to take into account that this data set is a pretty difficult one.

There is a lot of different external influences that disturb the "normal" operation of a truck, and the low quality of many of the sensors result in high levels of noise in the data. Also, for the predictive maintenance needs, it is not necessary to achieve Cent percentage or close accuracy - it is usually enough to detect faults some proportion of the time, since, often more interested in following gradual wear rather than abrupt failures.

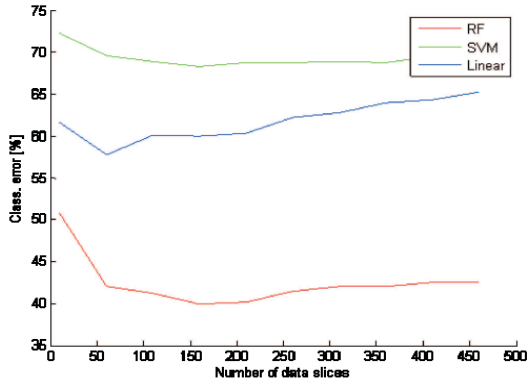


Fig 7: Classification error
(Least Absolute Shrinkage and Selection Operator Method)

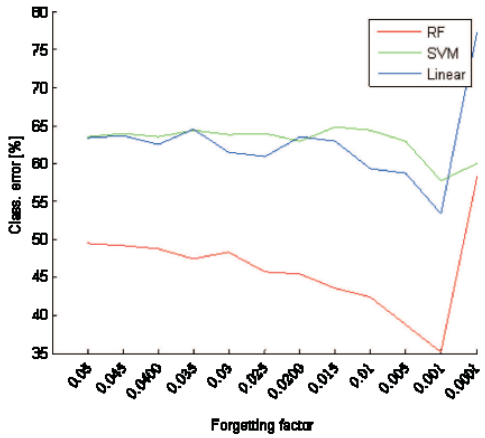


Fig 8: Classification error
(Recursive Least Square Method)

The seemingly low overall classification accuracy can also be partially attributed to the lack of dedicated sensors: in particular, neither of the four faults have analysed is currently being detected for in-production Products.*B*

VI. CONCLUSION

This paper presented a method for automatic discovery of interesting relations between time series of product signal data. On the data logged from a product and resulting models can be used for diagnostics using supervised learning. This is an important step towards a system that would be able to analyse onboard data on real products and detect anomalies in an autonomous way. This is very much work in progress and there are numerous directions to extend those results. An obvious thing is to look into ways of improving classification accuracy: used three well - known learning algorithms with defaults settings, but there is room for improvement in both the learning phase itself, as well as in the estimation of model parameters. Least Absolute Shrinkage and Selection Operator and Recursive Least Square implemented two methods, but there are many other potential solutions. Advantages and flaws of both of those methods are identified, so it would be interesting to look into possibility of developing some kind of hybrid approach.

REFERENCES

[1]. Ashish Kumar and Kavita Choudhary - Student, Dept. of Computer Science and Engineering, Jagannath Uni: A survey: Data Mining System for Mobile Devices, International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 2, February-2014

[2]. Byttner, Rognvaldsson, and Svensson., Consensus self-organized models for fault detection (COSMO). Engineering Applications of Artificial Intelligence, 24 (5): 833 – 839, 2011.

[3]. Lacaille and come. Visual mining and statistics for a turbofan engine fleet. In IEEE Aerospace Conference, pages 1–8, Mar, 2011.

[4]. Silva, - Diagnostics based on the statistical correlation of sensors. Technical Report 2008-01-0129, Society of Automotive Engineers (SAE), 2008.

[5]. Zhang, Gantt, Rychlinski, Edwards, Correia, and Wolf. Connected vehicle diagnostics and prognostics, concept, and initial practice. IEEE Transactions on Reliability, 58 (2): 286 – 294, 2009.

[6]. http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

[7]. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[8]. http://code.google.com/p/randomforest-matlab/.