APPLICATION OF RANDOM FORESTS TO ENGINE HEALTH MONITORING

Julien Ricordeau*, Jérôme Lacaille*
***Snecma, Rond-Point René Ravaud, Réau, 77550 Moissy-Cramayel cedex, France**

Keywords: *Health Monitoring, Random Forest, Diagnostic*

Abstract

Advanced Health Monitoring is becoming a standard for new engine applications, in order to enable in-service event reduction and engine maintenance optimization. The efficiency of these functionalities is driven by the capability to model the engine behavior and to identify engines on healthy or unhealthy conditions. Advances on health monitoring have been made possible by the use of machine learning techniques for both regression and classification. Among recent regression and classification techniques, decision tree bagging or so-called Random Forest appears to be quite promising in terms of accuracy and capacity to handle different types of inputs. This paper presents possible application of Random Forests to engine health monitoring.

1 General Introduction

Advanced Health Monitoring is becoming a standard for new engine applications, in order to enable in-service event reduction and engine maintenance optimization. The goal is to reduce operational events such as IFSD (In Flight Shut Down), ATO (Aborted Take-Off) et D&C (Delay & Cancellation) and to substitute them with maintenance operations that are planned long enough in advance in order to minimize their operational impacts for the airlines. IFSDs and ATOs are very seldom but still stressing for the pilots and they often produce secondary damages that might increase reparation costs. D&Cs are often not critical but occur more often; their consequences can be traffic disorganization, customer dissatisfaction. They are partially linked to procedures and controls to perform troubleshooting.

The performance of Engine Health Monitoring functionalities is driven by the capability to model the engine behavior and to identify engines on healthy or unhealthy conditions. Advances on Health Monitoring have been made possible by the use of machine learning techniques for both regression and classification. For instance, Random Forests have become very popular in literature the past years. Their capability to handle different types of inputs makes them quite versatile. The ability to treat discontinuities in the input space (operating conditions, change in engine configuration…) while controlling robustness offers good potential for this method in Engine Health Monitoring. These interesting properties come with a major drawback in terms of understanding for the expert in charge of validating the algorithm. Contrary to single decision trees, the Random Forest decision mechanism suffers from a lack of clarity for the expert who can only see it as a black-box model and loses the explanation from the input of the algorithm to the output diagnostic. In this paper, Random Forest is used for prediction and classification on some cases. Some ideas about the way to present the results to expert for validation and the link to the expert knowledge to the Random Forest structure are given.

2 Introduction to Engine Health Monitoring

2.1 EHM General Philosophy

As described in [1], fault detection algorithm can be described as the appending of two algorithms:

- CRN (Context Removal and Normalization): a standardization process that normalize observations to make them as if they were acquired always in the same context conditions;
- FDI (Failure Detection and Identification): a detection and classification algorithm that first diagnoses abnormalities and then identifies failures among a predefined list of possible degradation causes.

This low-level to high level diagnosis scheme also proposes a methodology to help experts in the analysis process of each possible failure.

This study will focus on CRN and Fault classification alternative to FDI.

2.2 CRN description:

CRN is extensively described in [1,4]. The step of interest here is the step 9 that corresponds to the indicator construction.

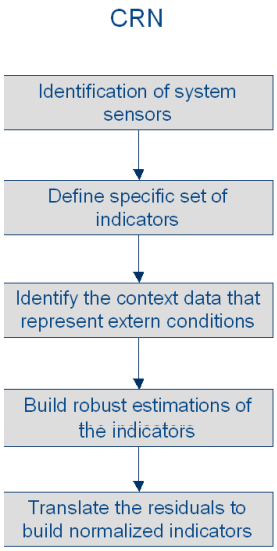

Fig. 1. CRN description

To implement such estimation for each indicator one need a set of observations Y (indicators) and X (context data) of good observations. “Good” refer to measurements done on engine that have no problem. In fact this is an easy task because most of the time there is no problem when using an engine. Thus building a regression on normal conditions is easily feasible. As we can get a lot of such observations, the quality of the model can be optimized.

Let $Y = (y_1, \mathrm{K}, y_m)$ be the vector of all indicators and X be the vector of context data. An estimation $\hat{Y}$ of an indicator Y is determined using an a priori parameterized function f and k operating conditions variables noted $x_1 \ldots x_k$ as shown in eq (1):

$$\hat{Y} = f(x_1, \ldots, x_k) \tag{1}$$

The result $\hat{Y}$ is a crude estimation of Y taking context data into account. The normalized data is thus obtained by computing $\tilde{Y}$ defined in eq (2):

$$\tilde{Y} = Y - \hat{Y} \tag{2}$$

This estimation is built from the difference of the real observation to the prediction. The whole result is readjusted to normal mean conditions (or defined standards).

3 Overview of Random Forests

3.1 Philosophy:

Random Forests have been introduced by Leo Breiman [3] in 2001 and quickly became popular for classification and regression tasks. Random Forests are a set of aggregated decision trees. All trees are taught on different bootstraps and there individual outputs are merged in a single output for the Random Forest. During the bootstrap, the data that are not used for an individual tree is put in a set called the out-of-the-bag (OOB).

An additional Randomness is introduced in the construction of the trees: at each node, a subset of all the possible variable is Randomly generated for split variable selection.

Leo Breiman has proposed an interesting way of estimating variable importance, which is useful to discuss the results with engine experts.

3.2 Variable importance:

The idea of Leo Breiman is to see the effect of breaking the dependencies between an input variable and the output variable on the Random Forest output.

Given a set of inputs and outputs, a model is trained and the out-of-the-bag error is estimated. The variable whose importance shall be estimated is shuffled in order to keep the variable marginal distribution but break the link with the output variable. This new set of data is passed through the Random Forest and the new out-of-the-bag error is evaluated.

For each variable, we can estimate the importance $I(x_i)$. The first step is to compute the difference $R(x_i)$ (see eq (3)) between errors obtained for the initial order and the random order:

$$R(x_i) = Error\ with\ random order\ of\ x_i - Error\ with\ initial order\ of\ x_i \quad (3)$$

The results is normalized to get $I(x_i)$ as defined in eq (4):

$$I(x_i) = 100 * \frac{R(x_i)}{\sum_k R(x_k)} \quad (4)$$

3.3 Feature selection:

The estimation of variable importance enables to select the relevant inputs. The idea is to introduce artificial inputs, randomly distributed. Variables that show less importance than the purely random variables are discarded.
Let us consider the data in Tab 1:

	Initial input			
Y	**x_1**	**x_2**	**…**	**x_k**
y_1	$x_{1,1}$	$x_{2,1}$	…	$x_{k,1}$
y_2	$x_{1,2}$	$x_{2,2}$	…	$x_{k,2}$
y_3	$x_{1,3}$	$x_{2,3}$	…	$x_{k,3}$
…	…	…	…	…
y_m	$x_{1,m}$	$x_{2,m}$	…	$x_{k,m}$

Tab. 1. Variable selection: initial dataset

This data is augmented with random inputs as shown in Tab 2:

	New input						
	Initial input				**Random variable**		
Y	**x_1**	**x_2**	**…**	**x_k**	**x_{k+1}**	**…**	**x_{k+n}**
y_1	$x_{1,1}$	$x_{2,1}$	…	$x_{k,1}$	$x_{k+1,1}$		$x_{k+n,1}$
y_2	$x_{1,2}$	$x_{2,2}$	…	$x_{k,2}$	$x_{k+1,2}$		$x_{k+n,2}$
y_3	$x_{1,3}$	$x_{2,3}$	…	$x_{k,3}$	$x_{k+1,3}$		$x_{k+n,3}$
…	…	…	…	…			
y_m	$x_{1,m}$	$x_{2,m}$	…	$x_{k,m}$	$x_{k+1,m}$		$x_{k+n,m}$

Tab. 2. Variable selection: augmented dataset

The variable importance estimation is performed and variables are ranked according to their importance:

Variable	**Variable importance**
x_m	15.4 %
x_2	10.5%
…	…
x_4	1.1%
x_{k+2}	0.9%
x_1	0.45 %
x_{k+3}	0.3 %
…	…
x_m	0.01%

Tab. 3. Variable selection: variable ranking according to importance

In this example the best-ranked artificial value is x_{k+2}, all variables with lower rank are not kept (Tab 4)

Variable	**Variable importance**
x_m	15.4 %
x_2	10.5%
…	…
x_4	1.1%
~~x_{k+2}~~	~~0.9%~~

~~x_1~~	~~0.45 %~~
~~x_{k+3}~~	~~0.3 %~~
~~…~~	~~…~~
~~x_m~~	~~0.01%~~

Tab. 4. Variable selection: variable elimination

3.4 Choice of Random Forests in EHM:

Several other mathematical objects such as neural networks and Support Vector Machines can perform the same kind of functionalities as Random Forests. The use of Random Forest has been motivated by the following properties that make Random Forests suitable for engineers use:

- Ease to understand for non experts:
- Ability to treat both discrete and continuous outputs
- Ease to tune:
- Toolbox aspect:
  - Regression
  - Classification
  - Embedded indication of variable importance

4 Model quality

One of the main difficulties in diagnostic system is to ensure a low false alarm rate, while improving fault detection. It is thus important to evaluate the quality of a model but also, when the model products a result, to be able to evaluate the quality of the result. The notions of MQV and PQV described in this paragraph are more extensively described in [2].

4.1 Mean Quality Value:

4.1.1 Philosophy:

The Mean Quality Value (MQV) is an indicator of a model intrinsic quality. It is a R2 coefficient based on out-of-the-bag error. The way to compute the estimate depends on the discrete or continuous case.

4.1.2 Continuous case:

For the continuous case, the error corresponds to an average estimation error. The data used for learning are run through trees for which they are in the out-of-the-bag set. A majority vote gives the result of the sub-Forest. This result is compared to the expected results; the error is thus, for N cases defined as eq (5) where σ is the standard deviation:

$$MQV = 1 - \left(\frac{\sigma(Y - \hat{Y})}{\sigma(Y)} \right)^2 \qquad (5)$$

4.1.3 Discrete case:

For the discrete case, the error corresponds to a misclassification count. The data used for learning are run through trees for which they are in the out-of-the-bag set. A majority vote gives the result of the sub-Forest. This result is compared to the expected results and the error is thus, for N cases:

$$MQV = 1/N * \sum_{i=1}^{N} 1_{\{x=0\}}(Y(i) - \hat{Y}(i)) \qquad (6)$$

An extension of the MQV is to compute a specific MQV for each target class.

4.2 PQV

4.2.1 Philosophy:

The Predictive Quality Value (PQV) is an estimate of the quality of the prediction when the model produces a result. It is an indicator going from 0 to 1, 0 corresponding to a poor quality of prediction.

4.2.2 Continuous case:

In the regression case, the error can be seen as a sum of errors for each tree of the Forest. The PQV is thus close to a gamma distribution. The PQV construction process is the following:

- the Random Forests error is estimated based on the inputs value; this is performed by another Random Forest. The new Random Forest, called PQV Random Forests, thus estimate the estimation error based on the inputs
- The error modeled by the PQV Random Forests as a marginal

distribution on which a gamma law is fitted, which gives parameters *a* and *b*.

When a result is computed, the PQV Random Forests is run on the input data to estimate the a priori error $\hat{e}$, used in the PQV estimation

$$PQV = P(e > \hat{e}) = \int_{\hat{e}}^{1} \Gamma(x, a, b)dx \qquad (7)$$

This definition implies that a high PQV is equivalent to a large possible area for the error.

4.2.3 Discrete case:

In the classification case, the mislabeling error is estimated thanks to an error Random Forest whose trees will estimate if the result is wrong or correct {true for a wrong result, false otherwise}.

$$PQV = 1 - \frac{number\ of\ true}{number\ of\ trees} \qquad (8)$$

4.2.4 An example:

The following figure (Fig 3) shows a prediction (based on a Random Forest model) made for engine start sequence monitoring. The output is an engine transient time estimation based on other engine parameters such as feeding pressure, oil temperatures… This example will be shown more extensively in part 5.

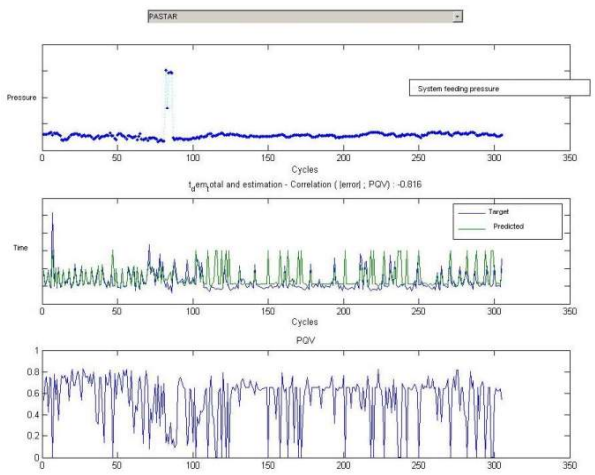

Fig. 2. Up: one of the external conditions, middle: prediction vs. target, down: predicted accuracy.

A correlation of – *0.816* is found between the norm of the error and the PQV. The negative sign means that the error and PQV evolve in opposite ways (a low error gives a high PQV) and the norm tells that those parameters are highly correlated.

5 Application 1: Start sequence monitoring CRN

5.1 Problem:

In order to monitor the engine during its start sequence, it is necessary to take into account external conditions such as feeding pressure, oil temperatures in order to detect abnormal conditions. The effect of these external conditions on start sequences indicators shall be removed. [5,6].

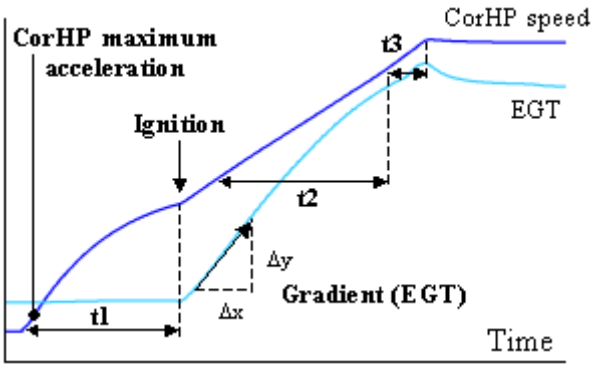

Figure 3: High Pressure compressor (CorHP) speed and exhaust gas temperature (EGT) during engine start

The graph below (Fig 4) shows the oil temperature just before an engine start. One immediately sees two kinds of temperatures that represent cold starts and hot starts. This is typically a case were an internal information (the oil temperature) must be considered as a context data. It clearly differentiates two classes of start procedures that the analysis algorithm must deal with.

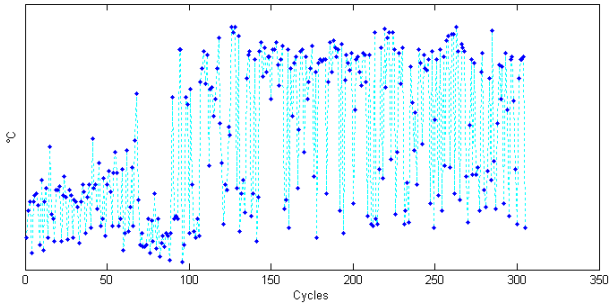

Figure 4: The oil temperature before start process. Each "cycle" corresponds to a new start sequence.

The false alarm rate has to be low in order to avoid useless repairs. Then the intersection between the distributions of the healthy and unhealthy starts has to be as small as possible so that the no detection rate is small. The variability introduced by the external conditions tends to increase this intersection.

5.2 Procedure:

Bench data from Snecma Villaroche industrial site on civil engines allowed establishing a predictive model for the different indicators.

These data are only healthy data, in other words, they are extracted from an engine without failure. Thus only the healthy distribution of the indicator could be learnt.

A set of 305 start sequences parameters with different contextual conditions has been recorded in a test cell. The 200 first data were used for training and the last ones for test.

5.3 Results of Random Forest :

The CRN information is provided through an IHM giving information about the estimate and the corresponding PQV. The context influences evaluated as previously described, are also shown. The following case shows the maximum Exhaust Gas Temperature (EGT) during the start sequence before and after normalization for a normal case:

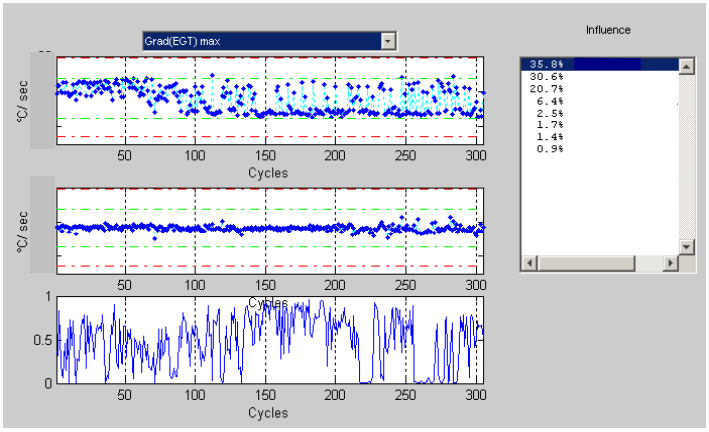

Fig. 5. up: data before normalization, middle: data after normalization, down: PQV.

The first graph represents the data before normalization and the data below after normalization. The normalization reduces the variability due to the external conditions.

5.4 Results exploration for experts:

When experts want to have a closer insight in the Random Forests response, the black-box aspect of the Random Forests can make the results interpretation difficult. It is then important to have tools to be able to better understand the model.

5.4.1 Variable importance:

The main tool is the variable importance computation that enables to link the variables. This is particularly valuable for experts who thus can check if the Random Forests have caught the physics of the phenomenon.

5.4.2 Visualization:

A second tool has been designed in order to navigate. The goal of this tool is to be able to navigate in the Random Forests outputs from the space of inputs.

The idea is to give the output mapping on a projection plan:

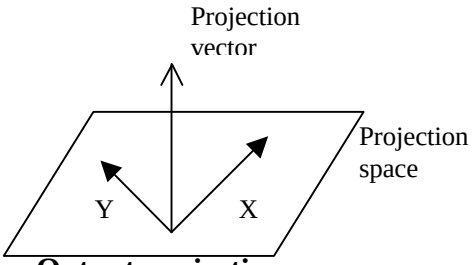

Fig. 6. up: Output projection

On the next figures, the labels represent:

1. Output value and PQV/Estimate selector:
- enables to choose the output data to display
- enables to choose between PQV and output estimate

2. Input variable influence display:
- displays the variable influences

3. Value color bar:
- sets the color code for the mapping

4. Output map:
- maps the projection of the multi-dimensional outputs to a 2-dimensional space. The color describes the value

5. Projection value selector:
- enables to change the projection direction with a scrollbar or by typing directly the desired value

6. Projection value display and plan variable selector:
- enables to see the projection direction
- enables to select X and Y plan axes.

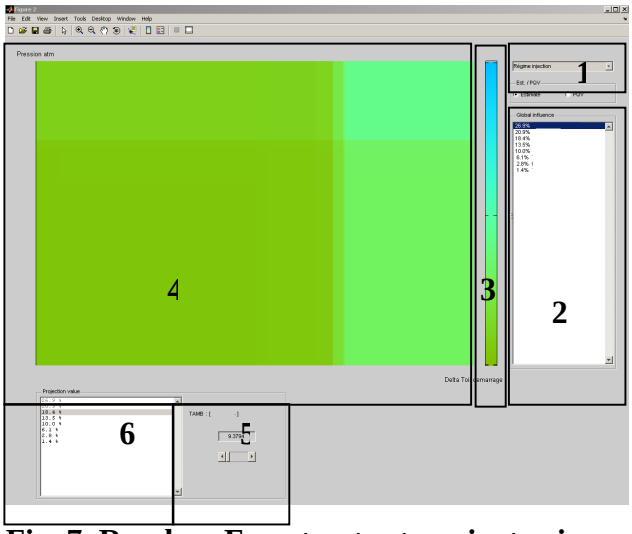

Fig. 7. Random Forest output navigator in estimate mode

The example illustrated by Fig 7 shows the estimation of the core speed versus feeding pressure and oil temperature with fixed values for other parameters (projection vector). Different levels of value are shown on the space. The variable importance estimation gives high importance for feeding pressure and oil temperature. The mapping shown in Fig 7 enables to see that the result depends on the values of those two parameters; otherwise, the output will be close to a single color plan.

By clicking on the point in the map, the user can have a precise value.

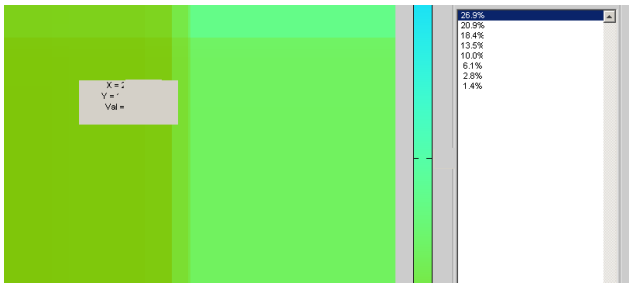

Fig. 8. up: Particular value descriptor

6 Application 2: Engine Gas Path CRN & Engine fault classification

6.1 Problem:

6.1.1 Description:

During flight operations, Aircraft records engine performance parameters during cruise in order to monitor the engine and detect any performance issue and localize them in the engine. We can consider that the engine is a set of five pieces a fan, booster, HP compressor, HP turbine, and LP turbine. Several techniques exist in order to evaluate faults, using for instance an estimation of the efficiencies and permeability of the different modules or directly classify the engine faults or absence of faults based on the recorded parameters [7].

The set of parameters taken for the study is the following:

Name	Description	Context
XM	Inlet Mach number	X
T12	Inlet Temperature	X
ALT	Altitude	X
PCN12	LP spool speed	X
PCN25	HP spool speed	
WF36	Fuel flow rate	
P25	Compressor inlet pressure	
PS3	Compressor Exit pressure	
PS13	Fan tip exit pressure	
T25	Compressor inlet temperature	
T3	Compressor exit temperature	
T495CC	HP turbine exit temperature	
T5	LP turbine exit temperature	

Fig. 9. Data available for performance analysis

We show here an example of CRN and an engine fault classifier.

6.1.2 CRN:

The idea here is still to remove context effect from the parameters we would like to study. In the case of our study, we use altitude, inlet temperature, Mach number and engine LP spool speed. The PLA could also be used instead of the LP spool speed since there is usually an immediate relationship between those two parameters.

6.1.3 Engine fault classifier:

Contrary to the FDI presented in [1], we will directly use the residuals coming from the CRN in order to classify the engine into six classes: health, fan issue, booster issue, HP compressor issue, HP turbine issue, LP turbine issue. There is no detection of abnormality

6.1.3 Algorithm description:

The proposed algorithm is composed of a CRN task and an engine fault classifier as described in Fig 10:

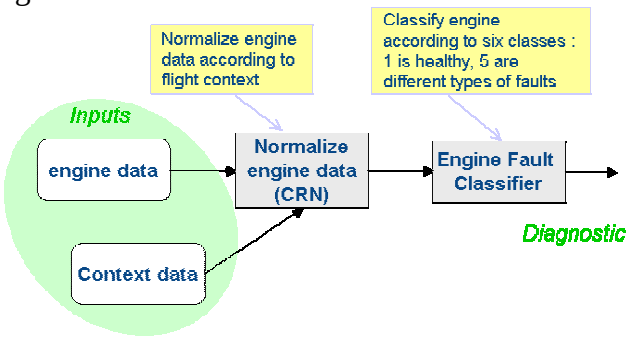

Fig. 10. Algorithm description

6.2 Procedure:

Data have been simulated through an engine model on which external conditions can vary and on which faults can be seeded.

A first set of 300 data with different conditions is used in order to train and test the CRN. The 200 firsts points are used for training and the 100 last for test.

A second test of 60 cases is simulated, corresponding to 10 examples of each class. 7 of each class are taken for training and 3 for test.

The set corresponds to all the available data including the context data. The learning phase is done with the variable selection procedure described previously.

6.3 Results of Random Forest:

6.3.1. CRN results:

On the next figure, the two first graphs show the input data (raw and renormalized) but with the same y-scale.

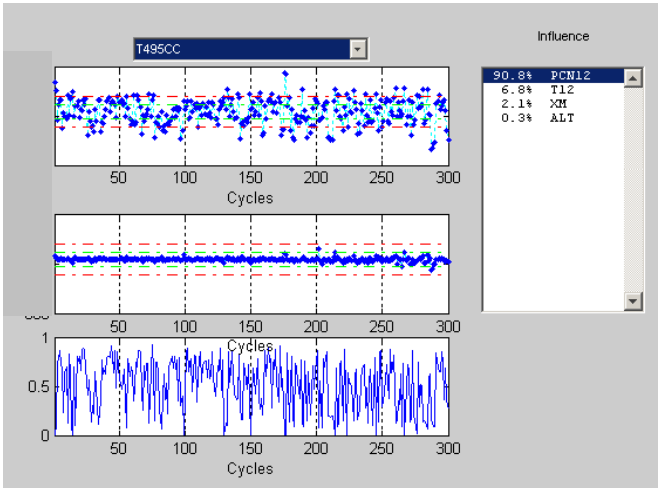

Fig. 11. up: data before normalization, middle: data after normalization, down: PQV.

The CRN enables to reduce drastically the dependence of engine sensors output from the context.
The Random Forest used for CRN shows MQVs higher than 0.96 for all the outputs.

6.3.2. Fault classification results:

The Random Forest Classifier shows good results since on this simplified case, all the good engine are well classified even though the MQV is not so high: 0.310 and the PQV does not give good results (Note: the error being zero, the correlation with the PQV is not well defined). This is due to the lack of training cases. The Random Forests used for PQV estimation has difficulties to estimate correctly the error.

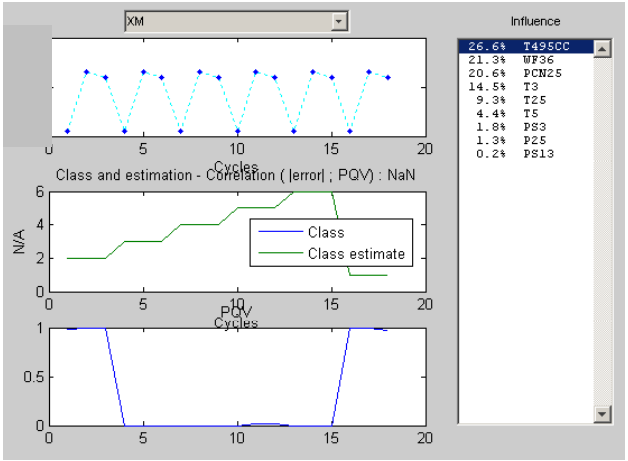

Fig. 12. Fault classification results with input variable selection. Up: input data, middle: classification result, down: PQV.

It is interesting to see that, during the learning procedure and the variable selection process, all the contextual variables have been discarded, which shows that main of the contextual effect has been removed through CRN. If we train the Random Forest with no input variable selection, we get very low influence for the context data:

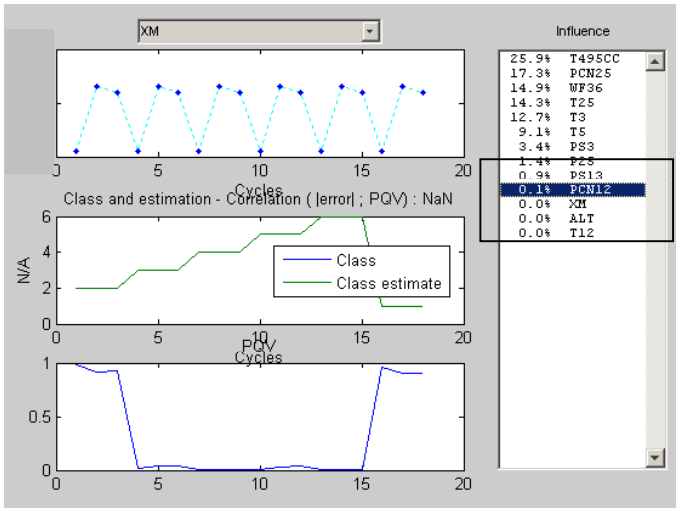

Fig. 13. Fault classification results without input variable selection, showing low influence of context data

A second interesting point is the variable importance for the classification. This influence is computed on all the faults, which does not mean that it is correct to discriminate each fault from the healthy condition. The three main parameters are HP turbine exit temperature, Fuel Flow and core speed, which are usually used by experts to monitor the engine.

In order to have a better insight of the result, it is possible to use the same visualization tool than in the start sequence case. Fig 14 shows the results on the core speed vs. HP turbine exit temperature. In this plot, several classes appear, showing the importance of those values for engine fault classification.

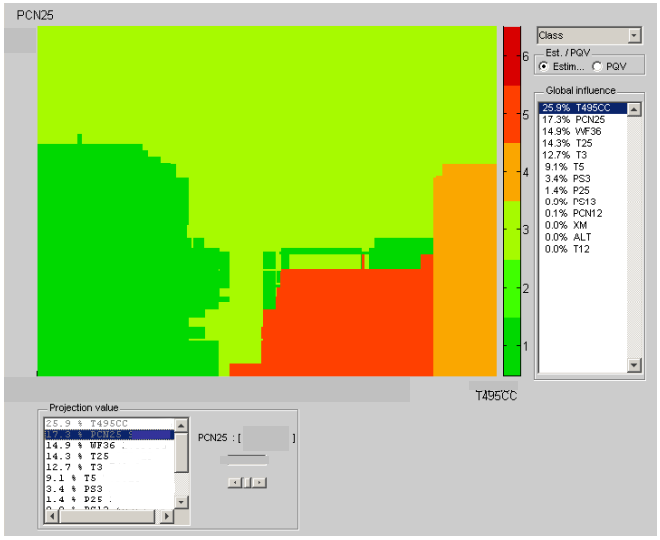

Fig. 14. Fault classification output navigator

The tool enables to show the effect of increasing in Fuel Flow; doing so, one gets more HP fault: Fig 15.

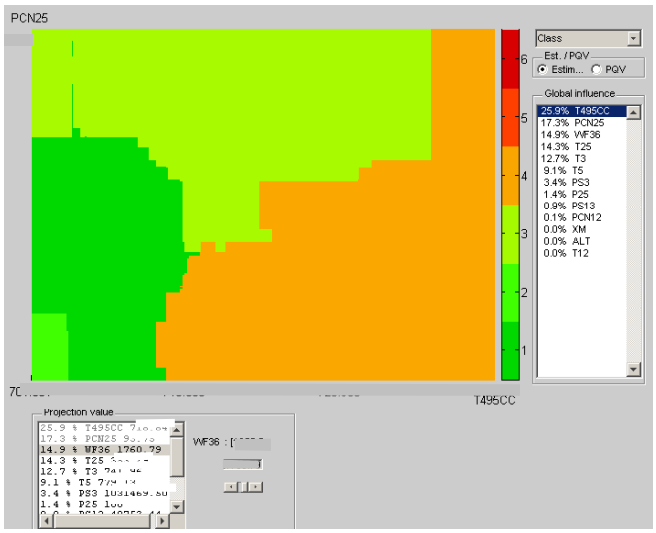

Fig. 15. Fault classification output navigator for higher Fuel Flow

Conclusion:

This paper shows how Random Forests can be used in Engine Health Monitoring. The power of Random Forests and the different toolboxes aspects of Random Forests make it particularly interesting for monitoring tasks. Some results have been shown on start sequence monitoring and engine performance faults monitoring. These results are defined on easy cases (specially true for performance) but enable to show possible uses of Random Forests. Proposals for MQV and PQV have been made and some paths on the way for an expert to interact with the Random Forests models have been shown.

References

[1] Lacaille J. Standardized Failure Signature for a Turbofan Engine, *in Proceedings of 2009 IEEE Aerospace Conference, Big Sky, MO.*

[2] Lacaille J. Validation of Health-Monitoring Algorithms for Civil Aircraft Engines, *in Proceedings of 2010 IEEE Aerospace Conference, Big Sky, MO.*

[3] Breiman L Random Forests *Machine Learning, 45(1), 5–32*. 2001

[4] Lacaille J., Snecma, Patented CRN Algorithm FR0858608, Context Removal and Normalization, 2008.

[5] C. Aurousseau, A. Aulsloos, J-R. Massé, P. Mouton, X. Flandrois, Snecma, Patented Engine start capability, FR0950839, Système de surveillance de l'état de santé des équipements intervenant dans la capacité de démarrage d'un turboréacteur, 2009.

[6] A. Ausloos et all. Estimation of monitoring indicators using regression methods; Application to turbofan start sequence, *ESREL 2009, Prague, Czech Republic*.

[7] Weizhong Yan, Application of Random Forest to Aircraft Engine Fault Diagnosis. *IMACS Multiconference on "Computational Engineering in Systems Applications"(CESA), October 4-6, 2006, Beijing, China*

Copyright Statement