

# Anomaly Detection Based on Confidence Intervals Using SOM with an Application to Health Monitoring

Anastasios Bellas<sup>1</sup>, Charles Bouveyron<sup>2</sup>, Marie Cottrell<sup>1</sup>, and Jerome Lacaille<sup>3</sup>

<sup>1</sup> SAMM, Université Paris 1 Panthé on-Sorbonne  
90 rue de Tolbiac, 75013 Paris, France

{anastasios.bellas,marie.cottrell}@univ-paris1.fr

<sup>2</sup> Laboratoire MAP5, Université Paris Descartes & Sorbonne Paris Cité  
45 rue des Saints-Pères, 75006 Paris, France

charles.bouveyron@parisdescartes.fr

<sup>3</sup> SNECMA, Rond-Point René Ravaud-Réau,  
77550 Moissy-Cramayel CEDEX, France

jerome.lacaille@sneema.fr

**Abstract.** We develop an application of SOM for the task of anomaly detection and visualization. To remove the effect of exogenous independent variables, we use a correction model which is more accurate than the usual one, since we apply different linear models in each cluster of context. We do not assume any particular probability distribution of the data and the detection method is based on the distance of new data to the Kohonen map learned with corrected healthy data. We apply the proposed method to the detection of aircraft engine anomalies.

**Keywords:** Health Monitoring, aircraft, SOM, clustering, anomaly detection, confidence intervals.

## 1 Introduction, Health Monitoring and Related Works

In this paper, we develop SOM-based methods for the task of anomaly detection and visualization of aircraft engine anomalies.

The paper is organized as follows : Section 1 is an introduction to the subject, giving a small review of related articles. In Section 2, the different components of the system proposed are being described in detail. Section 3 presents the data that we used in this application, the experiments that we carried out and their results. Section 4 presents a short conclusion.

### 1.1 Health Monitoring

Health monitoring consists in a set of algorithms which monitor in real time the operational parameters of the system. The goal is to detect early signs of failure, to schedule maintenance and to identify the causes of anomalies.

Here we consider a domain where Health Monitoring is especially important: aircraft engine safety and reliability. Snecma, the french aircraft engine constructor, has developed well-established methodologies and innovative tools: to ensure the operational reliability of engines and the availability of aircraft, all flights are monitored. In this way, the availability of engines is improved: operational events, such as D&C (Delay and Cancellation) or IFSD (In-flight Shut Down) are avoided and maintenance operations planning and costs are optimized.

## 1.2 Related Work

This paper follows other related works. For example, [9] have proposed the *Continuous Empirical Score* (CES), an algorithm for Health Monitoring for a test cell environment based on three components: a clustering algorithm based on EM, a scoring component and a decision procedure.

In [8,3,7], a similar methodology is applied to detect change-points in Aircraft Communication, Addressing and Reporting System (ACARS) data, which are basically messages transmitted from the aircraft to the ground containing on-flight measurements of various quantities relative to the engine and the aircraft.

In [4], a novel *star* architecture for Kohonen maps is proposed. The idea here is that the center of the star will capture the normal state of an engine with some rays regrouping normal behaviors which have drifted away from the center state and other rays capturing possible engine defects.

In this paper, we propose a new anomaly detection method, using statistical methods such as projections on Kohonen maps and computation of confidence intervals. It is adapted to large sets of data samples, which are not necessarily issued from a single engine.

Note that typically, methods for Health Monitoring use an extensive amount of expert knowledge, whereas the proposed method is fully automatic and has not been designed for a specific dataset.

Finally, let us note that the reader can find a broad survey of methods for anomaly detection and their applications in [2] and [10,11].

## 2 Overview of the Methodology

Flight data consist of a series of measures acquired by sensors positioned on the engine or the body of the aircraft. Data may be issued from a single or multiple engines. We distinguish between *exogenous* or *environmental* measures related to the environment and *endogenous* or *operational* variables related to the engine itself. The reader can find the list of variables in Table 1. For the anomaly detection task, we are interested in operational measures. However, environmental influence on the operational measures needs to be removed to get reliable detection.

The entire procedure consists of two main phases.

1. The first phase is the *training* or *learning* phase where we learn based on healthy data.

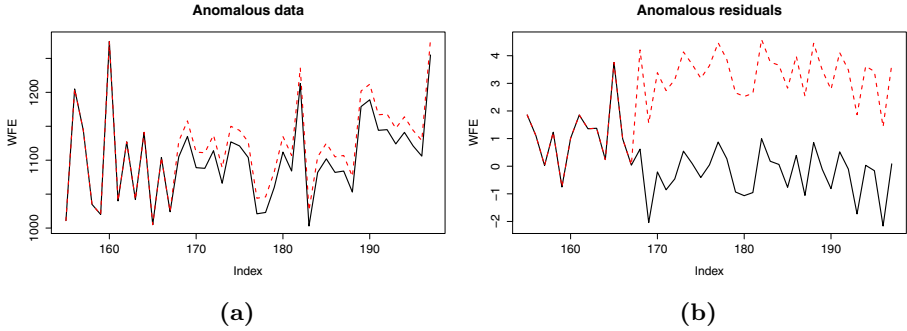
**Table 1.** Description of the variables of the cruise phase data

Name	Description
<b>Operational variables</b>	
EXH	Exhaustion gas temperature
N2	Core speed
Temp1	Temperature at the entrance of the fan
Pres	Static pressure before combustion
Temp2	Temperature before combustion
FF	Fuel flow
<b>Environmental variables</b>	
ALT	Altitude
Temp3	Ambient temperature
SP	Aircraft speed
N1	Fan speed
<b>Other variables</b>	
ENG	Engine index
AGE	Engine age

- We cluster data into clusters of environmental conditions using only environmental variables.
  - We correct operational measures variables from the influence of the environment using a linear model, and we get the residuals (corrected values).
  - Next, a SOM is being learned based on the residuals.
  - We calibrate the anomaly detection component by computing the confidence intervals of the distances of the corrected data to the SOM.
2. The learning phase is followed by the *test* phase, where novel data are taken into account.
- Each novel data sample is being clustered in one of the environment clusters established in the training phase.
  - It is then being corrected of the environment influence using the linear model estimated earlier.
  - The test sample is projected to the Kohonen map constructed in the training phase and finally, the calibrated anomaly detection component determines if the sample is normal or not.

**Clustering of the Environmental Contexts.** An important point is the choice of the clustering method. Note that clustering is carried out on the *environmental* variables. The most popular clustering method is the Hierarchical Ascending Classification [5] algorithm, which allows us to choose the number of clusters based on the explained variance at different heights of the constructed tree.

However in this work our goal is to develop a more general methodology that could process even high-dimensional data and it is well-known that HAC is not adapted to this kind of data. Consequently, we are particularly interested



**Fig. 1.** An example of an anomaly of the FF variable of the cruise flight data (a) Superposition of the healthy data (solid black lines) and the data with anomalies (dashed red line) (b) Superposition of the corrected data obtained from the healthy data and corrected data obtained from corrupted data. The anomaly is visible only on corrected data.

in methods based on subspaces such as HDDC [1], since they can provide us with a parsimonious representation of high-dimensional data. Thus, we will use HDDC for the environment clustering, despite its less good performance for low-dimensional data.

**Corrupting Data.** In order to test the capacity of the proposed system to detect anomalies, we need data with anomalies. However, it is very difficult to get them due to the extraordinary reliability of the aircraft engines and we cannot fabricate them because deliberately damaging the engine or the test cell is clearly not an option. Therefore, we create artificial anomalies by corrupting some of the data based on expert specifications that have been established following well-known possible malfunctions of aircraft engines.

Corrupting the data with anomalies is carried out according to a *signature* describing the defect (malfunction). A signature is a vector  $\mathbf{s} \in \mathbb{R}^p$ . Following  $\mathbf{s}$ , a corruption term is added to the nominal value of the signal for a randomly chosen set of successive data samples.

Figure 1a gives an example of the corruption of the FF variable for one of the engines. Figure 1b shows the corrupted variable of the corrected data, that is, after having removed the influence of the environmental variables.

## 2.1 Clustering the Corrected Data Using a SOM

In order to build an anomaly detection component, we need a clustering method to define homogeneous subsets of corrected data. We choose to use the SOM algorithm [6] for its well-known properties of clustering organized with respect to each variable of the data as well as its visualization ability.

The output of the algorithm is a set of prototype vectors that define an "organized" map, that is, a map that respects the topology of the data in the input space. We can then color the map according to the distribution of the data for each variable. In this way, we can visually detect regions in the map where low or high values of a given variable are located. A smooth coloring shows that it is well organized. In the next section, we show how to use these properties for the anomaly detection task.

## 2.2 Anomaly Detection

In this subsection, we present two anomaly detection methods that are based on confidence intervals. These intervals provide us with a "normality" interval of healthy data, which we can then use in the test phase to determine if a novel data sample is healthy or not.

We have already seen that the SOM algorithm associates each data sample with the nearest prototype vector, given a selected distance measure. Usually, the Euclidean distance is selected. Let  $L$  be the number of the units of the map,  $\{\mathbf{m}_l, l = 1, \dots, L\}$  the prototypes. For each data sample, we calculate  $\mathbf{x}_i$ , its distance to the map, namely the distance to its nearest prototype vector:

$$d(\mathbf{x}_i) = \min_l \|\mathbf{x}_i - \mathbf{m}_l\|^2 \quad (1)$$

where  $i = 1, \dots, n$ . Note that this way of calculating distance will give us a far more useful measure than if we had just utilized the distance to the global mean, *i.e.*  $d(\mathbf{x}_i) = \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2$ .

The confidence intervals that we use here are calculated using distances of training data to the map. The main idea is that the distance of a data sample to its prototype vector has to be "small". So, a "large" distance could possibly indicate an anomaly. We propose a global and a local variant of this method.

**Global Detection.** During the training phase, we calculate the distances  $d(\mathbf{x}_i)$ ,  $\forall i$ , according to Equation (1). We can thus construct a confidence interval by taking the 99-th percentile of the distances,  $P_{99}(\{d(\mathbf{x}_i), \forall i\})$ , as the upper limit. The lower limit is equal to 0 since a distance is strictly positive. We define thus the confidence interval  $\mathcal{I}$

$$\mathcal{I} = [0, P_{99}(\{d(\mathbf{x}_i), \forall i\})] \quad (2)$$

For a novel data sample  $\mathbf{x}$ , we establish the following decision rule:

$$\begin{cases} \text{The novel data sample is healthy, if } d(\mathbf{x}) \in \mathcal{I} \\ \text{The novel data sample is an anomaly, if } d(\mathbf{x}) \notin \mathcal{I}. \end{cases} \quad (3)$$

The choice of the 99-th percentile is a compromise taking into account our double-sided objective of a high anomaly detection rate with the smallest possible false alarm rate. Moreover, since the true anomaly rate is typically very small in civil aircraft engines, the choice of such a high percentile, which also serves as an upper bound of the normal functioning interval, is reasonable.

**Local Detection.** In a similar manner, in the training phase, we can build a confidence interval for every cluster  $l$ . In this way, we obtain  $L$  confidence intervals  $\mathcal{I}_l$ ,  $l = 1, \dots, L$  by taking the 99-th percentile of the *per* cluster distances as the upper limit

$$\mathcal{I}_l = [0, P_{99}(\{d(\mathbf{x}_i) : \mathbf{x}_i \text{ in SOM cluster } l\})] \quad (4)$$

For a novel data sample  $\mathbf{x}$  (in the test phase), we establish the following decision rule:

$$\begin{cases} \text{The novel data sample, affected to SOM cluster } l, \text{ is healthy, if } d(\mathbf{x}) \in \mathcal{I}_l \\ \text{The novel data sample, affected to SOM cluster } l, \text{ is an anomaly if } d(\mathbf{x}) \notin \mathcal{I}_l. \end{cases} \quad (5)$$

### 3 Application to Aircraft Flight Cruise Data

In this section, we present the data that we used for our experiments as well as the processing that we carried out on them.

Data samples in this dataset are snapshots taken from the cruise phase of a flight. Each data sample is a vector of endogenous and environmental variables, as well as categorical variables. Data are issued from 16 distinct engines of the same type. For each time instant, there are two snapshots, one for the engine on the left and another one for the engine on the right. Thus, engines appear always in pairs. Snapshots are issued from different flights. Typically, there is one pair of snapshots per flight. The reader can find the list of variables in Table 1. The dataset we used here contains 2472 data samples and 12 variables.

We have divided the dataset into a training set and a test set. For the training set, we randomly picked  $n = 2000$  data samples among the 2472 that we dispose of in total. The test set is composed of the 472 remaining data samples. We have verified that all engines are represented in both sets. We have sorted data based on the engine ID (primary key of the sort) and for a given engine, based on the timestamp of the snapshot. We normalize the data (center and scale) because the scales of the variables were very different.

**Selection of the Number of Clusters in Environment Clustering.** Clustering is carried out on environmental variables to define clusters of contexts. Due to the large variability of the different contexts (extreme temperatures very high or very cold and so on), we have to do a compromise between a good variance explanation and a reasonable number of clusters (to keep a sufficient number of data in each cluster). If we compare HDDC to the Hierarchical Ascending Classification (HAC) algorithm in terms of explained variance, we observe that the explained variance is about 50 % for five clusters for both algorithms. And as mentioned before, we prefer to use HDDC [1] to present a methodology which can be easily adapted to high-dimensional data. Let  $K = 5$  be the number of clusters.

**Correcting the Endogenous Data from Environmental Influence.** We correct the operational variables of environmental influence using the procedure we described in section 2. After the partition into 5 clusters based on environmental variables, we compute the residuals of the operational variables as follows: if we set  $X^{(1)} = \text{N1}$ ,  $X^{(2)} = \text{Temp3}$ ,  $X^{(3)} = \text{SP}$ ,  $X^{(4)} = \text{ALT}$  et  $X^{(5)} = \text{AGE}$ , we write

$$Y_{rkj} = \mu + \alpha_r + \beta_k + \gamma_{1k}X_{rkj}^{(1)} + \gamma_{2k}X_{rkj}^{(2)} + \gamma_{3k}X_{rkj}^{(3)} + \gamma_{4k}X_{rkj}^{(4)} + \gamma_{5k}X_{rkj}^{(5)} + \varepsilon_{rkj} \quad (6)$$

where  $Y$  is one of the  $d = 6$  operational variables,  $r \in \{1, \dots, 16\}$  is the engine index,  $k \in \{1, \dots, 5\}$  is the cluster number,  $j \in \{1, \dots, n_{rk}\}$  is the observation index. Moreover,  $\mu$  is the intercept,  $\alpha_r$  is the effect of the engine and  $\beta_k$  the effect of the cluster.

**Learning a SOM with Residuals.** By analyzing the residuals, one can observe that the model succeeds in capturing the influence of the environment on the endogenous measures, since the magnitude of the residuals is rather small (between -0.5 and + 0.5). The residuals therefore capture behaviors of the engine which are not due to environmental conditions. The residuals are expected to be centered, *i.e.* to have a mean equal to 0. However, they are not necessarily scaled, so we re-scale them.

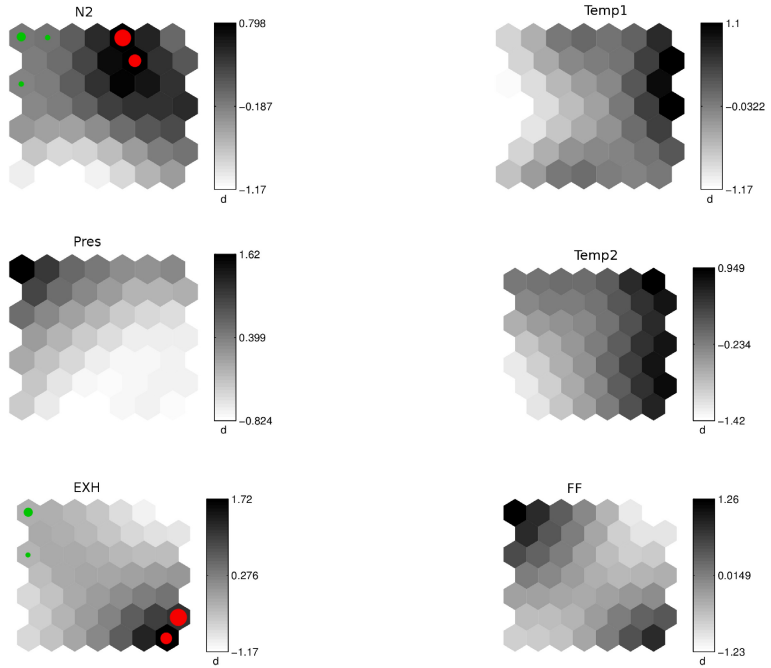
Generally speaking, since residuals are not smooth, we carry out smoothing using a moving average of width  $w = 7$  (central element plus 3 elements on the left plus 3 elements on the right). We note that by smoothing, we lose  $\lfloor \frac{w}{2} \rfloor$  data samples from the beginning and the end. Therefore, we end up with a set of 1994 residual samples instead of the 2000 that we had initially. Next, we construct a Self-Organizing Map (SOM) based on the residuals (Figure 2). We have opted here for a map of 49 neurons ( $7 \times 7$ ) because we need a minimum of observations per SOM cluster in order to calculate the normal functioning intervals with precision.

The last step is the calibration of the detection component by determining the global and local confidence intervals based on the distances of the data to the map. For the global case, according to Equation 2, we have:

$$\mathcal{I} = [0, 4.1707]$$

In a similar manner, we derive the upper limits of the local confidence intervals, ranging from 1.48 to 6.03.

**Test Phase.** In the test phase, we assume that novel data samples are being made available. We first corrupt these data following the technique proposed in Section 2. Snecma experts provided us with signatures of 12 known defects (anomalies), that we added to the data. For data confidentiality reasons, we are obliged to anonymize the defects and we refer to them as "Defect 1", "Defect 2" etc.



SOM 27–Jun–2013

**Fig. 2.** SOM built from the corrected training residuals for each of the  $p = 6$  endogenous variables. Black cells contain high values of the variable while white ones contain low values. Red dots refer to anomalies and green dots to healthy data for two different types of defects bearing on the variables N2 and EXH. The proposed method clusters them in different regions of the map. The size of each dot is proportional to the number of points of the cluster.

We start by normalizing test data with the coefficients used to normalize training data earlier. We then cluster data into environment clusters using the model parameters we estimated on the training data earlier. Next, we correct data from environmental influence using the model we built on the training data. In this way, we obtain the test residuals, that we re-scale with the same scaling coefficients used to re-scale training residuals.

We apply a smoothing transformation using a moving average, exactly like we did for training residuals. We use the same window size, *i.e.*  $w = 7$ . Smoothing causes some of the data to be lost, so we end up with 466 test residuals instead of the 472 we had initially.

Finally, we project data onto the Kohonen map that we built in the training phase and we compute the distances  $d(\mathbf{x})$  as in equation (1). We apply the decision rule, either the global decision rule of (3) or the local one of (5).



**Table 2.** Detection rate ( $tpr$ ) and false alarm rate ( $pfa$ ) for different types of defects and for both anomaly detection methods (global and local) for test data

Defect	Global detection		Local detection	
	tpr	pfa	tpr	pfa
Defect 1	100%	18,9%	100%	45,4%
Defect 2	100%	11,4%	100%	42,6%
Defect 3	100%	16,7%	100%	47,9%
Defect 4	100%	15,1%	100%	45,1%
Defect 5	96,7%	14,7%	100%	43,4%
Defect 6	100%	13,9%	100%	43,6%
Defect 7	96,7%	12,1%	96,7%	44,2%
Defect 8	100%	26,3%	100%	50%
Defect 9	100%	15,8%	100%	43,9%
Defect 10	100%	26,7%	100%	55,1%
Defect 11	100%	17,1%	100%	46,3%
Defect 12	100%	21%	100%	46,4%

In order to evaluate our system, we calculate the detection rate ( $tpr$ ) and the false alarms rate ( $pfa$ ):

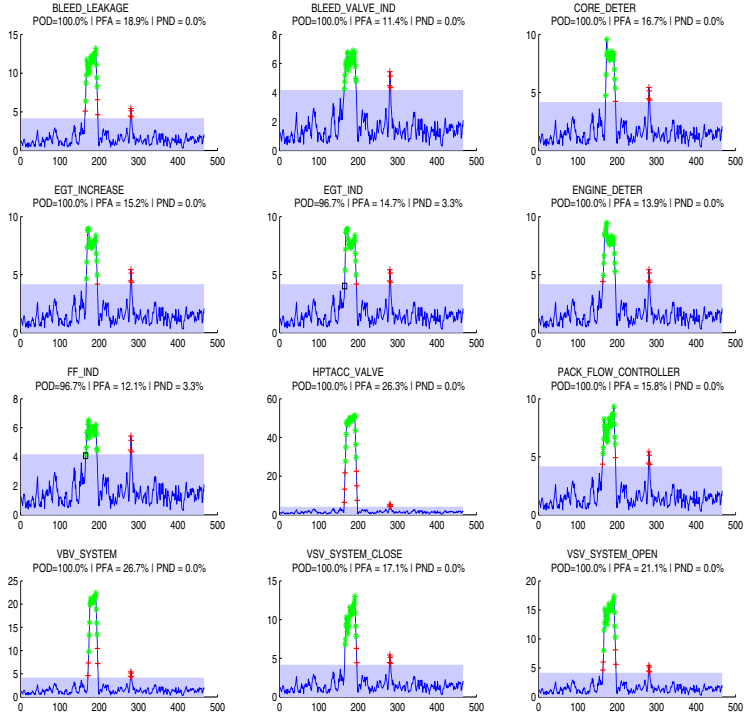
$$tpr = \frac{\text{number of detections}}{\text{number of anomalies}}$$

$$pfa = \frac{\text{number of non-expected detections}}{\text{number of detections}}$$

In Table 2, we can see detection results for all 12 defects and for both detection methods (global and local). It is clear that both methods succeed in detecting the defects, almost without a single miss. The global method has a lower false alarm rate than the local one. This is because in our example, confidence intervals cannot be calculated reliably in the local case since we have few data per SOM cluster.

Figure 3 shows the distance  $d$  of each data sample (samples on the horizontal axis) to their nearest prototype vector (Equation 1). The light blue band shows the global confidence interval  $\mathcal{I}$  that we calculated in the training phase. Red crosses show the false alarms and green stars the correct detections.

Due to limited space in this contribution, the figures related to the local detection can be found in the following URL: <https://drive.google.com/folderview?id=0B0EJciu-PLatZzdqR25oVjNNaTg&usp=sharing>



**Fig. 3.** Distances of the test data to their nearest prototype vector and the global confidence interval (in light blue). Red crosses show the false alarms and green stars show successful detection.

## 4 Conclusion and Future Work

We have developed an integrated methodology for the analysis, detection and visualization of anomalies of aircraft engines. We have developed a statistical technique that builds intervals of "normal" functioning of an engine based on distances of healthy data from the map with the aim of detecting anomalies. The system is first calibrated using healthy data. It is then fully operational and can process data that was not seen during training.

The proposed method has shown satisfying performance in anomaly detection, given that it is a general method which does not incorporate any expert knowledge and that it is, thus, a general tool that can be used to detect anomalies in any kind of data.

Another advantage of the proposed method is that the use of the dimension allows to carry out multi-dimensional anomaly detection in a problem of dimension 1. Moreover, the representation of the operational variables given by the use of the distance to the SOM is of a higher granularity than that of the distance from the global mean. Last but not least, the use of SOM allows us to give interesting visualizations of healthy and abnormal data, as seen in Figure 2.

An extension of our work would be to carry out anomaly detection for datastreams using this method. A naive solution would be to re-calibrate the components of the system with each novel data sample, but it would be very time-consuming. Instead, one can try to make each component of the system to operate on datastreams.

## References

1. Bouveyron, C., Girard, S., Schmid, C.: High-dimensional data clustering. *Computational Statistics & Data Analysis* 52(1), 502–519 (2007a)
2. Chandola, V., Banerjee, A., Kumar, V.: Outlier detection: A survey. *ACM Computing Surveys* 41(3) (2009)
3. Côme, E., Cottrell, M., Verleysen, M., Lacaille, J.: Aircraft engine health monitoring using self-organizing maps. In: Perner, P. (ed.) *ICDM 2010*. LNCS, vol. 6171, pp. 405–417. Springer, Heidelberg (2010a)
4. Côme, E., Cottrell, M., Verleysen, M., Lacaille, J., et al.: Self organizing star (sos) for health monitoring. In: *Proceedings of the European Conference on Artificial Neural Networks*, pp. 99–104 (2010b)
5. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York (1973)
6. Kohonen, T.: *Self-organizing maps*, vol. 30. Springer (2001)
7. Lacaille, J., Côme, E., et al.: Sudden change detection in turbofan engine behavior. In: *Proceedings of the Eighth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, pp. 542–548 (2011)
8. Lacaille, J., Gerez, V.: Online abnormality diagnosis for real-time implementation on turbofan engines and test cells, pp. 579–587 (2011)
9. Lacaille, J., Gerez, V., Zouari, R.: An adaptive anomaly detector used in turbofan test cells. In: *Proceedings of the Annual Conference of the Prognostics and Health Management Society* (2010)
10. Markou, M.: Novelty detection: a review-part 1: statistical approaches. *Signal Processing* 83(12), 2481–2497 (2003a)
11. Markou, M.: Novelty detection: a review-part 2: neural network based approaches. *Signal Processing* 83(12), 2499–2521 (2003b)