Automatic Detection of Rare Observations During Production Tests Using Statistical Models

Alex Mourer[1], Jérôme Lacaille[2], Madalina Olteanu[3] and Marie Chavent[4]

[1,2] *Safran Aircraft Engines, 77550 Réau, France*
alex.mourer@safrangroup.com jerome.lacaille@safrangroup.com

[1,3] *SAMM - EA 4543 Université Pantheon Sorbonne - France*
Madalina.Olteanu@univ-paris1.fr

[1,4] *INRIA Bordeaux Sud-Ouest CQFD team - France*
marie.chavent@math.u-bordeaux.fr

ABSTRACT

Engines are verified through production tests before delivering them to customers. During those tests, numerous measures are taken on different parts of the engine, considering multiple physical parameters. Unexpected measures can be observed. For this very reason, it is important to assess if these unusual observations are statistically significant.

However, anomaly detection is a difficult problem in unsupervised learning. The obvious reason is that, unlike supervised classification, there is no ground truth against which we could evaluate results. Therefore, we propose a methodology based on two independent statistical algorithms to double check the results. One approach is the Isolation Forest model which is specific to anomaly detection and able to handle a large number of variables. The goal of the algorithm is to find rare items, events or observations which raise suspicions by differing significantly from the majority of the data and, at the same time, it discriminates non-informative variables to improve estimation. One main issue of Isolation Forest is its lack of interpretability. Within this scope, we extend the Shapley values, interpretation indicators, to the unsupervised context to interpret the model outputs.

The second approach is the Self-Organizing Map (SOM) model which has nice properties for data mining by providing both clustering and visual representation. The performance of the method and its interpretability depend on the chosen subset of variables. In this respect, we first implement a sparse-weighted K-means to reduce the input space, allowing the SOM to give an interpretable discretized representation.

We apply both methodologies on aircraft engines data. Both approaches show similar results which are easily interpretable and exploitable by the experts.

1. INTRODUCTION

As an aircraft engines manufacturer, Safran verifies all individual engines before delivering to the customer during production tests. Those bench test operations generate lots of measures for different parts of the engines, resulting in multiple physical parameters acquisitions. As we may encounter unexpected measures, it is important to detect their causes and relevance. We build statistical methods to reach this goal.

Variations between performances of engines are common. Nevertheless, the production tests that verify essential engine functions before delivering it to an airline company are done in different bench test cells, under different ambient conditions, etc. A thermodynamic model is applied to compensate for context variations but there still exist some second level residuals we may have to compensate to enhance the quality of the measurements. They essentially depend on test bench components like slave cowls, but also sites and suppliers.

Therefore, one of the objectives is to take into account test bench components effects. Furthermore, there is no universally admitted way to evaluate unsupervised anomaly detection algorithms results. Hence, we proposed a new methodology based on two different algorithms.

- Large number of variables (>50) make statistical estimation challenging, especially w.r.t. the small number of engines (591). Therefore, a specific algorithm for anomaly detection, named Isolation Forest Liu et al. (2008), is proposed. Isolation-based methods measures the probability to be isolated and anomalies are those that have the highest probability. In randomly generated binary

trees, where instances are recursively partitioned, the trees produce noticeable shorter paths for anomalies. In fact, regions occupied by anomalies are low density regions, which result in a smaller number of partitions (shorter paths). Furthermore anomalies have, by definition, distinct feature-values and thus they are more likely to be separated early in the partitioning process.

- Then, another unsupervised algorithm named, Self-Organizing Map (SOM) is applied Kohonen (1982). It acts as an extension of the k-means algorithm that preserves as much as possible the topological structure of the data. Moreover, SOM has an intrinsic distance between prototypes and their direct neighbours. This latter representation can validate the estimation of Isolation Forest if the results coincide. Finally, SOM gives a discretized representation of the input space, which categorize anomalies. The categorization helps to understand the origin of the problem.

In addition to the rare events detection task, we need to provide explanations of the different models. Moreover, important parameters must be discovered at a local level to figure out flaws in a single engine and at global level to discern origins of unexpected variations and inherent bias.

2. DATA ANALYSIS

2.1. Structure of the Data

The characteristics of the test bench data used in the analysis are as follows:

- 14 variables are chosen in an expert-manner by domain experts of the performance team of Safran. They are not generally interested in other variables and thus we limit ourselves to this subset of variables. However, in future works we will consider a larger set of variables.
- 591 engines are observed.
- 4 stabilized points are considered.

A stabilized point, is a fixed level of performance for which all engines are tested and measurements acquired. They are ordered from the lowest to the highest level of performance. In the database, six are available but we do not consider the two first because they are taken at low engine speeds where there is a lot of variance in measures which makes them difficult to analyze. The measures of interest are listed below. Nine of them are numerical variables:

- FNIN1 : Thrust (FN: performance).
- XN12R : LP spool speed (N1: fan speed).
- XN25R : HP spool speed (N25: core speed).
- WF36 : Fuel flow.
- W2AR : Engine corrected air flow.
- P3 : HP compressor discharge pressure.
- T49C : LP turbine inlet temperature (EGT: exhausting gas temp).
- T3: HP compressor discharge temperature.
- P18QSC: Pressure section 18 normalized by standard conditions.

where HP and LP stand for High pressure and Low pressure respectively. The variables listed above are described in Figure 1. Moreover, four variables are categorical and represents test bench components:

- CELL : Bench.
- CNOZ : Primary nozzle.
- BMSN : Air nozzle.
- COWL : Nacelle.

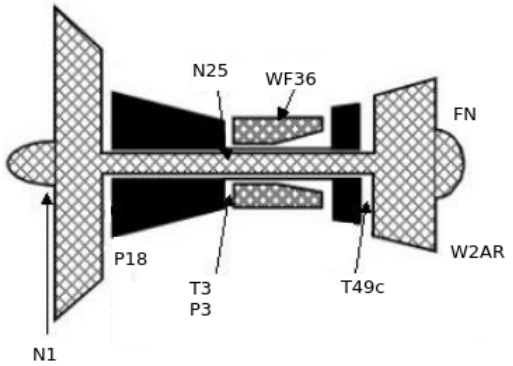

Figure 1. Simplified diagram of a turbofan engine where the different measured variables are specified.

2.2. Bias in Data

An analysis of the data shows that production tests data are biased by test bench components. However, normalizing variables successively by each test bench component is an acceptable method only if test bench components are independent with each others. In this case we found out, using a pairwise t-test, that this type of normalization does not remove the bias in the data. Thus, it is preferred to keep the non-standardized data and to include the bench components to our models, which will be able to handle interactions between variables.

3. EXPERT KNOWLEDGE TO DEFINE ANOMALIES

Some pre-treatments are needed before detecting unusual case in the data. As said before, it is normal to have variance in production tests data. Those fluctuations do not necessarily represent unexpected behavior. In practice, the behavior of an engine is not defined with measurements taken independently, but it is defined between pairs of measurements. Thus, an engine has a normal level of functioning if it has a "constant ratio" between some defined pairs of variables across the stabilized points considered.

In other words we do not define an anomaly considering the observed values but we construct a new data set from the observed one where each variable is constructed considering the dependence between pairs of measurements.

In the new data set the dependence between the pairs of variables is of importance, and especially the evolution of these dependencies across the different stabilized points. At each stabilized point a physical equation describes the relationship for each pair of variables and reveals the expected behavior of engines.

Figure 2 shows an example of the physical equation (red line) for the pair of variables (FNIN1, W2AR) and the observed values for the set of engines at the fourth stabilized point. The line gives one important information, that is the expected relationship between the thrust (FNIN1) and the mass flow rate (W2AR); which means that for a certain value of thrust we expect a certain mass flow rate.

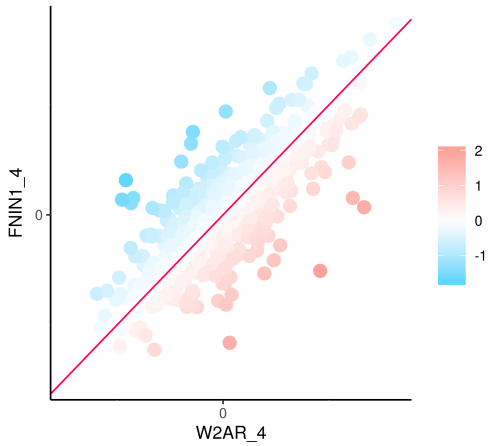

Figure 2. Representation of the functioning line for the pair of variables (FNIN1, W2AR) on the stabilized point four. The value of each engine is represented by the colors in the orthogonal space estimated with the RPCA.

However, the equation of the functioning line (red line) is unknown and its estimation can be done with the help of a Robust-Principal component analysis (PCA) as detailed in Candès et al. (2011). PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Therefore, the first axe of the PCA will model the functioning line, and the second axe will represent the space where the new variable will lie. The Robust PCA (RPCA) is an extension of PCA that is less sensible to extreme values, thus the slope of the functioning line will not depend on outliers and let them easier to detect.

3.1. Define Anomalies in Production Tests Data

For different stabilized points, we expect that an engine keeps a constant ratio between pairs of variables, i.e. their relationships do not change over the stabilized points. Therefore, we consider the engines values projected on the second axis obtained by the RPCA for each stabilized point. Then, we define a normal engine behavior as follow: "An engine has a normal behavior if it has similar projected values across the stabilized points".

Formally, let us consider an engine $i \in \{1, \dots, n\}$, a pair of variables $j \in \{1, \dots, J\}$ and a stabilized points $p \in \{1, \dots P\}$. Let $y^j_{i,p}$ be the projection of the engine i on the second axis of the RPCA for the pair of variables j at the stabilized point p. Then, the mean value over the stabilized points for an engine is

$$m^j_i = \frac{1}{P} \sum_{p=1}^{P} y^j_{i,p}. \tag{1}$$

The difference of an engine from its mean value for a stabilized point p is

$$x^j_{i,p} = y^j_{i,p} - m^j_i. \tag{2}$$

Thus given j, $\forall p$ a new variable $X^j_p = (x^j_{1,p}, \dots, x^j_{n,p})^T$ is created. The variable $x^j_{i,p}$ represents the deviation of the engine i at a specific point p compared to its mean value m^j_i to the pair j. Thus, small values of $x^j_{i,p}$ imply small deviations thus normal behavior, meanwhile large values lead to anomalies.

Nine pairs of measurements are defined in an expert manner are: (FNIN1, W2AR), (XN12R, W2AR), (P18QSC, W2AR) are the thrust, the LP spool speed and the pressure at section 18 given the engine corrected air flow. (FNIN1, WF36), (T49C, WF36) are the thrust and the exhausting gaz temperature given the fuel flow. The core speed given the fan speed, the pressure and the temperature at section 3 (HP) are also considered (XN25R, XN12R), (XN25R,T3), (XN25R, P3). Finally, the thrust function of the exhausting gaz is taken into consideration (FNIN1, T49C).

Each pair of variables is observed over four stabilized points, which give 36 new variables. In addition, the four test bench components, CELL, CNOZ, COWL and BMSN are kept in the set of variables that will be used in the model because the pre-treatment was not able to make the data independent from those ones. Finally, the models in following sections are applied on this set of 40 variables.

Note that, to keep an understanding of the new variables obtained from a pair, they are named as follow: "first variable name ___ second variable name ___ stabilized point ". For example, the variable created from T49C and FNIN1 on the fourth stabilized point will be called T49C_FNIN1_6 (the 4 stabilized point level are listed from 3 to 6).

4. ANOMALY DETECTION

4.1. Definition of the Method

Engines, in most of the cases, have solid and adequate measures. Thus, checking for unusual values can be seen as a statistical problem of outlier detection. For our purpose, a statistical method that is both efficient and interpretable is required. Density-based techniques are the most competitives and among these, Isolation Forest in Liu et al. (2008) showed best results on various studies Goix (2016). We employ this method on the new data to detect anomalies.

Isolation Forest is similar in principle to Random Forest Breiman (2001) and is built on the basis of decision trees. It identifies anomalies or outliers rather than profiling normal data points. Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that selected feature. Then, if an observation lies in a high-density region, the probability to isolate it is small because the values of the splits must be very close. On the other hand, if an observation lies in a low-density region, then many values of splits can isolate it, thus it has a higher probability to be isolated by a random split. Random partitioning produces noticeably shorter paths for anomalies. When a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

Let $h_t(x)$ the path length of x in the tree t and $h(x) = \sum_{t=1}^{T} h_t(x)$ the average value of $h(x)$ over the trees with T is the total number of trees in the forest. The number of splits required to isolate an observation is influenced by the number of samples n in the data. To account for this a normalized anomaly score, relying on a property of Binary Search Trees (BST)) Liu et al. (2008), is defined as

$$f(x,n) = 2^{\dfrac{-h(x)}{c(n)}}, \tag{3}$$

with $c(n)$ defined as

$$c(n) = \begin{cases} 2H(n-1) - \dfrac{2n-1}{n} & \text{for } n > 2, \\ 1 & \text{for } n = 2, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where n is the size of data set and H is the harmonic number. The value of $c(n)$ above represents the average of $h(x)$ given n, so we can use it to normalise $h(x)$ and get an estimation of the anomaly score for a given instance x. Note that $f \in [0;1]$ with value closer to 1 indicates that the observation is more likely to be an anomaly.

Once the Isolation Forest has been applied to the data, an anomaly score is estimated for each engine. This anomaly score is pointless if it cannot be completely understood by domain experts.

For complex models, such as ensemble methods, deep networks or Isolation Forest, we cannot use the original model as its own best explanation because it is not easily understandable. Instead, we must use a simpler explanation model, which we define as any interpretable approximation of the original model. We would like to have an average explanation of the model as well as explanation of single prediction and this is called as local explainability Guidotti et al. (2018) which is also known as "post-hoc" explainability. In, Doshi-Velez & Kim (2017) they assert that a useful local explanation should answer the following questions: What were the main factors in the decision? Would changing a certain factor have changed the decision? Why did two similar-looking cases get different decisions, or vice versa? More precisely, we would like to understand how variables contributed to the score of a single engine as well as how variables contributed in average. In addition, understand whether the contribution of a variable have a positive impact or a negative impact on the score, or in other words if a variable helps making an observation more normal or more abnormal. In this aim, Shapley values will be used.

4.2. Model Interpretability with Shapley Values

Shapley values have attracted a great deal of attention in recent years in the field of interpretability, which has been originally discussed in game theory Shapley (1953) and recently applied to statistics Štrumbelj & Kononenko (2014); Owen & Prieur (2017); Iooss & Prieur (2017); Lundberg & Lee (2017). Moreover, in the context of anomaly detection, few results have already been reported Antwarg et al. (2019); Giurgiu & Schumann (2019); Takeishi (2019); Takeishi & Kawahara (2020). These results have confirmed the usefullness of the Shapley value for anomaly interpretation. As described in their works, we will adopt general techniques in defining and computing the Shapley values derived from supervised learning.

Shapley values measure features importance for models in the presence of interaction between variables. This method requires retraining the model on all feature subsets $S \subseteq F$, where F is the set of all features. As described in Lundberg & Lee (2017), it assigns an importance value to each feature that represents the effect on the model prediction of including that feature. To compute this effect, a model $f_{S\cup j}$is trained with that feature present, and another model f_S is trained with the feature excluded. Then, predictions from the two models are compared on the current input $f_{S\cup j}(x_{S\cup j}) - f_S(x_S)$, where x_S represents the values of the input features in the set S. Since the effect of withholding a feature depends on other features in the model, the preceding differences are computed for all possible subsets $S \subseteq F\backslash j$. The Shapley values are

then computed and used as feature attributions. They are a weighted average of all possible differences:

$$\phi_j = \sum_{S \subseteq F \setminus j} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \big[f_{S \cup j}(x_{S \cup j}) - f_S(x_S) \big], \tag{5}$$

At first glance, the above equation seems ridiculously complicated, but it can be easily explained in one sentence: "The contribution of a feature j is the mean difference between a model trained on a subset of variables S with j and a model trained on the same subset S without j, and this is done for all the possible subsets of variables". All possible sets of feature values have to be evaluated with and without the j-th feature to calculate the exact Shapley value. For more than a few features, the exact solution to this problem becomes problematic as the number of possible coalitions exponentially increases as more features are added. In Štrumbelj & Kononenko (2014) an approximation with Monte-Carlo sampling is proposed

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \big(\hat{f}(x^m_{+j}) - \hat{f}(x^m_{-j}) \big), \tag{6}$$

where $\hat{f}(x^m_{+j})$ is the prediction for x, but with a random number of feature values replaced by feature values from a random data point z, except for the respective value of feature j. The x-vector x^m_{-j} is almost identical to x^m_{+j}, but the value x^m_j is also taken from the sampled z. Each of these M new instances is an "artificial object" assembled from two instances. In our case, $\hat{f}(x)$ is the anomaly score predicted by the Isolation Forest. If $\hat{\phi}_j$ is positive the value of the feature j increase the anomaly score, and it decreases if $\hat{\phi}_j$ is negative.

5. RESULTS ON THE DATA

5.1. Average Shapley Values

The Table 1 gives the average Shapley values by variable on all observations. It provides a nice interpretation of the effect of each variable on the anomaly score. A variable j with a ϕ_j value close to 0 will not affect the output of the model and thus the variable is not important to detect anomalies. A high absolute ϕ_j value points out that the variable plays an important role in model estimates. The sign of the contribution gives an additional information on the effect of the variable. A positive contribution denotes that the variable helped to increase the estimated anomaly score w.r.t. the average anomaly score whereas a negative contribution decreases it. Therefore, if in average a variable has a negative ϕ_j, it means that it does not globally contribute to make an engine significantly different from others.

Table 1. Average Shapley value by variables. Positive ϕ_j values indicate that the variable tends to increase the anomaly score while negative one indicates the opposite.

variable	ϕ
WF36_FNIN1_4	1.7839
WF36_T49C_4	1.4149
P3_XN25R_3	1.3275
T49C_FNIN1_6	1.0992
XN12R_XN25R_5	0.8475
WF36_T49C_3	0.7794
WF36_FNIN1_3	0.7782
P3_XN25R_6	0.7207
W2AR_XN12R_3	0.5513
WF36_T49C_6	0.4814
W2AR_XN12R_4	0.4696
XN12R_XN25R_6	0.4378
XN12R_XN25R_4	0.3720
WF36_T49C_5	0.2933
P3_XN25R_5	0.2284
W2AR_XN12R_6	0.1843
WF36_FNIN1_6	0.1367
CNOZ	0.1318
T3_XN25R_4	0.0514
CELL	0.0487
T49C_FNIN1_4	0.0212
WF36_FNIN1_5	-0.0029
W2AR_XN12R_5	-0.0201
XN12R_XN25R_3	-0.0446
P3_XN25R_4	-0.0780
W2AR_FNIN1_6	-0.1006
T49C_FNIN1_5	-0.1676
W2AR_FNIN1_4	-0.2927
COWL	-0.4981
W2AR_P18QSC_5	-0.5035
BMSN	-0.5225
W2AR_P18QSC_3	-0.6202
W2AR_P18QSC_4	-0.6486
W2AR_FNIN1_3	-0.7632
W2AR_FNIN1_5	-0.7816
W2AR_P18QSC_6	-0.7929
T3_XN25R_5	-0.7946
T3_XN25R_6	-0.8460
T49C_FNIN1_3	-1.0517
T3_XN25R_3	-1.1104

5.2. Single Prediction Explanation

Averaged explanation are useful and give insights but they are not sufficient. When an engine has a high estimated anomaly score, domain experts would like to understand which variables are responsible for this score. As an example, the engine 41 is observed. Shapley values are applied to explain how variables contributed to the score. On Figure 3, the average score of anomaly for engines is 0.40 and the engine 41 has an anomaly score of 0.47, which is significantly larger. This difference is decomposed variable by variable. The x-axis, ϕ, gives the weight of the contribution. On the y-axis, the engine values for each variable are displayed, ordered by decreasing importance. A value of 0 on the x-axis indicates that the variable does not play any role in the estimation of the score. Note that, most important variables for this engine correspond to the highest deviations $x^j_{41,p}$ obtained by RPCA. However, due to the possible high-order interaction between variables, a complex model was needed to assess a good estimation of the anomaly score.

Domain experts have access to the contribution of the variables

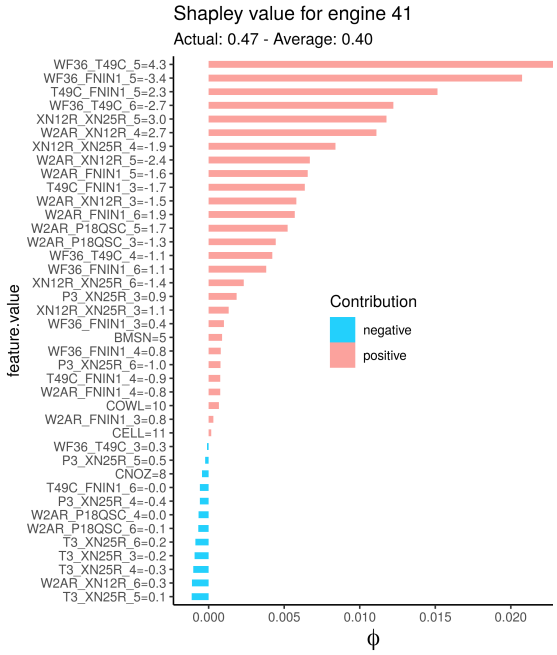

Figure 3. Shapley plot of the engine 41. 0.40 is the average anomaly score and 0.47 is the predicted score of the engine 41. Feature value with positive ϕ increase the score from 0.40 to 0.47, and negative value of ϕ decrease the anomaly score of the engine.

and they can control the validity of the results. Figure 4 shows the densities of variables with the highest (WF36_T49C_5) and the lowest (T3_XN25R_5) ϕ detected for engine 41. As expected, the value $x^j_{41,5}$ for j =WF36_T49C, is far from 0 and isolated in a low density area, which means that it has a really different behavior over the four stabilized points. On the other hand, for j =T3_XN25R, $x^j_{41,5}$ is close to 0 and the engine has a consistent behavior.

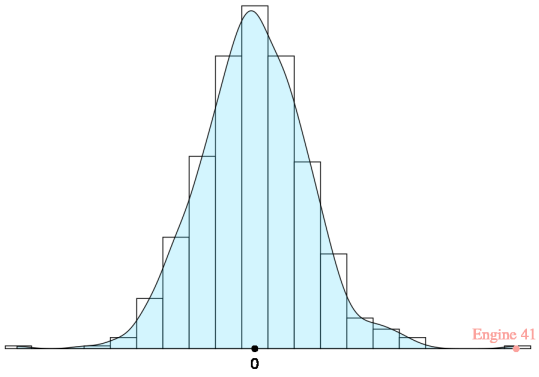

Figure 4. The density of the variable WF36_T49C_5. Engine 41 is isolated in a low density area which explains why this variable have a high contribution to increase the anomaly score.

A diagram that explains the process of engines tests validation

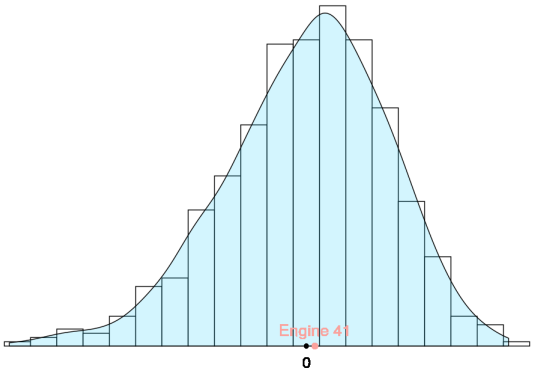

Figure 5. The density of the variable T3_XN25R_5. Engine 41 lies in a high density area which explains why this variable contributes to decrease the anomaly score.

using the statistical methodology is presented in Figure 6. Isolation Forest helps domain experts to identify few engines and Shapley values help them to focus on some specific measures. Then a complete inspection of the engine can be done before validating the production test.

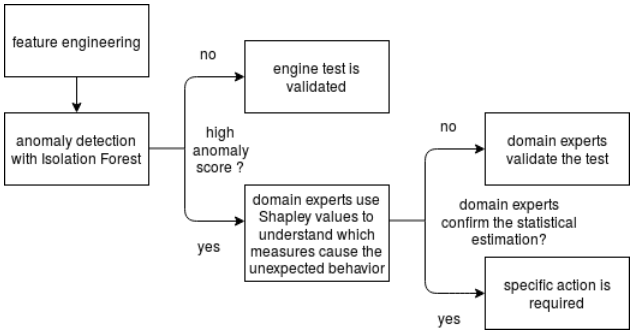

Figure 6. Diagram of the process of production tests validation using Isolation Forest and Shapley values. The statistical methodology helps to highlight specific engines and measures where further analyses are needed.

6. ANOMALY CATEGORIZATION USING SELF-ORGANIZING MAP

6.1. Definition of the Method

A SOM is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) discretized representation of the input space of the training samples, called a map. Each unit of the map corresponds to a prototype vector in the original high-dimensional space, and new data points are projected on the map by finding the closest prototype vector w.r.t. euclidean distance Kohonen (1982); Olteanu & Villa-Vialaneix (2015). Self-organizing maps have been used for aircraft engine fleet monitoring in Cottrell et al. (2009); Côme et al. (2010b,a); Forest et al. (2018) and to classify transient flight phases Faure et al. (2017). No specific study has been yet conducted on

using SOM to validate and categorize anomalies and especially on production tests data.

SOM has both intrinsic distances between clusters and nice two-dimensional visualization, which make it a good candidate. Nonetheless, clusters are still in high-dimensions and methods such as Shapley value are not tractable in this situation. Therefore, in the next section, before modelling a SOM, specific clustering algorithm for variable selection will be used.

6.2. Choose a Subset of Variables with the Help of Group-Sparse Weighted K-means

Group-sparse weighted K-means generalizes the sparse weighted K-means algorithm for numerical variables in Witten & Tibshirani (2010), by using the group regularization framework. Suppose that the numerical matrix of data $\mathbf{X}$ is described by p features that are divided into L priorly known distinct groups, such that $\mathbf{X} = \left[\mathbf{X}^1|...|\mathbf{X}^L\right]$, with $\mathbf{X}^\ell \in \mathbb{R}^{n\times p_\ell}$, p_ℓ being the size of group ℓ, and $p_1 + ... + p_L = p$.

In presence of group data, we would like to discriminate groups of variables $\mathbf{X}^\ell$ by using a specific L_1-group penalty, which has been already used in the regression framework Yuan & Lin (2006). This allows us to select variables by group, forcing the model to select or discriminate the entire group. As described in Chavent et al. (2020), the between-class variance of each variable is multiplied by a weight and a parameter λ penalizes the weights. The latter discriminates the groups of variables with the lowest between-class variance. There is a clustering solution (groups weights and clusters) for each fixed λ. The regularization path (clustering solution given lambda) is computed at a grid of values for the penalty factor λ, covering the entire range, from a model with all the groups included to a model with only one group. The optimization procedure is quite straightforward. The algorithm is optimized in an iterative fashion: first the K-means algorithm is performed on the weighted space of features, then the partition is held fixed and the weights are updated. This iterative procedure is continued until a (local) minimum is reached.

In this context, groups are clearly formed by the variables over the stabilized points. For example, T3_XN25R_3, T3_XN25R_4, T3_XN25R_5 and T3_XN25R_6 belong to the same group. Hence, there is 9 groups of variables, and we would like to know which are the most discriminative for clustering. Moreover, Table 1 shows that test bench components were not significant to detect unusual behavior. Shapley values attribute them in average a negative contribution, which implies that they are not useful to model unexpected behavior. Thus, these variables are not considered in the analysis. Furthermore, the number of clusters is set to five and was found with the Silhouette method Rousseeuw (1987).

In Figure 7 we provide the path of groups' weights against the λ sequence. Weights of groups, on the y-axis, are represented for each λ. The most important groups are 7, 4 and 5 which are respectively the groups P3_XN25R, T49C_FNIN1 and WF36_FNIN1.

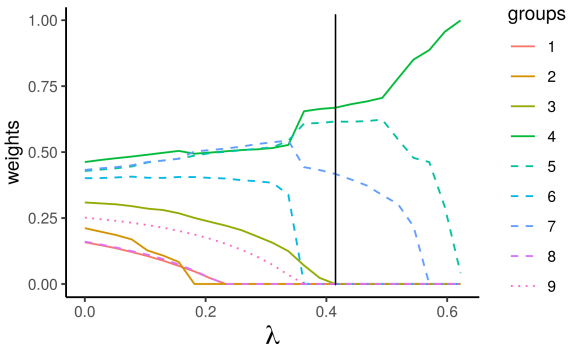

Figure 7. Groups weights for each value of λ. The vertical line represents the selected clustering solution where three groups of variables have non-zero weights.

In Figure 8 we see a big gap in terms of explained variance before $\lambda = 0.6$ If one give a closer look, the analysis points out that a subset of variables (3 groups) will give similar clustering, in terms of explained variance, to the one with all the variables included (see Figure 7 and Figure 8). We choose the clustering obtained for value of λ represented by the vertical line. This value of λ allows to select 3 groups with high explained variance. Higher value of λ lead to have a more similar solution to the one after the gap in Figure 8 in terms of weights, which seems to be a bad clustering solution. Therefore, the chosen value of λ seems to be a good trade-off between interpretability and performance. The weights obtained by groups are $\mathbf{w}_{\text{T49C\_FNIN1}} = 0.67$, $\mathbf{w}_{\text{WF36\_FNIN1}} = 0.62$ and $\mathbf{w}_{\text{P3\_XN25R}} = 0.42$. This subspace of 12 variables will be used to represent the data with the help of the SOM algorithm.

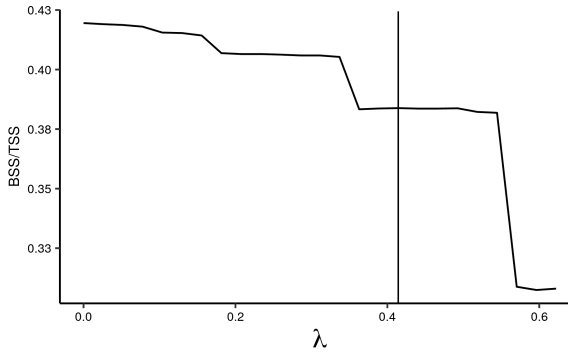

Figure 8. The ratio of between-sum of squares over the total sum of squares (BSS/TSS: explained variance) for each value of λ. The vertical line represents the selected clustering solution which has an explained variance that is close to the one with the full set of variables.

6.3. Data Representation with SOM

The SOM allows us to have access to several different visualizations. First, we plot the distances between prototypes (Figure 9), which gives a representation of the grid where the colors represents the mean distance to the neighbor prototypes. The color scale goes from blue to purple, where purple indicates a large distance. In Figure 10 the repartition of the engines on the map is provided. Finally the Figure 11 is the

Figure 9. Smooth distances between prototypes. The background colors indicate the distances between neighboring prototypes where pink corresponds to larges distances.

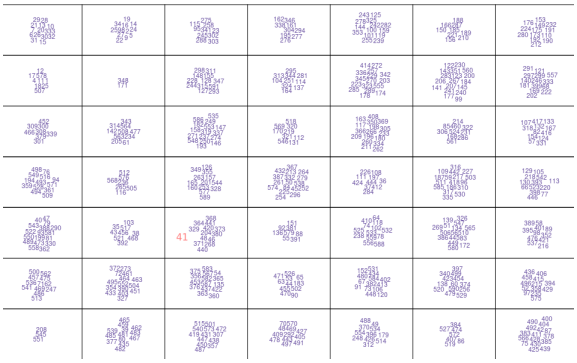

Figure 10. Repartition of the engines on the map.

SOM map where colors indicate the mean anomaly score of engines estimated with the Isolation Forest by clusters. The color scale goes from yellow to red, where red indicates a higher anomaly score. It is interesting to note that the engines with high anomaly score are distributed on the border of the map. The representation is very similar to the one given by the smooth distances between prototypes on Figure 9 which shows that the two methods agree. In addition, the map allows a categorization of the anomalies. In Figure 11, Super-clusters are identified thanks to the mean anomaly score of the prototypes and their proximities on the SOM map. Two map edges are thus identified as super-clusters. The comparison of these clusters of anomalies with the rest of the population will allow us to understand the discriminative variables.

The ANOVA method is applied to test significant difference between clusters (Figure 12). ANOVA provides a statistical

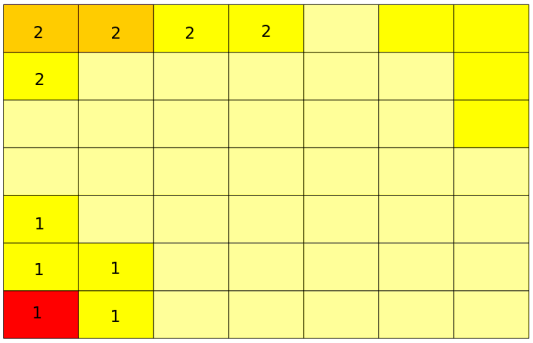

Figure 11. SOM map of engines where the colors represents the anomaly score estimated with Isolation Forest averaged by clusters. Color scale goes from yellow to red, where red corresponds to a higher score.

test of whether two or more population means are equal. The results show that, for cluster 2, the group of variables P3_XN25R seems to be important. On the other hand, for cluster 1, the group of variables WF36_FNIN1 may explain their unusual behavior. The ANOVA shows that the set of explaining variables has been reduced to only one group. For the sake of readability, we show the ANOVA test for only two variables at the stabilized point 3, but for the other stabilized points results are similar since they are all linked by construction. Moreover, the other groups of variables are not significantly different over the three clusters.

7. CONCLUSIONS

In this work, statistical methods in the context of rare event detection in production tests data demonstrated high degree of efficiency and interpretability in either local (one specific engine) or global level (groups of engines). We propose a multi-scale model, giving a hierarchy in the information allowing the experts to better understand flaws on a particular engine but also allowing them to detect more general problems. We apply two different methods: i) Isolation Forest to estimate anomaliy score and Shapley values to interpretate them; ii) SOM on a subset of variables obtained by group-sparse weighted K-means. Both methods provide similar results: the engines detected as anomalies with the Isolation Forest coincide to the engines that have the largest distances estimated with SOM.

Moreover, an other contribution of this work is the use of SOM to validate anomalies detection methods. On the contrary of Isolation Forest, SOM provides visualizations and categorizations of the anomalies which gives additional information to better understand and verify the estimated anomalies.

Explainability in unsupervised learning is a new field that needs to be explored, and this work is a step forward in this direction. Some further investigations are needed in both theoretical and applied domains. In future works, we plan to explore those points and also we will develop a method to

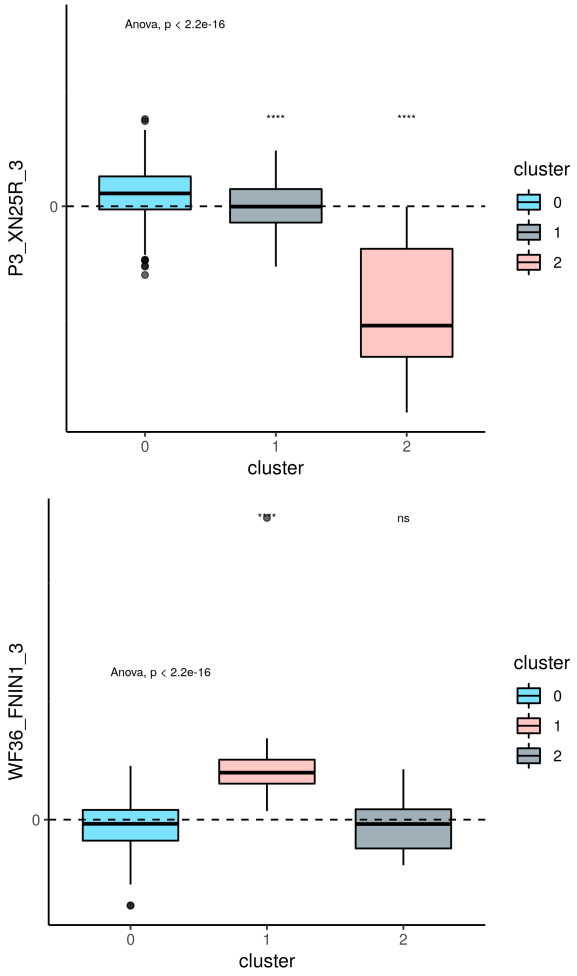

Figure 12. Boxplot of ANOVA test comparing two different area of the map Figure 9. For the two variables considered, only one cluster (the pink one) is truly different from the overall population of engines, implying that the two clusters of anomalies can be explained by different subsets of variables.

transform the anomaly score in a binary score which will help to domain experts in their decisions.

8. ACKNOWLEDGEMENT

This work was supported by the French National Agency for Research and Technology (ANRT) and Safran Aircraft Engines (Safran group).

REFERENCES

Antwarg, L., Shapira, B., & Rokach, L. (2019). Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, *58*(3), 1–37.

Chavent, M., Lacaille, J., Mourer, A., & Olteanu, M. (2020). Sparse k-means for mixed data via group-sparse clustering. *ESANN*.

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010a). Aircraft engine health monitoring using self-organizing maps. In *Industrial conference on data mining* (pp. 405–417).

Côme, E., Cottrell, M., Verleysen, M., & Lacaille, J. (2010b). Self organizing star (sos) for health monitoring..

Cottrell, M., Gaubert, P., Eloy, C., François, D., Hallaux, G., Lacaille, J., & Verleysen, M. (2009). Fault prediction in aircraft engines using self-organizing maps. In *International workshop on self-organizing maps* (pp. 37–44).

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Faure, C., Olteanu, M., Bardet, J.-M., & Lacaille, J. (2017). Using self-organizing maps for clustering anc labelling aircraft engine data phases. In *2017 12th international workshop on self-organizing maps and learning vector quantization, clustering and data visualization (wsom)* (pp. 1–8).

Forest, F., Lacaille, J., Lebbah, M., & Azzag, H. (2018). A generic and scalable pipeline for large-scale analytics of continuous aircraft engine data. In *2018 ieee international conference on big data (big data)* (pp. 1918–1924).

Giurgiu, I., & Schumann, A. (2019). Additive explanations for anomalies detected from multivariate temporal data. In *Proceedings of the 28th acm international conference on information and knowledge management* (pp. 2245–2248).

Goix, N. (2016). *Apprentissage automatique et extrêmes pour la détection d'anomalies* (Unpublished doctoral dissertation). Paris, ENST.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42.

Iooss, B., & Prieur, C. (2017). Shapley effects for sensitivity analysis with dependent inputs: comparisons with sobol'indices, numerical estimation and applications.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, *43*(1), 59–69.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422).

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Olteanu, M., & Villa-Vialaneix, N. (2015). On-line relational and multiple relational som. *Neurocomputing*, *147*, 15–30.

Owen, A. B., & Prieur, C. (2017). On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, *5*(1), 986–1002.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(C), 53–65. doi: 10.1016/0377-0427(87)90125-7

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, *2*(28), 307–317.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, *41*(3), 647–665.

Takeishi, N. (2019). Shapley values of reconstruction errors of pca for explaining anomaly detection. In *2019 international conference on data mining workshops (icdmw)* (pp. 793–798).

Takeishi, N., & Kawahara, Y. (2020). On anomaly interpretation via shapley values. *arXiv preprint arXiv:2004.04464*.

Witten, D. M., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, *105*(490), 713-726. Retrieved from `https://doi.org/10.1198/jasa.2010.tm09415` (PMID: 20811510) doi: 10.1198/jasa.2010.tm09415

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.