



Anomaly detection for aircraft electrical generator using machine learning in a functional data framework

Feriel Boulfani, Xavier Gendreau, Anne Ruiz-Gazen, Martina Salvignol

► To cite this version:

Feriel Boulfani, Xavier Gendreau, Anne Ruiz-Gazen, Martina Salvignol. Anomaly detection for aircraft electrical generator using machine learning in a functional data framework. GMC-ElecEng2020, Sep 2020, Valence, Spain. hal-02969954

HAL Id: hal-02969954

<https://hal.science/hal-02969954>

Submitted on 19 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Anomaly detection for aircraft electrical generator using machine learning in a functional data framework

1 st Ferial Boulfani	2 nd Xavier Gendre	3 rd Anne Ruiz-Gazen	4 th Martina Salvignol
<i>Toulouse School of Economics</i>	<i>ISAE-SUPAERO</i>	<i>Toulouse School of Economics</i>	<i>Electrical design department</i>
<i>Université de Toulouse</i>	<i>Université de Toulouse</i>	<i>Université de Toulouse</i>	<i>Airbus Operations S.A.S</i>
<i>Toulouse, France</i>	<i>Toulouse, France</i>	<i>Toulouse, France</i>	<i>Toulouse, France</i>
<i>feriel.boulfani@airbus.com</i>	<i>xavier.gendre@isae-supero.fr</i>	<i>anne.ruiz-gazen@tse-fr.eu</i>	
			

Abstract

To reduce the number of aircrafts on ground, the electrical design engineers are interested in predicting the oil temperature of the generator during a flight. Changes on the temperature value may indicate an incorrect functioning of the generator. An abnormal behavior can be identified by using machine learning algorithms trained on flights free from any anomalies to predict the generator oil temperature. These predictions can then be compared to the observed values, here the sensor data collected from the aircraft during flight. If the observed value is far from the predicted value, a failure warning is raised and a maintenance action shall be performed.

In this paper, we build a digital twin of the electrical generator which predicts the oil generator temperature at a given time thanks to the history of features. We compare several machine learning algorithms and the most promising appears to be a neural network which was implemented as part of the electrical generator digital twin.

The digital twin is tested by using real flight data containing generator failures and it is verified that the algorithm is able to detect an anomaly prior to the failure events (early failure detection).

Keywords

machine learning; failure detection; generator oil temperature prediction; digital twin; health monitoring;

1. Introduction

Several electrical generator failures can lead to a No-Go case, meaning that the aircraft is not allowed to take-off until the failure is fixed. The aircraft has the status of “Aircraft On the Ground” (AOG). The electrical design engineers want to reduce this cost by detecting the abnormal behavior before it turns into AOG and suggest to perform a maintenance action on the generator. Instead of doing a periodic maintenance inspection (preventive maintenance)

that immobilizes the aircraft, inspection is performed only when required (predictive maintenance).

The oil circuit cools the generator and its temperature may be used as a measure of proper functioning of the generator. An overheating or a very low oil temperature may indicate a generator anomaly. To monitor the health of the generator (health monitoring), a virtual model that distinguishes anomalies from normal behavior is built. This model can be seen as a Digital Twin (DT) that describes the normal behavior of the electrical generator based on the generator oil temperature. The oil temperature data recorded during the flight represents the physical model that describes the real health of the generator, while the predicted oil temperature represents the virtual model thus the DT.

The idea is to train the DT algorithm using only flights free of anomalies. The prediction of the oil temperature will diverge from the real value when the generator behaves abnormally. Figure 1 shows how the DT operates to detect an abnormal behavior on simulated flights. The plot of Figure 1 gives the tracking of the prediction error per flight. The flights are ordered in time, the flight number 50 corresponds to a generator failure. We see that the prediction error starts to diverge 7 flights before the generator failure. An anomaly warning can be raised after the flight number 45 to launch a second DT that investigates the anomaly cause and to select the appropriate maintenance actions. In this paper we will focus on the anomaly detection DT only.

The DT concept is widely used in the area of industrial systems maintenance as it saves costs by performing the maintenance only when a warning is raised. A recent state of the art and gap analysis is performed by [1] where various applications of the DT concept and its implementation for the maintenance in industrial manufacturing are presented. The authors give the two main reasons why the DT is used which are design validation and product mastering. Our DT model is based on product mastering where the product is the electrical generator.

The paper [17] referenced by [1] is related to the DT concept in the aeronautic field. The DT built by the authors concerns the design and the maintenance of the U.S. Air Force aircraft. The authors used a sub-model for different

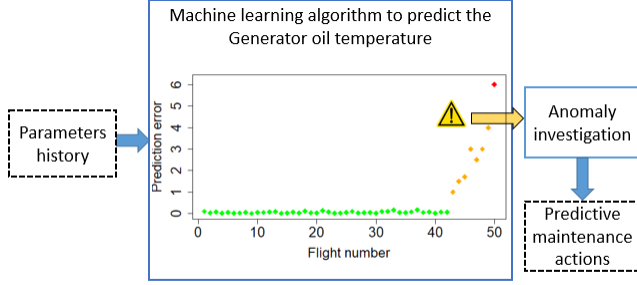


Figure 1: Predictive maintenance using DT for a simulated flights with anomaly

sub-systems and statistical methods to estimate uncertainty of each sub-model. Moreover in [13] and [14], the authors compute the remaining useful life of a system using several methodologies based on the survival analysis. On top of that we add the recent work done by [2] to design and monitor the ice protection system under IT infrastructure constraints. Our approach differs from the ones mentioned above. To build our DT we decided not to use any physical equations that describe the behaviour of the electrical generator, but to rely only on data measured in service. In fact, the electrical equipments in aeronautics become more and more complex, which makes it difficult to build a reliable generator model based on physical knowledge. The aim of this DT is to allow system designers to identify failures by combining their high level knowledge of the electrical system equipment and the data coming from the aircraft.

The novelty of our approach is to build a DT based on a machine learning algorithm that uses functional data to detect anomalies. We choose a functional data framework to predict the oil temperature at time T thanks to the T last records of selected features. This choice is motivated by the high time dependency between the recorded values and is adapted to capture the history of the features. Anomaly detection in the functional data framework is a quite recent research field. Papers have been published recently in the aerospace area (see [4] and [12]).

The Functional Data Analysis (FDA) encompasses the analysis and theory for functional data [10]. Such data include data recorded during a time interval. In this paper we use FDA tools to reduce the dimension by expanding our functions onto orthonormal bases and keep the first coefficients of each function. By reducing the dimension we reduce also the model complexity, but decrease the model accuracy. A trade off between complexity and accuracy needs to be done. The commonly used machine-learning procedures detailed in [9] were applied to the coefficients to predict the oil temperature. The procedure that has the smallest prediction error is selected and implemented in the DT.

In this paper we explain the functional data representation in Section II, we recall the prediction procedures and the model selection in Section III. Finally, we test the DT on real flights containing a failure event in Section IV. A perspective of

Table 1: Recorded features by flight at each second used to predict the generator oil temperature

Description	Unit	(Min, Mean, Max)
Generator oil temperature	C°	(38, 72, 136)
Engine speed	Knot (kts)	(24, 81, 95)
Static air temperature	C°	(-72, -29, 37)
Total air temperature	C°	(-46, -7, 41)
Computed air speed	kts	(116, 253, 345)
Altitude	ft	(151, 27140, 39070)
Generator load	KVA	(20, 26, 51)

this work is given in the concluding section.

II. Functional data

We consider 6 aircrafts and we sample $N = 606$ flights of distinct lengths. These flights were checked to be free of anomalies. For each flight, we observe the features defined in Table 1. The sampling rate of the records is one record per second.

For a given flight $\ell \in \{1, \dots, N\}$, we have n_ℓ observations of $(y, X^1, \dots, X^q)$ where y stands for the oil temperature and $X^1, \dots, X^q$ are the observed values taken by the features. The goal is to predict the oil temperature at a given time thanks to the observations of $X^1, \dots, X^q$ during the T last seconds. For that, we need to split each flight into segments of length T , where $T < \min\{n_1, \dots, n_N\}$. Then, for a flight $\ell \in \{1, \dots, N\}$ we obtain τ_ℓ segments where $\tau_\ell = \lfloor n_\ell / T \rfloor$ and $\lfloor \cdot \rfloor$ stands for the floor part function. Then for a given flight ℓ , the resulting segmented flight is given by a matrix

$$\begin{bmatrix} \dots & X^j(1) & \dots & X^j(T) & \dots \\ \dots & X^j(T+1) & \dots & X^j(2T) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & X^j(\tau_\ell T + 1) & \dots & X^j(\tau_\ell T) & \dots \end{bmatrix}$$

with τ_ℓ rows and qT columns. The output vector is defined by $y = (y(T), y(2T), \dots, y(\tau_\ell T))$. Figure 2 shows a visual example of the engine speed feature segmentation.

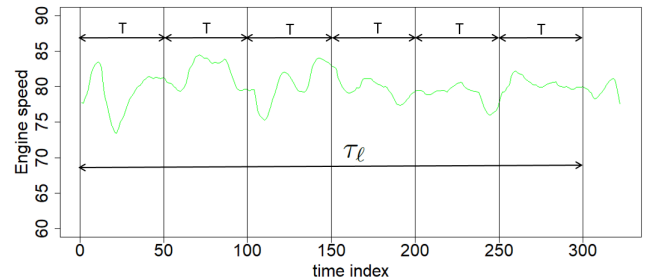


Figure 2: Split of engine speed feature into τ_ℓ segments of length T for the flight ℓ

Associated to the discrete observations of variables $X^1, \dots, X^q$, we consider, for any $j \in \{1, \dots, q\}$ and $k \in \{1, \dots, \tau_\ell\}$, the functions

$$x_k^j : [0, 1] \longrightarrow \mathbb{R} \\ t_i \longmapsto x_k^j(t_i) = X^j((k-1)T + i)$$

with $t_i = i/T$, $i \in \{1, \dots, T\}$.

As shown in Table 1 we are handling features with different units and scales. To normalize the data, we center and scale the functions. For a given function x^j , $j \in \{1, \dots, q\}$, we introduce its average function $\bar{x}^j$ defined by

$$\bar{x}^j(t) = \frac{1}{n} \sum_{i=1}^n x_i^j(t), \quad t \in \{1, \dots, T\}$$

where n is the row number of the segmented matrix for all flights ($n = \tau_1 + \dots + \tau_N$). The squared distance between x^j and the average $\bar{x}^j$ is given by

$$\|x_i^j - \bar{x}^j\|_T^2 = \frac{1}{T} \sum_{t=1}^T \left(x_i^j(t) - \bar{x}^j(t) \right)^2, \quad i \in \{1, \dots, n\}.$$

Thus, its variance is equal to

$$\text{Var}(x^j) = \frac{1}{n} \sum_{i=1}^n \|x_i^j - \bar{x}^j\|_T^2.$$

In what follows, we consider all functions centred and scaled as given by

$$\frac{x^j - \bar{x}^j}{\sqrt{\text{Var}(x^j)}}, \quad j \in \{1, \dots, q\}.$$

Let $\{\xi_d\}_{d \in \mathbb{N}}$ be an orthonormal basis of the space $L^2 = L^2([0, 1], dt)$ of square integrable functions on $[0, 1]$ with respect to Lebesgue measure dt . We consider the usual inner product,

$$\forall f, g \in L^2, \langle f, g \rangle_{L^2} = \int_0^1 f(t) g(t) dt.$$

The expansion of a function $f \in L^2$ according to the orthonormal basis is given by

$$\forall t \in [0, 1], f(t) = \sum_{d \in \mathbb{N}} c_d \xi_d(t), \quad \text{with } c_d = \langle f, \xi_d \rangle_{L^2}.$$

In our application, only discretizations of our auxiliary functions are observed. The c_d are estimated on a regular grid of step $1/T$ in $[0, 1]$ by

$$\hat{c}_d = \frac{1}{T} \sum_{i=1}^T f\left(\frac{i}{T}\right) \xi_d\left(\frac{i}{T}\right).$$

There are many ways to represent functional data. In this paper we consider two common ones, Fourier and Haar wavelet bases [10]. To reduce the dimension, we plan to handle a truncation by selecting the first $D \in \mathbb{N}$ coefficients for the features $X^1, \dots, X^q$. This means that we have the same number of coefficients D per auxiliary function.

On the matrix C of the coefficients $\hat{c}_d$, $d \in \{1, \dots, D\}$, we apply machine learning procedures for regression prediction based on Dq variables instead of Tq with $D \ll T$.

To satisfy orthonormality constraints and to stick to industrial practices, the length T is set to $256 = 2^8$ seconds in the sequel.

III. Prediction procedures

In this section we discuss the statistical procedures that we consider to predict the generator oil temperature through the regression model given by $Y = f(C) + \epsilon$ where $Y = (y_1, \dots, y_n) \in \mathbb{R}^n$ is the observed vector, f is the prediction procedure, $\epsilon \in \mathbb{R}^n$ is an error vector and C is a $n \times Dq$ coefficients matrix.

We compare the accuracy of neural network (NN), ridge regression (RR) and random forest (RF) procedures using the mean squared error (MSE) criterion given by

$$\text{MSE}(\hat{f}) = \frac{1}{n} \|Y - \hat{f}(C)\|^2$$

where $\|\cdot\|^2$ is the ℓ_2 -norm defined by $\|x\|^2 = \sum_{i=1}^n x_i^2$. Hereafter we recall the regression prediction procedures that we use:

- **Ridge regression** [8, Chap 3] assumes a linear relationship between the input matrix C and the output Y with the model $Y = C\beta + \epsilon$, where $\beta = (\beta_0, \beta_1, \dots, \beta_{Dq})^t \in \mathbb{R}^{Dq+1}$. The ridge regression regularizes the MSE by adding a penalty term to the MSE and the parameters β are obtained as

$$\hat{\beta} = \arg \min_{\beta} \left\{ \|Y - C\beta\|^2 + \lambda \|\beta\|^2 \right\},$$

where $\lambda > 0$ is the shrinkage parameter. The local minimum solution is given by

$$\hat{\beta} = (X^t X + \lambda I)^{-1} X^t Y.$$

- **Neural network** [5;8, Chap 11 is a collection of neurons connected together and organized in layers to predict the output Y . There are at least 3 layers: input, hidden and output. Each connection between two neurons has a weight. The output is a function of the weighted input features]

$$g(\beta_0 + \sum_{j=1}^{Dq} \beta_j c_i^j), \quad i = 1, \dots, n,$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_{Dq}) \in \mathbb{R}^{Dq+1}$ is a weight vector and $g : \mathbb{R} \rightarrow \mathbb{R}$ is called the activation function which quantifies the activation status of the neuron. Several activation functions can be considered, the most widely used are the rectified the linear unit, the sigmoid and the identity (see [18] for more details).

We use a multi-layer perceptron proposed by [15] with one hidden layer that contains $\lambda \in \mathbb{N}^*$ number of neurons and the Sigmoid activation function is applied to the hidden layer. For the output layer we use the identity activation function. The weight matrix is trained using the backpropagation learning.

- **Random forest** [8, Chap 15;16 is an aggregation of a multitude of decision trees. Each decision tree

is built according to the CART approach. Such an approach is obtained by recursively splitting the data space and fitting a simple prediction model within each partition. To reduce the variance of the CART algorithm, individual trees are combined by using bagging (bootstrap aggregation) together with features selection. The idea of bagging is to draw m new training sets with replacement from X and Y . These samples are known as bootstrap samples. A decision tree is built for each bootstrap. The output variable is fitted by averaging the m predictions from all the individual decision trees.]

Random forests improve the bagging technique by adding randomness in the choice of the features during the learning process. For each bootstrap sample, the decision tree is built by selecting $\lambda \in \mathbb{N}^*$ features draw randomly from Dq features at each splitting iteration. In the sequel, the number of trees to grow is set to $m = 20$.

The hyperparameter λ of each procedure needs to be calibrated. To this end, we use a common approach known as 10-fold cross-validation [9, Chap 5].

To compare our prediction procedures, we need first to set the number D of selected coefficients for the features. The choice of D is crucial, because the higher D the higher the model complexity. Thus, we need to balance between model performance and model complexity to find a tradeoff. Moreover, D is also limited by the features history length T ($D \ll T$).

To set the optimal D , we train the selected procedures with 85% of random flights and keep the 15% to test the accuracy of the procedures. In Figure 3, we plot the link between the model performance MSE and the model complexity which is equal to Dq using Fourier basis coefficients.

We differentiate the prediction models RR, NN and RF by respectively solid, dashed and dotted lines. This comparison shows that the RF is the most accurate model followed by NN and RR models. Similar results are observed for Haar basis and are not reported in the present paper.

For all models and both bases, the MSE increases for a complexity higher than 200. To keep Haar and Fourier basis comparable, we set $D = 31$ and have a complexity $Dq = 186$. By using Fourier and Haar bases expansion we reduce our model from 1 536 to 186 parameters to estimate. Table 2 gives the results of the 10-fold cross-validation MSE for $T = 256$, $D = 31$ in order to compare the procedures accuracy. For both bases we obtain similar cross-validation MSE for all procedures. Once again, the RF approach is the best method. In what follows we focus on the NN and RF models and use the Fourier basis only.

To compare NN and RF models performance we plot in Figures 4 and 5 the average MSE per flight respectively for the NN and RF model. The flights are separated by aircraft using vertical lines. The ones used in the training are represented by diamonds and those used in the test by crosses. Horizontal lines (distance mark) are added at the MSE (solid line), $2 \times$ MSE (dotted line) and $4 \times$ MSE (dashed line) values on both figures.

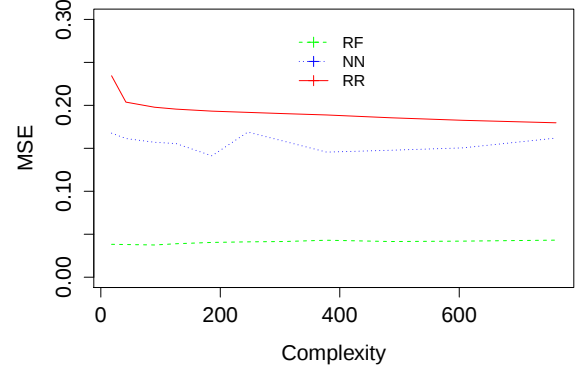


Figure 3: Prediction accuracy versus model complexity for Fourier basis

Table 2: Cross-validation error for RR, NN and RF by basis

MSE	Fourier basis	Haar basis
RR	0.19	0.20
RF	0.04	0.04
NN	0.10	0.13

The MSE per flight for the NN are more centred around the global MSE than the RF. It is noticeable that the RF model suffers from overfitting as the MSE of the test flights are far from the training set. In what follows, we then keep the NN model and Fourier basis to detect the anomalies.

From Table 2, we define a reference $MSE_{ref} = 0.10$ for NN model using Fourier basis. This value is used in the next section as reference to detect anomalies.

IV. Anomaly detection using digital twin

To validate that our DT can detect anomalies we need to test it with flights that contain a real anomaly. Unfortunately

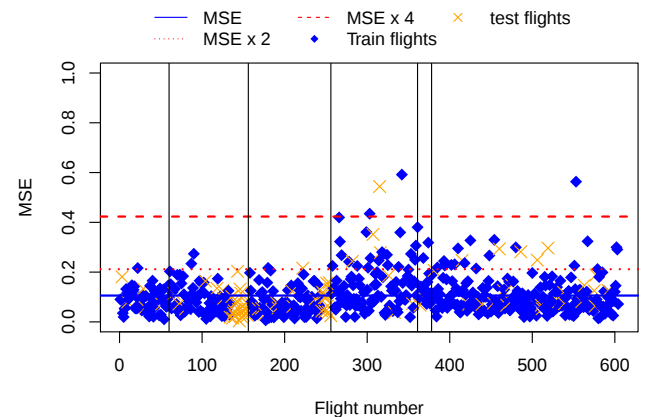


Figure 4: Prediction of the generator oil temperature for the flights test using the NN model and Fourier basis

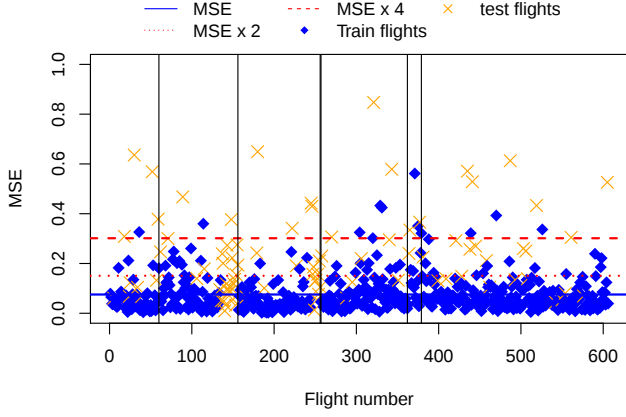


Figure 5: Prediction of the generator oil temperature for the flights test using the RF model and Fourier basis

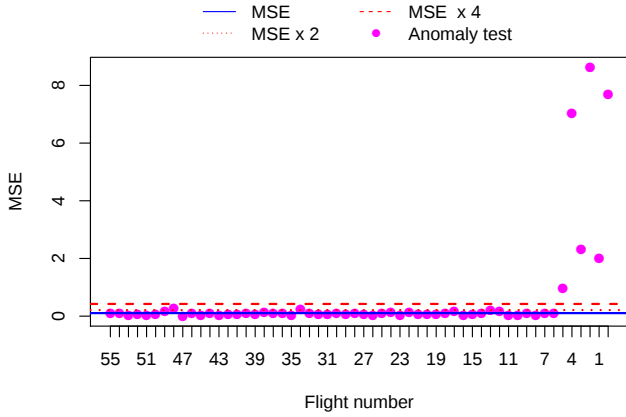


Figure 6: MSE by flight to detect anomaly of case 1

the anomalies are not easy to identify or to label, and failure occurrences are very rare. We identified two cases of generator failures which represent a loss of one generator in cruise phase in our data set.

For each case, the DT was tested on the flights preceding the generator loss and the results are presented in Figures 6 and 7. The distance markers used in Section III are kept and the flights preceding the generator loss are presented by circles. The flights are numbered using a countdown to the failure, thus the flight number 0 represents the failure.

In case 1 (Figure 6), the DT is tested on 55 flights preceding the generator loss. The DT starts to detect anomalies 5 flights before it turns into a failure. In Figure 8 we put all the flights used on this aircraft to benchmark the anomaly test flights and the training flights. It shows that we are predicting well normal behavior and abnormal behavior is identified by an important divergence between prediction and real values.

For the second case (Figure 7), we have less flights available but the DT is able to detect the failure 9 flights before the failure event.

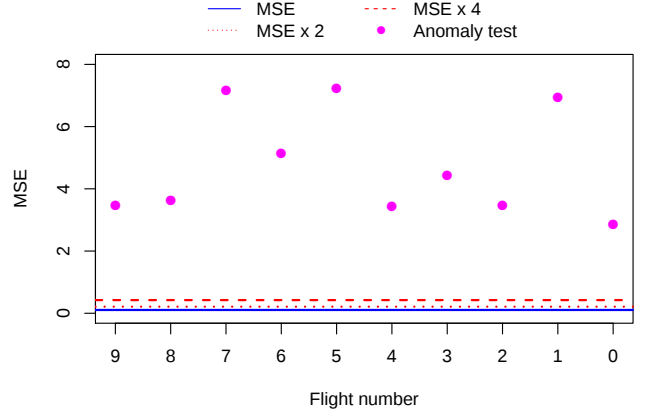


Figure 7: MSE by flight to detect anomaly of case 2

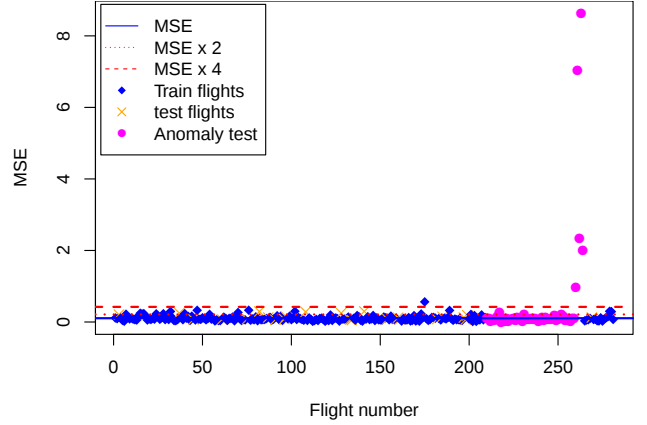


Figure 8: Anomaly detection for case 1 using all available data sets (train, test and anomaly flight)

In these two cases we are facing a huge gap between the MSE of abnormal flights and the distance markers. To identify an anomaly profile, we propose to define a limit defined as a ratio of the MSE of an observed flight over MSE_{ref} . If this limit is exceeded several times in consecutive flights, a warning is raised.

The limit needs to be chosen carefully, as a too high limit may lead to a DT that misses a lot of anomalies and a too low limit to a DT that may detect a lot of false anomalies. From the 670 flights with a normal behavior used in this DT the maximum observed MSE by flight is 0.6, thus a limit of 1 is reasonable which represents 10 times MSE_{ref} .

V. Conclusion

In this paper we present a way to detect generator anomalies before they turn into failures. We build a digital twin model based on the predicted generator oil temperature value. We use the functional data theory and machine learning procedures to predict the oil temperature thanks to the history of

features. The procedures are compared and the best one is selected as part of the digital twin.

We test the DT built on two cases with anomalies and it shows that the DT is able to detect anomalies several flights before it turns into failure. The amount of data available for this study is not enough to conclude on which different failure types can be detected by this method. However, as AGOs are very costly for airlines and aircraft manufacturers, the here proposed DT has an interest even if it allows to predict a few percentage of failure events.

The main challenges to build this DT was to label the flights correctly to normal and abnormal flights. To improve the anomaly profile, we propose to increase the number of flights with anomalies and to design a limit that will take into account the history of the detected anomalies.

A perspective of this work is to reinforce the learning by injecting the normal flights into the DT and the abnormal flights detected to the anomaly profile to update the limit. The DT could also be adjusted to the real behavior model by adding the generator components remaining lifetime to raise a warning. The warning should raise the probability and the remaining time to get a failure.

The concept of our DT can be adapted to other fields and especially for the industrial maintenance (see [6], [3], [7] and [11]). The most important is to identify the features that are correlated with the process under study and have a labeled dataset with abnormal and normal behaviors. Taking into account the dataset free of anomalies, we are then able to train the semi-supervised learning algorithms.

Acknowledgment

This work has been partly supported by the French *Agence Nationale de la Recherche* through CIFRE contract 2017/1354 and through the Investments for the Future (Investissements d’Avenir) program, grant ANR-17-EURE-0010.

References

- [1] P. Aivaliotis, K. Georgoulas, and K. Alexopoulos. Using digital twin for maintenance applications in manufacturing: State of the art and gap analysis. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–5. IEEE, 2019.
- [2] G. Bachelor, E. Brusa, D. Ferretto, and A. Mitschke. Model-based design of complex aeronautical systems through digital twin and thread concepts. *IEEE Systems Journal*, 2019.
- [3] X. Baril, O. Coustie, J. Mothe, and O. Teste. Application performance anomaly detection with lstm ontemporal irregularities in logs. *ACM International Conference on Information and Knowledge Management (CIKM)*, 2020.
- [4] C. Barreyre. *Statistiques en grande dimension pour la détection d’anomalies dans les données fonctionnelles issues des satellites*. PhD thesis, 2018.
- [5] C. N. Bergmeir, J. M. Benítez Sánchez, et al. Neural networks in r using the stuttgart neural network simulator: Rsnns. American Statistical Association, 2012.
- [6] E. Chu, D. Gorinevsky, and S. Boyd. Detecting aircraft performance anomalies from cruise flight data. In *AIAA Infotech@ Aerospace 2010*, page 3307. 2010.
- [7] M. Flores, S. Naya, R. Fernández-Casal, S. Zaragoza, P. Raña, and J. Tarrío-Saavedra. Constructing a control chart using functional data. *Mathematics*, 8(1):58, 2020.
- [8] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA., 2001.
- [9] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
- [10] Jim, J. Ramsay, and B. Silverman. *Functional Data Analysis*. Springer Science & Business Media, 2005.
- [11] B. Kroll, D. Schaffranek, S. Schriegel, and O. Niggemann. System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–7. IEEE, 2014.
- [12] C. Lejeune, J. Mothe, and O. Teste. Outlier detection in multivariate functional data based on a geometric aggregation. In *EDBT*, pages 383–386, 2020.
- [13] P. Papachatzakis, N. Papakostas, and G. Chrysosouris. Condition based operational risk assessment an innovative approach to improve fleet and aircraft operability: Maintenance planning. In *1st European Air and Space Conference, Berlin, Germany*, pages 121–126, 2007.
- [14] V. Quintana, L. Rivest, R. Pellerin, F. Venne, and F. Kheddouci. Will model-based definition replace engineering drawings throughout the product lifecycle? a global perspective from aerospace industry. *Computers in industry*, 61(5):497–508, 2010.
- [15] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [16] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.
- [17] E. Tuegel. The airframe digital twin: some challenges to realization. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, page 1812, 2012.
- [18] B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.