Corrosion Risk Estimation and Cause Analysis on Turbine Stator Blades of Turbofan Engine

Jérôme Lacaille[1] and Raphaël Langhendries[2]

[1]*Safran Aircraft Engines, Moissy-Cramayel, 77550, France*
jerome.lacaille@safrangroup.com

[2]*Université Paris 1 Panthéon-Sorbonne, Paris, 75013, France*
r.langhendries@gmail.com

ABSTRACT

Turbofan engine parts are subject to corrosion. This degradation is very difficult to observe outside of engine dismantling on predefined inspection dates. Moreover, it is necessary to specifically observe the impacted parts to notice this wear. This inspection process is long, expensive, and rarely carried out with the aim of detecting a corrosion effect. The provision of an indicator to alert on potential corrosion and to target parts is highly anticipated. Better still, if the origin of this corrosion can be identified by numerical methods, it will then be possible to improve the calculation of the lifespan of the part and to space out engine inspections. We propose a digital tool able to offer an indicator estimating if an engine part is corroded. In addition to the alert subject to this estimate, we also give a graphical means to interpret this detection. This application is intended for personnel involved in monitoring engines in operation (in-service support, airlines) or maintenance workshops. It not only alerts on degradation but also helps to optimize the calculation of engine inter-inspection intervals and even possibly helps to design new materials or coatings.

1. INTRODUCTION

1.1. Inputs

Currently, engines are inspected by endoscopy (BSI or BoroScope Inspection) only at the entrance to Shop Visit (SV), or under wing according to the aircraft maintenance manual (repetitive inspection, post-event, etc.). In all cases, a BSI inspection (images or video) requires an interpretation of the image, highly dependent on its quality (lens, light, zoom level, sensitivity of the inspector, etc.). The current means of detection are limited because they are mainly linked to planned inspections on each engine (SV or under wing). The appearance of corrosion on a part does not affect the evolution of engine parameters. It is therefore not a detectable phenomenon via conventional means of monitoring engine performance.

Our proposal is to define a probabilistic indicator which calculates a risk of corrosion for each of the engine parts monitored. In addition to the indicator, a graphic tool is used to interpret the alerts by presenting the potential causes.

The probabilistic indicator is constructed in two stages:

- A neural network calculates a rate of exposure to a corrosive environment from engine information (temperatures, pressures, speeds of rotation, etc.), flight information (duration of the journey, rate of climb, etc.), meteorological information on departure and arrival airports, information on pollution...
- The second element is a lifespan model that estimates a time before the onset of corrosion using a survival law and from the cumulative information of exposure rates.

The parameters of the neural network and the parameters of the survival law (Weibull in our case) modeling the fatigue of the monitored metallic components are learned simultaneously from a history of past information (Langhendries and Lacaille 2020).

The interpretation of the results is obtained using a self-organizing classification (Kohonen map) (Kohonen 1988). This map categorizes aircraft uses and displays the instantaneous corrosion rate accordingly. The observation of the trajectory of an aircraft, flight after flight on this map gives a clear interpretation of the risk of corrosion announced. We deduce the missions or the sequence of events that led to the suggested state.

2. DATA MANAGEMENT

The data we need to ensure our ability to track stator engine blades corrosion state comes from a variety of sources. One of the challenges was to reconcile those elements.

2.1. Inputs

There was information emitted by the engine itself, they are called ACARS (Aircraft Communication Addressing and Reporting System) snapshots. These data consist of different measurements taken at certain predefined times during each flight (see Table 1). It may also contain online calculations implemented by our PHM teams. For example, we have one takeoff snapshot, one climb snapshot, up to eight cruise snapshots, one descent snapshot and one post-flight snapshot. Most of the specific online PHM computation are stored in the post-flight snapshot, among them we retrieve vibration power spectrum values, some durations under specific conditions and other calculations based on engine physics. Each snapshot is broadcasted in real time and received via satellite channel. Then Safran stores each document in a large cluster of computers. One problem was that since an aircraft and not the engine sends a snapshot, it does not contain the engine ID but the aircraft tail number and the positions of the engines on the wings.

Table 1. A non-exhaustive list of parameters recorded during flight snapshots.

Parameter	Description
TAT	External temperature
Altitude	Altitude
Mach number	Instant speed
Ground Speed	speed with respect to the ground
N2	Core speed
T25	Compressor Inlet Temperature
EGT	Exhaust Gas Temperature
N1	Fan Speed
PS3	Compressor Discharge Pressure
T3	Compressor Discharge Temperature
T12	Inlet temperature
PT2	Inlet Total Pressure
Fuel Flow	Fuel flow
VBV Position	Variable Bleed Valve Position
Oil Quantity/Temperature/Pressure	Oil Quantity/Temperature/Pressure

As hot corrosion depends on pollution, we also need to incorporate some environmental data into our inputs. For this purpose, we use the open databases METAR (https://www.aviationweather.gov/metar) for airport weather and SATAVIA (https://satavia.com/) for pollution data.

To link the different tables together we also use the open FlightAware database (https://fr.flightaware.com/) that presents a view of each flight with dates and airports of departure and arrival. This identifies snapshots and flight cycles but not engine IDs.

We also need output data to calibrate and validate the algorithm. Engines are regularly maintained, and an internal database called FDM (Flight Data Management) records all events relating to each engine serial number. This new table contains the inspection dates where notifications may be regarding corrosion, which will be mandatory for our purpose. Nevertheless, this table also contains engine installation details on a given aircraft. Thus, it becomes possible to connect an engine ID with its aircraft for a period.

2.2. Outputs

We only know if a metal part that interests us is corroded or not when an inspection is carried out during a shop visit. Therefore, we have no information about the state of the component during a long time. If the last inspection showed that the component was operational, it does not mean that the component is still not corroded now. It corresponds to right-censored data. Likewise, if the last inspection concludes that a component is corroded, it is not known how long the component has been corroded. This corresponds to left-censored data. All observation data will be censored, and it explains why we use a stochastic distribution as output for our model. Moreover, as our objective is to anticipate a remaining useful life (RUL) we select a reliability measure based on the Weibull law.

3. ALGORITHMIC MODEL

The idea behind our proposal is to combine a cumulative damage model (CDM) with survival analysis (Figure 1).

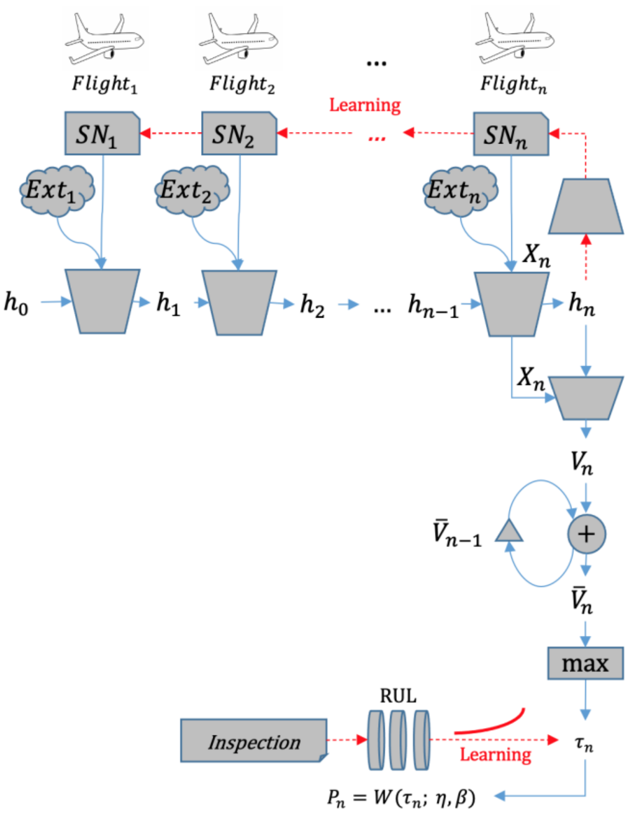

Figure 1. Design of the cumulative damage model.

Inputs from the engine during each flight n and external data (X_n) are fed in a recurrent neural network that maintains a state (h_n) of the engine. Another network calculates the exposure rates (V_n) which cumulative maximum (τ_n) is used as an exposure time for the survival law.

3.1. Cumulative damage model

A CDM is an indicator that increases with specific use of a component and reflects its current wear. However, the hot corrosion may not be explained by a single physical phenomenon. It is mainly a combination of events whose interaction leads to damage. To build this indicator we will record all the history of a given engine and use it with a neural model to maintain a current state of wear and come up with a cumulative output that we hope to relate with corrosion.

Figure 1 is a symbolic design of an algorithm able to learn a corrosion CDM. The inputs (snapshots and external data) are provided flight after flight and a state vector h_n of the engine is calculated. This state stores information relating to corrosion events and helps to build a global vector of cumulative indicators $\bar{V}_n$, each coordinate representing a factor for a specific accumulation of wear. Then we select our CDM τ_n as the maximum of the different wear factors.

3.2. Survival analysis

On the output size, survival analysis aims to predict a duration until failure.

The main goal is to approach the survival function $S(t)$ which gives the probability that a subject did not meet the event before time t. In our case we will replace the notion of time by our cumulative damage model which can be interpreted as a cumulative exposure time τ. Here we select to approach the cumulative distribution function $W(\tau) = 1 - S(\tau)$ that gives the probability that a subject meets the event before τ by a Weibull law (Weibull 1951) with parameters η (scale) and β (shape).

$$W(\tau) = 1 - \exp\left(-\left(\frac{\tau}{\eta}\right)^{\beta}\right) \tag{1}$$

Like all statistical models, survival models must be fitted to data. Survival data can be represented by a pair (n, c) where n is an engine cycle number (or a count of successive flights) and c is a label that give the observation status.

- $c = 0$ if an inspection occurs just after flight n and no corrosion was observed.
- $c = 1$ if an inspection that occurs just after flight n shows some corrosion of the component of interest.

Then, for each inspection i we have a set of observations $(n_i, c_i, \hat{p}_i)$ where $\hat{p}_i = W(\tau_{n_i})$. Finally, we may optimize our model by minimization of the negative log-likelihood:

$$L = -\sum_i c_i \ln(1 - \hat{p}_i) + (1 - c_i) \ln \hat{p}_i \tag{2}$$

This loss function corresponds to the binary cross-entropy loss.

3.3. Neural network

State models using recurrent networks have already shown interesting results (Ren et al. 2019). We just apply a simple GRU (Gated Recurrent Unit) on the input data X_n (snapshots and external context) to calculate the hidden state h_n, which represents the health status of the engine just after flight n. For other purpose like the monitoring of performance of the engine and not just a specific wear we used a more complex recurrent network including layers of attention (Langhendries and Lacaille 2022), but it was not necessary here. Then X_n and h_n are concatenated and given as input to a dense neural network that produces a vector output V_n (see Figure 2).

V_n is an instantaneous damage exposure vector corresponding to flight n. The instantaneous exposure vectors are summed after each flight to calculate a cumulative exposure vector $\bar{V}_n$ (simplified by V in the diagram below). And we select the maximum τ of the wear exposure coordinates as our specific "time" of exposure to corrosion conditions.

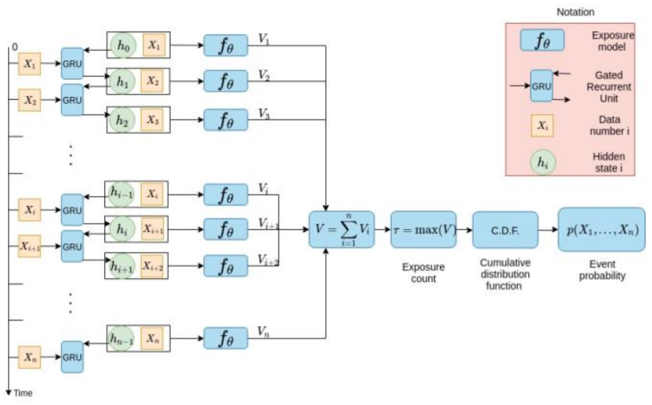

Figure 2. Detailed design of the recurrent neural network. Here f_θ is a dense neural network with weights parameters θ. The learning procedure will also adapt the cumulative distribution function of the survival law (the pair of Weibull parameters η and β).

4. APPLICATION TO TURBINE STATOR BLADES

A set of turbine stator blades is subject to hot corrosion. The same types of stators are positioned at different stages of the turbine (3 on this specific engine). Each is subject to different work conditions, but the same model can be used because the same causes apply on the same metal alloy. We are testing our model on 32 engines but with the different turbine stages and the large number of blades, we have been able to seriously increase the volume of observations.

Hot corrosion is a physical phenomenon that can cause damages to different metals exposed at high temperatures (Pettit 2011). The prediction of hot corrosion in turbofan engines is a difficult problem because it involves several different physical phenomena. Three different types of hot corrosion may arise in turbofan (Figure 3).

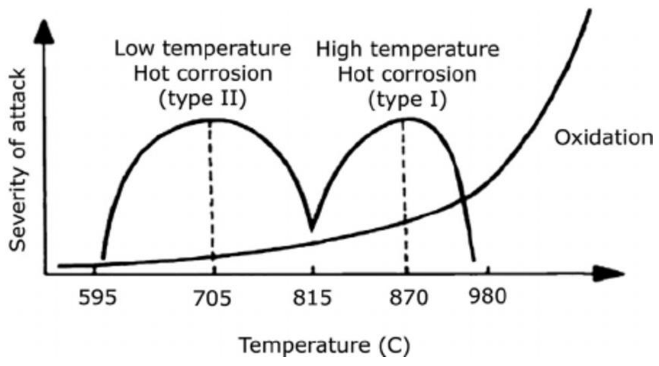

Figure 3. Hot corrosion severity according to the temperature.

Depending on temperature and pressure conditions, corrosion of type 1, 2 or oxidation can be preponderant. When the turbofan is operated, these conditions vary, therefore all three types of hot corrosion can appear and even coexist. Moreover, when a component shows wear due to corrosion, we do not know what type of corrosion is involved. Finally, the severity of hot corrosion attack is known to be influenced by chemicals which can act as a catalytic (for example SO3). Thus, air pollution can influence corrosion as well as the combustion reaction in the turbofan.

4.1. Implementation

The model is implemented in PyTorch. The exposure model is a 3 layers dense neural network with 90 neurons by layer (10 for the GRU). Learning procedure uses ADAM optimizer (Kingma and Ba 2015). We add a dropout regularization scheme (Srivastava et al. 2014) with $p = 0.4$ and a L^2 penalty to reduce over-fitting (with decay of 10^{-3}). The learning rate is equal to 10^{-3}.

The learning procedure uses all engines in a dataset and builds a common model for all engines (same network parameters). The data from each engine is viewed as a multivariate time series.

4.2. Results and comparisons

We evaluate the model using a 10-folds cross-validation procedure. Each engine is used only once in the validation set and capitalize thousands of flights. Hence we get only 3 or 4 engines per validation set in the cross procedure. We aggregate all results obtained on validation sets and present them (Figure 4) in the form of a boxen plot, or letter-value plot (Hofmann, Wickham, and Kafadar 2017).

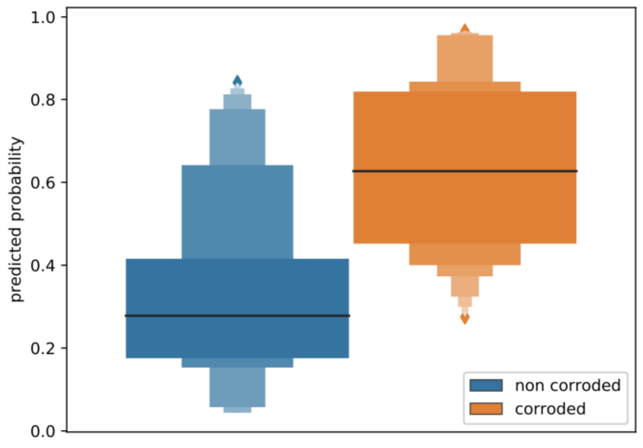

Figure 4. Boxenplot obtained by cross validation on test sets. On this graph each rectangle area is proportional to the number of observed engines.

This solution is compared to several others implemented using PySurvival (Fotso 2019). It includes the Cox model (Cox 1972), the Weibull parametric model and DeepSurv (Katzman et al. 2018). We used the package tsfresh (Christ et al. 2018) to extract a vector of features from each input time series (X_n^k) describing flights achieved by the engine k (we ignored the k indices in the previous description for clarity). Then, feature vectors are used by Cox model and DeepSurv. Figure 5 shows on ROC (Receiving Operating Curve) curves (Fawcett 2006) that our model performs better than baselines (Cox model and Weibull parametric model) and a state-of-the-art alternative solution (tsfresh features extraction + DeepSurv).

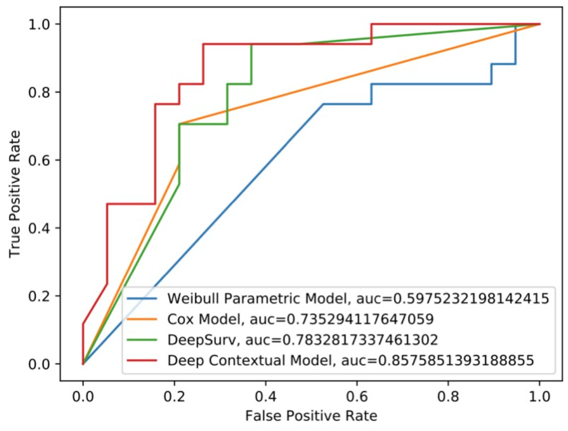

Figure 5. Comparisons with state-of-the-art survival models using ROC curves. The one we propose (red curve) behave better than the others on our set of data.

The area under the ROC curve (given in the legend Figure 5) may be interpreted as the proportion of pairs of observations that are well compared. In the binary case it approaches the probability of discriminating between a random selection of a pair of observations, that is to say whether the two blades are corroded, not corroded or only one is corroded (Clémençon, Lugosi, and Vayatis 2008).

5. INTERPRETATION OF THE RESULTS

The use of this algorithms produces a probability of corrosion for a stator turbine stage. The MRO (Maintenance Repair Overhaul) operator must decide to open the engine turbine to replace some blades, but this decision has an important maintenance cost. It would be preferable if an interpretable explanation was given.

We use Kohonen's algorithm (Kohonen 1995) to construct a self-organized map (SOM) of the input data. The quality of the algorithms results can also be seen when plotting the output probability of corrosion as the map background. This output was not used as input for data categorization by Kohonen's algorithm. Figures 6 and 7 presents such maps. The axis have no meaning, each cell of the map correspond to a category of similar inputs and the cells are organized in a way that try to conserve the proximity of the observations in the real space. The color of a cell may be any coordinate of the cell prototype or a computation from this prototype or observations that belongs to this cell. In our case we present the mean value of the corrosion prediction on figure 6 and mean flight leg duration (an input) on figure 7.

Figure 6 shows that the corrosion probability calculated by the model are concentrated on small areas. This proves that specific missions have a greater impact on the wear than others. In addition, one can also plot the average value of an input measurement in each cell of the map and look at the previous graphs as overlays to infer relationships between flight data and the risk of corrosion.

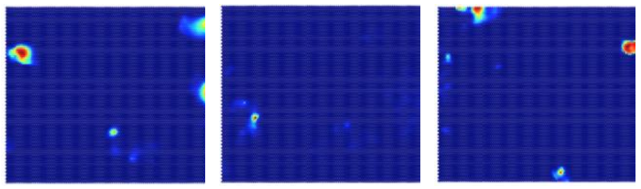
Figure 6. Observation of the probability of corrosion for each of the three stator stages on the topological cartography of the input data.

Figure 7 is a trivial example which shows that when the duration of the flight is superimposed on the third map (Figure 6), it can be deduced that flight leg impacts one type of corrosion on the third turbine stage. In fact, it was not the only factor, but obviously it has less of an impact on the other two stages.

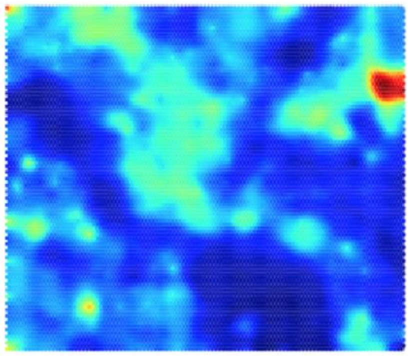
Figure 7. Example of the flight leg duration on the map. Blue flights are short and red ones are long.

6. CONCLUSION

This work introduces a new model for survival analysis. It uses past observations recoded over the life of an aircraft engine to construct a partial state of wear sufficient to expose corrosion on turbine stator blades.

The actual algorithm relies on a single GRU unit to memorize the current state. On other applications it can be replaced by a more consistent architecture. The next step is a projection of the current observation modulated by the current state to estimate an instantaneous wear risk corresponding to each degradation mode: all the factors, not necessarily known, causing one of the corrosion mechanisms. At any instant, after each flight, the preponderant cumulated risk is retained to serve as an exposure time for a Weibull distribution.

Although the results are not bad, they are still difficult to interpret. We therefore use an a posteriori solution based on a self-organizing map (SOM) to help engineers identify the main causes of degradation. This last tool mainly helps to gain confidence in the application, but its main strength is that it opens visibility into the causes of corrosion and can help designers deduce a better material or coating for the next generation of blades.

REFERENCES

Christ, Maximilian, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. "Time Series FeatuRe Extraction on Basis of Scalable Hypothesis Tests (Tsfresh – A Python Package)." *Neurocomputing* 307:72–77.

Clémençon, Stéphan, Gabor Lugosi, and Nicolas Vayatis. 2008. "Ranking and Empirical Minimization of U-Statistics." *Annals of Statistics* 36(2):844–74.

Cox, D. R. 1972. "Regression Models and Life-Tables." *Journal of the Royal Statistical Society. Series B (Methodological)* 34(2):187–220.

Fawcett, Tom. 2006. "An Introduction to ROC Analysis." *Pattern Recognition Letters* 27(8):861–74.

Fotso, Stephane. 2019. "PySurvival: Open Source Package for Survival Analysis Modeling."

Hofmann, Heike, Hadley Wickham, and Karen Kafadar. 2017. "Letter-Value Plots: Boxplots for Large Data." *Journal of Computational and Graphical Statistics* 26(3):469–77.

Katzman, Jared L., Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. 2018. "DeepSurv: Personalized Treatment Recommender System Using a Cox Proportional Hazards Deep Neural Network." *BMC Medical Research Methodology* 18(1):24.

Kingma, Diederik P. and Jimmy Lei Ba. 2015. "Adam: A Method For Stochastic Optimization." in *International Conference for Learning Representations (ICLR)*. San Diego.

Kohonen, Teuvo. 1988. *Self-Organization and Associative Memory*. Second Edi. Berlin: Springer.

Kohonen, Teuvo. 1995. *Self-Organizing Maps*. Vol. 30. Springer.

Langhendries, Raphaël and Jérôme Lacaille. 2020. "Operability Forecasting Combining Neural Network and Survival Analysis with an Application to Hot Corrosion in Turbofan." in *International Conference on Maintenance, Condition Monitoring and Diagnostics (MCMD).*

Langhendries, Raphaël and Jérôme Lacaille. 2022. "Turbofan Exhaust Gas Temperature Forecasting and Performance Monitoring with a Neural Network Model." in *European Conference on Safety and Reliability (ESREL)*. Dublin.

Pettit, Fred. 2011. "Hot Corrosion of Metals and Alloys." *Oxidation of Metals* 76(1–2):1–21.

Ren, Kan, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. 2019. "Deep Recurrent Survival Analysis." Pp. 4798–4805 in *AAAI Conference on Artificial Intelligence*. Vol. 33.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15(56):1929–58.

Weibull, Waloddi. 1951. "A Statistical Distribution Function of Wide Applicability." *Journal of Applied Mechanics* 18:7.

BIOGRAPHIES

Raphaël Langhendries is a doctoral student in applied mathematics at the statistics laboratory of the University of Paris 1 Panthéon Sorbonne. He is doing his thesis in collaboration with Safran Aircraft Engines and is a member of the company's DataLab. During his work Raphaël collaborated with business teams on engine performance, design, and operations. Its most striking result is the demonstration of a concentration inequality that validates the use of learning algorithms on data from successive flights and therefore clearly dependent. Within the company, he paved the way for the exploitation of Deep Survival techniques to modernize the design of new damage indicators and the prediction of lifetimes.

Jérôme Lacaille is an expert emeritus at Safran Aircraft Engines whose successive missions have been to develop an algorithmic environment for prognostic and health monitoring (PHM) of engines, the creation of a DataLab for the analysis of company data and the animation of a mathematics, data analysis and scientific computing business network. Doctor in mathematics, authorized to direct research and a former normalien agrégé in mathematics, Jérôme supervises a team of doctoral students within the company and continues to teach at the university. Member of the board (ex-vice-president) of the society of applied and industrial mathematics (SMAI), as well as of numerous other academic committees, Jérôme assumes the role of scientific ambassador for Safran.