Least squares smoothed k-nearest neighbors online prediction of the remaining useful life of a NASA turbofan

Least Squares Smoothed k-Nearest Neighbors Online Prediction of the Remaining Useful Life of a NASA Turbofan

Luca Viale [1,*], Alessandro Paolo Daga [1], Alessandro Fasana [1] and Luigi Garibaldi [1]

[1] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy;

[*] Correspondence: luca.viale@polito.it

Abstract

An accurate prediction of the Remaining Useful Life (RUL) of aircraft engines plays a fundamental role in the aerospace field since it is both mission and safety critical. Remaining Useful Life (RUL) estimation models consequently allow improving the performance and reducing maintenance costs. They can be classified into several approaches: experimental, data-based, physics-based and hybrid. This paper proposes a novel data-driven method to increase accuracy on the RUL prediction and to get a real-time prognostic system, considering multiple degradation mechanisms and making the model easy to implement. The proposed method exploits a modified k-Nearest Neighbors Interpolation (kNNI) with an a posteriori Least Square Smoothing (LSS) implementation. A double optimization by means of a Genetic Algorithm (GA) and a Root-Mean-Square Error (RMSE) minimization has been carried out in order to obtain the best performing model parameters according to the related application dataset. In addition, based on the application under analysis, the LSS method has been generalized as an online Cumulative and Moving Average (CMA) mixture filter, which uses an optimized sliding window. The generation of the prognostic model is based on NASA data generated with the dynamic model Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) with run-to-failure trajectories relative to a small fleet of aircraft engines under realistic flight conditions. Finally, a comparison with a reference kNN-based method already known in literature was used to demonstrate the goodness of the results and the performance improvements.

Keywords

Remaining Useful Life (RUL) – Aircraft turbofan – Prognostics – k-Nearest Neighbors (kNN) – Least Square Smoothing – Cumulative and Moving Average

1. Introduction

Aircraft engines are highly complex rotating thermal machines [1] and are a critical element of airships. Their failures are particularly significant and occur quite frequently, given their functioning in extreme environmental conditions for long periods. Since safety and reliability are key concepts in the aerospace and aeronautic fields, an accurate prediction of the Remaining Useful Life (RUL) of aeroengines is a fundamental element in the Prognostic and Health Management (PHM) context. In addition, the State Of Health (SOH) of engines affects also the performance and costs of aircrafts. Given that maintenance involves high costs and complex procedures, research in the field of diagnostics and prognostics of mechanical systems has been rapidly evolving in recent years. Both diagnostic and prognostic techniques allow recognizing or predicting a mechanical system failure thanks to degradation data which, for instance, consists of vibration signals [2–7], potentially integrated with temperature measurements [8] or signals of various nature [9] and analyzed in the time or frequency domain [10,11].

The RUL refers to the remaining time in terms of the utility of a component or machinery, where the utility usually has an economic rather than a technical conception [12]. The economic life of a machine is very often shorter than the technical one. In addition to this, the RUL depends on many other factors, such as the operating environment, the current age of the mechanical system, health information and monitoring of observed conditions. Therefore, the model developed in this study aims to predict the RUL of an aircraft engine in conditions of highly variable flight parameters considering possible failure modes. The multivariate NASA's N-CMAPSS [13] dataset has been used for this purpose because it includes data on a fleet of aircraft engines under realistic flight conditions. This dataset mainly contains run-to-failure physical measurements as pressures and temperatures from the turbofan engines, flight parameters as altitude and Mach number, and other auxiliary data. In the aeronautical field, the RUL can be expressed as the number of flights that an engine can perform from the present moment before reaching failure conditions [14] and, therefore, it is essential to predict it in an accurate and timely manner because this application is both mission and safety critical.

Numerous models and different approaches for RUL prediction exist in the literature [15,16]. The choice of the most appropriate approach to the application (physics-based, data-driven, experimental or hybrid) mainly depends on the available information and a trade-off between the model complexity and the required accuracy. For example, there are prognostic models based on particle filters [17] which, as any non-deterministic prediction techniques, can lead to a high error as the time horizon increases, as well as they strongly depend on the initial prediction step. Other data-driven models of aircraft engine prognostics are already present in some studies. For instance, Ordóñez et al. [18] developed a hybrid model which combines an Auto-Regressive Integrated Moving Average (ARIMA) with a Support Vector Regression (SVR) model that, however, is able to predict few time units in advance. Instead, Zhao et al. [19] based their study on pattern recognition thanks to neural networks. Similarly, Liu et al. [20] developed a RUL prediction model combining clustering and Long Short Term Memory (LSTM) network. Another not very different approach was carried out by Zheng et al. [21] with the development of a model that combines the Time Window (TW) and Extreme Learning Machine (ELM). Although they are quite effective methods, the model creation and training are considerably complex and time-consuming. Furthermore, neural networks can be considered as a "black box" due to the nature of their functioning, consequently reducing

the confidence in the results. This know-how of the overall system behavior without the specific intermediate steps knowledge, which is typical of neural networks, is a concept referred to as unpredictability of Artificial Intelligence (AI) [22]. In the context of AI safety [23–26] and AI governance [27], tools subject to unpredictability are not safe to use.

Considering methods of different types present in the literature – although the field of application is different – Zhou et al. [28] adopted a method based on k-Nearest Neighbors (kNN) [29,30] to estimate the RUL for lithium-ion cells. Their study proposes a RUL prediction by means of a weighted kNN, which is intrinsically smoothed.

The present paper proposes a data-driven method for the RUL prediction of an aeroengine based on a modified k-Nearest Neighbors Interpolation (kNNI). Unlike the reference kNN-based method [28], this paper proposes a new method that optimizes a kNNI using Genetic Algorithms (GA) [31] and integrating it with an a posteriori Least Squares (LS) fitting technique. The Least Square Smoothing (LSS) applied to this specific case is afterwards generalized as a Cumulative and Moving Average (CMA) mixture filter. In this way, it is possible both to obtain good results in terms of the predicted RUL and to develop a real-time model.

The proposed method mainly permits the following four performance improvements of the RUL prediction over the reference method:

1) First of all, the obtained results show that the a posteriori LSS allows obtaining greater accuracy for the RUL prediction without increasing the model complexity. This improvement is considerable in terms of Root-Mean-Square Error (RMSE) reduction, which is the considered cost function.
2) Furthermore, it can be observed that the reference method, as generally do the methods based on moving regression [32] (e.g., LOESS and LOWESS), applies an intrinsic smoothing to the kNN-based method through a weighted average. Therefore, this smoothing is performed in the multidimensional space of the features and can be considered similar to a Gaussian kernel filter [33]. On the other hand, the proposed method provides a dedicated smoothing application in the time domain thanks to its a posteriori LSS. This would be equivalent to a Gaussian kernel in the space domain which varies over time and depends on different RUL values. In addition, the present method is independent from the k parameter (which represents the number of considered nearest neighbors), which constrains the smoothing degree in the weighted kNN model.
3) Given the independence of the proposed method with respect to the k parameter, a further advantage consists in being less subject to the curse of dimensionality. This implies a lower computational effort, especially considering Big Data applications.
4) Finally, studies in literature that consider multiple degradation mechanisms are limited since the hypothesis of the presence of a single category of failures is assumed in most cases [34]. However, the proposed model can consider multiple types of degradation tendencies thanks to the recognition of different failure paths in the multivariate space. This can also be observed from the obtained results since the dataset includes various failure modes.

This paper is structured as follows. Section 2 provides a brief description of the analyzed dataset and the related turbofan. Section 3 describes all the techniques of the developed method: pre-processing and features extraction, kNNI and filtering, LSS with its improvements and the

generalization as CMA. Section 4 regards the method parameters optimization, both thanks to the RMSE minimization and by means of GA. Finally, Sections 5 and 6 respectively report the results and conclusions.

2. System description and dataset

The system under analysis corresponds to a high-bypass, twin-spool commercial turbofan engine. The engine consists of six main components: fan, Low-Pressure Compressor (LPC), High-Pressure Compressor (HPC), combustor or burner, High-Pressure Turbine (HPT), and Low-Pressure Turbine (LPT). The core shaft (also known as high-speed shaft) connects the HPC and HPT; while the fan shaft (also known as low-speed shaft) connects the fan, LPC, and LPT [35]. In addition to these components, the engine has a combustor which needs a fuel flow W_f, an inlet at the front, a nozzle at the rear, a bypass duct, an inter-stage bleed valve, and a variable-angle stator. A simplified representation of the engine [36] with the related sensors positions is shown in Figure 1.

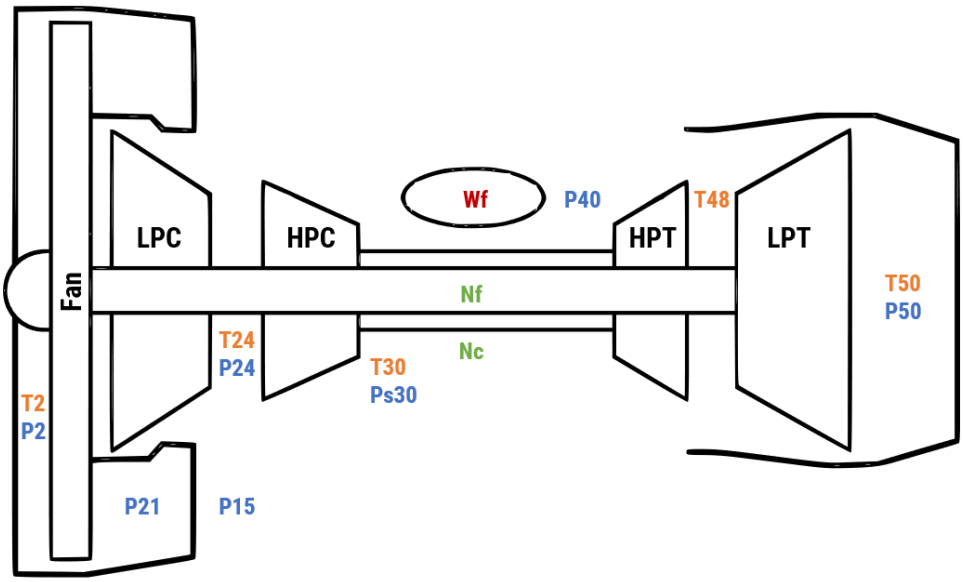


Figure 1 - Schematic representation of the turbofan engine model.

The data are multivariate time-series of sensors readings, and they contain the corresponding remaining useful life label (RUL) from a fleet of $U_d = 54$ units. Each observation is a vector of $p = 14$ sensor readings taken at certain operating conditions. Sensors mainly record fuel flow, speed, pressures, and temperatures at the different turbofan stages. Pressures and temperatures are mainly reported as total, i.e., they refer to the condition in which the fluid is theoretically slowed down to zero speed with an adiabatic, non-ergodic and isentropic transformation. The length of the sensory signal can, in general, differ from unit to unit. Given this set-up, the task is to optimize a predictive model that provides a reliable RUL estimate on a further dataset of $U_t = 36$ units, the test dataset.

The N-CMAPSS dataset provides synthetic run-to-failure degradation trajectories of a fleet of turbofan engines with unknown initial health states subject to real flight conditions. Since this dataset contains run-to-failure degradation trajectories, it was not necessary to apply any data augmentation techniques, e.g., the dynamic time warping which allows to increase the availability of data precisely in conditions of run-to-failure data scarcity [37]. The initial health conditions of each unit are unknown, and their degradation starts at some time during the flight history, slowly developing with different evolutions. Concretely, all the rotating sub-components of the engine can be affected by flow and efficiency deterioration. The flight conditions contained in the dataset follow real trends typical of operating aircrafts. The dataset contains three flight classes depending on the flight length performed by each (i.e., short, medium, and long length). Each flight cycle covers climb, cruise, and descent conditions which correspond to different flight routes with different records lengths.

The N-CMAPSS dataset contains eight files describing run-to-failure degradation trajectories from 90 units and seven different failure modes of the rotating sub-components. The failure modes of each set of data are summarized in Table 1. Two datasets, development and test, are present and each of them contains the operative conditions variables w, the measured signals X_s, the RUL label, and some auxiliary data (i.e., the unit number, the flight cycle number, the flight class which indicates the flight length, and the health state which is a binary index representing the hypothetical deterioration starting point). Tables 2 – 4 provide a description of each variable present in the dataset [14]. In particular, matrices partition was slightly changed based on the use of each feature (e.g., P2 was moved to the scenario descriptors despite being originally included in the measurements matrix).

Table 1 - Overview of the datasets containing different failure modes affecting the flow (F) and/or the efficiency (E) of the rotating sub-components.

Name	# Units	# Failure Modes	Fan		LPC		HPC		HPT		LPT	
			E	F	E	F	E	F	E	F	E	F
DS01	10	1							X			
DS03	15	2							X		X	X
DS04	10	3	X	X								
DS05	10	4					X	X				
DS06	10	5			X	X	X	X				
DS07	10	6									X	X
DS08a	15	7	X	X	X	X	X	X	X	X	X	X
DS08d	10	7	X	X	X	X	X	X	X	X	X	X

Table 2 - Scenario descriptors w (i.e., flight data).

#	Symbol	Description	Units
1	alt	Altitude	Ft
2	Mach	Flight Mach number	-
3	TRA	Throttle-resolver angle	%
4	T2	Total temperature at fan inlet	°R
5	P2	Total pressure at fan inlet	psia

Table 3 - Auxiliary data A.

#	Symbol	Description	Units
1	unit	Unit number	-
2	cycle	Flight cycle number	-
3	Fc	Flight class	-
4	h_s	Health state	-

Table 4 - Measurements X_s.

#	Symbol	Description	Units
1	Wf	Fuel flow	pps
2	Nf	Physical fan speed	rpm
3	Nc	Physical core speed	rpm
4	T24	Total temperature at LPC outlet	°R
5	T30	Total temperature at HPC outlet	°R
6	T48	Total temperature at HPT outlet	°R
7	T50	Total temperature at LPT outlet	°R
8	P15	Total pressure in bypass-duct	psia
9	P21	Total pressure at fan outlet	psia
10	P24	Total pressure at LPC outlet	psia
11	Ps30	Static pressure at HPC outlet	psia
12	P40	Total pressure at burner outlet	psia
13	P50	Total pressure at LPT outlet	psia

3. Methodology

The proposed method exploits Genetic Algorithms (GA) to find the optimal parameters of a modified kNN Interpolation (kNNI) integrated with a Least Squares Smoothing (LSS). To reach an online RUL value, the pre-processing and features extraction, kNNI and filtering, LSS and GA optimization phases were mainly carried out. All these steps are detailed in the following Sections and summarized in the right branch of the flowchart in Figure 2. In addition, Figure 2 includes the reference method in the left branch in order to highlight the differences between the proposed and benchmark methods.

3.1 Pre-processing and features extraction

First, pre-processing of the data was carried out. Since the dataset is mainly composed of pressure and temperature values referring to the different stages of the engine, these measurements have been calculated in relative terms by subtracting the temperature and pressure values at the fan inlet (i.e., T2 and P2).

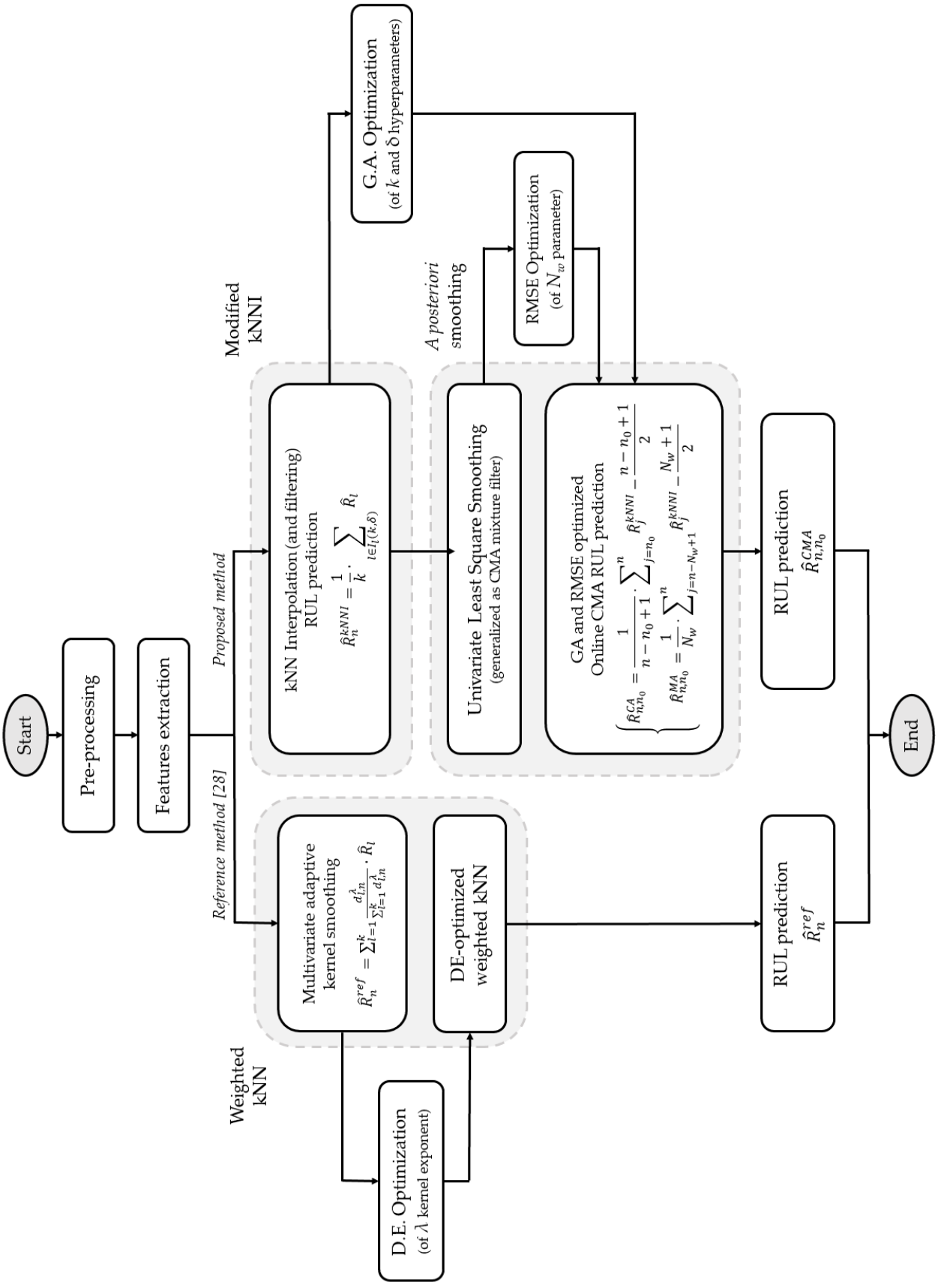


Figure 2 – Flowchart representing the proposed (in the right branch) and reference (in the left branch) methods. As can be seen, the main difference is the following observation: the reference method has a multivariate intrinsic a priori smoothing, while the proposed one implements a univariate a posteriori smoothing phase which allows improving RUL prediction performances.

This first pre-processing phase made it possible to obtain a single matrix describing all the flights (both for development and test) with the data sampled at a frequency of $f_s = 1\ \mathrm{Hz}$. Remember that the RUL value is given in cycle units and, consequently, it respects the following characteristics: it has constant value for each flight and decreases linearly with increasing flights. Since it turned out to be the best choice, after having recognized and appropriately separated the individual flights, the average value of each parameter has been calculated per each flight. Other features were also tested (RMS, skewness, and kurtosis), but only the mean value was used since it turned out to be the feature with the best performance. Attempts to extract the features by recognizing the various phases of the flight (take-off, cruise, and landing) were made, but the average value for the entire flight proved to be the most promising in terms of accuracy. In this way two $r \times c$ matrices were obtained, where $c_d = 4089$ and $c_t = 2736$ respectively represent the total number of flights including all the damages and all the units of the development and test dataset, while $r = 13$ represents the considered features.

These matrices represent all flights within a multivariate space. It is assumed that, as the wear and the type of damage (and, therefore, the RUL value) vary, the position of the points within the multivariate space also varies. This is highlighted with a simple Principal Component Analysis (PCA) [38]. Indeed, representing the 6th and 7th principal directions, it can be observed in Figure 3 how the different types of damage are partially distinguished in certain regions of the PCA subspace. Its ramifications represent the different failure paths that correspond to the multiple degradation mechanisms (and, therefore, allow recognizing them). The first five Principal Components (PCs) have not been used since they collect information from the features with greater variance and are not necessarily those influenced by damages and units' health conditions. In other words, PCs #6 and #7 representation corresponds to an orthogonal regression in which the residuals are analyzed. Furthermore, from Figure 4 it is also possible to notice how the RUL varies within the described subspace. This particular composition of the features allows noting that points describing each flight have high RUL values around the origin and move with directions depending on the type of damage as the RUL decreases. On these bases, after calculating a new point representing a generic flight with conditions w and X_s, this should be located in the proximity of flights with similar wear, damage and RUL.

3.2 k-Nearest Neighbors Interpolation

After the extraction and adequate pre-processing of the features, the proposed method exploits the kNNI. Indeed, the Euclidean distances of the new point are calculated with respect to all points of the development dataset. The predicted RUL value is equal to the mean of the RUL of the nearest k points, where the optimized value of k is obtained by means of GA. Before calculating the distances between the flight under analysis and those of the development dataset, its features were standardized in z-scores using the average value and standard deviation of the development data.

This method is more effective for low RUL values, i.e., towards the end of the engine's life. This is mainly because the points cloud is the densest in the region of space representing

healthy conditions, with high RUL values. These conditions produce more noise for high RUL values. For this reason, specific filters have been designed and adopted for the selection of flights to be considered, based on auxiliary information from Table **3** (i.e., current flight cycle number and flight class, F_c). In this way, the kNNI allows examining the k closest flights to the one under analysis among those present in the development dataset with the most similar conditions. A novel parameter δ has been added to establish the range of flight cycles to be considered for the use of these filters. This parameter is also optimized through GA. In addition to the use of these filters, the smoothing that will be used in the next phase will also reduce the noise at high RULs and, consequently, also the prediction error.

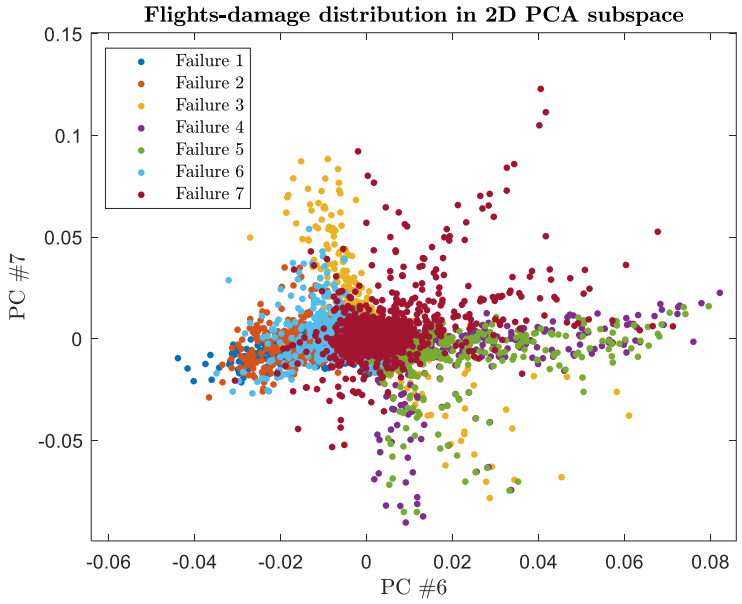
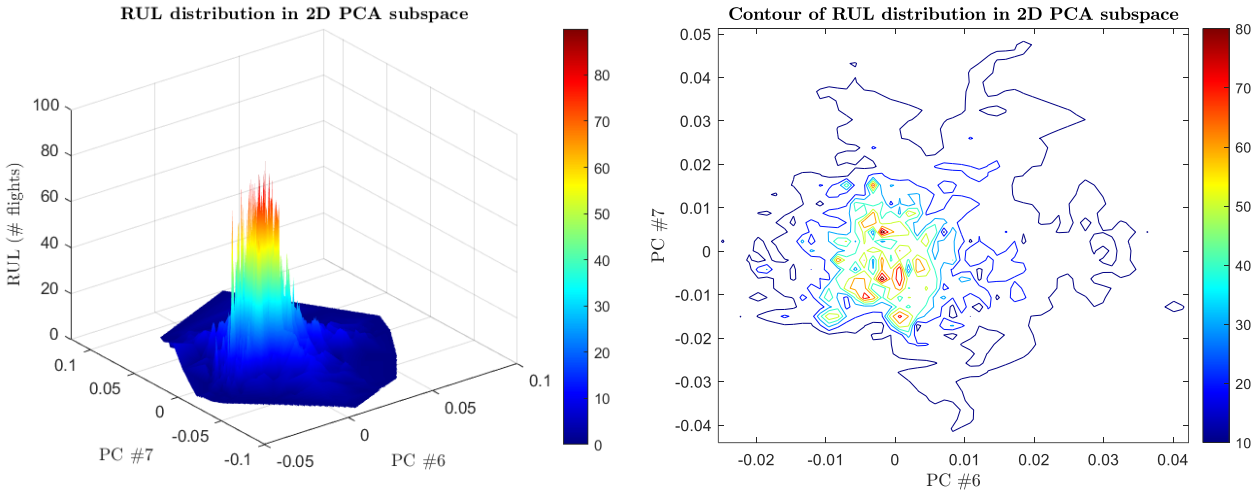


Figure 3 - Representation of flights as the type of damage varies in a subspace generated by PCA.



(**a**) (**b**)

Figure 4 - (a) RUL distribution and (b) contour of RUL distribution in a subspace generated by PCA.

3.3 Least Square Smoothing

As mentioned in the previous Sections, the kNNI has been integrated in a novel approach with an LS [39] smoothing method to improve the RUL prediction, and consequently to reduce the error. The importance of adding this smoothing technique is substantial for obtaining more precise results. Indeed, the LSS allows considering a greater amount of information for each prediction. This is due to the fact that LSS technique considers all flight data starting from n_0 to predict the RUL ($\hat{R}^{LSS}_{n,n_0}$) of the n-th flight, while the RUL ($\hat{R}^{kNNI}_{n}$) obtained through kNNI would only consider this last flight information. In addition, it is possible to notice that the proposed smoothing method finds its application after the kNNI. For this reason, it performs the smoothing in the univariate time domain rather than in the multidimensional space domain. The latter is equivalent to a Gaussian kernel convolution, which does not consider the time variable and potentially flattens the distribution of different RULs.

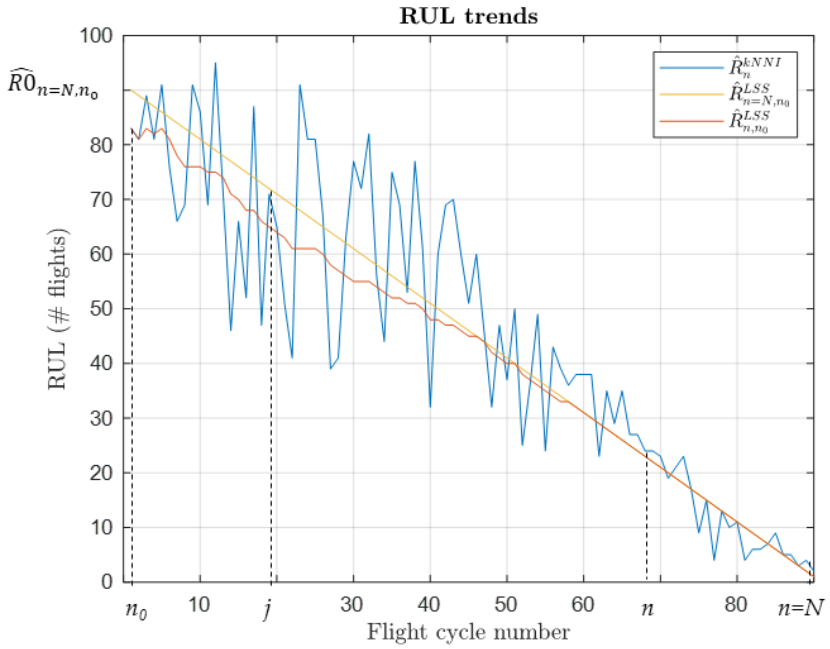


Figure 5 – Exemplary chart representing the trends of the kNNI-predicted RUL ($\hat{R}^{kNNI}_{n}$), LSS-predicted RUL ($\hat{R}^{LSS}_{n,n_0}$) on the n-th flight (Eq.8), and ($\hat{R}^{LSS}_{n=N,n_0}$) on the flight $n = N$ (where N represents the last available flight) with an offline backpropagation, performed for one of the considered units. To clarify the nomenclature, the main variables used in the demonstrations have been reported.

The objective to be achieved with this fitting technique is to calculate the predicted RUL, denominated $\hat{R}^{LSS}(n|n_0, k, \delta)$, minimizing the error $\varepsilon(n)$ of the kNNI prediction $\hat{R}^{kNNI}(n|k, \delta)$ with respect to the theoretical RUL $R(n)$ thanks to the proposed LSS, as shown in Figure 5. The variable n refers to a generic flight under analysis, n_0 is the first recorded flight (which does not strictly correspond to the aircraft's first flight), k and δ are the hyperparameters intrinsic to the kNNI method. For the sake of simplicity, the previous

notation is abbreviated by removing the k and δ parameters and reporting n and n_0 variables to the subscript, if any. The theoretical discrete RUL line follows the equation:

$$R_n = R0 + m \cdot n \tag{1}$$

where R_n is the theoretical RUL on the n-th flight, $R0$ is the total number of flights that an aircraft can perform (which corresponds to the RUL prediction before the turbofan's first flight), and $m = -1$ is the line slope since the RUL is measured as number of flights. Given a filtered set of data I_l depending on k and δ parameters, the kNNI RUL prediction is equal to:

$$\hat{R}_n^{kNNI} = \frac{1}{k} \cdot \sum_{l \in I_l(k,\delta)} R_l \tag{2}$$

That said, the LSS method could be applied to minimize the error ε_n which coincides with the squared difference between the kNNI-predicted RUL ($\hat{R}_n^{kNNI}$) and the theoretical RUL (R_n):

$$\varepsilon_n = \sum_{j=n_0}^{n} \left(\hat{R}_j^{kNNI} - R_j\right)^2 = \sum_{j=n_0}^{n} \left(\hat{R}_j^{kNNI} - R0 + j\right)^2 \tag{3}$$

where j is a generic flight in the $[n_0, n]$ interval. By deriving Eq. (3), it is possible to obtain the estimate of the intercept $\widehat{R0}_{n,n_0}$ (which corresponds to $R0$ plus an error fraction) which minimizes the error:

$$\widehat{R0}_{n,n_0} = \frac{1}{n - n_0 + 1} \sum_{j=n_0}^{n} \left(\hat{R}_j^{kNNI} + j\right) \tag{4}$$

Considering that the predicted RUL line follows the same behavior of the theoretical one in Eq. (1):

$$\hat{R}_{n,n_0}^{LSS} = \widehat{R0}_{n,n_0} - n \tag{5}$$

it is possible to obtain the LSS-predicted RUL:

$$\hat{R}_{n,n_0}^{LSS} = \frac{1}{n - n_0 + 1} \sum_{j=n_0}^{n} \left(\hat{R}_j^{kNNI} + j\right) - n = \tag{6}$$

$$= \frac{1}{n - n_0 + 1} \sum_{j=n_0}^{n} \hat{R}_j^{kNNI} + \frac{1}{n - n_0 + 1} \sum_{j=n_0}^{n} j - n = \tag{7}$$

$$= \frac{1}{n - n_0 + 1} \sum_{j=n_0}^{n} \hat{R}_j^{kNNI} - \frac{n - n_0 + 1}{2} \tag{8}$$

Thanks to Eq. (8), it can be noted that the LSS-predicted RUL $\hat{R}_{n,n_0}^{LSS}$ is equal to the difference between the kNNI-predicted RUL $\hat{R}_j^{kNNI}$ average value on flights in the $[n_0, n]$ interval and half the number of flights performed by the unit in the same interval. Furthermore, Eq. (8) allows directly using the predicted RUL $\hat{R}_{n,n_0}^{LSS}$ on the n-th flight to calculate the next flight RUL $\hat{R}_{n+1,n_0}^{LSS}$ through:

$$\hat{R}_{n+1,n_0}^{LSS} = \widehat{R0}_{n+1,n_0} - (n+1) = \frac{(n-n_0+1)\cdot \widehat{R0}_{n,n_0} + \left(\hat{R}_{n+1}^{kNNI} + n + 1\right)}{n-n_0+2} - n - 1 \qquad (9)$$

In this way, the computational effort is lightened by avoiding the calculation of a sum on $n - n_0 + 1$ terms at each new flight to be predicted. This aspect is of considerable importance for real-time applications as it allows to significantly reduce the computational effort thus making online implementation faster.

3.4 Cumulative and moving averages mixture filter adapted for an online application

The described model exploits the LS principle to generate smoothing. Focusing on the LSS result in Eq. (11), this can be implemented both as a Cumulative Average (CA) [40] and as a Moving Average (MA) [41] by changing the initial data point n_0. Indeed, the model analyzed and used so far only considers the variation of the flight n under analysis given a starting flight n_0. However, the method can potentially predict the RUL by varying both parameters n_0 and n, introducing a sliding window of length N_w. Then, a generalization as a Cumulative and Moving Averages (CMA) model was proposed to consider a more generic LS-based smoothing, exploiting an additional parameter to optimize the model and reach more precise predictions.

The main difference that distinguishes traditional CMA from the proposed method is that they cannot be used in a real-time model. Indeed, they allow to smooth the curve by calculating the new corrected value with $N_w/2$ flights delay (i.e., centering the new calculated value with respect to the considered $[n_0, n]$ interval). For this reason, it is necessary to adapt the CMA to make it applicable to a real-time prognostic system. Therefore, the considered windows of length N_w are not centered with respect to the n-th flight under analysis (as it is typically performed) but they are aligned on the n-th flight so that the window allows considering the previous N_w flights.

Therefore, it is possible to introduce the LSS model generalization in terms of a CMA mixture filter adapted for an online application. Indeed, the model generalization for the online RUL prediction during the n-th flight can be rewritten as in the following equation:

$$\begin{cases} \widehat{R0}_{n,n_0} = \dfrac{1}{n-n_0+1}\displaystyle\sum_{j=n_0}^{n}\left(\hat{R}_j^{kNNI} + j\right) & if\ (n-n_0) < N_w \\ \widehat{R0}_{n,n_0} = \dfrac{1}{N_w}\displaystyle\sum_{j=n-N_w+1}^{n}\left(\hat{R}_j^{kNNI} + j\right) & if\ (n-n_0) \geq N_w \end{cases} \qquad (10)$$

$$\begin{cases} \hat{R}_{n,n_0}^{CA} = \dfrac{1}{n-n_0+1}\displaystyle\sum_{j=n_0}^{n} \hat{R}_j^{kNNI} - \dfrac{n-n_0+1}{2} & if\ (n-n_0) < N_w \\ \hat{R}_{n,n_0}^{MA} = \dfrac{1}{N_w}\displaystyle\sum_{j=n-N_w+1}^{n} \hat{R}_j^{kNNI} - \dfrac{N_w+1}{2} & if\ (n-n_0) \geq N_w \end{cases} \quad (11)$$

where N_w is the sliding window dimension that will be optimized taking into account the RMSE optimization, as described in Section 4. It can be observed that the $(n-n_0) < N_w$ condition causes the RUL prediction as an online CA, while an online MA is used for the second condition. The implementation of the N_w window allows considering a reduced number of flights, diminishing the noise peculiar to the first predictions.

4. Optimization

The model developed so far consists of the integration of LSS with the kNNI and the related use of filters. As explained in the previous Sections, it turns out to be a parametric model. Indeed, the choice of k and δ (which allow identifying within the development dataset the flights most similar to the case under analysis) influences the model precision. For this reason, their optimal values are obtained using a Genetic Algorithm (GA) hyperparameter tuning. Furthermore, the optimization of the sliding window dimension N_w is added to these parameters if the generalized version of the proposed method is considered, as per Section 3.4.

The optimization was carried out by predicting the flights contained in the test dataset thanks to the development dataset. In the prognostics field, there are several cost functions for the prediction evaluation [42]. In this case, the RMSE – usually employed in the prognostics field – has been chosen as score function both for GA optimization and for the following choice of N_w intrinsic in the LSS phase:

$$RMSE = \sqrt{E\left[\left(\hat{R}_{n,n_0,N_w}^{CMA} - R_n\right)^2\right]} \quad (12)$$

where $E[\,]$ represents the expected value operator, R_n and $\hat{R}_{n,n_0,N_w}^{CMA}$ are respectively the real and estimated RUL via the generalized and adapted CMA mixture filter. Please note that the RMSE value is obtained as the average of the RMSEs inherent to each unit. An overall RMSEs mean value of all flights was not directly calculated because it would have entailed a greater importance to the units with longer life, while neglecting those with shorter lifetime.

This model optimization can be divided into two phases: the first point concerns the parameters k and δ, which vary the preceding kNNI conditions; the second phase concerns N_w, which allows modifying the a posteriori smoothing. In particular, the optimization of N_w is implicit in the GA hyperparameter tuning as it is established as the RMSE minimization. Therefore, the following Subsections will describe the bias-variance optimization applied for N_w and the ensuing GA implementation, which allows finding k and δ parameters, which consequently determine the choice of N_w.

4.1 RMSE optimization

The RMSE optimization was introduced to minimize the prediction error and, consequently, to derive the related sliding window length N_w. This RMSE optimization allows minimizing the mean square discrepancy between the predicted and the real RULs. This procedure can be observed with a different and useful interpretation thanks to the RMSE decomposition into bias β and variance σ [43]. The bias is the error that represents how much the expected value of the predictions differs from its real value, while the variance describes the predictions dispersion. The bias-variance trade-off was conceived as an attempt to simultaneously minimize these two types of errors to avoid underfitting and overfitting [44]. The RMSE and, consequently, the sum of the square bias and the variance usually have a parabolic-like trend that allows identifying a value of the reference index N_w which minimizes both errors. Therefore, the RMSE optimization can be interpreted as the identification of the window length N_w which allows generating a smoothing effect by reducing the variance to the detriment of a smaller bias increase. Figure 6 shows the RMSE trend with the related bias-variance decomposition calculated on the test dataset using the model with the GA optimized parameters $k = 1$ and $\delta = 13$ and the related value of $N_w = 72$ which allows minimizing both errors. The value $N_w = 72$ means that the described filter behaves as CA for the flights $N \leq 72$ and as MA for flights $N > 72$. It can be further noticed that the error is approximately constant as the N_w value increases. Consequently, the use of the LSS model described in Eq. (8) could be chosen and it is equivalent to maximizing the value of N_w (i.e., using only an online CA).

Finally, the predicted total life trends and the related histograms in Figure 7 show an example of how the proposed smoothing method allows considerably improving the precision of RUL predictions, substantially reducing the noise. This amelioration is observable since the mean value (explanatory of the bias) and the variance differs as the methods vary. The proposed method has been applied both using the general CMA form as in Eq. (11), and applying the CA form (which corresponds to Eq. (8) and represents a specific case where all the predictions are considered for the smoothing). Both the general CMA and the characteristic CA methods allow considerably reducing the variance while keeping the average value approximately equal.

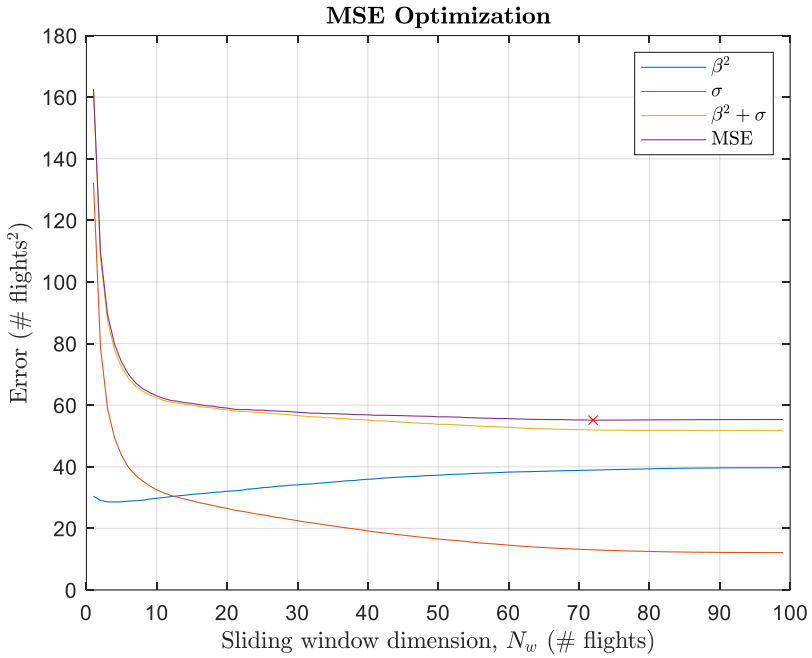


Figure 6 – MSE and bias-variance trade-off trends with $k = 1$ and $\delta = 13$ and the related identification of the optimal value $N_w = 72$ which allow minimizing the prediction error.

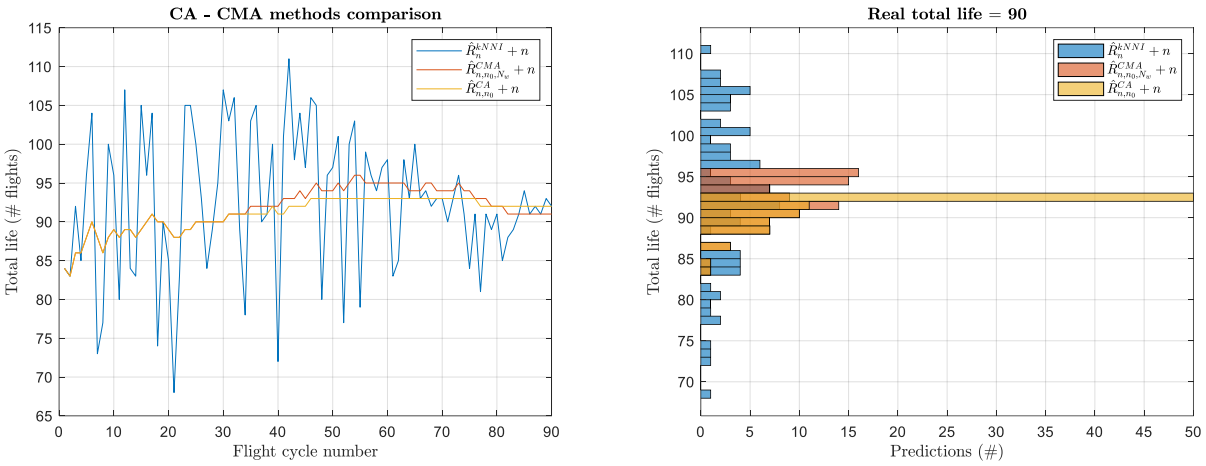


Figure 7 – On the left, an example of total life trends (i.e., sum of the predicted RUL, $\hat{R}_n$, and performed flights, n) predicted with kNNI, and smoothed through an adapted CMA ($k = 1$, $\delta = 13$) in a general (with a sliding window of length N_w) and specific form (CA). On the right, histograms of the same data which show the differences between mean values and variances of the three predictions distributions.

4.2 Optimization by means of GA hyperparameter tuning

After describing the RMSE minimization, it is necessary to optimize the k and δ values necessary for the kNNI definition. The GA allows finding the optimal values of these two parameters, and implicitly obtain the related optimal N_w value.

A global optimization algorithm as GA has been chosen because it is well suited for optimization problems with a few design variables, when the cost function is discontinuous and when global optimum research is pursued [45,46]. Nevertheless, GA can recognize local minima of the cost function. For this reason, it has been decided to use the Monte Carlo method to make the cost function converging and to determine the optimal parameters. In particular, the optimal values that have been obtained are $k = 1$, $\delta = 13$, and $N_w = 72$. This means that the kNNI will choose the most similar flight (i.e., with the shortest Euclidean distance within the multidimensional features space), filtered considering the units with +/- 13 performed flights. The best sliding window dimension with these conditions has been subsequently found equal to $N_w = 72$, which means that the generalized CMA filter behaves as an online CA if the flight under analysis $n \leq 72$ and as an online MA if $n > 72$.

5. Results

The proposed methodology is meant to easily predict the aircraft RUL in real-time with a good degree of precision. This Section reports the results extracted with the optimized parameters obtained with the generalized LSS model. These CMA results are mainly compared with the reference kNN-based method [28] optimized with an evolutionary algorithm. It is noted that the optimized parameters are consistent with those of the method developed in this article. With this last comparison, it will be possible to observe the improvements with respect to the existing methods. Given that the reference method does not contemplate the use of specific filters (such as those introduced by this work in the interpolation phase), a further comparison between the CMA results and those obtained with the improved reference method with the same filters was considered. This further comparison with the same input data highlights the substantial improvements produced by the generalized a posteriori LSS.

The results and the related comparisons have been carried out predicting the RUL on the entire test dataset, which includes 2736 flights on 36 units. The considered performance index is the RMSE, described by Eq. (12). The results obtained in terms of RUL predictions using the three considered methods (the reference method $\hat{R}_n^{ref}$, the reference method upgraded with filters $\hat{R}_{n,\delta}^{ref}$, and the proposed method $\hat{R}_{n,n_0,N_w}^{CMA}$) are reported in Table 5. The improvements generated by the CMA method are highlighted thanks to the RMSE variations. The following paragraphs report some comments on the obtained results and the generated improvements (mentioned in the Introduction) observable from several aspects.

Table 5 – Results obtained with the reference method [28] ($\hat{R}_n^{ref}$) in terms of RMSE, and comparison with the related results obtained with the same method improved with the proposed filters ($\hat{R}_{n,\delta}^{ref}$) and the proposed method (online CMA mixture filter). The percentage variations of the improved and proposed methods with respect to the reference one are reported in brackets.

Unit phases	**Reference method [28] (#flights)**	**Filter-improved reference method (#flights)**	**Proposed method (#flights)**
Overall	16.07	12.75 ***(-20.7 %)***	7.43 ***(-53.8 %)***
I Quartile	18.31	14.76 ***(-19.4 %)***	8.88 ***(-51.5 %)***
II Quartile	15.64	14.73 ***(-5.8 %)***	7.15 ***(-54.3 %)***
III Quartile	15.67	10.89 ***(-30.5 %)***	6.65 ***(-57.6 %)***
IV Quartile	11.09	8.50 ***(-23.3 %)***	5.71 ***(-48.5 %)***

First of all, the main amelioration brought about by the present method consists in the RUL prediction precision compared to the literature methods, as shown in Figure 8. Indeed, the proposed model allows minimizing the real-time prediction error by means of the LSS method (generalized as an adapted CMA mixture filter for this specific application), which substantially corrects the kNNI-predicted RUL thanks to the N_w previous flights results. In particular, the developed method allows reducing the RMSE on the RUL estimation by 54% on average with respect to the original reference method ($\hat{R}_n^{ref}$). The RMSE decrease could be also interpreted as a noise reduction in terms of oscillations on the prediction. In addition, comparing the CMA with respect to the reference method applied to the filtered dataset ($\hat{R}_{n,\delta}^{ref}$), it is possible to note that the proposed smoothing alone improves the RMSE on the RUL prediction on average. As initially mentioned, this means that the proposed smoothing – which acts in the time domain – allows improving the performance with respect to the invariant time method, which acts in the multivariate space domain.

It is also worth to highlight that the CMA method has the advantage of increasing the precision for low RUL values, which concern the life phase of a mechanical system of greater criticality and relevance in the prognostic field. In this way, the closer you get to the end of a unit's life (i.e., the life phase with the highest criticality in terms of safety and maintenance), the greater the accuracy. For instance, this can be seen in Figure 5, where the flight-by-flight prediction of one unit of the test dataset is shown. It can be noted that, as the RUL decreases, the error decreases. This aspect can be observed in Table 5, where the RMSE on the predictions were calculated both considering the entire unit life and dividing it into quartiles. Indeed, it is possible to observe how the amelioration of the prediction improves even further at low RULs (i.e., III-IV quartiles). Nevertheless, the RMSE percentage reductions that the present method allows in comparison to the two benchmarks do not attain the maximum value in the last quartiles. This is explained by the fact that the very same benchmark prediction is more precise as the RUL decreases due to the data distribution in the multivariate space.

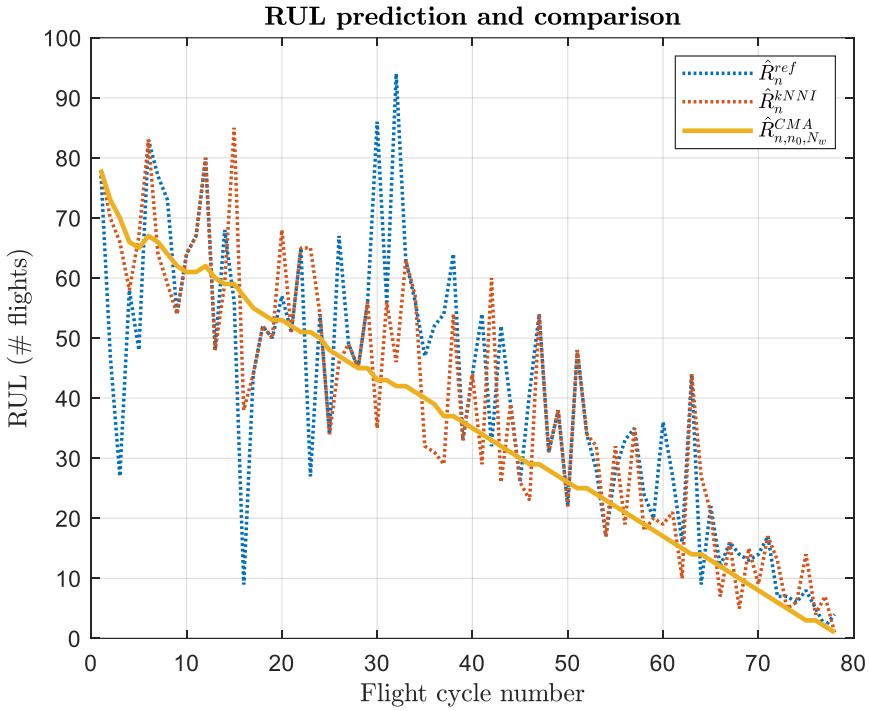


Figure 8 – Comparison of RUL prediction for a test dataset unit obtained with the original reference method ($\hat{R}_n^{ref}$), the reference method implemented with the proposed filters ($\hat{R}_{n,\delta}^{ref}$) and the generalized CMA ($\hat{R}_{n,n_0,N_w}^{CMA}$) method.

6. Conclusions

In this paper, a data-based approach has been developed for the RUL prediction of an aircraft engine. The method consists of the integration of a kNNI improved by applying the LSS principle and optimized by GA. It allows to obtain real-time RUL values with an online correction based on previous flights, which permits reducing the error. The correction obtained with the LSS both reduces the prediction error and achieves an improved real-time estimate. In addition, the LSS method has been generalized for this specific case as an adapted online CMA mixture filter, which uses an optimized sliding window of length N_w. The results obtained with the latter method show an overall RMSE of about 7 cycles. Compared to kNN-based methods existing in the literature which use an a priori multivariate space smoothing, the proposed procedure reduces the RMSE by about 54% on average (where the application of the smoothing alone generates a significant improvement). The methods implementation is not complex and allows obtaining greater accuracy for flights with low RULs. This means that the error decreases in the more critical phase in terms of safety and maintenance (i.e., flights close to the end of life of each unit). This is evident from the results calculated on the predictions divided into quartiles.

In addition to the performance improvements, the proposed method is theoretically less subject to the curse of dimensionality as it does not need to calculate k distances to evaluate the weights to be adopted in the kNN-based reference method. Therefore, in cases where the optimal k value is greater than 1, the present method would be subject to a lower computational effort.

A further advantage of the proposed method consists in the independence of smoothing from the k parameter, while the reference method only generates a smoothing restricted to the optimal k value (the lower the optimal k value, the more the smoothing will be consequently reduced). On the other hand, the proposed method allows performing an a posteriori smoothing and, hence, is not dependent on the k parameter choice. Furthermore, the proposed method is highly flexible since its three parameters (k, δ, N_w) are automatically optimized according to the specific application.

To conclude, it should be remembered that the present method further allows considering multiple types of damage thanks to the recognition of different failure paths in the multivariate space, unlike most of the methods existing in the literature which assume the presence of a single failure category.

Author Contributions:

Conceptualization, L.V. and A.D.P.; methodology, L.V. and A.D.P.; software, L.V.; validation, L.V., A.D.P., A.F. and L.G.; formal analysis, L.V.; investigation, L.V. and A.D.P.; resources, L.V. and A.D.P.; data curation, L.V.; writing—original draft preparation, L.V.; writing—review and editing, L.V., A.D.P. and A.F.; visualization, L.V.; supervision, A.F. and L.G.; project administration, A.F. and L.G.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement:

The "Turbofan Engine Degradation Simulation" dataset was downloaded online at https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan-2 [M. Chao, C.Kulkarni, K. Goebel and O. Fink (2021). "Aircraft Engine Run-to-Failure Dataset under real flight conditions", NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA].

Conflicts of Interest:

The authors declare no conflict of interest.

Bibliography

[1] J.L. Kerrebrock, Aircraft engines and gas turbines, MIT press, 1992.

[2] A.P. Daga, A. Fasana, L. Garibaldi, S. Marchesiello, Big Data management: A Vibration Monitoring point of view, in: 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, IEEE, 2020: pp. 548–553.

[3] A.P. Daga, L. Garibaldi, Machine vibration monitoring for diagnostics through hypothesis testing, Information. 10 (2019) 204.

[4] F. Castellani, L. Garibaldi, A.P. Daga, D. Astolfi, F. Natili, Diagnosis of faulty wind turbine bearings using tower vibration measurements, Energies. 13 (2020) 1474.

[5] F. Natili, A.P. Daga, F. Castellani, L. Garibaldi, Multi-Scale Wind Turbine Bearings Supervision Techniques Using Industrial SCADA and Vibration Data, Applied Sciences. 11 (2021) 6785.

[6] A.P. Daga, A. Fasana, S. Marchesiello, L. Garibaldi, The Politecnico di Torino rolling bearing test rig: Description and analysis of open access data, Mechanical Systems and Signal Processing. 120 (2019) 252–273.

[7] A.P. Daga, L. Garibaldi, C. He, J. Antoni, Key-Phase-Free Blade Tip-Timing for Nonstationary Test Conditions: An Improved Algorithm for the Vibration Monitoring of a SAFRAN Turbomachine from the Surveillance 9 International Conference Contest, Machines. 9 (2021) 235.

[8] A.P. Daga, L. Garibaldi, L. Bonmassar, Turbomolecular high-vacuum pump bearings diagnostics using temperature and vibration measurements, in: 2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT), IEEE, 2021: pp. 264–269.

[9] L. Viale, A.P. Daga, A. Fasana, L. Garibaldi, From Novelty Detection to a Genetic Algorithm Optimized Classification for the Diagnosis of a SCADA-Equipped Complex Machine, Machines. 10 (2022) 270. https://doi.org/10.3390/machines10040270.

[10] A.P. Daga, A. Fasana, L. Garibaldi, S. Marchesiello, A. Moshrefzadeh, Fast Computation of the Autogram for the Detection of Transient Faults, in: European Workshop on Structural Health Monitoring, Springer, 2020: pp. 469–479.

[11] A.P. Daga, L. Garibaldi, A. Fasana, S. Marchesiello, Performance of envelope demodulation for bearing damage detection on cwru accelerometric data: Kurtogram and traditional indicators vs. targeted a posteriori band indicators, Applied Sciences (Switzerland). 11 (2021). https://doi.org/10.3390/app11146262.

[12] F. Ahmadzadeh, J. Lundberg, Remaining useful life estimation, International Journal of System Assurance Engineering and Management. 5 (2014) 461–474.

[13] M. Arias Chao, C. Kulkarni, K. Goebel, O. Fink, Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics, Data. 6 (2021) 5.

[14] X.-S. Si, W. Wang, C.-H. Hu, D.-H. Zhou, Remaining useful life estimation–a review on the statistical data driven approaches, European Journal of Operational Research. 213 (2011) 1–14.

[15] H.M. Elattar, H.K. Elminir, A.M. Riad, Prognostics: a literature review, Complex & Intelligent Systems. 2 (2016) 125–154.

[16] J. Guo, Z. Li, M. Li, A review on prognostics methods for engineering systems, IEEE Transactions on Reliability. 69 (2019) 1110–1129.

[17] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, N. Zerhouni, Particle filter-based prognostics: Review, discussion and perspectives, Mechanical Systems and Signal Processing. 72 (2016) 2–31.

[18] C. Ordóñez, F.S. Lasheras, J. Roca-Pardinas, F.J. de Cos Juez, A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines, Journal of Computational and Applied Mathematics. 346 (2019) 184–191.
[19] Z. Zhao, B. Liang, X. Wang, W. Lu, Remaining useful life prediction of aircraft engine based on degradation pattern learning, Reliability Engineering & System Safety. 164 (2017) 74–83.
[20] J. Liu, F. Lei, C. Pan, D. Hu, H. Zuo, Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion, Reliability Engineering & System Safety. 214 (2021) 107807. https://doi.org/10.1016/j.ress.2021.107807.
[21] C. Zheng, W. Liu, B. Chen, D. Gao, Y. Cheng, Y. Yang, X. Zhang, S. Li, Z. Huang, J. Peng, A data-driven approach for remaining useful life prediction of aircraft engines, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018: pp. 184–189.
[22] R.V. Yampolskiy, Unpredictability of AI: On the impossibility of accurately predicting all actions of a smarter agent, Journal of Artificial Intelligence and Consciousness. 7 (2020) 109–118.
[27] A. Ramamoorthy, R. Yampolskiy, Beyond mad? the race for artificial general intelligence, ITU J. 1 (2018) 1–8.
[28] Y. Zhou, M. Huang, M. Pecht, Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization, Journal of Cleaner Production. 249 (2020) 119409.
[29] O. Kramer, K-nearest neighbors, in: Dimensionality Reduction with Unsupervised Nearest Neighbors, Springer, 2013: pp. 13–23.
[30] C. Sammut, G.I. Webb, Encyclopedia of machine learning, Springer Science & Business Media, 2011.
[31] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools and Applications. 80 (2021) 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.
[32] B. Berthelot, E. Grivel, P. Legrand, New Variants of DFA Based on Loess and Lowess Methods: Generalization of the Detrending Moving Average, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021: pp. 5140–5144. https://doi.org/10.1109/ICASSP39728.2021.9414216.
[33] J. Babaud, A.P. Witkin, M. Baudin, R.O. Duda, Uniqueness of the Gaussian Kernel for Scale-Space Filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-8 (1986) 26–33. https://doi.org/10.1109/TPAMI.1986.4767749.
[34] N. Raghavan, D.D. Frey, Remaining useful life estimation for systems subject to multiple degradation mechanisms, in: 2015 IEEE Conference on Prognostics and Health Management (PHM), IEEE, 2015: pp. 1–8.
[35] R. May, J. Csank, T. Lavelle, J. Litt, T.-H. Guo, A high-fidelity simulation of a generic commercial aircraft engine and controller, in: 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2010: p. 6630.
[36] D.K. Frederick, J.A. DeCastro, J.S. Litt, User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS), (2007).
[37] S. Kim, N.H. Kim, J.-H. Choi, Prediction of remaining useful life by data augmentation technique based on dynamic time warping, Mechanical Systems and Signal Processing. 136 (2020) 106486.
[38] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdisciplinary Reviews: Computational Statistics. 2 (2010) 433–459.
[39] C.L. Lawson, R.J. Hanson, Solving least squares problems, SIAM, 1995.

[40] J.G. Everett, S.H. Farghal, Data Representation for Predicting Performance with Learning Curves, Journal of Construction Engineering and Management. 123 (1997) 46–52. https://doi.org/10.1061/(ASCE)0733-9364(1997)123:1(46).
[41] F.R. Johnston, J.E. Boyland, M. Meadows, E. Shale, Some properties of a simple moving average when applied to forecasting a time series, Journal of the Operational Research Society. 50 (1999) 1267–1271.
[42] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, IEEE, 2008: pp. 1–9.
[43] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation. 4 (1992) 1–58.
[44] O. Maimon, L. Rokach, eds., Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2010. https://doi.org/10.1007/978-0-387-09823-4.
[45] G. Venter, Review of optimization techniques, Encyclopedia of aerospace engineering. (2010). https://scholar.sun.ac.za:443/handle/10019.1/14646 (accessed April 29, 2022).
[46] M. Sreedhar, S. Reddy, S.A. Chakra, T.S. Kumar, S.S. Reddy, B.V. Kumar, A review on advanced optimization algorithms in multidisciplinary applications, in: Recent Trends in Mechanical Engineering, Springer, 2020: pp. 745–755.