Features clustering and XAI for Remaining useful life estimation using LSTM

YOUNESS Genane[1,2], AALAH Adam[3]

[1] *Laboratoire LINEACT CESI, Nanterre, IDFC*
gyouness@cesi.fr
[2] *Laboratoire Cedric-MSDMA, Paris, France*
genane.youness@lecnam.net

[3] *Institut polytechnique de Paris, Palaiseau, France*
aalah.adam@gmail.com

ABSTRACT

Prognosis and health management (PHM) depend on sufficient prior knowledge of the degradation process of critical components to predict the Remaining Useful Life (RUL). This task is composed of two phases: learning and prediction. The first phase uses the available information to learn the system's behaviour. The second phase predicts future behaviour based on the available information of the system and estimates its remaining lifetime. Deep learning approaches achieve good prognostic performance but usually suffer from a high computational load. Complex feature extraction models do not solve this problem, as they lose information in the learning phase and thus have a poor prognosis for the remaining lifetime. To address this issue, a new prepossessing approach is used with feature clustering. The proposed approach allows for restructuring the data into homogeneous groups strongly related to each other using a simple architecture of Long Short-Term Neural Networks (LSTM). The use of this model is advantageous in terms of learning time and the possibility of using limited computational capabilities. Finally, we will focus on the interpretability of deep learning prognosis using explainable AI to achieve interpretable RUL prediction. Experimental results on the available NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset show the performance of the proposed model compared to other common methods.

1. INTRODUCTION

In recent years, the manufacturing sectors have taken a new approach involving technological innovations aimed to modernize and optimize production. This new concept is characterized by automizing mechanical processes using artificial intelligence. These techniques allow what is called "Industry 4.0" for the evolution of maintenance, and more particularly predictive maintenance. Due to artificial intelligence, predictive maintenance can anticipate anomalies (e.g., the life of computer servers, and the electrical installation of dysfunctional subway trains). Besides, intelligent maintenance requires a multi-step approach, from monitoring, through analysis and decision support, to validation and verification. These steps are part of a discipline called PHM (Prognostics and Health Management), linking the study of failure mechanisms and life cycle management.

Prognostics, in general, require two types of techniques: (1) an application-dependent technique that aims to detect precursors to estimate the system's health state, and (2) a prediction technique to predict the remaining useful life (RUL). Prognostic and Health Management (PHM) is an engineering field whose goal is to provide users with a thorough analysis of the health condition of a machine and its components [Lee et al., 2014]. Relying on human operators for managing atypical events is quite difficult when dealing with complex equipment and attempting to retrieve information and patterns of different equipment failures. It aims to better manage the health status of physical systems while reducing operation and maintenance costs [Zio, 2012]. The implementation of PHM solutions is becoming increasingly important, and the prognostic process is now considered one of the main levers of action in the quest for overall performance [Saxena et al., 2010]. It is generally described as a combination of seven layers that can be divided into three phases: observation, analysis, and action.

The Remaining useful life (RUL) is the length of time a machine is likely to operate before it requires repair or replacement. By taking RUL into account, maintenance can be scheduled in order to optimize operating efficiency and avoid unplanned downtime. For this reason, estimating RUL is a top priority in predictive maintenance programs. Based on

ageing models (degradation of the monitored system), prognostics determine the health status of a system and predict the RUL. This prediction can be obtained using a physical model-based approach, a data-driven approach, and a new hybrid approach [Karasu and Altan, 2021] merging the first two approaches. Different works have shown the effectiveness of data-driven approaches [Gugulothu et al., 2018] and especially deep learning, in estimating the RUL of a turbofan engine. A turbofan engine consists of several key components including a fan, compressor, combustion chamber, turbine, exhaust nozzle and bypass duct as well as various sensors to understand the condition of the engine. One of the advantages of data-driven approaches over the physics-based approach is that they can take into account the complexity of the structure and variety of sensors, which can affect the RUL of the turbofan engine, including operating conditions, maintenance history, and design features of the engine. To predict the RUL of a turbofan engine, several algorithms have been used such as multilayer perceptrons, long short-term memories (LSTMs), bidirectional LSTMs (Bi-LSTMs), and convolutional neural network (CNN) algorithms, as well as combined algorithms to achieve higher accuracy. However, the accuracy of its predictions depends on the quality and relevance of the data that influence the engine's RUL. It is difficult to build accurate models with noisy data or complex features and to identify patterns. To address the challenge a number of techniques can be used such as dimensionality reduction which reduces the number of dimensions, and feature selection which identify the most relevant features. In this paper, a deep learning method based on a feature clustering approach is proposed using a single hidden layer LSTM neural network, offering better accuracy than considerably complex approaches. Learning and validation are performed using the NASA C-MAPSS dataset. The effects of some critical parameters are analyzed with 30 random independent trials. The accuracy obtained is then compared to that provided by all sensors and the Principal Component Analysis (PCA) dimension reduction technique. Another key contribution of this paper is to highlight the benefits of explainable AI in the prognostic estimation of the RUL of turbofans. To this end, explainable AI is adopted through two phases of the study: first, during preprocessing with the use of feature clustering to allow for a better understanding of the relationships between features. This preprocessing step could be considered as part of the explainable AI domain, since it promotes a better comprehension of the model. Second phase, in the post-model, Shapley's Additive Explanation (SHAP) is applied for the interpretation of the RUL prediction estimate and the determination of the cause of the engine failure.

To present our work, this paper is organized as follows: Section 2 explains XAI's need in PHM and reviews related work in this field. Section 3 briefly describes the techniques used in this study. An overview of the used data is illustrated in Section 4. Section 5 provides an overview of previous related work to our study. Section 6 is devoted to the proposed framework, experiments, results, and discussions. Finally, Section 7 concludes the article.

2. XAI'S NEED IN PHM

System prognostics is a safety-sensitive industry field. It is therefore crucial to ensure the use of properly regulated AI in this area. Deep learning networks are artificial neural networks that include many hidden layers between the input and the output layers. These approaches often offer better performance but are accused of being black boxes. They suffer from a lack of interpretability and analysis of prediction results. It is difficult to know how information is processed in the hidden layer. Therefore explanatory capabilities are needed in RUL prediction to improve system reliability and provide insight into the parts that caused the engine failure. Explainable AI (XAI) has been presented as a solution to this problem. It is able to analyze the black box inside, verify its processes and provide an understandable explanation of the logic behind the prediction. Several approaches can be applied to explainable AI systems for prediction [Burkart and Huber, 2021]. [Carvalho et al., 2019] classifies the interpretability methods in 3 groups according to the moment when these methods are applicable: before (pre-model), during (in-model) or after (post-model) the construction of the model to predict. Some authors [Tjoa and Guan, 2021], [Carvalho et al., 2019] consider that PCA, Distributed Stochastic Neighbor Embedding (t-SNE), and clustering methods can be classified under pre-model interpretability and be part of the XAI field. The field of in-model interpretability is mainly focused on intrinsically interpretable models that are specifically designed to be transparent, such as decision trees, Rule-based models, Bayesian models, and Hybrid models. Post-model interpretability refers to the improvement of interpretability after a model has been built (post hoc); Two approaches can be model-specific or model-agnostic. Model-specific interpretability includes self-explanation as the additional output such as attention mechanisms [Lai et al., 2022] and capsule network (CapsNets) [Sabour et al., 2017] that focus on the most relevant parts of the input data when making a prediction. Other examples of model-specific interpretability techniques include saliency maps and layer-wise relevance propagation. Post-hoc model-agnostic interpretability approach consists to analyse root causes by manipulating inputs and visualizing outputs without relying on the trained model. These approaches include local model analysis, global model analysis, and feature importance analysis. These methods can be used to identify the input features that contribute to the predictions and have the greatest impact on the output of the model. In this study, Shapley additive explanation (SHAP) [Lundberg and Lee, 2017] is used to identify the key features that affect the RUL of a turbofan engine.

3. METHODOLOGY

3.1. Feature clustering: ClustOfVar

As mentioned previously, feature clustering can be classified as pre-model interpretability and part of the XAI domain. It facilitates visualization and understanding of relationships between features and pattern identification. Feature clustering obtains clusters of related and redundant features. It creates synthetic features that best summarize the relevant information provided by the initial features. This approach can be considered as a dimension reduction step and seems to be a good alternative to the principal component analysis. Indeed, it allows for removing the redundancies of information in the p available features. The synthetic features will be constructed only with the features of the group, contrary to the principal component analysis where the synthetic features are constructed with all the features. It thus makes it easier to interpret the synthetic features. The homogeneity criterion of a cluster is the squared correlations summarizing the features in the cluster. The ascendant hierarchical clustering method is used. At each stage, the two clusters that minimize the loss of homogeneity are aggregated. Each feature is then reassigned to the cluster that is closest to it, i.e. the one with the highest squared correlation between the feature and the cluster representative. To evaluate the stability of the partition, a bootstrap approach is used; it allows the determination of suitable numbers of clusters. ClustOfVar approaches [Chavent et al., 2012] implemented in the R packages are used.

3.1.1. Definition of the synthetic feature of the cluster

Each cluster C_k can be summarized by a numerical synthetic feature, noted y_k defined as follows:

$$\mathbf{y}_k = \arg\max_{\mathbf{u}\in\mathbb{R}^n} \sum_{j\in C_k} r^2\left(\mathbf{x}_j, \mathbf{u}\right) \tag{1}$$

$r^2\left(x_j, u\right) \in [0,1]$ is the square of the linear Pearson correlation between the feature x_j and the synthetic feature u.

3.1.2. Homogeneity H of the cluster C_k

The homogeneity H of the partition P_k = (C_1,..., C_k) is defined as the sum of the homogeneities of its k clusters :

$$\mathcal{H}\left(P_K\right) = \sum_{k=1}^{K} H\left(C_k\right) \tag{2}$$

The goal of the hierarchical ascending classification algorithm is to find a partition of a set of p features that maximizes the homogeneity H(P_k).

3.1.3. The hierarchical clustering algorithm

The aim is to find a partition that maximizes the homogeneity function H defined in (2) in such a way that the features within a cluster are strongly related to each other. The hierarchical bottom-up classification algorithm starts with the partition into p clusters At each step, the two classes A and B that minimize the loss of homogeneity are aggregated. Thus, the two clusters with the smallest dissimilarity d are chosen. It is defined as follows:

$$d(A,B) = H(A) + H(B) - H(A \cup B) \tag{3}$$

Using this aggregation measure, the new partition maximizes H among all partitions obtained by aggregating two clusters.

3.1.4. Stability of feature partitions

To assess the stability of all partitions into 2 to p-1 classes resulting from the hierarchical clustering, a bootstrap resampling procedure is used to determine the appropriate number of feature classes that can be retained for analysis. The right choice for this number of classes is the one that retains the most stable partition. B bootstrap replications of the original data are generated and the associated B dendrograms are obtained. The partitions of these B dendrograms are compared to the partitions of the original hierarchy through the adjusted Rand index [Hubert and Arabie, 1985]. The stability of the partition is evaluated based on the average of the values of the adjusted B Rand indices. The closer the Rand index value is to 1, the more similar the partitions are.

3.2. Long Short-Term Memory Neural Network

A type of recurrent neural network has been developed by [Hochreiter and Schmidhuber, 1997], Long Term Memory, LSTM, becoming a powerful model for processing temporal data with complex structures. It was a solution to the problem of the optimal gradient of the back-propagation to change the weights of the network. This is a promising approach to service life prediction, providing valuable short and long-term information. The LSTM network is shown in Figure 1 for a unit at step t with no indication of weight.

Each unit is linked to a hidden state h and to a state c of the cell that serves as a memory. Three gates regulating the flow of information represent the structure of the LSTM network: the input gate allows or blocks the update of the cell c, the forget gate allows the reset of the cell state and the output gate controls the communication from the cell c to the output of the unit. The activation function used is the hyperbolic tangent function. It transforms the values between -1 and 1 and prevents the outputs from becoming large.

$$
\begin{aligned}
F_t &= \sigma\left(W_F x_t + U_F h_{t-1} + b_F\right) \quad \text{(forget gate)} \\
I_t &= \sigma\left(W_I x_t + U_I h_{t-1} + b_I\right) \quad \text{(input gate)} \\
O_t &= \sigma\left(W_O x_t + U_O h_{t-1} + b_O\right) \quad \text{(output gate)} \\
c_t &= F_t \circ c_{t-1} + I_t \circ \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \\
h_t &= O_t \circ \tanh\left(c_t\right) \\
o_t &= f\left(W_o h_t + b_o\right)
\end{aligned} \tag{4}
$$

The inputs of each gate are weighted by the gate weights and by a bias. There are 4 weight matrices whose dimensions depend on the dimensions of h_{t-1} and x_t:
W_F: weight of the forget gate.
W_I: weights of the input gate.
W_c: weights of the data that will be combined with the input gate to update the cell state.
W_o: weights of the output gate.

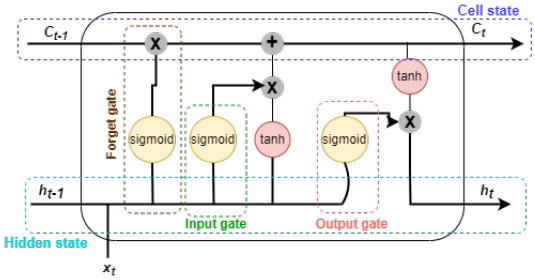

Figure 1. Structure of an LSTM network Unit.

In our proposed model, we used one LSTM layer to extract the temporal synthetic features included in the previous time windows with size T_w before estimating the RUL.

3.3. SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive Explanations) proposed by [Lundberg and Lee, 2017], is a method used to improve the interpretability of artificial intelligence methods. Issued from Game Theory, it computes Shapley values [Shapley, 1953] to each predictor to indicate its contribution to the final result. The idea is to average the impact of a feature over all possible combinations of features in a linear model. The Shapley value ϕ_j is defined as follows:

$$
\phi_j(f,x) = \sum_{S \subseteq \{x_1,..,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} \left(\mathrm{f_x}\left(S \cup \{j\}\right) - \mathrm{f_x}(S)\right) \tag{5}
$$

With p the number of all features, S is a subset of features used in the model, and j is the j^{th} feature. f_x the prediction function of a single input x defined as:

$$
\mathrm{f_x}(S) = \int \hat{f}\left(x_1, \ldots, x_p\right) dP_{x \notin S} - E_X\left(\hat{f}(X)\right) \tag{6}
$$

It performs several integrations for each feature that does not contain S. A linear g model is finally fitted to the features and their impacts, represent as follows:

$$
f(x) = g\left(x'\right) = \phi_0 + \sum_{i=1}^{p} \phi_i x_i \tag{7}
$$

with $f(x)$ representing the original model, $g(x)$ is the explanation model, $x'-$ is simplified input, such that $x = h_x\left(x'\right)$, it has several omitted features, $\phi_0 = f\left(h_x(0)\right)$, it represents the model output with all simplified inputs missing. Shapley values are the only set of values that satisfy the following three properties: local accuracy, missingness, and consistency.

4. DATA OVERVIEW

NASA has developed a generic military turbofan engine simulation [Frederick et al., 2007], the Modular Aero-Propulsion System Simulation (C-MAPSS), which has been released for public use to rapidly implement and test control algorithms on a validated engine model. C-MAPSS simulates an engine model of the 90,000 lb thrust class and the package includes an atmospheric model capable of simulating operations at an altitude ranging from sea level to 40K ft, Mach number from 0 to 0.90, and sea-level temperatures from -60 to 100°F.

Table 1. Operational conditions at which the data was simulated

Operational Conditions

ID	Name	Alt, ft	Mach	TRA, deg	Fan speed, rpm	Core speed, rpm	Fuel Flow, pps	T48, °R
1	FC09	42K	0.84	100	2212	8317	1.518	1744
2	FC08	35K	0.84	100	2223	8346	2.120	1750
3	FC07	25K	0.62	60	1915	8006	1.670	1534
4	FC06	20K	0.70	100	2324	8719	3.863	1909
5	FC05	10K	0.25	100	2319	8774	4.661	1947
6	FC01	0K	0	100	2388	9051	6.835	2072

The C-MAPSS engine contains five rotating components. These components can be divided into two subcategories a low-pressure shaft that contains: Fan, Low-Pressure Compressor (LPC), a low-pressure turbine (LPT), and a high-pressure shaft that contains: a high-pressure Compressor (HPC) and a high-pressure turbine (HPT).

To comprehend how the engine operates, we will have to understand each component and its role, Figure 2. The fan is the first component located at the entrance of the turbofan engine. The rotation of its blades causes the suction of large quantities of air. Followed by the compressor, being the first component in the engine core. It contains two sub-components low-pressure compressor (LPC) and a high-pressure compressor (HPC). The compressed air obtained is then pumped

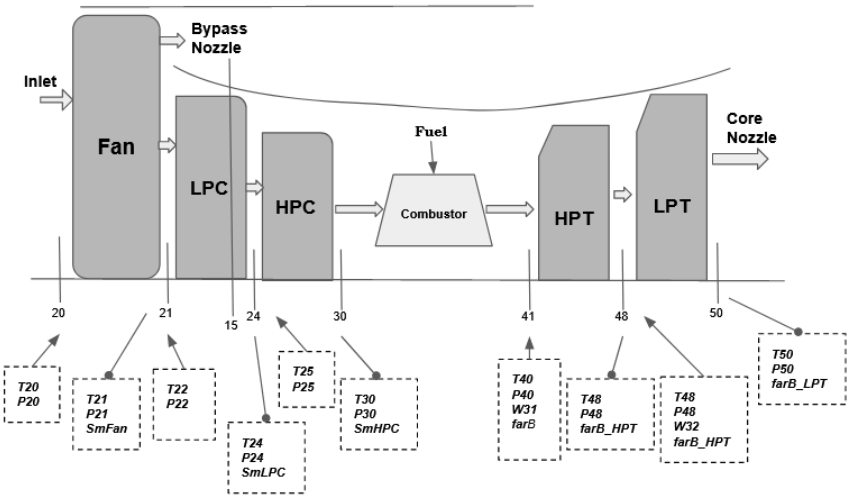

Figure 2. C-MAPSS engine components.

into the combustion chamber, whereabouts the injected fuel is mixed and burnt. This provides a high-temperature airflow that is supplied to the turbine. The turbine contains two sub-components a high-pressure turbine (HPT) and a low-pressure turbine (LPT). The high-energy airflow comes out of the combustor chamber into the turbine, causing the turbine's blades to rotate. The air sucked into the turbofan engine, is split into two parts, the first part goes to the combustor chamber passing through the compressor and the second part goes through the bypass. The bypass role is to increase thrust without increasing fuel consumption, as well as to provide cooling air.

Table 2. Simplified engine's five rotating components' inputs and outputs

	input sensors	output sensors
Fan	T20,P20	T21 , P21, SmFan
LPC	T22 , P22	T24 , P24 , SmLPC
HPC	T25 , P25	T30, P30, SmHPC
HPT	T40, P40, W31, farB	T48, P48 ,farB$_{HPT}$
LPT	T48, P48, W32, farB$_{HPT}$	T50 , P50 , farB$_{LPT}$

The C-MAPSS dataset consists of four subsets: FD001, FD002, FD003 and FD004 detailed in Table 3, it is considered the benchmark for predictive maintenance problems. In each subset the data records a collection from turbofan engine sensors at each time cycle, which includes 26 columns in total: the first column represents engine ID, the second column represents the current time cycle, columns 3 to 5 are the three operational conditions that have a substantial effect on engine performance, the remaining columns from 6 to 26 represent the 21 sensors, displayed in Table 4, these sensor measurements are contaminated with noise. The engine is operating normally at the start of each time series and develops a fault at some point in time. The fault, then, grows in magnitude until the engine reaches failure.

Table 3. Detailed information of the C-MAPSS subsets

	FD001	FD002	FD003	FD004
Units (Train)	100	260	100	249
Units (Test)	100	259	100	248
Operation condition	1	6	1	6
Fault modes	1	1	2	2

In this work, in order to predict the RUL, we choose the subset FD004, a complex structured subset in C-MAPSS having two faults and six operation conditions.

5. RELATED WORK

The C-MAPSS dataset is one of the most used datasets for the goal of improving RUL estimations. Whether using machine learning models or deep learning models, many approaches and methods were developed and proposed throughout the years. [Wang et al., 2022] built a fusion model that extracts features from the data based on a Broad Learning System (BLS) and integrates LSTM to process time series information. [Heimes, 2008] was the first to implement a recurrent neural network (RNN) for RUL prediction. [de Oliveira da Costa et al., 2020] proposed a Domain Adversarial Neural Network (DANN) approach to learning domain-invariant features using LSTM. [Jiang et al., 2020] used a Fusion network combined With a Bidirectional LSTM Network to estimate the RUL with the use of sequenced data. [Zhao et al., 2020] used a Bidirectional LSTM (BiLSTM) approach for RUL estimation by taking sequence data in bi-direction, a model optimization was necessary to obtain the best results possible. [Zhang et al., 2020] proposed an LSTM-Fusion architecture, concatenating separate LSTM sub-networks, on sensor signals with feature window sizes. [Listou Ellefsen et al.,

Table 4. C-MAPSS 21 sensor readings

Sensor readings		
Symbol	Description	Unit
T20	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P20	Pressure at fan inlet	psia
P15	Total Bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P20)	–
Ps30	Static pressure at HPC outlet	psia
ϕ	Ratio of fuel flow to Ps30	pps/psia
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	–
farB	Burner fuel-air ratio	–
htBleed	Bleed enthalpy	–
Nf_{dmd}	Demanded fan speed	rpm
$PCNfR_{dmd}$	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

2019] proposed a semi-supervised approach based on LSTM using a Genetic Algorithm (GA) to adjust the diverse amount of hyper-parameters in the training procedure. [Zheng et al., 2017] used a deep LSTM model for RUL estimation feeding sequenced sensor data to the model to reveal hidden features when multiple operational conditions are present. [Lee, 2008] adopted a new approach by normalizing the sensor data per operational condition, as well as rectifying the RUL; it is done by setting an early RUL removing the initial cycles where the engine is at a healthy state and only focusing on the degradation period. Recently, [Palazuelos et al., 2020] extend capsule neural networks for fault prognostics, particularly remaining useful life estimation. [Chen et al., 2020] applied an attention-based deep learning framework hybrid LSTM with feature fusion able to learn the importance of features. [Qin et al., 2022] proposed a Slow-varying Dynamics assisted Temporal CapsNet (SD-TemCapsNet) to simultaneously learn the slow-varying dynamics and temporal dynamics from measurements for accurate RUL estimation. [Li et al., 2022] proposed a cycle-consistent learning scheme to obtain a new representation space, considering variations in the degradation patterns of different entities. [Ren et al., 2022] proposed a lightweight and adaptive knowledge distillation for enhancing industrial prediction accuracy. To address the sensor malfunction problem, [Li et al., 2023] introduced adversarial learning to extract generalized sensor-invariant features.

6. EXPERIMENTS AND RESULTS

In this section, we explain the experiments using the proposed model to predict the RUL. The Flowchart of the proposed methodology is described in Figure 3. The experimental environment is DELL-Intel Core vPRO i9, 64GB RAM, 16GB GPU NVIDIA RTX A5000, and Microsoft 10 operating system. The models and algorithms were implemented in python language, with an exception when using the R language for the ClustOfVar package to compute the feature clustering. The frameworks used are TensorFlow and Keras on Jupyter-Notebooks IDE. The experiment's code is available on GitHub [1].

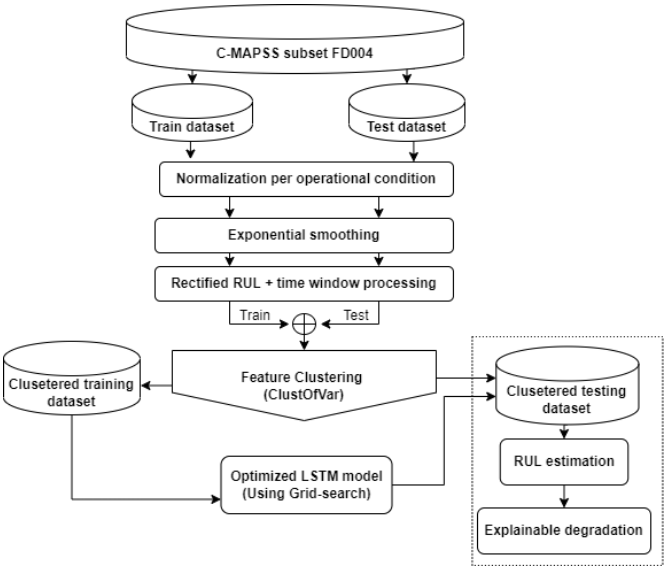

Figure 3. Flowchart of the proposed LSTM framework.

As mentioned, we chose to work with the C-MAPSS subset FD004, as it is the most complex dataset out of four subsets, containing 2 fault modes and 6 operational conditions (see Table 3). The experiments can be conducted through three stages: Pre-processing, Training and RUL estimation, and then explaining the degradation.

6.1. Pre-processing approach

6.1.1. Normalization according to Operation condition

Since FD004 was simulated under six different operational conditions [Lee, 2008], normalization is then done to each operational condition. The operational condition is identified by the Mach number, altitude, and ambient temperature of an aircraft engine. It is a controlled setting or environmental parameters. Normalizing the sensors to this information has a significant impact on the performance of the model. Normalized values are found within the range [-1,1] using the

[1] https://github.com/adam-aalah/Feature-clustering-and-XAI-for-RUL-estimation

min-max normalization method:

$$\text{norm}\left(x_t^{i,j}\right) = \frac{x_t^{i,j} - \min\left(x^j\right)}{\max\left(x^j\right) - \min\left(x^j\right)} - 1 \tag{8}$$

6.1.2. Data smoothing

Normalizing the data is not enough to produce a reasonable and accurate RUL estimation due to the large amount of noise present in the data. Exponential smoothing is applied. It uses a parameter α that controls the smoothing factor. The value of α lies between 0 and 1. The fitted values are written as follows:

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1} \tag{9}$$

- α = 0: signifies those future forecasted values are the average of historical data, giving more weight to historical data.
- α = 1: signifies that future forecast values are the results of the recent observation, giving more weight to recent observations.

A value close to 1 indicates fast learning (that is, only the most recent values influence the forecasts), whereas a value close to 0 indicates slow learning (past observations have a large influence on forecasts). A grid-search was performed to find the best value of α in the range of [0.1 ; 0.9] which turned out to be equal to 0.3. However, the effect of the smoothing value α on the performance of the model will be investigated in further experiments.

6.1.3. Feature clustering

Feature clustering is illustrated on the combined pre-processed train and test sets and then carried out a hierarchical feature clustering reducing the inputs of the LSTM neural network. The stability criterion is employed based on a bootstrap approach to choose the number of clusters using the "ClustOfVar" R-package [Chavent et al., 2012].

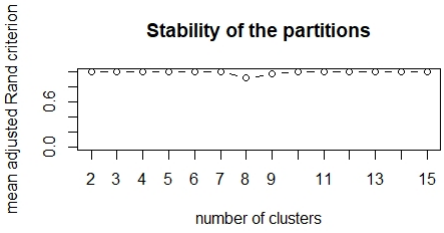

Figure 4. Stability criterion for the partitions and dendrogram obtained by the hierarchical clustering of the features.

The stability in Figure 4, is achieved up to 7 clusters and then re-stabilizes from 9 clusters. Based on the stability, the choice of 3 clusters gives a gain in cohesion equal to 66.98%, while the choice of 7 clusters gives a gain in cohesion equal to 90.09%. Therefore, it seems reasonable to keep 7 clusters, as it allows to give a deeper insight into how the engine's degradation occurs.

According to Figure 5, the sensors are distributed across the 7 clusters. These synthetic features are structured as follows:

$$\begin{aligned} cluster1 = {} & 7.856395 - 3.887295 \times T24 - 3.847008 \\ & \times T30 - 3.245412 \times T50 - 2.964724 \times Ps30 \\ & - 3.533685 \times htBleed \end{aligned} \tag{10}$$

$$cluster2 = -3.447993 + 4.611694 \times P15 \tag{11}$$

$$\begin{aligned} cluster3 = {} & -3.345230 + 5.308207 \times P30 \\ & + 5.164441 \times \phi \end{aligned} \tag{12}$$

$$cluster4 = -4.890251 + 6.789197 \times Nf + 6.588968 \times NRf \tag{13}$$

$$cluster5 = -3.642547 + 7.101891 \times Nc + 7.607409 \times NRc \tag{14}$$

$$cluster6 = -2.868859 + 8.863718 \times epr \tag{15}$$

$$\begin{aligned} cluster7 = {} & -2.891498 - 4.090116 \times BPR + 5.704820 \\ & \times W31 + 5.630470 \times W32 \end{aligned} \tag{16}$$

The synthetic feature scores for each observation are extracted. This represents the new dataset applied to predict the RUL.

6.1.4. Time window and rectified RUL

After reducing the noise using exponential smoothing, a fixed-length sliding Time Window (TW) processing is applied to convert multivariate time series data into sequential measurement cycle data. A time window is moved each time by one measurement cycle to generate a new TW. A longer length of TW can contain more local information and therefore impacts the performance of the proposed model. Simultaneously, it slows down the learning speed of the model. A grid search was performed on time window size for values $\in$ [30,60], using the LSTM model, and leads to a TW value equal to 30. The engine fails at an unknown time and is healthy before the failure, so a linear degradation model is essential for model convergence. After fixing the time window to 30, and based on different values of rectified RUL [Zheng et al., 2017], we set an early RUL value equal to 130. The engine is considered healthy until it reaches cycle 130 where degradation can be observed. Since all engines are healthy from cycle 130 onward, the remaining life of the train and test set has been rectified. This allows the model to learn from the critical cycle interval instead of the whole life of the engine

We will examine the effect of the time window size and the rectified true RUL setting on the model performance in the subsequent experiments.

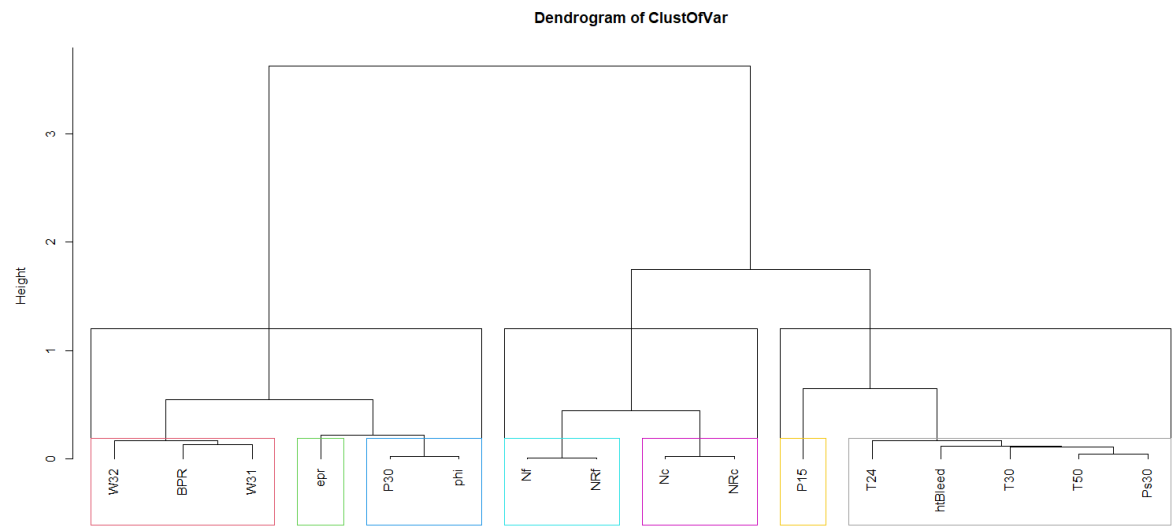

Figure 5. The hierarchical clustering of the features.

6.2. LSTM configuration and metrics

6.2.1. Hyper-parameters

The proposed model is built with a simple Long Short-Term Memory Neural Network, LSTM. Its architecture is composed of one LSTM layer with an activation 'tanh', one dense layer with an activation function 'ReLu' followed by a dropout layer with an activation layer 'ReLu' and a Dense(1) layer. The grid search cross-validation method is used for parameter optimization. The best hyper-parameters with the lowest RMSE, used in this study, are shown in Table 5.

Table 5. Hyper-parameters used for LSTM model.

Hyper-parameter	value
number of Hidden Layers	1
Nodes	128
Dropout	0.2
Batch-size	384
Epochs	20
Learning-Rate	0.01

Experiments were conducted on seven synthetic features With a time window equal to 30 and an early RUL value equal to 130. Adam optimizer was applied given the best results with a learning rate set to 0.01. For the proposed model, the LSTM uses a single hidden layer with 128 nodes, and the dropout is set to 0.2.

6.2.2. Prognostic metrics

To evaluate the model's performance two metrics are used in this paper including Root Mean Square Error (RMSE) and the Scoring function.

- RMSE evaluates the prediction error. it is represented as:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(d_i)^2} \tag{17}$$

  with d_i = RUL'_i - RUL_i

- Based on the PHM data challenge in 2008 [Saxena et al., 2008], a scoring function, S-score, was defined to penalise a lot more late predictions in comparison to early predictions of the remaining useful life. It is described as:

$$S\_score = \begin{cases} \sum_{i=1}^{N}\left(e^{-\frac{d_i}{13}}-1\right) & \text{for } d_i < 0 \\ \sum_{i=1}^{N}\left(e^{\frac{d_i}{10}}-1\right) & \text{for } d_i \geq 0 \end{cases} \tag{18}$$

  Where n denoted the total number of test sets, RUL'_i is the predicted RUL, RUL_i is the actual RUL, and d_i is the prediction error between the predicted.

6.3. Results analysis

In order to better estimate the remaining useful life of the units, we study the effect of different parameters on the model's performance, such as data smoothing parameter α, the time window size and the piecewise rectified RUL. The proposed model was used on our new pre-processed dataset

containing synthetic features. Experimental results are averaged by 30 independent trials to ensure the accuracy of the results and reduce the effect of randomness.

6.3.1. Parameters study

We first adjust the data smoothing parameter α to 0.2, 0.3, 0.7, and 0.9, consistent with previous studies. After 30 independent trials, the results are shown in Figure 6. The 95% confidence interval of the RMSE is displayed for each α value. The red dot indicates the average RMSE. The best value of the smoothing parameter α is equal to 0.3, which leads to the smallest root mean square error (RMSE).

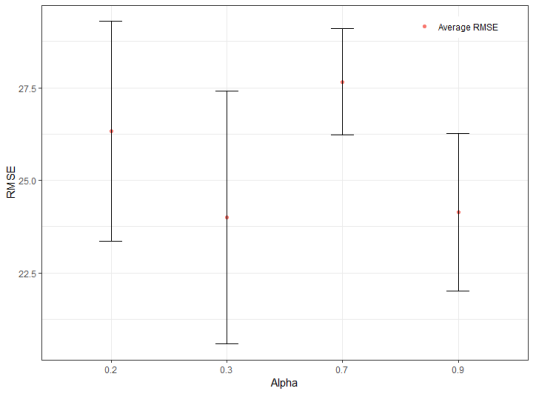

Figure 6. 95% CI for average RMSE by smoothing parameters α.

Regarding the size of the time window, most studies using the C-MAPSS FD004 database set it at 30. Nevertheless, we believe that this parameter should be investigated. Setting α to 0.3, we adjust the time window to 10, 20, 30, 40, and 50. Each configuration is repeated 30 times independently. The results in Figure 7 show that the RMSE first decreases as the window size increases and exceeds the value of 30, indicating that a window size that is too large does not help improve performance. This is because the current RUL is correlated with data from the most recent period, and the correlation decreases as the time interval increases. Including this data may result in too much noise. On the other side, the test results' variance increases with the time window's size, indicating that the model learning is indeed perturbed by noise. The best results are obtained with a time window size of 30. In addition, we study the impact of the rectified RUL as for the window size experiments. Setting a time window to 30, we adjust RUL early to 130, 140, and 150 and repeated each trial 30 times independently. The results in Figure 8 show that the lowest RMSE is achieved with a rectified RUL equal to 130.

6.3.2. Feature clustering effect

In this section, we perform an ablation study to show the effectiveness of using feature clustering in preprocessing, improving model performance and over-fitting problems by reducing the influence of input sensors. First, our results are compared to those obtained by deploying a one, two, and three-hidden-layer LSTM network without dimension reduction. Second, the results are compared to those obtained by the PCA dimension reduction method with 3 PCs, 4 PCs, and 16 PCs using a single hidden layer LSTM network. This comparison mainly shows the motivation behind using feature clustering. This approach extracts information from correlated features to reveal hidden patterns, thus providing a simpler model with fewer hidden layers. Our subsequent experiments validate this assumption.

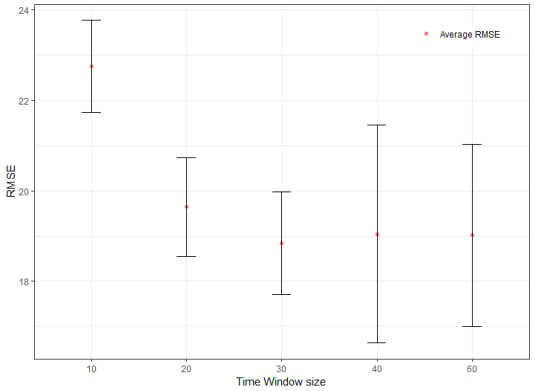

Figure 7. 95% CI for average RMSE by time window sizes.

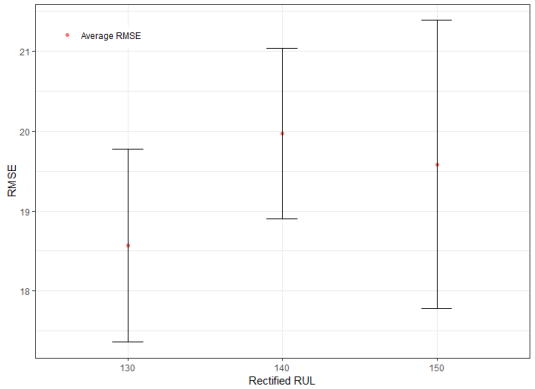

Figure 8. 95% CI for average RMSE by rectified RUL values.

The sensors were smoothed using α equal to 0.3, the time window equal to 30 and rectified RUL set to 130. The results in Table 6 and in Figure 9 show that when no dimension reduction was performed, two layers are required to achieve better performance with the LSTM network. Furthermore, the experimental results in Figure 11, show also that increasing the complexity of the LSTM network does not lead to the prediction accuracy found with feature clustering.

According to Figure 10, the best results using the PCA method were obtained with 4 principal components catching all the variance in the original data. This result indicates

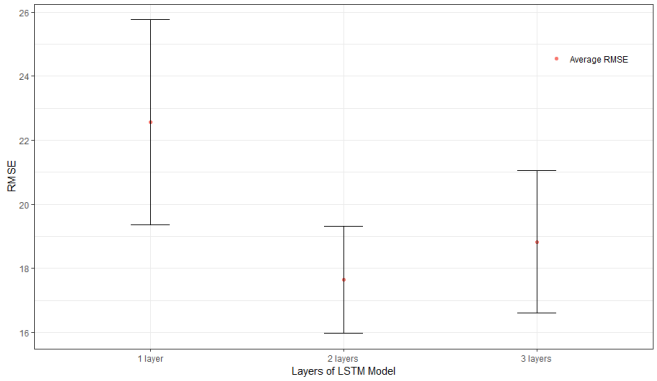

Figure 9. Impact of LSTM layers using all sensors in terms of RMSE.

that reducing the number of components reduces the complexity of the model which improves the prediction accuracy. However, with excessive dimension reduction, the features required for the prediction cannot be learned. In addition, our model performed better than the model using PCA dimension reduction, Figure 11.

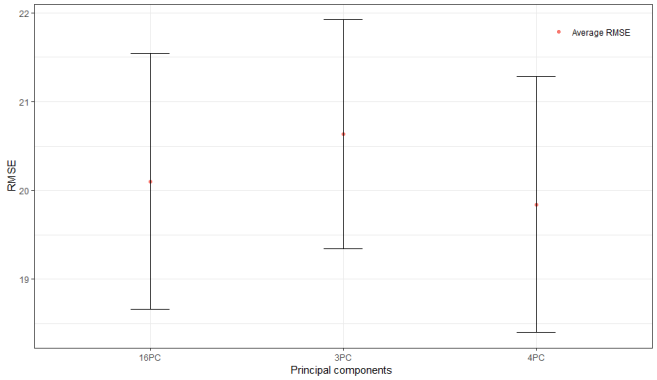

Figure 10. Impact of PCA dimension reduction with LSTM(1-layer) in terms of RMSE.

Compared to the other models, in Table Table 6, the proposed model performs better, with the lowest average RMSE value (16.14) and the lowest average S-score (299.19). It also provides the smallest standard deviation (values in brackets), showing that the performance of our model is more accurate. These results indicate that, with feature clustering as a part of the pre-processing, reducing the complexity of the model does not decrease the information needed for learning.

6.3.3. Case study

The estimated RUL values at each run are compared to the true RUL values for 4 different engines randomly selected when using RUL. The aqua-clear blue, orange and red lines correspond respectively to the estimated RUL values found by 2-layer all-sensor LSTM, 4PCA+LSTM(1-layer) and our

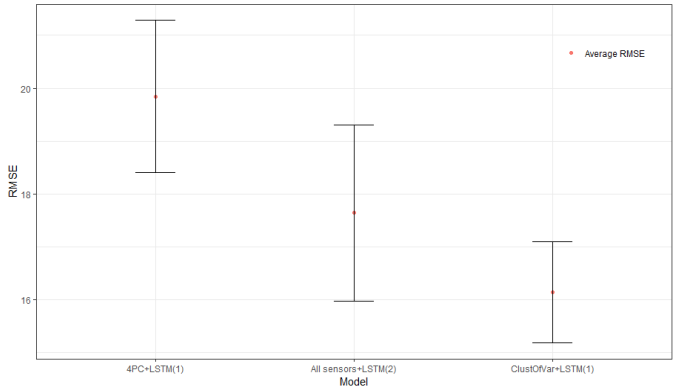

Figure 11. Comparative study.

Table 6. Results using different inputs: All sensors, feature clustering and PCA components.

	FD004 Test set		
Method	**Input features**	**RMSE (std)**	**S-score (std)**
All sensors +LSTM	1 layer	22.56 (3.20)	5127.79 (1679.31)
	2 layers	17.64 (1.66)	1823.05 (649.20)
	3 layers	18.83 (2.23)	3063.82 (833.38)
PCA + LSTM (1-layer)	16 PC	20.10 (1.44)	2949.82 (394.42)
	4 PC	19.84 (1.44)	2524.4 (1165.41)
	3 PC	20.63 (1.29)	2924.69 (1166.29)
Proposed method	**7 clusters**	**16.14 (0.96)**	**1299.19 (255.25)**

proposed model with 7 CustOfVar+ LSTM(1-layer). The engines are sorted by ascending order of real RUL represented in dark blue, to observe the performance of our model, indicating the error between the true and the predicted RUL. As shown in Figure 12, the RUL values estimated with the proposed method are closer to the true RUL values than those estimated using the 4 principal components as input or the one estimated with LSTM (2 hidden layers) without performing any type of dimension reduction.

Figure 13 shows the prediction results of randomly selected engines on FD004 test sets. The majority of the errors based on [Fu et al., 2021] fall within the confidence interval, defined by the lower bound d_i = 10 and the upper bound d_i = -13, proving that the model's performance was well trained.

In order to evaluate the model's performance, graphically, the predicted error d_i of the test set is represented. The green and purple lines correspond respectively to the lower and upper limit of d_i. As illustrated in Figure 14, most of the values fall within this confidence interval, showing the efficiency of the model estimation.

A comparison between the predicted RUL and the rectified true RUL can be seen in Figure 15. The red coloured line in both figures represents the predicted RUL values of the 248

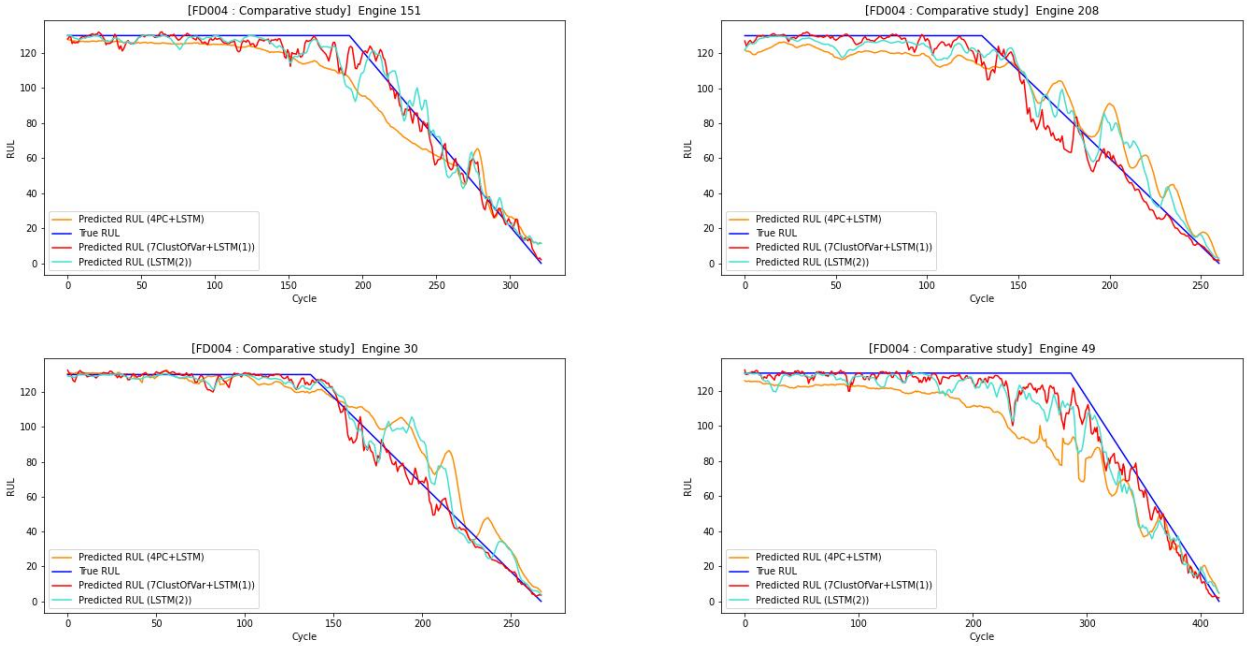

Figure 12. Comparing the estimated RUL using different methods 7ClustOfVar+LSTM(1), 4PC+LSTM(1) and LSTM(2), for randomly selected engines.

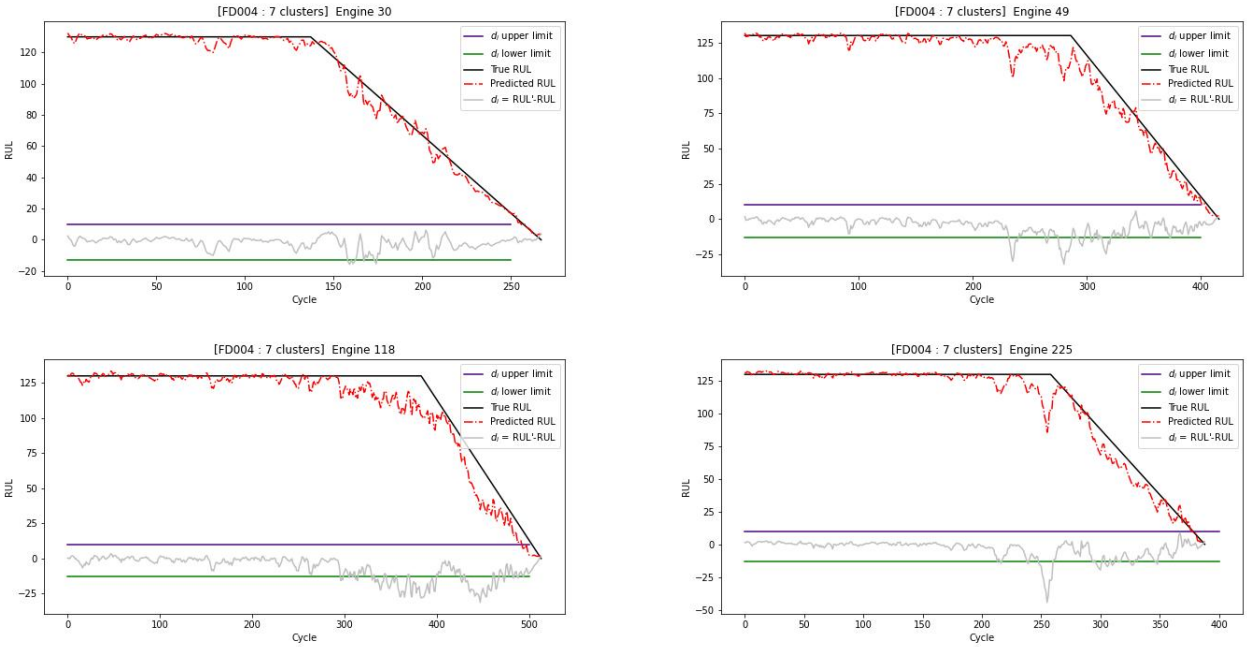

Figure 13. Estimated RUL using our proposed model for randomly selected engines.

engines of the test subset, and the dashed grey coloured line represents the given ground truth RUL of the test subset. Generally, the predicted values for each engine have a small difference from the true RUL. Overall, our model performs well in estimating the remaining useful life of the engines, despite some engines being too noisy and affecting the model's training and performance.

6.4. Comparison with other work

A comparison of the proposed method with state-of-art studies is presented in Table 7. It illustrates a comparison of works using the FD004 C-MAPSS dataset and summarizes the researcher with the publication date, network algorithm, predicted RMSE, and S-score results from previous studies. The value in brackets denotes the standard deviation of the metrics for the experiments in the study.

Table 7. Results comparison of different approaches using LSTM model.

Authors	Approach	RMSE	S-score
[Sateesh Babu et al., 2016]	semi-supervised setup	22.66	2840
[Zheng et al., 2017]	Deep LSTM	28.17	5550
[Listou Ellefsen et al., 2019]	GA+LSTM	22.66	2840
[Zhang et al., 2020]	LSTM-fusion	21.97	7726.9
[Zhao et al., 2020]	BiLSTM	24.86	5430
[Jiang et al., 2020]	BiLSTM-fusion	29.16	7886
[de Oliveira da Costa et al., 2020]	DANN-LSTM	21.30	2904
[Palazuelos et al., 2020]	CapsNet	18.96 (0.27)	2625.64 (266.83)
[Chen et al., 2020]	Hybrid LSTM with attention	27.08	5649.14
[Wang et al., 2022]	B-LSTM	16.24	5220
[Qin et al., 2022]	SD-TemCapsNet	16.49	804.05
[Ren et al., 2022]	Lightweight and Adaptive KD	15.10	1508.84
[Li et al., 2023]	Advertisal neurons network	26.64 (0.29)	10343 (783)
Proposed method	feature clustering + LSTM	16.14 (0.96)	1299.19 (255.25)

Accuracy has been improved by using advanced deep learning algorithms, with a deeper network layer or by using fusion algorithms. Compared with work done in recent years, the RUL predictions of turbofan engines show the effectiveness of our approach. As some methods tend to have a lower S-score or RMSE, the proposed method achieves better results in terms of the trade-off between performance and explainability, while using a simple architecture with only a single layer LSTM. For a PHM solution, the more insightful the model, the greater its reliability.

6.5. Explained proposed model

Shapely additive explanation (SHAP) is a great tool to understand complex tree-based and deep network model outputs. The SHAP method can link local and global interpretations. Deep neural networks, known as black-box are difficult to understand. However, it can be explained by the SHAP method which determines the factors responsible for the outcome and decision. The SHAP method is used to understand the RUL for a given engine of an aircraft.

The results of the average SHAP values of the predictive model are shown in Figure 16. It shows the clusters with the greatest influence on the RUL prediction, placing the largest cluster on top. The x-axis represents the average absolute SHAP value of each cluster. Clusters with larger absolute SHAP values correspond to the most important clusters. Cluster 1 represents the features that contribute the most to the prediction, followed by cluster 5, cluster 3, cluster 2 and cluster 7.

Based on Figure 17, the x-axis represents the SHAP value, and the y-axis represents all clusters. Red colour means a high value of a cluster; blue means a lower value of a cluster. Clusters are presented in order of importance of all clusters, with the first cluster being the most important and the last being the least important. This distribution shows the overall impact of the cluster directions. Clusters 1 and 7, with high values, contributed positively to the prediction, while with low values, they contribute negatively. On the other hand, clusters 5 and 3 have a negative influence on the prediction when they have high values, and a positive influence when they have low values.

Cluster 2 has a low contribution to the prediction. Clusters 6 and 4 have almost no contribution to the prediction, whether their values are high or low. To relate these results to the initial features, we use the equations found by ClutOfVar in section 6.1.3. Cluster 1, represented in Eq.(10) by the combination of features T24, T30, T50, PS30, and htBleed, has a positive contribution when its values are high, and a strong negative contribution when its values are low. A high value of cluster 1 (For each additional value of these features, cluster 1 decreases by 9.6) corresponds to a low value of total LPC outlet temperature T24, total HPC outlet temperature T30, total LPT outlet temperature T50, static pressure at HPC outlet PS30, and purge enthalpy htBleed. They have the greatest power to predict failures; their higher values increase the remaining useful life and subsequently decrease failure degradation. Cluster 5 has a strong negative contribution when its values are high, and a positive contribution when its values are low. A high value of cluster 5 corresponds to a high value of the physical core speed Nc, and a high value of the cor-

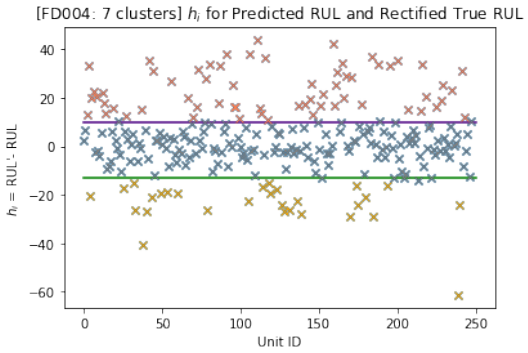

Figure 14. Predicted error values using the proposed model.

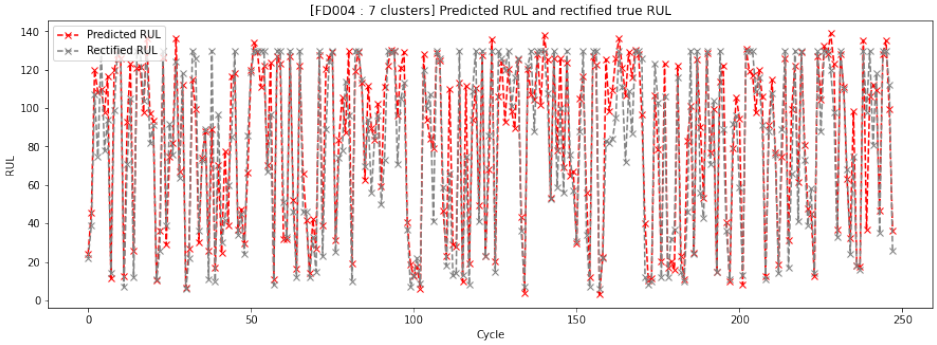

Figure 15. Comparison of the rectified true values and predicted values of RUL, of all engines in the test set using the proposed model.

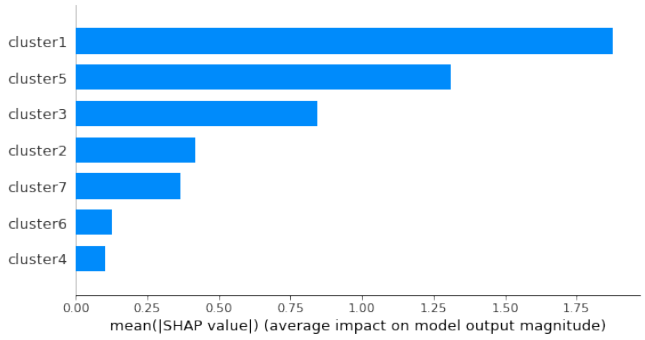

Figure 16. SHAP importance ranking of clusters.

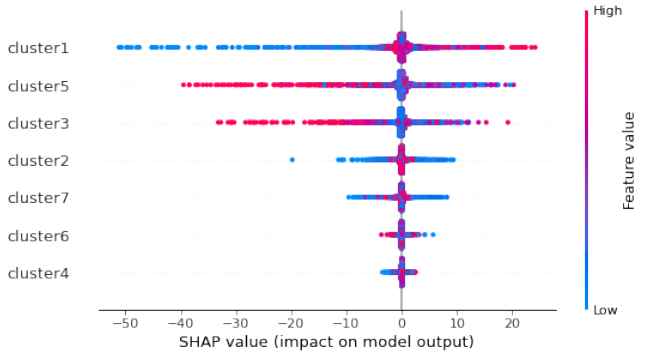

Figure 17. SHAP value plots for the proposed model.

rected core speed NRc. That means, a low core speed increases the remaining useful life and so the failure degradation decrease. Cluster 3 with low values does not contribute to prediction but high values cannot specify contribution to the prediction. This cluster is represented in Eq.(12) by the Ratio of fuel flow to Ps30 and the total pressure at the HPC outlet. Cluster 2 is represented in Eq.(11) by Total Bypass-duct. It has a low contribution to the prediction. A high value of Total Bypass-duct does not contribute to RUL prediction. On the other hand, a high value cannot confirm the way of contribution to the prediction. Cluster 7 is represented in Eq.(16) by Bypass Ratio, BPR negatively, and strongly positively by HPT coolant bleed, W31, and LPT coolant bleed, W32. Low values of HPT coolant bleed and LPT coolant bleed correspond to decrease RUL so increased failure degradation. We can conclude that:

1. The higher the values of total LPC outlet temperature T24, total HPC outlet temperature T30, total LPT outlet temperature T50, static pressure at HPC outlet PS30, and purge enthalpy htBleed, the higher chance of failure due to degradation.

2. The higher the values of physical core speed Nc, and a high value of the corrected core speed NRc, may result in lower RUL values, therefore a higher chance of failure due to degradation.

3. The higher the values of ϕ, the Ratio of fuel flow to static pressure at the HPC outlet, and P30, the total pressure at the HPC outlet, may result in a lower RUL values, as a consequence a higher chance of failure due to degradation.

4. Low values of HPT coolant bleed, W31, and LPT coolant bleed, W32, correspond to decrease RUL so increased failure degradation.

7. CONCLUSION

In this paper, we proposed a data-driven approach to predictive maintenance. The proposed model applies normalization based on operating conditions, and clustering of sensor features during the pre-processing phase and is built with the LSTM algorithm. Our model has been implemented and evaluated to predict the RUL of engines on the NASA turbofan engine dataset. An ablation study was performed compared to those obtained without dimension reduction using LSTM with different hidden layers and those with a PCA dimension reduction algorithm. The experimental results indicate the effectiveness and suitability of the proposed method. The XAI method was used to explain decisions based on the predictive model outputs, help understand the operation of the LSTM model, and describe the most important features contributing to the predicted degradation of turbofan engines.

As for future work, a promising direction could be followed by applying the technique to other data sets to validate the proposed approach. In addition, another exciting topic is driving causality in post-hoc XAI tools to provide more reliable and interpretable explanations and a better understanding of which part of the machine needs maintenance. It provides valuable information to perform proactive maintenance before the aircraft engine fails.

REFERENCES

Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Chavent, M., Kuentz-Simonet, V., Liquet, B., and Saracco, J. (2012). Clustofvar: An r package for the clustering of variables. *Journal of Statistical Software*, 50(13):1–16.

Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., and Li, X. (2020). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics*, 68(3):2521–2531.

de Oliveira da Costa, P. R., Akçay, A., Zhang, Y., and Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering System Safety*, 195:106682.

Frederick, D., DeCastro, J., and S., J. (2007). User-guide for the commercial modular aero-propulsion system simulation (C-MAPSS). Technical report, NASA.

Fu, X., Xia, X., Zhong, S., Lin, L., Fu, S., Liu, D., and Cui, Z. (2021). A novel label correction method for remaining useful life prediction of turbofan engines. In *2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, pages 74–80.

Gugulothu, N., Malhotra, P., Vig, L., and Shroff, G. (2018). Sparse neural networks for anomaly detection in high-dimensional time series.

Heimes, F. (2008). Recurrent neural networks for remaining useful life estimation. *2008 International Conference on Prognostics and Health Management*, pages 1–6.

Hochreiter and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(193-218):1–16.

Jiang, Y., Lyu, Y., Wang, Y., and Wan, P. (2020). Fusion network combined with bidirectional lstm network and multiscale cnn for useful life estimation lstm network and multiscale cnn for useful life estimation. pages 620–627.

Karasu, S. and Altan, A. (2021). Crude oil time series prediction model based on lstm network with chaotic henry gas solubility optimization. *Energy*, 242.

Lai, Z., Liu, M., Pan, Y., and Chen, D. (2022). Multi-dimensional self attention based approach for remaining useful life estimation. *arXiv preprint arXiv:2212.05772*.

Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1):314–334.

Li, X., Xu, Y., Li, N., Yang, B., and Lei, Y. (2023). Remaining useful life prediction with partial sensor malfunctions using deep adversarial networks. *IEEE/CAA Journal of Automatica Sinica*, 10(1):121–134.

Li, X., Zhang, W., Ma, H., Luo, Z., and Li, X. (2022). Degradation alignment in remaining useful life prediction using deep cycle-consistent learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5480–5491.

Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., and Zhang, H. (2019). Remaining useful life pre-

dictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering System Safety*, 183:240–251.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Palazuelos, A. R.-T., Droguett, E. L., and Pascual, R. (2020). A novel deep capsule neural network for remaining useful life estimation. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 234(1):151–167.

Qin, Y., Yuen, C., Shao, Y., Qin, B., and Li, X. (2022). Slow-varying dynamics-assisted temporal capsule network for machinery remaining useful life estimation. *IEEE Transactions on Cybernetics*.

Ren, L., Wang, T., Jia, Z., Li, F., and Han, H. (2022). A lightweight and adaptive knowledge distillation framework for remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, pages 1–11.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, 30.

Sateesh Babu, G., Zhao, P., and Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In Navathe, S. B., Wu, W., Shekhar, S., Du, X., Wang, X. S., and Xiong, H., editors, *Database Systems for Advanced Applications*, pages 214–228, Cham. Springer International Publishing.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management (PHM)*, pages 1–9, Denver, CO.

Saxena, A., Roychoudhury, I., Celaya, J., Saha, S., Saha, B., and Goebel, K. (2010). Requirements specifications for prognostics: An overview. *AIAA infotech*.

Shapley, L. S. (1953). *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton.

Tjoa, E. and Guan, C. (2021). A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813.

Wang, X., Huang, T., Zhu, K., and Zhao, X. (2022). Lstm-based broad learning system for remaining useful life prediction. *Mathematics*, 10(12).

Zhang, Y., Hutchinson, P., Lieven, N., and Nunez-Yanez, J. (2020). Remaining useful life estimation using long short-term memory neural networks and deep fusion. *IEEE Access*, PP:1–1.

Zhao, C., Huang, X., Li, Y., and Yousaf, M. (2020). A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction. *Sensors*, 20:7109.

Zheng, S., Ristovski, K., Farahat, A., and Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95.

Zio, E. (2012). Prognostics and Health Management of Industrial Equipment. In Kadry, S., editor, *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, pages 333–356. IGI Global.

BIOGRAPHIES

Genane YOUNESS Ph.D., is currently an associate professor and AI researcher at LINEACT Laboratory (Laboratoire d'Innovation Numérique pour les entreprises et l'Apprentissage pour la Compétitivité des Territoires)). Dr. Youness is an associate researcher at the MSDMA-CEDRIC Laboratory - CNAM (Conservatoire National des Arts et Métiers) in Paris, since 2021. In 2004, she was a lecturer and head of the statistics department for 16 years at CNAM in Lebanon. Dr. Youness holds a Bachelor's degree in Applied Mathematics from the Faculty of Science of the Lebanese University in 1998, a Master's degree and a PhD in Statistics from the University Pierre-et-Marie-Curie (Paris VI) in 2000 and 2004 respectively. Her research activities concerned the comparison of close partitions, data mining and machine learning theory and its applications. Her current research efforts are focused on data-driven predictive maintenance using artificial intelligence, explainable artificial intelligence, and AI model's confidence.

Adam AALAH received his Engineering degree in Business Intelligence from the Engineering school of Applied Sciences in Morocco, in 2019. He completed his M.S. in Data Science at the Polytechnic Institute of Paris, in September 2022. His research interests include data-driven predictive maintenance using artificial intelligence and explainable AI as well as medical image processing, pattern recognition and classification using machine learning and deep learning.