Technium Vol. 4, Issue 6 pp.58-83 (2022)
ISSN: 2668-778X
www.techniumscience.com

Modeling and fault detection of a turbofan engine by deep-learning approach

Offole Florence[1], Essola Dieudonné[1], Fohoue Kennedy[1], Kamgua Piam[1], Mela Rodrigue[1], Issonj Nelson

[1] Laboratory of Energy, Materials, Modeling and Methods (E3M), National Higher Polytechnic School of Douala, University of Douala, Cameroon
Corresponding author: Offole Florence, e-mail: florenceoffole@yahoo.fr

Abstract. Throughout the world, thousands of passengers travel by air, their quality depends on that of the equipment used. Predictive maintenance is increasingly used to estimate. The remaining useful life of system components and in particular turbofan engines as an essential component. It is used to predict failure before it occurs, optimize component design, extend equipment life, and reduce maintenance costs. However, the algorithms proposed in the literature to date to determine the remaining useful life lack precision with a quadratic error around 20 while the physical models have errors of the order of 0.02. The problem here is how to increase the accuracy of predicting the remaining useful life of a turbofan engine. The objective of this study is to develop a more realistic and accurate algorithm for calculating the remaining useful life of a turbofan engine. To do this, we considered the degradation of the high pressure compressor and the fan as essential organs of the turbojet engine and we used deep learning, known for its high precision linked to a great capacity for extracting information. More specifically, it involved acquiring data on a turbojet engine in operation, pre-processing this data, developing the prediction model, training the model and finally validating the approach in comparison with other diagnostic methods. and to model these defects. We compared two deep learning architectures per application against the CMAPSS dataset to assess their performance. The LSTM architecture we developed prevailed with an RMSE of 13.76, well positioned compared to the literature architecture.

Keywords. Time series prediction, remaining lifetime, turbofan engine, deep learning.

1. Introduction

In recent years, the development of modern aircraft technology has led to a complex aircraft system, where high reliability, quality and safety are required in a competitive environment [1] . Indeed, according to the Aviation Safety Network [2] , millions of passengers fly around the world every day and their safety is a critical aspect. In the past, engineers were content with corrective maintenance [3], today the abundance of operating data has opened up the prospect of fault prognosis based on operating conditions. Therefore, the time has come for predictive maintenance, which makes it possible to predict failure before it occurs, but also to optimize the design of components, extend the life of equipment and reduce maintenance costs [4] .

Thus, three main methods of calculating remaining useful life can be used: an approach based on a physical/mathematical model of degradation, an approach based on operating data from the healthy state to the fault state and a hybrid approach [4] . The approach by a a physical model has a high accuracy

with an error around 0.02 [5] but considerable knowledge of the studied system is necessary to create a model. On the other hand, the method based on operating data does not require expert knowledge of the system but has a low precision of the order of 20 in squared error due to the absence of a precise physical model.

The predicting of the remaining useful life time therefore becomes approximate and can have serious consequences. A late prediction implies that the failure has already occurred, which can affect the safety of passengers and crew in flight [6] and an early prediction leads to additional maintenance costs at non-optimized periods [7].

1.1. The turbojet

The turbojet is a propulsion system which transforms the chemical energy potential contained in a fuel, associated with the oxidizer that is the ambient air, into kinetic energy making it possible to generate a reaction force in a compressible medium in the opposite direction to the ejection [8]. Turbofans are jet engines derived from turbojet engines. They are distinguished from them essentially by the fact that the thrust is not obtained only by the ejection of hot gases, but also by a flow of cold air.

1.2. The main indicators of deterioration of a turbojet engine

The aircraft's dashboard has a set of instruments for controlling engine power and supervising its operating parameters. It is important to understand that none of the isolated instruments can assess the health of the engine, they must be considered together. The essential control parameters on a turbojet engine are:

- Engine compression ratio (EPR around 25): this parameter measures the thrust generated by the engine, it is the ratio of the air pressure at the turbine outlet to the air pressure at the compressor inlet . It is a recognized and certified engine performance indicator [9].
- Rotor rotation speed, it represents the number of revolutions per minute (RPM). It is expressed as a non-dimensional ratio that compares rotor RPM to 100% rated speed representing a high power situation. The high pressure compressor RPM (N2) is used to supervise the operating limit conditions [9].
- Exhaust Gas Temperature (EGT). It is used to monitor the health and mechanical integrity of the engine. It is high for values >2500.
- Fuel flow indicator: this parameter indicates the quantity of fuel supplied to the injectors in kilograms per hour. It is used to monitor fuel consumption, engine performance and speed regulation. Abnormally high fuel flow may indicate a leak between the fuel control system and the injection nozzles. Particularly when EPR and RPM are normal or low [9].
- Air temperature indicator: outside the aircraft. This temperature can be recorded from specific locations and the actual value can mean different things depending on the particular aircraft. This temperature is generally used to help select the EPR in those engines where the thrust is defined by the EPR [9].

Of all the indicators cited above, [10] has established that the static pressure at the outlet of the high pressure compressor (Ps30) and the ratio of fuel flow to Ps30 (Phi) have the greatest impact on engine degradation, this which suggests that the most critical component is the high pressure compressor. Thus, major recent works use datasets that only model high-pressure compressor degradation as the cause of failure. However, we question this assumption, indeed although the high pressure turbine is the central element of propulsion for the turbojets for military use, in the turbojets for civil use, it is the fan which contributes for more than 80% thrust [11]. In the rest of the work we will be particularly interested in the high pressure compressor and the fan, this can be justified by the recurrence and the gravity of the phenomena which directly impact these two organs during the flight: overvoltage, stall, ingestion of debris, hot start, seizing have direct effects on the high pressure compressor [6] . But also, they are the organs at the origin of the propulsion and without which, there is no flight. Finally, the part of the engine cycle most susceptible to instability is the compression phase [9].

1.3. Fault diagnosis techniques

1.3.1. Remaining useful life (RUL) prediction

Engine Optimization will allow us to model its functional or dysfunctional behavior: it is a question of identifying which are the parameters and entries which will make it possible to have an optimization of the system to improve its performances. This involves simulating the mathematical system under different conditions and thus determining which parameter will produce the best result [12-14].

To model a defect, a model of the system is required. Generally, a model is an official formulation of all the knowledge about the functioning of the system and the related faults. The remaining useful life (RUL) of a machine is the expected life or time of use remaining before the machine will require repair or replacement. Units of life can be quantities such as distance traveled (miles), fuel consumed (gallons), repeat cycles performed, or time elapsed since start of operation (days). Similarly, the evolution of time can mean the evolution of a value with such an amount [15].

Generally, estimating the RUL of a system comes down to developing a model that can perform the estimation based on the temporal evolution of the condition indicator values, such as the model that compares the evolution from a condition indicator to time series of systems that worked until failure. Such a model can calculate the most likely failure time of the current system. This model corresponds better to the CMAPSS data set which contains the operating-to-failure data. Figure 1 presents steps to built a RUL prediction algorithm [16].

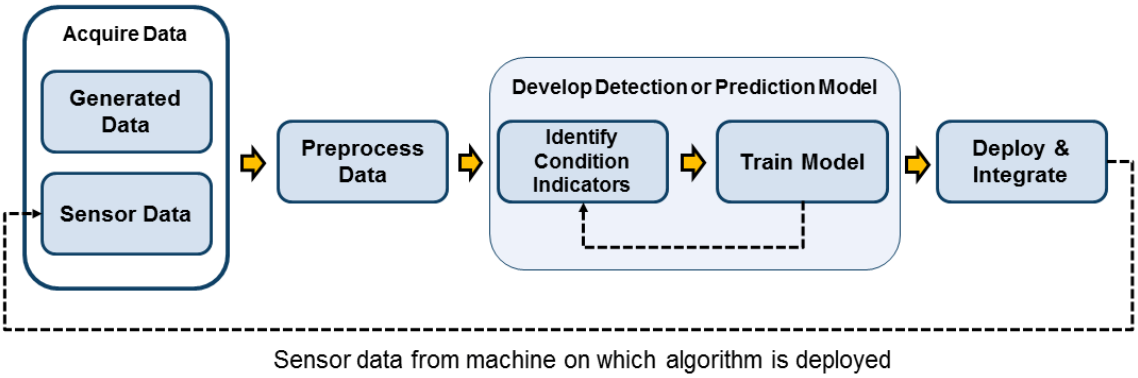

Figure 1: Steps to build a prediction algorithm

In recent years, deep learning techniques have outperformed traditional machine learning models. Deep learning has proven to be an effective solution in view of its ability to automatically learn the temporal dependencies present in time series [17].

1.3.2. Short-long term memory network (LSTM)

LSTM network are an improvement of recurrent neural networks (RNN). RNNs were introduced as a variant of ANN (Artificial Neural Network) for time-dependent data. While MLPs ignore temporal relationships within the input data, RNNs connect each time step with previous ones to model the time dependence of the data, providing native RNN support for the given sequences [18].

In 1997, LSTM networks were introduced as a solution to the problems of ERNN (Elman Recurrent Neural Networks). LSTMs are able to model temporal dependencies in broader horizons without forgetting short-term patterns. LSTM networks are differentiated from ERNNs in the hidden layer, also known as the LSTM memory cell. LSTM cells use a multiplicative input gate to control the memory units, preventing them from being modified by irrelevant disturbances [16].

1.3.3. Short-long term memory network (LSTM)

CNNs are a family of deep architectures originally designed for computer vision tasks. They are considered state-of-the-art for many classification tasks such as object recognition, speech recognition and natural language processing. CNNs can automatically extract features from high-dimensional raw data with a grid topology such as pixels in an image, without the need for any feature engineering. The model learns to extract meaningful features from the raw data using the convolutional operation, which is a sliding filter that creates feature maps and aims to capture repeated patterns in different regions of the data. This feature extraction process provides CNNs with an important characteristic called distortion invariance, which means that features are extracted regardless of their location in the data [16].

The work of [16] demonstrated that among all recent deep learning models, long short-term memory networks (LTSM) and convolutional networks (CNN) are the best alternatives. While LSTM gets the most accurate prediction, CNN makes the most repeatable prediction. In the rest of our study, we will present its two approaches. Our objective being to build an algorithm of great precision, we have therefore chosen to use the deep learning approach because of its adaptability and the reliability it offers to model the faults of a turbojet engine, therefore the purpose of optimize it better.

2. Materials and methods

To determine the remaining useful life, we will develop two deep learning algorithms, one based on convolutional neural networks and the other based on short and long memory networks, then we will apply the prediction algorithms useful remaining lifetime on the CMAPSS dataset.

2.1. Materials

It is a question here of presenting not only the experimental device used for data collection, but also the different software used for numerical simulations.

2.1.1. Digital simulation device

In this work, we used the digital simulation data C-MAPSS (Commercial Modular Aero Propulsion System Simulation) Commercial Modular AeroPropulsion System Simulation, is a state-of-the-art turbofan engine simulator developed by NASA. It is a 90,000 thrust class engine model and the package includes an atmospheric model capable of simulating operations at altitudes ranging from sea level to 40,000 ft, Mach numbers from 0 to 0.90 and sea level temperatures from -60 to 103°F. In addition, all gains for the fan speed controller and the four limit regulators are programmed so that the controller and regulators operate as expected. Over the full range of flight conditions and power levels.

CMAPSS provides 14 inputs for simulating the effect of faults and deterioration on any of the 5 rotating engine elements and 58 outputs which include the response surfaces of various sensors and operating margins. Inputs are: Fuel Flow, Blower Efficiency, Blower Flow, Blower Compression Ratio, LPC Efficiency, LPC Flow, LPC Compression Ratio, HPC Efficiency, HPC Flow, HPC compression, HPT efficiency, HPT throughput, LPT efficiency, HPT throughput. The table presents some information about the state of degradation and deterioration of the engine.

Technium Vol. 4, Issue 6 pp.58-83 (2022)
ISSN: 2668-778X
www.techniumscience.com

Table 1. Information on the data set generated by the sensors.

name	FD003
Fault modes	HPC and blower degradation
Number of motor trajectories for training	100
Number of motor trajectories for the test	**100**
Default training sample	24720
flight condition	Stationary at sea level

2.1.2. Software used

MATLAB was at the heart of this work, allowing us to carry out all the stages of construction of our algorithm, in particular, the preprocessing of the data, the construction of the architecture of the neural network, the training of the neural network and the neural network performance evaluation. To do this, we used Matlab's Deep Learning toolbox, which offers an environment for designing and implementing deep neural networks.

2.2. Methods

The flowchart of figure 2 presents the stages of construction of the algorithm 1. The stages presented in the flowchart will be examined in more detail in the following.

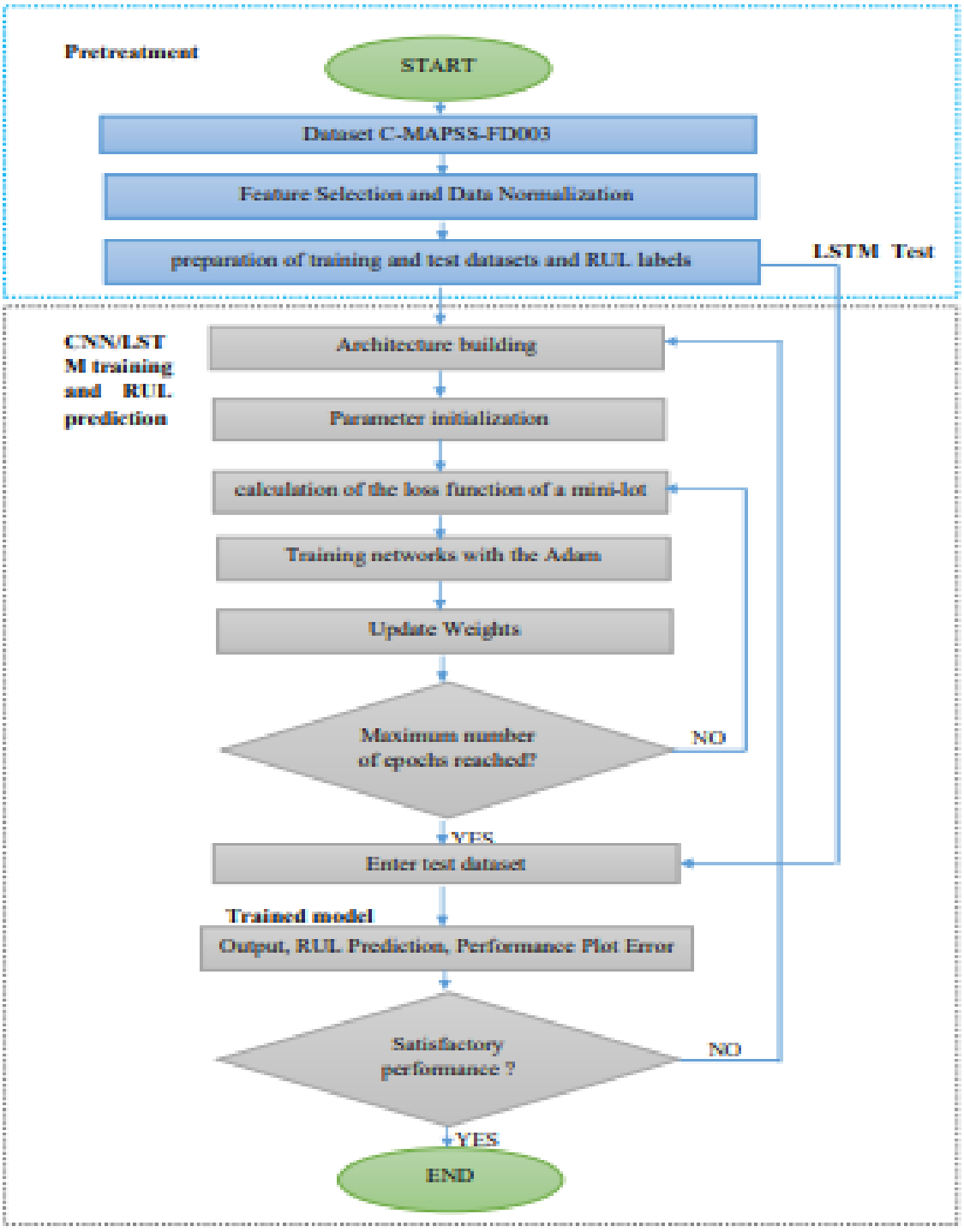

Figure 2: Prediction Algorithm Construction Flowchart

2.2.1. Acquiring c-mapss data.

In the CMAPSS, the sensor data generated by this simulation based on the input parameters are summarized in **Erreur ! Source du renvoi introuvable.**1, which are processed and produce the output data, the main ones of which are in Table 2.

Table 2. CMAPSS output data to measure system response.

Sensor No.	Symbol	Description and unit
	Output parameters available in data	
1	T2	Total temperature at blower inlet °R
2	T24	Total outlet temperature LPC °R
3	T30	Total temperature at the HPC outlet °R
4	T50	Total outlet temperature LPT °R
5	P2	Psia blower inlet pressure
6	P15	Total pressure in the auxiliary air circuit psia
7	P30	Total pressure at HPC outlet psia
8	nf	Blower speed rpm
9	N/a	Engine core speed rpm
10	Epr	Engine compression ratio (P50/P2) --
11	PS30	HPC outlet static pressure psia
12	Phi	Fuel Flow Ratio at Ps30 pps/psi
13	NRf	Corrected blower speed rpm
14	NRc	Corrected heart rate rpm
15	OPI	Dilution ratio --
16	farB	Combustion fuel-air ratio --
17	htBleed	Purge enthalpy --
18	Nf_dmd	Requested blower speed rpm
19	PCNfR_dmd	Corrected blower speed requested rpm
20	W31	HPT Cooling Purge lbm/s
21	W32	Cooling Purge LPT lbm/s

To ensure that CMAPSS generates realistic data, the data was compared to those published by Goebel [18]. It turned out that the qualitative responses were similar. Indeed, by increasing the flow rate and the efficiency, the response surfaces were similar to those of [18-19]. To obtain the output data, the performance and throughput were incremented for each module (HPC, HPT and LPT) and CMAPSS was compiled under different random conditions. Functional parameters were varied exponentially as ordered in [18]. Maintenance times between missions have been modeled such as noise, which influences performance parameters, namely throughput and efficiency; which improves by remaining within certain limits, therefore the performance losses are not monotonous.

The mechanical failure model, including those of Arrhenius, Coffin Manson and Eyring, makes it possible to establish an engine health equation [19]:

$$h(t) = 1 - exp\{at^b\} \quad (1)$$

Or

$e^a = \frac{A}{th_w}$, $t^b\ B(t)$, th_w *est upper wear thresold*, A *est une constante* , B *fonction du temps*

Generally, the system will be observed from an initial non-zero degradation denoted d, which allows data generation to start at an arbitrary point in the evolution of the crack. This initial degradation is modeled as an additive term to the engine health equation from an initial state of degradation [19] we then obtain:

$$h(t) = 1 - d - exp\{at^b\} \ (2)$$

The health index can be used to model various phenomena across the subsystem. Especially for aircraft engine modules like compressor section and turbine section, health index is described as both efficiency (e) and throughput (f). The trajectories for throughput and efficiency vary for different fault modes and are modeled as two separate health indices as shown below :

$$e(t) = 1 - d_e - exp\{a_e(t).t^{b_e(t)}\} (3)$$

Equation of health according to flow [20]:

$$f(t) = 1 - d_f - exp\{a_f(t).t^{b_f(t)}\} \ (4)$$

The terms e(t) and f(t) are then combined to form the general health index H(t), the response of the simulated engine to the input data.

General engine health index equation [19]:

$$H(t) = g(e(t), f(t)) \quad (5)$$

Where g is the minimum of all operating margins considered for the fan, HPC, HPT, EGT…

Expression of g

$$g(e(t), f(t)) = min(m_{Soufflante}, m_{HPC}, m_{HPT}, m_{EGT}) (6)$$

Where the margins m are functions of yield e(t) and throughput f(t).

The margins are calculated by estimating the distance between the states of the engine at a time t with respect to the defined operational limits. Each of these margins is in the interval [0,1] where 1 means a perfectly healthy system and 0 a system where the margin is below the acceptable limit. In the

simulation, the limit was set at 15% for the HPC, LPC and fan stall margin. The margin is 2% for the margin of EGT. The procedure for generating data with CMAPSS is as follows:

- Choose an initial state of deterioration (f_0,e_0) domain of definition of e and f

$e_0 \in [0.99,1], f_0 \in [0.99,1]$ (7),

- Impose an exponential variation on the loss of throughput and efficiency as explained in [14] . This leads to the general health index H(t) (Equation 5) which varies with time. The direction and evolution of faults is defined by:
  $f_i, e_i \leq 1\%$ (Equation 8: constraint on e and f for a direction and a random evolution of the deteriorations),
  $a_k \in [0.001,0.003]$
  $b_k \in [1.4,1.6], k = 1,2$
- Stop when H health = 0,
- Impose noise on sensor measurements.

As output, we obtain time series in the form of vectors of the output parameters. They are produced by modifying the HPC throughput and efficiency input parameters from the initial state to the failure threshold.

The training data have trajectories that top at the failure threshold while the test and validation data represent trajectories that stop before the failure threshold.

2.2.2. Pretreatment

To do this you need:

- Download and unzip the Turbofan Engine Degradation Simulation dataset.
- Perform basic preprocessing: MATLAB® includes many useful functions for basic preprocessing of data in arrays. Data is loaded using the localLoadData [20] function. The function extracts data from filenamePredictors and returns a table that contains the training predictors and the corresponding response sequences (i.e. RUL). Each row represents a different engine.
- Remove features with less variability: Features that remain constant or do not provide relevant information for the RL calculation can have a negative impact on training. We use the prognostic function to measure the variability of characteristics in the event of a failure.
- Normalize training predictors: The measurement data collected from each sensor is normalized to fall within the range of [-1, 1] using the min-max normalization method,

Data Normalization Formula Equation :

$$x_{norm}^{i,j} = \frac{2(x^{i,j} - x_{min}^{j})}{x_{max}^{j} - x_{min}^{j}} - 1, \forall i,j \ (8)$$

Where $x^{i,j}$ denotes the original i-th data point of the j-th sensor, and $x_{norm}^{i,j}$ is the normalized value of $x^{i,j}$. x_{max}^{j} and x_{min}^{j} designate respectively the maximum and minimum values of the original measurement data of the j-th sensor.

- Operate time windows: More information can usually be obtained from time sequence data compared to the multivariate data point sampled at a single time step. Let Ntw be the size of the time window. At each time step, all sensor data passed in the time window is collected to form a high-dimensional feature vector and used as inputs for the network.
  It should be noted that in the test set, the data cycles recorded for the test engine units have a different length, and the shortest is only 31 cycles specifically. We have a normalized data

sample of the 14 selected sensors in a time window of size 30 relative to a single motor unit in the training dataset.

Pay attention to RUL less than RULlim: in general , the engine block operates normally from an early age and degrades linearly thereafter, so that the network concentrates on the part of the data where the engines are more likely to break down (end of engine life), the responses are cut at the RULlim threshold of 125 [21]. This forces the network to process instances with lower or equal RUL values [22]. In this work, we will use for the detection the generalized model of structure based on artificial intelligence, we will present the bases of neural network.

2.3. Deep convolutional networks with attention

We use two types of architectures:

2.3.1. CNN network architecture

We constructed a CNN consisting of four consecutive sets of CNN layers, with filterSize and numFilters as two input arguments to convolution2dLayer, followed by a fully connected layer of size numHiddenUnits and a dropout layer with a dropout probability of 0.5. A final convolution layer is used with 1 filter to combine the previous feature maps into one. Filter size is 3 × 1. num Responses is set to 1. All layers use ReLU (Rectified Linear Unit) as activation functions.

Figure 3 presents the graph of the CNN neural network layers to visualize the architecture of the underlying network and Table 3: the values of the parameters of the defects [10].

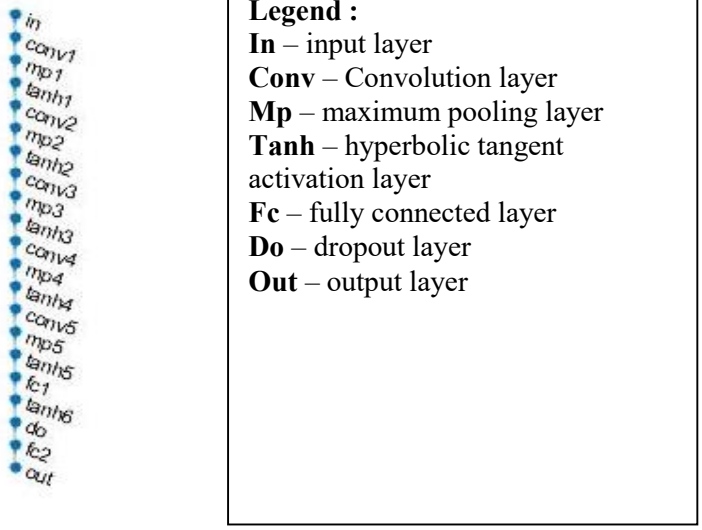

Figure 3: Visualization of the proposed deep CNN architecture

Table 3. Default parameters of the method used for the experiment

Setting	Value	Setting	Value
Outlet size (F_L)	10x1	Number of convolution layers	5
Convolution filter size (F_N)	10x1	Number of Neurons in a Fully Connected Layer	100
Number of features (F_{tr})	14	Dropout rate	0.5
Time window length (N_{tw})	30	Batch size	512
RULlim	125	Number of epochs	250
Initial learning rate	0.01	Learning rate after 200 epoch	0.0001

Where RULlim is the limit of RUL from which the degradation is taken into account.

2.3.2. LSTM architecture

We created an LSTM network consisting of an LSTM layer with 200 hidden units, followed by a fully connected layer of size 50 and a dropout layer with a dropout probability of 0.5. Figure 4 presents the layer graph of LSTM neural networks. To specify the training options, we set the maximum number of epochs to 300 with mini-lots of size 512 using the 'Adam' solver. We specified a variable learning rate of 0.01 up to the 200th epoch and 0.001 from the 201st. To keep sequences sorted by length, we set "Shuffle" to "never".

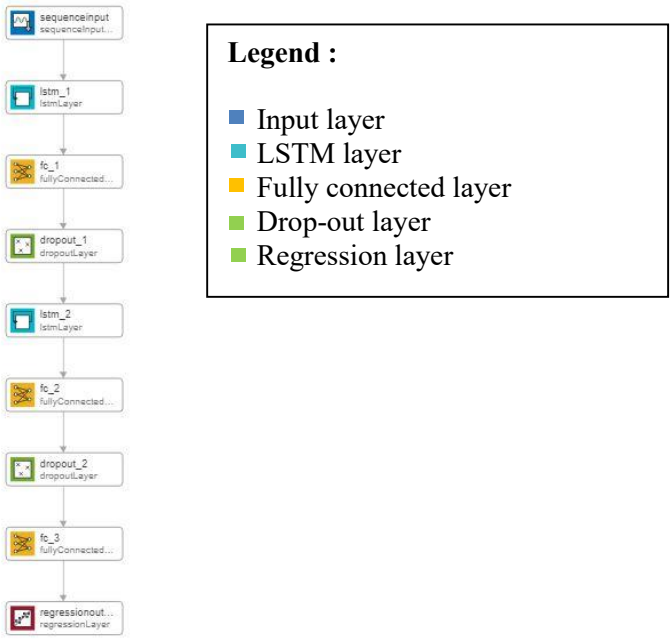

Figure 4: Proposed LSTM architecture

Table 4. Default parameters of the method used for the experiment

Setting	Value	Setting	Value
Outlet size (F_L)	1x1	Number of LSTM layers	1
Number of hidden layers	200	Batch size	512
Number of features (F_{tr})	14	Dropout rate	0.5
Number of Neurons in a Fully Connected Layer	50		
RULlim	125	Number of epochs	300
Initial learning rate	0.001	Learning rate after 200 epoch	0.0001

Where RULlim is the limit of RUL from which the degradation is taken into account.

2.3.3. Performance indicators

In [27] , 2 metrics were used to evaluate the performance of the proposed prognostic method: the scoring function and the mean square error. The notation function is defined below:

$$s = \sum_{i=1}^{N} s_i \quad (9)$$

With:

$$s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1 \ \ pour\ d_i < 0 \\ e^{\frac{d_i}{10}} - 1, \ \ pour\ d_i \geq 0 \end{cases} \quad (10)$$

Where the score and N is the total number of test data samples. di = RUL′ i − RUL i, i.e. the error between the estimated RUL value and the actual RUL value for the ith sample of test data. The scoring function penalizes late prediction more than early prediction because late prediction usually leads to more serious consequences in many fields such as aerospace industries.

Another method to evaluate the effectiveness of the proposed method is the root mean square error (RMSE) calculation over all time cycles of the test sequences to compare the performance of the network on the test data. The wording of the RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \quad (11)$$

3. Results and discussion

The performance in terms of prognosis of the proposed method for RUL estimation will be presented. The effects of different factors on the results will be studied, including the number of hidden layers and

the length of the time window. Comparisons with other popular neural network architectures will be made to show the efficiency of the proposed structure. The result of the proposed approach will be compared to the latest state-of-the-art prognostic results on the same C-MAPSS dataset including errors.

3.1. Data preprocessing

Before exploiting the training data, it is necessary to remove the "empty shells", this is data that does not provide relevant information. Figure 5 presents the variability of some indicators.

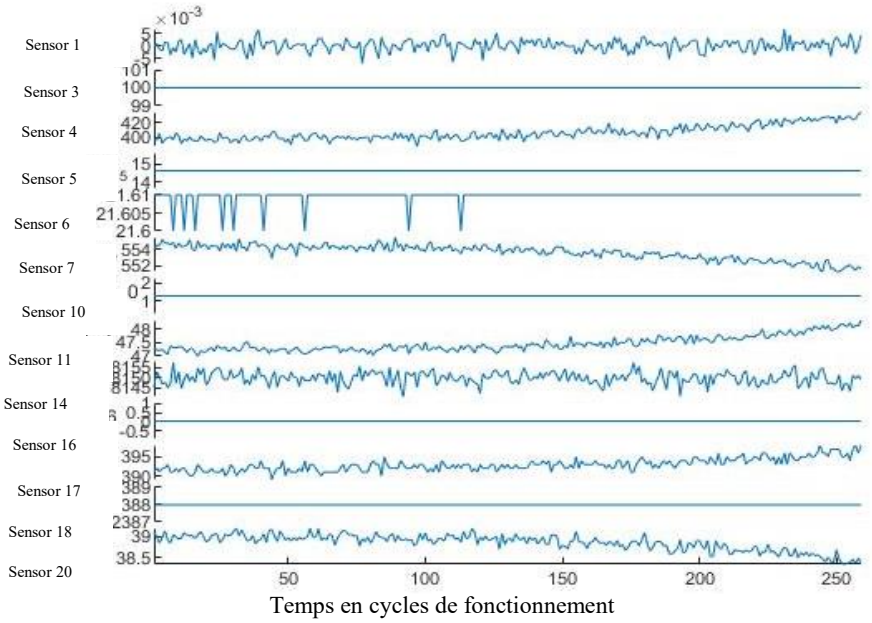

Figure 5: variation of certain predictors with cycles

We observe certain data which vary with the operating cycles and become monotonous when the end of the cycle approaches, in particular the sensors 4; 7; 18 and 20 which are not monotonous. Other characteristics, on the other hand, remain constant throughout the operating cycle orhave almost zero variability.

Data varies with duty cycles as life decreases. Figure 5 offers relevant information as to the prediction of the remaining useful life, especially since when they become monotonous and the remaining useful life tends towards the end. They are in fact linked to the degradation of the high pressure compressor, for example sensor 4 is the total temperature at the LPT outlet, sensor 20 the HPT cooling purge (lbm/s). Data with consistent variability and/or monotonicity is suppressed. This allowed us to keep 14 data whose indices are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21. Recent work by [23-24] related to prediction of remaining useful life using CMAPSS dataset FD001 also retained the same data for training.

3.2. Performance and prognosis of CNN-like deep neural networks

The RUL prediction results are shown in the figures below. The test engine units are sorted into best performance, average performance and worst performance for better observation and analysis.

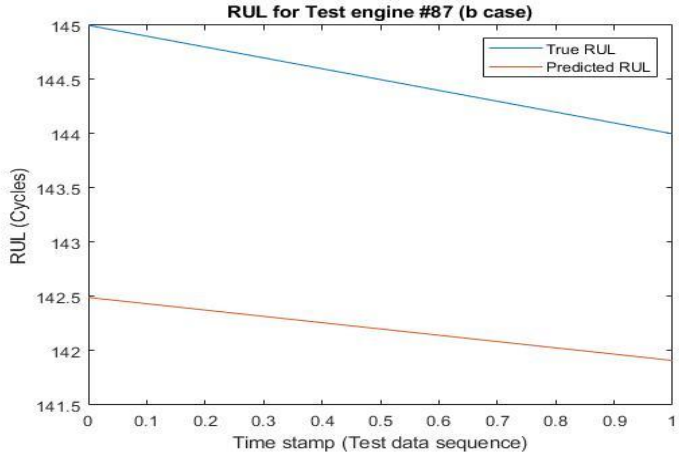

Figure 6: RUL prediction results, best performance for engine 87

The predicted curve and the test curve have a decreasing trend, the first has a lifetime of around 142 cycles (red) while the second has a lifetime of around 144 cycles (blue). As shown in Figure 6, the decrease of the two curves reassures that the remaining useful life time decreases with the number of cycles or the operating time in other words. The prediction indicates that the end of life will occur two days before the actual date, which is significant in the sense that it is better to predict the defect sooner rather than too late.

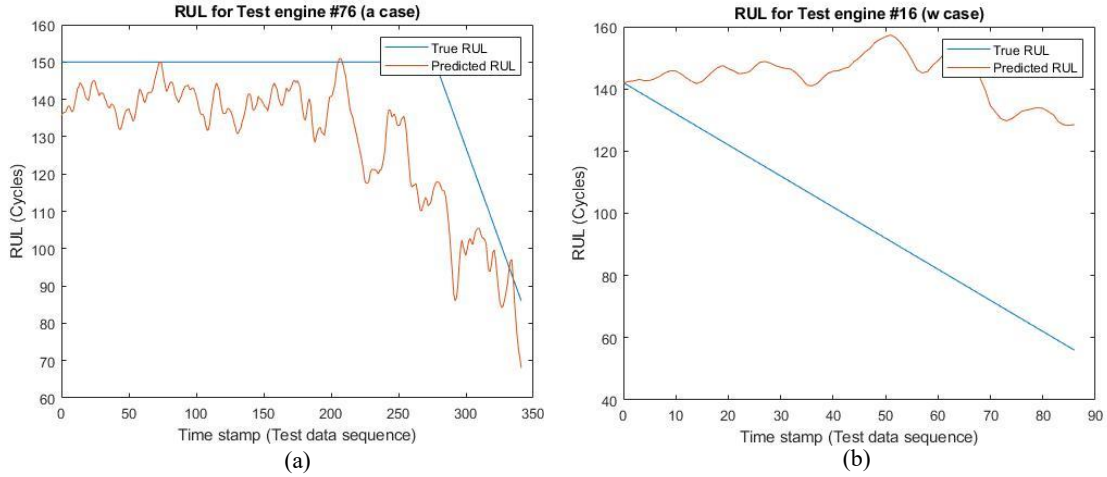

Figure 7: RUL prediction results; average performance for engine 76 (a); worst performance for engine 16(b)

Figure 7 (a) shows the average prediction performance, for engine 76. We see fluctuations in the predicted curve and stabilizes around 70 cycles (red) while the actual curve is smooth and stabilizes around 90 cycles (blue).

Figure 7 (b) shows the worst prediction, this is reached for engine 16. The predicted curve has fluctuations and stabilizes around 130 (red) while the actual curve is smooth and stabilizes around 60 (blue). The fluctuations reflect the sensitivity of the prediction algorithm which calculates the useful life at each iteration without taking into account the previous one.

The prediction of figure 7 (a) is 20 cycles off while that of Figure 7(b) is 70 cycles off. This demonstrates that the algorithm has difficulty in generalizing its learning to the prediction of RUL for these engines.

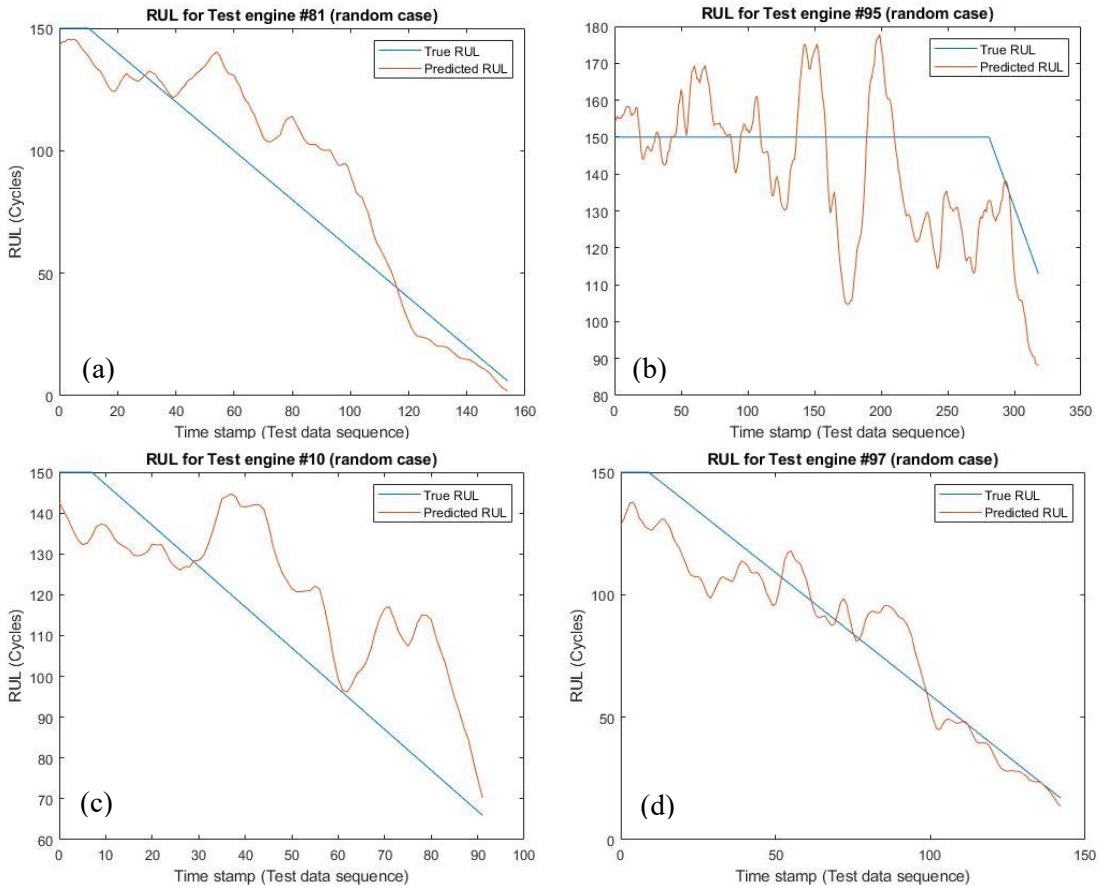

Figure 8: Random RUL prediction for engines 81, 10, 97, 95

Figure 8 (a) shows the prediction for engine 81. The predicted curve has slight fluctuation and is closer to reality at the beginning of the cycle and at the end of life.

Figure 8 (b) shows the prediction of the randomly chosen engine 95. The predicted curve fluctuates a lot but remains around an average value close to the real value. It is closer to reality at the beginning of the cycle and at the end of life with a difference of 20 days.

Figure 8 (c) shows the prediction for engine 10. The predicted curve is sometimes far away sometimes close to the actual curve but is very close at the end of life.

Figure 8 (d) shows the prediction for engine 97. The predicted curve is very close to the actual curve at the beginning and even merges at the end of life.

Figure 8(a),), figure 8 (b), figure 8 (c) and figure 8 (d) show that the prediction is more accurate when approaching the end of engine life and generally at the start of operation as well. This was predictable, because as observed in figure 3, the readings of certain sensors are strictly monotonous when approaching the end of the engine's life, this trend is therefore detected by the prediction models, which facilitates its prognosis. It can be observed that the RUL values predicted by the proposed methods are generally close to the real values. In particular, the prognostic accuracy tends to be higher in the region where the RUL value is low. Indeed, when the motor unit operates in fault mode, the fault function is enhanced and can be captured by the proposed network for better prognosis.

Note that the RUL estimates for the latter parts of the engine life are not shown. This is because in the test data set, the last parts of the sensor measurements are not provided in order to examine prognostic performance. It can be observed that in the early periods in the 4 cases, the proposed method estimates the RUL values close to the constant RULlim. Then the estimates decrease almost linearly with time until the end of the available test samples. Specifically, despite some noticeable error existing between predictions and actual RUL values in general, prognosis accuracy is high, especially when motor units are close to failure.

Figure 9 presents a capture of the evolution of the RMSE objective function and loss function during training.

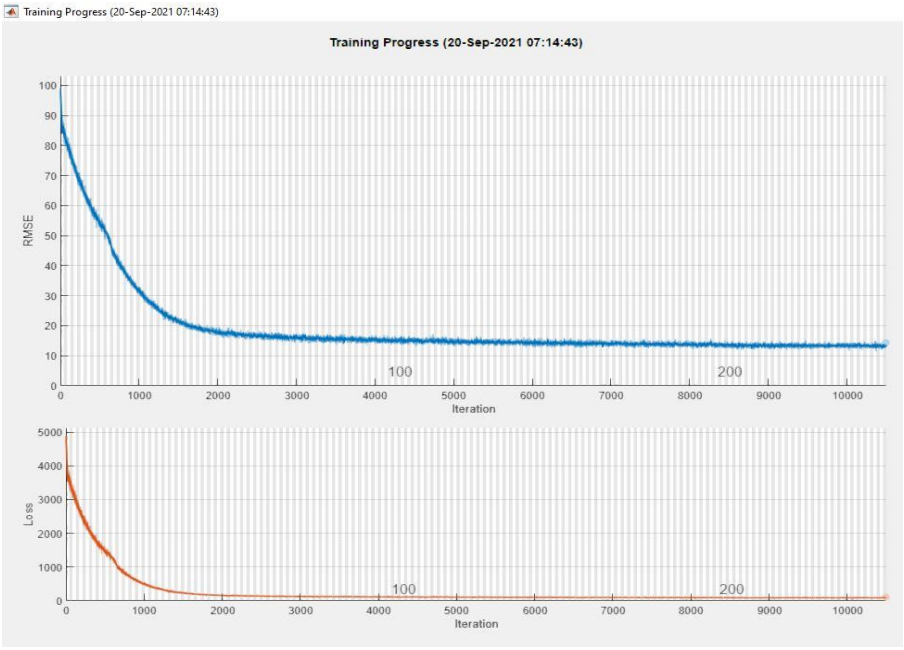

Figure 9: Objective function curve (blue) and loss function (red) as a function of mini lots

Both curves haves a similar decreasing overall trend and stabilize around a certain value (RMSE = 12). However, slight local fluctuations can be observed. These results are in line with expectations, because the objective is to have a convergence of the cost function towards a global minimum.

The overall decrease of the two cost functions reassures that the neural networks actually learns from the samples provided to it in the data set. The results obtained are compared with the results of the literature in table 5. It is observed that the training curve tends to descend at the 250 th epoch, which means that the prognostic performances could be improved if the number of epoch to 1000 for example. It can be considered that the performance discrepancies between the literature and the present study are related to this. However, hardware constraints prevent us from increasing the number of epochs above 250. The RMSE drops rapidly at the beginning to stabilize and decreases slowly thereafter, this is due to the characteristics extracted at the start being coarse and very basic. As one makes iterations, one seeks to optimize the function loss, from where a slowing down of the average quadratic error. The following histogram in figure 10 shows the distribution of the RMSE values over all the test engines.

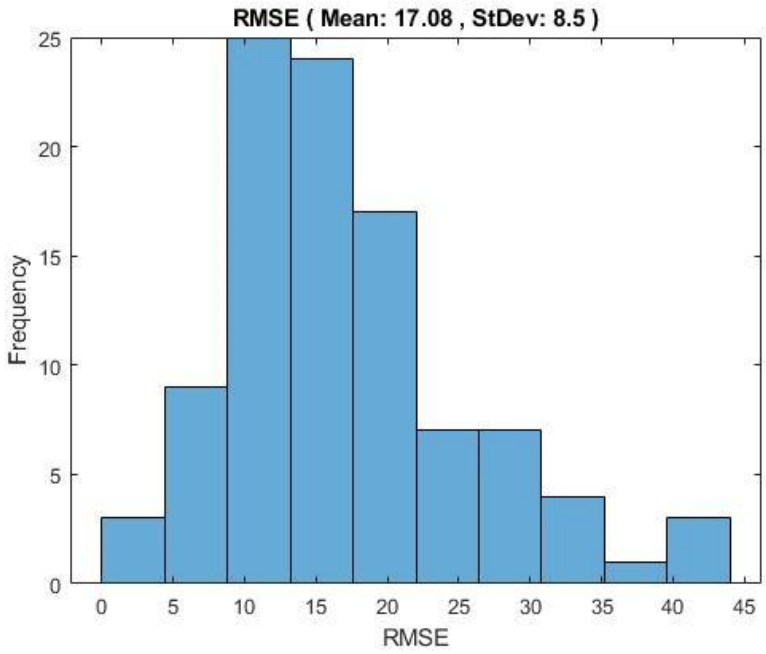

Figure 10: Distribution of RMSE values across all test engine

Note that the quadratic error is maximum around engine 15, which corresponds to poor generalization performance for engines whose indices are 11, 12, 13, 14, 15, 16, and 17. Furthermore, the RMSE value without adequate pre-processing of 19,76 [20] shows that data from useless sensors is very detrimental to learning, from which we can recommend that maintenance teams do not take these sensors into account to preventive maintenance work to determine the remaining useful life or the importance of knowing how each sensor behaves at the start of the process.

3.3. Effects of number of convolution layers

It can be observed that generally, more hidden convolution layers lead to lower RMSE values. This indicates that the deep architecture is able to capture more useful information than the shallow ones. However higher prognostic accuracy can be obtained by a deeper structure, the computation time for the learning process increases almost linearly with the number of hidden layers. It should be noted that neural networks with 5 convolution layers achieve good performance with medium computational load, and that is used as the default hidden layer number of the architecture proposed in this study. A comparison between RMSE and the scoring function is provided in figure 11. It is observed that both functions have curves that vary, although there is a gap between the magnitude scales. They are therefore both relevant for the evaluation of prognostic performance in this case study. The peaks in the score curve are due to the good performance of the system to operate.

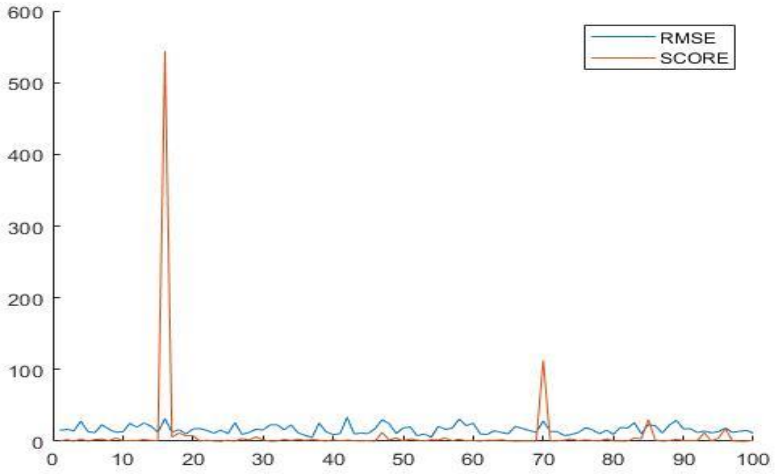

Figure 11: comparison between the RMSE and the score function

3.4. Performance and prognosis of LSTM type deep neural networks

The training data was ordered according to run length to reduce padding effect during mini-batch separation as shown in figure 12.

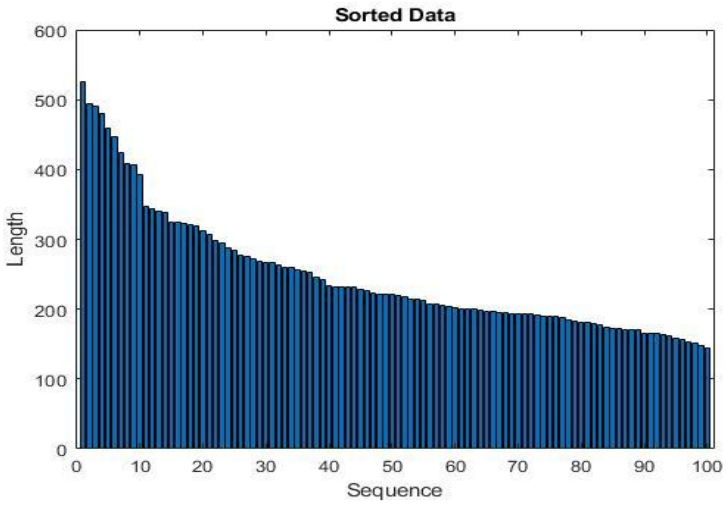

Figure 12: Training data classified according to sequence length

The data sequences are arranged from largest to smallest. This order allows LSTM networks to better extract the temporal characteristics between sequences, especially during training, hence it is recommended to set the "breath" parameter to "never". The RUL prediction results are shown in figures 13 through 17. The test engine units are sorted by labels from small to large for better observation and analysis.

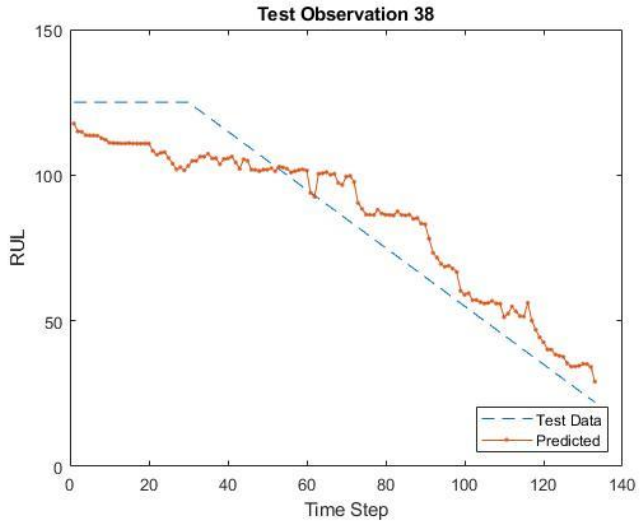

Figure 13: RUL prediction results, best performance for engine 38

The predicted and test curves have a similar trend. The similar tendencies of the two curves and the proximity between them reassure that the prognosis is very close to reality.

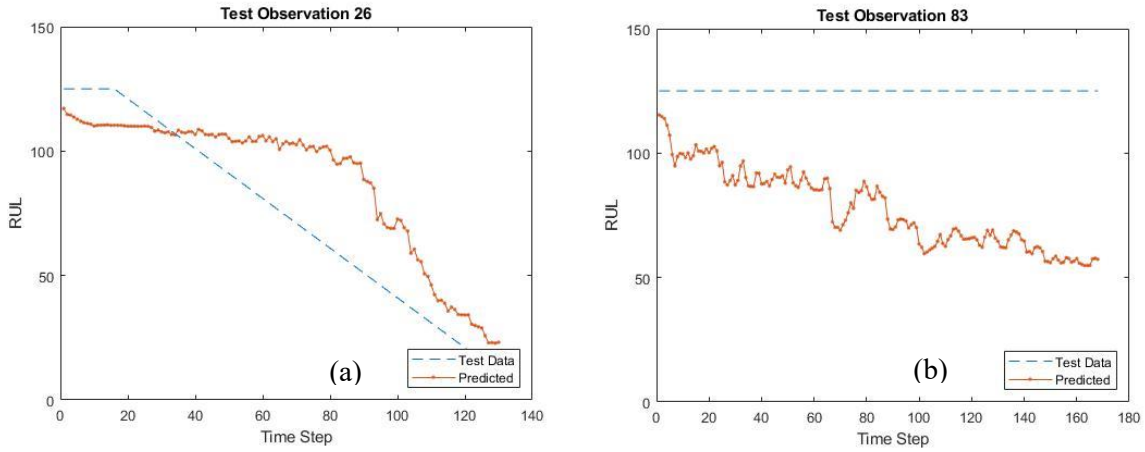

Figure 14: RUL prediction results; average performance for engine 26 (left); worst performance for engine 83 (right)

Figure 14 (a) shows the mean prediction performance, this is achieved for the motor 26. The closer the predicted curve is to the actual curve, the closer the end of life is approached after a sufficiently long time.

Figure 14 (b) presents the worst prediction, this one is reached for the engine 83 (red). The predicted curve has fluctuations. It is closer to reality at the beginning of the cycle but gradually moves away; it stabilizes around 60 while the real curve is stable at 125 (blue).

The fluctuations reflect the sensitivity of the prediction algorithm which calculates the useful life at each iteration without taking into account the previous one.

The prediction of figure 14 (a) is 5 cycles off while that of Figure 14(b) is 60 cycles off. This demonstrates that the algorithm has difficulty in generalizing its learning to the prediction of RUL for these engines.

It can be observed that the RUL values predicted by the proposed method are closer to the real values than the values predicted by CNN. As with CNN; the prognostic accuracy tends to be higher in the region where the RUL value is low. The failure is already almost felt, causing the responsible components to show a degraded performance response. In addition, the RUL estimates for the life of the test engine units before the last recorded cycle are shown in Figure 15. Here we have 4 examples out of one hundred test engine units, with the number unit is respectively 14, 49, 79, and 96, presented for demonstrations.

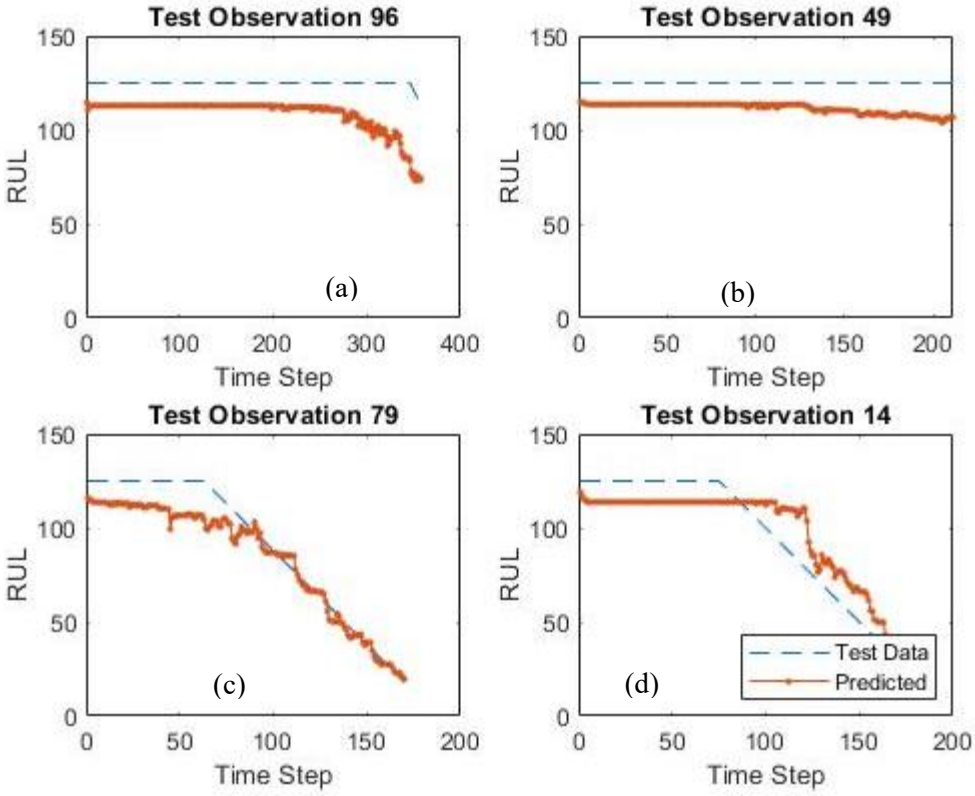

Figure 15: Random RUL prediction for engines 14, 49, 79, 96

Note that that the RUL estimates for the latter parts of the engine life are not shown. This is because in the test data set, the last parts of the sensor measurements are not provided in order to examine prognostic performance. The current RUL value for the last recorded cycles is in the data set, and the corresponding RUL labels for the previous lifetime can be obtained accordingly. It can be observed that in the early periods in the 4 cases, the proposed method estimates the RUL values close to the constant RULlim = 125 which is the value set by the manufacturer for normal use of the engine. Then the estimates decrease over time until the end of the available test samples. Despite some noticeable error existing between predictions and actual RUL values in general, prognostic accuracy is high, especially when motor units are close to failure.

Figure 16: curve of objective function (blue) and loss function (red) according to mini lots

In figure 16, a capture of the training process makes it possible to collect several parameters. The two curves are similar (have the same profile) despite a distortion, have a decreasing overall trend and stabilize around a certain value (RMSE = 12). However, slight local fluctuations can be observed. These results are in line with expectations, because the objective is to have a convergence of the cost function towards a global minimum.

Compared to figure 9, where the distortion is greater, it is noted that the error is minimal by approaching the LSTMs.

The overall decay of the two cost functions reassures that the neural networks are indeed learning from the samples provided to it in the dataset. The results obtained are compared with the results of the literature in table 5. The RMSE falls slowly at the beginning for a notable descent from the epoch 100, this can be explained by the character of the LSTM networks to capture the temporal dependencies at long term. The following histogram shows the distribution of the RMSE values on all the test engines.

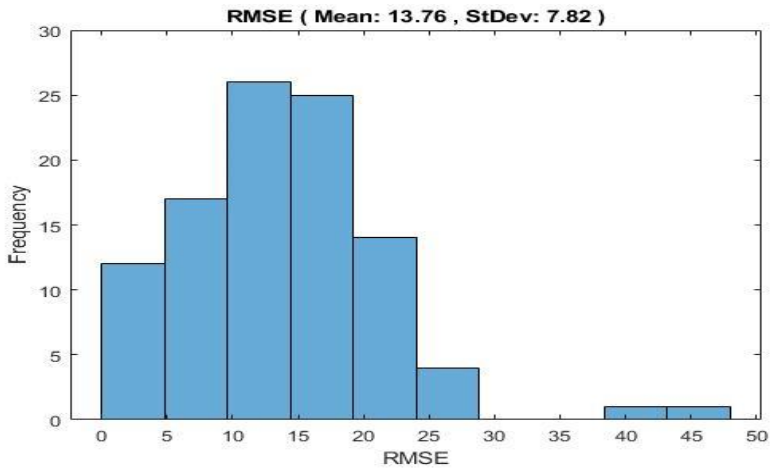

Figure 17: distribution of RMSE values on all test engines

It is noticed that the RMSE value of LSTM neural networks is significantly smaller than that of CNN, which demonstrates a better performance of LSTM compared to CNN. In addition, the RMSE value without adequate pre-processing of engines 22 and 34 [24] shows that data from useless sensors is very detrimental to learning, hence maintenance teams can be recommended not to take these sensors into account. For preventive maintenance work aimed at determining the remaining useful life, provided that the location of the sensors and their classification are entered at the start of the process.

3.5. Effects of number of LSTM layers

We observed that the more LSTM layers there are, the longer the training time. Moreover, it was shown in 3.3 that more hidden convolution layers lead to lower RMSE values. This indicates that the deep architecture is able to capture more useful information than the shallow ones.

On the other hand, while higher prognosis accuracy can be achieved by deeper structure, the computation time for the learning process increases almost linearly with the number of hidden layers. It should be noted that neural networks with 5 convolution layers achieve good performance with medium computational load, and that is used as the default hidden layer number of the architecture proposed in this study.

A comparison between RMSE and the scoring function is provided in figure 18.

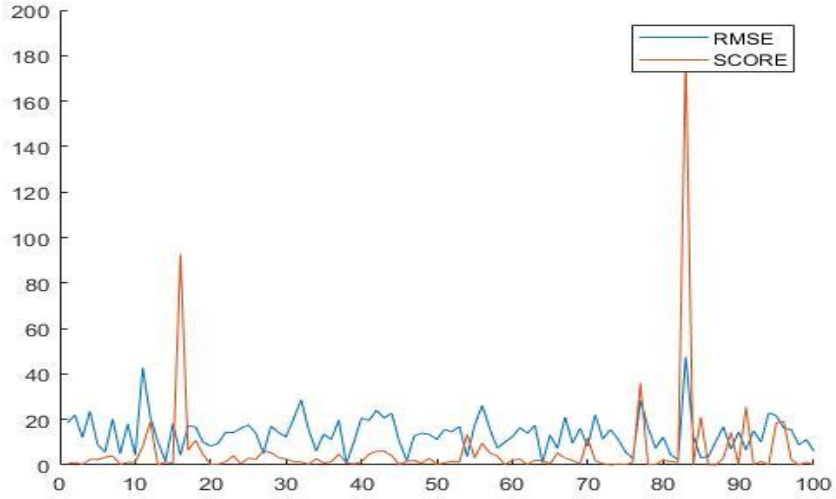

Figure 18: comparison between the RMSE and the score function

It is observed that the two functions have similar trends, although there is a gap and peaks between the magnitude scales. They are therefore both relevant for the evaluation of prognostic performance in the case study.

3.6. Comparison with other architectures

A comparison of the different prognostic performance results using different methods are shown in table 5.

Table 5. Performance comparisons between DCNN and LSTM neural networks and the latest methods on CMAPSS.

N	Methods	RMSE	Score	RULlim	Year
1	3 CNNs + 3 LSTMs + 2 Bi-LSTMs	10.41			2020
2	DAG (CNN/LSTM hybrid) + LSTM	11.96			2019
3	DCNN with RULlim threshold	12.61	273.7	125	2017
4	LSTM	12.81		125	2012
5	RULCLIPPER [21]	13.27		135	2014
6	Bi-LSTM	13.65			2018
7	Sequential Extreme Learning Machine	13.74	641.17		2020
8	**PROPOSED LSTM**	**13.76**	**641.01**	**125**	**2022**
10	NN with time window	15.16			2016
11	LSTM	16.17	338		2018
12	**PROPOSED DCNN**	**17.08**	**1148.07**	**125**	**2022**
13	CNN	18.45	1296.7		2016
14	Support Vector Regression	20.96			2016
15	Revelance Vector Regression	23.80			2016
16	MultiLayer Perceptron	37.56			2016
17	ARIMA SVM	39.68			2019

We can observe that the proposed deep learning method is competitive, compared to several. The results show that the proposed deep learning architecture is well suited to the prognosis problem. The stacked convolution layers contribute to the learnability of the network. The RNN structure and the second best with its information flow.

NN and DNN basic neural networks are also competitive. This indicates that sample preparations with raw feature selection, data preprocessing, and time window application are effective for subsequent feature extraction.

The proposed deep learning method has achieved promising performance compared to state-of-the-art results. It should be noted that in this work, we have artificially set a threshold RULlim for the healthy condition, which has a noticeable effect on the experimental performance. Non-threshold prognostic results are also provided. Despite the increase in the RMSE value of the RUL estimate, the results of the proposed method are still competitive.

In summary, the comparison results presented above suggest that the proposed method is promising for prognostic problems and able to provide reliable RUL estimates in different cases.

4. Conclusion

Prognosis is currently at the heart of industrial system health management. Reliably estimating the state of health of a system holds the promise of enormous cost savings by, for example, avoiding unscheduled maintenance, similarly the prognosis improves safety in the operation of equipment, and allows decision makers to change the operating parameters to extend the life of the components, finally it also allows them to better organize maintenance.

Various fault detection and diagnostic techniques have been developed for solving diagnostic problems. These techniques include model-based approaches, knowledge-based approaches, qualitative simulation-based approaches, classical multivariate statistical techniques, and neural network-based approaches. In this work, a new deep learning method for prognosis is proposed, based on deep convolutional neural networks and long-term memory neural networks. Experiments were conducted on the popular C-MAPSS dataset to show the effectiveness of the proposed methods.

With raw feature selection, data preprocessing and sample preparation using a time window, good prognostic performance is obtained with the CNN method, including an RMSE value of 17.08 and an 8.5 type for test data. Even better performance is obtained with the LSTM network, namely an RMSE value of 13.76 and a standard deviation of 7.8. LSTM neural networks offer better performance probably because of its ability to learn temporal relationships between time series data, unlike CNN neural networks.

We observed that the RUL in engine life can be predicted well, especially for the late period close to failure. In comparison with most of the existing deep methods which take more into account the degradation of the high pressure compressor, the proposed method is more realistic because it also takes into account the degradation of the fan. Moreover, No homemade signal processing functions are needed, such as asymmetry, flattening, etc. Therefore, no prior expertise on prognosis and signal processing is required in the proposed method. We recommend for future research to use data sets that take into account the combined degradation of all engine components rather than just the high pressure compressor and fan. This will constitute a closer approach to reality.

References

[1] X. ZHOU, F. LU, J. HUANG: Fault Diagnosis Based on Measurement Reconstruction of HPT Exit Pressure for Turbofan Engine », *Chinese Journal of Aeronautics*., April 2019.

[2] H. RANTER: Aviation Safety Network Statistics. https://aviation-safety.net/statistics/ (consulted the October 23, 2021).

[3] PINTELON, PARODI-HERZ: Maintenance: An Evolutionary Perspective In Complex System Maintenance Handbook. *Springer*, (1) 21-28, (2008).

[4] P. CHENG, C. YUFENG, C. QING, T. ZHAOHUI, L. LINGLING, G. WEIHUA: A Remaining Useful Life Prognosis of Turbofan Engine Using Temporal and Spatial Feature Fusion. *MDPI*, January 2021.

[5] N. BOLANDER, H. QIU, N. EKLUND, E. HINDLE, T. ROSENFELD: Physics-based Remaining Useful Life Prediction for Aircraft Engine Bearing Prognosis. Annual Conf. of the Prognostics and Health Management Society, 2009.

[6] F. Engine, P. Directorate:Turbofan Engine Malfunction Recognition and Response Final Report. ANE-110, 12 New England Executive Park Burlington, MA 01803, 07/17/2009.

[7] I.C. DUȚU, C. FRAȚILA, T. AXINTE, M.G. MUNTEANU, L. CALANCEA, M. DIACONU, C. DRAGAN: Control Efficiency Improvement of an Electro-hydraulic Winch. *Technium Science Applied Sciences and Technology*. (3) 9, (2021).
[8] Turboréacteur, *Wikipédia*. avr. 14, 2021. Consulté le: juill. 10, 2021. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Turbor%C3%A9acteur&oldid=181900363
[9] FAA (FEDERAL AVIATION ADMINISTRATION): Airplane Turbofan Engine Operation and Malfunctions Basic Familiarization for Flight Crews. Course Jawaharlal Nehru College, Aviation. 2014.
[10] D. QIAN, L. XIANG, S. JIAN-QIAO: Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks. *Elsevier Science*, juin 2017.
[11] Turborfan engine. *Wikipédia*. April 18, 2021. Accessed : July 10, 2021. [Online]. Available on:https://fr.wikipedia.org/w/index.php?title=Turbor%C3%A9acteur_%C3%A0_double_flux&oldid=182030755
[12] N. J. ISSONDJ BANTA, F. OFFOLE, D. ESSOLA, V. T. YOTCHOU, C. V. NGAYIHI ABBE, R. MOUANGUE: Simulation of Performance and Emissions Related Parameters in a Thermal Engine Using a Deep Learning Approach. *SN Computer Science*, under exclusive licence to Springer Nature Singapore, 17 November 2021.
[13] D. ESSOLA, F.OFFOLE, K. T. FOHOUE, L. T. NJENJI, R. KOUSSOUKE: Mapping and functional optimization of control parameters in W18V50DF dual-fuel engines during combustion. *Eng. Res. Express,* 3 May 2021.
[14] Florence OFFOLE, Dieudonné ESSOLA, Nelson ISSONDJ and Charly BOUHEUL. Failure Prediction of Highly Requested Complex Technical Systems: Application to W18v50df Engines./ International Journal of Advances in Scientific Research and Engineering (ijasre). E-ISSN : 2454-8006 DOI: 10.31695/IJASRE.2020.33798 Volume 6, Issue 4 May – 2020.
[15] Three Ways to Estimate Remaining Useful Life for Predictive Maintenance - MATLAB & Simulink.https://www.mathworks.com/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html (Accessed: September 12, 2021).
[16] Predict Remaining Useful Life - MATLAB & Simulink. https://www.mathworks.com/help/predmaint/predict-remaining-useful-life.html (Accessed: September 12, 2021).
[17] P. LARA-BEN´ITEZ, M. CARRANZA-GARCIA, J. C. RIQUELME: An Experimental Review on Deep Learning Architectures for Time Series Forecasting », *International Journal of Neural Systems*, (31), p. 28, 2021.
[18] S. KALEKAR, PRAJAKTA: Time Series Forecasting using Holt-Winters Exponential Smoothing. *Kanwal Rrkhi School of Information Technology*, December 6, 2004.
[19] XINXIN, HUI YANG, NONGCHENG AND L. OING: Remaining Useful Life Prognostics of Aircraft Engines Based on Damage Propogation Modeling and Data Analysis. 8th International Symposium on Computational Intelligence and Design (ISCID), 2015.
[20] Remaining Useful Life Estimation using Convolutional Neural Network - MATLAB & Simulink. https://www.mathworks.com/help//predmaint/ug/remaining-useful-life-estimation-using-convolutional-neural-network.html (Accessed: September 20, 2021).
[21] Z. ZHAO, L. BIN, X. WANG, W. LU: Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 2017, Vol.164, 74-83.
[22] Sequence-to-Sequence Regression Using Deep Learning - MATLAB & Simulink ». https://www.mathworks.com/help/deeplearning/ug/sequence-to-sequence-regression-using-deep-learning.html (Accessed: September 20, 2021).
[23] C. ZHANG, P. LIM, A. K. QIN, K. C. TAN: Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE*, (99), 2016, 1-13.
[24] F. O. HEIMES: Recurrent Neural Networks for Remaining Useful Life Estimation. Proceedings of Inter. Conf. on Prognostics and Health Management, 2008, 1-6.

Technium Vol. 4, Issue 6 pp.58-83 (2022)
ISSN: 2668-778X
www.techniumscience.com

Abbreviations
RUL: Remaining useful life
CMAPSS: Commercial Modular AeroPropulsion System Simulation
LSTM: Long Short Terms Memory
ANN: Artificial Neural Network
RNN : Recurrent neural network
CNN: Convolutional neural network
MLP: Multilayer perceptrons
HPC: High Pressure Compressor
LPC: Low Pressure Compressor
EGT: Exhaust Gas Temperature
RMSE: Mean square error
SM: Stall Margin
SmFan: Fan stall margin
SmLPC: Stall margin of LPC
SmHPC: Stall margin of HPC
PCNfR: Corrected blower speed requested
EPR: Engine Pressure Ratio