Learning methods for remaining useful life prediction in a turbofan engine *

Andréia Seixas Leal[1][0000−0001−6291−3284], Lilian Berton[1][0000−0003−1397−6005], and Luis Carlos de Castro Santos[2,3][0000−0001−6158−6517]

[1] Universidade Federal de São Paulo, São José dos Campos/SP, Brazil
andreia.leal@unifesp.com, lberton@unifesp.com
[2] Universidade de São Paulo, São Paulo/SP, Brazil
lsantos@ime.usp.br
[3] Embraer S.A.
luis.castro@embraer.com.br

Abstract. In industry 4.0 there is a growth in the Industrial Internet of Things (iIoT) with a lot of information generation and consequent big data challenges. Thus, it is imperative to have techniques able to process all this data and predict the maintenance of equipment and systems. The development of algorithms for remaining useful life (RUL) estimators is critical for the full functioning of the company's assets. Especially the aeronautical sector needs to guarantee safety and quality flights. The turbofan, a propulsion engine, is a critical element for an airplane operation. This paper proposes a model to perform prediction of the remaining useful life of an aircraft's turbo engine. In this work, we focus on the run-to-failure data from an N-CMAPSS turbofan, the data used were provided by NASA in 2021. After training and validating different algorithms such as MLP and CNN, we find CNN as the best approach with an RMSE of 9.11, a score of 5.14, and computed score of 1.17. The results have improved when compared to the literature over 25% in RMSE and 15% in computed score.

Keywords: Industry 4.0 · RUL · PHM · Machine Learning · Neural Network · Aeronautical · Turbofan.

1 Introduction

The current manufacturing companies have been heavily affected by the globalized economy, which conveys the constant pursuit of increased productivity, high-quality products, reduction of the product life cycle and the processes of digitalization [33]. The application of new technologies and usage of numerous types of sensors in different sectors of the market makes companies produce an increasing volume of data and become data companies [13]. Data management in the big data environment is critical for achieving self-aware and self-learning machines and for supporting manufacturing decisions. Data analytics is categorized into three main stages characterized by different levels of difficulty, value, and intelligence [6]. In a study by Forrester Consulting, 98% of organizations surveyed said analytics are important in driving business priorities. However, less than 40% of workloads are using advanced analytics or artificial intelligence (AI) [5]. Through AI, industrial production can achieve higher efficiency, and autonomy and reduce costs. AI is also a fundamental base of Industry 4.0.

In the modern economy, the dependence on equipment in industrial production is inevitable, so equipment maintenance is a critical activity for any sector that involves equipment and systems. Repairing the production line or equipment after the breakdown can be more costly than conducting predictive maintenance ahead of the breakdown [19]. Predictive maintenance performs maintenance schedule based on the prediction and analysis of the state of equipment in general. This maintenance can save a lot of time, cost, and energy as it avoids the need to carry out unnecessary maintenance activities periodically [22]. The prediction can be made by analyzing the measurement data from the sensors used and then a model is developed and used to predict the machine failure before it happens [15].

* This work was possible with the support of DAI CNPq.

The association of big data with the importance of maintenance corroborates the development of algorithms for the prediction of remaining useful life (RUL). This task is part of prognostics and health management (PHM), which uses prognostic information to help improve safety, condition-based maintenance, and product life extension. By studying the behavior of the equipment during its useful life, RUL techniques can allow the estimation of reliability and degradation over time, or the time for the occurrence of failures [8].

The methodologies of remaining life determination generally fall into two broad categories: (1) model-based methods and (2) data-based methods. In (1) starting from a solid idea of how a physical system works and the states or events that are important to be detected, a hypothesis is generated about which aspects can be discarded and the target morphology is achieved. Obtaining confirmations of the correlations between what is recorded and what is desired in the detection, an anomaly detection system (in this case failures or degradation) is designed. In (2) an approach that allows finding correlations and connections directly from the data, using statistics or machine learning (ML). The advantage of the data-based method (2) is that, unlike (1), it does not require prior knowledge and can be performed directly if there are good quality data [31]. Among them, machine learning is a very common data-driven method that has been widely used in the field of RUL prediction.

The aerospace sector has many studies to determine estimators for RUL due to the severity of equipment failure in an airplane. Among these, the turbofan engine is the most critical power unit for the aircraft and of course, its safety and stable operation are of utmost importance. However, the turbofan engine usually works in an extremely complex and hostile environment, which makes it imperative to predict its useful life [13, 15].

NASA makes available a series of data sets from satellites, aircraft, and planes. Among them, is the turbofan engine degradation data. The first data made available by NASA was generated by the dynamic model simulation of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). CMAPSS is a high-fidelity computational model for simulating a large commercial turbofan engine, which allowed the construction of the Turbofan data set - 1 (DS01) [17]. The DS01 data set was built from samples from a system that was already degraded. Therefore the beginning of the failure state cannot be predicted but only the evolution of the failure. In 2021 a new data set named N-CMAPSS was built, containing the complete history starting in a good condition of the equipment and going up to the state in which the failure occurs [1].

Although big data analytics has been used for real-time PHM for this dataset, their utilization in decision-making algorithms is still in its early stages. Currently, deep learning architectures have been widely used in data-based implementations for RUL estimation of a turbofan engine [4, 27, 20, 10, 31, 14], most studies use the data set DS01 from [17]. These works are mainly focused on simulation degradation data. However, a new realistic data set turbofan dataset-2 (DS02) from [18] of run-to-failure turbofan engine degradation was published in 2021. DS02 presents a significant difference from the simulation data, its main challenge is the difference in flight duration for each cycle [13].

This work employed different models, such as the traditional algorithm Multi-Layer Perceptron (MLP) and the deep learning approach Convolutional Neural Networks (CNN). The implementation's purpose was to develop a model that is a good estimator for predicting the remaining useful life of an aircraft's turbofan engine. The models were built based on turbofan engine data from NASA's Prognostics Data Repository (2021) in [18]. Using a training set, a model was built and validated with a test data set. The results are competitive with literature.

2 Related work

Previous works proposed techniques and methodologies to predict Turbofan aircraft engines' remaining useful life (RUL). Past research on the original CMAPSS data set demonstrated the applicability of convolutional neural networks (CNNs) [11, 29], long short-term memory (LSTM) [2, 32, 28] and the combination of CNN and LSTM [30].

Regarding the new version of the Turbofan dataset (N-CMAPSS), it was used in the 2021 PHM Society Data Challenge. The winner solution [12] used a deep convolutional neural network that can take inputs of variable length. The authors also pre-processed the data by training a feedforward

neural network on a non-degraded subset of the data that maps from the flight conditions to the sensor outputs.

The second place was reached by [3] that proposed a deep convolutional neural network with inception architecture for estimating the RUL of turbofan engines. The third place was reached by [25] which employed two deep convolutional neural networks for turbofan engine degradation. The first network extracts a low-dimensional feature vector using the normalized raw data as input and the second ingests these vectors and estimates the RUL.

In [9] a virtual health indicator was developed for degradation monitoring of safety-critical engineered systems that operate under time-varying conditions. It incorporates limited domain knowledge with a two-phase heuristics approach to select a causal driver and a set of measuring parameters. With many learning architectures: long short-term memory and autoencoder combined (LSTM-AE), autoencoders (AE), anti-causal regression (RR+), and a simple multi-layer perceptron (MLP+). The performance of all health indicators was tested for each of the units in terms of root mean square and the PHM of the RUL in cycles, it was not presented the average results. In [9] a framework with anti-causal learning is said to outperform existing deep learning architectures, reducing the average RMSE across all investigated units by nearly 65%.

3 Methodology

This section contains data explanation, problem definition and the steps involved in processing the data and implementing the proposed architectures.

3.1 Data

The data sets used in this work were obtained using N-CMAPSS and were made available by [18]. These data are composed of several multivariate time series of sensor measurements made for each operating cycle of a turbofan engine that was simulated. Over time, engines start to degrade and the goal is to predict the number of cycles to failure (RUL). Each dataset is divided into subsets, which may have different numbers of failure modes and operating conditions.

As stated in the introduction, the construction of the new N-CMAPSS data set was made to contain the complete history of the trajectories, starting from a good condition of the equipment until the state in which the failure occurs. The DS02 contains information regarding ninety simulated flights, extracted from the N-CMAPSS [18].

Table 1. Scenario descriptors - w

♯	Symbol	Description	Units
1	alt	Altitude	ft
2	Mach	Flight Mach number	-
3	TRA	Throttle-resolver angle	%
4	T2	Total temperature at fan inlet	°R

The problem revolves around the development of a model capable of predicting the RUL of the system, using four types of data. First, the scenario descriptors w: altitude (alt), flight Mach number (Mach), throttle-resolver angle (TRA), and total temperature at fan inlet (T2). Second, the sensor outputs x_s: fuel flow (Wf), physical fan speed (Nf), physical core speed (Nc), total temperature at LPC outlet (T24), total temperature at HPC outlet (T30), total temperature at HPT outlet (T48), total temperature at LPT outlet (T50), total pressure in bypass-duct (P15), total pressure at fan outlet (P21), total pressure at LPC outlet (P24), static pressure at HPC outlet (Ps30), total pressure at burner outlet(P40), and total pressure at LPT outlet (P50). Furthermore, we have the data from the virtual sensors x_v. Lastly, the auxiliary data a: flight constant (Fc), and equipment health status (hs). These features and their units are shown in the tables 1 to 4 [1].

Seven different failure modes were defined related to flow degradation or efficiency of subcomponents that may be present in each flight. Flights are divided into three classes (Fc) depending

Table 2. Measurements - x_s

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

on the duration of the flight. In class 1, autumn flights last 1 to 3 hours, class 2 comprises flights between 3 and 5 hours, and class 3 contains flights lasting more than 5 hours. Each flight is divided into cycles, covering climb, cruise and descent operations [1].

Table 3. Virtual Sensors - x_v

♯	Symbol	Description	Units
18	T40	Total temp. at burner outlet	°R
19	P30	Total pressure at HPC outlet	psia
20	P45	Total pressure at HPT outlet	psia
21	W21	Fan flow	pps
22	W22	Flow out of LPC	lbm/s
23	W25	Flow into HPC	lbm/s
24	W31	HPT coolant bleed	lbm/s
25	W32	HPT coolant bleed	lbm/s
26	W48	Flow out of HPT	lbm/s
27	W50	Flow out of LPT	lbm/s
28	epr	Engine pressure ratio (P50/P2)	-
29	SmFan	Fan stall margin	-
30	SmLPC	LPC stall margin	-
31	SmHPC	HPC stall margin	-
32	NRf	Corrected fan speed	rpm
33	NRc	Corrected core speed	rpm
34	PCNfR	Corrected core speed	pct
35	phi	Ratio of fuel flow to Ps30	pps/psi

Table 4. Auxiliar Data - a

♯	Symbol	Description
36	Fc	Flight constant
37	hs	Equipment health status

3.2 Problem definition

This work goal is the development of a model G capable of predicting the remaining useful life Y of the system, using the scenario descriptors (w), the sensor measurements (x_s), virtual sensors (x_v), and auxiliary data (a). The length of the sensory signals w and x_s is not constant, the model needs to incorporate variable lengths of input data. The optimization problem can be denoted as: $\arg\min \sum S(y - \hat{y})$. Where y and $\hat{y} = G(w, x_s, x_v, a)$ are the expected and estimated RUL. S is a scoring function defined as the average of the root-mean-square error (RMSE) or any of the other metrics that will be presented in section 3.4. In this case, RUL is the predicted number of cycles to failure.

3.3 Pre-processing

The data attributes consist of the characteristics that make up the states of the turbofan engine. A thorough data analysis was carried out and the following attributes were selected: alt, Mach, TRA, T2, Wf, Nf, Nc, T24, T30, T48, T50, P15, P21, P24, Ps30, P40, P50.

As multiple sensors will produce multiple features, there are different levels of influence on prediction results for different dimensions. In order to eliminate the influence of different data dimensions on the prediction accuracy of the model, the data are usually normalized with *z-score* [26, 16]. Mean and variance are computed for each validation set in training and are used to normalize the validation data. The selected features that will compose the neural networks input are those presented in Table 5 (the data are already normalized in the table).

Table 5. Features from data and their description

♯	mean	std	min	25%	50%	75%	max
alt	0.444286	0.250428	0.000000	0.235733	0.459790	0.655313	1.000000
Mach	0.743903	0.159245	0.000000	0.621803	0.770104	0.869869	0.999579
TRA	0.598073	0.276963	0.000000	0.385445	0.691375	0.839623	1.000000
T2	0.574023	0.174039	0.000000	0.433476	0.573452	0.726228	1.000000
T24	0.554066	0.138047	0.000205	0.457650	0.533719	0.642121	0.997950
T30	0.562168	0.140717	0.000000	0.470928	0.555018	0.642536	0.996721
T48	0.673533	0.112808	0.000000	0.601374	0.681773	0.744591	1.000000
T50	0.648686	0.090298	0.000000	0.587279	0.630993	0.701897	1.000000
P15	0.450200	0.195194	0.000386	0.288172	0.429492	0.602082	1.000000
P2	0.462437	0.209776	0.000000	0.285250	0.435201	0.638353	1.000000
P21	0.450022	0.195267	0.000000	0.287931	0.429314	0.601971	1.000000
P24	0.434952	0.173394	0.000008	0.296277	0.415502	0.559120	1.000000
Ps30	0.401762	0.152141	0.000000	0.290850	0.363922	0.487988	1.000000
P40	0.402490	0.152244	0.000043	0.291443	0.364889	0.488995	0.999622
P50	0.430238	0.213956	0.000000	0.251239	0.395621	0.594237	0.999878
Nf	0.610273	0.219932	0.000000	0.477884	0.673676	0.788916	1.000000
Nc	0.571083	0.142057	0.000000	0.480133	0.567537	0.654085	1.000000
Wf	0.403752	0.140458	0.000000	0.305801	0.365684	0.474504	1.000000
Fc	0.593950	0.491094	0.000000	0.000000	1.000000	1.000000	1.000000
hs	0.207140	0.405257	0.000000	0.000000	0.000000	0.000000	1.000000

Then, after performing the normalization of the data, a sliding window in time is used to scroll through them. Generally speaking, as the name suggests, this technique involves a window that is formed over a piece of data. This window can slide over the data to capture different parts of it. The window size used is defined during the model definition phase. Each data window is used as a network input.

The labels, for defining the base value of the RUL, were defined as a linear function of the cycles for each unit. Then, this linear function is defined as the difference between the total life of the unit in cycles and the number of cycles passed since the beginning of the experiment in time.

Thus, after the data are prepared and normalized, it is necessary to build the network model. The models were trained using the RMSE as the loss function, in addition to other metrics such as Mean Absolute Error (MAE), the computed score [24], and the combination score. These metrics will be defined in the next section.

3.4 Metrics

The performance of the models is evaluated using the metrics described in this subsection. First, the mean absolute error (MAE) is a measure of error between paired observations, this method used the difference between values predicted by a model and the values observed. MAE is calculated as the sum of absolute errors divided by the sample size:

$$MAE = \frac{\sum_{i=1}^{n_v} \left(y^{(i)} - \hat{y}^{(i)} \right)}{n_v} \tag{1}$$

where n_v is the number of test samples, y is the actual, and $\hat{y}$ is the predicted RUL values.

Additionally, the root-mean-square error (RMSE) is more frequently used than MAE as a measure of the differences between predicted and observed values.

$$RMSE = \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \left(y^{(i)} - \hat{y}^{(i)} \right)} \tag{2}$$

The computed score (CP score) was represented using [24] as a reference:

$$CP\ score = \frac{1}{n_v} \sum_{i=1}^{n_v} \exp \left(\alpha \times |y^{(i)} - \hat{y}^{(i)}| \right) \tag{3}$$

$$\alpha = \begin{cases} \frac{1}{13}, & \text{if} \quad |y^{(i)} - \hat{y}^{(i)}| \leq 0. \\ \frac{1}{13}, & \text{if} \quad |y^{(i)} - \hat{y}^{(i)}| > 0. \end{cases} \tag{4}$$

Finally, a score was calculated using the combination of RMSE and the computed score using the following proportions:

$$CO\ score = \frac{RMSE}{2} + \frac{computed\_score}{2} \tag{5}$$

3.5 Convolutional Neural Networks (CNN)

The model that uses the CNNs was built according to the architecture shown in Figure 1. The objective is to find a good model to produce a time-window coding of the raw input signals. This encoding is necessary due to the high dimensionality of the input data. In addition to reducing the size of the input, another objective is to remove as much noise as possible. The ultimate goal of this model is to define a good RUL estimator. CNN uses parameter sharing and sub-sampling to extract maps with more significant features. The main operations on a CNN are convolution and "probing". The convolution operation implements the sharing of parameters and local receptive fields.

The kernel array is sized and loops through the input array looking for patterns. The *pooling* operation scales down by applying a statistical operation to each region of the input. *Max-pooling* was also used, which reduces the input along its spatial dimensions (height and width) taking the maximum value in an input window (of defined size) for each input channel. The window is shifted along each dimension. The *max-pooling* operation reduces the computational requirements for the upper layer. It also reduces the number of parameters for the fully connected upper layers, as well as helps to mitigate the risk of *over-fitting*.

This neural network has several hyper-parameters that can be adjusted, in addition to the sliding window size adjustment. The hyper-parameters of this work were defined based on what

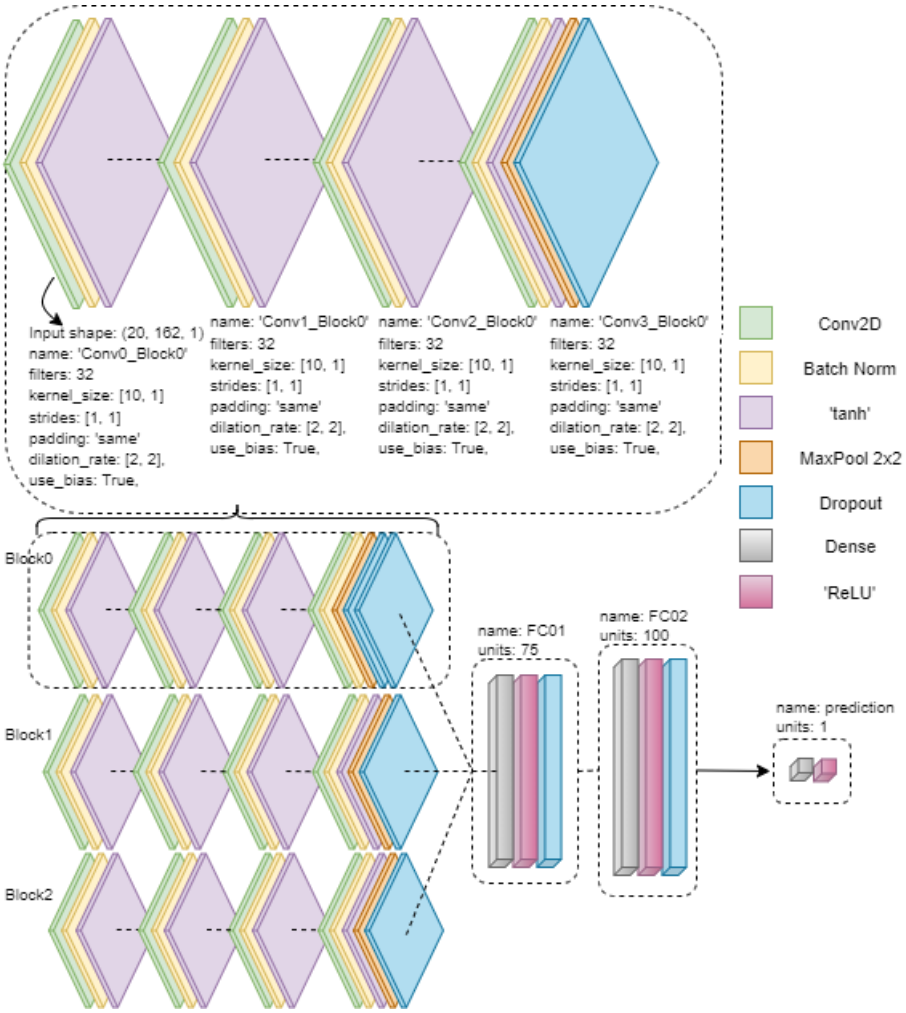

Fig. 1. Representation of the CNN network architecture.

was found in the literature [3, 25, 9], but mainly by trial and error. There were 100 training epochs with batch size of 63 and an input batch format of [20, 162, 1].

The models were trained using RMSE as a loss function. The RMSE is used as a reference metric by the author who provided the data from this set [18, 1]. A stopping criterion of 25 epochs was defined to end the training if there is no improvement in the loss. The learning rate decreases by a factor of 1E-1 if there is no improvement in the last 10 epochs.

3.6 Multi-layer Perceptron (MLP)

The model that uses the MLPs was built according to the architecture shown in Figure 2. One more time a time window was defined to find a good model, producing a time-window coding of the raw input signals.

The input of the MLP network is formed from the flattening of the 20 features selected with 162 window sizes, i.e., $20 \times 162 = 3240$. The first layer (FC01) has 712 neurons that receive input of size 3240. In the sequence, there is a second (FC02) and third (FC03) layers that contain 100 and 75 neurons, respectively. As the purpose is to predict the RUL the last layer has only one neuron. The hyper-parameters of this work were defined mainly by trial and error.

The model was also trained using RMSE as a loss function [18, 1]. The same stopping criterion of 10 epochs to end the training if there is no improvement in the loss is used. As well a learning rate that decreases by a factor of 1E-1 if there is no improvement in the last 10 epochs was defined.

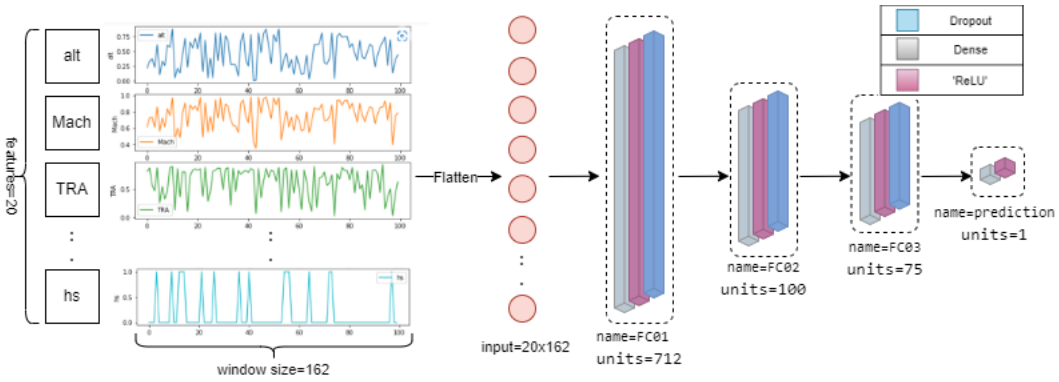

Fig. 2. Representation of the MLP network architecture.

4 Results

The data with normalized parameters used for training, which were summarized in Section 3.1, were splitted with 80% for training and 20% for testing. The execution of the algorithm was performed using CUDA 11.5 and cuDNN 8.2.2, on a computer $i7-1165G7$ with 16GB of RAM and 500GB SDD.

The used CNN architecture is represented in Figure 1. The top four blocks (enlarged) in the figure contain layers composed of convolution, *batch* normalization, activation functions, *max-pooling* and *dropout*.

The bi-dimensional convolutional layers used kernel size of [10,1] with jump size 1. Normalizing the *batches* between layers makes the optimization scenario significantly smoother. This smoothness induces more predictable and stable behavior, allowing for faster training [7, 23]. The activation function used for the convolutional layers was the *tanh* (hyperbolic tangent). The *max-pooling* layer has *pool size* [2,2] and jumps of [2,2]. *Dropout* and *max-pooling* were performed in the fourth convolutional layer, with a rate of approximately 0.13.

The architecture explained above (Block0) is represented in the highlighted block at the top of Figure 1. Three blocks of this structure (from *Block0* to Block2) were used and represent the number of channels in our network. The results of these three blocks are '*flattened*' using *Flatten()* and fed to the fully connected layers FC01 and FC02. The last fully connected layer, which will generate only one output unit referring to the prediction, has been named *prediction* in the figure.

The last three blocks presented in the architecture of Figure 1, which represent fully connected layers (FC01, FC02 and *prediction*), use the activation function *ReLU* and *dropout* in a rate of 0.13.

Adam was the optimizer, with an initial learning rate of 1E-03, and remembering that the learning rate decreases by a factor of 1E-1 if there is no improvement in the last 10 epochs. After the last iteration, the following were obtained for the CNN model: RMSE 9.11, score 5.14 and the CP Score of the set was 1.17. The learning rate finished at 1E-07.

In contrast, Figure 2 presents the architecture used to implement the MLP. The left blocks in the figure contain features with their graphs representations which are flattened to form the first layer input. The flattened input goes to the first layer, and second and third layers with 712, 100, and 50 neurons, respectively. The MLP model results are worse than CNN (Table 6), as depicted in the Figure 3 curves.

Figure 3 shows the RMSE loss curves with different models (CNN, and MLP).

It is important to consider that lower RMSE values indicate a better fit of the model and that the hidden validation set to be scored was composed of 38 units. Given this, the results of the predictions generated by the final solution are exhibited in Table 6.

The article [9] was published this year and stated that there was a reduction of the average RMSE in all the investigated units by almost 65%, however, the results presented by the authors were inferior to this work. Note that the results presented for LSTM-AE (network resulting from the combination of Long short-term memory and auto-encoders), auto-encoders (AE), RR+ (least-

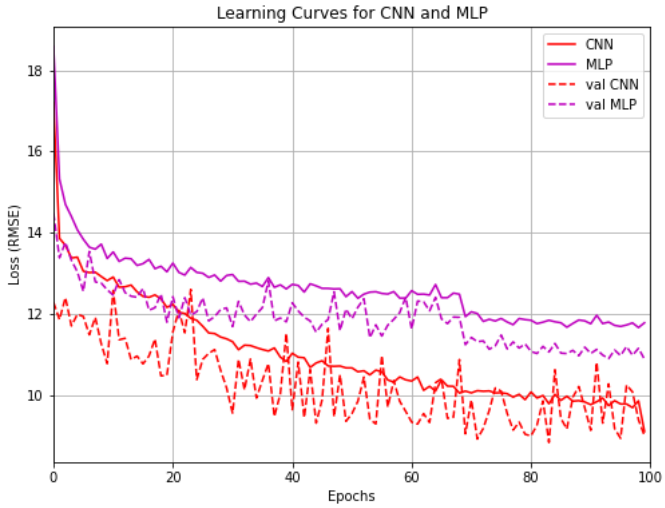

Fig. 3. RMSE loss curve for MLP, and CNN architectures, with dashed for the validation curves.

Table 6. Comparison results between our proposed MLP and CNN models, and literature approaches. The results presented are related to RMSE, Combination score (CO), Computed score (CP), and 2021 PHM Competition.

Model	RMSE	CO Score	CP Score	PHM21 Score
Proposed MLP	11.67	6.79	1.92	
Proposed CNN	**9.11**	**5.14**	**1.17**	
CNN [25]	10.46	6.30	2.13	3.651
CNN [3]	12.5	7.52	2.53	3.327
LSTM-AE [9]	15.57	125.43	236	
AE [9]	14.86	117.43	220	
RR+ [9]	11.32	70.16	129	
MLP+ [9]	13.64	91.32	169	
Winner PHM Comp. [12]	-	-	-	3.006

squares with l2-regularization), and MLP+ (simple MLP architecture with l2-regularization) [9] in Table 6 are the average values of the 11, 14 and 15 results obtained from the paper.

As seen, the **CNN** average results scored 4.86 with RMSE loss of 9.11, which was better than the winner [12], the second [3] and third [25] places of the 2021 PHM Competition [21]. The winner's result [12] of the competition was perceived in terms of score and used a feed-forward network for pre processing the data and a deep CCN for classification. The second place published their work [3] with a deep convolutional neural network architecture with different parameters and configuration than those used in this work. The third place also published their work [25] where they built a method that combines AE and deep CNN. The **MLP** implemented generated worse results than the other models.

Considering the results of the 2021 PHM Competition it is assumed that the competition gave more importance to the weight of the computed score than RMSE for its PHM21 Score. However, unfortunately, we do not have the necessary information to compare with the winner's result.

5 Conclusions

The purpose of this work is to produce a model for RUL prediction in airplane turbofans using the simulated N-CMAPSS data. As it is critical equipment for an aircraft, the improvement of the model results is extremely important.

It was not possible to compare with the winner of the competition, but the results were better than the second [3] and third [25] places in RMSE, Computed Score, and Combination Score.

Furthermore, when this work results (9.11 of RMSE and 1.17 of Computed Score) are compared with [3] which obtained an RMSE of 12.5 and a Computed Score of 2.13 for its CNN configuration, results have improved over 25% in RMSE and 15% in Computed Score.

There is room for improvement in future work by carrying out more architecture tests, in order to obtain better results with lower computational costs. In addition to the possible optimization of the network parameters.

References

1. Chao, M.A., Kulkarni, C., Goebel, K., Fink, O.: Aircraft engine run-to-failure dataset under realflight conditions. A preprint from NASA (Apr 2020)
2. Da Costa, P.R.d.O., Akcay, A., Zhang, Y., Kaymak, U.: Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation. International Journal of Prognostics and Health Management **10**(4) (2019)
3. DeVol, N., Saldana, C., Fu, K.: Inception based deep convolutional neural network for remaining useful life estimation of turbofan engines. In: Annual Conference of the PHM Society. vol. 13 (2021)
4. Ellefsen, A.L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., Zhang, H.: Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering & System Safety, **183**, 240–251 (Mar 2019)
5. Forrester: (2019), https://www.forrester.com/consulting/, acessed: June, 15 2022
6. Frazzetto, D., Nielsen, T., Pedersen, T., Siksnys, L.: Prescriptive analytics: A survey of emerging trends and technologies. VLDB **28**(4), 575–595 (2019)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning pp. 448–456 (2015)
8. Kang, Z., Catal, C., Teikinerdogan, B.: Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks. Sensors **21**(3), 932 (2021)
9. Koutroulis, G., Mutlu, B., Kern, R.: Constructing robust health indicators from complex engineered systems via anticausal learning. Engineering Applications of Artificial Intelligence **113**, 104926 (2022)
10. Li, J., Li, X., He, D.: A directed acyclic graph network combined with cnn and lstm for remaining usefullife prediction. Special section on advances in prognostic and system management **7** (2019)
11. Li, X., Ding, Q., Sun, J.Q.: Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety **172**, 1–11 (2018)
12. Lövberg, A.: Remaining useful life prediction of aircraft engines with variable length input sequences. In: Proceedings of the Annual Conference of the PHM Society (2021)
13. Ma, Q., Zhang, M., Xu, Y., Song, J., Zhang, T.: Remaining useful life estimation for turbofanengine with transformer-based deep architecture. 26th International Conference on Automation and Computing (ICAC). IEEE pp. 1–6 (2021)
14. Malhotra, P., TV, V., Ramakrishnan, A.: Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. 1st ACM SIGKDD Workshop on Machine Learning for Prognosticsand Health Management (2016)
15. Mathew, V., Toby, T., Singh, V., Rao, B.M., Kumar, M.G.: Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning. IEEE International Conference on Circuits and Systems (ICCS) (2017)
16. Molugaram, K., Rao, G.S., Shah, A., Davergave, N.: Statistical techniques for transportation engineering. Butterworth-Heinemann (2017)
17. NASA: Turbofan engine degradation simulation data set-1 (2008), https://ti.arc.nasa.gov/c/6/
18. NASA: Turbofan engine degradation simulation data set-2 (2021), https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/turbofan
19. Ozcan, S., Simsir, F.: A new model based on artificial bee colony algorithm for preventive maintenance with replacement scheduling in continuous production lines. Engineering Science and Technology, an International Journal **22**(6), 1175–1186 (2019)
20. Pasa, G., Medeiros, I., Yoneyama, T.: Operating condition-invariant neural network-based prognostics methods applied on turbofan aircraft engines. Annual conference of the prognostics and health management society 2019 (2019)
21. PHMsociety: 2021 phm conference data challenge results (2021), https://data.phmsociety.org/2021-phm-conference-data-challenge-winners/
22. Salonen, A., Gopalakrishnan, M.: Practices of preventive maintenance planning in discrete manufacturing industry. Journal of Quality in Maintenance Engineering (2020)
23. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? Advances in neural information processing systems **31** (2018)

24. Saxena, A., , Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 international conference on prognostics and health management IEEE pp. 1–9 (Oct 2008)
25. Solís-Martín, D., Galán-Páez, J., Borrego-Díaz, J.: A stacked deep convolutional neural network to predict the remaining useful life of a turbofan engine. arXiv preprint arXiv:2111.12689 (2021)
26. Tian, Y., Ma, T., Khan, M.K., Sheng, V.S., Pan, Z.: Big data and security: Third international conference. ICBDS 2021. Proceedings (Nov 2021)
27. Wen, L., Dong, Y., Gao, L.: A new ensemble residual convolutional neural network for remaining useful life estimation. Mathematical Biosciences and Engineering **16**(2), 862–880 (2019)
28. Wu, J., Hu, K., Cheng, Y., Zhu, H., Shao, X., Wang, Y.: Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. ISA transactions **97**, 241–250 (2020)
29. Yang, H., Zhao, F., Jiang, G., Sun, Z., Mei, X.: A novel deep learning approach for machinery prognostics based on time windows. Applied Sciences **9**(22), 4813 (2019)
30. Zhao, C., Huang, X., Li, Y., Iqbal, M.Y.: A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction. Sensors **20**(24), 7109 (2020)
31. Zhao, Z., Liang, B., Wang, X., Lu, W.: Remaining useful life prediction of aircraft engine based on degradation pattern learning. Reliability Engineering System Safety **164**, 74–83 (2017)
32. Zheng, S., Ristovski, K., Farahat, A., Gupta, C.: Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management (ICPHM). pp. 88–95. IEEE (2017)
33. Zutin, G.C., Barbosa, G.F., de Barros, P.C., Tiburtino, E.B., Kawano, F.L.F., Shiki, S.B.: Readiness levels of industry 4.0 technologies applied to aircraft manufacturing—a review, challenges and trends. The International Journal of Advanced Manufacturing Technology pp. 1–17 (2022)