Prognostic techniques for aeroengine health assessment and Remaining Useful Life estimation

A. Caricato[1, a)], A. Ficarella[1, b)] and L. Spada Chiodo[1,c)]

[1]*Department of Engineering for Innovation, University of Salento, Lecce, 73100, Italy*
Phone: +39 0832 297320; Fax: +39 0832 297777

[a)] antonio.caricato1@unisalento.ii
[b)] antonio.ficarella@unisalento.it
[c)] ludovica.spadachiodo@studenti.unisalento.it

Abstract. Predictive maintenance is the latest frontier in the management and maintenance of many industrial assets, including aeroengines. Made possible by last decades advances in monitoring equipment and machine learning algorithms, it permits individual-based maintenance schedules, on the basis of performance monitoring and estimates resulting from the application of diagnostic and prognostic techniques, whether on ground or real time. Predictive maintenance results in operational cost reduction and asset usage optimization, if compared with traditional maintenance strategies, which instead may suffer from unanticipated failure or unnecessary maintenance and therefore higher operational costs. In the study, Remaining Useful Life (RUL) estimates will be carried out for different turbofan engines, based on historical individual and fleet data made available by the Prognostics Center of Excellence at NASA.
The design of Prognostics and Health Management (PHM) algorithms requires at first an analysis of available data to identify which of them is effectively related to equipment degradation and hence could be useful in determining future system evolution and predicting failure.
In particular, RUL prediction of test engines suffering from high pressure compressor fault with exponential degradation trend has been carried out with both regression and Artificial Neural Networks (ANNs). In turn, different regression models and neural network architectures have been compared, namely tree regression with different levels of tree depth, Gaussian Process Regression (GPR) with different kernel functions and Multilayer Perceptron (MLP) with one to three hidden layers and varying number of nodes. The objective is to demonstrate the capability of such machine learning algorithms to predict engine failure and thus their importance in supporting predictive maintenance planning, and to evaluate the quality of results in relation to the algorithm structure.
Results show comparable performance in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of predicted with respect to actual RUL, in particular predictions obtained through recourse to multilayer perceptron reveal to be the most accurate, with a RMSE of 17.38 and a MAE of 12.50.

Nomenclature

ANN (NN)	*Artificial Neural Network (Neural Network)*	MLP	*Multilayer Perceptron*
GPR	*Gaussian Process Regression*	MSE	*Mean Square Error*
HPC (HPT)	*High Pressure Compressor (Turbine)*	PHM	*Prognostics and Health Management*
LPC (LPT)	*Low Pressure Compressor (Turbine)*	RMSE	*Root Mean Square Error*
MAE	*Mean Absolute Error*	RUL	*Remaining Useful Life*

INTRODUCTION

Reliability, availability, safety and maintenance cost effectiveness have been an important concern in many industries.

For this reason, predictive maintenance is one of the key figures within the field of Industry 4.0, since it guarantees high equipment availability and reduced downtime, allowing for an optimal exploitation of the item itself and its maintenance process and thus reduced operational costs.

Predictive maintenance relies on Prognostics and Health Management (PHM) systems, which provide overall health state of machines or complex systems and assists in making correct decisions on machine maintenance.

The main duties of PHM technology are monitoring of key features, assessing engine health, identifying and isolating potential faults and establishing degradation trends to be used to predict the engine remaining useful life (RUL).

Starting from either a fault detection or a degradation trend, prognostic models can be developed to predict future evolution scenarios of the engine state until a predefined threshold of acceptability.

This can be intended as the point at which the impact of the potential fault on the system performance is no longer tolerable or as the remaining useful life of life limited parts.

In fact, health monitoring plays the preliminary role of monitoring system performance, identifying eventual faults or abnormal behavior and then enabling a future perspective to be delineated by means of prognostic algorithms.

This in turn permits a different maintenance approach, which provides for a maintenance action to be taken when necessary but however before failure occurrence: in this way, the system can be exploited at a maximum and the maintenance operation can be planned in optimal manner.

In the present paper, fleet data of several turbofan units available at NASA PCoE Data Repository are analyzed and employed to implement different predictive models by means of three distinct machine learning algorithms, namely Tree Regression, Gaussian Process Regression and Neural Networks, whose scope is the prediction of Remaining Useful Life of test engines.

Algorithm performance in terms of RMSE and MAE of predicted with respect to actual values of RUL are given on varying algorithm characteristic parameters.

The most common practice in engine prognostics algorithms is that of fusing data collected from the engines into a single health index and use it as the target of models to be trained. Following this approach, Kang et al. [1] have already demonstrated the high potentiality of MLP in predicting failure.

A similarity-based approach has been employed in [2] and [3]; several neural network solutions have been used for prognostic purposes, such as Recurrent Neural Network in [4], Deep Convolutional Neural Network in [5] and Long Short-Term Memory in [6]. Though, such algorithms require a much higher computational cost if compared with MLP. Thus, this study aims to develop a MLP for RUL prediction trained with separate engine data and compare algorithm performance in relation to its architectural parameters.

As for regression kind prognostics, Taha et al. [7] compare different regression methodologies, ranging from linear to random forest, while Tree Regression is used in [8]. In the present study, the potentiality of Tree Regression and result optimization with tree development are analyzed.

Finally, Gaussian Process Regression appears to be a novelty inside engine prognostics, since its previous applications mainly concern different equipment ([9], [10]).

DATASET AND ANALYSIS

The present work is focused upon data processing and training of a model which would be useful for RUL prediction of several individuals from a fleet of similar turbofan engines, on the basis of data made available at NASA PCoE Repository.

Data is extracted from datasets FD001 and FD002, which in turn contain a training dataset plus a test dataset each, whose actual values of RUL are given, too.

The former (training) is made of degradation trajectories of different units until failure occurrence while the latter consists of degradation trajectories of other units interrupted at an unknown time prior failure occurrence.

For FD001, both training and testing datasets contain 100 degradation trajectories, while for FD002 260 training engines and 259 testing engines are given.

Degradation trajectories are obtained through simulations of a turbofan engine model in C-MAPSS (Commercial Modular Aero Propulsion System Simulation).

When building the model to be simulated, the user inserts a degradation law that can be applied to each of the rotating components (fan, LPC, HPC, HPT, LPT), to simulate performance decay during engine life.

Each of the modules can be characterized by 3 types of degradation: efficiency loss, flow capacity loss and pressure ratio loss.

In particular, data available in dataset FD001 and FD002 have been obtained from simulations run with different exponential degradation trajectories affecting only HPC (this form was chosen since it reflects common degradation trends experienced in practice), with randomly chosen coefficients a and b ($0.001 \leq a \leq 0.003$ and $1.4 \leq b \leq 1.6$)[1], being the exponential degradation *d* (non-dimensional) expressed as a function of time *t* in the form

$$d = \exp(a \cdot t^b) \quad (1)$$

Moreover, an initial deterioration not higher than 0.01 is added, to account for initial wear ascribable to manufacturing inefficiencies that is commonly observed in real systems.

The output of the simulations carried out with the above-described deteriorated models, is given as a time-series of the parameters listed in Table 1, corrupted with a certain amount of random measurement noise [11].

TABLE 1. C-MAPSS simulation outputs.

Sensor number	Sensed parameter	Description
Sensor 1	T2	Total temperature at fan inlet [°R]
Sensor 2	T24	Total temperature at LPC outlet [°R]
Sensor 3	T30	Total temperature at HPC outlet [°R]
Sensor 4	T50	Total temperature at LPT outlet [°R]
Sensor 5	P2	Total pressure at fan inlet [psia]
Sensor 6	P15	Total pressure in bypass duct [psia]
Sensor 7	P30	Total pressure at HPC outlet [psia]
Sensor 8	Nf	Fan speed [rpm]
Sensor 9	Nc	Core speed [rpm]
Sensor 10	EPR	P50/P2
Sensor 11	Ps30	Static pressure at HPC outlet
Sensor 12	Phi	Fuel flow/Ps30 [pps/psi]
Sensor 13	NRf	Corrected fan speed [rpm]
Sensor 14	NRc	Corrected core speed [rpm]
Sensor 15	BPR	Bypass Ratio
Sensor 16	FARb	Burner fuel air ratio
Sensor 17	htBleed	Bleed enthalpy
Sensor 18	Nf_dmd	Demanded fan speed [rpm]
Sensor 19	PCNfR_dmd	Corr. demanded fan speed [rpm]
Sensor 20	W31	HPT coolant bleed [lbm/s]
Sensor 21	W32	LPT coolant bleed [lbm/s]

[1] For the details of C-MAPSS engine simulations refer to [11], since only simulation outputs were made available for the purpose of data-driven algorithm implementation for RUL predictions.

Being very large and noisy datasets, they should be properly pre-processed before being employed to construct a certain predictive model with higher accuracy.

In fact, not all available data are necessarily useful for prognostic purposes.

Hence, at first parameters assuming constant or nearly constant values throughout the engines' lives were discarded from training datasets, since they carry no information about performance decay.

They have been identified by imposing a lower threshold to 0.005 to parameters' standard deviations.

After this step, parameters (T2 - P2 - P15 – EPR – FAR – Nf_dmd – PCNfR_dmd), corresponding to sensors s1, s5, s6, s10, s16, s18 and s19 were deleted from the original dataset FD001 and (T2 - P2 - EPR - FAR - Nf_dmd - PCNfR_dmd), and so sensors s1, s5, s10, s16, s18 and s19 are discarded from FD002.

Although engine degradations of all members of FD002 is ascribable to HPC failure as for FD001, the main difference between them lies in the fact that in this case engines do not operate at a unique combination of settings: 6 different operating conditions are present, which makes it necessary to preliminarily individuate operating condition clusters and associate each instance to the pertaining cluster, so that all measurements can be normalized with respect to the corresponding cluster mean and standard deviation values: in this manner it becomes possible to compare sensed parameters independently from operational settings.

All significant parameters to be retained for further analysis show a visible degradation trajectory towards end of life, with common trend for all sensors but sensor 9 and sensor 14 (Fig.1).

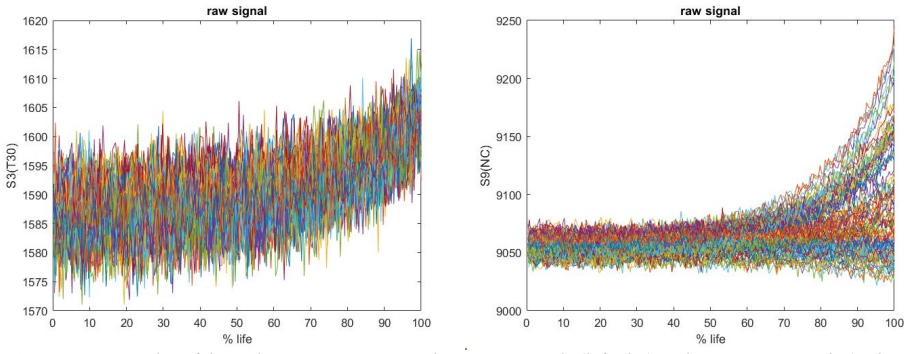

FIGURE 1. *Examples of degradation trajectories with common trends (left plot) and uncommon trends (right plot).*

Proposed Methodology

After parameter selection, there is the need to filter remaining parameters to reduce the impact of noise on subsequently implemented algorithms. A moving mean filter has been employed for this scope.

Filtered sensor measurements retained after preliminary analysis will be used as inputs to train a model capable of predicting desired responses for a set of different data.

Of course, the response of interest is remaining useful life (i.e. output of the models), at first simply defined as progressive time to failure, or

$$RUL(t)=|t-t_{max}| \quad (2)$$

Though, it can be noticed that performance decay is not appreciable in an engine early stage of life, becoming much more evident and steep after some degree of usage is reached.

This would result in very low accuracy when predicting RUL for test engines which have run only few cycles, since their degradation path is not noticeable yet.

Such issue can be dealt with by modeling a piecewise linear RUL function, which shows a constant output at the beginning of the equipment employment followed by a linear segment preceding failure, reflecting real behavior.

This means that, once individuated the point at which the RUL knee appears, a constant value is attributed to RUL for each cycle before the knee, as supported from many literature sources [4-12-13].

In the present work, it has been assumed that degradation becomes manifest in the last 125 cycles before failure, resulting in a RUL function shaped as in the picture below (Fig.2).

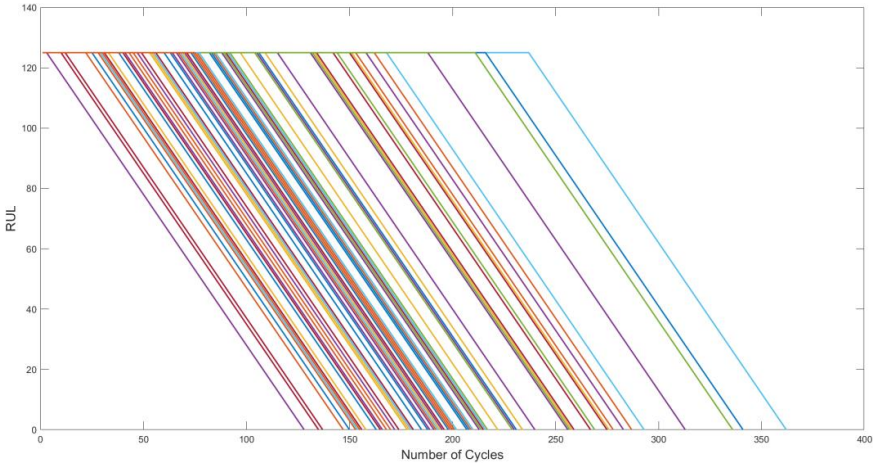

FIGURE 2. *Piecewise linear RUL for each engine in training dataset.*

The first approach consists in training several regression models in Matlab Regression Learner, with a 25% holdout validation method. That means that only 75% of training data fed into the algorithm will be used to train the model, while remaining 25% serves to test and thus validate it.

The most suitable regression models for the problem at hand, in terms of performance in predicting validation sets, appear to be either Tree Regression or Gaussian Process Regression.

Tree based models split the data multiple times according to certain cutoff values in the features. Through splitting, different subsets of the dataset are created, with each instance belonging to one subset.

The final subsets are called terminal or leaf nodes and the intermediate subsets are called internal nodes or split nodes [14].

To predict the outcome in each leaf node, the average outcome of the training data in this node is used:

$$\hat{y}= \hat{f}(x)= \sum_{m=1}^{M} c_m I\{x \in R_m\} \quad 3)$$

with M subsets, c_m constant for each subset and $I\{x \in R_m\}$ is the identity function that returns 1 if x is in the subset R_m and 0 otherwise.

Each instance falls into exactly one leaf node (that is R_m).

If an instance falls into a leaf node R_l, the predicted outcome is c_l, where c_l is the average of all training instances in leaf node R_l.

Splitting points are chosen so as to minimize squared error of predictions in the two subsets identified with the split.

A medium tree with minimum leaf size of 12, a coarse tree with leaf size of 36 and 50 and an optimizable tree have been trained. In fact, although a finer tree could behave better when trained, it should be considered that trees with too small leaf size may incur in overfitting and perform worse when applied to a test dataset.

Then, a Gaussian Process Regression with both exponential and squared exponential kernel function is modeled.

Gaussian Process Regression is a non-parametric regression which incorporates Bayesian approach in the creation of the model.

Given a set of data $X = \{x_i\}$ and observations $Y = \{y_i\}$, $i = 1,...,n$, such that

$$y_i = f(x_i) + \varepsilon \quad 4)$$

and ε additive gaussian noise with 0 mean and σ^2 variance, the distribution of latent functions $f(x_i)$ is assumed to be a gaussian process, that means that $P(f|x_1,\ldots,x_n) = N(0, K)$.

K is the covariance matrix obtained from the covariance function (kernel) k.

A common shape for kernel function is squared exponential, expressed by the following equation:

$$k(x_i,x_j|\theta) = \sigma_f^2 \exp[-\frac{1}{2}\frac{(x_i-x_j)^T(x_i-x_j)}{\sigma_l^2}] \quad 5)$$

where σ_f^2 is the prior variance and σ_l a length scale parameter (θ defines the vector of kernel parameters).

Another possible assumption for kernel function is that of exponential kernel, defined as

$$k(x_i,x_j|\theta) = \sigma_f^2 \exp[-\frac{r}{\sigma_l^2}] \quad 6)$$

where r is the Euclidean distance between x_i and x_j.

The model is trained when its hyperparameters are such that they maximize the likelihood of training dataset [15].

Once assumed a mean function and the kernel function of the prior distribution, acquiring new data X^* allows to update the posterior distribution of possible fitting functions in a Bayesian perspective, that is:

$$\begin{bmatrix} Y \\ f^* \end{bmatrix} \sim N(0, \begin{bmatrix} (K+\sigma^2 I) & k \\ k^T & k(X,X^*) \end{bmatrix}$$

which represents the joint distribution over observed Y and f* and consequently allows to calculate the conditional probability $P(f^*|f,X,X^*)$

The second method employed to predict RUL of test engines resorts to ANNs.

Different network architectures were built and trained for dataset FD001 and FD002: in particular, 4 multilayer perceptrons with one hidden layer and varying number of neurons and a cascade-forward network were applied to dataset FD001, while FD002 was tested with four different MLP with one hidden layer, three MLP with 2 hidden layers and a MLP with 3 hidden layers.

Data fed as input is randomly split up into a training portion, a validation and a testing portion. Default values for training, validation and testing fractions are 0.7, 0.15, 0.15.

All previously mentioned networks gave a MSE lower than 300 and a R-squared higher than 0.91 with validation subset.

The recourse to higher complexity networks required an increment in training time, not accompanied by an equivalent improvement in performance: in fact, this consideration is supported by the common assumption that MLP with single hidden layer is a potential universal approximator for continuous mapping functions from one finite space to another [16].

The net is trained by backpropagation with Levenberg-Marquardt algorithm, which is an iterative method used to solve nonlinear least squares problems: in order to minimize a determined loss function (generally MSE) it updates the network parameters according to the following equation:

$$w_{i+1} = w_i - (J^T \cdot J + \mu I) - 1 \cdot (2J^T \cdot e_i) \quad 7)$$

where the Hessian has been approximated with J^TJ, e_i is the error associated to the i-th instance and μ is the damping factor, whose value determines whether the algorithm approximates more closely a steepest descent or a Gauss method: μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function.

In this way, the performance function is always reduced at each iteration of the algorithm [17].

Training arrests whether the minimum magnitude of gradient descent is reached or maximum number of epochs (in the sense of iterations) or alternatively maximum μ.

Figures 3A, 3B and 3C depict a flowchart for the three reported algorithms.

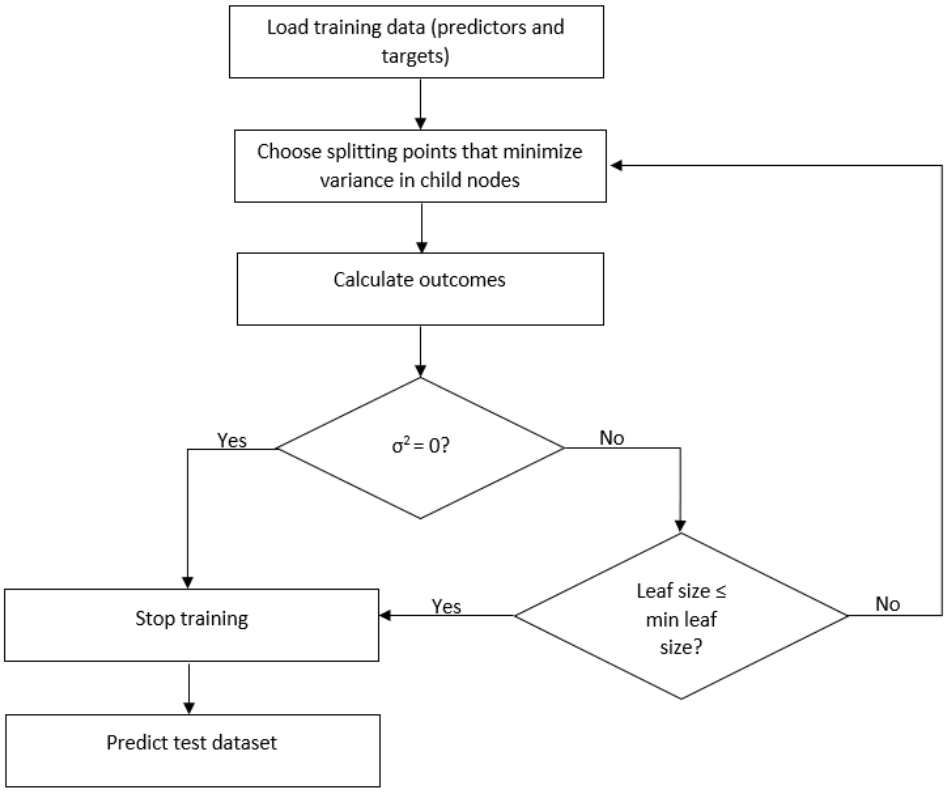

FIGURE 3A. *Tree Regression flowchart.*

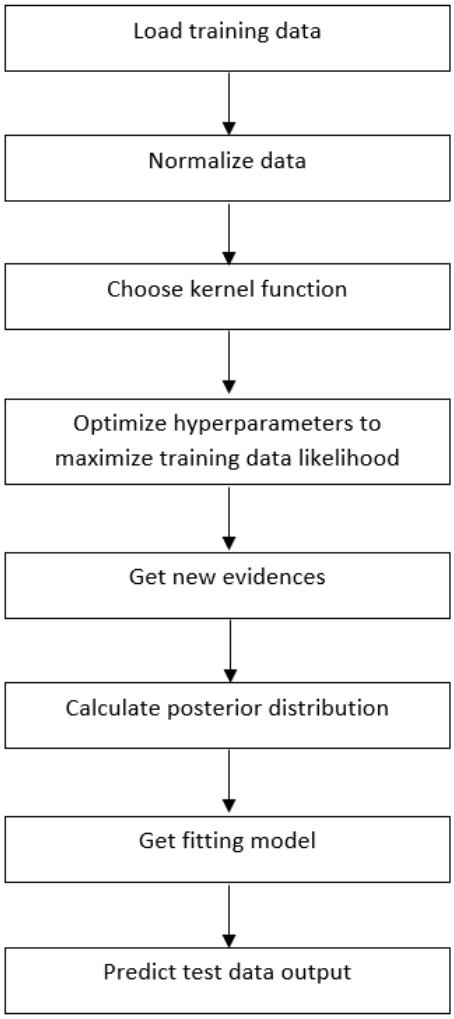

FIGURE 3B. *Gaussian Process Regression flowchart.*

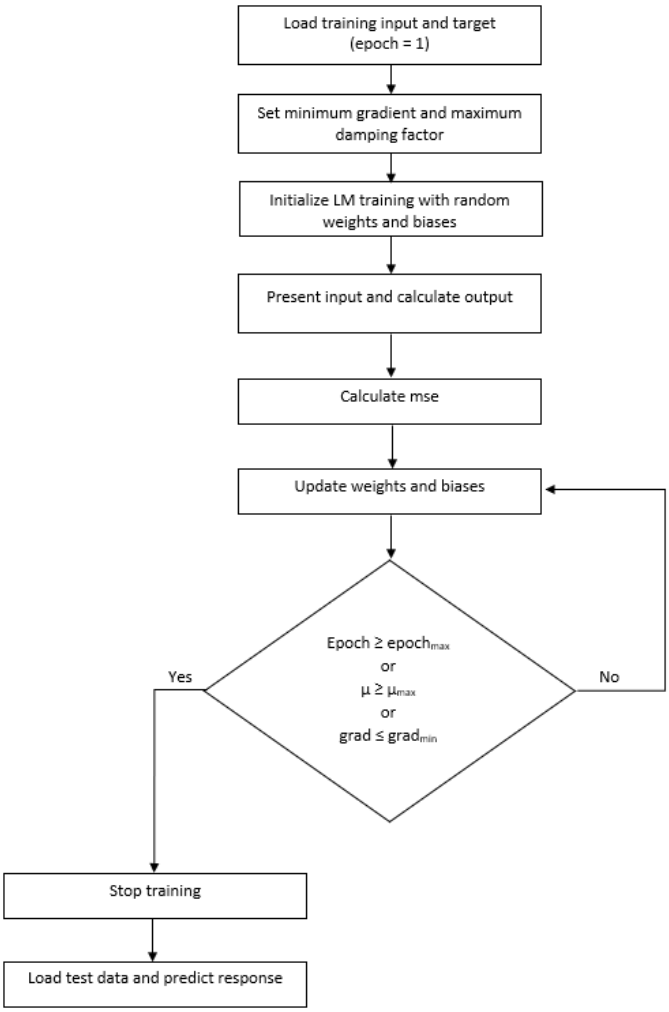

FIGURE 3C. *Neural Network flowchart.*

The trained models are then applied to test datasets to predict Remaining Useful Life of test units. Since actual RUL of test instances is known, it will be used to evaluate the performance of the tested algorithms, in terms of prediction Root Mean Square Error and Mean Absolute Error.

RESULTS AND DISCUSSION

The following histograms display the performance of all trained predictive models when applied for test dataset predictions, showing the distribution of prediction errors on all instances and global algorithm performance in terms of RMSE and MAE.

When predicting RUL of test engines by means of a Tree Regression algorithm, it can be noticed (Figures 4a, 5a) that a slight performance improvement is attained when leaf size is increased, since the model becomes more capable of generalizing [18].

Gaussian Process Regression with exponential and squared exponential kernel predicted test outputs with similar accuracy as the coarse tree, with a RMSE ~ 20 and MAE ~ 14.6 for dataset FD001 and RMSE ~ 17.6 and MAE ~ 13 for FD002 tests (Figures 4b, 5b).

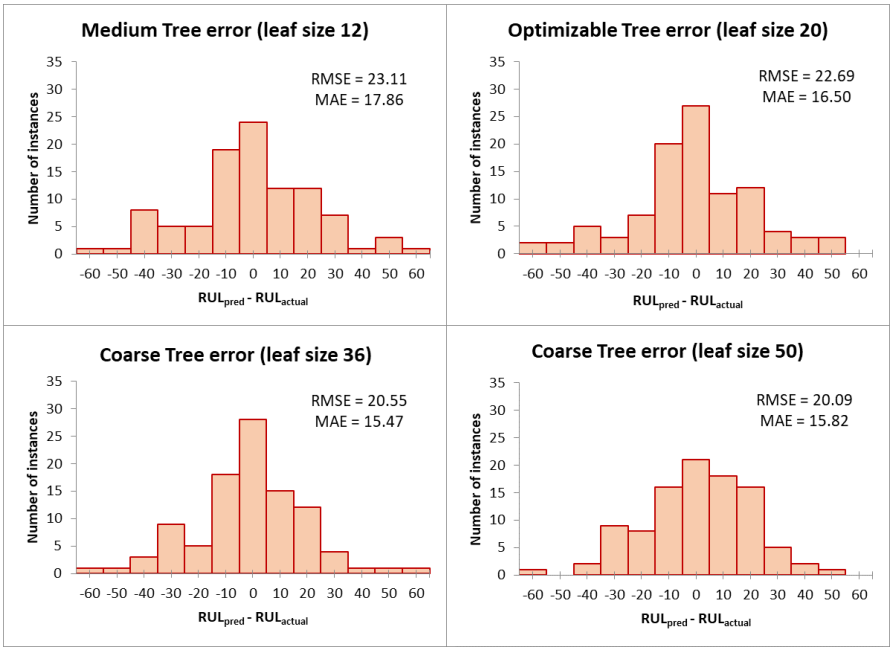

FIGURE 4A. *RMSE, MAE and error distribution for Tree Regression on test dataset FD001.*

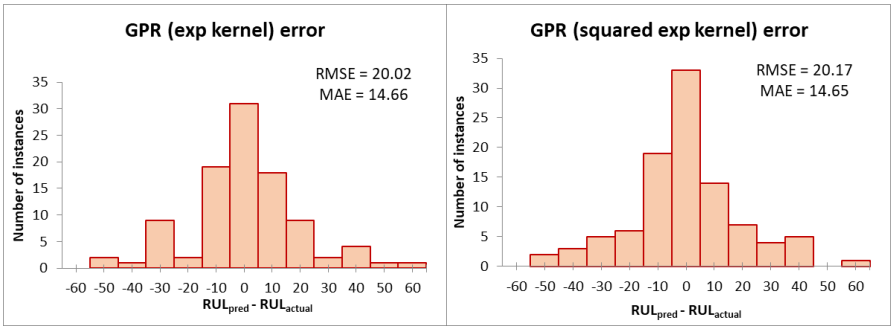

FIGURE 4B. *RMSE, MAE and error distribution for GPR on test dataset FD001.*

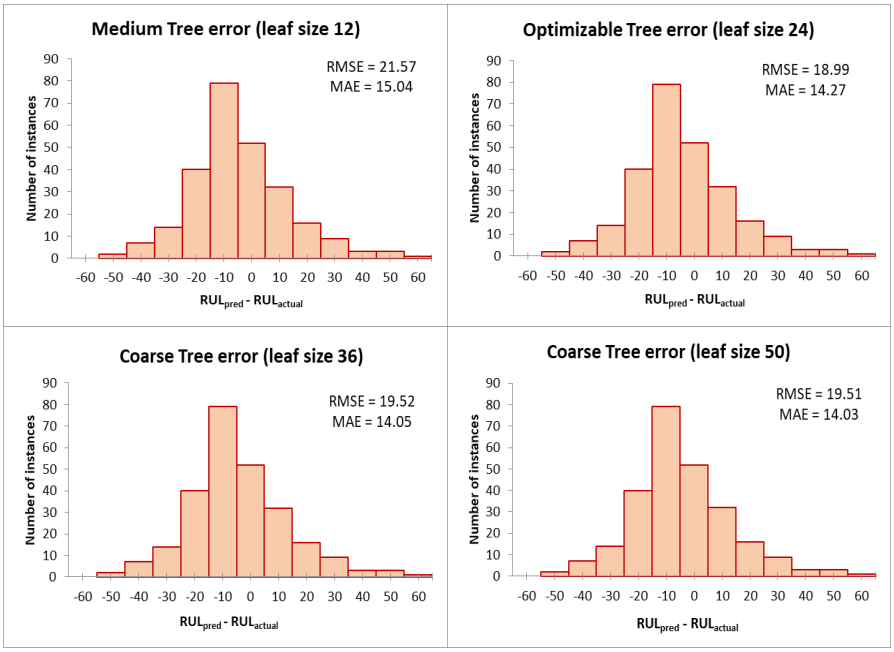

FIGURE 5A. *RMSE, MAE and error distribution for Tree Regression on test dataset FD002.*

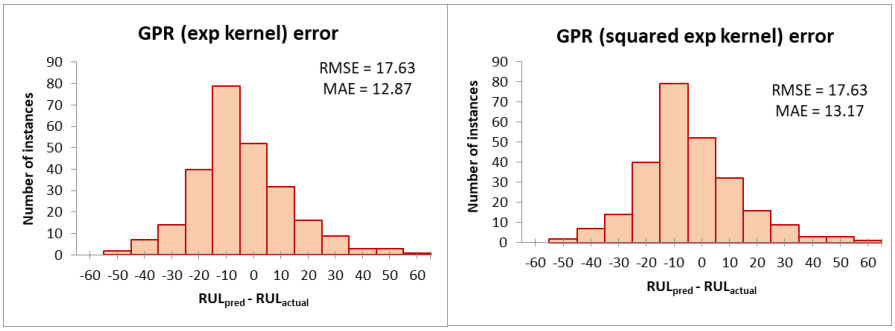

FIGURE 5B. *RMSE, MAE and error distribution for GPR on test dataset FD002.*

The following plots (Figures 6a, 6b) show the actual RUL compared with RUL predictions obtained with each regression model described above for each unit in both datasets.

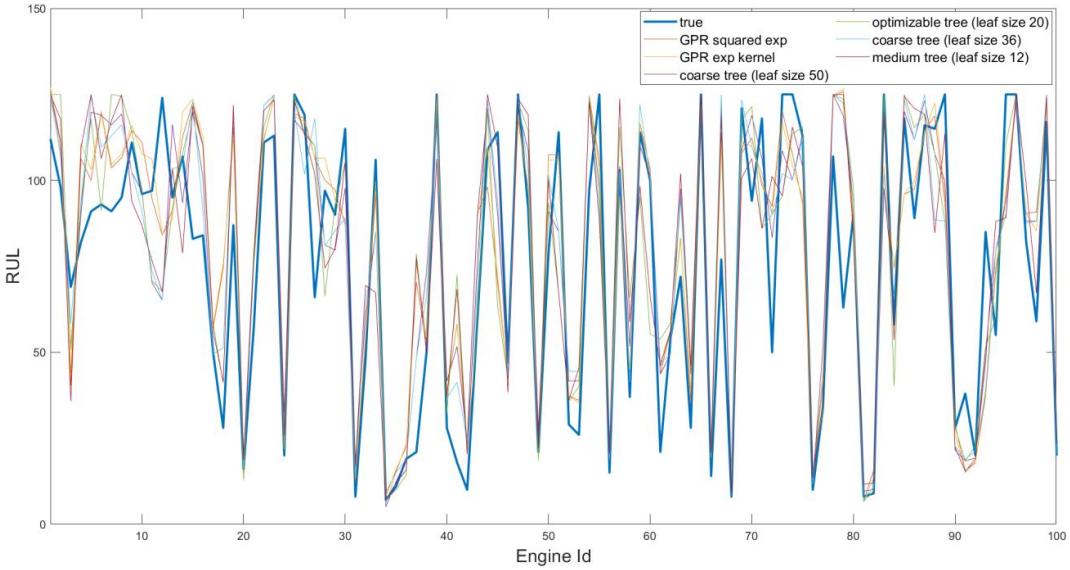

FIGURE 6A. *RUL predictions for test dataset FD001 with Tree Regression and GPR.*

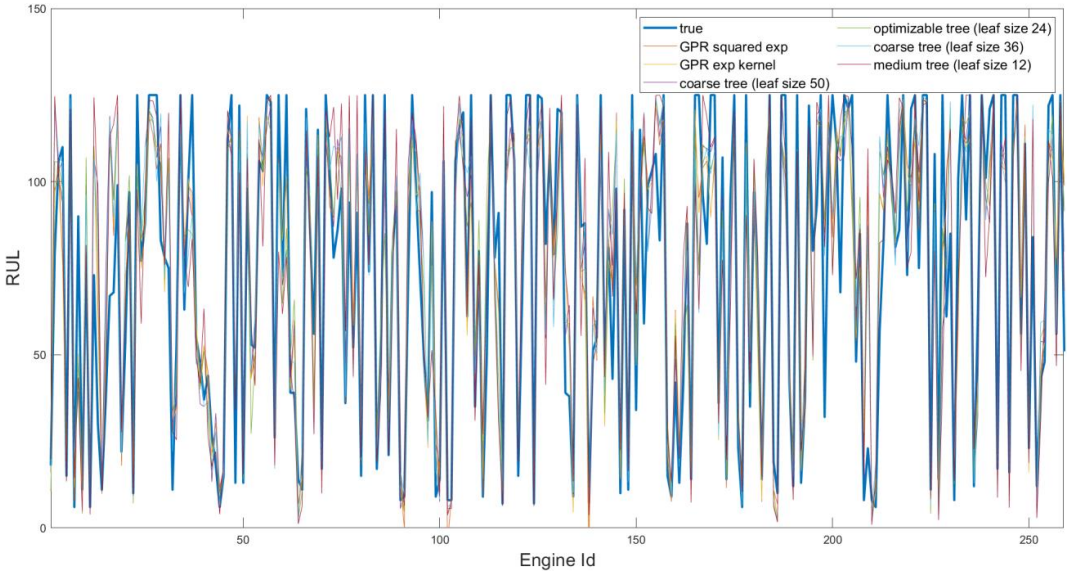

FIGURE 6B. *RUL predictions for test dataset FD002 with Tree Regression and GPR.*

The performances of ANNs for FD001 and FD002 test units predictions are shown in the following pictures (Figures 7a, 7b):

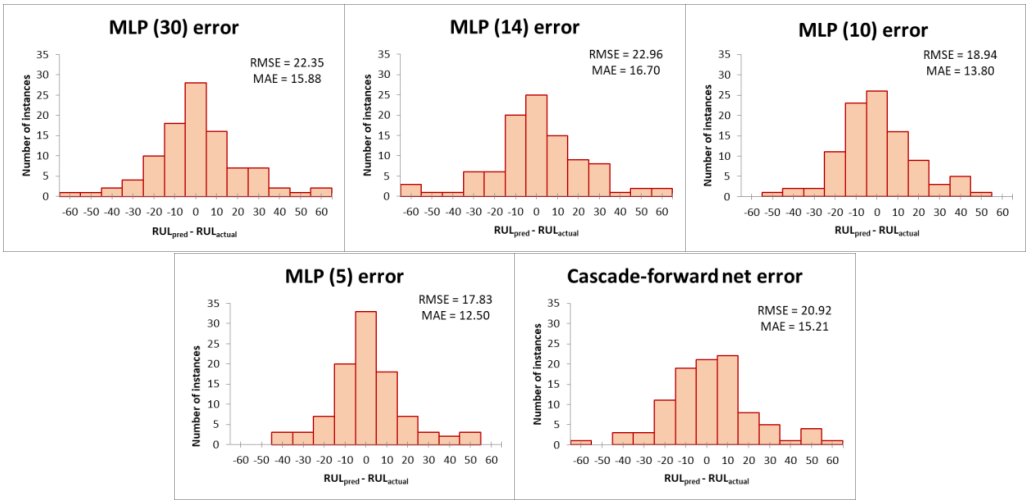

FIGURE 7A. *RMSE, MAE and error distribution for different NN predictions of test dataset FD001.*

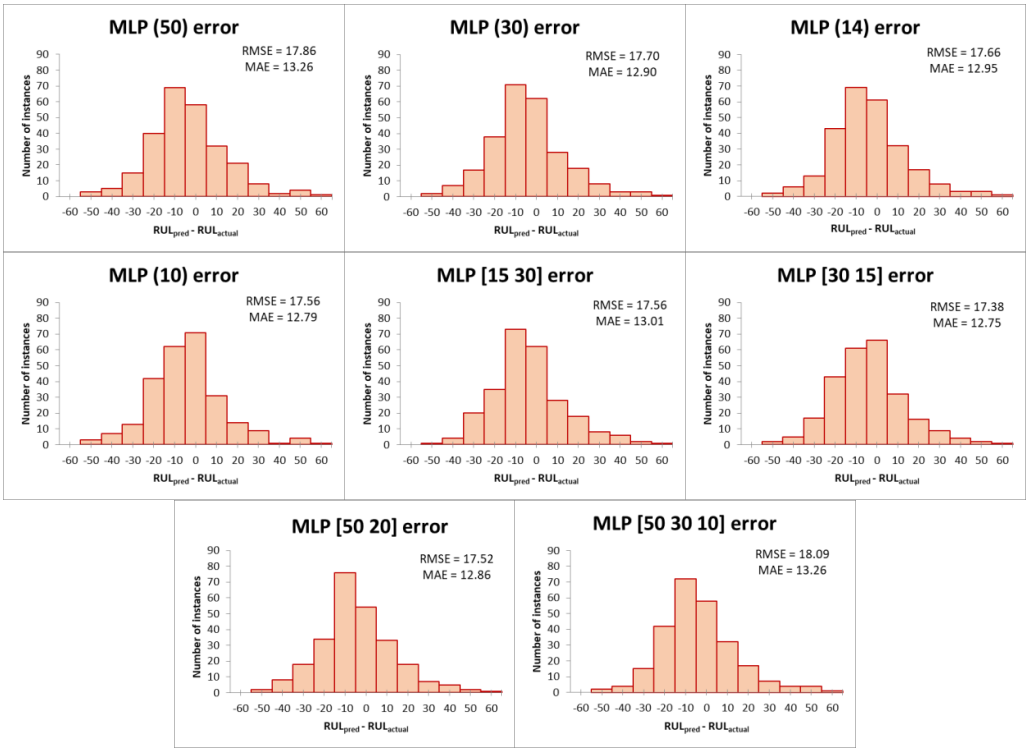

FIGURE 7B. *RMSE, MAE and error distribution for different NN predictions of test dataset FD002.*

RUL predictions are shown in Figure 8a (FD001 dataset) and Figure 8b (FD002 dataset).

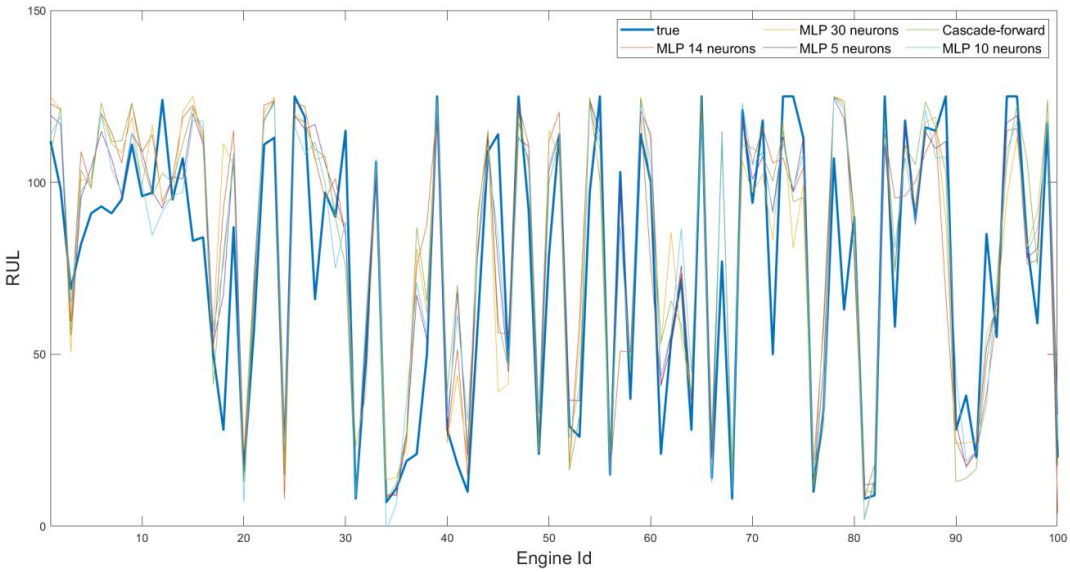

FIGURE 8A. *RUL predictions for test dataset FD001 with ANNs.*

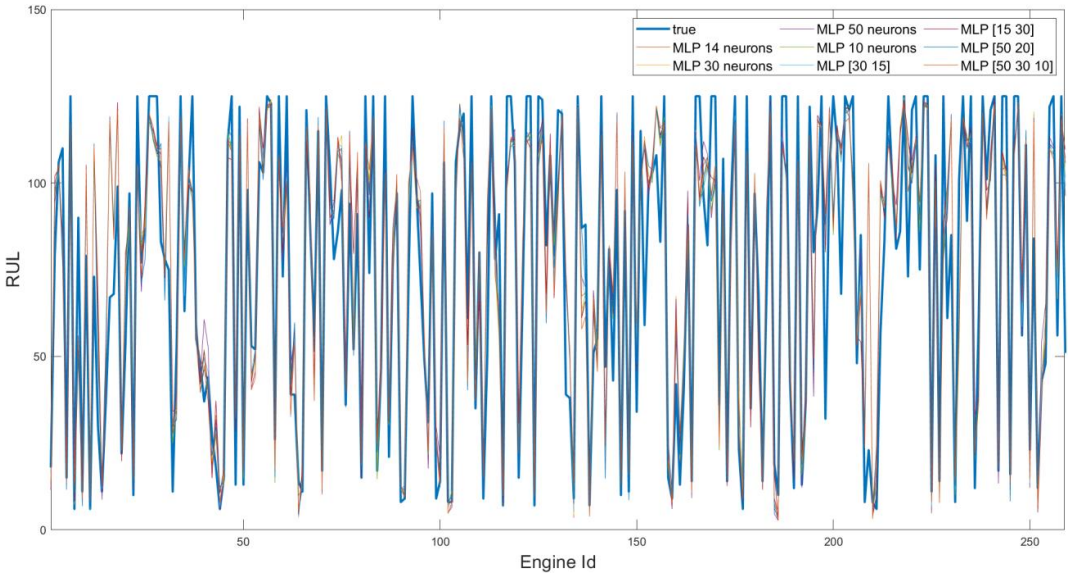

FIGURE 8B. *RUL predictions for test dataset FD002 with ANNs.*

The level of accuracy achieved with all proposed methods is in line with those reported in current literature studies.

Among all tested algorithms, the best predictive performance was achieved with the lightest MLP for test engines of dataset FD001: so the network with best prediction accuracy is a multilayer perceptron with one hidden layer and five nodes.

In general, the error increase with increasing depth of tree regressions and number of hidden neurons in the networks, as can be evinced from plots in Figs. 4-7 and Tables 2-3, confirms the reduced generalization capability of too fine trees and NNs with too many hidden nodes, as a consequence of probable overfitting when trained.

TABLE 2. Algorithm performances on FD001 test dataset.

	RMSE	**MAE**
MEDIUM TREE REGRESSION	23.11	17.86
COARSE TREE REGRESSION (LEAF SIZE 36)	20.55	15.47
COARSE TREE REGRESSION (LEAF SIZE 50)	20.09	15.82
OPTIMIZABLE TREE (LEAF SIZE 20)	22.69	16.50
GPR EXP KERNEL	20.02	14.66
GPR SQUARED EXP KERNEL	20.17	14.65
MLP 14 NEURONS	22.96	16.70
MLP 30 NEURONS	22.35	15.86
MLP 5 NEURONS	17.83	12.50
MLP 10 NEURONS	18.94	13.80
CASCADE-FORWARD NET	20.92	15.31

TABLE 3. Algorithm performances on FD002 test dataset.

	RMSE	**MAE**
MEDIUM TREE REGRESSION	21.57	15.04
COARSE TREE REGRESSION (LEAF SIZE 36)	19.52	14.05
COARSE TREE REGRESSION (LEAF SIZE 50)	19.51	14.03
OPTIMIZABLE TREE (LEAF SIZE 24)	18.99	14.27
GPR EXP KERNEL	17.63	12.87
GPR SQUARED EXP KERNEL	17.63	13.17
MLP14 NEURONS	17.66	12.95
MLP30 NEURONS	17.70	12.90
MLP50 NEURONS	17.86	13.26
MLP10 NEURONS	17.56	12.79
MLP [30 15]	17.38	12.75
MLP [15 30]	17.56	13.00
MLP [50 20]	17.52	12.86
MLP [50 30 10]	18.09	13.26

CONCLUSIONS

Inside the framework of Industry 4.0, implementation of accurate prognostic algorithms for failure prediction is a key factor towards the transition from a time-based maintenance policy to a condition-based maintenance, with annexed benefits in terms of reduced maintenance costs, reduced equipment failure and unscheduled maintenance occurrences, optimization of equipment life cycle exploitation.

In such contest, the development and refinement of diagnostic and prognostic tools is fundamental to perform the task of condition monitoring.

When developing PHM algorithms, available data should be analyzed at first to identify which of them is meaningful as performance indicator, being effectively related to equipment degradation and hence could be useful in forecasting system evolution and predicting failure.

This work has investigated the development of data-driven algorithms for assessment of RUL estimates of turbofan engines, on the basis of fleet data available at NASA PCoE (NASA dataset FD001 and FD002), obtained from C-MAPSS simulation: RUL prediction of test engines suffering from HPC fault with exponential degradation trend has been carried out with recourse to regression models and ANNs.

Different regression models and NN architectures have been implemented and applied to test datasets, namely tree regression with varying tree growth, GPR with different kernel functions and MLP with one to three hidden layers and varying number of nodes.

Generally, they have exhibited similar performance in terms of MAE and RMSE of RUL predictions with respect to actual known values, with highest accuracy achieved with recourse to MLP (lowest RMSE 17.38 and lowest MAE 12.50).

However, prediction performances could be further refined, especially for those test cases whose RUL has been predicted with low accuracy: in these cases, an in-depth inspection of data should be taken at first, to try to locate the source of error and consequently reduce its impact with proper pre-processing.

Moreover, further development could concern the optimization of algotithm performance itself on the basis of an optimal combination of hyperparameters, which should be investigated.

REFERENCES

1. Kang, Z., Catal, C. and Tekinerdogan, B. “Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks.”, Sensors, vol. 21, 932, 2011.
2. Wang, T. “Trajectory Similarity Based Prediction for Remaining Useful Life Estimation.”, 2010.
3. Wang, T., Yu, J., & Siegel, D., and Lee, J. “A similarity-based prognostics approach for Remaining Useful Life estimation of engineered systems.”, 1 - 6. 10.1109/PHM.2008.4711421, 2008.
4. Heimes, F. “Recurrent neural networks for remaining useful life estimation.”, 1-6. 10.1109/PHM.2008.4711422, 2008.
5. Li, X., Ding, Q., and Sun, J. Q. “Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks.”, Reliability Engineering & System Safety. 172. 10.1016/j.ress.2017.11.021, 2017.
6. Yuan, M., Wu, Y. and Lin, L. “Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network.”, 2016 IEEE International Conference on Aircraft Utility Systems (AUS), pp. 135-140, 2016.
7. Taha, H., Sakr, A., and Yacout, S. “Aircraft Engine Remaining Useful Life Prediction Framework for Industry 4.0”, 2019.
8. Bakir, A., Zaman, M., Hassan, A., and Hamid, M. “Prediction of remaining useful life for mech equipment using regression.”, Journal of Physics: Conference Series. 1150. 012012. 10.1088/1742-6596/1150/1/012012, 2019.
9. Aye, S.A. and Heyns, S. “An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission.”, Mechanical Systems and Signal Processing. 84. pp. 485-498. 10.1016/j.ymssp.2016.07.039, 2017.
10. Liu, J. and Chen, Z. "Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Health Indicator and Gaussian Process Regression Model", IEEE Access, vol. 7, pp. 39474-39484, 2019.
11. Saxena, A., Goebel, K., Simon, D., and Eklund, N. “Damage propagation modeling for aircraft engine run-to-failure simulation.”, 2008 International Conference on Prognostics and Health Management, pp. 1-9, 2008.
12. Zhao, C., Huang, X., Li, Y., Yousaf Iqbal, M. “A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction”, Sensors, vol. 20, 7109, 2020.
13. Peng, C., Chen, Y., Chen, Q., Tang, Z., Li, L., Gui, W., (2021). “A Remaining Useful Life Prognosis of Turbofan Engine Using Temporal and Spatial Feature Fusion”, Sensors, vol. 21, 418, 2021.
14. Molnar, C. “Interpretable machine learning. A Guide for Making Black Box Models Explainable”, https://christophm.github.io/interpretable-ml-book/, p.49, 2018.

15. Wang, J. “An Intuitive Tutorial to Gaussian Processes Regression”, 2020.
16. https://www.heatonresearch.com/2017/06/01/hidden-layers.html
17. https://www.mathworks.com/help/deeplearning/ref/trainlm.html
18. Sambasivan, R. and Das, S. "Big Data Regression Using Tree Based Segmentation," 2017 14th IEEE India Council International Conference (INDICON), pp. 1-6, 2017.