

Enhancing Predictive Maintenance: A Machine Learning Approach to Turbine RUL Prediction

Joniel Bastos Barreto

PPG Electronic and Computer Eng.
Instituto Tecnológico de Aeronáutica

São José dos Campos, São Paulo, Brazil
jonielbarreto@ita.br

Acélio Luna Mesquita

PPG Electronic and Computer Eng.
Instituto Tecnológico de Aeronáutica

São José dos Campos, São Paulo, Brazil
acelioluna@ita.br

Rafael Silva Godoy

PPG Operations Research Eng.
Instituto Tecnológico de Aeronáutica

São José dos Campos, São Paulo, Brazil
godoyrsg@ita.br

Abstract—Accurately predicting the Remaining Useful Life (RUL) of critical components is essential for effective predictive maintenance, as it allows maintenance to be scheduled based on the actual condition of the equipment rather than arbitrary timelines, thereby preventing failures and optimizing operational efficiency. This paper presents a machine learning-based system for predicting the RUL of Turbofan jet engines, aiming to improve predictive maintenance strategies by using sensor telemetry data to identify when engines will require attention. Furthermore, exponential degradation, LSTM, and XGBoost models are compared to determine which one performs best in accurately capturing degradation trends. The models' performance was evaluated using a 10x 10-fold cross-validation, generating 100 metrics per model pair. These metrics were analyzed with a hierarchical Bayesian approach, utilizing Markov chain Monte Carlo (MCMC) sampling to estimate the probabilities of one model outperforming another. The LSTM model consistently outperformed exponential degradation and XGBoost and was selected as the most suitable for production due to its superior performance.

Index Terms—Predictive Maintenance, RUL Prediction, Machine Learning System

I. INTRODUCTION

Turbines play a key role in several applications, such as power generation, aerospace, and manufacturing. Over time, they experience wear and degradation, necessitating careful maintenance to prevent unexpected failures and costly downtimes. According to [1], degradation is the state in which the performance capability of equipment is reduced to perform a required function but continues with acceptable performance. In the era of Industry 4.0, the industrial sector increasingly depends on the continuous operation of critical machines to maintain productivity and ensure safety, making the safety, efficiency, and reliability of industrial machines an elementary concern in commercial sectors [2]. To avoid critical damage and abrupt stops of machine operation, faults in rotating machines must be detected as early as possible [3].

Traditional maintenance strategies, such as reactive and preventive maintenance, often yield suboptimal results, including unexpected failures and unnecessary maintenance activities [4]. Reactive maintenance, which addresses failures after they occur, frequently leads to significant operational disruptions. Preventive maintenance, based on fixed schedules, can result in unnecessary maintenance actions, thereby increasing oper-

ational costs without ensuring the avoidance of unexpected failures. In contrast, predictive maintenance has emerged as a key strategy for proactively resolving potential failures [5]. By leveraging historical and real-time data, predictive maintenance aims to forecast potential failures before they happen, enabling timely and targeted maintenance actions.

A crucial aspect of predictive maintenance is the accurate prediction of the Remaining Useful Life (RUL) of critical components[2]. For the calculation of RUL, it is necessary to identify the End of Life (EOL) of the equipment, which is based on the failure threshold [6]. The equation is defined by:

$$RUL(t_p) = t_{eol} - t_p \quad (1)$$

Where t_{eol} is the predicted failure time and t_p is the current time of the prediction. Accurate RUL forecasting allows maintenance activities to be planned based on actual equipment conditions rather than arbitrary schedules or reactive measures. In turbofan engines can effectively avoid serious air disaster due to engine failure by mining its component degradation characteristics[4].

In recent years, advancements in machine learning and the availability of extensive sensor data have transformed RUL prediction. The continuous collection of high-dimensional, high-frequency data from sensors enables a rich dataset for analysis. Machine learning models, especially those designed for time-series, excel in identifying complex, nonlinear patterns in machinery degradation processes [7]. These models surpass traditional statistical methods by capturing temporal dependencies and subtle failure indicators, significantly improving predictive accuracy. Consequently, the integration of advanced machine learning with extensive sensor data has enhanced predictive maintenance systems, providing early warnings, optimizing maintenance schedules, and reducing operational costs.

This paper presents a machine learning-based system for predicting Turbofan Jet Engine RUL, with the aim of improving predictive maintenance strategies. Based on historical machine data, the system presents RUL predictions informing which machines need the most urgent attention. The system processes the data and searches for the best model for the data set. The proposed approach not only captures the degradation

trend behavior but also presents how to choose the best model for predicting the RUL.

The primary contributions of this study are as follows:

- Development and evaluation of a data science project for RUL prediction using historical data from turbine sensors.
- Comparative analysis of various machine learning models, including mathematical model, ensemble learning method and Recurrent Neural Network.
- Exploratory data analysis to understand the behavior of sensor readings up to the moment of machine failure.
- Discussion of the practical implications of the findings for implementing predictive maintenance in industrial settings.

The remainder of this paper is structured as follows. Section 2 reviews the existing literature on RUL prediction and predictive maintenance. Section 3 explains the experimental plan for the system, including the data preprocessing, models development and the search for the best model. Section 4 details the system deployment. Section 5 discuss the results and compares the performance of different models, while Section 6 concludes the paper and suggests directions for future research.

II. LITERATURE REVIEW

In recent years, fault detection algorithms based on neural networks have significantly improved in their accuracy and comprehensiveness. According to [8], it is possible to use a simpler technique for data collection in rotating electrical machines. The author, using only current data and accelerometer data for vibration capture, was able to diagnose both electrical and mechanical faults in an induction motor. In the study by [9], the authors concluded that the use of deep convolutional neural networks (DCNN) can assist in detecting datasets from different equipment, thus avoiding poor network generalization.

Prognostics and Health Management (PHM) methodologies have been increasingly utilized in power systems. [10] conducted research on capacitor degradation within DC-DC converters. [11] applied monitoring methods combined with fuzzy logic to diagnose faults in hydraulic generating units. [12] explored the use of diverse signal processing techniques to monitor bearings and gearboxes in industrial machinery through PHM strategies. [13] introduced a prognostic framework based on models to assess the remaining life of gas turbines in aircraft. [14] used artificial neural networks and support vector machines (SVM) to forecast the remaining service life in the railway sector. Adding to this, [15] demonstrated that neural networks provide effective results in monitoring the health of various systems.

III. EXPERIMENTAL PLAN

A. Data Collection

The dataset employed for this project is the NASA Turbofan Jet Engine dataset, which simulates the run-to-failure degradation of turbofan jet engines. This dataset consists of multiple multivariate time series, each representing an individual engine

within a fleet of similar engines. Initially, all engines operate under normal conditions but eventually develop faults. The experimental plan involved data from 100 engines exhibiting High Pressure Compressor (HPC) degradation, with operating durations ranging from 128 to 356 cycles.

The dataset includes engine ID, cycle times, three columns of operating settings, and 21 sensor readings. A comprehensive list of sensors is provided in the appendix of this paper. Since the dataset's telemetry is simulated up to the point of failure, the cycles can be used to create a RUL column. Analysis of sensor readings over the cycles revealed that some sensors remained constant or lacked significant trends, indicating they do not influence the RUL and can thus be excluded from the training process.

B. Models Pipeline

a) *Long Short-Term Memory (LSTM)*: The data extract, transform and load (ETL) process begins by reading the train and test sets. Following this, a column for RUL is created, representing the target variable for prediction. Next, six columns with constant sensor readings are discarded as they do not provide useful information for the model. The operating settings and time cycles are also discarded. The features are then normalized using the standard scaler, which transforms the data as follows:

$$z = \frac{x - \mu}{\sigma} \quad (2)$$

where x is the input data, μ is the mean, and σ is the standard deviation. This normalization ensures that all features contribute equally to the model training process.

As the LSTM model can capture sufficient historical information, an observation window of 5 samples is employed, meaning the model considers the previous 5 time steps to make predictions about the RUL. In this way, the dataset is transformed from one dimension $(n, 14)$, with n being the number of samples and 14 being the number of features, to $(n - 4, 5, 14)$ for each machine.

The model architecture used consists of two LSTM layers to capture temporal dependencies in the sensor data, followed by two dropout layers to prevent overfitting. The training is performed using the Symmetric Mean Absolute Percentage Error (sMAPE) as the loss function. This metric is defined by:

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{\frac{|y_i| + |\hat{y}_i|}{2}} \times 100 \quad (3)$$

where n is the number of observations, \hat{y}_i is the predicted value, and y_i is the actual value. The advantage of this metric is that it avoids division by zero, treats positive and negative errors symmetrically, and is particularly useful when the absolute difference between the predicted and actual values does not mean the same for values on different scales.

b) *Exponential Degradation*: Like LSTM, the exponential degradation model uses the same etl process, however, after normalization, a Principal Component Analysis (PCA) is performed, used for dimension reduction and feature fusion.

PCA transforms the 14 features into 3 components, with the first component being chosen.

The Exponential degradation model can be defined as below

$$h(t) = \varphi + \theta \exp(\beta t) \quad (4)$$

where $h(t)$ is the health indicator as a function of time, φ is the intercept term considered as a constant, and θ and β are random parameters determining the slope of the model, where θ is lognormal-distributed and β is Gaussian-distributed.

In the training step the training data is used to obtain the values of ϕ , θ , and β for each machine ID. The threshold was defined as the RUL equal to zero of the elements of the first PCA.

For the test, the exponential model fit used the mean parameters of the training data with a lower bound of 25% and an upper bound of 75%. Then, the total remaining cycles to reach the failure threshold.

c) *XGBoost*: For the ETL process, a pipeline was created to handle incoming data by verifying and standardizing column names, as well as checking for the existence of null fields. Sensors and information with no variance, detected earlier in the exploratory phase, are removed from the dataset. The next step in the pipeline is to normalize the data using the *StandardScaler*, as described in Equation 2.

With the data thus processed, an XGBoost regressor was trained with the following hyperparameters:

- Objective function: squarederror
- Number of estimators: 100
- Learning rate: 0.1
- Maximum depth: 5
- random_state: 42 (to ensure data reproducibility)

C. Model Search

In the model search, a 10-fold cross-validation with 10 replications using different seeds is employed. This approach ensures that 10 different combinations of 10 folds are created, providing 100 different test set predictions.

For each prediction set, a metric adapted from Mean Absolute Error (MAE) is used to measure the error. In this metric, the MAE is only calculated if the predicted value or the actual value is equal to or less than 30 cycles. The MAE equation is defined by:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

Since there are 100 different test set combinations, a vector with 100 MAE values is saved. For each model, the mean and the standard deviation of this vector are calculated. To decide the best model, a Bayesian comparison of each pair of models is used, providing the probability of practical equivalence and probability of one model being better than another.

IV. DEPLOYMENT

Once the productive model was selected, a productive environment was created on Databricks through Azure cloud. In this environment, two distinct workflows were created:

the first one for model training and the second one for predicting RUL. The prediction workflow was configured to automatically trigger every day at a specific time, with alerts set up for process failures. The training workflow requires manual triggering according to established model performance rules.

The architecture ensures robustness at both the hardware level, supported by the cloud service with backups and elastic auxiliary clusters as needed, and at the software level, processes were modularized to streamline maintenance, enhance error handling, and enable almost complete automation.

In this environment, both turbine telemetry data and prediction data were saved in separate databases that are versioned and managed using Hive. From the prediction database, a Power BI dashboard was connected for automatic updates. In this dashboard, we can view turbine predictions with a focus on the most critical ones (lowest RUL). In another tab of the dashboard, the model's performance regarding model error in a hypothetical scenario where a turbine is allowed to fail to assess the accuracy of predictions made up to that point is displayed.

The architecture discussed is illustrated in Figure 1.

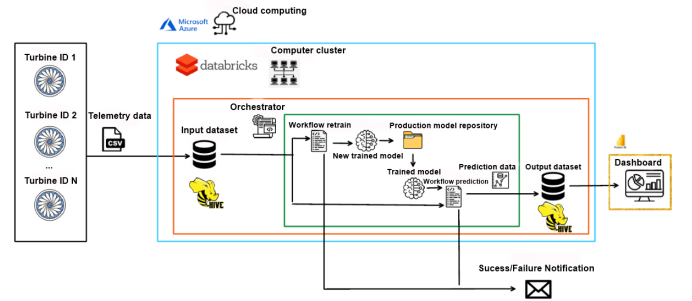


Fig. 1: Cloud System Architecture

V. RESULTS AND DISCUSSION

With the model pipelines constructed, the models were subjected to a 10x 10-fold cross-validation, resulting in 100 metrics for each pipeline. These metrics were summarized in Table I:

Pipeline	Mean Error	Standard Deviation Error
Exponential	20.40	18.92
LSTM	6.56	1.76
XGBoost	6.97	1.46

TABLE I: Summary of 10x 10-cross-validation pipeline errors

The performance of models was evaluated by comparing error differences between each pair using a 10x 10-fold cross-validation, yielding 100 metrics per pair. These metrics are modeled using a hierarchical Bayesian approach to account for data correlation and variability across datasets. Posterior distributions are obtained via MCMC sampling, estimating the probabilities of one model outperforming another. This methodology was presented by [16]. The model with the

highest probability of superior performance is chosen. An example comparison between LSTM and XGBoost was shown in Figure 2. In that example, the LSTM consistently performed better than XGBoost. Between all the models, the LSTM was selected as the most suitable for production.

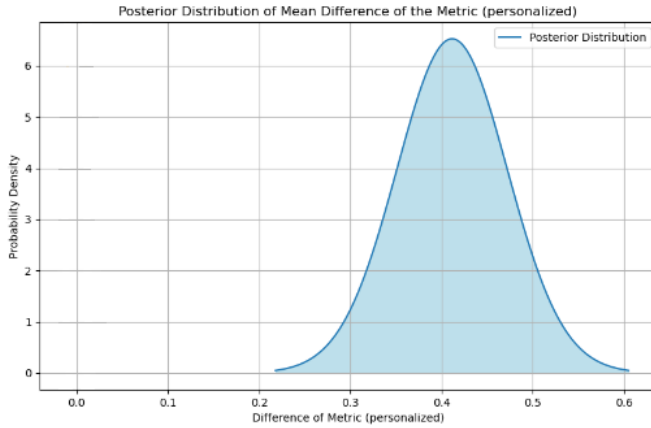


Fig. 2: Error Distribution LSTM x XGBoost

VI. CONCLUSION

The proposed system demonstrated efficiency in the case study, reliably ensuring RUL prediction with only a few days of error. Considering that the user needs a 30-day margin to adjust the logistics for part replacement, it is feasible to plan the replacement when the prediction indicates 45 days. This approach accounts for the slight margin of error in the modeling, maximizes the usage of the part, and reduces costs.

The system's architecture also proved to be robust. In terms of hardware, it is scalable in both processing power and data storage capacity, with redundancy in place to handle potential failures. It includes multiple layers of error detection and alarming to ensure the reliability of the presented data. The processing is nearly fully automated, requiring minimal human intervention.

There were some points identified for future improvements. Two key areas, which are interrelated, were enhancing the modeling and optimizing sensor selection. Reducing the number of required sensors without compromising prediction quality has a significant financial impact, as it decreases the need for sensor maintenance and purchase, and streamlines telemetry data, thereby improving computational efficiency. Improvement in modeling could be achieved through better data handling and testing of different models or the same models with different hyperparameters. Another crucial aspect was adding a user feedback channel to the architecture, allowing the reporting of issues not related to part fatigue, ensuring these cases do not interfere with the system's learning process.

REFERENCES

- [1] *International Electrotechnical Vocabulary - Part 192: Dependability*, International Electrotechnical Commission Std. IEC 60 050-192, 2015. [Online]. Available: <https://webstore.iec.ch/publication/2435>

- [2] O. Asif, S. A. Haider, S. R. Naqvi, J. F. W. Zaki, K.-S. Kwak, and S. M. R. Islam, "A deep learning model for remaining useful life prediction of aircraft turbofan engine on c-mapss dataset," *IEEE Access*, vol. 10, pp. 95 425–95 440, 2022.
- [3] A. Muneer, S. M. Taib, S. Naseer, R. F. Ali, and I. A. Aziz, "Data-driven deep learning-based attention mechanism for remaining useful life prediction: Case study application to turbofan engine analysis," *Electronics*, vol. 10, no. 20, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/20/2453>
- [4] J. Li, Y. Jia, M. Niu, W. Zhu, and F. Meng, "Remaining useful life prediction of turbofan engines using cnn-lstm-sam approach," *IEEE Sensors Journal*, vol. 23, no. 9, pp. 10 241–10 251, 2023.
- [5] CheckProof, "Predictive maintenance vs preventive maintenance: Maximize your industrial efficiency," 2023. [Online]. Available: <https://checkproof.com/en/blog/preventive-vs-predictive-maintenance/>
- [6] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, pp. 1–16, 2010. [Online]. Available: <https://www.phmsociety.org/node/1231>
- [7] A. Muneer, S. M. Taib, S. M. Fati, and H. Alhussian, "Deep-learning based prognosis approach for remaining useful life prediction of turbofan engine," *Symmetry*, vol. 13, no. 10, 2021. [Online]. Available: <https://www.mdpi.com/2073-8994/13/10/1861>
- [8] M. Z. Ali, M. N. S. K. Shabbir, X. Liang, Y. Zhang, and T. Hu, "Machine learning-based fault diagnosis for single- and multi-faults in induction motors using measured stator currents and vibration signals," *IEEE Transactions on Industry Applications*, vol. 55, no. 3, pp. 2378–2391, 2019.
- [9] S. Li, F. Jia, J. Shen, and J. Ma, "Deep transfer attention network for intelligent fault diagnosis of rolling bearings," *Journal of Physics: Conference Series*, vol. 1983, no. 1, p. 012011, jul 2021. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/1983/1/012011>
- [10] C. Kulkarni, G. Biswas, and X. Koutsoukos, "A prognosis case study for electrolytic capacitor degradation in dc-dc converters," 09 2009.
- [11] I. Marcos, A. Álvarez, and M. Abadia, "Metodologia para diagnóstico de falhas em unidades geradoras hidráulicas usando conceitos de lógica difusa," *ABCM Symposium Series in Mechatronics*, vol. 6, pp. 1–10, 2014.
- [12] A. Soualhi, Y. Hawwari, K. Medjaher, G. Clerc, R. Hubert, and F. Guillet, "Phm survey: implementation of signal processing methods for monitoring bearings and gearboxes," *International Journal of Prognostics and Health Management*, vol. 9, no. 2, pp. 1–14, 2018.
- [13] O. Alozie, Y.-G. Li, X. Wu, X. Shong, and W. Ren, "An adaptive model-based framework for prognosis of gas path faults in aircraft gas turbine engines," *International Journal of Prognostics and Health Management*, vol. 10, no. 2, pp. 1–12, 2019.
- [14] T. Böhm, "Remaining useful life prediction for railway switch engines using classification techniques," *International Journal of Prognostics and Health Management*, vol. 8, no. Sp7, pp. 1–15, 2017.
- [15] E. F. Costa, "Predição da descarga e capacidade de baterias li-íon utilizadas em vants," *Dissertação de Mestrado em Engenharia Elétrica e da Computação*, Universidade Federal do Ceará, Sobral, 2019.
- [16] A. Benavoli, G. Corani, J. Demsar, and M. Zaffalon, "Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis," *Journal of Machine Learning Research*, vol. 18, pp. 1–36, 2017. [Online]. Available: <http://jmlr.org/papers/v18/16-305.html>

APPENDIX A

List of sensors reading:

- sensor 1: (Fan inlet temperature) (°R),
- sensor 2: (LPC outlet temperature) (°R),
- sensor 3: (HPC outlet temperature) (°R),
- sensor 4: (LPT outlet temperature) (°R),
- sensor 5: (Fan inlet Pressure) (psia),
- sensor 6: (bypass-duct pressure) (psia),
- sensor 7: (HPC outlet pressure) (psia),
- sensor 8: (Physical fan speed) (rpm),
- sensor 9: (Physical core speed) (rpm),
- sensor 10: (Engine pressure ratio(P50/P2)),

- sensor 11: (HPC outlet Static pressure) (psia),
- sensor 12: (Ratio of fuel flow to Ps30) (pps/psia),
- sensor 13: (Corrected fan speed) (rpm),
- sensor 14: (Corrected core speed) (rpm),
- sensor 15: (Bypass Ratio),
- sensor 16: (Burner fuel-air ratio),
- sensor 17: (Bleed Enthalpy),
- sensor 18: (Required fan speed),
- sensor 19: (Required fan conversion speed),
- sensor 20: (High-pressure turbines Cool air flow),
- sensor 21: (Low-pressure turbines Cool air flow)