JABATAN KEJURUTERAAN MEKANIK
Department of Mechanical Engineering

Tugasan Kumpulan
Group Assignment

Sesi & Sem **Session & Sem**	Session 2023/2024 (Sem 2)
Kod & Nama Kursus **Course Code & Name**	KIG4068 Machine Learning
Tajuk projek **Title of Project**	Predictive Maintenance of Turbofan Jet Engine
Kumpulan **Group**	G14
Tarikh Hantar **Submission Date**	14 June 2024

Group Members:

No.	Name	Matric No.
1	Ch’ng Zhen Huo	S2131858/1
2	Muhammad Shahril Bin Zainol Abidin	17205720/2

GitHub Repository: [ZhenHao03/KIG4068-Assignment (github.com)]

1. ABSTRACT

Predictive maintenance of turbofan jet engines is crucial for enhancing the reliability, efficiency, and safety of aircraft operations. This project focuses on developing a machine learning model to accurately predict the Remaining Useful Life (RUL) of turbofan engines using time-series sensor data from the NASA CMAPSS dataset. A baseline Linear Regression model was initially employed to establish a performance benchmark. Comprehensive exploratory data analysis (EDA) informed the creation of a robust preprocessing pipeline, which included scaling, rolling window transformations, and feature extraction. Various advanced models, including SVM, Decision Tree, Random Forest, XGBoost, and NGBoost, were evaluated. The tuned NGBoost Regressor, enhanced through meticulous hyperparameter tuning, demonstrated superior performance with the lowest test RMSE and MAE. The final model provided highly accurate RUL predictions with 40% reduction of both RMSE and MAE when compared to the baseline model, underscoring the effectiveness of combining sophisticated preprocessing techniques and advanced machine learning algorithms for predictive maintenance applications.

2. INTRODUCTION

In the modern aviation industry, the reliability and efficiency of aircraft engines are paramount. Predictive maintenance, which involves forecasting potential failures before they occur, has become a critical component in maintaining aircraft fleets' safety and operational readiness. By leveraging advanced data analytics and machine learning techniques, predictive maintenance aims to optimize the lifecycle management of critical engine components, thereby reducing downtime, maintenance costs, and the risk of in-flight failures.

Turbofan jet engines, the powerhouse of modern commercial and military aircraft, operate under extreme conditions, making them susceptible to wear and degradation over time. Traditional maintenance strategies, which rely on scheduled inspections and repairs, are often reactive and may lead to unexpected engine failures, costly repairs, and extended periods of aircraft downtime. Given the complexity and high operational demands of jet engines, there is a need for a more proactive approach to maintenance.

This machine learning (ML) project addresses the challenge of predicting the Remaining Useful Life (RUL) of turbofan jet engines using time-series data collected from various sensors monitoring engine performance. The primary goal is to develop a ML model capable of accurately predicting the RUL, enabling timely and targeted maintenance actions. Such a predictive maintenance system will enhance aircraft operations' reliability, efficiency, and safety.

3. IDENTIFYING GAPS IN AVAILABLE CODES

1. Inadequate data exploration:
    - A large number of current codes don't function well with data exploration.
    - Their lack of knowledge about the properties and structure of the data will result in poor feature selection and preprocessing.
    - In order to provide a clear and thorough data exploration, we had addressed this issue by performing some analysis with the data, such as finding the distribution, correlation, or any patterns we can identify within the data.

2. Limited preprocessing:
    - The other scripts have few and badly executed preprocessing processes. They performed poorly during data training because they neglected to complete important preprocessing tasks such data segmentation, normalization, scaling, and handling missing values.
    - We standardize and normalize the sensor data in our codes so that it is identical at the beginning point of each different engine unit. We also set up a pipeline for preprocessing that is more suitable. The poor correlation features were eliminated, the data was scaled using StandardScaler and a rolling window, and PCA was carried out in the pipeline.
3. Basic Feature Engineering:
    - Presently available codes lack feature engineering or have very little of it. They frequently rely on unprocessed sensor data without identifying any important aspects.
    - In contrast, we performed the feature engineering using TSFresh. Models may underperform if sophisticated feature engineering is not used to identify the underlying patterns in the data.
4. Lack of Hyperparameter Tuning:
    - There is a lack of systematic hyperparameter adjustment in many codes. Without optimization, the default hyperparameters are employed. When inappropriate hyperparameters are used, models may perform poorly.
    - We will perform hyperparameter tuning of the best model using GridSearchCV.
5. Limited Model Exploration:
    - Few codes investigate a wide range of models. A small number of widely used models, such as Linear Regression, SVM, and Random Forest are frequently used. There is no investigation into the performance of novel models like NGBoost. In our code, the NGBoost model proved to be the most effective one for data prediction.

4. DATASET DESCRIPTION

4.1 Details of the NASA C-MAPSS Turbofan Jet Engine Dataset

The Prognostics Center of Excellence (PCoE) at NASA Ames contributed the dataset used for this project, which simulates engine degradation using the Commercial Modular Aero Propulsion System Simulation (C-MAPSS) (Saxena & Goebel, 2008).

Table 4.1: Details of the C-MAPSS Datasets

Dataset	FD001	FD002	FD003	FD004
Train size (# of engine units)	100	260	100	249
Test size (# of engine units)	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

As shown in Table 4.1, the dataset consists of four unique sets, each of which simulates engines under various combinations of operating conditions and fault modes. Subsets for testing and training are created from each dataset. The data can be regarded as coming from a fleet of engines of the same type because each time series comes from a distinct engine. The user is unaware of the varying degrees of initial wear and manufacturing variance present in each engine. This wear and variance is not seen as a defect; rather, it is deemed typical. The engine performance is significantly impacted by three operational settings. The data also contains these settings. There is sensor noise tainting the data. More details can be seen in Table 4.2 and Table 4.3.

Table 4.2: Summary of the FD001 Dataset

Features	Engine Unit	Time, in cycles	Operational Setting	Sensors
Column Number	1	2	3-5	6-26

Table 4.3: Pandas head() of FD001 dataset

	unit	time	OS1	OS2	OS3	SM1	SM2	SM3	SM4	SM5	...	SM12	SM13	SM14	SM15	SM16	SM17	SM18	SM19	SM20	SM21
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044

4.2 Exploratory Data Analysis

From Figure 4.1, it is observed that the engine units in the test set have shorter lifetimes than the train set. From the describe() function, on average, test set lifetimes are 70 time cycles shorter compared to the train set.

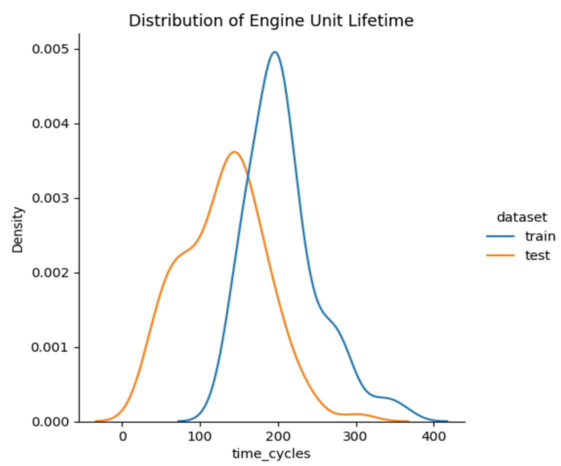


Figure 4.1: Distribution of engine unit lifetime

As expected, the train set contains engine units with RUL way higher than the true RUL as shown in Figure 4.2. This is because the train set has the full lifetime data of an engine while the test set only contains data until a period of time before its end of life. Hence, the target is to predict the RUL at the last cycle of each engine unit in the test set.

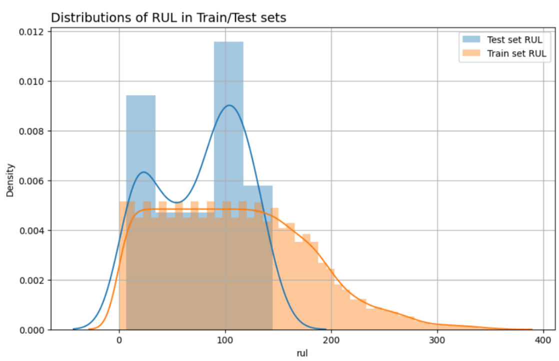


Figure 4.2: Distribution of True RUL

For Figure 4.3, engine 1, 2, 5, and 10 have been chosen to visualize the correlation between RUL and time cycles. It is obvious that as the engines have been running for many time cycles, the RUL will be reduced.

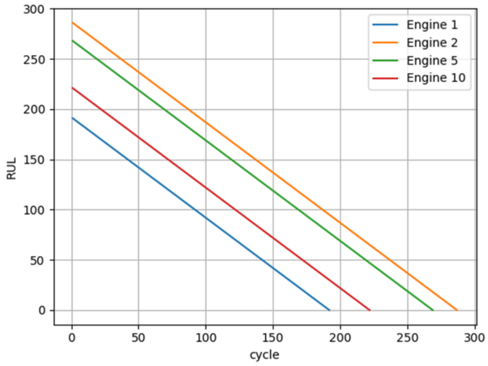


Figure 4.3: RUL against time cycles

Figure 4.4 shows the time series data of each sensor (sensor 1-21) for each engine unit. It is observed that sensors 1, 5, 10, 16, 18, and 19 are constant throughout the time cycles. This rendered these sensors useless as they have zero variance and can be dropped during data preprocessing. For other sensors, it can be seen that the sensor values will change with respect to time cycles, whether increasing or decreasing. Thus, this indicates that they will have correlation to the target. In addition, feature scaling is needed in data preprocessing due to the different range of sensor values.

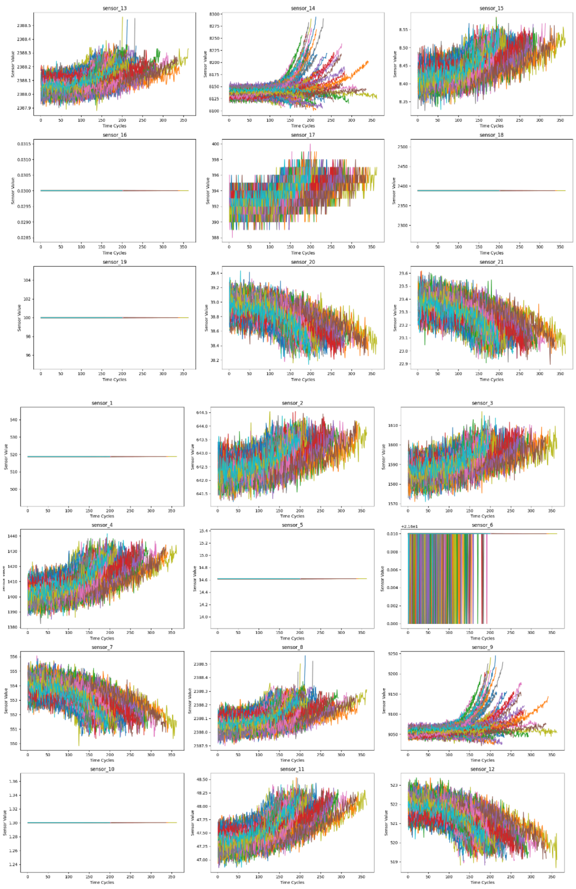


Figure 4.4: Sensor time-series data

Figure 4.5 shows the plots of two units of engine with the biggest difference in mean values for sensor 8, 9, 11, 12, 13, and 14. Each unit has a different starting point and the scaling is off. Thus, this suggests that we might need to scale the sensor time series with respect to the start of every individual engine's time series. Scaling with respect to the individual engines starting values allows us to bring all the engines time series to the same scale.

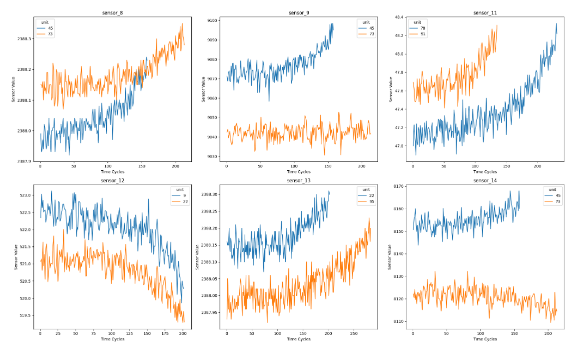


Figure 4.5: Uneven scaling across engine unit

5. METHODOLOGY

5.1 Building Preprocessing Pipeline

To prepare the time-series sensor data for predictive modeling, we implemented a comprehensive preprocessing pipeline. The pipeline consists of several sequential steps designed to clean, transform, and reduce the dimensionality of the data, ensuring it is suitable for training machine learning models. The preprocessing pipeline was constructed using the following steps:

1. Low Variance Feature Removal: Features with little to no variance do not contribute significant information for the predictive model and can be considered noise. Thus, they can be dropped.
2. Scaling Per Engine: This step involves normalizing the sensor data for each engine individually. Normalization ensures that the data for each engine unit is scaled appropriately, accounting for variations in operational conditions.
3. Time-Series Rolling: To capture temporal patterns and trends in the data, a rolling window transformation is applied. This involves shifting the time-series data by a specified number of cycles.
4. Feature Extraction with TSFresh: TSFresh (Time Series Feature extraction based on scalable hypothesis tests) is employed to extract a comprehensive set of features from the time-series data. This step transforms the raw time-series data into a set of meaningful statistical features.
5. Dimensionality Reduction with PCA: To reduce the complexity of the dataset and mitigate the risk of overfitting, Principal Component Analysis (PCA) is performed. PCA helps in reducing the number of features while retaining most of the variance in the data.
6. Feature Selection with TSFresh: A final feature selection step is applied to retain only the most significant features. This step helps in further reducing the dimensionality and enhancing the model's performance by focusing on the most relevant features.

5.2 Model Training and Evaluation

To develop a robust machine learning model for predicting the Remaining Useful Life (RUL) of turbofan jet engines, we followed a systematic approach that involved building a baseline model, evaluating several advanced models, and performing hyperparameter tuning. The steps in this process are detailed below:

1. Baseline Model Development: As a reference point, we built a baseline model using the simplest regression algorithm, Linear Regression, without applying any preprocessing or feature engineering techniques. This baseline model serves to establish a performance benchmark.
2. Model Training with Preprocessing Pipeline: Next, we trained a variety of machine learning models (SVM, Decision Tree, Random Forest, XGBoost, and NGBoost) after applying the preprocessing pipeline described earlier.
3. Cross-Validation: Stratified group k-fold cross-validation method was employed because our data is grouped in terms of engine units. Cross-validation was performed with 5 folds (k=5). Here, root mean squared error (RMSE) and mean absolute error (MAE) for both training and cross-validation were recorded.

4. Best Model Selection: Models were ranked based on their validation performance, and the model with the best trade-off between bias and variance was selected for further tuning.
5. Hyperparameter Tuning: The best-performing model underwent hyperparameter tuning to optimize its performance further. GridSearchCV was used to systematically explore a range of hyperparameters for the selected model.
6. Model Evaluation: The tuned model was then tested on the test set to predict the RUL of each engine unit. The predicted RUL will be compared to the true RUL provided by the NASA CMAPSS dataset. The final performance metrics (RMSE and MAE) were computed to assess the model's accuracy

6. RESULTS AND DISCUSSIONS

6.1 Cross-Validation and Best Model Selection

Table 6.1: Results of model training and cross-validation

No	Model	RMSE		MAE	
		Train	Validation	Train	Validation
1	Linear Regression (Baseline)	23.80	23.42	19.42	19.32
2	Linear Regression + Preprocessing	15.47	15.62	12.43	12.61
3	SVM + Preprocessing	14.54	15.82	10.71	12.23
4	Decision Tree + Preprocessing	0.00	19.45	0.00	14.97
5	Random Forest + Preprocessing	2.74	14.26	1.81	10.99
6	XGBoost + Preprocessing	4.79	14.55	3.37	11.12
7	NGBoost + Preprocessing	13.11	14.09	10.21	10.82

Table 6.1 above shows the training and cross-validation results of various ML models for predicting the RUL of a turbofan jet engine. The baseline model, Linear Regression without any preprocessing, exhibited high error metrics with an RMSE of 23.80 and an MAE of 19.42 on the training set, and an RMSE of 23.42 and an MAE of 19.32 on the validation set. Incorporating the preprocessing pipeline improved the performance of Linear Regression markedly, reducing the RMSE and MAE to 15.47 and 12.43 on the training set, and 15.62 and 12.61 on the validation set, respectively. This significant improvement highlights the effectiveness of the preprocessing pipeline, which included steps such as scaling, rolling window transformations, and feature extraction, in enhancing the model's predictive capability.

Advanced models such as Support Vector Machine (SVM), Decision Tree, Random Forest, XGBoost, and NGBoost were evaluated with the preprocessing pipeline applied. While SVM showed no notable improvement over Linear Regression, the Decision Tree model suffered from severe overfitting, with perfect training scores but poor validation performance. In contrast, ensemble methods like Random Forest and XGBoost demonstrated significant reductions in errors, highlighting their ability to capture complex patterns. However, Random Forest and XGBoost still can be considered overfitting to the training set.

NGBoost emerged as one of the best-performing models, achieving low validation errors and demonstrating robust generalization capabilities. It showed balanced performance between training and cross-validation while obtaining the lowest RMSE and MAE when compared among others. Thus, NGBoost is selected for further tuning and evaluation with the test sets.

6.2 Hyperparameter Tuning

Table 6.2: Hyperparameter tuning of NGBoost Regressor using GridSearchCV

No	NGBoost Parameters	Grid	Best Value
1	minibatch_frac	[1, 0.8, 0.5]	1
2	n_estimators	[100, 200, 300, 400]	300
3	learning_rate	[0.01, 0.05, 0.1]	0.05
4	Base	[DecisionTreeRegressor(criterion='friedman_mse', max_depth=4), DecisionTreeRegressor(criterion='friedman_mse', max_depth=5, max_features=0.8, min_samples_leaf=50)]	DecisionTreeRegressor(criterion='friedman_mse', max_depth=5, max_features=0.8, min_samples_leaf=50)

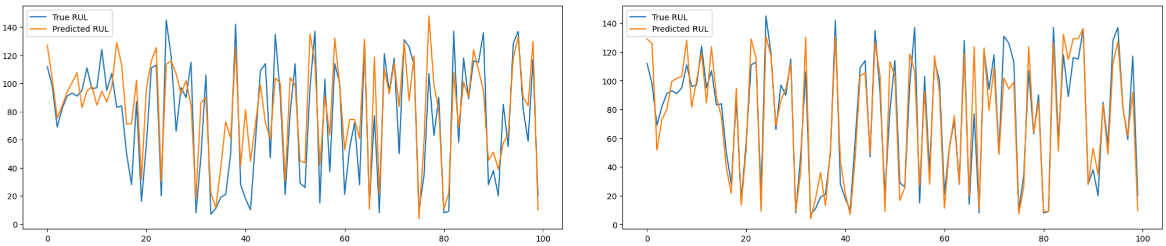
Further, the NGBoost model underwent hyperparameter tuning using GridSearchCV to optimize its performance as shown in Table 6.2. Initially, the hyperparameters for learning rate, minibatch fraction, and the number of estimators were tuned, resulting in the best parameters: a learning rate of 0.05, a minibatch fraction of 1, and 300 estimators. Subsequent tuning focused on finding the best base learner, comparing two configurations of the DecisionTreeRegressor. The best base learner was identified as a DecisionTreeRegressor with a max depth of 5, max features of 0.8, and a minimum of 50 samples per leaf. Evaluating the tuned NGBoost model using the same stratified group k-fold cross-validation method (k=5) showed an improvement in performance metrics. The training RMSE and MAE improved from 13.11 to 10.80 and from 10.21 to 7.92, respectively. Similarly, the cross-validation RMSE and MAE improved from 14.09 to 13.62 and from 10.82 to 10.40, respectively.

6.3 Model Testing

Table 6.3: Test RMSE results from model testing

No	Model	Test RMSE
1	Linear Regression (Baseline)	22.37
2	LSTM (Ebrahim, 2022)	15.71
3	Random Forest (Kirstein, 2022)	17.96
4	XGBoost Regressor (Tyagi, 2020)	20.62
5	Tuned NGBoost Regressor (our best model)	13.19

Table 6.3 above presents the test RMSE results of various models for predicting the Remaining Useful Life (RUL) of turbofan jet engines, including our baseline Linear Regression model, our best-performing tuned NGBoost model with preprocessing, and several other models sourced from online repositories such as Kaggle. It is observed that Our best model, the tuned NGBoost with preprocessing, achieved the lowest RMSE of 13.19, significantly outperforming all other models. This success is attributed to a very comprehensive exploratory data analysis (EDA) that enabled us to establish an appropriate preprocessing pipeline. Additionally, we experimented with many ML algorithms and determined that the NGBoost Regressor provided the best results. The model's performance was further enhanced by meticulous hyperparameter tuning.



Baseline Model

Tuned NGBoost Regressor

Figure 6.1: Comparison of Predicted RUL against True RUL

Figure 6.1 compares the performance of the baseline Linear Regression model and the tuned NGBoost Regressor in predicting the Remaining Useful Life (RUL) of turbofan jet engines. These visualizations provide a clear depiction of how each model's predicted RUL values align with the true RUL values across various engine units.

In the plot for the Baseline Model, the predicted RUL values (orange line) show considerable deviation from the true RUL values (blue line) across many engine units. This discrepancy indicates that the baseline model struggles to capture the underlying patterns and complexities in the time-series sensor data, leading to less accurate predictions. The high RMSE and MAE values for the baseline model, as discussed earlier, further confirm this observation.

Conversely, the plot for the Tuned NGBoost Regressor demonstrates a much closer alignment between the predicted RUL values and the true RUL values. The orange line representing the predicted RUL follows the blue line of the true RUL more closely, indicating that the tuned NGBoost model is better at capturing the temporal dependencies and degradation patterns inherent in the sensor data. This improved alignment is reflected in the significantly lower RMSE and MAE values for the tuned NGBoost model, highlighting its superior predictive performance.

7. CONCLUSION

In conclusion, this machine learning project demonstrated the successful development and evaluation of ML models for predicting the Remaining Useful Life (RUL) of turbofan jet engines, with the tuned NGBoost Regressor emerging as the best-performing model. The NGBoost Regressor achieved test RMSE and MAE of 13.19 and 10.14 respectively, which indicates around -40% reduction when compared to the Baseline Model. Through comprehensive exploratory data analysis, an effective preprocessing pipeline, and meticulous hyperparameter tuning, the NGBoost model achieved superior accuracy and robust generalization, significantly outperforming the baseline Linear Regression and other models available online. This model provides a reliable tool for proactive maintenance planning, enhancing the safety, efficiency, and cost-effectiveness of aircraft operations.

8. CONTRIBUTIONS OF EACH MEMBER

8.1 Muhammad Shahril

- Leader of Group 14. Scheduled meetings virtually or physically when needed.
- Performed exploratory data analysis (EDA) and model training on the chosen dataset.
- Responsible for Introduction, Methodology, Results, Discussions, and Conclusion in report writing.

8.2 Zhen Huo

- Developed preprocessing pipeline.
- Prepared GitHub repository template for report writing.
- Responsible for Identifying Gaps in Available Code and Dataset Description in report writing.

9. REFERENCES

1. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137, 106024. https://doi.org/https://doi.org/10.1016/j.cie.2019.106024
2. Ebrahim, I. (2022). RUL prediction using Long Short Term Memory (LSTM). *Kaggle*. https://www.kaggle.com/code/easmeileabrhim/rul-prediction-using-long-short-term-memory-lstm
3. Ferreira, C., & Gonçalves, G. (2022). Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. *Journal of Manufacturing Systems*, 63, 550-562. https://doi.org/https://doi.org/10.1016/j.jmsy.2022.05.010
4. Kirstein, C. (2022). Predictive Maintenance NASA turbofan Regression. *Kaggle*. https://www.kaggle.com/code/carlkirstein/predictive-maintenance-nasa-turbofan-regression#notebook-container
5. NASA. (2021). Turbofan Engine. https://www.grc.nasa.gov/www/k-12/airplane/aturbf.html
6. Saxena, A., & Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data
7. Tyagi, V. (2020) Damage Propagation Modeling for Aircraft Engine. *Kaggle*. https://www.kaggle.com/code/vinayak123tyagi/damage-propagation-modeling-for-aircraft-engine