Jet Engine Lifecycle Survival Analysis and Remaining Useful Life (RUL) LSTM Prediction

Team 7: Yang Lu, Sijian Tan, Cheng Zhang, Siming Zhao

Abstract

The aerospace industry is heavily dependent on rigorous maintenance of jet engines to ensure safety and operational efficiency. This project introduces an innovative approach for predicting the Remaining Useful Life (RUL) of jet engines, with the goal of optimizing maintenance schedules and preventing severe damage. The Cox Proportional Hazards (CoxPH) model is utilized to conduct a survival analysis, estimating the probability of engine survival. Furthermore, a Long Short-Term Memory (LSTM) neural network is employed for a detailed numerical prediction of the RUL based on current sensor readings. The results illustrate the potential of combining traditional statistical models with advanced machine learning techniques for effective predictive maintenance in complex engineering systems.

1 Motivations & Objectives

Prognostics and Health Management (PHM) is a critical area in the aerospace industry that focuses on understanding and predicting the condition of aircraft engines [1]. PHM is essential to figure out when an engine might need maintenance before it actually breaks down. By doing so, PHM helps to keep planes flying without interruption and prevents engine problems before they happen. This is important for keeping flights on schedule, reducing repair costs, and making sure that safety is always the top priority.

Traditional approaches to predicting the Remaining Useful Life (RUL) of jet engines typically involve analyzing sensor readings to assess the engine's current condition. The challenge with this method is the vast scale of statistical data generated by these sensors, which can make accurate predictions difficult using conventional statistical techniques. In contrast, machine learning offers significant benefits for processing and making sense of this large-scale data. For instance, the Cox Proportional Hazards (CoxPH) model provides a probabilistic approach to identify and understand the various risk factors that could affect the lifespan of a jet engine [2]. It also helps in estimating the engine's probability of continuing to function without failure. Additionally, the Long Short-Term Memory (LSTM) model [3] is particularly effective for time series data, allowing for predictions that evolve in response to the engine's historical and current operating data. Thus, this paper addresses the challenge of accurately predicting jet engine RUL by harnessing the strengths of both CoxPH and LSTM models to improve prediction reliability and maintenance scheduling.

2 Data Description

The dataset used in this project is sourced from a public repository overseen by NASA's Prognostics Center of Excellence (CoE). It is specifically crafted for modeling asset degradation and employs the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) framework. This comprehensive dataset comprises four unique subsets, each representing a spectrum of operational conditions and possible fault modes.

The dataset is available in Kaggle, including the training, testing, and RUL data files for 4 different operation condition combination sets. In training and testing data files, there are 26 columns. The variable of each column is shown in Figure 1.

Parameters	Parameters
Engine number	Sensor 9: Physical core speed (Nc, rpm)
Time in cycles	Sensor 10: Engine pressure ratio (epr, P50/P2)
Operation Condition 1: Altitude	Sensor 11: Static pressure at HPC outlet (Ps30, psia)
Operation Condition 2: Mach number	Sensor 12: Ratio of fuel flow to Ps30 (phi, pps/psi)
Operation Condition 3: Throttle resolver angle (TRA)	Sensor 13: Corrected fan speed (NRf, rpm)
Sensor 1: Total temperature at fan inlet (T2, R)	Sensor 14: Corrected fan speed (NRc, rpm)
Sensor 2: Total temperature at LPC outlet (T24, R)	Sensor 15: Bypass ratio (BPR)
Sensor 3: Total temperature at HPC outlet (T30, R)	Sensor 16: Burner fuel-air ratio (farB)
Sensor 4: Total Temperature LPT outlet (T50, R)	Sensor 17: Bleed Enthalpy (htBleed)
Sensor 5: Pressure at fan inlet (P2, psia)	Sensor 18: Demanded fan speed (Nf_dmd, rpm)
Sensor 6: Total pressure in bypass-duct (P15, psia)	Sensor 19: Demanded corrected FS (PCNfR_dmd, rpm)
Sensor 7: Total pressure at HPC outlet (P30, psia)	Sensor 20: HPT coolant bleed (W31, lbm/s)
Sensor 8: Physical fan speed (Nf, rpm)	Sensor 21: LPT coolant bleed (W32, lbm/s)

Figure 1: Variables

Besides this, there are 6 operation conditions, which include:

1. Condition 1: Altitude = 0, Mach = 0, TRA = 100
2. Condition 2: Altitude = 10, Mach = 0.25, TRA = 100
3. Condition 3: Altitude = 20, Mach = 0.7, TRA = 100
4. Condition 4: Altitude = 25, Mach = 0.62, TRA = 60
5. Condition 5: Altitude = 35, Mach = 0.84, TRA = 100

And 4 combination sets are:

1. FD001: Condition 1 only
2. FD002: Mix of all the conditions
3. FD003: Condition 1 only
4. FD004: Mix of all conditions

In the training data file, the maximum time cycle will be the maximum life cycle of the engine. In the testing data file, the maximum time cycle is the latest observation cycle, which needs to be added by the corresponding cycles in RUL data file to get the maximum life cycle.

3 Data Preprocessing

Data preprocessing is a critical step in the data analysis process, as it prepares raw data for further processing and analysis. It involves cleaning and transforming the data to correct inaccuracies, handle missing values, and standardize formats, ensuring that the dataset is consistent and reliable.

Two feature selections method are chosen in this project:

1. **Variance threshold** [4]: by eliminating features with low variance, this method focuses on the most influential variables, effectively reducing the dimensionality of the data. Variance threshold method could reduce redundant data but also enhance the performance of machine learning models by retaining only the most relevant predictors.

2. **Random Forest** [5]: effective in ranking features based on importance in making accurate predictions by the use of multiple decision trees; also in handling non-linear relationships. Combining these two balances the simplicity and computational efficiency, improve the effectiveness and performance of the model.

   The chosen features for each dataset are shown below (S: sensor, OP: operation condition):

   (a) FD001: S2, S3, S4, S7, S9, S11, S12, S14, S20

   (b) FD002: OP1, OP2, S2, S3, S4, S7, S8, S9, S11, S12, S13, S14, S15, S20, S21

   (c) FD003: S2, S3, S4, S7, S8, S9, S11, S12, S13, S14, S20, S21

   (d) FD004: OP1, OP2, S2, S3, S4, S7, S8, S9, S11, S12, S13, S14, S15, S20, S21

MinMaxScaler method is utilized for normalization, to transform features by scaling them to a given range, typically between 0 and 1, but preserve the shape of the original distribution.

4 CoxPH Survival Analysis

4.1 Introduction

Survival Analysis, commonly known time-to-event analysis, is a statistical method employed to analyze the duration until an event takes place. **Time-to-event data** usually refers to a type of data that measures the time it takes for a specific event to occur. This event could encompass a range of scenarios, including the occurrence of an subject death in a medical study, machine component failure, customer unsubscribing and so on. Therefore, for a time-to-event dataset, it often includes the following variables:

- **Time Variable:** Represents the time until the event of interest to occur.
- **Event Indicator:** A binary variable indicating whether the event has occurred or not at the observation point
- **Feature Variables:** Additional Variables provide information about the subjects in the study.

In our study, we are analyzing the remaining useful life of jet engines, which will be our time variable and the engine failure will be our event of interest.

Cox Proportional Hazard (CoxPH) model is considered a foundational model for survival analysis. One of the strengths of CoxPH model lies in its ability to properly handle **right-censored data** where the event of interest has not occurred by the end of the study. CoxPH model is able to utilize the information up to the censoring time for estimation. In our scenarios, all of the engine failures have been observed, so CoxPH model could use the exact event times for estimation.

CoxPH model is aimed to estimate the hazard function $\mathbf{h(t)}$ (1), which describes how the hazard (instantaneous risk of an event) changes over time.

$$h(t) = h_0(t)e^{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n} \tag{1}$$

In this equation:

- $h_0(t)$ is a baseline hazard function (Non-parametric).
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients corresponding to the covariates $x_1, x_2, \ldots, x_n$.

The baseline hazard function captures the underlying risk at t in the absence of the covariates, while the coefficients β quantify the impact of the covariates on the hazard, providing insights into how the risk evolves over time.

The CoxPH model is optimized by maximizing the partial likelihood function $\mathbf{L(\boldsymbol{\beta})}$ (2)

$$L(\beta) = \prod_{Y_i \; Uncensored} \frac{\exp(X_i^\top \beta)}{\sum_{Y_j \geq Y_i} \exp(X_j^\top \beta)} \tag{2}$$

The partial likelihood only considers the ordering of event times while discarding the information about the baseline hazard function, because it doesn't make any assumption on the baseline hazard.

4.2 Implementation

The datasets we selected are time-series, providing information on the status of each engine at every cycle. However, CoxPH model is typically more suitable for time-to-event datasets. To develop a CoxPH model to estimate RUL of jet engines, we firstly preprocessed the training datasets by transferring the *current cycle* to RUL using the following equations (3):

$$\text{RUL} = \text{Max cycle} - \text{Current cycle} \tag{3}$$

In order to maximize the utilization of the available information, we decided to calculated RUL for each machine at every cycle and treat each machine at each cycle independently. For the testing datasets, we utilized the RUL of each machine at the max cycle provided for validation.

After preparing the RUL(time variable) for our input, we also created an event indicator to denote whether the engine failure is observed or not. In our training datasets, as every instance corresponds to an observed engine failure, we designated all instances as *True*.

By creating proper time variable and event indicator, we transformed our time-series datasets to time-to-event datasets. This transformation enables us to apply CoxPH model to predict RUL.

4.3 Model Build

We trained the CoxPH model using Efron's method to handle tied event times because the default method assumes the time when objects experience the event to be distinct. The default assumption may not hold in the presence of the tied events (i.e multiple engine failures take place at the same time) due to the way we prepared our training datasets.

In addition to modeling the hazard function $\mathbf{h(t)}$ (1), CoxPH allows us to derive the survival function $\mathbf{S(t)}$ (4) from the hazard function as well [6].

$$S(t) = e^{-\int h(t)} dt \tag{4}$$

The survival function is aimed to estimate the probability of an event occurring after time t. In other words, it estimates the probability that the engine could survive until time point t. The official definition of $\mathbf{S}(\mathbf{t})$ (5) is as follow:

$$S(t) = P(T > t) = 1 - F(t) \tag{5}$$

where T is random variable and $F(t)$ is the cumulative distribution function.

In pursuit of analyzing RUL of each jet engine, predicting survival function of each engine would yield more comprehensive and straightfoward information as needed, we decided to use CoxPH to output the predicted survival functions for each machine

5 Long Short-Term Memory (LSTM) Prediction

5.1 Introduction

Traditional Recurrent Neural Network (RNN) is prone to the problems of gradient vanishing or gradient explosion when dealing with long sequences, which limits their ability to capture long-term dependencies [7]. While LSTM is a variant of RNN and represent a type of temporal recurrent neural network. LSTM effectively address these issues through their cell state and gating mechanisms [8]. As shown in Figure 2, the gating mechanism of LSTM involves three key components:

- **Forget Gate:** The forget gate determines which information in the memory cell needs to be retained. It produces an output value between 0 and 1 through a sigmoid activation function, indicating the extent to which information at the corresponding should be forgotten.
- **Input Gate:** The input gate regulates the entry of new information into the memory cell. It includes a sigmoid activation function that determines which information will be written to the memory cell. If the output of the input gate is close to 1, it signifies that the network should memorize this information.
- **Output Gate:** The output gate controls the flow of information from the memory cell to the hidden state. It contains a sigmoid activation function, as well as a hyperbolic tangent activation function to adjust the values of the memory cells.

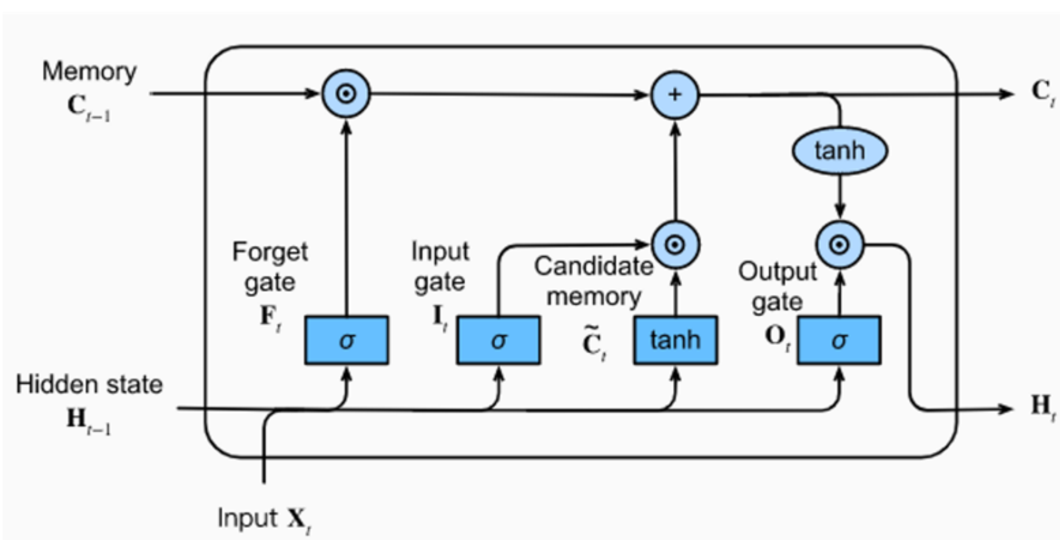

Figure 2: LSTM - Cell

In summary, the input gate, forget gate, and output gate empower LSTM more efficiently than RNN to become a good choice for processing time series data.

5.2 Implementation

We leverage a sliding window approach to preprocess time series data, prior to inputting data into the LSTM model. The primary purpose of this approach is to better consider the data preceding the current time point and transform the data into a 3D array format, including batch size, time steps, and feature size [9]. By utilizing a sliding window, the LSTM model can effectively capture short-term patterns and trends within a small time range at each time step.

The input shape of training and testing datasets are shown in Figure 3. In the training data, we generate sequences suitable for model training by applying a sliding window to all the data. For the testing data, we select the sliding window corresponding to the last recorded RUL in the dataset. The testing RUL dataset takes the values from the expected RUL file. Utilizing the last few time cycles of the engine allows for a more realistic simulation of the model's performance in real-world applications.

	FD001	FD002	FD003	FD004
Train Dataset Shape	(19731, 10, 9)	(51419, 10, 15)	(23820, 10, 12)	(59008, 10, 15)
Train RUL Dataset Shape	(19731,)	(51419,)	(23820,)	(59008,)
Test Dataset Shape	(100, 10, 9)	(259, 10, 15)	(100, 10, 12)	(248, 10, 15)
Test RUL Dataset Shape	(100,)	(259,)	(100,)	(248,)

Figure 3: LSTM - Train & Test Dataset Shape

5.3 Model Build

In the development of our LSTM-based predictive model for estimating the RUL of jet engines, we implement a neural network architecture. The core of our model is an LSTM layer with 128 units, designed to process 3D sequence data. This layer is adept at capturing the temporal dependencies characteristic of time-series data [10]. Following the LSTM layer, a fully connected dense layer transforms the LSTM output into a singular value representing the RUL [11]. Specifically, we utilize the Keras API to build our model, compiling it with the Adam optimizer and mean squared error as the loss function [12]. Each model is trained over 50 epochs with a batch size of 128, ensuring robustness and generalizability of our predictions. The trained models and their corresponding training logs are recorded, providing valuable insights for further analysis and model refinement.

6 Results

6.1 CoxPH Survival Analysis Results and Model Evaluation

We firstly developed CoxPH model to predict the survival function of each engine at the max cycle. We plotted the survival curves for the first 10 engines in each dataset (Figure 4). The x-axis represents the RUL of each engine, and the y-axis refers to the probability of the engine to survive until a specific time point. From the Figure 4, it is observed that the the survival curves drop in the beginning. Until a certain point, the survival curves level off, forming a plateau and exhibiting a low probability of survival.

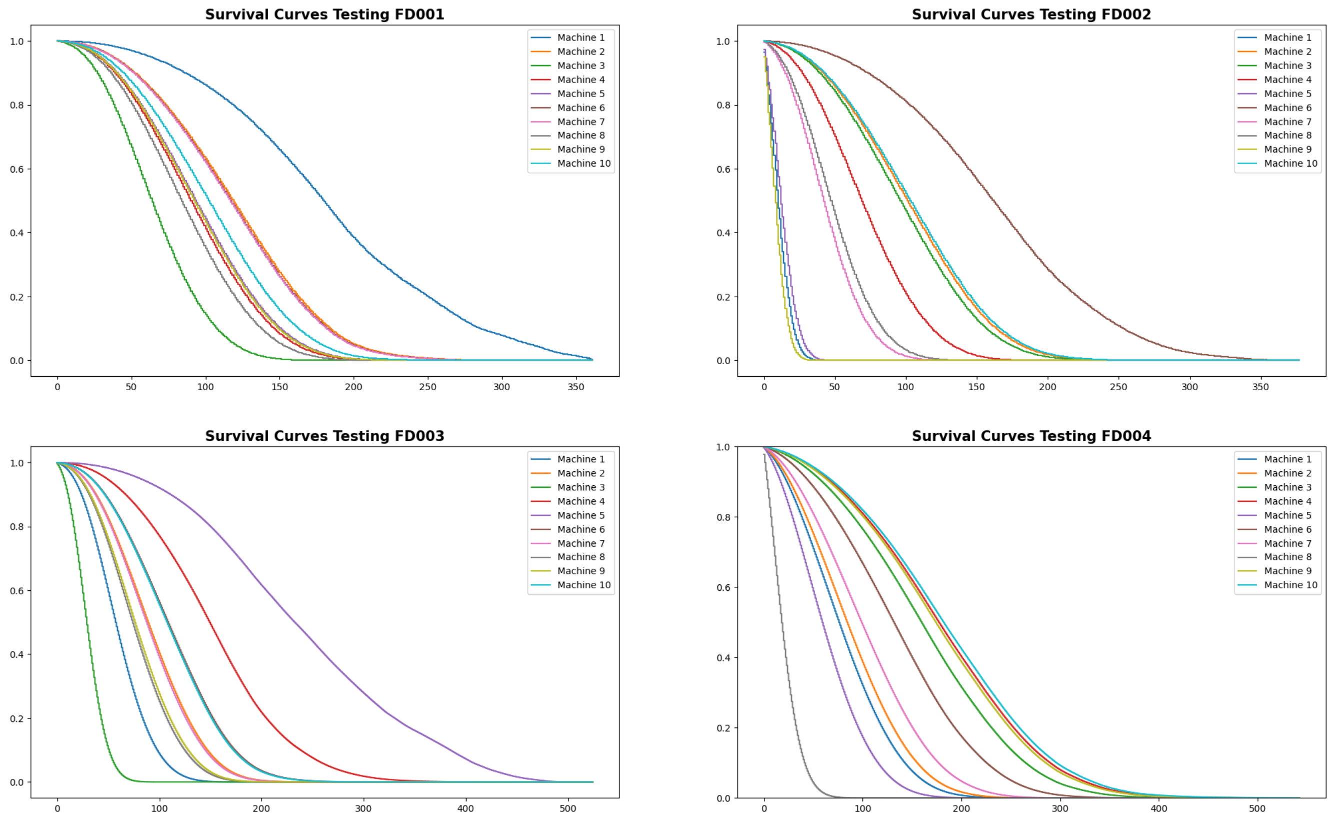

Figure 4: CoxPH Survival Curves: Testing FD001, FD002, FD003 and FD004

To further evaluate our model by assessing the alignment of the predicted survival distribution with the ground truth RUL of the testing datasets, we computed the expectation of the survival distribution of each engine (6). Additionally, we we quantified the alignment by computing the Mean Squared Error using the expectations.

$$E(t) = \int_t^{\infty} S(t)\,dt = \int_t^{\infty} t\,f(t)\,dt \quad \text{where} \quad f(t) \text{ is the Probability Density Function.} \tag{6}$$

Further more, to obtain a more comprehensive understanding of the model performance, we computed the 95 % Confidence Interval of each survival distribution using the following formula (7):

$$CI = (0, t_0) \quad \text{subject to} \quad F(t_0) = P(T \leq t_0) = 0.95 \tag{7}$$

$F(t_0)$ is the cumulative distribution function and $S(t_0) = 1 - F(t_0)$. Therefore, it indicates that there's 95% probability that the engine failure has occurred before t_0. To demonstrate our model performance more directly, the proportion of the testing RUL values that fall within the predicted 95% confidence interval of all the data points of each dataset.

According to the Figure (5), it is evident that the majority of the true RUL values fall within our computed confidence interval. However, However, it is noteworthy that the RMSEs are relatively large when compared to the mean RUL of each dataset. This situation may be attributed to a substantial variance in the predictions.

Datasets	Proportion	MSE with Expectation	RMSE	RUL Mean of the Dataset
FD001	0.99	1346.08	36.69	75.52
FD002	0.97	1223.91	34.98	81.19
FD003	1.0	3348.64	57.87	75.32
FD004	0.99	2467.39	49.67	86.55

Figure 5: CoxPH Model Performance. Proportion: proportion of the testing RUL values that fall within the predicted 95% CI; MSE: Mean Squared Error; RMSE: Root MSE

Overall, while our model performance is moderate, our model shows its unique strength. Instead of predicting a single RUL value, we are able to predict a survival distribution for each object, which is particular valuable. The CoxPH model is designed for survival analysis on the time-to-event data, especially to handle the right-censored data. It has a semi-parametric nature, which provides balance between flexibility and structure and allows to adapt to different data. However, our time-series datasets pose challenges as they are not an ideal match for the CoxPH model. Moreover, the assumptions that the hazard ratios for covariates are constant over time in the CoxPH model, implying a linear relationship between the covariate and the hazard function on the log scale. The linear relationship limits the robustness of the model performance.

6.2 LSTM Prediction and Model Evaluation

In our study, the LSTM model is employed to forecast the RUL of jet engines across four different datasets, labeled FD001 through FD004. The analysis charts depicting 'Difference vs Predicted RUL' reveal the differences between the model's predicted RUL and the actual engine lifespans for the FD001 and FD002 datasets (Figure 6). Despite fluctuations, the 'Difference' line graphs an overall trend that approximately mirrors the engines' actual degradation patterns. Conversely, results for the FD003 and FD004 datasets (Figure 7) varied, with FD003, in particular, showing a lower prediction accuracy with larger difference values, possibly indicating the influence of unique operational conditions or data noise within that dataset. For FD004, the model demonstrates a more robust trend-following ability, reflecting a stronger predictive performance.

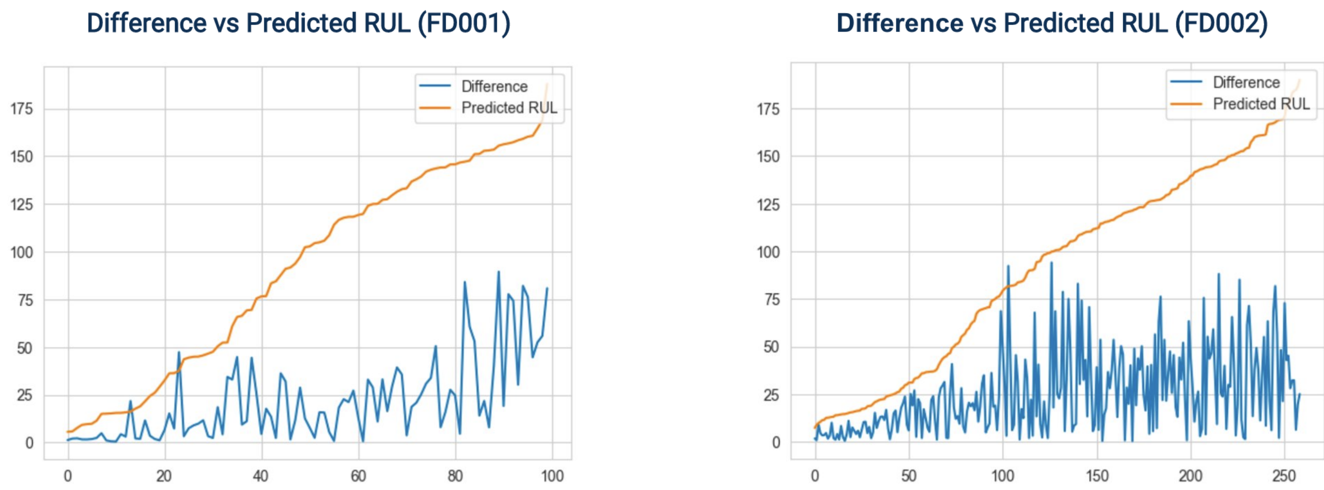

Figure 6: LSTM Prediction: FD001 and FD002

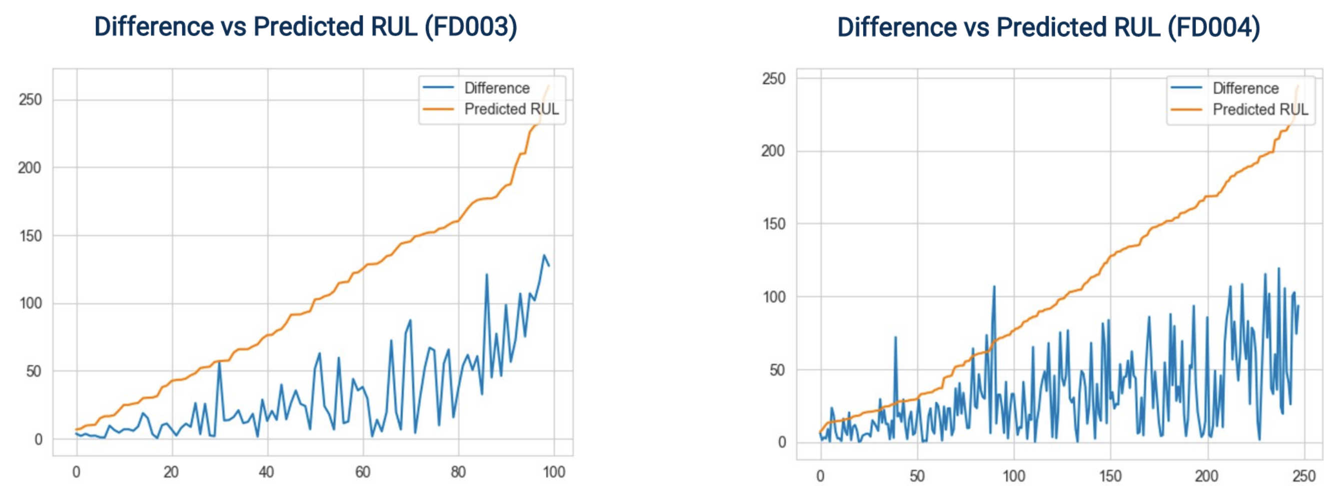

Figure 7: LSTM Prediction: FD003 and FD004

Our model's performance is quantified using Root Mean Square Error (RMSE) and the coefficient of determination (R^2) [13]. The performances of our model and baseline model are shown in the Figure 8. For the FD001 and FD002 datasets, the model achieves RMSE values of 28.943 and 31.585, with R^2 values of 0.515 and 0.655, respectively, indicating moderate explanatory power over the data variability. On the FD003 dataset, the model's RMSE is 36.137 with an R^2 of 0.238, suggesting limited predictive efficacy. However, the FD004 dataset shows improved RMSE and R^2 values of 39.674 and 0.471, respectively, denoting a better fit compared to FD003.

When compare to the baseline performance from the 'Turbo Jet Engine Predictive Maintenance with LSTM' case on Kaggle, our model displays superior or competitive performance on the FD003 and FD004 datasets, notably on FD004 where our R^2 score surpassed the baseline. Overall, these metrics underscore the practicality of our LSTM model in predictive maintenance applications, particularly in its capacity to process and learn from sequential data for RUL prediction.

Our Model Performance

	RMSE	R^2
FD001	28.943	0.515
FD002	31.585	0.655
FD003	36.137	0.238
FD004	39.674	0.471

Baseline Performance

	RMSE	R^2
FD001	28.301	0.536
FD002	30.049	0.688
FD003	40.921	0.023
FD004	42.321	0.397

Figure 8: LSTM Evaluation Metrics

7 Insights and Contributions

In this project, we explore two distinct approaches, CoxPH model and LSTM neural network, for predicting the RUL of jet engines. This dual-method approach allowed us to delve into the strengths and limitations of each model independently, providing valuable insights and contributions to the field of predictive maintenance.

We stretch ourselves beyond the standard homework where you can follow set instructions. Both the CoxPH model and the LSTM neural network are new to us, and we have to put in a lot of extra time to understand how they work. We spend hours reading up on these models, learning how to apply them, and then actually putting them into practice for our jet engine RUL prediction task. We are also grateful for the guidance and support from Professor Zhao, who provides valuable insights and directions.

For the CoxPH model, we have acquired a foundational understanding of the survival analysis model that could be widely applied to various time-to-event data analyses. In our specific application, we utilized the model to predict the survival distributions of the jet engines. Beyond this, CoxPH could also predict hazard function, risk scores and so on, offering comprehensive insights into the survival dynamics of of each object. In addition to the standard CoxPH model, alternative survival analysis models, including gradient boosting survival analysis and random forest survival analysis, offer diverse options for addressing various scenarios. Our project demonstrates a practical example of survival analysis for real-world data.

For the LSTM, we also gain valuable insights and make important contributions. We learn the mechanisms of LSTM and its application conditions where it is adept at handling data that constantly evolves over time. This knowledge is crucial for estimating how long the engines will last. Another important discovery is how the correct data preparation significantly improves the LSTM's predictions. Our work with the LSTM model also makes some valuable contributions. It proves this model can be used in real-life scenarios, particularly in predicting when jet engines require maintenance. This finding is vital for the aerospace industry, where timely engine servicing is essential for safe and efficient airplane operation. Our project shows that LSTM is not just a theoretical concept; it is a practical tool for making critical decisions in fields like aerospace.

8 Conclusion

In conclusion, this research has demonstrated a significant advancement in predictive maintenance strategies for aircraft engines. By employing survival analysis, we have developed a robust method to forecast an aircraft engine's survival function. This approach allows for a more accurate prediction of potential failures, contributing to enhanced safety and reliability.

Additionally, the integration of Long Short-Term Memory (LSTM) neural networks has been crucial in estimating the Remaining Useful Life (RUL) of aircraft engines. These advanced predictive models have proven to be effective in processing and analyzing complex data patterns derived from engine sensors.

The synergy of these methodologies enables the engine maintenance team to accurately determine the most opportune moments for maintenance activities. This is not only critical for preventing unexpected engine failures but also ensures that maintenance efforts are carried out in a cost-effective manner. By analyzing current sensor readings, our approach proactively prevents critical engine damage, thereby safeguarding the engine's operational efficiency and longevity.

In summary, the combination of survival analysis and LSTM neural networks marks a significant step forward in the field of predictive maintenance. This integrated approach paves the way for more reliable, efficient, and safer aircraft engine operations, underlining the importance of continuous innovation in aviation technology.

References

[1] Al Volponi. Data fusion for enhanced aircraft engine prognostics and health management. Technical report, 2005.

[2] David G Kleinbaum, Mitchel Klein, David G Kleinbaum, and Mitchel Klein. The cox proportional hazards model and its characteristics. *Survival analysis: a self-learning text*, pages 97–159, 2012.

[3] Hao Xue, Du Q Huynh, and Mark Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194. IEEE, 2018.

[4] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[5] Miron B Kursa and Witold R Rudnicki. Feature selection with the boruta package. *Journal of statistical software*, 36:1–13, 2010.

[6] Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *arXiv preprint arXiv:1907.00825*, 2019.

[7] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650–655, 2021.

[8] A. Fahim, Q. Tan, M. Mazzi, M. Sahabuddin, B. Naz, and S. Ullah Bazai. Hybrid LSTM Self-Attention Mechanism Model for Forecasting the Reform of Scientific Research in Morocco. *Comput Intell Neurosci*, 2021:6689204, 2021.

[9] Manqing Wang, Liangyu Zhang, Haixiang Yu, Siyu Chen, Xiaomeng Zhang, Yongqing Zhang, and Dongrui Gao. A deep learning network based on cnn and sliding window lstm for spike sorting. *Computers in Biology and Medicine*, 159:106879, 2023. ISSN 0010-4825.

[10] Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I. Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey, 2023.

[11] Luyu Zhou, Chun Zhao, Ning Liu, Xingduo Yao, and Zewei Cheng. Improved lstm-based deep learning model for covid-19 prediction using optimized approach. *Engineering Applications of Artificial Intelligence*, 122:106157, 2023. ISSN 0952-1976.

[12] K.E. ArunKumar, Dinesh V. Kalaga, Ch. Mohan Sai Kumar, Masahiro Kawaji, and Timothy M. Brenza. Comparative analysis of gated recurrent units (gru), long short-term memory (lstm) cells, autoregressive integrated moving average (arima), seasonal autoregressive integrated moving average (sarima) for forecasting covid-19 trends. *Alexandria Engineering Journal*, 61(10):7585–7603, 2022. ISSN 1110-0168.

[13] N. Absar, N. Uddin, M. U. Khandaker, and H. Ullah. The efficacy of deep learning based LSTM model in forecasting the outbreak of contagious diseases. *Infect Dis Model*, 7(1):170–183, Mar 2022.