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

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

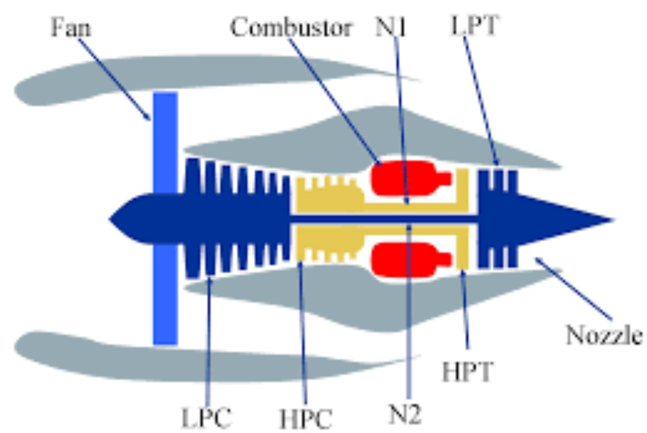
Georgia Tech

Motivations

- **Prognostics and health management (PHM):** crucial field within the aerospace industry, instrumental in predicting the state of aircraft engines to prevent downtime and avoid failures.
- **Leveraging machine learning knowledge:** enable facilitating a predictive maintenance approach based on the current state of the engines and enhances the reliability of aircraft operations.

Dataset

- The dataset employed in this project originates from the public repository managed by **NASA's Prognostics Center of Excellence (CoE)**
- Specifically designed for **asset degradation modelling**, utilizing the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) framework.
- Including four distinct datasets, representing a variety of operational conditions and fault modes.



Objectives

- Utilize machine learning techniques to provide precise prediction of the **survival probability and Remaining Useful Life (RUL)** for specific engine types.
- Analyze the engine's **operational conditions** and interpret data collected from different sensors
- Provide a **proactive maintenance approach** for the industry

Georgia Tech

Data Description

- The dataset from Kaggle includes the training, testing, and RUL data files for 4 different operation condition combination sets.
- In training and testing data files, there are 26 columns, including:

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)

Data Description

- There are 6 operation condition combinations:

Condition 1: Altitude = 0, Mach Number = 0, TRA = 100

Condition 2: Altitude = 10, Mach Number = 0.25, TRA = 100

Condition 3: Altitude = 20, Mach Number = 0.7 TRA = 100

Condition 4: Altitude = 25, Mach Number = 0.62, TRA = 60

Condition 5: Altitude = 35 Mach Number = 0.84, TRA = 100

Condition 6: Altitude = 42, Mach Number = 0.84, TRA = 100

- And the 4 combination sets are:

FD001: Condition 1 only

FD002: Mix of all the conditions

FD003: Condition 1 only

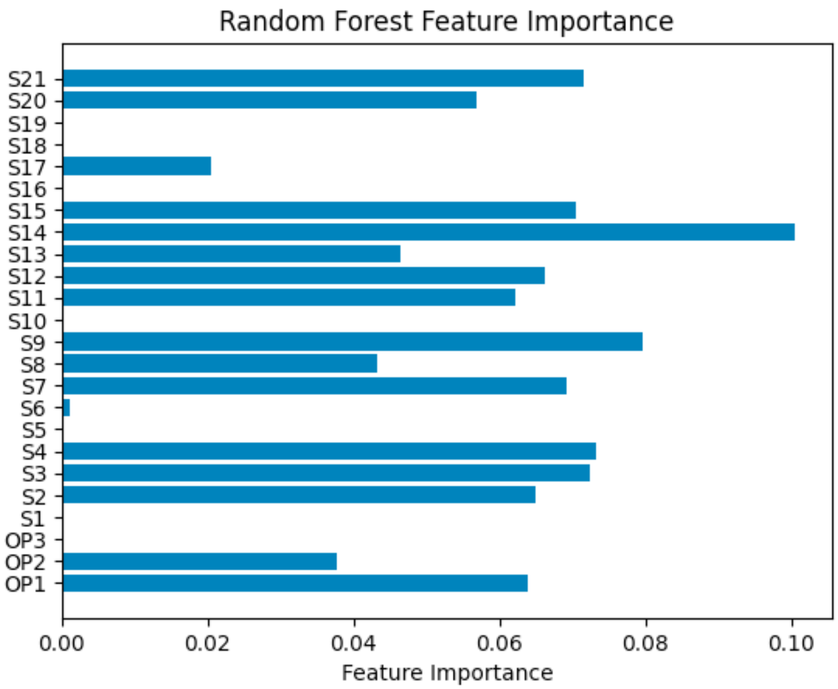
FD004: Mix of all conditions

Data Preprocessing

- Check missing value in dataset
- Feature selection
  - **Variance thresholding:** remove low-variance features, and reduce the dimensionality of the dataset
  - **Random forest:** 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
- Normalization
  - **MinMaxScaler:** transforms features by scaling them to a given range, typically between 0 and 1, but preserve the shape of the original distribution

Georgia Tech

Feature Selection



- FD001(Left Figure):
  'S2', 'S3', 'S4', 'S7', 'S9', 'S11', 'S12', 'S14', 'S20'
- FD002:
  'OP1', 'OP2', 'S2', 'S3' , 'S4', 'S7', 'S8', 'S9', 'S11', 'S12', 'S13', 'S14', 'S15', 'S20', 'S21'
- FD003:
  'S2', 'S3', 'S4', 'S7', 'S8', 'S9','S11', 'S12', 'S13', 'S14', 'S20', 'S21'
- FD004:
  'OP1', 'OP2', 'S2', 'S3' , 'S4', 'S7', 'S8', 'S9', 'S11', 'S12', 'S13', 'S14', 'S15', 'S20', 'S21'

Georgia Tech

Cox Proportional Hazard Model

Georgia Tech

CoxPH Dataset Preprocessing

- Transfer the training dataset current cycle to RUL:
  - $RUL = Max\ cycle\ - Current\ cycle$
- Label all the cycles in training dataset as **true, or observed** (Cox Proportional Hazards Model is not designed to directly handle time-varying covariates or **time series data**).
- The features will be new x, the event indicator (labels), follow-up timepoint (RUL) will be new y
- Transfer the test dataset current cycle to RUL:
  - $RUL = Max\ cycle\ in\ test\ data + RUL\ from\ validation\ - Current\ cycle$
- Treat each machine at **each cycle independently**

Georgia Tech

CoxPH Survival Analysis

- **Survival Analysis** is a statistical method for analysing the length of time required for an event to occur. Survival Analysis is often referred to as **time-to-event analysis**
  - **Time-to-Event Data**: refers to a type of data that measures the time until a specific event of interest occurs. This event could be death in a medical study, failure of a machine component, or a customer unsubscribing from a service.
    - **Time Variable:** This is the primary variable of interest and represents the time until the occurrence of the event
    - **Event Indicator:** A binary variable indicating whether the event of interest has occurred or not
    - **Right censored:** Observations that have not experienced the event by the end of the observation period.
      - Our training datasets -- all the events have occurred. --> no censoring --> Event Indicator = True

CoxPH Survival Analysis

- **Model:** The model is used to estimate the hazard function, which describes how the hazard (instantaneous risk of an event) changes over time.

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

  - **Non-parametric $h_0(t)$**: Baseline Hazard, representing the hazard at time t when all covariates are zero.
  - **Parametric h(t)**: Characterization of how the hazard function changes based on the variables

CoxPH Survival Analysis

- **Objective Function of CoxPH model:**
  - Maximize Partial Likelihood

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

Georgia Tech

CoxPH Survival Analysis

- Input:
  - X -> features
  - Y -> (event indicator, follow-up timepoint)

```
[( True, 148.) ( True, 147.) ( True, 146.) ( True, 145.) ( True, 144.)]
```

- Output:
  - predict(X) -> Risk Scores
  - predict_cumulative_hazard_function(X)
  - predict_survival_function(X)

$$e^{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n}$$

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

$$S(t) = e^{-\int h(t)}dt \qquad h(t) = -\frac{d}{dt} ln(S(t))$$

CoxPH Survival Analysis – Survival Function

- Probability Density Function:
  - *f(t) = P(T= t) (T random variable)*
  - the probability that an event occurs at time t

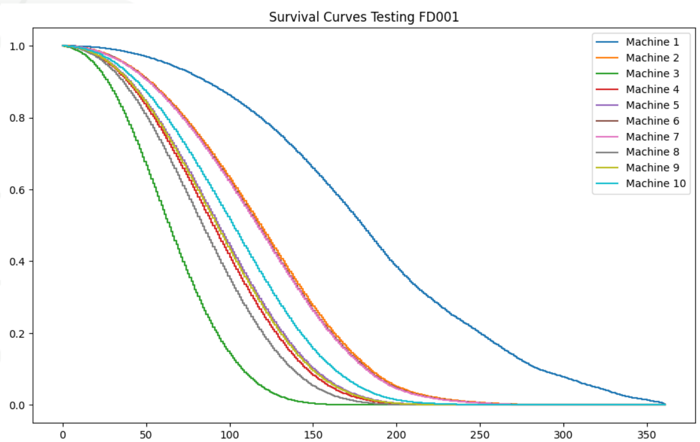
- Cumulative Distribution Function:
  - $F(t)=P(T<=t)$
  - the probability that an event occurs before time t

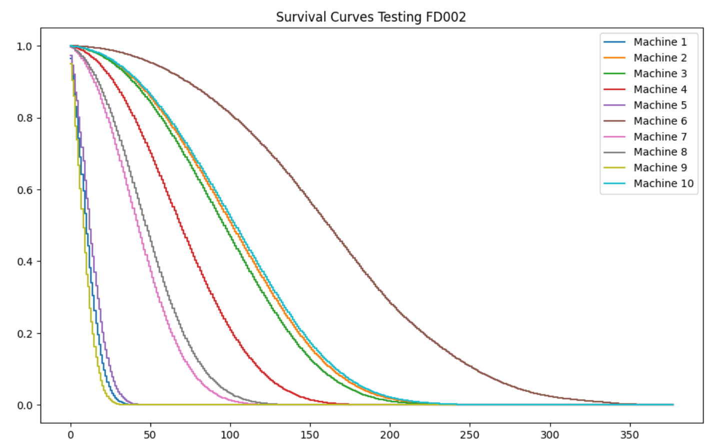
$$F(t) = \int_0^t f(s)ds$$

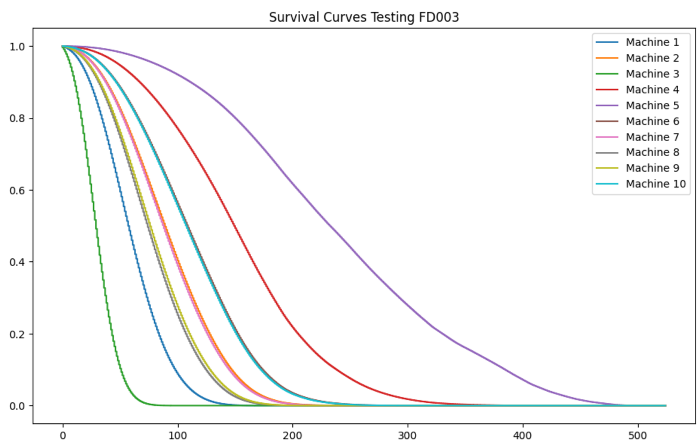
- Survival Distribution Function:
  - $S(t)=P(T>t)$
  - the probability that an event occurs after time t

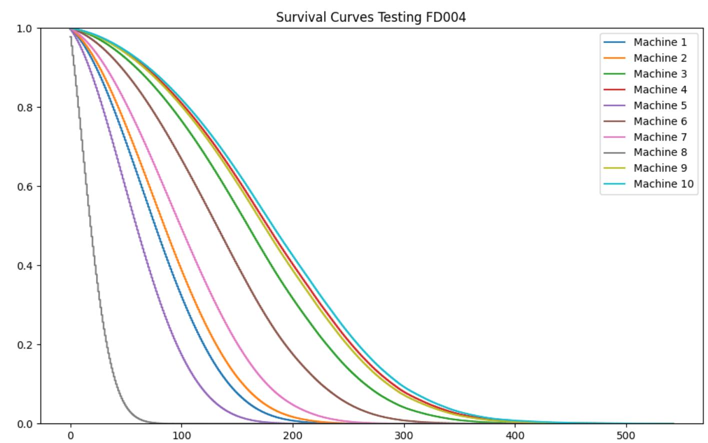
$$S(t) = 1 - F(t)$$

Survival Curves









Georgia Tech

Evaluation of the Model on the testing datasets

- Proportion: the proportion of the testing Remaining Useful Life (RUL) values that fall within the predicted 95% confidence interval of all my data points.
- MSE: MSE of Predicted Expectation VS Testing RUL

$$S(t_0) = 0.05$$

$$CI = (0, t_0)$$

$$E(T) = \int_0^\infty t f_T(t)\, dt = \int_0^\infty S(t)\, dt.$$

Datasets	Proportion	MSE with Expectation	RMSE
FD001	0.99	1346.08	36.69
FD002	0.97	1223.91	34.98
FD003	1.0	3348.64	57.87
FD004	0.99	2467.39	49.67

Georgia Tech

Advantages and Limitations of CoxPH

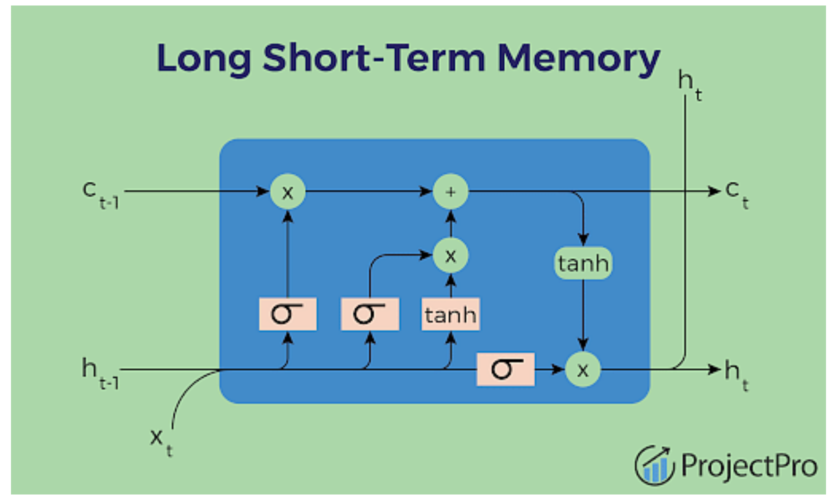
- Advantages
  - Predict Hazard function and Survival function instead of a single value
  - Semi-Parametric Nature:
    - Not making assumptions about the shape of the hazard
    - Allows to adapt to different data patterns
  - Handling Censored Data
    - Our dataset failed to utilize this strength

- Limitations
  - Our dataset is a time-series dataset which is not a perfect fit for the CoxPH.
  - The CoxPH model assumes that the hazard ratios for covariates are constant over time, implying a linear relationship between the covariate and the hazard function on the log scale.
  - We treated the datapoints of the same machine at different cycles independently while they are actually not.

Long Short Term Memory Model

LSTM (Long Short-Term Memory) Introduction

- **Recurrent Neural Network (RNN)** used in deep learning to process and make predictions on sequential data.
- Overcome the limitations of traditional RNNs in handling **long-term dependencies** within data sequences.
- Well-suited for **survival analysis**, model how risk factors evolve over time and how they influence the probability of an event (like **failure or death**) occurring.
- Adaptability in dealing with censored data, accurately predicting **time-to-event outcomes**.

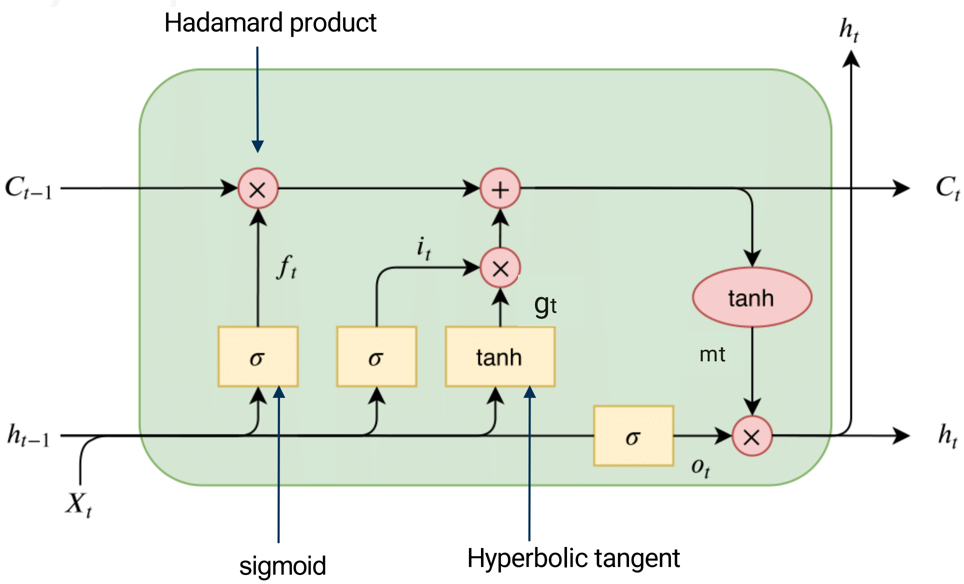


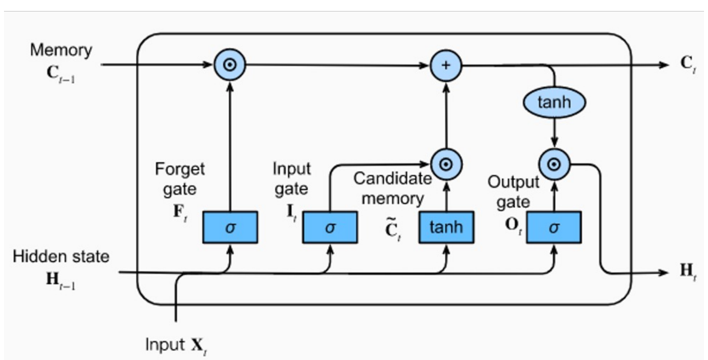
Georgia Tech

LSTM Structures

- **Cell state:** central component of the LSTM, acting as a 'memory' of the network, runs straight down the entire chain of the network, allowing information to flow along it
- **Forget gate:** decides what information should be discarded from the cell state.
- **Input gate:** determines what new information will be added to the cell state.
- **Output gate:** decides what the next hidden state should be, which includes the information that will be passed to the next LSTM cell.
- **Hidden state:** output of the LSTM cell, passed to the next time step and influences the gates in the next cell

LSTM Governing Equations





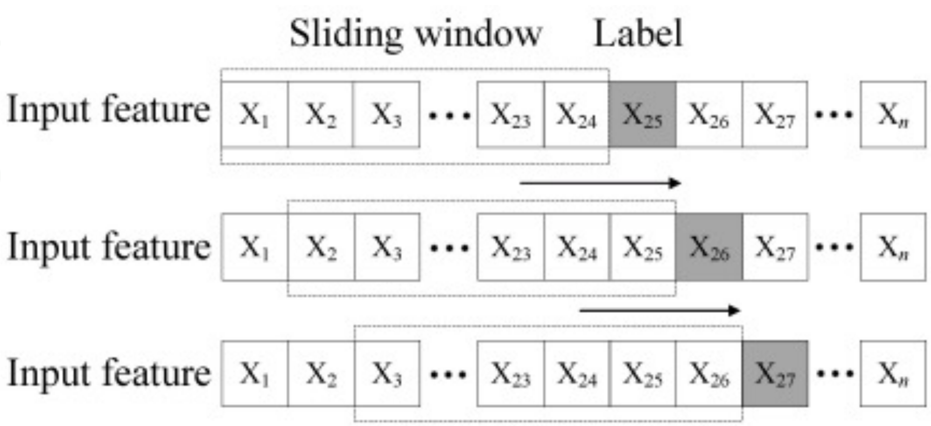
$$
\begin{cases}
i_t = \sigma(\tilde{i}_t) = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t = \sigma(\tilde{f}_t) = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
g_t = \tanh(\tilde{g}_t) = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\
o_t = \sigma(\tilde{o}_t) = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
c_t = c_{t-1} \odot f_t + g_t \odot i_t \\
m_t = \tanh(c_t) \\
h_t = o_t \odot m_t
\end{cases}
$$

LSTM Prediction Procedures

1. Load Dataset & Check Null Values
2. Feature Selection
3. Dataset Normalization & RUL Calculation
4. Creating Sliding Window Sequences
5. LSTM Model & Implementation
6. Training and Validation
7. Prediction
8. Evaluation Metrics
9. Conclusion

4. Creating Sliding Window Sequences

Sliding window prediction principle



Source: https://ieeexplore.ieee.org/document/9366425

Capture short-term dependencies

- Sliding Windows allow the model to capture shorter-term dependencies. At each time step, the model can see a piece of past historical data, which helps to better learn and understand complex patterns in the time series.

Input data shape

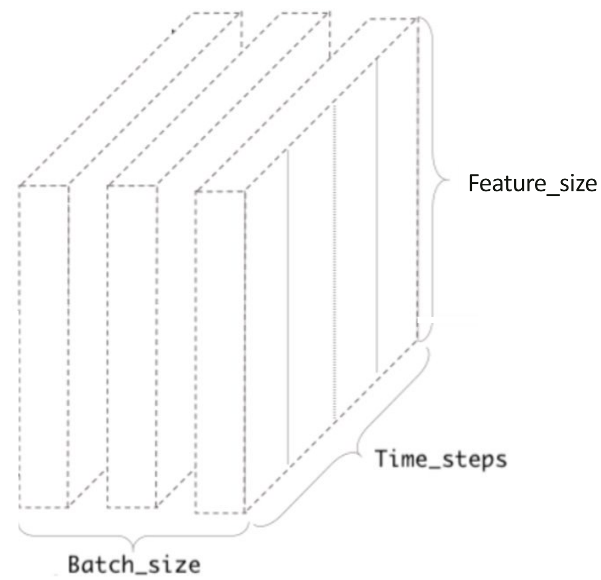
- The time series data can be transformed into a 3D array.(Number of Samples, Time Steps, Number of Features)

```
Train Dataset Shape:  (19731, 10, 9)
```

Georgia Tech

5. Training and Testing Dataset

	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,)



Georgia Tech

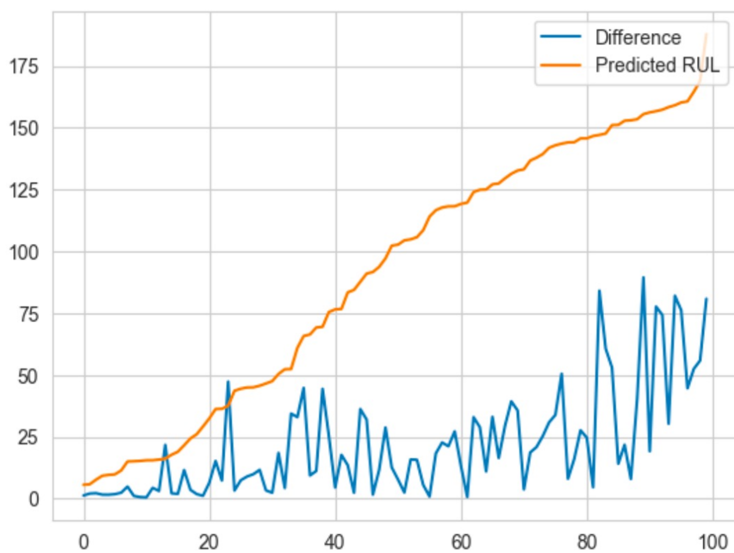
6. LSTM Model Build

- **LSTM Layer**: An LSTM layer with 128 units, the input to the LSTM layer is 2D sequence data.
  - Input shape: (10, 9)

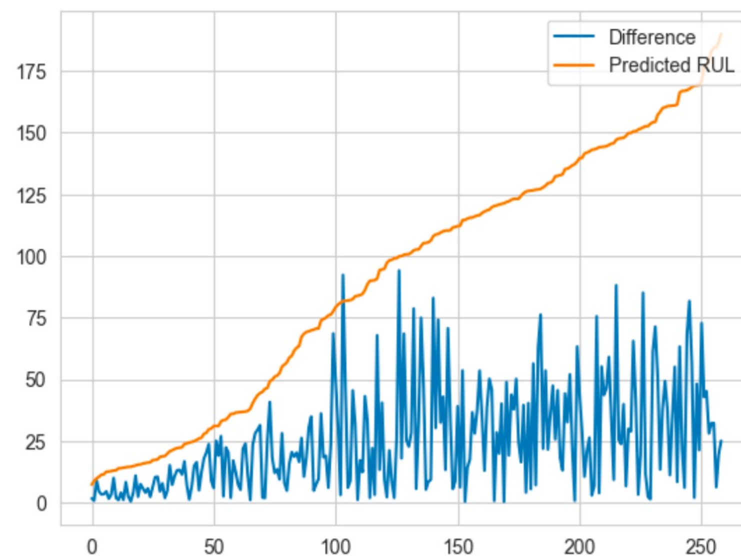
- **Dense Layer** (fully connected): Takes the output of the LSTM layer and converts it into the final prediction.
  - Output: RUL

7. Prediction

Difference vs Predicted RUL (FD001)



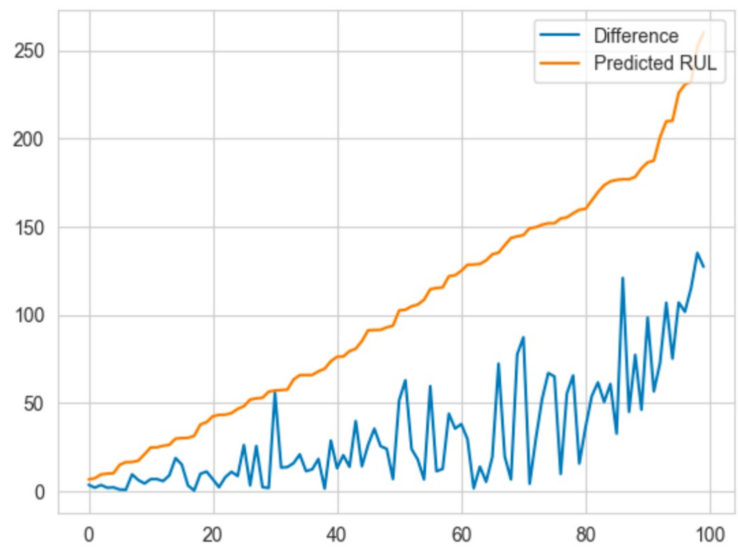
Difference vs Predicted RUL (FD002)



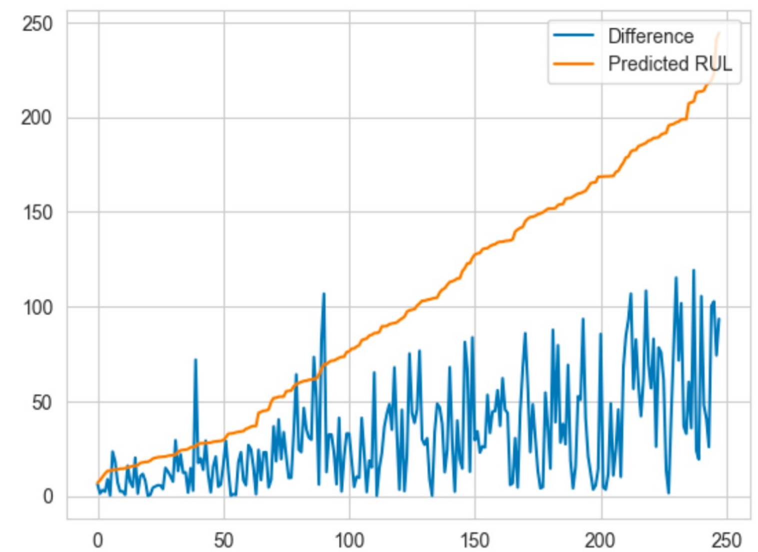
Georgia Tech

7. Prediction (cont.)

Difference vs Predicted RUL (FD003)



Difference vs Predicted RUL (FD004)



Georgia Tech

8. Evaluation Metrics

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 LSTM Performance

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

Source: https://www.kaggle.com/code/chiraggupta7410/turbo-jet-engine-predictive-maintainance-with-lstm/notebook

Georgia Tech

Conclusion

In this project, we:

- Enhance predictive maintenance strategies by employing survival analysis to forecast an **aircraft engine's survival function**.
- Concurrently leverage Long Short-Term Memory (**LSTM**) neural networks to estimate the engine's Remaining Useful Life (**RUL**).
- These advanced predictive models enables the engine maintenance team to accurately determine the **optimal scheduling** for upcoming maintenance activities.
- This is achieved by analyzing current sensor readings, thereby proactively prevent critical engine damage and ensuring operational efficiency.

Reference:

- https://blog.klm.com/jet-engine-propulsion-the-comparison-of-power-between-a-car-and-an-aircraft/
- https://global-engine.com/
- https://gpm.nasa.gov/science-team/resources/nasa-logo
- https://ntrs.nasa.gov/api/citations/20070034949/downloads/20070034949.pdf
- https://blog.klm.com/jet-engine-maintenance-this-is-how-we-do-it/
- https://www.kaggle.com/code/chiraggupta7410/turbo-jet-engine-predictive-maintainance-with-lstm/notebook
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9366425

Georgia Tech

Thank you for listening