PREDICTIVE MAINTENANCE AND PERFORMANCE OPTIMIZATION FOR JET ENGINES BASED ON ROLLS-ROYCE ENGINE MANUFACTURER AND SERVICES WITHIN THE AEROSPACE SECTOR

SITI SYAHIRAH BINTI MOHD YUNUS

UNIVERSITI TEKNOLOGI MALAYSIA

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : SITI SYAHIRAH BINTI MOHD YUNUS

Date of Birth : 10/10/1993

Title : PREDICTIVE MAINTENANCE AND PERFORMANCE OPTIMIZATIONFOR JET ENGINES BASED ON ROLLS-ROYCE ENGINE MANUFACTURER AND SERVICES WITHIN THE AEROSPACE SECTOR

Academic Session : 2024/2025

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
✓	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
MCS241012	
MATRIX NUMBER	**NAME OF SUPERVISOR**
Date: 17 JANUARY 2025	Date:

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my opinion this thesis is suffcient in term of scope and quality for the award of the degree of Master’s degree in Data Science”

Signature	:	______________________________
Name of Supervisor I	:	
Date	:	

Signature	:	______________________________
Name of Supervisor II	:	
Date	:	

Signature	:	______________________________
Name of Supervisor II	:	
Date	:	

BAHAGIAN A - Pengesahan Kerjasama*

Adalah disahkan bahawa projek penyelidikan tesis ini telah dilaksanakan melalui kerjasama antara Click or tap here to enter text. dengan Click or tap here to enter text.

Disahkan oleh:

Tandatangan : Tarikh :

Nama :

Jawatan :

(Cop rasmi)

** Jika penyediaan tesis atau projek melibatkan kerjasama.*

BAHAGIAN B - Untuk Kegunaan Pejabat Sekolah Pengajian Siswazah

Tesis ini telah diperiksa dan diakui oleh:

Nama dan Alamat Pcmeriksa Luar :

Nama dan Alamat Pcmeriksa Dalam :

Nama Penyelia Lain (jika ada) :

Disahkan oleh Timbalan Pendaftar di SPS:

Tandatangan : Tarikh :

Nama :

PREDICTIVE MAINTENANCE AND PERFORMANCE OPTIMIZATIONFOR JET ENGINES BASED ON ROLLS-ROYCE ENGINE MANUFACTURER AND SERVICES WITHIN THE AEROSPACE SECTOR

SITI SYAHIRAH BINTI MOHD YUNUS

A project reportsubmitted in partial fulfilment of the
requirements for the award of
Master's Degree in Data Science

School of Computing
Universiti Teknologi Malaysia

JANUARY 2025

DECLARATION

I declare that this thesis entitled *"Predictive Maintenance and Performance Optimization for Jet Engines Based on Rolls-Royce Engine Manufacturer and Services Within the Aerospace Sector"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	....
Name	:	SITI SYAHIRAH BINTI MOHD YUNUS
Date	:	17 JANUARY 2025

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

I would like to thank Prof. Madya Dr. Mohd Shahizan for his consistent support and guidance during running this project. Prof. continuously provided encouragement and always willing to assist in any way with full of enthusiasm throughout the project. I would also like to thank Asyraf for providing advice regarding analysis. Lastly, to my care assistant Cica, whom I cherished the most.

ABSTRACT

Intelligence maintenance system facilitate the capability to differentiate traditional with modern prognostics, which enable remaining useful life (RUL) of its components. Data driven prognostics model studies involving datasets with trajectories of run-to-failure. Rolls Royce as a safety-critical system in aerospace sectors ensure the failures are rarely to happen. In order to produce a predictive maintenance and performance optimization model with the intention to avoid potently high social cost and significant economic disruptions, dataset from Commercial Modular Aero-Propulsion System Simulation (CMAPSS) model developed by NASA is being used to do the prognostic; RUL of the jet engine. This study focus is to assess past engine data for factors identification that causing engine failures, cultivate predictive models for foreseen problems, and advocate strategies to amplify engine performance and fuel efficiency.

ABSTRAK

Sistem penyelenggaraan perisikan memudahkan keupayaan untuk membezakan tradisional dengan prognostik moden, yang membolehkan baki hayat berguna (RUL) komponennya. Kajian model prognostik dipacu data yang melibatkan set data dengan trajektori run-to-failure. Rolls Royce sebagai sistem kritikal keselamatan dalam sektor aeroangkasa memastikan kegagalan jarang berlaku. Untuk menghasilkan penyelenggaraan ramalan dan model pengoptimuman prestasi dengan niat untuk mengelakkan kos sosial yang tinggi dan gangguan ekonomi yang ketara, set data daripada model Simulasi Sistem Aero-Pendorongan Modular Komersial (CMAPSS) yang dibangunkan oleh NASA sedang digunakan untuk melakukan prognostik; RUL enjin jet. Fokus kajian ini adalah untuk menilai data enjin lepas untuk mengenal pasti faktor yang menyebabkan kegagalan enjin, memupuk model ramalan untuk masalah yang diramalkan, dan menyokong strategi untuk menguatkan prestasi enjin dan kecekapan bahan api.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVR	-	Support Vector Regression
RUL	-	Remaining Unit Life
CMAPSS	-	Commercial Modular Aero-Propulsion System Simulation
NASA	-	National Aeronautics and Space Administration
EDA	-	Explanatory Data Analysis
RMSE	-	Root Mean Square Error
UTM	-	Universiti Teknologi Malaysia
ICAO	-	International Civil Aviation Organization
ANN	-	Artificial Neural Network
ICAO	-	International Civil Aviation Organization
FAA	-	Federal Aviation Administration

LIST OF SYMBOLS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In engine manufacturing and services, Rolls-Royce is a trailblazer; ahead in the field within the aerospace sector and well known as a pioneer in the area of advancements in technology. With the aim of pledging the trustworthiness and successfulness of contemporary jet engines, predictive maintenance must be implemented. This is a data-driven strategy to predict potential failures in advanced that being facilitate by manufacturers and service providers.

Rolls-Royce have developed TotalCare program to exhibit the magnitude of predictive maintenance by providing proactive maintenance and live monitoring for their airplane engines, conclusively reducing operational expenses and airlines' downtime in the long run.

Predictive maintenance is becoming crucial from time to time to boost the jet engine's efficiency as to comply with the augment needs for fuel efficiency, sustainability in aviation industry and lower operational expense. Engine malfunctions prediction scale and detecting performance reduce in advanced can minimizing unanticipated maintenance expenses. Else, this can helps with reducing the fuel economy which is a crucial aspect for business that prioritize carbon reduction target.

The aims of this project is to utilize extensive datasets of jet engine performance and sensor data to generate predictive maintenance models and approach for optimizing performance. This project focus is to assess past engine data for factors identification that causing engine failures, cultivate predictive models for foreseen problems, and advocate strategies to amplify engine performance and fuel efficiency.

This research's significance is within the capability to lower aviation's environmental outcome while refining aircraft efficiency. This project seeks assistance for companies such as Rolls-Royce by improving the efficiency, sustainability, and reliability of their engine systems with collaborating predictive analytics and machine learning.

In conclusion, this project will apply data-driven insights to boost performance and fulfil maintenance needs, supporting operational improvements while also foster Rolls-Royce's innovation and sustainability goal beside conveying the aviation industry’s prospective challenges.

1.2 Problem Background

The aerospace industry encounter a major hurdle due to the costly nature of sudden maintenance. Unscheduled maintenance causing major disruptions, resulting in expensive repairs and lengthen downtime that becoming an operation’s bottlenecks as aircraft operators has their own schedule on aircraft engines for their daily activities. It is crucial for engine makers such as Rolls-Royce to minimize these incidents from occurs to guarantee airlines' operational efficiency and cost-effectiveness. Besides, unforeseen engine breakdown has various impacts. This includes flight schedule, airline revenue, customer contentment, and overall fleet supervision. Failure to detect engine wear and tear beforehand consequence in costly and dangerous incidents while in mid-air operation. By endorsing predictive maintenance, possible malfunction can be avoid in advance, hence reducing the possibility of unexpected incidents and improve maintenance schedule.

As climate change has become a vital in environment concern, the aviation sector encounter such increasing demand to reduce its environmental footprint particularly its carbon emissions. Jet engines are a main contributor to fuel consumption, and even little inefficiencies in engine performance can cause in significant rises in fuel usage and carbon emissions. Rolls-Royce which is a top player in the aerospace sector, is dedicated to improve the fuel efficiency and reducing carbon emissions of its engines. Nevertheless, enhancing engine efficiency is complex and continuous evaluation of various factors including flight environments, operating conditions and engine degradation are necessary. Maintenance's forecast, together with methods to magnify performance, could identify the opportunities to improve engine efficiency, resulting fuel and emission reductions align with Rolls-Royce's sustainability goals.

Modern jet engines are embedded with vast amount of sensors that producing abundance of data about engine performance, pressure, vibration, temperature, and other operational variables. Moreover, many companies still enduring obstacle in fully utilizing this data for real-time live monitoring and decision-making process. Basically, Rolls-Royce's TotalCare program provide advanced monitoring of engine performance, other aircrafts still do not receive the most from the maximal potential of real-time data analytics. Generally, data is not processed at the speed of light hence it is not easy to make proactive decisions, leading to inadequacy in identifying issues beforehand. The capability to integrate machine learning models with real-time sensor data with the purpose to predict engine failures or recognizing performance degradation may enhance the effectiveness of predictive maintenance, reduce downtime, and improve overall operational efficiency.

1.3 Problem Statement

Unplanned maintenance continues to be a notable issue in the aerospace sector, causing considerable disruptions to operations and increased expenses for airlines. Aircraft engines, especially those utilized in commercial air travel, are intricate systems that need constant monitoring to guarantee they operate at their best. Unforeseen engine failures resulting frequent and costly repairs by various airlines and operators although engine technology has been evolving from time to times. These aircraft breakdown may lead to flights delay, high cost of urgent fixes, and the worse is aircraft being out of service hence increasing overall operational expenses. By using low capability predictive maintenance system, it is restricting early detection of possible failures before they have becoming worsen. Creating a capable predictive maintenance system is the goal for this study as the system can predict engine failures through the collection of sensor data with the aim to reduce unscheduled maintenance costs.

Enhancing the fuel efficiency of airplane engines has become a top priority as the aviation sector faces growing demands to lessen its impact on the environment. Carbon emissions and operational expenses will increase even with the slightest engine inefficiencies whereby it consumes more fuel usage. Enhancing engine efficiency to minimize fuel consumption has been an absolute way to achieve sustainability goal hence engine makers like Rolls-Royce are collaborating with the airlines. Moreover, reducing the fuel consumption is a difficult task because there are various factors that directly impact engine efficiency, such as operational configurations, engine degradation plus environmental conditions. Good data analytics skills are much needed for discovering performance decline thus procedure to enhance the engine configurations for optimizing fuel efficiency can be considered.

Various sensors in modern jet engines have potential to produce a fast real-time data on larger scale. This included crucial engine variables like temperature, vibration, pressure, and fuel usage. Generally, major difficulties in fully utilizing the potential of this data are still encountered by aerospace industries until today. Frequently in industry practices, sensor data are not being evaluate in an instant lead to the delays in decision-making hence quick interventions are failed to make. Consequently, problems such as decreased engine performance, decreased efficiency, and upcoming failures might go unnoticed until it is too late. Real-time monitoring and predictive maintenance services has been addressed by Rolls-Royce's TotalCare program for utilization of sensor data up to the full potential for proactive interventions. Aim of this project is to fill this gap through a predictive maintenance system that uses real-time data from jet engines to predict decline in performance for a better maintenance timing so that operational effectiveness can be leverage.

1.4 Research Question

1. How can good predictive systems improve the schedule for maintenance timing for the purpose of minimizing downtime while maximizing engine efficiency?

2. How data exploration methods able to uncover the patterns in indicating engine wear and tear from a past data monitoring system?

3. How interactive dashboard being designed to show actionable insights in an easy way to understand for operational management and maintenance team?

1.5 Research Gap

The research gap is as below:

1. Lack of frameworks that integrate real-time IoT data streams with predictive models for volatile maintenance schedule.

2. Has limited focus on domain-specific parameters such as thermal stress, engine turbofan, engine vibration signatures and wear patterns.

3. The unavailable systems that enable cross-functional collaboration and prompt-decision making in time based on maintenance insights.

4. Lack of Real time prognostic dataset from NASA and engine manufacturer as this is totally private assets.

1.6 Aim of the Study

The aim of this project is to create a predictive maintenance system for jet engines. This can be done by harnessing large-scale datasets from engine performance and sensor information with the intention to forecast possible failures besides improving maintenance schedule and boost up overall engine performance. This study aims to apply good data analytics method and machine learning models to figure out early signs of engine wear and tear which lead to less unexpected maintenance expenses. Besides, this could improve the sustainability and fuel efficiency of contemporary jet engines. This study will take the insights on how past data monitoring systems can be beneficial for continuous improvement in the future of engine performance, leading to lower operational costs and lessen environmental impact in the aerospace sector.

1.7 Objective of the Study

The proposed project aims to achieve the following objectives:

1. To bring forth predictive maintenance system for jet engines by utilizing extensive sets of engine performance data and sensor information to the full potential for predicting possible engine malfunction, enhance maintenance plans to increase engine effectiveness.

2. To investigate exploratory data analysis on real data monitoring systems that can be used for continuously improve engine performance, oversee early signs of engine wear and tear, reduce unscheduled maintenance expenses, and revamp the eco-friendliness and fuel economy for contemporary jet engines throughout the application of inventive data analytics methods and machine learning models.

3. To conduct comprehensive evaluations on the developed predictive model and build and interactive dashboard.

1.8 Scope of the Research

The Scope of this study "Predictive Maintenance and Performance Optimization for Jet Engines Based on Rolls-Royce Engine Manufacturer and Services Within the Aerospace Sector" is extensive, multifaceted to meet the unique challenges. The study focus on data collection and preprocessing by using publicly accessible and simulated obstacles in jet engine performance and sensor dataset around 5 figures data points. Task during preprocessing is involving managing missing data, cleaning raw data, and ensure the dataset is ready for analysis. Additionally, Exploratory Data Analysis (EDA) method is applicable for analysis of engine performance trend, obtaining all the parameters such as pressure, temperature, vibration and fuel consumption. The purpose is to detect the patterns and irregularities which signals possible decline in engine performance following maintenance requirements. The project further delves into the predictive modelling which consist of development and evaluation of machine learning model (e.g., regression, classification, or time-series models) to predicting engine failures rate or performance degradation. Both machine learning approach and traditional statistical techniques are being compares to identify the most effective predictive maintenance solution. Consecutively, strategies development to improve operational efficiency and further lessen carbon emissions through performance optimization techniques being made after validation and evaluation of these model to ensure robustness and reliable. The insights derived from these predictive models will be translated into data visualization; dashboard using tools like Tableau or power BI to visualize predictive maintenance outcomes and performance optimization insights. Results are presented in user-friendly format for conveying actionable information for stakeholders, aiming to assist manufacturing operations, customer, environmentalist and policymakers in making informed decisions. The project's scope also encompasses the acknowledgement of limitation and ethical considerations from using simulated and publicly available datasets. Lastly, the study scope's exploration is applicable to Rolls-Royce's *TotalCare program* identical predictive maintenance series in the aerospace sector. The analysis could maximize the efficiency and sustainability goals of aviation companies.

1.9 Expected Contribution of the Study

1. Produce a strong framework for predictive maintenance to foresee engine malfunctions in advanced.

2. Harnessing Big Data and Machine Learning: Leverage vast datasets and exploring analytics methods are vital for the project.

3. Optimization and Cost Reduction: Focus on minimizing unscheduled maintenance and improving fuel efficiency.

4. Environmental sustainability: Certify the project comply with encompassing goals to reduce carbon emissions.

5. Monitoring in real-time: Consolidation of the systems for monitoring engine performance continuous improvement.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss the related issues and the previous studies that have been done. Aerospace engineering is a fast pace, high evolving environment and the critical factor for operational excellence and safety is predictive maintenance. By applying the comprehension of machine learning, big data analytics and the Internet of Things (IoT), predictive maintenance systems has lead the traditional maintenance paradigm into data-driven approach and proactive solution. These technologies are being used to allow jet engine's real-time monitoring, maintenance schedules optimization, enable potential fault's early detection and extending engine lifespan.

As a pioneer in implementing predictive maintenance technologies, Rolls-Royce lead the innovation by using "Power by the Hour" service model that applying IoT-enabled systems and digital twins technology. These technologies derive the benefits from sensor information and engine performance data to predict malfunctions, optimize maintenance schedules and extending engine life. Despite these advancements, the challenges remains significance where real-time adaptability, scalability and multi-sensor fusion frameworks development are volatile.

This literature review synthesizes on recent advancements in the areas of predictive maintenance for jet engines, riveting on the use of IoT, data analytics and

AI to helps in optimizing maintenance schedules planning and the reliability of the system. The review draw a parallel with existing gaps and renders a roadmap for thriving comprehensive, real-time predictive systems.

2.2 Synthesis of existing Studies and Gap Identification

Technological foundations for predictive maintenance will emphasize on IoT-based real time data monitoring, role of big data and cloud platform, as well as machine learning models for predictive analysis.

2.2.1 IoT-Based Real-Time Data Monitoring

IoT has made far-reaching changes in predictive maintenance by perpetual monitoring of critical components. Concurrent data input from IoT sensors such as vibration, pressure readings, temperature helps to provide the foundation of anomaly analysis and fault detection. Observation has been made, that shows low latency data transmission is crucial during continuous jet engine's health monitoring based on IoT-based frameworks exploration (S. Nasir et al, 2022). The challenge is to achieve good performance in extreme operating conditions consistently. IoT-enabled fault detection system has been proposed by the leverages of pressure and vibration sensors and coupled with unsupervised learning models. The feasibility of early stages of fault detection being demonstrates but facing real-time deployment challenges (L. Zhang et al, 2019)

2.2.2 Big Data and Cloud Platforms

Big data analytics and cloud platforms integration has enhances the capacity significantly in the progress of processing huge amounts of operational data from jet engines. (R. Mohanty et al, 2021). The absolute needs for hybrid architectures being emphasizes from the edge solutions and cloud computing trade-offs. By using cloud-based IoT systems, scalable data pipeline for concurrent-time fault prediction being develops and the theoretical models provided (D. Lee et al, 2018). However, the practical implementation for large-scale aerospace real applications is lacking.

2.2.3 Machine Learning Models for Predictive Analysis

Predictive Models and Algorithms

Machine learning is a powerful tools that became a benchmark for predictive maintenance with the purpose of analysing patterns in engine performance data. Previously Random Forests which is a supervised learning models being implemented The purpose is to predict engine faults (R. Mohanty et al, 2021). Random Forests has showing the capability to demonstrates high accuracy for historical data while struggling with real-time implementation.

Besides, a study on anomaly detection with the focus on unsupervised learning techniques including Support Vector Regression (SVR) and neural networks helps in identifying the rare fault pattern (L. Zhang et al, 2019). Scalability challenges in multi-engines environments and has been highlighted in Fault Diagnosis in Jet Engines.

Digital Twin Integration

Digital twins system integration has creating the chances to improve predictive systems by continuous optimization stimulated by the real-time jet engines state in applications. Performance optimization and concurrent fault prediction in aircraft engine health monitoring by using digital twins system showing a strong theoretical framework, only the real-world application is lacking in real world applications (P. Li et al., 2020).

CMAPPS Model

Theoretically, The CMAPSS system model has the nonlinear equations as a form of a coupled system. The inputs of the system model are divided into scenario descriptor operating conditions w and unobservable model health parameters 0. The outputs of the system model are estimates of the measured physical properties s and unobserved properties x, that are not part of the condition monitoring signals (i.e., virtual sensors). The nonlinear system model is denoted as:

$$[x_s^{(t)}, x_v^{(t)}] = F(w^{(t)}, \theta^{(t)})$$

The unobservable model health parameters 0 are model tuners and fall in the class referred to as quality parameters (i.e., component efficiencies, flow, input scalars, output scalars, a n d / o r adders). These model parameters are used to simulate the deteriorated behaviour of the system. Concretely, all the rotating sub-components of the engine i.e., fan, low pressure compressor (LPC), high pressure compressor (HPC), low pressure turbine (LPT) and high pressure turbine (HPT) can be affected by degradation in flow and efficiency as shown in Figure 2.1 and Table 2.1. (Wu Ju et al., 2020)

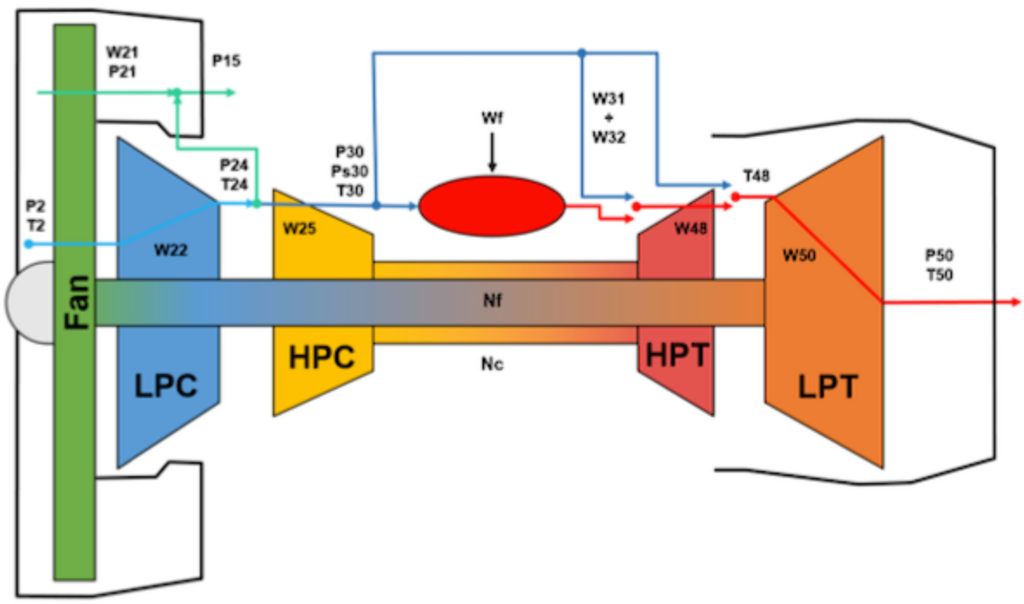

Figure 2.1 Schematic representation of the CMAPSS model as depicted in the CMAPSS documentation (Frederick et al., 2020)

A **low pressure compressor (LPC)** and **high pressure compressor (HPC)** supply compressed high temperature, high pressure gases to the combustor. **Low pressure turbine (LPT)** can decelerate and pressurize air to improve the chemical energy conversion efficiency of aviation kerosene. **High pressure turbines (HPT)** generate mechanical energy by using high temperature and high pressure gas strike turbine blades. **Low-pressure rotor (N1), high-pressure rotor (N2), and nozzle** guarantee the combustion efficiency of the engine.

#	Symbol	Description	Units
1	fan_eff_mod	Fan efficiency modifier	-
2	fan_flow_mod	Fan flow modifier	-
3	LPC_eff_mod	LPC efficiency modifier	-
4	LPC_flow_mod	LPC flow modifier	-
5	HPC_eff_mod	HPC efficiency modifier	-
6	HPC_flow_mod	HPC flow modifier	-
7	HPT_eff_mod	HPT efficiency modifier	-
8	HPT_flow_mod	HPT flow modifier	-
9	LPT_eff_mod	LPT efficiency modifier	-
10	LPT_flow_mod	HPT flow modifier	-

Table 2.1 Model Health Parameter

2.3 Research Methodology and Gaps

The IoT Framework design is the development of a system architecture which combining cloud computing for processing and IoT sensors for data collection as presented in Real-Time Monitoring for Predictive Maintenance in Aerospace (S. Nasir et al., 2022) This study identifying the faults by using real-data streams sourcing from engine-mounted sensors. The data handling which involving mass-volume sensor data are being centralized at cloud system and early detection of anomalies are based on continuous monitoring. Validation purpose involving testing fault detection performance by using simulated engine data, besides fault detection accuracy and latency reduction as primary performance metrics.

Hybrid architecture that merging cloud computing for long-term storage and real-time analytics from edge computing is being implemented by data pipeline development. Data processing in cloud systems storing data that is historical for trend identification purpose plus more complex analysis being conducted as presented by Real-Time IoT Data Processing for Predictive Maintenance in Aerospace Applications (D. Lee et al., 2018). Simulated jet engine data helps in validating the framework while fault detection accuracy and latency aid in assessing the performance.

High-frequency sensor data collection have its own data processing pipeline by using hybrid IoT-cloud framework. The machine learning models involving fault detection models by applying Random Forest and SVM Big Data Analytics as supervised learning models. These models are focused on data pre-processing for the purpose of handling noise and missing values. Predictive Maintenance in Aircraft Engines (R. Mohanty et al., 2021). Case studies were conducted on the scenarios of simulated engine wear. Metrics such as precision, recall, classification accuracy were used to evaluate model performance.

Based on article Fault Diagnosis in Jet Engines Using IoT-Based Sensors (L. Zhang et al., 2019), engine components producing acoustic signals and vibration; these two were collected for IoT sensors deployment. The methodology of using unsupervised machine learning involving anomaly detection by applying clustering techniques (e.g., k-means) plus detecting novel fault patterns by endorsing multi-dimensional sensor data. To validate this study, isolated engine components were tested and accuracy for anomaly detection besides interpretability being a focus points.

Digital Twins for Aircraft Engine Health Monitoring (P. Li et al., 2020) presenting Digital twin design by creating real-time sensor data of virtual replicas for jet engines and endorsing AI-driven predictions with physics-based modelling for scenario simulation. For data integration purpose, multiple failure conditions being stimulate by ensure digital twin and real time operational data are updated continuously. This simulations are labelled as lack of real-world data application as it were tested under controlled scenarios. The study aid in predictive accuracy emphasizing which have the ability to simulate the conditions for future engines.

```
[ IoT Sensors ] ----> [ Edge Computing ] ----> [ Cloud Processing ]
      |              |                |
      |              |                |
 [ Real-Time Data ] ----> [ Predictive Models ] ----> [ Dashboard Visualization ]
```

Predictive Maintenance Architecture

Article Title	Theme	Key Summary
Real-Time Monitoring for Predictive Maintenance in Aerospace (S. Nasir et al., 2022)	IoT-Based Data Monitoring	Real-time IoT systems for engine health monitoring.
IoT and Big Data Analytics for Aircraft Engines (R. Mohanty et al. 2021)	Big Data and Machine Learning Success stories on predictive maintenance	Success stories on predictive maintenance with big data.
Digital Twins for Aircraft Engine Health (P. Li et al. 2020)	Digital Twin Technology	Virtual replicas for operational optimization.
Fault Diagnosis Using IoT-Based Sensors (L. Zhang et al., 2019)	Fault Detection and Diagnostics	Innovative fault detection algorithms leveraging IoT.
Real-Time IoT Data Processing for Predictive Maintenance in Aerospace Applications (D. Lee et al. 2018)	IoT and Cloud Platforms	Framework for cloud-based predictive systems.

Table 2.2 Insights from articles

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This section explains the methodology and applications that will be used in the research. Before getting to the data gathering, this chapter will examine the data description. The data must first be examined in order to accomplish the research's goal. The procedure would begin with gathering and evaluating raw data from many websites, such as NASA's Open Data Portal, UC Irvine Machine Learning Repository and ICAO Aircraft Engine Emissions Databank.

The predictive maintenance is a method which is used in order to determine the condition of in-service equipment for the purpose of determining when the maintenance should be carried out by using the data from turbofan jet engine. A turbofan or fanjet is a type of airbreathing jet engine that is widely used in aircraft propulsion. This approach has the potential of reducing costs over routine or time based preventive maintenance since here the tasks are only done when necessary. Throughout discussion of the research findings and methodology will be covered in this chapter.

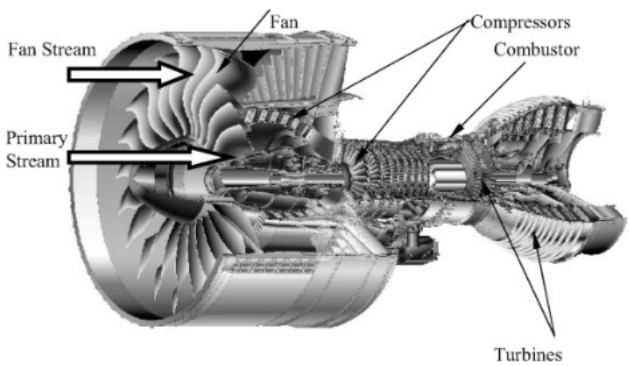

Figure 3.1 Turbofan engine

3.2 The Framework

I. Problem Formulation
II. Data Collection
III. Data Pre-processing
IV. Modelling
V. Testing and Validation
VI. Performance Evaluation

The details of the research framework for this study are shown in the Figure below.

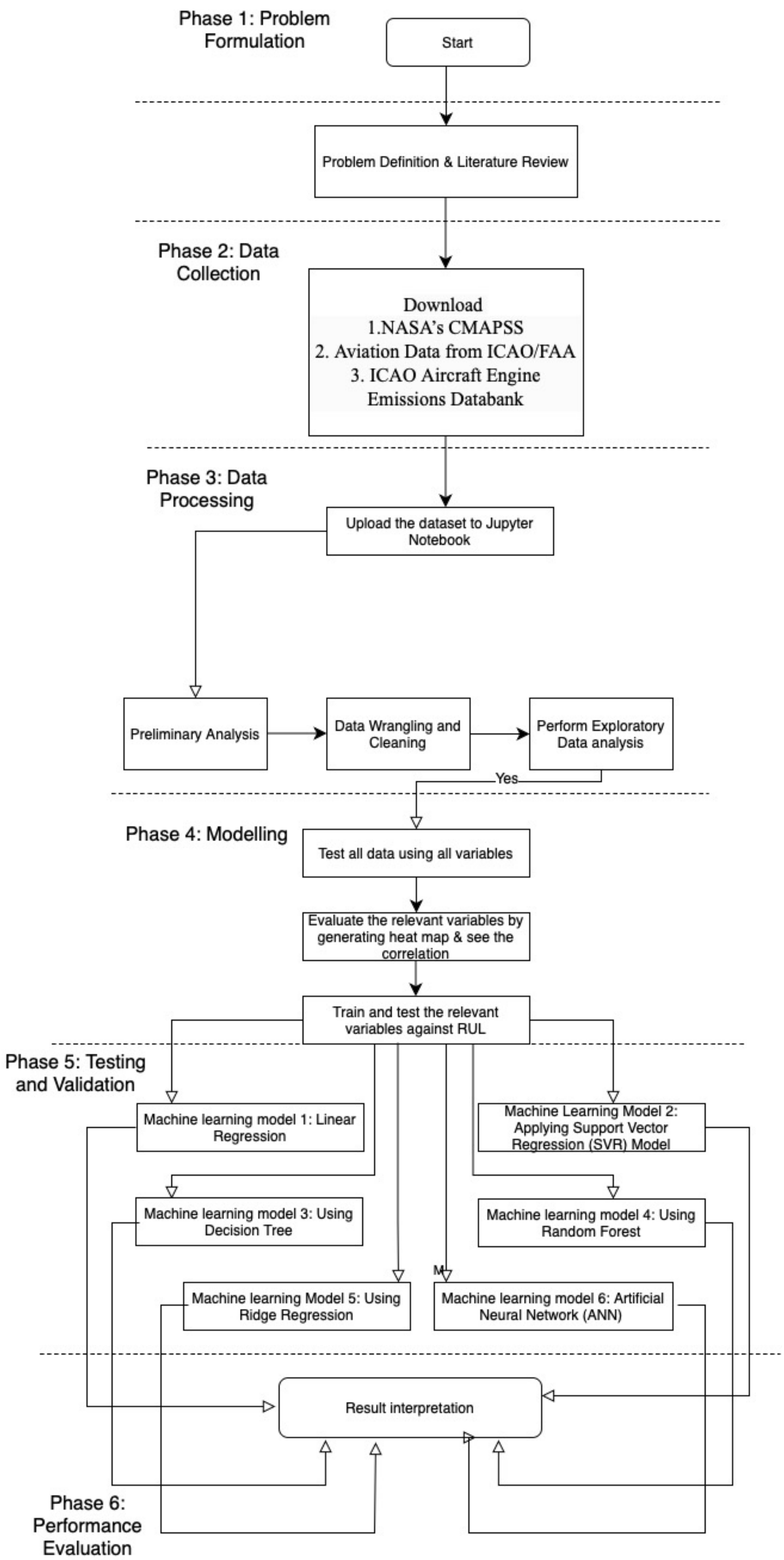

Figure 3.2: Research Framework

Building Model

(a) Model 1: Linear Regression

(b) Model 2: Applying Support Vector Regression (SVR) Model

(c) Model 3: Using Decision Tree

(d) Model 4: Using Random Forest

(e) Model 5: Using Ridge Regression

(f) Model 6: Neural Network Model

3.3 Problem Formulation

The principal objective for this research is to employ descriptive prognostic and health management of aircraft engine for predicting the conditions of the assets in order to avoid downtime and failures plus improving the predictive maintenance schedules. Predictive maintenance on NASA'S turbofan engine degradation dataset (CMAPSS) will be used to run the machine learning. The machine learning will be apply to perform a predictive RUL (Remaining Useful Life of Engine) by applying various ML Model on FD001 dataset. This dataset is the least complex and the first in the series.

3.4 Data Collection

The data collection framework are as follow:

```
jet-engine-project/
│
├── data/
│   ├── cmapss_data.csv
│   ├── icao_faa_data.csv
│   ├── simulated_iot_data.csv
│
├── notebooks/
```

```
│   ├── EDA.ipynb
│   ├── cmapss_modeling.ipynb
│   ├── icao_optimization.ipynb
│   ├── iot_anomaly_detection.ipynb
│
```

Figure 3.3: The data collection frameworks.

Data Set	Train Trajectories	Test Trajectories	Conditions	Fault Modes
FD001	100	100	ONE (Sea Level)	ONE (HPC Degradation)
FD002	260	259	SIX	ONE (HPC Degradation)
FD003	100	100	ONE (Sea Level)	TWO (HPC Degradation, Fan Degradation)
FD004	248	249	SIX	TWO (HPC Degradation, Fan Degradation)

Table 3.1 Data Set Organization

Predictive maintenance on NASA's turbofan engine degradation dataset (CMAPSS) are selected to assist in achieving the objectives. The data is obtained from official website of NASA's Open Source Data. Dataset FD001 contains :

a) Train trajectories: 100 engines.
b) Test trajectories: 100 engines.
c) Fault Modes: ONE.

Datasets consist of simulations from multiple turbofan engines over period of time, each row contains the following information:

1. Engine unit number
2. Time, in cycles
3. Three operational settings
4. Readings from 21 sensors

There is no additional information regarding the sensors has been provided. Additionally, if there are some information regarding sensor type, for example vibration sensor, pressure sensor , temperature sensor etc., only then we are able to grab more information about degradation of engine by using domain knowledge.

3.5 Data Pre-Processing

Preliminary analysis is crucial prior to pre-processing processes. Understanding view of dataset chosen to be used in the study is necessary to have the firm grasp of all the available features in the dataset and how it will use following the creativity to fully utilize it.

Data Pre-processing	Purpose
Preliminary analysis	To evaluate the provided dataset and obtain insightful knowledge for the modelling phase that follows.
Data Cleaning	Find the missing value, identify the outliers and eliminate the rows that does not contain missing value
Data Concatenation	Compile every CSV file from January 2008 to the latest file by NASA; 16 May 2024
Data Visualization	Plotted a chart illustrating the trend of each variable following the sensor and RUL of jet engine

Table 3.2 Data Pre-processing step

3.5.1 Preliminary Analysis

Preliminary analysis is such a crucial steps to be taken in every data analysis. To be well-versed in data collection, deep learning is involved to understand in the area of format, structure and all sorts of variables in the dataset. The observation made in at the early stage can helps in identifying the issues that have to be tackled for reliable analysis. This including outliers, missing values or contradictions.

FD001 will be used in this study. It contains times series for 21 sensors and 3 settings for 100 units of turbofan engine. As per engine nature, it works perfectly fine at the beginning of each time series and it will gradually fail by the end of the time series. Each row is a showing the data taken during single operation cycle which means one complete cycle.

FD001 subset are corresponding to HPC failure of the engine. These including RUL_FD001, test_FD001 and train_FD001. Data inspection including importing train data using 'train_FD001.text'. 20630 data are in train_FD00.

```
#Importing Pandas & Numpys libraries.
import pandas as pd
import numpy as np

#Importing Matplotlibs and Seaborns for bar chart, pie chart and scatter plot.
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#Reading dataset into Data Frame
dataframe = pd.read_csv(r'C:\Users\65917\Downloads\archive (19)\CMaps\train_FD001.txt')
dataframe
```

	1 1 -0.0007 -0.0004 100.0 518.67 641.82 1589.70 1400.60 14.62 21.61 554.36 2388.06 9046.19 1.30 47.47 521.66 2388.02 8138.62 8.4195 0.03 392 2388 100.00 39.06 23.4190
0	1 2 0.0019 -0.0003 100.0 518.67 642.15 1591.82...
1	1 3 -0.0043 0.0003 100.0 518.67 642.35 1587.99...
2	1 4 0.0007 0.0000 100.0 518.67 642.35 1582.79 ...
3	1 5 -0.0019 -0.0002 100.0 518.67 642.37 1582.8...
4	1 6 -0.0043 -0.0001 100.0 518.67 642.10 1584.4...
...	...
20625	100 196 -0.0004 -0.0003 100.0 518.67 643.49 15...
20626	100 197 -0.0016 -0.0005 100.0 518.67 643.54 16...
20627	100 198 0.0004 0.0000 100.0 518.67 643.42 1602...
20628	100 199 -0.0011 0.0003 100.0 518.67 643.23 160...
20629	100 200 -0.0032 -0.0005 100.0 518.67 643.85 16...

20630 rows × 1 columns

Figure 3.3 Importing the libraries

3.5.2 Data Cleaning

Data is being analysed to see the outlier or missing data. There are none hence the data to train has remained 20630.

```
#Exploring number of Columns and Rows
numcolumns = len(dataframe.axes[1])
numrows = len(dataframe.axes[0])

print ("Number of columns:",numcolumns,"\nNumber of rows:",numrows)
```

```
Number of columns: 1
Number of rows: 20630
```

```
#Viewing dataset info to see datatypes for attributes
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20630 entries, 0 to 20629
Data columns (total 1 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   1 1 -0.0007 -0.0004 100.0 518.67 641.82 1589.70 1400.60 14.62 21.61 554.36 2388.06 9046.19 1.30 47.47 521.66 2388.02 8138.62 8.4195 0.
03 392 2388 100.00 39.06 23.4190    20630 non-null  object
dtypes: object(1)
memory usage: 161.3+ KB
```

Figure 3.4 Showing column and rows

The 'train_FD' then being defined to show all the sensors setting. Initial_setting, setting_2 and end_setting is the default setting used to imitate the engine performance. s_1 till s_21 is the sensor numbering for the 21 sensors. 20631 rows including column name.

```
# define column names for easy indexing
index_names = ['unit_nr', 'time_cycles']
setting_names = ['initial_setting', 'setting_2', 'end_setting']
sensor_names = ['s_{}'.format(i) for i in range(1,22)]
col_names = index_names + setting_names + sensor_names

#Reading dataset into Data Frame (2nd try)
dataframe = pd.read_csv(r'C:\Users\65917\Downloads\archive (19)\CMaps\train_FD001.txt',sep='\s+', header=None, names=col_names)
dataframe
```

	unit_nr	time_cycles	initial_setting	setting_2	end_setting	s_1	s_2	s_3	s_4	s_5	...	s_12	s_13	s_14	s_15	s_16	s_17
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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20626	100	196	-0.0004	-0.0003	100.0	518.67	643.49	1597.98	1428.63	14.62	...	519.49	2388.26	8137.60	8.4956	0.03	397
20627	100	197	-0.0016	-0.0005	100.0	518.67	643.54	1604.50	1433.58	14.62	...	519.68	2388.22	8136.50	8.5139	0.03	395
20628	100	198	0.0004	0.0000	100.0	518.67	643.42	1602.46	1428.18	14.62	...	520.01	2388.24	8141.05	8.5646	0.03	398
20629	100	199	-0.0011	0.0003	100.0	518.67	643.23	1605.26	1426.53	14.62	...	519.67	2388.23	8139.29	8.5389	0.03	395
20630	100	200	-0.0032	-0.0005	100.0	518.67	643.85	1600.38	1432.14	14.62	...	519.30	2388.26	8137.33	8.5036	0.03	396

20631 rows × 26 columns

Figure 3.5 Defining column names for easy indexing

Data cleaning including the task of checking for missing values in each column for the sensor.

```
# Check for missing values and null values
print("\nMissing Values in Each Column:")
print(dataframe.isnull().sum())

checknull = dataframe[dataframe.isna().any(axis=1)]
checknull
```

```
Missing Values in Each Column:
unit_nr            0
time_cycles        0
initial_setting    0
setting_2          0
end_setting        0
s_1                0
s_2                0
s_3                0
s_4                0
s_5                0
s_6                0
s_7                0
s_8                0
s_9                0
s_10               0
s_11               0
s_12               0
s_13               0
s_14               0
s_15               0
s_16               0
s_17               0
s_18               0
s_19               0
s_20               0
s_21               0
dtype: int64
```

unit_nr	time_cycles	initial_setting	setting_2	end_setting	s_1	s_2	s_3	s_4	s_5	...	s_12	s_13	s_14	s_15	s_16	s_17	s_18	s_19	s_20	s_21

0 rows × 26 columns

Figure 3.6 Checking missing values and null values

3.5.3 Data Concatenation

Dataframe showing the engine cycle count, mean, standard deviation, minimum cycle which is 1, 25% , 50%, 75% to maximum 100% running capacity. The descriptive inspected for unit number, clearly shown that the unit numbers started with 1 and finish at 100 as showing the total number of turbofan jet engine as expected; 100 units. This means the reading has been successfully recorded for all units. However, the mean and quantiles does not align with the descriptive statistic based on 1-100 vector hence this showing each unit have their own different reading of max time_cycles. The mean which is the average cycles is 108.81 besides the large spread of standard deviation with the reading 68.88. Else, the 50% running number recorded as 104 cycles while the minimum number as the cycle starter is 1 and maximum cycles operated before the engine stop running due to maximum capacity or what we called broke down and in need of maintenance is 362 cycles. From the recorded data observation, there are possibly outliers at the low end and high end of the distribution because maximum and minimum values are considered as far from the quartiles. Further visualization on other steps are needed to have a better perspective.

```
dataframe.loc[:,['unit_nr','time_cycles']].describe()
```

	unit_nr	time_cycles
count	20631.000000	20631.000000
mean	51.506568	108.807862
std	29.227633	68.880990
min	1.000000	1.000000
25%	26.000000	52.000000
50%	52.000000	104.000000
75%	77.000000	156.000000
max	100.000000	362.000000

Figure 3.7 Engine data

Sensors 1 to 21 with the reading details of mean, standard deviation, quartile, starting cycle values and max cycle values. The values are recorded at the exact time.

```
dataframe.loc[:,'s_1':].describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
s_1	20631.0	518.670000	6.537152e-11	518.6700	518.6700	518.6700	518.6700	518.6700
s_2	20631.0	642.680934	5.000533e-01	641.2100	642.3250	642.6400	643.0000	644.5300
s_3	20631.0	1590.523119	6.131150e+00	1571.0400	1586.2600	1590.1000	1594.3800	1616.9100
s_4	20631.0	1408.933782	9.000605e+00	1382.2500	1402.3600	1408.0400	1414.5550	1441.4900
s_5	20631.0	14.620000	3.394700e-12	14.6200	14.6200	14.6200	14.6200	14.6200
s_6	20631.0	21.609803	1.388985e-03	21.6000	21.6100	21.6100	21.6100	21.6100
s_7	20631.0	553.367711	8.850923e-01	549.8500	552.8100	553.4400	554.0100	556.0600
s_8	20631.0	2388.096652	7.098548e-02	2387.9000	2388.0500	2388.0900	2388.1400	2388.5600
s_9	20631.0	9065.242941	2.208288e+01	9021.7300	9053.1000	9060.6600	9069.4200	9244.5900
s_10	20631.0	1.300000	4.660829e-13	1.3000	1.3000	1.3000	1.3000	1.3000
s_11	20631.0	47.541168	2.670874e-01	46.8500	47.3500	47.5100	47.7000	48.5300
s_12	20631.0	521.413470	7.375534e-01	518.6900	520.9600	521.4800	521.9500	523.3800
s_13	20631.0	2388.096152	7.191892e-02	2387.8800	2388.0400	2388.0900	2388.1400	2388.5600
s_14	20631.0	8143.752722	1.907618e+01	8099.9400	8133.2450	8140.5400	8148.3100	8293.7200
s_15	20631.0	8.442146	3.750504e-02	8.3249	8.4149	8.4389	8.4656	8.5848
s_16	20631.0	0.030000	1.556432e-14	0.0300	0.0300	0.0300	0.0300	0.0300
s_17	20631.0	393.210654	1.548763e+00	388.0000	392.0000	393.0000	394.0000	400.0000
s_18	20631.0	2388.000000	0.000000e+00	2388.0000	2388.0000	2388.0000	2388.0000	2388.0000
s_19	20631.0	100.000000	0.000000e+00	100.0000	100.0000	100.0000	100.0000	100.0000
s_20	20631.0	38.816271	1.807464e-01	38.1400	38.7000	38.8300	38.9500	39.4300
s_21	20631.0	23.289705	1.082509e-01	22.8942	23.2218	23.2979	23.3668	23.6184

Figure 3.8 Sensors Reading details

3.6 Data Modelling

Data being trained to show numbering for 100 engine, starting from 1 to 100.

```
train['unit_nr'].unique()

# There are 100 no unique engines.
```

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,  39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52,
        53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,
        66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,
        79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,
        92,  93,  94,  95,  96,  97,  98,  99, 100], dtype=int64)
```

Figure 3.9 Using 100 unique engines in the dataset.

The dataset provided the RUL (Remaining Useful Life of Engine) values (y_test) for the final cycle test from each engine hence the test set is subset to represent as such.

```
test.shape
```

```
(13096, 26)
```

```
# Since the true RUL values (y_test) for the test set are only provided for the last time cycle of e
# the test set is subsetted to represent the same
test = test.groupby('unit_nr').last().reset_index().drop(['unit_nr','time_cycles'], axis=1)
```

```
y_test.shape

# RUL value for 100 no of engines.
```

```
(100, 1)
```

```
test.shape
# Now test data contains entries for 100 no of engines with their RUL.
```

```
(100, 24)
```

Figure 3.10 Checking for 100 entries with respected RUL

```
train.describe()
```

	unit_nr	time_cycles	setting_1	setting_2	setting_3	s_1	s_2	
count	20631.000000	20631.000000	20631.000000	20631.000000	20631.0	2.063100e+04	20631.000000	20631.0
mean	51.506568	108.807862	-0.000009	0.000002	100.0	5.186700e+02	642.680934	1590.
std	29.227633	68.880990	0.002187	0.000293	0.0	6.537152e-11	0.500053	6.
min	1.000000	1.000000	-0.008700	-0.000600	100.0	5.186700e+02	641.210000	1571.0
25%	26.000000	52.000000	-0.001500	-0.000200	100.0	5.186700e+02	642.325000	1586.2
50%	52.000000	104.000000	0.000000	0.000000	100.0	5.186700e+02	642.640000	1590.1
75%	77.000000	156.000000	0.001500	0.000300	100.0	5.186700e+02	643.000000	1594.3
max	100.000000	362.000000	0.008700	0.000600	100.0	5.186700e+02	644.530000	1616.9

8 rows × 26 columns

Figure 3.11 Train the model to check the outliers again

```
# Remove setting_3 column as we can see that it's value is not changing theerfore will not not add a
# prediction
train=train.drop('setting_3',axis=1)
```

```
# Adding RUL (Ramining Useful Life) to the train dataset
```

```
def add_remaining_useful_life(df):
    # Get the total number of cycles for each unit
    grouped_by_unit = df.groupby(by="unit_nr")
    max_cycle = grouped_by_unit["time_cycles"].max()

    # Merge the max cycle back into the original frame
    result_frame = df.merge(max_cycle.to_frame(name='max_cycle'), left_on='unit_nr', right_index=Tru

    # Calculate remaining useful life for each row
    remaining_useful_life = result_frame["max_cycle"] - result_frame["time_cycles"]
    result_frame["RUL"] = remaining_useful_life

    # drop max_cycle as it's no longer needed
    result_frame = result_frame.drop("max_cycle", axis=1)
    return result_frame

train = add_remaining_useful_life(train)
train[sensor_names+['RUL']].head()
```

Figure 3.12 Remove setting_3 column

Removing setting_3 column as the values is counted as outliers and does not add any information to the prediction model. The RUL has been added to train the dataset

CHAPTER 4

INITIAL RESULTS

4.1 Data Visualization

Data visualization is crucial as it shows the pattern and exact illustration in graph to aid with better understanding on each testing condition.

4.2 EDA & Feature engineering

Turbofan Engines Lifetime showing maximum cycles time for each unit form engine 1-100. Based on the plotted chart, it is clearly shown the performance readings are varies.

```
#Maximum time cycle of each unit
max_time_cycles=dataframe[index_names].groupby('unit_nr').max()


plt.figure(figsize=(20,50))
plt.title('Turbofan Engines LifeTime',fontweight='bold',size=30)
ax=max_time_cycles['time_cycles'].plot(kind='barh',width=0.8, stacked=True,align='center')

plt.ylabel('Time cycle',fontweight='bold',size=20)
plt.yticks(size=15)
plt.xlabel('unit',fontweight='bold',size=20)
plt.xticks(size=15)
plt.grid(False)
plt.tight_layout()

#Show the plot
plt.show()
```

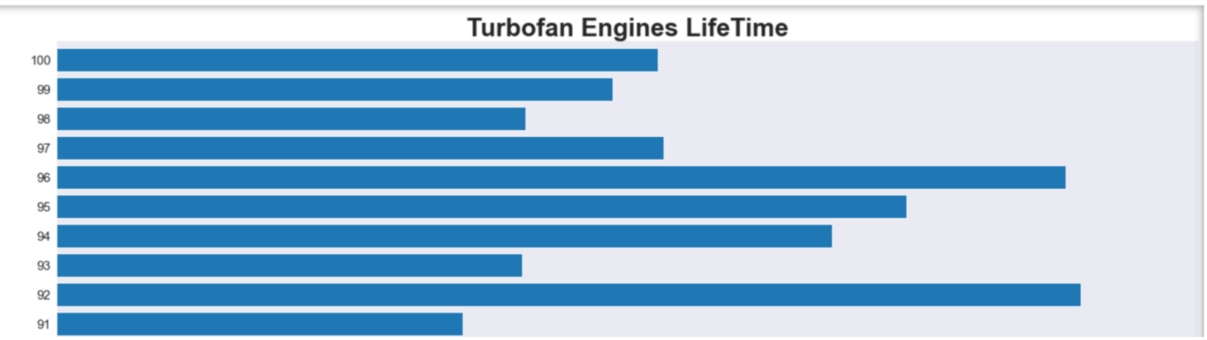

Figure 4.1 The chart showing example reading for engine 91-100.

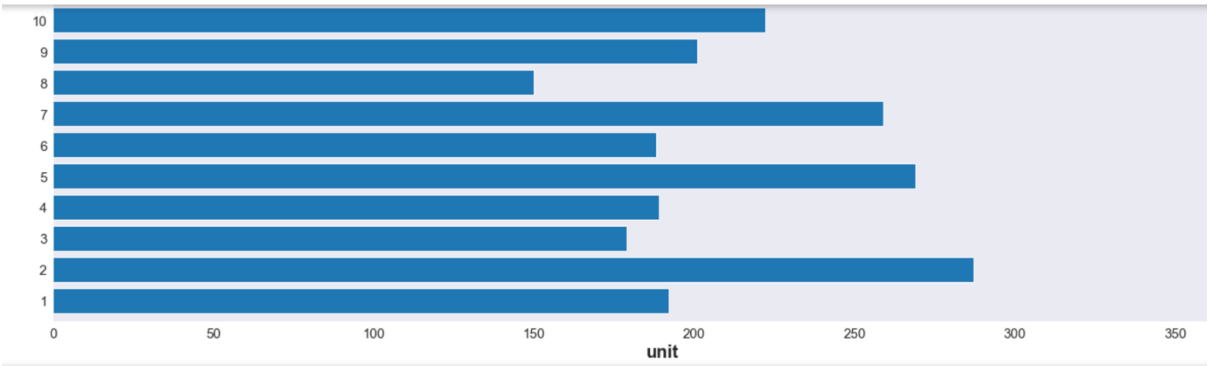

Figure 4.2 The chart showing example reading for engine 1-10.

The distribution of maximum time cycles are shown below indicate the engine can achieve the maximum cycles time is the range 190 and 210 before the HPC Failure. The RUL is skewed to the left side showing the greatest engine performance is at first 50% or up to second quartiles and the trend goes down going through the next third and fourth quartile.

```
#Distribution of maximum time cycles
sns.displot(max_time_cycles['time_cycles'],kde=True,bins=20,height=6,aspect=2)
plt.xlabel('max time cycle')
```

```
Text(0.5, 9.444444444444459, 'max time cycle')
```

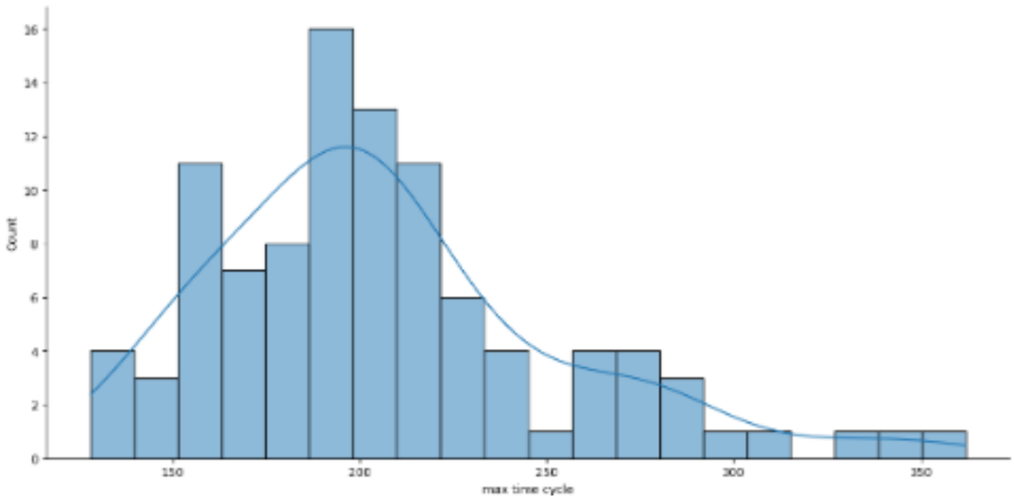

Figure 4.3 : Distribution of maximum time cycles

Plotting the Sensor Values vs Engine RUL. Sensor s_1 to s_21 all are having different reading to show the correlation between each sensor and the function to determine the RUL. Based on the observation, sensor 1,5,6,10,16,18 and 19 have remained constant throughout the cycles test hence this means these sensors have no useful information and not have significant function to the RUL. Figures 4.1-4.6 Showing sensors graph plotted against engine RUL.

```
def plot_sensor(sensor_name):
    plt.figure(figsize=(13,5))
    for i in train['unit_nr'].unique():
        if (i % 10 == 0):  # only plot every 10th unit_nr
            plt.plot('RUL', sensor_name,
                     data=train[train['unit_nr']==i])
    plt.xlim(250, 0)  # reverse the x-axis so RUL counts down to zero
    plt.xticks(np.arange(0, 275, 25))
    plt.ylabel(sensor_name)
    plt.xlabel('Remaining Use fulLife')
    plt.show()

for sensor_name in sensor_names:
    plot_sensor(sensor_name)
```

Figure 4.5: Sensor values vs RUL s_1 s_3

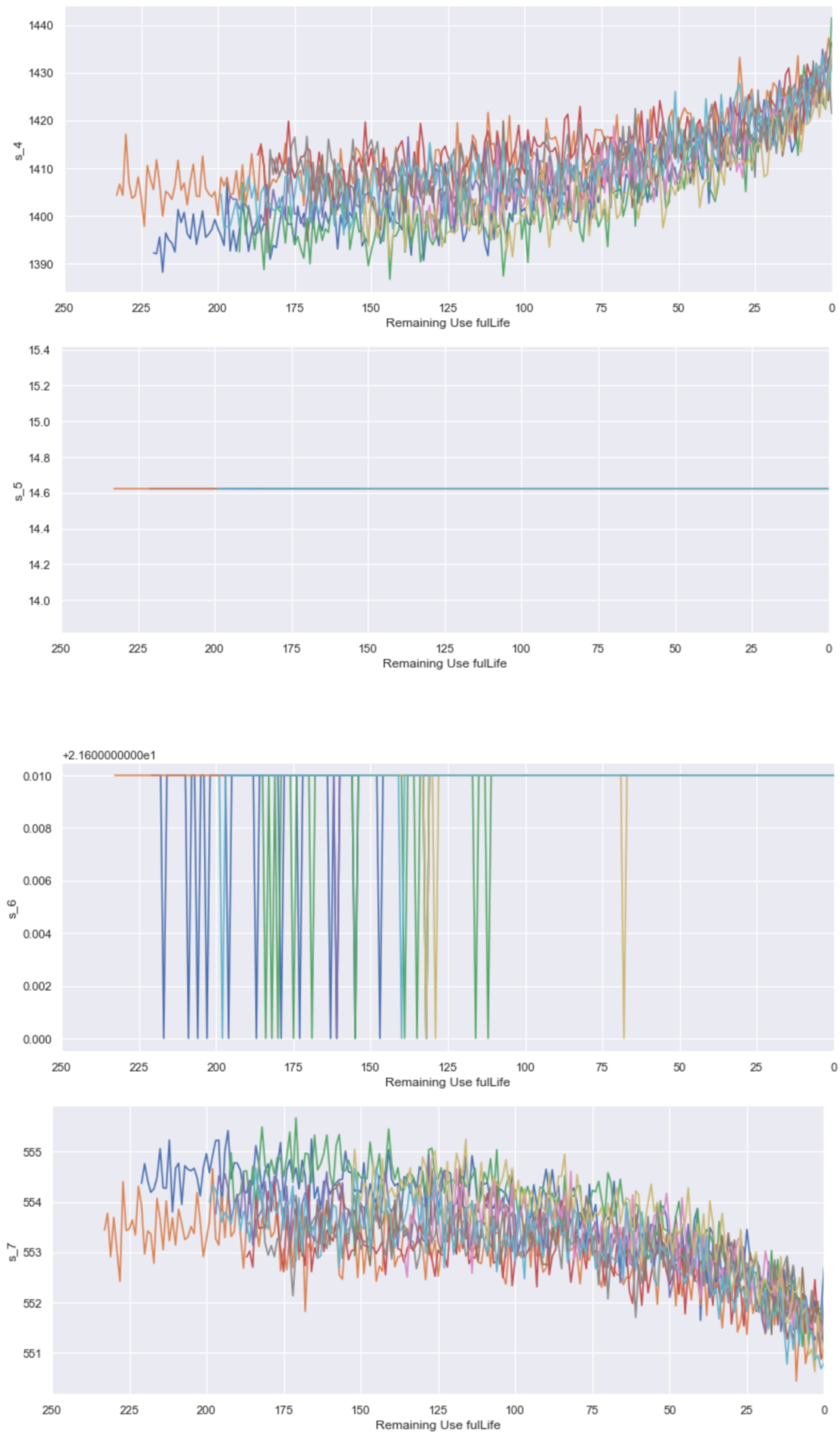

Figure 4.2: Sensor values vs RUL s_4 till s_7

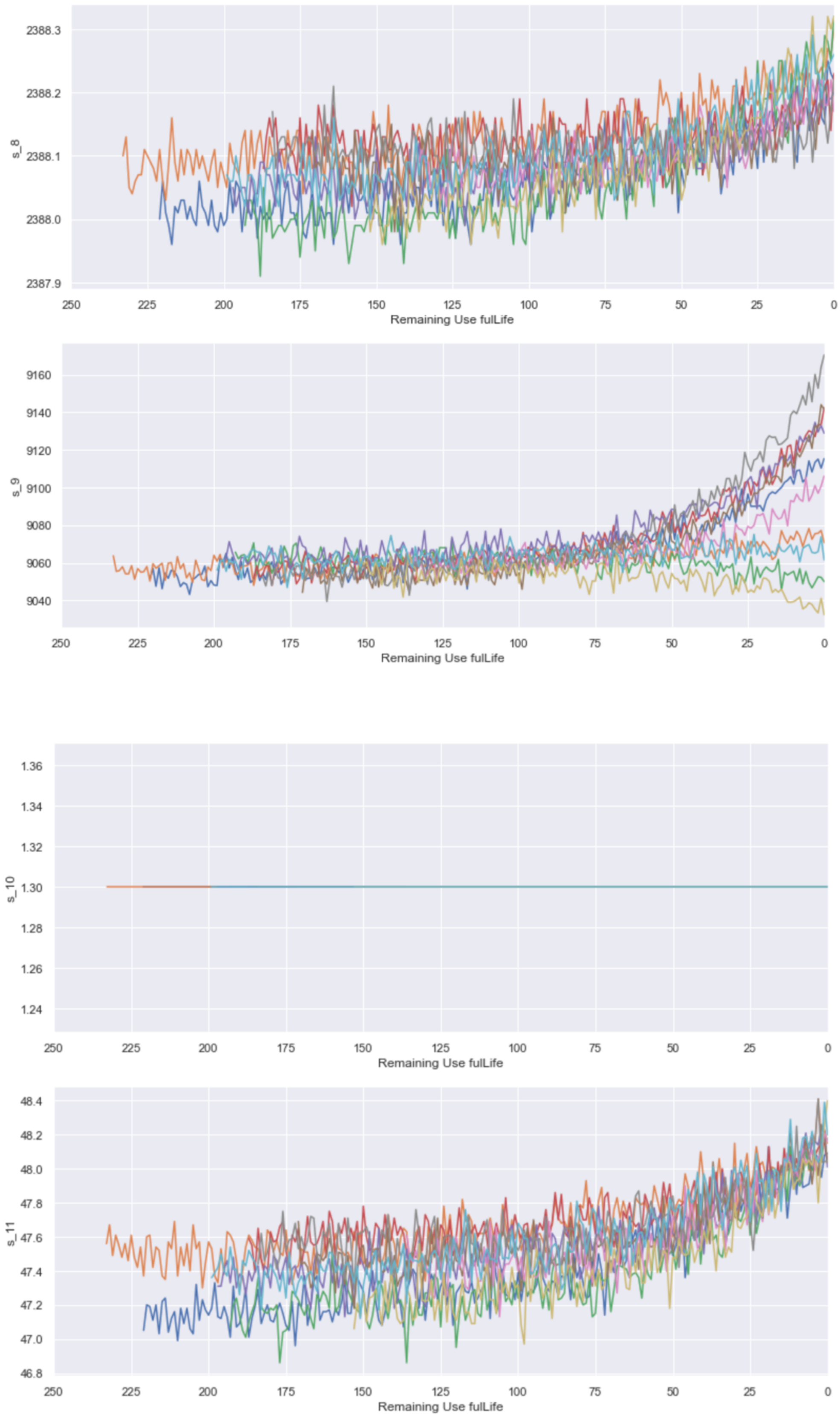

Figure 4.3: Sensor values vs RUL s_8 till s_11

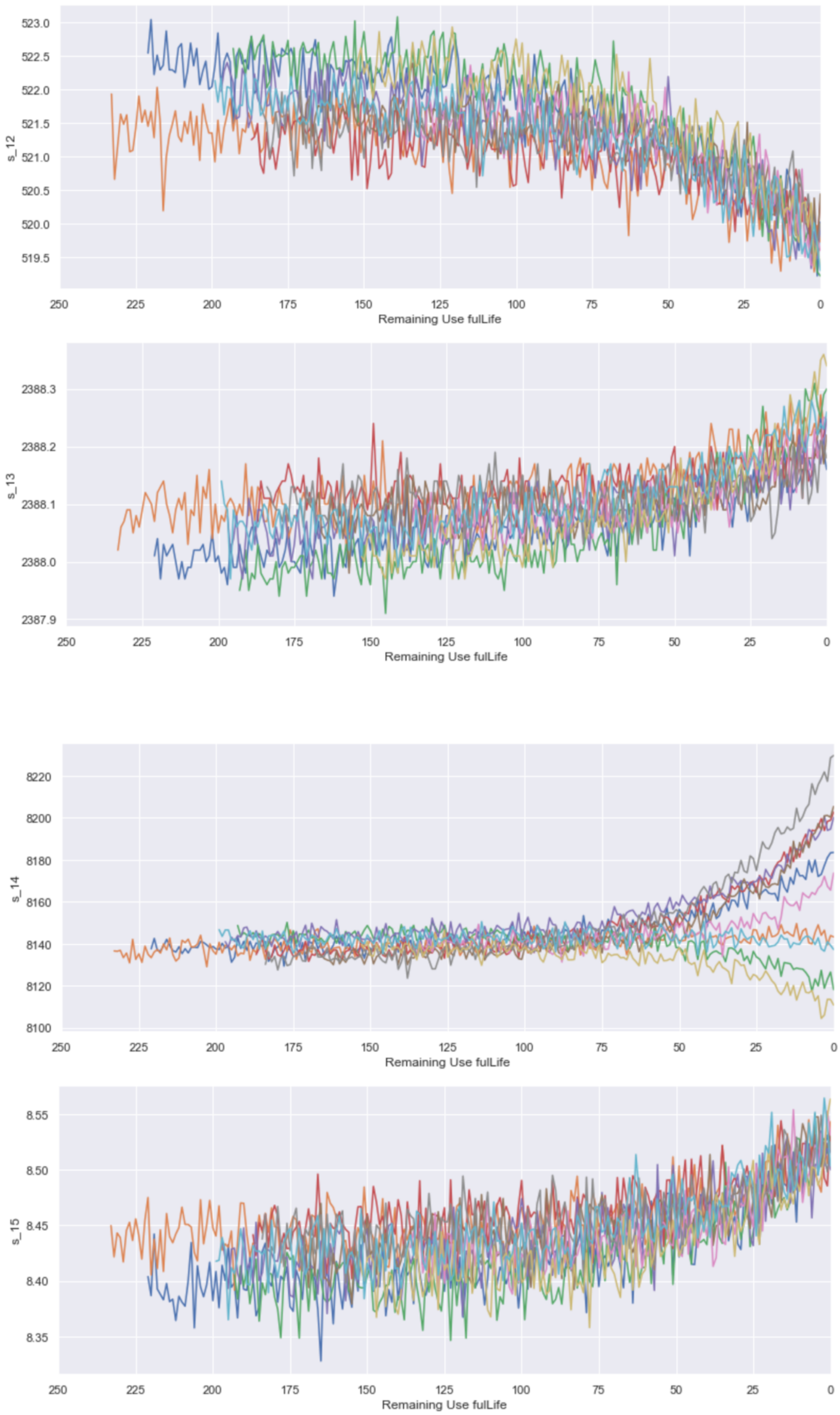

Figure 4.4: Sensor values vs RUL s_12 till s_15

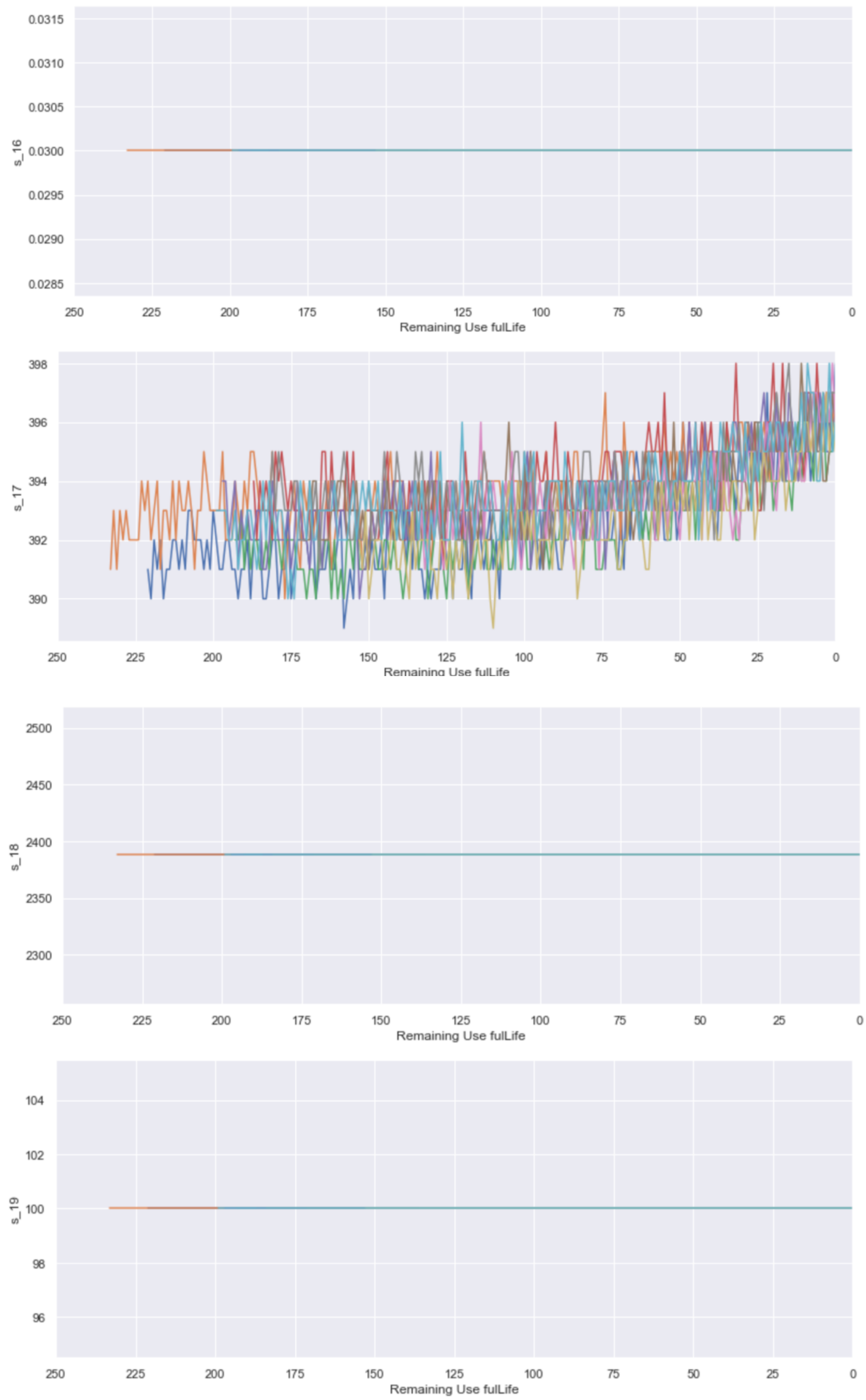

Figure 4.5: Sensor values vs RUL s_16 till s_19

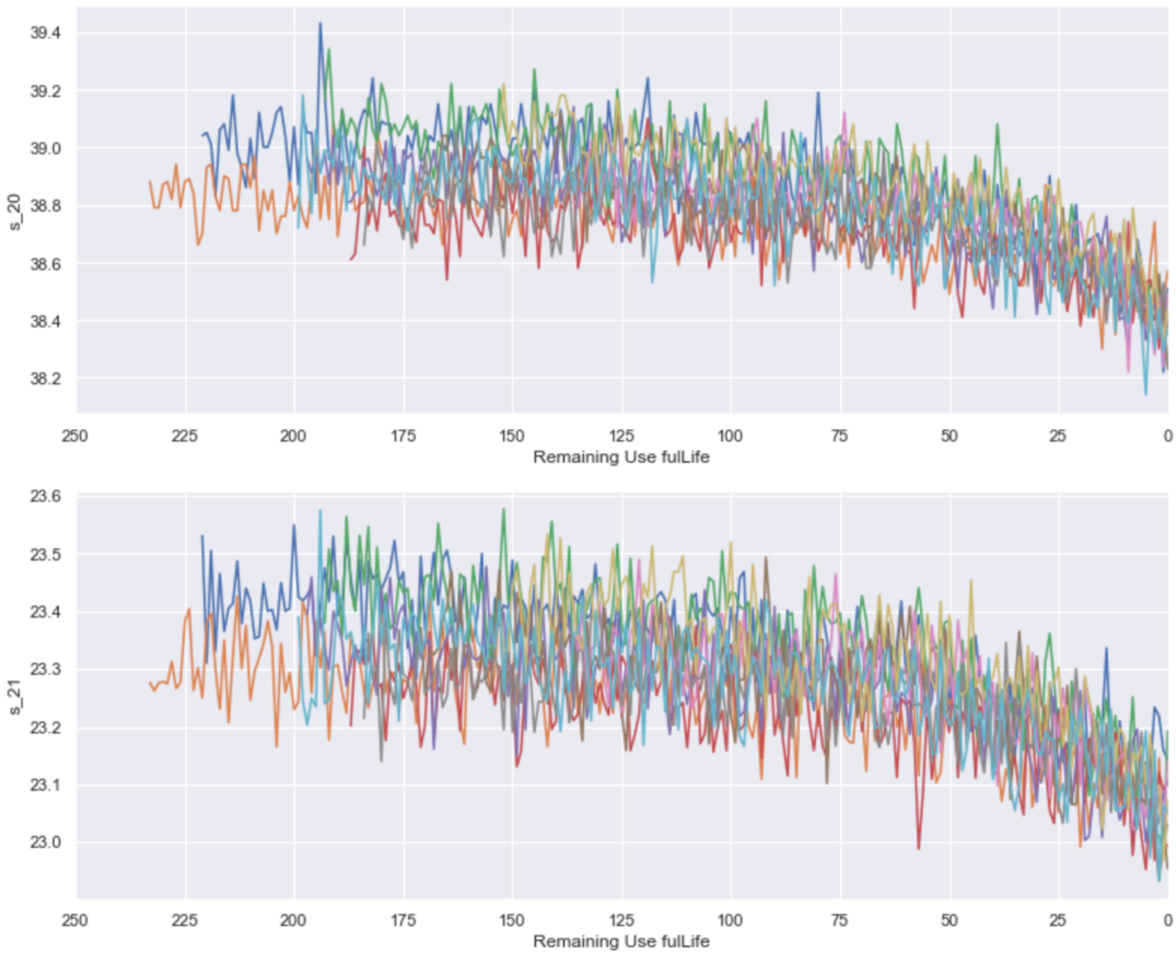

Figure 4.6: Sensor values vs RUL s_20 and s_21

The EDA sensor dictionary 1,5,6,10,16,18 and 19 has remained constant throughout the cycles test hence this means these sensors have no useful information and does not have significant function to the RUL. The figure 4.7 below match each sensor with its significant function.

```
dict_list=[ "(Fan inlet temperature) (◦R)",
"(LPC outlet temperature) (◦R)",
"(HPC outlet temperature) (◦R)",
"(LPT outlet temperature) (◦R)",
"(Fan inlet Pressure) (psia)",
"(bypass-duct pressure) (psia)",
"(HPC outlet pressure) (psia)",
"(Physical fan speed) (rpm)",
"(Physical core speed) (rpm)",
"(Engine pressure ratio(P50/P2)",
"(HPC outlet Static pressure) (psia)",
"(Ratio of fuel flow to Ps30) (pps/psia)",
"(Corrected fan speed) (rpm)",
"(Corrected core speed) (rpm)",
"(Bypass Ratio) ",
"(Burner fuel-air ratio)",
"(Bleed Enthalpy)",
"(Required fan speed)",
"(Required fan conversion speed)",
"(High-pressure turbines Cool air flow)",
"(Low-pressure turbines Cool air flow)" ]
i=1
for x in dict_list :
    Sensor_dictionary['s_'+str(i)]=x
    i+=1
Sensor_dictionary
```

```
{'s_1': '(Fan inlet temperature) (◦R)',
 's_2': '(LPC outlet temperature) (◦R)',
 's_3': '(HPC outlet temperature) (◦R)',
 's_4': '(LPT outlet temperature) (◦R)',
 's_5': '(Fan inlet Pressure) (psia)',
 's_6': '(bypass-duct pressure) (psia)',
 's_7': '(HPC outlet pressure) (psia)',
 's_8': '(Physical fan speed) (rpm)',
 's_9': '(Physical core speed) (rpm)',
 's_10': '(Engine pressure ratio(P50/P2)',
 's_11': '(HPC outlet Static pressure) (psia)',
 's_12': '(Ratio of fuel flow to Ps30) (pps/psia)',
 's_13': '(Corrected fan speed) (rpm)',
 's_14': '(Corrected core speed) (rpm)',
 's_15': '(Bypass Ratio) ',
 's_16': '(Burner fuel-air ratio)',
 's_17': '(Bleed Enthalpy)',
 's_18': '(Required fan speed)',
 's_19': '(Required fan conversion speed)',
 's_20': '(High-pressure turbines Cool air flow)',
 's_21': '(Low-pressure turbines Cool air flow)'}
```

Figure 4.7: Sensor dictionaries

In the context of a jet engine, a **heatmap** is used to visualize the sensor training vs RUL where it showing the correlation between these two or even its components. Heatmap act as a visual way to show the relationship between sensor data (such as vibration, pressure, temperature etc.) with the engine performance degradation over period of time. This method is being used to highlight which variables are more significant affecting the RUL.

Brighter colours showing high intensity regions indicate strong correlation between sensor readings and RUL as for example, the higher temperatures strongly indicate faster degradation hence the RUL being reduced.

Darker colours act as indicator of low intensity regions hence the correlation between sensors reading and RUL is weaker, meaning the sensors are less significant in RUL prediction.

The figure showing colour patterns and the pattern can indicate critical points as for example when engine is getting to the end of its life, the sensors that shows vibration will give higher reading and on heatmap it will show brighter colour.

```
sns.heatmap(dftrain.corr(),annot=True,cmap='RdYlGn')
fig=plt.gcf()
fig.set_size_inches(20,20)
plt.show()
```

Figure 4.8 Train the sensor data to display Heatmap

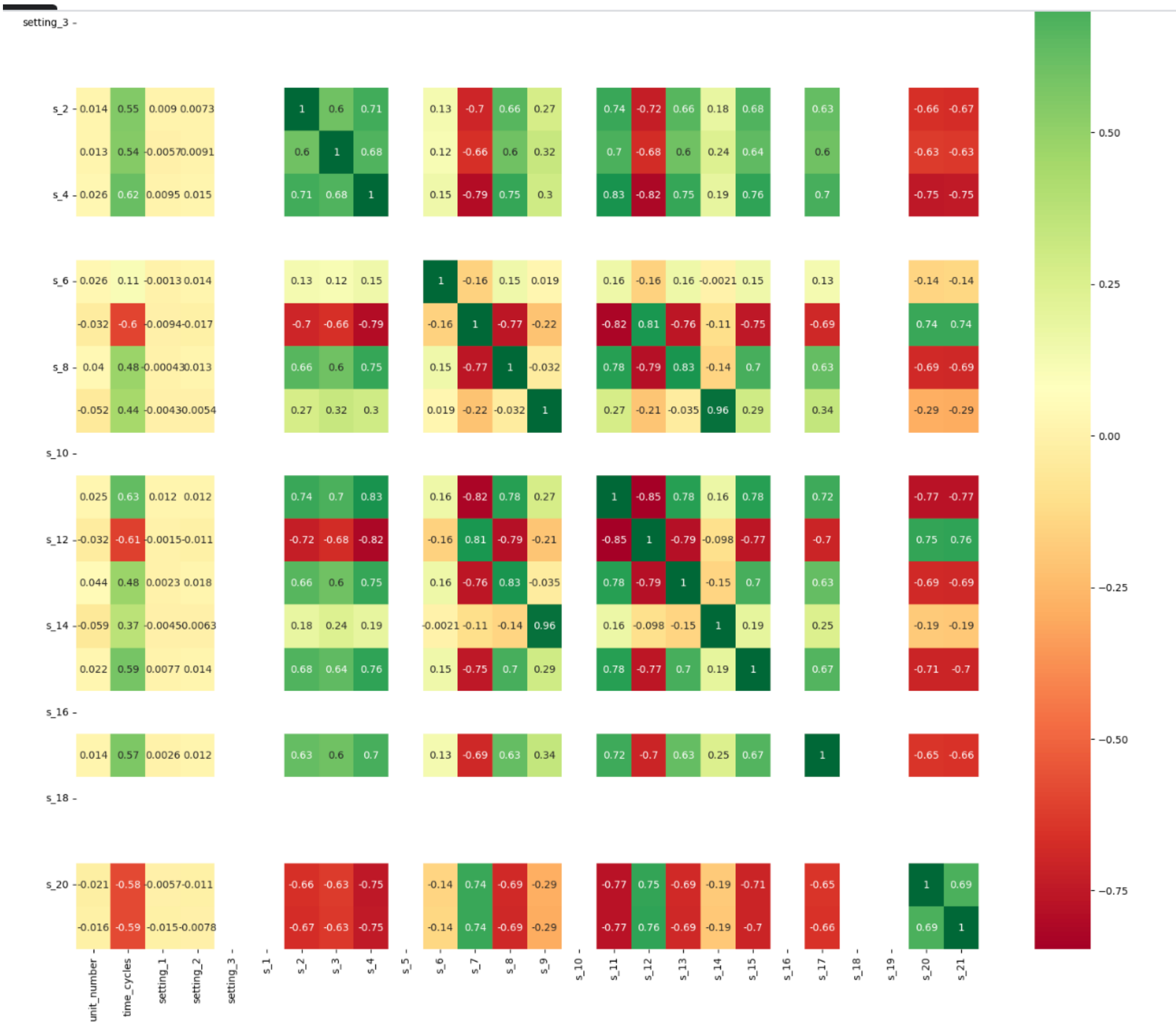

Figure 4.9 Heatmap indicate the correlation between sensors & RUL

Based on the input from heatmap, it is beneficial to identify which sensors that indicate which variables need to be focus onto and which that need to be eliminate. This is call as dimensionality reduction to reduce unnecessary features to reduce the overfitting.

By selecting highly correlated features, this can helps to reduce overfitting and avoid the wastage for the engine performance. Training relevant features only are so important to ensure the credibility of RUL. The details shown as per Figure 4.10.

```
cor=train.corr()
#cor_target = abs(cor["RUL"])
#Selecting highly correlated features
train_relevant_features = cor[abs(cor['RUL'])>=0.5]

train_relevant_features['RUL']
```

```
time_cycles   -0.736241
s_2           -0.606484
s_3           -0.584520
s_4           -0.678948
s_7            0.657223
s_8           -0.563968
s_11          -0.696228
s_12           0.671983
s_13          -0.562569
s_15          -0.642667
s_17          -0.606154
s_20           0.629428
s_21           0.635662
RUL            1.000000
Name: RUL, dtype: float64
```

```
list_relevant_features=train_relevant_features.index
list_relevant_features=list_relevant_features[1:]
list_relevant_features
```

```
Index(['s_2', 's_3', 's_4', 's_7', 's_8', 's_11', 's_12', 's_13', 's_15',
       's_17', 's_20', 's_21', 'RUL'],
      dtype='object')
```

Figure 4.10 Select and train relevant sensors features

Out if 21 sensors, only 12 sensors contributing relevant aspects to the RUL hence only 12 sensors are train to see the correlation with RUL

```
# Above list contains important features have correlation of magnitude greater and equal to 0.5 with our target variable RUL.
```

```
# Now we will keep only these important features in both train & test dataset.
train=train[list_relevant_features]
```

```
train.head(5)
```

	s_2	s_3	s_4	s_7	s_8	s_11	s_12	s_13	s_15	s_17	s_20	s_21	RUL
0	641.82	1589.70	1400.60	554.36	2388.06	47.47	521.66	2388.02	8.4195	392	39.06	23.4190	191
1	642.15	1591.82	1403.14	553.75	2388.04	47.49	522.28	2388.07	8.4318	392	39.00	23.4236	190
2	642.35	1587.99	1404.20	554.26	2388.08	47.27	522.42	2388.03	8.4178	390	38.95	23.3442	189
3	642.35	1582.79	1401.87	554.45	2388.11	47.13	522.86	2388.08	8.3682	392	38.88	23.3739	188
4	642.37	1582.85	1406.22	554.00	2388.06	47.28	522.19	2388.04	8.4294	393	38.90	23.4044	187

Figure 4.11 Train the relevant sensors features

Among all sensors, selected features are as below dictionaries. Only these variables are being used to test and train for modelling purpose.

```
[['cycle',
  '(LPC outlet temperature) (◦R)',
  '(HPC outlet temperature) (◦R)',
  '(LPT outlet temperature) (◦R)',
  '(bypass-duct pressure) (psia)',
  '(HPC outlet pressure) (psia)',
  '(Physical fan speed) (rpm)',
  '(Physical core speed) (rpm)',
  '(HPC outlet Static pressure) (psia)',
  '(Ratio of fuel flow to Ps30) (pps/psia)',
  '(Corrected fan speed) (rpm)',
  '(Bypass Ratio) ',
  '(Bleed Enthalpy)',
  '(High-pressure turbines Cool air flow)',
  '(Low-pressure turbines Cool air flow)']]
```

Figure 4.12 Sensors variables used in modelling

4.3 Initial insights

In this chapter, only 2 machine learning model are being used to do initial finding for turbofan engine RUL. The model chosen are the simplest machine learning model which are Linear regression and Support Vector Regression (SVR). **Table 4.1** explain the comparison aspects.

Aspect	**Linear Regression**	**Support Vector Regression (SVR)**
Data Relationship	Assumes a **linear relationship** between sensor features and RUL.	Captures **non-linear relationships** in sensor data using kernels.
Robustness to Outliers	**Sensitive** to outliers; large deviations can distort predictions.	**Robust** to outliers; uses a margin of tolerance (ε) to ignore noise.
Feature Complexity	Requires explicit **feature engineering** for non-linear trends (e.g., adding polynomial terms).	Automatically handles **complex patterns** through kernel transformations.
Interpretability	**Highly interpretable**; easy to explain relationships between features and RUL.	**Less interpretable**; kernel transformations obscure feature contributions.
Scalability	**Efficient**; suitable for large datasets with many features.	Computationally **expensive** for large datasets, especially with RBF kernels.
Performance on Non-linear Data	Performs poorly if the sensor data has **non-linear degradation patterns**.	Performs well on **non-linear degradation trends** in sensor data.
Ease of Implementation	Simple and fast to implement as a baseline model.	More complex; requires tuning of parameters (C, ε, kernel).
Training Time	**Very fast**, even for large datasets.	**Slower**, especially for high-dimensional data or large datasets.
Example Use Case	Effective when RUL decreases **linearly** with operating cycles or sensor degradation.	Effective for **complex, non-linear degradation** patterns over time.

Table 4.1 Linear regression and SVR comparison aspects

Model 1: Linear Regression

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train1 = sc.fit_transform(X_train)
X_test1 = sc.transform(X_test)
```

```
# create and fit model
lm = LinearRegression()
lm.fit(X_train1, y_train)

# predict and evaluate
y_hat_train1 = lm.predict(X_train1)
RMSE_Train,R2_Train=evaluate(y_train, y_hat_train1,'train')

y_hat_test1 = lm.predict(X_test1)
RMSE_Test,R2_Test=evaluate(y_test, y_hat_test1,'test')
```

```
train set RMSE:22.734164950962253, R2:0.7023848970100307
test set RMSE:22.914265328584634, R2:0.6959448729951723
```

```
# Make Dataframe which will contain results of all applied Model
Results=pd.DataFrame({'Model':['LR'],'RMSE-Train':[RMSE_Train],'R2-Train':[R2_Train],'RMSE-Test':[RMSE_Test],
Results
```

	Model	RMSE-Train	R2-Train	RMSE-Test	R2-Test
0	LR	22.734165	0.702385	22.914265	0.695945

Figure 4.13 Linear Regression

After training, actual versus predicted graph are plotted as a comparison to see if Linear regression model are capable to show the predicted trend (red graph) related to the actual RUL (blue graph)

```
# Plot Actual Vs Predicted RUL for Train Data
#c = [i for i in range(1,81,1)]
fig = plt.figure();
plt.figure(figsize=[20,12])
plt.plot(y_train,color="blue", linewidth=2.5, linestyle="-",label="Actual")
plt.plot(y_hat_train1,color="red",  linewidth=2.5, linestyle="-",label="Predicted")
fig.suptitle('Actual and Predicted', fontsize=20)              # Plot heading
#plt.xlabel('Index', fontsize=18)                              # X-label
plt.ylabel('RUL', fontsize=16)                                 # Y-label
plt.legend()
plt.title("Actual RUL Vs Predicted RUL for Train Data")
```

```
Text(0.5, 1.0, 'Actual RUL Vs Predicted RUL for Train Data')
<Figure size 432x288 with 0 Axes>
```

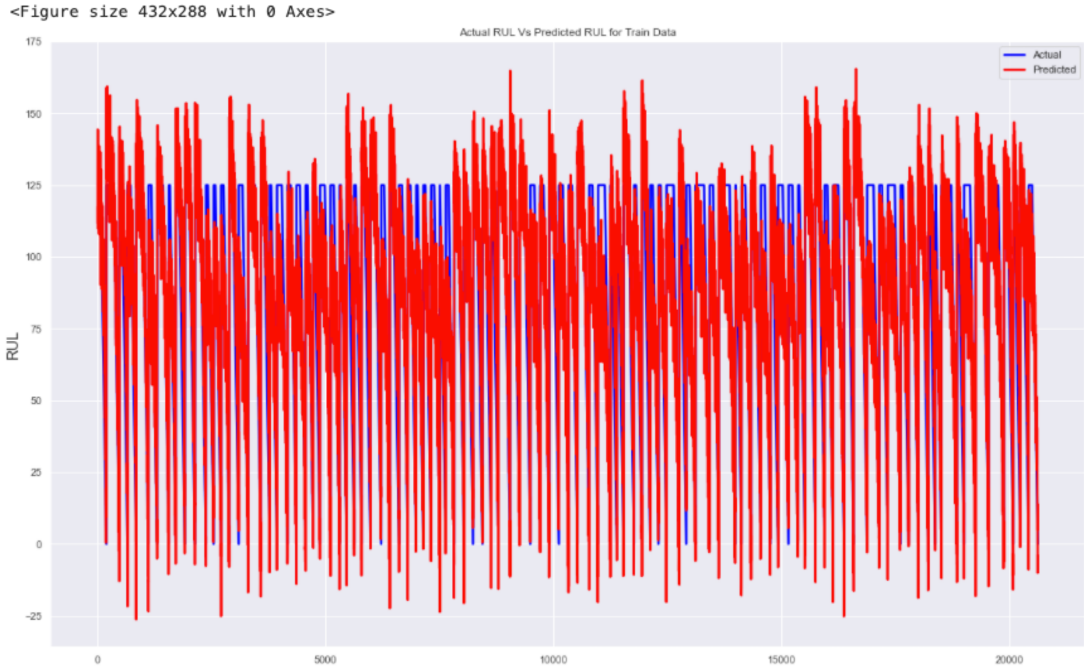

Figure 4.14 Actual vs predicted RUL for train data

The test data showing Root Mean Square Error ,RMSE = 22.9 while R2 score = 0.695. Other machine learning model need to be run, compare and evaluate to see which machine learning model have a better RUL prediction.

```
# Making predictions using test set
y_hat_test1 = lm.predict(X_test1)
evaluate(y_test, y_hat_test1)
```

```
test set RMSE:22.914265328584634, R2:0.6959448729951723
(22.914265328584634, 0.6959448729951723)
```

```
# Plot Actual Vs Predicted RUL for Test Data
#c = [i for i in range(1,81,1)]
fig = plt.figure();
plt.figure(figsize=[20,12])
plt.plot(y_test,color="blue", linewidth=2.5, linestyle="-",label="Actual")
plt.plot(y_hat_test1,color="red",  linewidth=2.5, linestyle="-",label="Predicted")
fig.suptitle('Actual and Predicted', fontsize=20)              # Plot heading
#plt.xlabel('Index', fontsize=18)                               # X-label
plt.ylabel('RUL', fontsize=16)                                # Y-label
plt.legend()
plt.title("Actual RUL Vs Predicted RUL for Test Data")
```

```
Text(0.5, 1.0, 'Actual RUL Vs Predicted RUL for Test Data')
<Figure size 432x288 with 0 Axes>
```

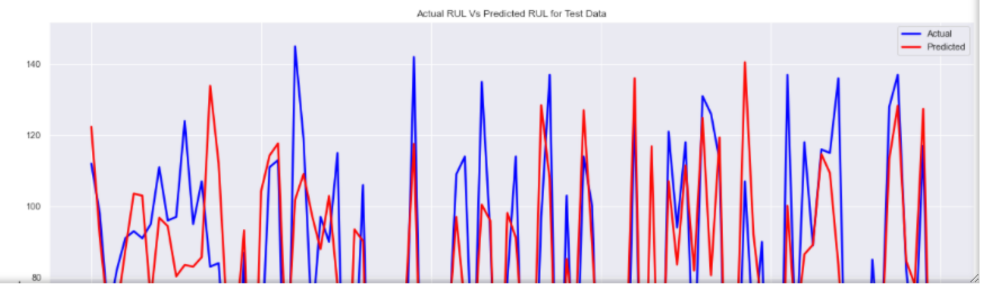

Figure 4.15 Actual vs predicted RUL for test data

<u>Model 2 : Applying Support Vector Regression (SVR)</u>

The RMSE Result for test and train model has been increased after applying SVR model. The value now is lower than Linear regression models meaning magnitude of error made by prediction model is become smaller hence the result are more accurate as shown in **Figure 4.16**

```
from sklearn.svr import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X_train1, y_train)

# predict and evaluate
y_hat_train1 = regressor.predict(X_train1)
RMSE_Train,R2_Train=evaluate(y_train, y_hat_train1)

y_hat_test1 = regressor.predict(X_test1)
RMSE_Test,R2_Test=evaluate(y_test, y_hat_test1)
```

```
test set RMSE:21.42864070146049, R2:0.7355849328257191
test set RMSE:21.828222300276334, R2:0.7240838052912153
```

```
# Make Dataframe which will contain results of all applied Model
Results=Results.append(pd.DataFrame({'Model':['SVM'],'RMSE-Train':[RMSE_Train],'R2-Train':[R2_Train],'RMSE-T
Results
```

	Model	RMSE-Train	R2-Train	RMSE-Test	R2-Test
0	LR	22.734165	0.702385	22.914265	0.695945
1	SVM	21.428641	0.735585	21.828222	0.724084

Figure 4.16 Result for Linear Regression LR and SVR

All 21 sensors has been through data cleaning, training and test before producing the heatmap. The heatmap indicate the sensor correlation with the turbofan jet engine RUL. Then only sensor with the strong correlation with RUL are being used to test and train under 2 machine learning models.

Since only 2 models are used for feature engineering which are Linear Regression and Support Vector Regression and run only once, further repetition is advisable to do. Although the RMSE values has decreased and the result for RUL is not far from the actual RUL recorded at the beginning of this chapter, there are still 4 models that will be run in near future so that better comparison can be made.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This study aims to bring forth predictive maintenance system for jet engines by utilizing extensive sets of engine performance data and sensor information to the full potential for predicting possible engine malfunction, enhance maintenance plans to increase engine effectiveness. From the analysis, it can be concluded that the predictive analysis following Rolls Royce approach are vital in foresee the RUL for jet engine.

Through the use of CMAPPS turbofan jet engine data from NASA together with the 21 sensors for EDA and feature engineering by applying first two machine learning model which are Linear regression and SVR, we can determine The RMSE Result for test and train model has been increased after applying SVR model. The value of SVR model is lower than Linear regression models meaning magnitude of error made by prediction model is become smaller hence the result are more accurate for comparison the RUL before machine learning model and after for the 100 units of jet engines. The sensor's features that are having high correlation with the RUL of the jet engines also has been successfully identified.

To summarize based on initial insights, jet engine can achieve the maximum cycles time is the range 190 and 210 before the High Pressure Compressor failure. The greatest engine performance is at first 50% or up to second quartiles and the trend goes down going through the next third and fourth quartile. As a result, this discovery offers insightful information to apply data-driven strategy to predict potential failures in advanced facilitate by manufacturers and service providers improve the predictive maintenance services for aerospace manufacturer and services sector.

5.2 Future Works

Few gaps have been identified as the outcome of this research and these shall be addressed as improvements in the future. This study suggested a few suggestions and insights that could be useful for potential researchers to further explore the research scope on the predictive maintenance of jet engine for engine health monitoring topic within the aerospace sector.

There are few aspects that can be taken into count for expanding the research further. First and foremost, sensor data can be altered by scaling the data with certain the range given by applying method of MinMaxScaler to transform the sensor features. At the same moment, the machine learning model used are only 2 out of 6 model listed and 4 more model shall be applied to continue the research. While running the model, it is advisable to train and test at least 5-10 times for 5 figures dataset to be able to view the clearer output pattern before deciding the RUL besides plotting the visualization for each sensor output to view the trend and further evaluate the finding.

On the contrary, the classification on RUL could be done by group it into 3 classes which are 'No risk, Moderated risk and Risk zone". This is believed to be beneficial for improving the predictive maintenance schedule. In succession, there are 2 more set of data listed in the study which are Aviation Data from ICAO/FAA aircraft engine emissions databank and IoT simulated sensor dataset hence these data shall undergo phase 3 to 6 dataset to explore more about the validation purpose involving testing fault detection performance by using simulated engine data, besides fault detection accuracy and latency reduction as primary performance metrics.

Subsequently, it would be beneficial to have multi-sensor fusion whereby exploration in algorithms needed to combine vibration, acoustic and thermal data for the purpose of enhancing fault detection accuracy. On this occasion, data from the engine emission databank could be used to investigate the predictive maintenance's role in improving fuel efficiency and lowering the carbon emission. In summation, actionable insights are useful for dashboard designing with the sole purpose of providing ultimate understanding in between maintenance teams and the decision-makers for further improving predictive maintenance schedules.

REFERENCES

Nasir, S., Zainab, H., & Hussain, H. K. (2022). *Real-Time Monitoring for Predictive Maintenance in Aerospace*.

Mohanty, R., Ramasamy, A. K., & Mohanty, A. (2021). *IoT and Big Data Analytics for Predictive Maintenance in Aircraft Engines*.

Zhang, L., Wang, Y., & Zhao, X. (2019). *Fault Diagnosis in Jet Engines Using IoT-Based Sensors*.

Li, P., Chen, T., & Xiang, J. (2020). *Digital Twins for Aircraft Engine Health Monitoring*.

Herceg, M., Raff, T., Findeisen, R., & ... (2006). Nonlinear model predictive control of a turbocharged diesel engine. *2006 IEEE Conference …*, *Query date: 2024-12-12 14:04:27*. https://ieeexplore.ieee.org/abstract/document/4777076/

Jafari, S., & Nikolaidis, T. (2019). Meta-heuristic global optimization algorithms for aircraft engines modelling and controller design; A review, research challenges, and exploring the future. *Progress in Aerospace Sciences*, *Query date: 2024-12-12 14:11:01*. https://www.sciencedirect.com/science/article/pii/S0376042118301416

https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data A. Saxena, K. Goebel, D. Simon, and N. Eklund, 'Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation', in the Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08), Denver CO, Oct 2008. (accessed on 3 November 2024)

https://www.nasa.gov/intelligent-systems-division/#turbofan

https://www.easa.europa.eu/en/domains/environment/icao-aircraft-engine-emissions-databank?utm_source=chatgpt.com https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset

Wu, J.; Hu, K.; Cheng, Y.; Zhu, H.; Shao, X.; Wang, Y. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. ISA Trans. 2020, 97, 241-250.

Fink, O.; Wang, Q.; Svensén, M.; Dersin, P.; Lee, W.J.; Ducoffe, M. Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng. Appl. Artif. Intell. 2020, 92, 103678.

Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mech. Syst. Signal Process. 2018, 104, 799-834.

Booyse, W.; Wilke, D.N.; Heyns, S. Deep digital twins for detection, diagnostics and prognostics. Mech. Syst. Signal Process. 2020, 140, 106612

Da Costa, P.R.d.O.; Akcay, A.; Zhang, Y.; Kaymak, U. Remaining useful lifetime prediction via deep domain adaptation. Reliab. Eng. Syst. Saf. 2020, 195, 106682.

Prognostics Center of Excellence-Data Repository. 2020. Available online: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository / (accessed on 3 November 2024).

Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab. Eng. Syst. Saf. 2018, 172, 1-1.

Listou Ellefsen, A.; Bjorlykhaug, E.; Asoy, V.; Ushakov, S.; Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab. Eng. Syst. Saf. 2019, 183, 240-251.

Remaining Useful Life Prediction of Airplane Engine Based on PCA-BLSTM. Sensors 2020, 20, 4537.

LIST OF PUBLICATIONS