Cluster-Based Fault Analysis and Comparative Modelling for Predicting Aircraft Engine RUL

by

Ravivarman Devarajan

A thesis submitted in partial fulfilment of the requirement of Glasgow Caledonian University for the degree of Master of Science in Applied Data Science in Engineering

May 2025

Abstract

The ability to accurately predict the Remaining Useful Life (RUL) of aircraft engines is crucial for enhancing operational efficiency, minimizing downtime and ensuring flight safety. This study integrates Cluster-Based Sensor Trend Analysis with Machine Learning (ML) and Deep Learning (DL) models to develop a robust predictive maintenance framework using C-MAPSS dataset.

A cluster-based approach is employed to analyse sensor trends, distinguishing fault conditions and assessing sensor data as a reliable predictor of fault propagation. Advanced feature engineering techniques, including rolling statistics and exponential weighted means are applied to enhance data representation. Multiple ML and DL models, such as Random Forest, XGBoost, Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) are trained and evaluated for RUL prediction. Model performance is assessed using key metrics like Root Mean Squared Error (RMSE), R-squared scores and computational efficiency.

The results demonstrate that sensor-driven clustering provides deeper insights into fault progression, supporting the development of fault-specific predictive models. The findings contribute to the advancement of data-driven Prognostics and Health Management (PHM) systems, facilitating more reliable predictive maintenance strategies in the aerospace industry.

Acknowledgements

This thesis was done in the Glasgow Caledonian University over the period of Feb 2025 to May 2025.

I would foremost like to thank my supervisor Professor Dr. Amit Kumar Jain from the School of Engineering and Computing at Glasgow Caledonian University for the helpful comments and unlimited support that has been given to me throughout this work. Furthermore, I would like to thank especially my Module Leader, Professor Dr. Octavian Niculita, for his advice, all the helpful thoughts and inspiring discussions as well as for his overall support and encouragement he has given to me all the time.

Dedication

This thesis is dedicated to my family, without whom none of this would have been even possible. A special thank you goes to Gokul Nathan, my biggest inspiration and a brother who has been supporting my work with energy and faith. Last but not least I would like to thank all my friends for their encouragement and support during the creation of this work.

Author's Declaration

I declare that this thesis is original and my own work and that all materials not my own have been identified and referenced.

Abbreviations

EDA	Exploratory Data Analysis
PHM	Prognostics and Health Management
RUL	Remaining Useful Life
ML	Machine Learning
DL	Deep Learning
HPC	High Pressure Compressor
LPC	Low Pressure Compressor
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
RNN	Recurrent Neural Network
NASA	National Aeronautics and Space Administration
AE	Auto Encoder
EPR	Engine Pressure Ratio
TRA	Throttle Resolver Angle
PCA	Principal Component Analysis
DBN	Deep Belief Network
SVM	Support Vector Machines
AI	Artificial Intelligence
GMM	Gaussian Mixture Model
EM	Expectation Maximization
WPCA	Weighted Principal Component Analysis
LSTM	Long-Short Term Memory
RF	Random Forest
XGB	Extreme Gradient Boosting
MLP	Multi Layered Perceptron
RFE	Recursive Feature Elimination
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
GRU	Gated Recurrent Unit
CNN	Convolutional Neural Networks
RBM	Restricted Boltzmann Machine
T-SNE	t – Distributed Stochastic Neighbour Embedding
UMAP	Uniform Manifold Approximation and Projection
EGT	Exhaust Gas Temperature
HPT	High Pressure Turbine
LPT	Low Pressure Turbine

PFE	Polynomial Feature Extraction
FFT	Fast Fourier Transform
WT	Wavelet Transform
GA	Genetic Algorithm
RFE	Recursive Feature Elimination
FIRF	Feature Importance Ranking Framework
LASSO	Least Absolute Shrinkage and Selection Operator

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Project Justification and Motivation

This project integrates an engineering system specifically, an aircraft engine with advanced data science methodologies, making it highly relevant to my MSc Applied Data Science in Engineering. The intersection of these two domains allows for a comprehensive application of data-driven techniques in a critical engineering context, aligning well with the core learning objectives of my course.

A key motivation for selecting this project is its ability to encapsulate and apply the diverse set of skills and knowledge acquired throughout my MSc program. The project effectively incorporates key concepts from multiple core modules, as outlined below:

Module Name	Key Skills
Data Visualization	Conducting EDA to derive insights from data.
Digital Twins	Fault Analysis, characterisation of fault modes and data modelling for predictive insights.
Software Development for Data Science	Implementing EDA and leveraging supervised and unsupervised learning techniques.
System Health Management	Monitoring system health through data acquisition, failure mode analysis and fault propagation modelling.
Data Capture	Understanding sensor functionalities and data collection methodologies.
Predictive Maintenance	Applying machine learning algorithms for predictive maintenance, statistical analysis and evaluating model performances.
Internet of Things (IoT)	Addressing aspects of data acquisition and security within an industrial setting.

Table 1-1: Mapping of Core Modules to Key Skills

By undertaking this project, I aim to demonstrate the integration of these competencies in a practical, real-world application. The project serves as an opportunity to bridge theoretical knowledge with industrial application, reinforcing my ability to develop data-driven solutions for engineering challenges.

1.2 Project Background and Overview

The increasing complexity of modern aircraft engines demands advanced PHM systems to ensure operational reliability and safety. A key challenge in PHM is accurately predicting the RUL of engine components to facilitate predictive maintenance, reduce unplanned downtime and enhance cost efficiency in fleet management. Traditional maintenance strategies, such as fixed inspection schedules are often inefficient and lead to either premature maintenance or unexpected failures.

In response to this challenge, data-driven approaches leveraging ML and DL techniques have emerged as powerful tools for RUL estimation. These models rely on sensor data from engine operation, capturing degradation patterns over time and forecasting when a failure is likely to occur. However, different fault modes and operational conditions introduce variability, making accurate RUL prediction complex.

To improve fault understanding and model performance, this project integrates Cluster-Based Fault Analysis with Comparative Modelling of ML and DL techniques. Clustering helps distinguish HPC degradation and Fan degradation, providing insights into how different fault conditions affect sensor readings. This knowledge contributes to the development of robust RUL estimation models.

The project utilizes the C-MAPSS dataset, a widely used benchmark for aircraft engine prognostics. The dataset consists of multi-sensor time series data collected from a fleet of engines under varying operating conditions and fault modes. The objective is twofold:

1. Cluster-Based Fault Analysis:
    - ➢ Identify and differentiate fault types (HPC vs. Fan degradation) using unsupervised learning (clustering techniques).
    - ➢ Examine how sensor readings vary across fault conditions which ensures sensor data reliability and helps in determining a robust modelling approach for RUL estimation.
2. Comparative Modelling for RUL Prediction:
    - ➢ Evaluate multiple ML and DL models (e.g., RF, XGBoost, LSTM, CNN) for RUL estimation.
    - ➢ Assess the effectiveness of sensor data and the impact of different fault conditions and operational settings on model performance.

1.2.1 C-MAPSS Dataset

The dataset is divided into four subsets (FD001 – FD004), each representing different operational conditions and fault scenarios:

Dataset	Training Units	Test Units	Operating Conditions	Fault Modes
FD001	100	100	1 (Sea Level)	HPC Only
FD002	260	259	6 (Multiple)	HPC Only
FD003	100	100	1 (Sea Level)	HPC + Fan
FD004	248	249	6 (Multiple)	HPC + Fan

Table 1-2: Overview of C-MAPSS Datasets with Operating Conditions and Fault Modes

Each dataset contains multivariate time series data, where each engine starts in a healthy state and degrades over time until failure. The training data contains full degradation trajectories, while the test data ends before failure, requiring RUL estimation. The dataset includes:

- Three operational settings affecting engine performance.
- 21 sensor measurements capturing system behaviour and degradation patterns.
- Sensor noise simulating real-world uncertainty.

The RULs for each of the four test data set were given for validation.

1.2.2 Significance of the Study

- Enhanced Fault Differentiation: Clustering techniques provide deeper insights into how different faults evolve, helping refine sensor selection.
- Improved Predictive Models: By understanding fault progression and sensor reliability, RUL models can be optimized for greater accuracy and robustness.
- Impact on Aerospace Industry: The findings contribute to data-driven maintenance strategies, enabling airlines and manufacturers to transition towards predictive maintenance frameworks, reducing operational costs and enhancing safety.

1.3 Key Project Objectives

The key objectives of this project are outlined below, focusing on fault differentiation through cluster analysis and comparative modelling for RUL prediction. The achievement of these objectives, along with the challenges encountered and limitations identified are discussed in detail in the subsequent chapters of this dissertation.

- Conduct cluster-based fault analysis on sensor data to effectively distinguish between HPC degradation and Fan degradation, providing insights into fault differentiation.

- Develop and implement advanced feature engineering and data preprocessing techniques to enhance the accuracy and robustness of RUL prediction models.
- Compare and evaluate multiple ML and DL models for predicting the RUL of aircraft engines, assessing their effectiveness under varying operational conditions and fault modes.
- Assess the reliability of sensor data in estimating damage accumulation and degradation trends, establishing its significance in predictive maintenance strategies.
- Identify and recommend optimization techniques and best practices to improve the precision, stability and computational efficiency of RUL prediction models.

Chapter 2 – Literature Review

The literature review conducted in support of this project explores a range of research and published works that examine damage propagation in aircraft engines and its impact on RUL prediction. The review integrates studies from the scope of PHM, focusing on key areas relevant to this research. These include:

- Damage Propagation Modelling in Aircraft Engines: Understand how faults develop in HPC and Fan components and how degradation affects system performance.
- Sensor-Based Monitoring for Fault Diagnosis: Investigating the role of multi-sensor data in tracking degradation trends and enabling predictive maintenance.
- Clustering and Fault Differentiation for RUL Estimation: Exploring how unsupervised learning techniques can distinguish between different fault modes.
- Selection of ML and DL models for RUL prediction: Benchmarking different predictive modelling approaches, including traditional ML methods (RF, XGB etc.) and advanced DL models (LSTM, CNN etc.).
- Feature Engineering Techniques for Time Series Data Modelling (Lag features, Principal Component Analysis etc.)

As the literature review investigates a wide area of interconnected topics that, when combined, provide an essential background knowledge for understanding the subject matter, the author has deemed it appropriate to include a condensed summary of the key insights in Table 2-2 at the end of this chapter.

2.1 Damage Propagation Modelling in Aircraft Engines

A fundamental challenge in RUL prediction lies in understanding and modelling damage propagation, how faults develop in aircraft engines over time. Several approaches have been explored in the literature, including physics-based damage models, data-driven methods and hybrid approaches that integrate both.

2.1.1 Physics-Based Models for Damage Propagation

C-MAPSS dataset, developed by NASA, provides an effective simulation framework for modelling engine degradation and generating data for prognostic algorithm development. C-MAPSS is a MATLAB/Simulink-based tool designed to simulate the performance of a large commercial turbofan engine under various operational conditions. It provides a closed-loop engine control system that allows for realistic simulation of component degradation. The key features of C-MAPSS include:

- Simulation of altitude, Mach number and TRA variations.
- Input parameters for different engine modules.
- Output of sensor measurements that reflect the effects of degradation.
- Integration of operational noise and maintenance variability.

Traditional physics-based models describe the gradual degradation of engine components, often using exponential functions to simulate wear and tear. Saxena *et al.* (2008)[15] discuss some key models such as the Arrhenius Model which is commonly used for chemical and diffusion-related failures, where the time to failure is an exponential function of temperature. The Coffin-Manson Model is applied in cases of mechanical fatigue and crack growth, establishing a correlation between failure cycles and stress conditions. The Eyring Model provides a more generalized approach, incorporating multiple stressors, including temperature and mechanical loads, to assess component degradation. Similar to the ones mentioned, C-MAPSS is also an exponential degradation model which is employed to simulate wear and tear over time. The health index, defined as a function of efficiency and flow parameters [15], decreases until a failure criterion is met. This approach enables the generation of realistic degradation trajectories for use in predictive modelling. According to Xiong Xinxin *et al.* (2015)[19], damage propagation follows an exponential growth pattern, meaning faults start small and increase in severity over time. The study utilized C-MAPSS data to develop an exponential damage model, which serves as a reference for comparing real-time sensor readings and estimating RUL. They explored methods to enhance damage propagation modelling RUL prediction for aircraft engines by fusing multiple sensors into a single health index. Their research emphasized that individual sensor readings often exhibit high variability and noise, making direct RUL estimation challenging. To address this, they proposed a sensor fusion approach that integrates multiple degradation-related signals into a comprehensive health index, providing a more stable and accurate representation of the engine's condition. The study employed statistical feature extraction, dimensionality reduction and fusion techniques to derive a unified degradation metric. This health index was then used to track engine wear and predict failure progression over time. Their results demonstrated that multi-sensor fusion significantly improved the reliability and robustness of predictive models, outperforming traditional methods that rely on isolated sensor readings.

By incorporating sensor fusion into damage models, Xiong Xinxin *et al.* (2015)[19] highlighted the importance of integrating multiple degradation indicators to develop more accurate and interpretable prognostic models.

2.1.2 Data-Driven Approaches to Modelling Damage Propagation

While physics-based models provide insight into failure mechanisms, their effectiveness is often limited by complex real-world conditions and unmodeled uncertainties. As a result, researchers have shifted toward data-driven methods, which leverage sensor data to track degradation trends. Alomari, Mátyás Andó and Baptista (2023)[2] proposed an advanced feature engineering and dimensionality reduction approach to enhance the accuracy of RUL prediction for aircraft engines. Their study focused on extracting meaningful degradation indicators from multivariate sensor data, reducing feature redundance and improving model efficiency. By applying dimensionality reduction techniques, such as PCA and autoencoders, the study demonstrated how refining input features could lead to better generalization and reduced computational complexity in predictive maintenance models. Huthaifa Al-Khazraji *et al.* (2022)[6] introduced a hybrid deep learning framework combining an Autoencoder and a DBN for RUL estimation. Their approach aimed to automatically learn hierarchical feature representations from raw sensor data, capturing underlying degradation trends more effectively than traditional machine learning techniques. The integration of unsupervised feature extraction (Autoencoder) with probabilistic deep learning (DBN) allowed the model to enhance predictive accuracy and adapt to varying failure patterns.

Both studies emphasize the importance of feature selection and transformation in aircraft engine prognostics. Together these contributions highlight the significance of data preprocessing and advanced modelling techniques in developing reliable and efficient predictive maintenance systems for aerospace applications. These studies confirm that sensor-driven degradation analysis can complement physics-based models, leading to more adaptable RUL estimation frameworks.

2.2 Sensor-Based Monitoring for Fault Diagnosis

Sensor-based monitoring has become a critical component in industrial systems, enabling real-time fault detection and predictive maintenance. With the advent of Industry 4.0, multi-sensor systems are increasingly deployed for continuous monitoring of equipment health, allowing for early detection of faults and anomalies.

2.2.1 Role of Multi-Sensor Data in Fault Diagnosis

Multi-sensor data provides a comprehensive picture of system health by capturing different operational parameters, which helps in fault detection, isolation and prognosis. According to Trapani and Longo (2023)[17], the integration of sensor networks enhances real-time monitoring, improving fault diagnosis efficiency in industrial settings. However, the reliability of sensor data remains a key challenge, as sensor failures can lead to incorrect predictions. To address that, the research by Karakose (2016)[7] presents predictive maintenance as a structured approach that integrates condition monitoring, detection, diagnosis and prognosis to improve system reliability.

- **Condition Monitoring:** Involves real-time data collection from sensors measuring key parameters such as temperature, pressure, vibration and electrical currents. By continuously tracking performance trends, this stage helps identify deviations from normal operations using signal processing techniques, statistical analysis and AI-based pattern recognition. Accurate condition monitoring ensures that potential faults can be detected early before they escalate into significant failures.
- **Fault Detection:** Comparing sensor readings against predefined thresholds or historical baseline data to identify anomalies. Traditional threshold-based methods may miss subtle faults, making ML techniques like support SVMs, neural networks and clustering algorithms essential for improving detection accuracy. One of the key challenges in this stage is minimizing false positives and false negatives, ensuring that real faults are detected while avoiding unnecessary maintenance actions.
- **Diagnosis:** It focuses on pinpointing the root cause and location of the fault within the system. This stage often integrates data from multiple sensors to improve accuracy through AI-driven methods such as decision trees, Bayesian networks and fuzzy logic systems. Fault classification is crucial as it helps determine the severity of the issue and provides actionable insights for maintenance.
- **Prognosis (RUL Estimation):** Estimates the RUL of components by analysing historical trends and real-time data. Advanced ML models, DL approaches and Monte Carlo simulations help predict degradation patterns, enabling industries to implement predictive maintenance strategies effectively. A robust prognosis system allows for proactive repair scheduling, reducing unexpected failures and minimizing costly unplanned downtime.

By integrating these four stages, predictive maintenance transforms traditional maintenance approaches, enhancing system efficiency and longevity. The research [7] also highlights the role of ML and optimization algorithms, such as genetic algorithms, artificial immune systems and neural networks in enhancing predictive maintenance.

2.2.2 Fault Detection and Isolation in Sensor Networks

Sensor fault diagnosis plays a crucial role in ensuring the accuracy of predictive maintenance systems. Vasso Reppa, Polycarpou and Panayiotou (2016)[18] discuss how model-based and data-driven approaches can be used to detect sensor faults. They categorize fault isolation architectures into:

- Centralized architectures: All sensor data is processed at a single location, increasing computational burden but allowing holistic analysis.
- Distributed architectures: Fault diagnosis is performed locally at different sensor nodes, reducing communication costs but requiring coordination.
- Decentralized architectures: Sensors operate independently without data exchange, improving robustness against network failures but limiting fault isolation capabilities.

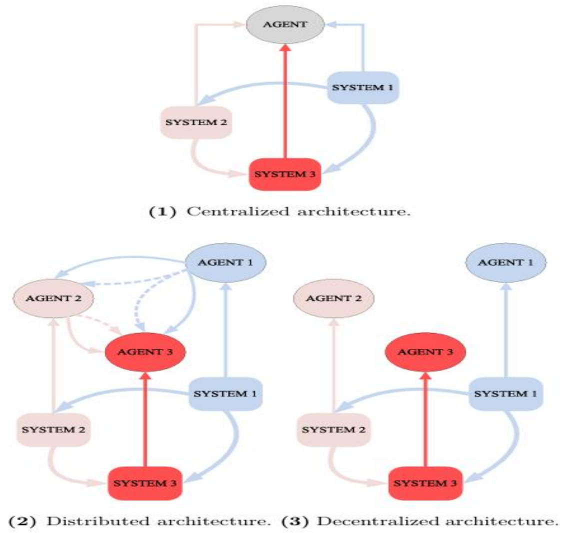


Figure 2-1: Fault Isolation Architectures [18]

The study emphasizes that observer-based methods and adaptive thresholding techniques can improve sensor fault detection in cyber-physical systems.

2.2.3 Machine Learning and AI for Predictive Maintenance

Advancements in AI and ML have significantly improved sensor-based fault diagnosis. The research [7] emphasizes that AI plays a critical role in improving fault detection accuracy and optimizing predictive maintenance strategies. Traditional fault detection methods, which rely on predefined thresholds or rule-based systems, often struggle with complex industrial environments where faults may develop gradually or exhibit subtle patterns that are difficult to detect. AI-based methods, such as neural networks, SVMs and clustering algorithms offer more advanced and adaptive solutions by learning from historical data and identifying complex relationships between sensor measurements.

Soft sensors, also known as virtual sensors, are computational models that estimate physical measurements using ML and statistical techniques instead of relying on direct physical sensors. These models are particularly useful in fault detection and diagnosis because they provide redundancy, enhance accuracy and help mitigate sensor failures by generating reliable data when physical sensors are unavailable. A statistical analysis was made as part of research [17] were recent papers in literature (from 2000 to 2022) with a particular focus on fault and anomaly detection for sensor and sensor networks were chosen and a keyword analysis was carried out with an aim of detecting a common theme in those papers. The results of the analysis shown in Figure 2-2.

From that analysis, it is evident that most of the sensor faults are detected mainly through the analysis of the signals they return [17]. Which proves that the soft sensors are becoming increasingly popular for fault detection and diagnosis.

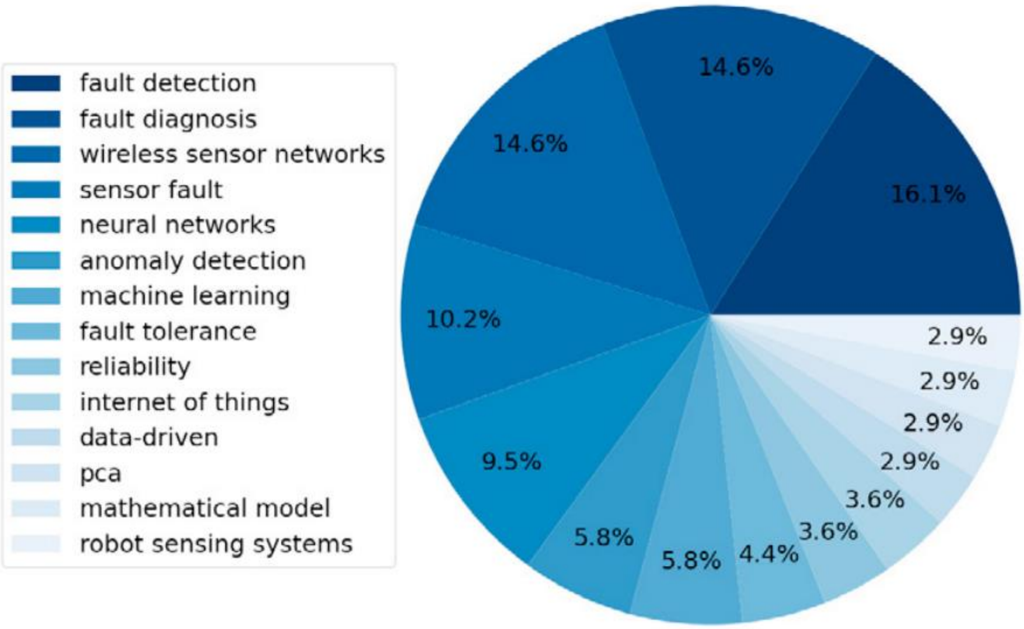


Figure 2-2: Percentage keyword distribution in selected papers [17]

2.2.4 Challenges and Future Research Directions

Despite significant advancements in multi-sensor fault diagnosis, several challenges continue to hinder the widespread adoption of these systems. These challenges primarily revolve around data reliability, computational complexity and scalability, all of which must be addressed to enhance fault detection and predictive maintenance.

- Data Reliability: Sensor failures and noisy data can lead to incorrect predictions.
- Computational Complexity: Processing large scale multi sensor data in real-time requires efficient algorithms.
- Scalability: As sensor networks grow, decentralized and distributed architectures must be optimized for large-scale systems.

To overcome these challenges, future research must focus on developing more efficient, adaptive and intelligent solutions for multi-sensor fault diagnosis. Several promising areas of innovation include hybrid diagnostic approaches, improved AI models and edge computing solutions.

Multi-sensor monitoring is transforming fault diagnosis by enabling predictive maintenance and reducing operational downtime. While AI-driven approaches improve fault detection accuracy, challenges related to data reliability and computational efficiency must be addressed. By integrating soft sensors, distributed architectures and machine learning techniques, future sensor-based monitoring systems can achieve higher reliability and performance.

2.3 Clustering and Fault Differentiation for RUL estimation

One of the primary challenges in RUL estimation is accurately distinguishing between different fault modes and degradation patterns. The ability to differentiate faults is crucial for ensuring reliable prognostics, as treating multiple failure types as a single degradation trend can lead to misleading predictions and reduced model accuracy. In this context, clustering techniques have emerged as powerful tools for fault differentiation, particularly in unsupervised learning scenarios where labelled failure data is scarce. These methods not only facilitate the identification of distinct degradation trajectories but also aid in feature selection and improved predictive modelling, making them especially valuable for users who lack domain expertise in complex industrial systems

2.3.1 Clustering for Fault Identification and Isolation

Traditional fault detection methods rely on supervised learning, requiring extensive labelled datasets. However, in real-world scenarios, labelled failure data is often scarce or unavailable. Unsupervised learning approaches such as clustering, help in differentiating fault conditions without prior fault labels. Clustering is a powerful technique for fault differentiation, as it allows similar degradation patterns to be grouped together, facilitating the development of customized failure thresholds for each cluster. Cho, Carrasco and Ruz (2022)[4] introduce a clustering-based RUL estimation framework that identifies critical segments of degradation signals and groups them into clusters with similar failure progression. Unlike traditional RUL models that apply a single global failure threshold, this approach defines multiple cluster-specific failure thresholds, improving fault differentiation and RUL prediction accuracy.

One key advantage of clustering is that it accounts for variability in degradation trajectories, which can be influenced by factors such as environmental conditions, operating loads and manufacturing inconsistencies. By clustering degradation signals, the study ensures that each group has homogenous failure characteristic, making it easier to develop more reliable RUL prediction models.

Lindström (2021)[10] investigated unsupervised clustering using a modified GMM with EM for fault classification. The study demonstrated that fault signals tend to diverge along specific trajectories which can be modelled using Gaussian distribution. The proposed approach successfully differentiated fault severities and was applied to engine degradation modelling, providing a probabilistic framework for tracking damage propagation.

Additionally, the study utilized WPCA to extract the fault vector as the first principal component, enhancing fault separation. Unlike standard PCA, which finds the directions of maximum variance, WPCA assigns weights to each data point, prioritizing features that contribute significantly to degradation patterns. This approach helped in reducing dimensionality while retaining the most critical fault-related information, improving the clustering accuracy for fault differentiation.

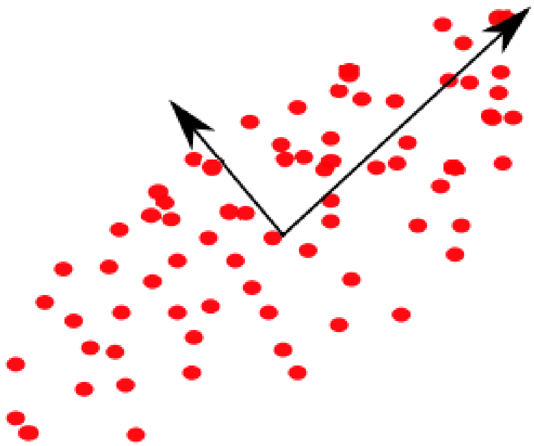


Figure 2-3: An example of PCA displaying the two principal components for fault differentiation [10]

Similarly, Yoo (2020)[21] proposed a correlation-based clustering approach for fault detection. Unlike conventional clustering techniques that rely on dimensionality reduction (e.g., PCA, ICA KPCA), Yoo's method preserves the physical relationships between sensor variables. By selecting highly correlated sensor pairs, this approach enables fault detection without losing interpretability, making it highly relevant for aircraft engine diagnostics. This methodology can be leveraged for RUL estimation by identifying early-stage fault patterns that influence degradation trajectories.

2.3.2 Clustering for RUL Prediction

Recent literatures have explored the integration of clustering techniques with RUL estimation models, bridging the gap between fault differentiation and predictive modelling.

Mohamed *et al.* (2020)[11] presented a distribution-based clustering approach for LSTM-based RUL prediction. The method involved pre-clustering sensor data before training an LSTM model, ensuring that the network learns degradation-specific features rather than treating all engines as a single entity. The study demonstrated that incorporating clustering enhances model generalization and improves RUL prediction accuracy across different failure conditions. The run-to-failure clustering method for defining degradation patterns by Cho, Carrasco and Ruz (2022)[4] improved failure threshold definition leading to more accurate RUL predictions. The study highlighted the importance of distinguishing early-stage degradation trends from terminal failure conditions, which aligns with the objectives of fault differentiation in aircraft engines.

2.3.3 Overall Implications for Aircraft Engine Prognostics

All the reviewed studies thus provide strong evidence that clustering-based fault differentiation can enhance sensor-driven RUL estimation by:

- Improving fault isolation: Clustering enables separation of fault modes, refining input features for predictive models.
- Reducing feature redundancy: By identifying highly correlated sensor pairs, clustering improves model interpretability and efficiency.
- Enhancing model generalization: Clustering-based pre-processing ensures that RUL models capture failure-specific degradation trends rather than generalizing across mixed failure types.
- Defining more accurate thresholds: Clustering critical degradation segments helps in adaptive thresholding, improving RUL predictions under varying operational conditions.

By integrating clustering into aircraft engine PHM, future research can develop hybrid fault-aware RUL models that not only predict failure time but also differentiate degradation sources, optimizing maintenance strategies and reducing downtime.

2.4 Selection of ML and DL models for RUL prediction

Traditional rule-based maintenance approaches have proven inefficient, leading to the adoption of data-driven methods using ML and deep DL. The effectiveness of an RUL prediction model depends on its ability to capture complex degradation trends, generalize across different failure conditions and handle multivariate sensor data. This section explores the selection and working mechanisms of ML and DL models for RUL estimation, focusing on regression-based approaches, feature extraction and hybrid model architectures.

2.4.1 Machine Learning Models for RUL Prediction

The study by Alomari, Mátyás Andó and Baptista (2023)[2] investigates the use of RF, XGB and MLP for predicting the RUL of aircraft engines. Their study highlights the importance of feature engineering and dimensionality reduction in improving ML model performance.

1. Random Forest (RF): It is an ensemble learning algorithm that constructs multiple decision trees during training and outputs the average prediction of the individual trees (as shown in figure 2-4).

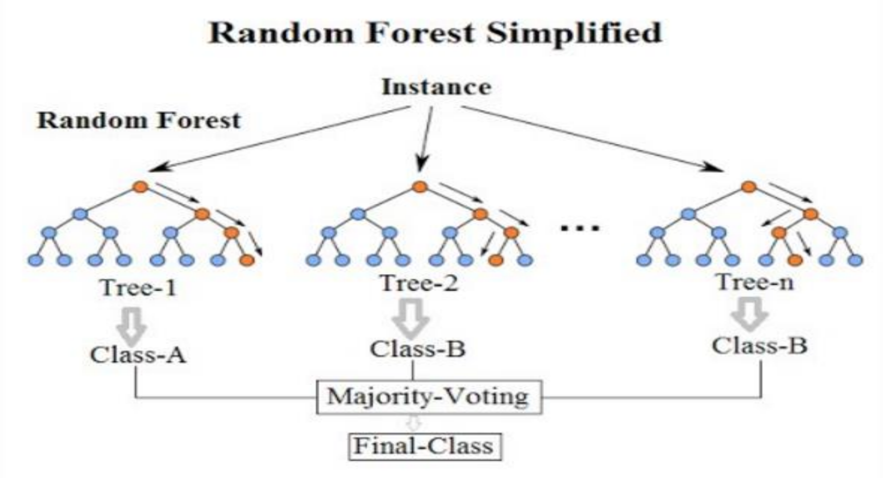


Figure 2-4: RF Model Structure [8]

It is effective in handling non-linear relationships and noisy sensor data. In the context of RUL prediction, RF models are trained on historical failure data, learning degradation patterns from engine sensor readings. The study[2] demonstrates that RF achieves reliable performance when combined with RFE for optimal feature selection. The key significance of using RF model predicting RUL is as follows:

- RF is effective in handling high-dimensional sensor data, making it useful for processing multivariate time series data from aircraft engines.
- It provides feature importance scores, allowing the identification of critical degradation indicators.
- RF is computationally efficient and resistant to overfitting, making it a preferred choice for real-time RUL estimation in industrial applications.

2. Extreme Gradient Boosting (XGB): It is an optimized gradient boosting algorithm designed for high-performance predictions. It operates by sequentially training decision trees, where each tree corrects the errors of its predecessor (shown in figure 2-5).

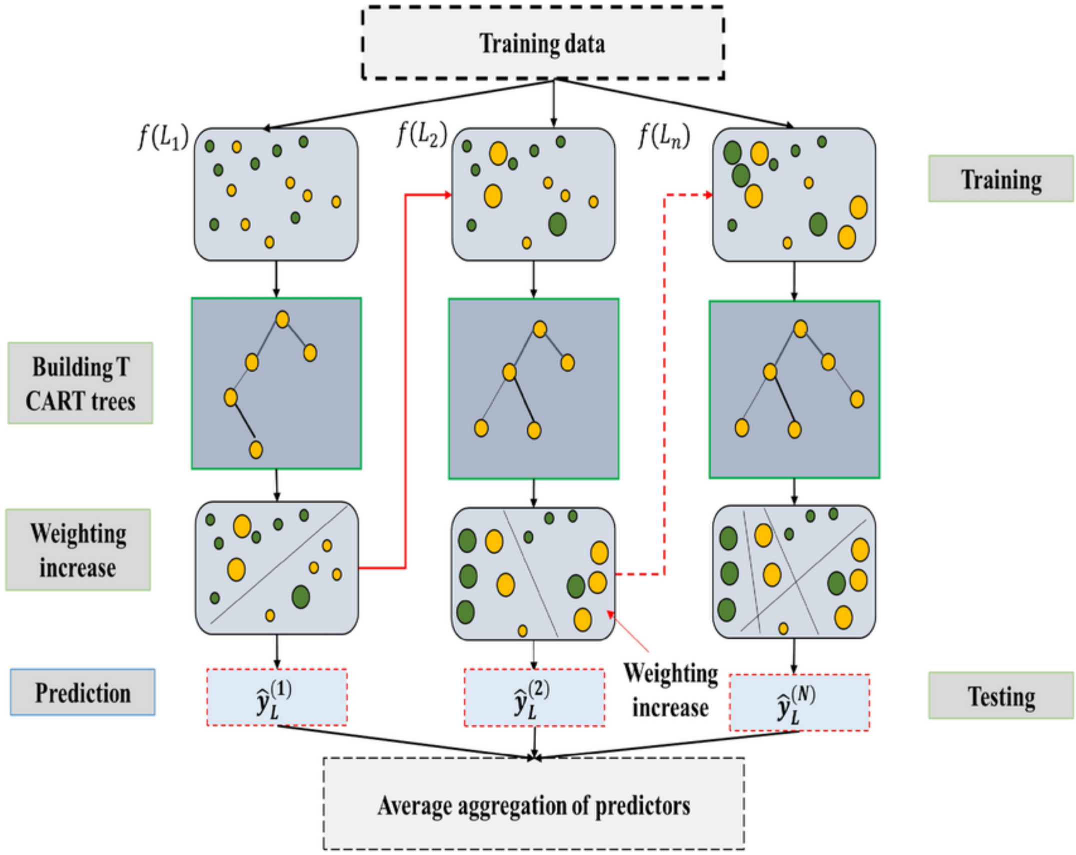


Figure 2-5: Graphical scheme of XGB model [22]

Compared to RF, XGB is more efficient in handling high-dimensional datasets and offers faster convergence. The study by Safavi *et al.* (2024)[14] finds that XGB with hyperparameter tuning (XGB-HT) outperforms other ML models in battery RUL prediction, achieving lower RMSE and MAPE. This suggests that gradient-boosting techniques are particularly well-suited for time-series degradation modelling. The key significance of using XGB model for RUL predictions is as follows:

- Its ability to capture long-term dependencies without requiring a deep architecture makes it useful for short-term RUL estimation.
- It has been widely used in feature engineering to extract relevant degradation trends from raw sensor data, as demonstrated by Alomari, Mátyás Andó and Baptista (2023)[2] in their study on aggregated feature importance.
- It outperforms traditional ensemble methods by handling missing values, sensor noise and non-linear degradation patterns effectively.

3. Light Gradient Boosting Machine (LGBM): It is an advanced gradient boosting algorithm designed for high efficiency and scalability. Unlike XGB, LGBM uses histogram-based binning (shown in figure 2-6), reducing computational overhead and improving training speed.

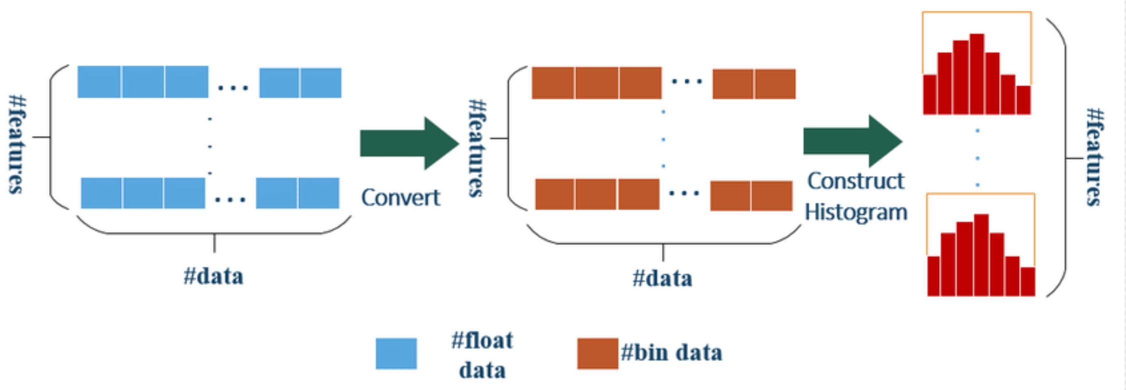


Figure 2-6: Histogram algorithm of LGBM [20]

The study by Safavi *et al.* (2024)[14] demonstrates that LGBM performs comparably to XGB in battery RUL prediction, with faster execution times while maintaining high prediction accuracy. Additionally, LGBM's ability to handle imbalanced datasets and missing sensor readings makes it highly suitable for real-time RUL estimation in industrial applications. The significance of LGBM model for RUL predictions is given below:

- Its improved computational efficiency and lower memory usage make it more suitable for RUL estimations without sacrificing accuracy.
- It has been widely used in scenarios requiring rapid model training and inference, as demonstrated by Safavi *et al.* (2024)[14], where LGBM outperformed XGB in handling high dimensional sensor data with faster execution times.
- It works well in managing imbalanced datasets and sparse features, making it highly effective for RUL prediction when sensor failures or missing values are present.
- Its histogram-based feature selection reduces noise more efficiently than XGB, leading to enhanced predictive stability in non-linear degradation patterns.

4. Multi-Layer Perceptron (MLP): MLP is a feedforward neural network that learns complex nonlinear relationships in sensor data. Alomari, Mátyás Andó and Baptista(2023)[2] show that MLP, when trained on PCA-selected features, improves RUL estimation accuracy. However, MLP struggles with time-series dependencies and requires hyperparameter tuning to avoid overfitting, making it less effective than deep learning models for sequential degradation modelling.

2.4.2 Deep Learning Models for RUL Predictions

Deep learning models have outperformed traditional ML techniques in RUL prediction due to their ability to capture long-term dependencies and extract hierarchical degradation patterns. The study by Huthaifa Al-Khazraji *et al.* (2022)[6] introduces a hybrid deep learning model (AE-DBN), combining AE for feature extraction and DBN for sequential prediction.

1. Long Short-Term Memory (LSTM): LSTM is a specialized RNN designed for long-term dependency learning. Unlike traditional RNNs, LSTM employs gates (input, forget and output) to regulate information flow, mitigating the vanishing gradient problem (shown in figure 2-7).

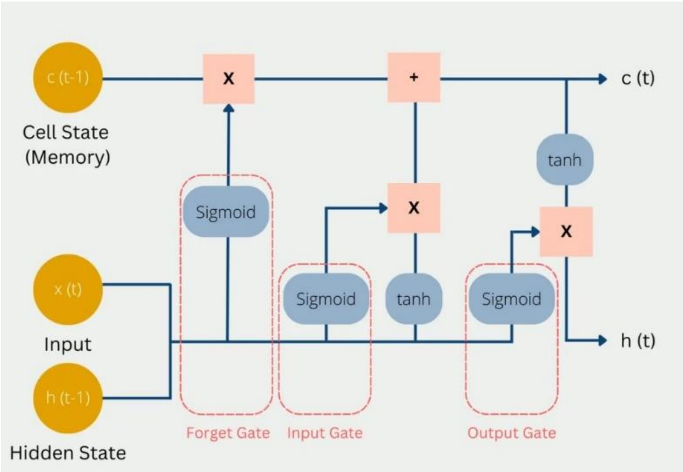


Figure 2-7: LSTM Architecture [3]

Alomari, Mátyás Andó and Baptista (2023)[2] demonstrates that Bidirectional LSTM (Bi-LSTM) and Deep LSTM models effectively capture engine degradation trends over time. The key significance of LSTM models are listed below.

- Unlike tree-based models, LSTM can learn sequential dependencies from degradation patterns, making it ideal for long-term RUL prediction.
- It is effective in handling irregular failure trajectories in complex systems, as shown in (Mohamed *et al.*, 2020)[11] where LSTM was combined with clustering to enhance generalization.
- LSTM-based models can incorporate real-time sensor inputs, making them well-suited for dynamic RUL estimation.

However, LSTM models require high computational resources and large training datasets, making them less practical for real-time RUL estimation in edge computing environments.

2. Gated Recurrent Unit (GRU):

GRU is a simplified version of LSTM that reduces computational complexity while maintaining accuracy. Unlike LSTM, which uses input, forget and output gates GRU has only an update and reset gate, making it faster to train (shown in figure 2-8).

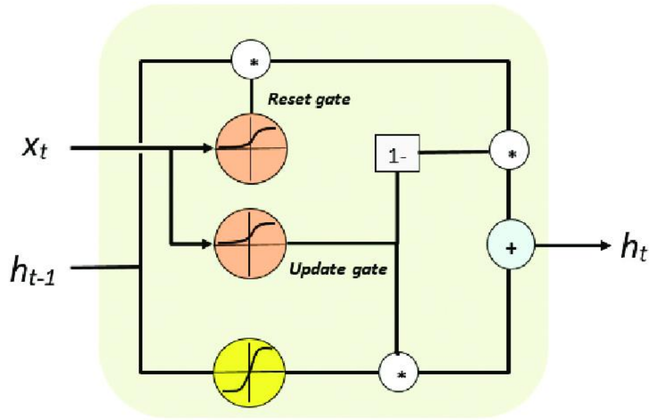


Figure 2-8: GRU architecture[12]

According to Adryan Fitra Azyus, Sastra Kusuma Wijaya and Mohd Naved (2023)[1], GRU is highly effective for time-series regression problems including RUL prediction for aircraft engines. Their study demonstrates that CNN-GRU outperforms standalone GRU models, as CNN extracts spatial degradation features, while GRU captures temporal dependencies. The CNN-GRU model successfully predicted turbofan engine degradation patterns using the C-MAPSS dataset, showing that GRU can handle sequential failure efficiently while reducing computational overhead compared to LSTM [1].

However, GRU has limitations in handling very long-term dependencies, making it less effecting than LSTM for extensive degradation monitoring. It also requires careful hyperparameter tuning to prevent overfitting on small datasets.

3. 1D Convolutional Neural Networks (Conv 1D): It is a variant of CNN that applies 1D convolutional filters to sequential data, making it highly effective for extracting short-term degradation patterns from sensor signals. Unlike CNNs, which operate on 2D image data, Conv1D is specifically designed for time-series forecasting, making it a powerful tool for RUL prediction (figure 2-9 shows a simple Conv 1D architecture).

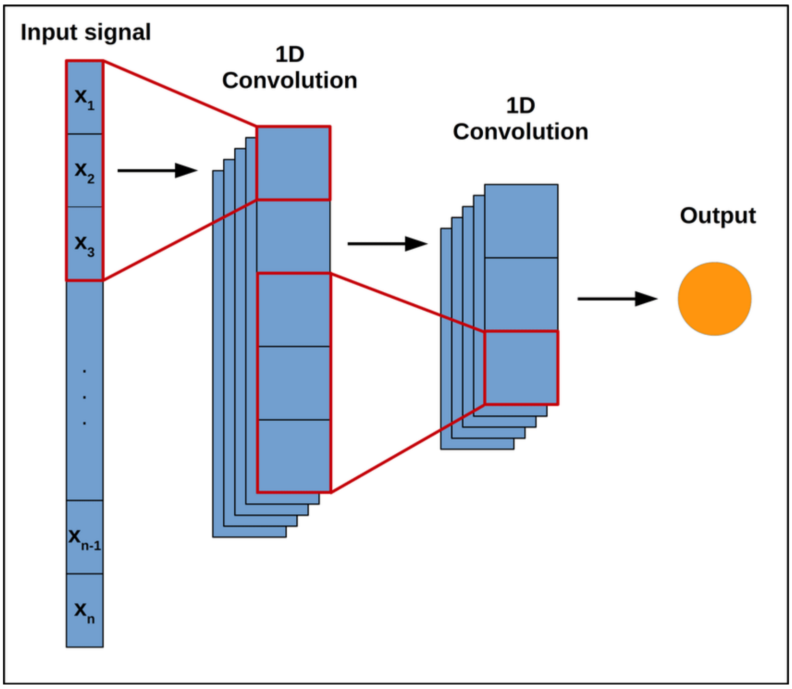


Figure 2-9: Simple Conv 1D architecture with two convolutional layers[16]

The Conv1D model applies convolutional filters over sensor readings to detect local failure trends, improving feature extraction before passing the data to temporal models like LSTM or GRU. Research has shown that Conv1D-LSTM and Conv1D-GRU hybrid models achieve higher accuracy than standalone RNNs, as they can capture both spatial and temporal dependencies [1]. Their research highlight that Conv1D enhances the CNN-GRU model by improving feature extraction from raw sensor signals, reducing preprocessing efforts. Their study on C-MAPSS datasets demonstrates that Conv1D-CNN-GRU provides better RUL estimates by leveraging hierarchical feature learning and sequential modelling.

4. Autoencoder-Based Deep Belief Network (AE-DBN): The AE-DBN hybrid model proposed by Huthaifa Al-Khazraji *et al.* (2022)[6] combines:

- AE which Compresses sensor data, filtering noise and preserving degradation features.
- DBN which uses stacked RBMs to model long-term failure trends.

AE-DBM outperforms traditional LSTM models, as it efficiently captures multi-scale degradation patterns. However, pretraining the AE separately before training the DBN is necessary for optimal performance.

2.4.3 Comparative Analysis of ML and DL Models

MODEL	STRENGTHS	LIMITATIONS
RF	Robust to noise, interpretable	Lacks sequential modelling
XGB	High accuracy, feature selection efficiency	Computationally expensive
LGBM	Fast training, handles missing data	Slightly less accurate than XGBoost
MLP	Captures nonlinear relationships	Overfitting risk, lacks time-series modelling.
LSTM	Captures long-term dependencies	Requires large training datasets
GRU	Faster than LSTM, effective for short-term RUL	Struggles with long-term dependencies
Conv1D	Extracts spatial degradation features	Less effective alone for long-term trends
CNN-GRU	Combines spatial and temporal learning	Requires large datasets for training
AE-DBN	Strong feature extraction and sequential modelling.	Requires pretraining, complex architecture.

Table 2-1: Comparison of ML and DL Models

The selection of ML and Dl models for Rul prediction depends on data complexity, model interpretability and real-time requirements. While traditional ML models like RF, XGB and LGBM are effective for structured datasets, deep learning models like GRU, LSTM and CNN-GRU provide advanced sequential degradation tracking. Future research should focus on interpretable AI techniques and hybrid ML-DL models.

2.5 Feature Engineering Techniques for Time Series Data Modelling

The quality of engineered features significantly impacts forecasting accuracy, anomaly detection and predictive performance in ML and DL models. Traditional statistical methods and modern machine learning techniques rely on effective feature extraction and transformation to capture temporal patterns, seasonality and trends.

2.5.1 Basic Feature Engineering Techniques for Time Series Data

1. Time-Based Features: One of the fundamental steps in feature engineering for time series data is extracting time-based features, such as hour, day, month, week, quarter and year components. These features help models recognize seasonal and periodic trends in time series data.

- Day of the week & month features: Useful for business or retail data, where sales patterns often follow weekly or monthly cycle [5].
- Holidays & special events: Adding holidays flags or external event indicators can enhance forecasting models, especially in e-commerce and stock market predictions [13].
- Cyclical encoding: Instead of treating time as a linear variable, encoding time-based features helps to preserve cyclic properties [9].

2. Lag Features: It capture previous time steps values, allowing models to learn temporal dependencies. Common techniques include:

- **Fixed Lags:** Using values from previous **n** time steps as new features (e.g., x_t-1, x_t-2).
- **Rolling window statistics**: Computing mean, standard deviation and min/max values over a defined window (e.g, last 10 cycles or 30 cycles).
- **Exponential weighted moving average (EWMA)**: Assigns greater importance to recent values useful in financial and sensor-based forecasting [13].

2.5.2 Advanced Feature Engineering Techniques

3. Frequency Domain Transformations (Fourier and Wavelet Transforms): Many time-series patterns are better analysed in the frequency domain rather than the time domain. Fourier Transform decomposes time-series signals into different frequency components, enabling the identification of seasonal and cyclical behaviours.

- **Fast Fourier Transform (FFT):** Useful for identifying dominant frequencies in periodic data.
- **Wavelet Transform (WT):** Unlike FFT, WT captures both time and frequency information, making it valuable for detecting non-stationary trends.

2.5.3 Machine Learning and Hybrid Feature Engineering Approaches

4. Principal Component Analysis (PCA): High-dimensional time-series datasets often contain redundant or noisy features. PCA helps to reduce dimensionality while retaining the most important variance in the data. It transforms high-dimensional datasets into a smaller number of uncorrelated variables, known as principal components, while retaining the most important variances in the data. This is particularly useful in time series modelling, where datasets often contain redundant noisy or highly correlated features.

- Feature selection using PCA can improve the performance of ML models by removing correlated and less informative features.
- T-SNE and UMAP are alternatives for visualizing complex time-series feature spaces.

5. Combining Statistical and Learned Features: Modern prediction models often combine handcrafted statistical features with learned features from deep learning models to enhance predictive accuracy. This hybrid approach leverages the strengths of both traditional statistical methods (e.g., rolling statistics and frequency-domain transformations) and data-driven feature extraction from neural networks (e.g., embeddings from LSTMs or Transformers). By integrating these features, models can capture short-term fluctuations, long-term trends, seasonality and complex temporal dependencies.

Feature engineering is a crucial step in time series modelling, influencing the effectiveness of both statistical and machine learning-based forecasting models. Techniques such as time-based encoding, lag features, rolling statistics and frequency transformations help extract meaningful patterns from raw time series data. Recent advancements hybrid models have further improved predictive performance. Future research should focus on automated feature engineering using AI-driven approaches, enabling more efficient and scalable time series analysis.

2.6 A Condensed Summary of the Literature Review

The literature review conducted for this study explores key research and methodologies related to damage propagation, fault differentiation and RUL prediction in aircraft engines. The review focuses on five critical areas within PHM, emphasizing the role of ML and DL models in predictive maintenance. The insights gained for these studies provide a foundation for developing more accurate, interpretable and fault-specific RUL prediction models. Table 2-2 summarizes the key findings from the literature review.

TOPIC	KEY INSIGHTS
Damage Propagation Modelling in Aircraft Engines	➢ Faults in HPC and Fan components follow distinct degradation patters, impacting engine performance differently. ➢ Degradation often follows exponential trends, necessitating sensor-based monitoring for early detection. ➢ Physics-based models (Arrhenius, Coffin-Manson) are useful but limited by real-world uncertainties.
Sensor-Based Monitoring for Fault Diagnosis	➢ Multi-sensor data is crucial for tracking wear and tear in engine components. ➢ Sensor fusion techniques improve diagnostic accuracy by combining multiple sensor signals into a single health index.
Clustering and Fault Differentiation for RUL estimation	➢ Unsupervised clustering (e.g K-Means) enables differentiation between HPC degradation and Fan degradation. ➢ Clustering-based pre-processing enhances fault-aware RUL modelling, reducing noise in training data. ➢ Correlation-based clustering(Yoo, 2020)[21] helps preserve physical relationships between sensor variables.
Selection of ML and DL models for RUL Prediction	➢ RF and XGBoost effectively handle high-dimensional sensor data and are useful for feature selection. ➢ LSTM and GRU models capture long-term dependencies in degradation patterns, improving sequential RUL prediction.
Feature Engineering Techniques for Time Series Data Modelling	➢ Lag features and rolling statistics help in capturing temporal dependencies in degradation trends. ➢ Principal Component Analysis (PCA) reduces dimensionality while retaining critical fault- related degradation information.

	➢ Wavelet transforms and exponential weighted averages are effective for noise reduction and trend extraction.

Table 2-2: Key Insights from the Literature Review

Chapter 3 – System Model and Degradation Analysis

The system model utilized for this research is based on the C-MAPSS dataset, a widely used benchmark for aircraft engine prognostics. C-MAPSS provides a realistic simulation of a large commercial turbofan engine, enabling researchers to study engine degradation under various operational conditions. This section discusses the system model used for tracking damage progression in aircraft engine turbomachinery. The approach follows a methodology that relies on response surfaces generated through the simulation models.

The model represents an engine with 90,000lb thrust capacity and accounts for altitude variations (sea level to 42,000 ft), Mach numbers (0-0.84) and TRA (20 – 100).C-MAPSS operates in either open-loop or closed-loop configurations, with the latter including controllers for fan speed, pressure and turbine temperature regulations. The control system prevents exceeding design limits, ensuring accurate degradation simulation.

3.1 Turbofan Engine Architecture

The turbofan engine consists of several key components that work together to generate thrust efficiently. The Fan is the first stage, responsible for drawing in ambient air and increasing mass flow, contributing significantly to thrust production. The air then passes through the LPC, which raises its pressure before sending it to the HPC, where it undergoes further compression to enhance combustion efficiency. In the Combustor, the highly compressed air is mixed with fuel and ignited, producing high-energy gases that drive the turbines. The HPT extracts energy from these hot gases to power the HPC, while the LPT extracts the remaining energy to drive both the LPC and the fan. Finally, the exhaust gases are expelled through the nozzle, generating thrust and propelling the aircraft forward. The simplified diagram of the simulated engine and the connection between various modules as modelled in the simulation is shown in figure 3-1 and figure 3-2.

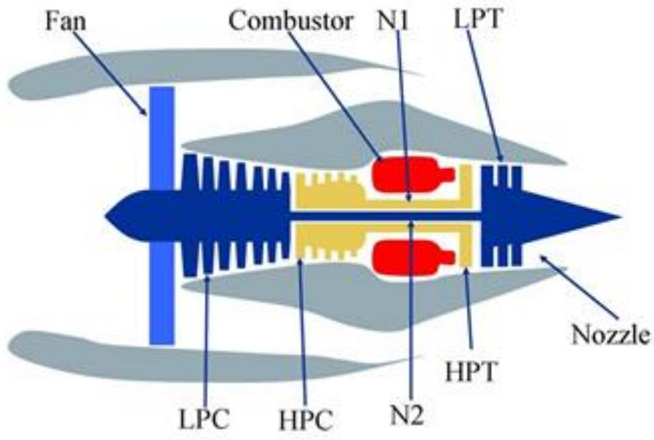


Figure 3-1: Simplified diagram of the simulated engine[15]

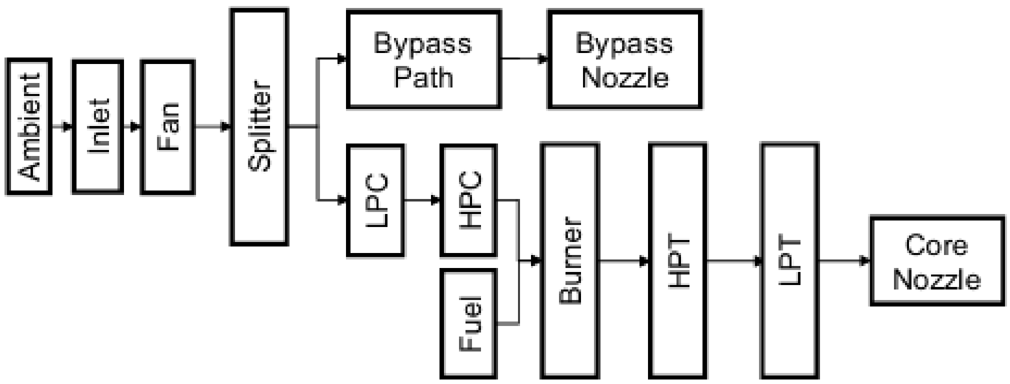


Figure 3-2: A Layout showing various module connections as modelled in the simulation[15]

The bypass flow and core flow are separately modelled. Various inputs such as fuel flow rate and efficiency/flow modifiers for rotating components allow simulation of degradation scenarios. The simulation generates multiple sensor outputs, which serve as performance indicators (shown in table 3-1). These include:

- Temperature and pressure measurements at different engine stages.
- Physical and corrected speeds of fan and core components.
- EPR and fuel-air ratio, crucial for performance assessment.
- Bleed flow energy.
- Health indices, including stall margins and EGT (not given).

SYMBOL	DESCRIPTION	UNITS
T2	Total temperature at fan inlet	^{0}R
T24	Total temperature at LPC outlet	^{0}R
T30	Total temperature at HPC outlet	^{0}R
T50	Total temperature at LPT outlet	^{0}R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNFR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

Table 3-1: Parameters as sensor data (C-MAPSS outputs)[15]

3.2 Exponential Damage Growth Model

The damage propagation model used in C-MAPSS is based on an exponential degradation function that captures the way faults evolve in aircraft engines. The approach is inspired by real-world degradation trends and aims to provide a physics-based model for wear and tear in engine components.

The degradation of an aircraft engine component due to wear is represented using an exponential function:

$$\omega = Ae^{B(t)}$$

Where:

- ω represents the accumulated wear.
- A and B are parameters that define the degradation rate over time.
- t represents time.

Since each engine component has an upper wear threshold (th_ω), beyond which it is considered unusable, the health of the component over time is defined as:

$$h(t) = 1 - Ae^{B(t)} / th_\omega$$

To simplify, parameters are redefined as:

$$A/th_\omega = e^{a}, B(t) = tb$$

Thus, the equation for health index (HI) becomes:

$$h(t) = 1 - e^{atb}$$

where:

- a and b are degradation parameters
- The health index starts at 1 (fully healthy) and decreases as wear accumulates.

In reality, engine components may not start in a fully healthy state due to initial wear and manufacturing inefficiencies. This initial degradation, d, is incorporated into the model:

$$h(t) = 1 - e^{atb} - d$$

Each engine module (e.g., compressor, turbine) is evaluated using two parameters. The efficiency (e) which determines how well the component performs and flow (f) which determines the amount of air passing through the component. Thus, these parameters follow the same exponential degradation pattern:

$$e(t) = 1 - d_e - e^{atbe}$$

$$f(t) = 1 - d_f - e^{atbf}$$

The overall health index $H(t)$ for the engine is determined as:

$$H(t) = g[e(t),f(t)]$$

Where g is a function that takes the minimum of various component margins,

$$g(e,f) = \min(m_{fan}, m_{HPC}, m_{HPT}, m_{EGT})$$

The overall health index of an aircraft engine, denoted as H(t), is determined by evaluating the operability margins of critical components such as the fan, HPC, HPT and EGT. The degradation of these components follows an exponential pattern, meaning that as wear accumulates over time, the efficiency and flow properties of these parts deteriorate. The function governing H(t) takes the minimum value among these operability margins, indicating that the most degraded component dictates the overall health of the engine. When this health index reaches zero, the engine is considered to have failed.

3.3 Application Scenario

This study is based on the PHM'08 data challenge, where multiple aircraft engines are tracked throughout their operational lifecycle. The primary objective is to estimate the RUL of engines based on historical sensor data with a focus on predictive maintenance.

1. Initial Wear:

To create the C-MAPSS dataset, initial wear conditions were assigned to engine components, considering that real-world manufacturing imperfections often introduce minor inefficiencies. Components start with slight deviations from their ideal state, typically within a 1% degradation range. Over time, the degradation accelerates, with efficiency and flow values decreasing based on pre-defined degradation models. The dataset includes wear conditions for 3000 and 6000 cycles showing how different components degrade at varying rates (shown in table 3-2). For example, fan efficiency drops by approximately 1.5% after 3000 cycles and 2.85% after 6000 cycles, while HPC efficiency degrades by 1.46% and 2.61% over the same periods.

COMPONENT	INITIAL WEAR (%)	WEAR 3000 CYCLES (%)	WEAR 6000 CYCLES (%)
Fan Efficiency	-0.18	-1.5	-2.85
Fan Flow	-0.26	-2.04	-3.65
LPC Efficiency	-0.62	-1.46	-2.61
LPC Flow	-1.01	-2.08	-4.00
HPT Efficiency	-0.48	-2.63	-3.81
HPT Flow	+0.08	+1.76	+2.57
LPT Efficiency	-0.10	-0.54	-1.08
LPT Flow	+0.08	+0.26	+0.42

Table 3-2: Engine wear as manifested in flow and efficiency changes[15]

2. Noise Considerations:

Noise plays a crucial role in making the dataset realistic. Various sources of noise, including manufacturing variations, process disturbances and sensor measurement errors, introduce unpredictability into the data. To simulate these uncertainties, noise was incorporated at different stages, such as initial efficiency and flow values, degradation trajectory parameters and final sensor readings. A mixture of distributions was used to ensure the noise characteristics resembled real-world scenarios. This approach accounts for random fluctuations and occasional improvements due to maintenance actions, allowing for a more comprehensive understanding of degradation trends. Therefore, the steps involved in data generation are,

- Choose initial values e_0 , f_0 randomly in [0.99, 1].
- Simulate exponential degradation for efficiency and flow.
- Stop simulation when H = 0 (engine failure).
- Add measurement noise to sensor data.

3. Health Index Calculation:

The health index is calculated based on operability margins, n. These margins include constrains such as core speed limits, EGT limits and stall margins for the fan, HPC and LPC. The final health index is normalized and is based on the worst-performing component. If any margin drops below the predefined critical threshold, the engine is deemed to have failed. These margins vary as degradation takes place and also change as a function of operational conditions (e.g. TRA, altitude, ambient temperature, etc.) as shown in figure 3-3. For instance, a 15% stall margin loss was considered critical for the HPC, LPC and fan while a 2% loss in EGT was sufficient to indicate failure.

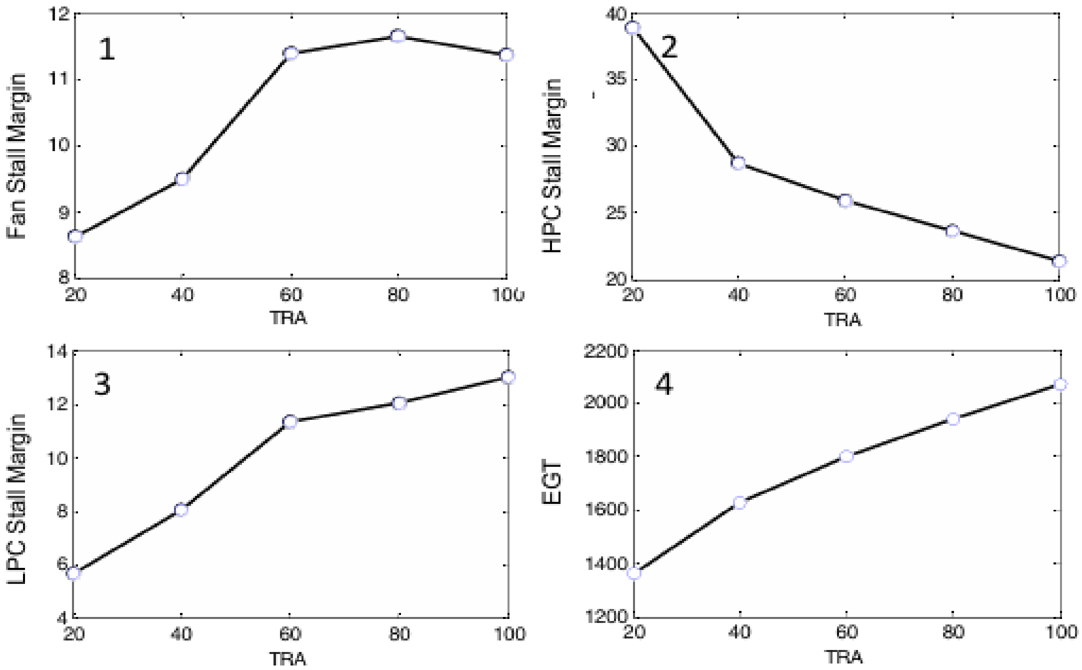


Figure 3-3: Stall margins vary as a function of operational conditions (TRA in the above example) [15]

For the given dataset, the degradation in the HPC module was simulated under six different flight conditions that comprises of a range of values for the three operational conditions: altitude (0 – 42k ft), Mach number (0 – 0.84) and TRA (20 – 100). Sensed margins (fan, HPC, LPC and EGT) were used to compute health index to determine the simulation stopping criteria. Time series of observables (see table 3-1) including the operating conditions were collected and divided into training set, test set and validation set. The training set had trajectories that ended at the failure threshold while the test and validation sets were pruned to stop some time before the failure threshold. The range of RUL variation was expanded for the validation dataset to test robustness of the algorithms on the test data. The test data set RULs ranged between 10 and 150 cycles, whereas validation RULs ranged between 6 and 190 cycles.

3.4 Key Takeaways

- Degradation follows an exponential model and affects efficiency and flow.
- Health index H(t) is derived from the worst-performing component.
- Noise is incorporated at different stages to make the data more realistic.
- The C-MAPSS dataset models gradual wear and tear, rather than sudden failures.
- Maintenance effects are indirectly modelled through noise, allowing occasional improvements in efficiency.

Understanding this degradation model is vital for predicting the RUL of aircraft engines. By leveraging this model, predictive maintenance systems can be developed to assess engine health in real time. Machine learning models trained on the C-MAPSS dataset can learn to distinguish normal wear patterns from abnormal degradation, allowing for more accurate predictions of engine failure. Additionally, the incorporation of realistic noise ensures that the models generalize well to real-world conditions, making them suitable for deployment in operational aircraft monitoring systems.

Chapter 4 – Fault Clustering

4.1 Single Operating Condition Selection for Fault Clustering

Fault clustering in aircraft engines is inherently complex due to the variability in operability margins across different operating conditions. Operational settings including TRA, Mach number and Altitude significantly influence sensor readings and engine degradation characteristics. Separating and normalizing the effects of these operational factors requires high domain expertise and extensive preprocessing to ensure meaningful fault differentiation. Therefore, this study focuses on fault clustering under a single operating condition (Sea Level) for the following reasons:

1. Minimizing the influence of Operating Conditions on Fault Signatures:

Variations in TRA, Mach number and altitude introduce additional complexity in clustering, as the same fault can exhibit different sensor trends under different conditions. By limiting the analysis to Sea Level conditions, we ensure that observed sensor deviations are primarily attributed to fault progression rather than operational variability, leading to more reliable fault differentiation.

2. Ensuring Sensor Data Consistency for Clustering Algorithms:

Clustering algorithms, especially distance-based methods like K-Means and DBSCAN, are sensitive to feature scaling and distribution. Operating conditions introduce non-stationary effects that make clustering less effective unless extensive feature engineering and domain-specific transformations are applied. Restricting the analysis to a single operating condition ensures a consistent feature space, allowing the clustering model to capture fault-specific degradation trends more accurately.

3. Reducing the Need for High-Domain Knowledge in Data Preprocessing:

Separating and aligning data across six operating conditions require an in-depth understanding of engine physics, aerodynamics and operational limits which is beyond the scope of this study. Instead of applying complex transformations to compensate for operational variability, using a single operating condition naturally eliminates these variations, making the clustering results more interpretable and actionable.

4. Enhancing Fault Differentiation Between HPC and Fan Degradation:

The primary objective of this clustering analysis is to differentiate between HPC degradation and Fan degradation. Using multiple operating conditions could introduce overlapping patterns that obscure the distinction between these two fault modes, making clustering less effective.

5. Aligning with Existing Literature on Clustering-Based Fault Analysis:

Previous studies on unsupervised clustering for fault diagnostics have demonstrated that clustering is more effective when external operational influences are minimized (e.g. (Yoo, 2020) [21]). Similar approaches in PHM suggests that clustering within a controlled operational regime leads to better fault separation and improved RUL prediction.

Therefore, by performing fault clustering under a single operating condition (Sea Level), this study ensures that clustering results are driven by fault characteristics rather than operational variations. This approach simplifies the clustering process, improves fault differentiation accuracy and enhances the interpretability of results, making it a practical and effective choice for unsupervised fault analysis in aircraft engines.

4.2 Clustering Analysis for Fault Differentiation

This section presents the clustering-based fault analysis conducted on aircraft engine sensor data from the C-MAPSS dataset. The primary objective is to differentiate between High-Pressure Compressor (HPC) degradation and Fan degradation using unsupervised learning techniques. The analysis focuses on two datasets:

- FD001: Contains engines experiencing only HPC degradation under a single operating condition (Sea Level).
- FD003: Includes engines with both HPC and Fan degradation, also under Sea Level Conditions.

By applying K-Means clustering this study aims to identify degradation patterns and explore how fault propagation varies across engine components.

1. Data Preprocessing and Feature Selection: To perform clustering effectively, the analysis begins with loading and preprocessing the datasets. Since the C-MAPSS dataset contains multiple sensor readings, a subset of nine critical features related to HPC and Fan degradation is selected, including temperature, pressure, speed and BPR. These features are chosen based on basic domain knowledge and their known influence on fault progression. Before applying clustering, the data is standardized using StandardScaler, ensuring tat all sensor values are on a comparable scale. This is crucial for distance-based clustering algorithms like K-Means, which are sensitive to differences in feature magnitudes.

2. Determining the Optimal Number of Clusters: To identify the optimal number of clusters, the Elbow Method is applied. This involves:

- Running K-Means clustering with varying number of clusters (k = 2 to 10).
- Recording the inertia (within-cluster sum of squared distances) for each k-value.
- Plotting an Elbow Curve to determine where the inertia starts to level off.

The elbow curves of both the fault conditions are shown in figure 4-1.

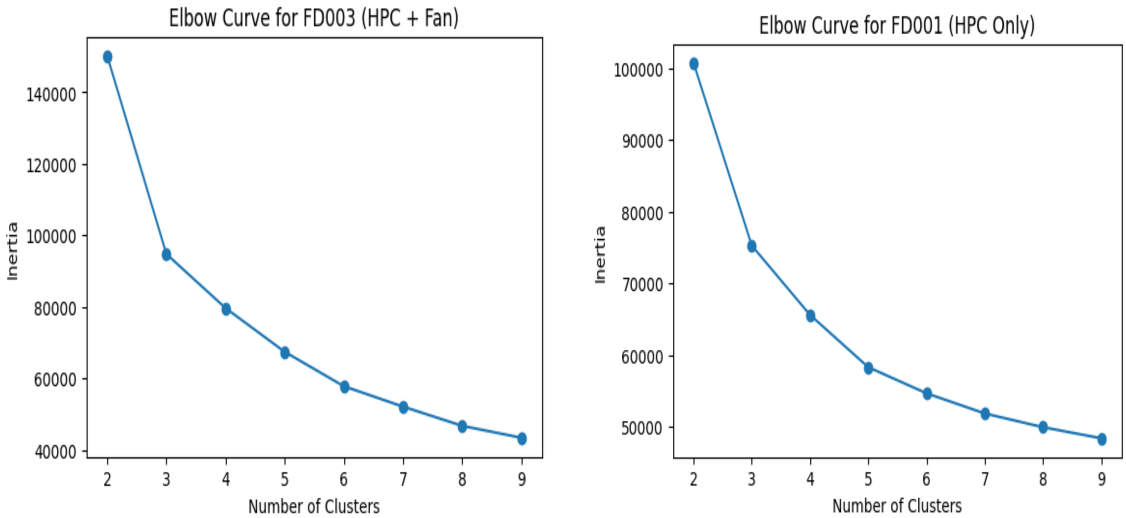


Figure 4-1: Elbow curves for the fault conditions

From the curves it is clear that the optimal number of clusters for both the fault condition is three. The Elbow Curve thus, helps in identifying the most natural grouping of data points, ensuring that the clusters effectively separate degradation trends.

3. Clustering with K-Means: Once the optimal number of clusters is determined, K-Means clustering is applied to the sensor data for both FD001 and FD003. This unsupervised algorithm groups data points based on their similarity in feature space, allowing for data-driven fault differentiation without prior labelling. Each engine cycle is assigned a cluster label, representing different degradation stages or fault progression patterns. The resulting clusters provide insight into how faults evolve over time in each dataset. To analyse how engine degradation progresses, clusters are visualized over engine cycles using scatter plots (shown in figure 4-2, figure 4-3).

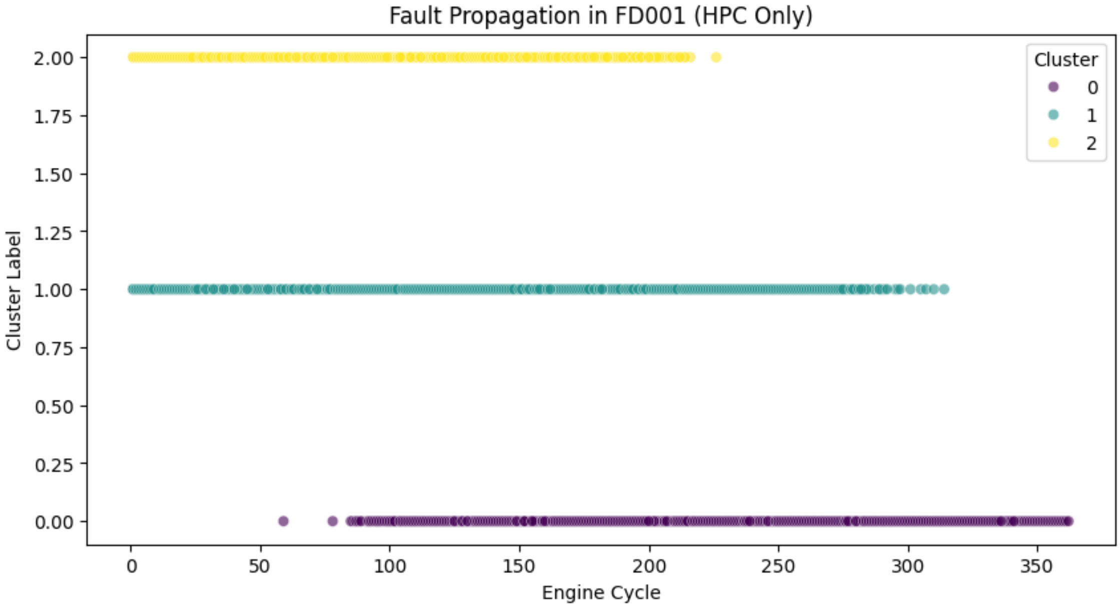


Figure 4-2: Fault Propagation Over Time across Clusters (FD001)

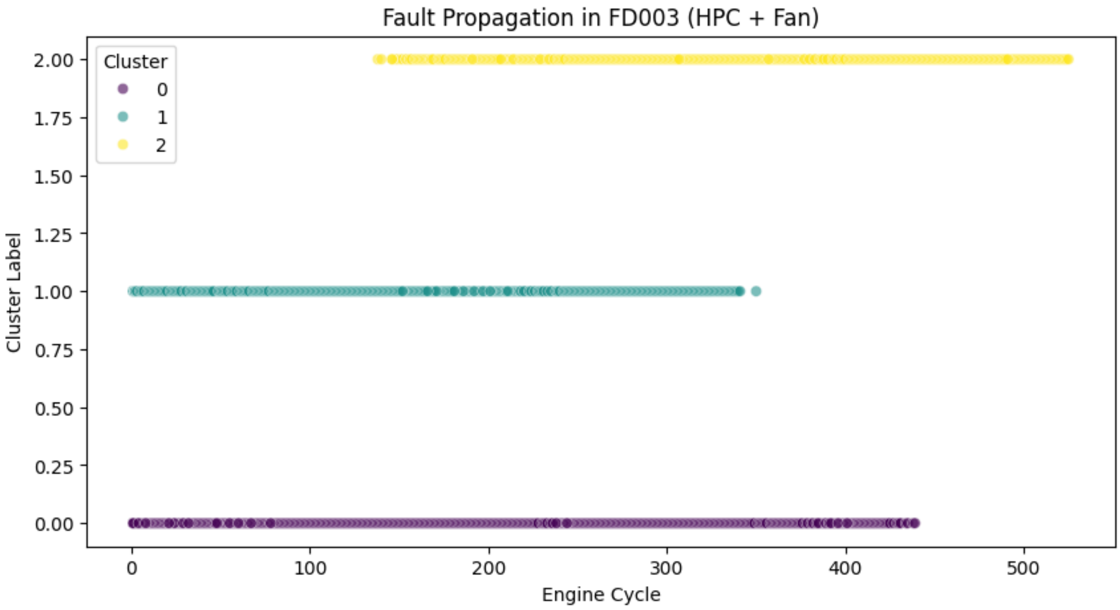


Figure 4-3: Fault Propagation Over Time across Clusters (FD003)

In FD001 dataset, three distinct clusters are evident, representing different stages of degradation. Cluster 1 in FD003 and cluster 2 in FD001 are concentrated in the early engine cycles, indicating a healthy or slightly degraded state. As the engine experiences wear, it transitions to cluster 0 and cluster 1 respectively, which occupies the mid-stage, suggesting a moderate shift in fault characteristics. Eventually as degradation becomes severe the engine enters cluster 2 for FD003 and cluster 0 for FD001 which appears at later cycles, likely corresponding to a critical failure state.

4. PCA Visualization: Since the original dataset contains multiple sensor features, PCA is applied to reduce dimensionality and visualize clusters in two-dimensional space. PCA helps in simplifying complex, high-dimensional data while preserving key patterns, providing a clearer view of fault separation in the clustered data and validate whether clusters are well-defined and physically meaningful. Figure 4-4 depicts how sensor readings vary across different fault clusters, allowing for a more intuitive interpretation of engine degradation trends.

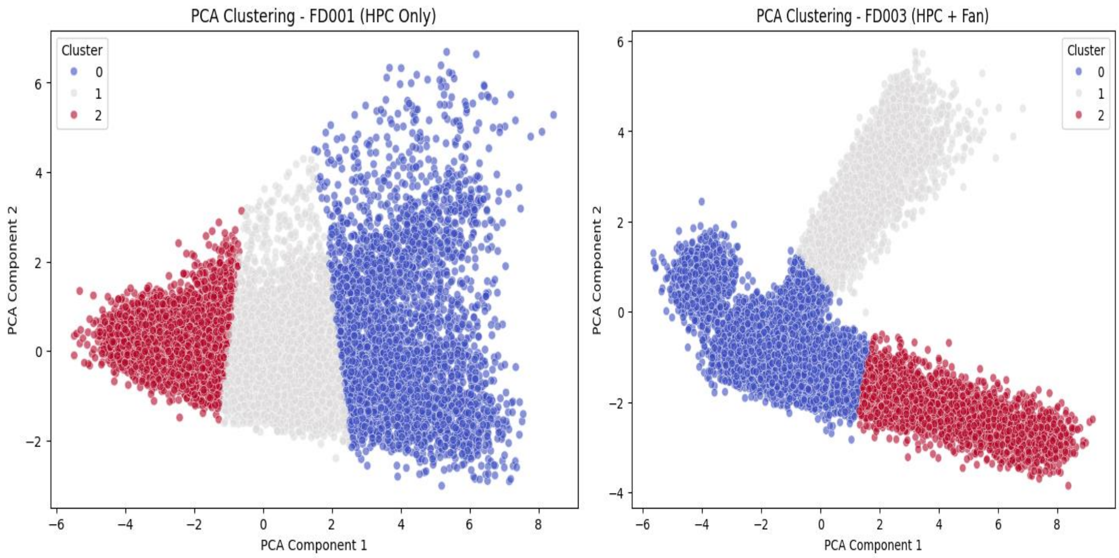


Figure 4-4: Sensor readings variations across clusters

In FD001, the degradation is more gradual and progresses in a linear direction as there is only HPC degradation. Whereas in FD003, the separation between the clusters is more pronounced, indicating that the introduction of Fan degradation creates additional variations in sensor readings. Thus, from these plots it is clear that:

- The cluster analysis successfully separates different health states in both datasets.
- In FD003, two different failure modes emerge, suggesting that Fan degradation introduces a different degradation pattern.
- In FD001, the failure progression is more continuous, as only one degradation mode exists.
- This visualization confirms that sensor data can be useful in identifying failure progression and distinguishing between fault conditions.

4.3 Sensor Trend and RUL Distribution Analysis Across Clusters

This section presents the sensor trend analysis and RUL distribution across different clusters obtained from the fault clustering analysis. The primary objective is to assess how sensor readings evolve over engine cycles for different fault conditions and examine how RUL is distributed within each identified cluster. This analysis helps in understanding degradation patterns, verifying the effectiveness of clustering in distinguishing fault modes and exploring how sensor trends correlate with RUL.

4.3.1 Sensor Trend Analysis

The sensor trends are visualized for FD001 and FD003 datasets using line plots of selected sensor measurements over engine cycles, grouped by cluster labels. The purpose of this analysis is to determine whether clustering effectively separates engine with different degradation profiles (fan and HPC), observe if specific sensor trends change significantly across clusters, which would indicate distinct fault progression patterns and analyse how fault propagation differs between FD001 and FD003. Figure 4-5, figure 4-6, figure 4-7 and Figure 4-8 shows sensor trends over engine cycles, with different clusters represented by colour coded lines in FD001 and FD003.

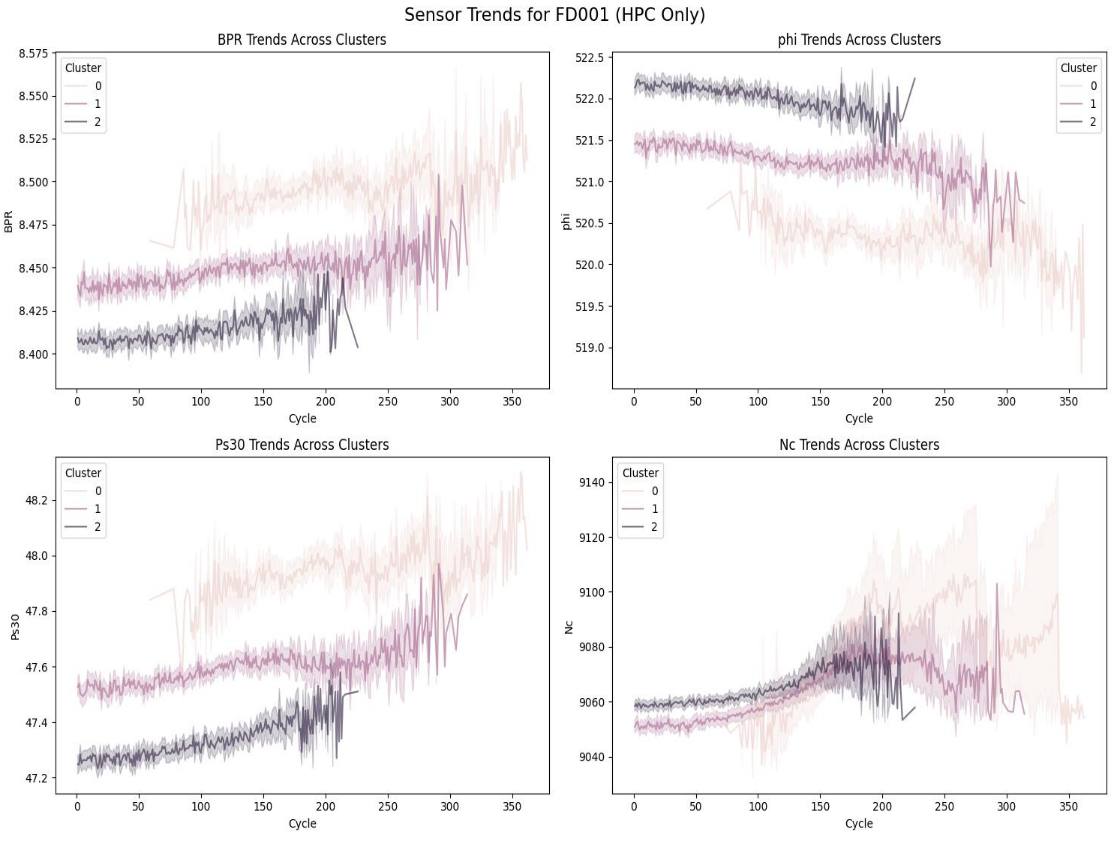


Figure 4-5: Sensor trends for FD001 (BPR, phi, Ps30 and Nc)

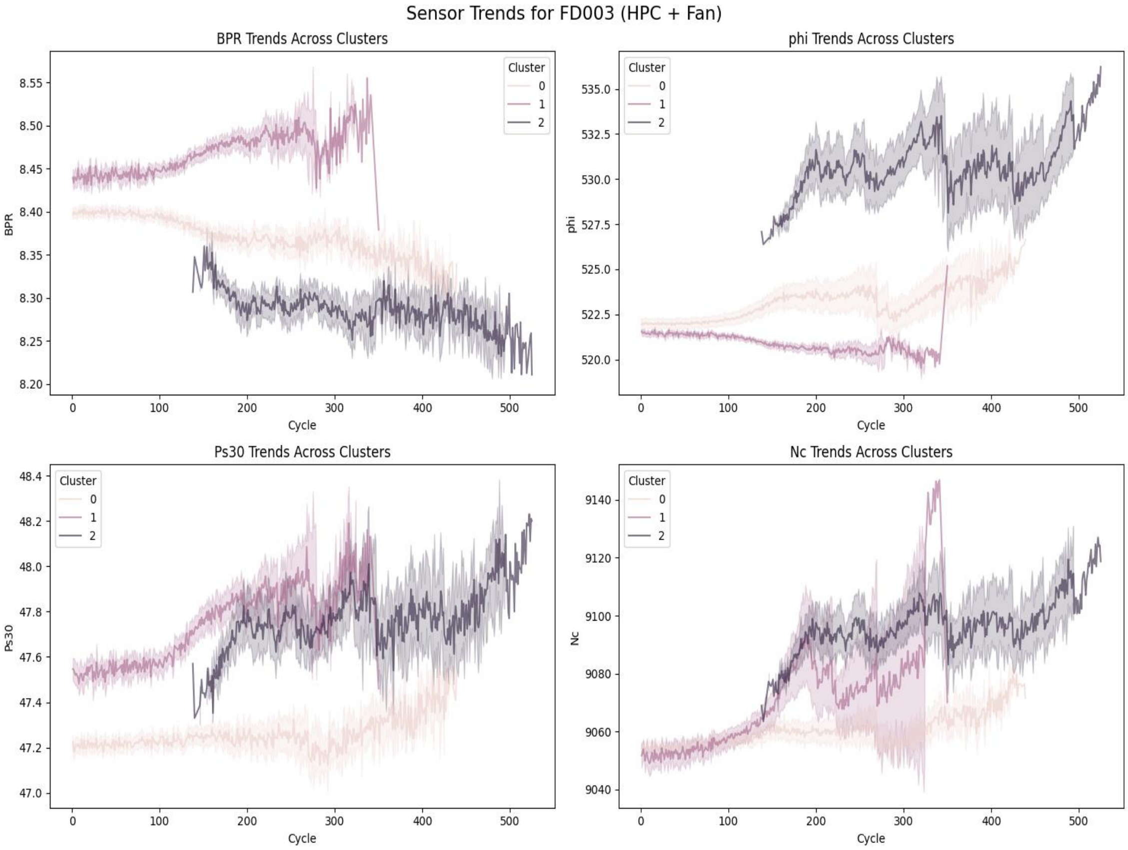


Figure 4-6: Sensor trends for FD003 (BPR, phi, Ps30 and Nc)

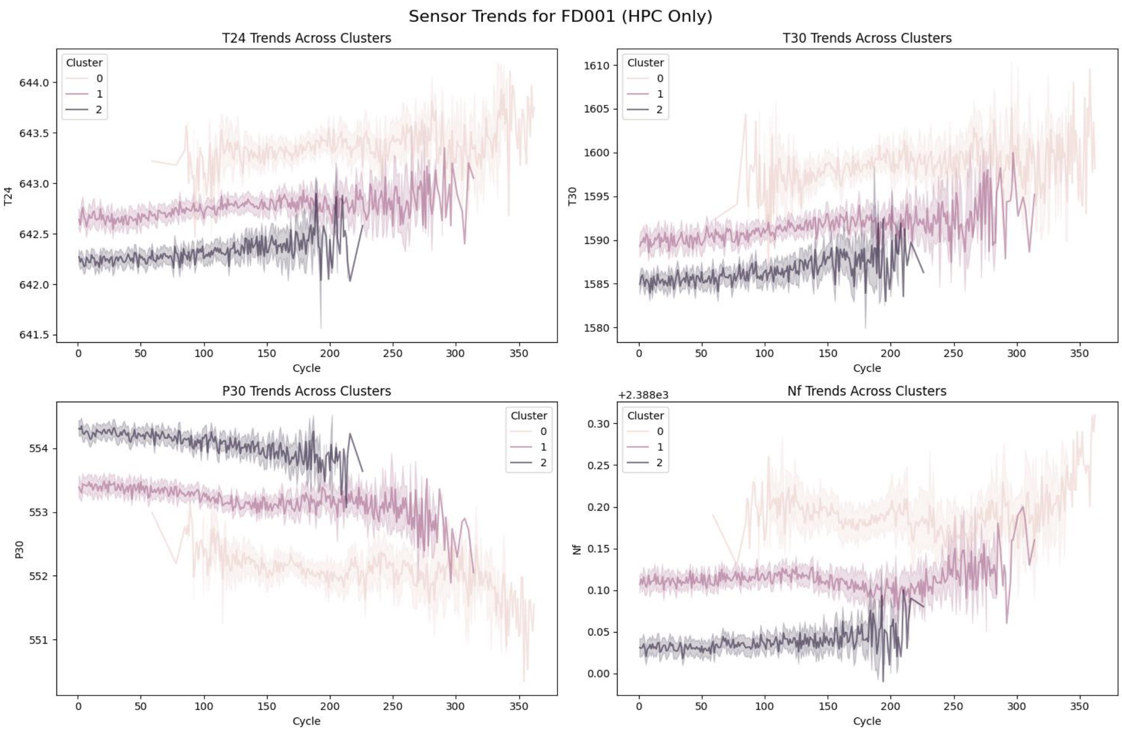


Figure 4-7: Sensor trends for FD001 (T24, T30, P30 and Nf)

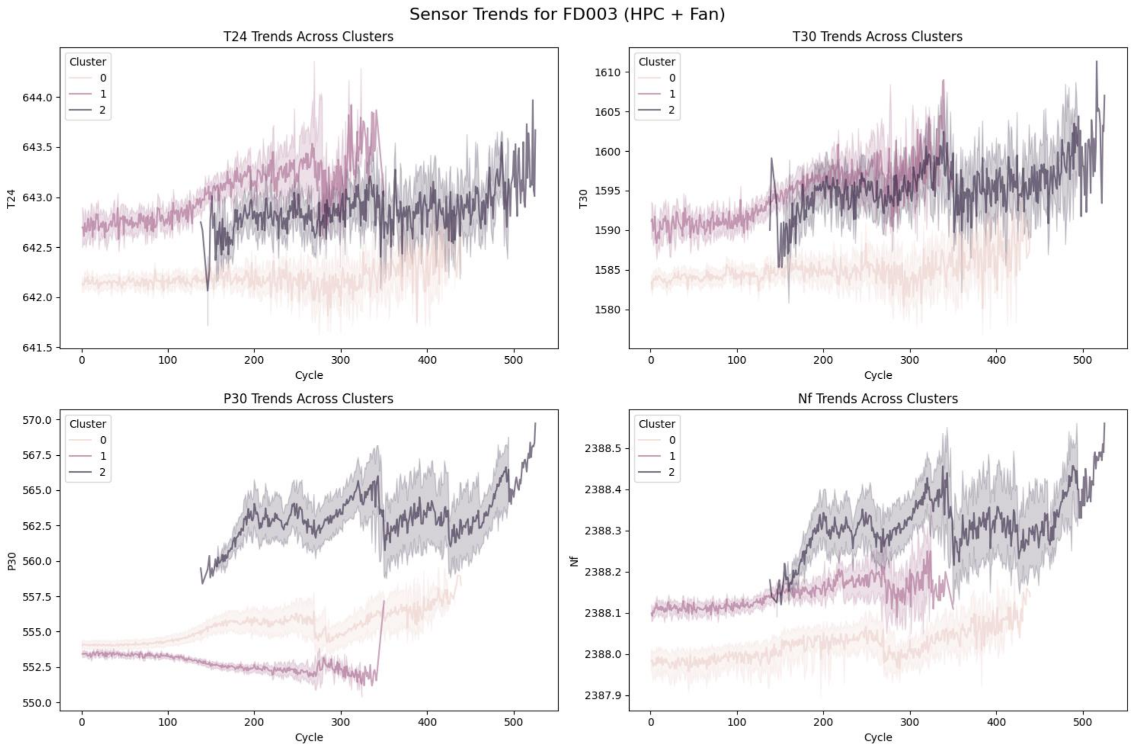


Figure 4-8: Sensor trends for FD001 (T24, T30, P30 and Nf)

In FD001, the variations are relatively smoother across cycles. The trendlines remain somewhat stable before degradation accelerates. Clusters appear closer together, suggesting a more gradual fault progression. Whereas in FD003, more abrupt changes and fluctuations, especially in phi and Nc. The separation between clusters is more pronounced, indicating different degradation modes. The effect of fan degradation is evident, making the trends more irregular. The key takeaways from the plots are listed as follows:

In FD001,

- Smooth degradation trends with gradual shifts.
- Less fluctuations especially in Nc, P30, Ps30 and Nf.

In FD003,

- Erratic shifts in the sensor measurements suggest-fan related failures.
- Sharper drops in BPR, phi and P30, higher fluctuations in Ps30 and steeper rises in Nc and T30.
- Clusters are more distinct, meaning different engine degrade in different patterns.

The clustered trends help to categorize degradation types: engines with similar degradation paths are grouped. HPC-only degradation is more predictable, while fan + HPC faults introduce greater variability. A sudden drop in phi and an increase in Nc is a red flag for fan degradation. Monitoring cluster separations over time can improve fault detection models for RUL prediction.

4.3.2 RUL Distribution Across Clusters

To further evaluate the effectiveness of clustering, the distribution of RUL is analysed across different clusters. The RUL values are calculated for each engine using:

$$\textbf{RUL = Max Cycles for Each Engine – Current Cycle}$$

A histogram of RUL values is generated for each dataset, where different clusters are represented using distinct colours. The purpose of this RUL distribution analysis is to examine if clusters are associated with progressive degradation stages, where some clusters correspond to engines closer to failure (lower RUL), validate whether engines with severe degradation are grouped separately from those in early stages of fault progression and explore differences in RUL trends between HPC-only degradation (FD001) and combined HPC + Fan degradation (FD003). Figure 4-9 shows the RUL distributions for FD001 and FD003.

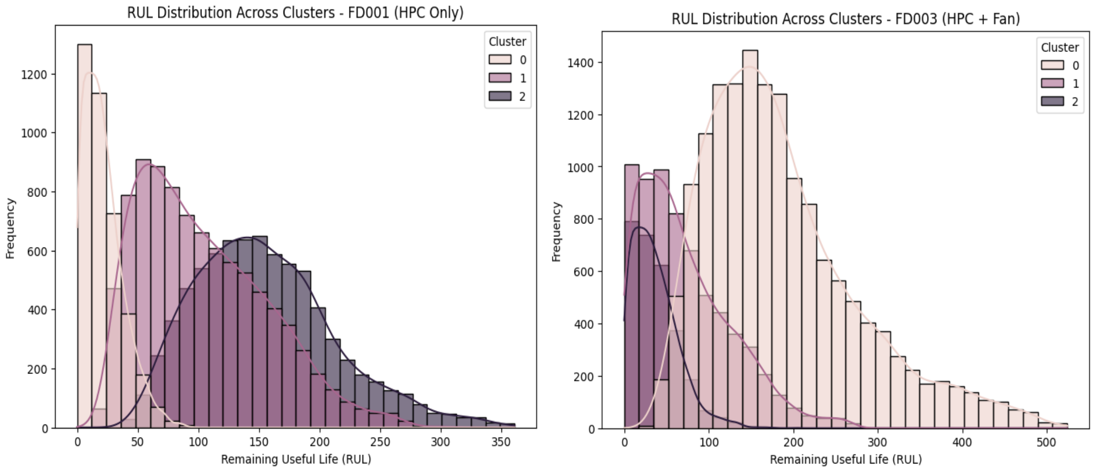


Figure 4-9: RUL distributions for FD001 and FD003

1. FD001 (HPC Only):

The distribution follows a more structured progression, with clear peaks at different degradation stages. Cluster 0 (Severe Degradation) peaks near 0 -50 cycles, indicating rapid wear towards the end of engine life. Cluster 1 (Intermediate Degradation) peaks around 70 cycles, showing a transition phase where engine still have a moderate RUL. Cluster 2 (Healthy Engines) has a peak around 150 – 180 cycles, with a longer tail extending beyond 300 cycles, suggesting a steady and predictable degradation process.

2. FD003 (HPC +Fan):

The inclusion of fan degradation creates a wider spread in the RUL distribution, leading to more variation in failure timelines. Cluster 2 (Severe Degradation) still peaks near 0 cycles, but there is a more gradual decline, indicating that failures are not sharply defined as in FD001. Cluster 1 (Intermediate Degradation) shows a broader distribution, with a more gradual transition from healthy to failed states, unlike FD001's sharper separation. Cluster 0 (Healthy Engines) has a more dispersed tail, highlighting that some engines remain functional for much longer, while others degrade more quickly due to the additional impact of fan deterioration.

3. Key Differences:

- Wider spread in FD003: The presence of fan degradation causes more variation in RUL, making the failure progression less predictable compared to FD001.
- More Gradual Cluster Transitions: FD003 shows a smoother progression between healthy and failed states, whereas FD001 exhibits a more structured decline in RUL values.
- Higher Uncertainty in FD003: Since fan degradation introduces an additional failure mode, engines in the same cluster can have more varied RUL values, leading to an overall broader and less concentrated distribution.

4.4 Overall key findings

The comparison highlights how additional degradation sources (fan) influence the failure progression, making prediction more complex for FD003. While FD001 shows a clearer degradation pattern, FD003 suggests multiple interacting failure mechanisms, emphasizing the importance of advanced predictive maintenance strategies to account for multiple degradation sources. Thus, training a model on mixed fault types requires high computational power and could lead to inaccurate predictions, as the degradation characteristics of one failure mode may not generalize well to another. By creating separate models for each dataset, we ensure that each model learns the specific failure dynamics of its respective fault condition improving prediction accuracy, interpretability and reliability.

Hence, a fault-specific approach simplifies analysis, allowing for clearer insights into sensor trends, degradation paths and maintenance strategies tailored to each failure mode.

Chapter 5 – Comparative Modelling

This chapter presents the modelling phase of the project, where various ML and DL techniques were applied to predict the RUL of aircraft engines under different fault modes, as provided in the C-MAPSS dataset. Considering the variation in degradation patterns and fault characteristics across the four datasets (FD001 to FD004), a fault-specific modelling approach was adopted. This means that a separate predictive model was developed and tuned for each dataset, tailored to its operational complexity and fault modes.

The primary objective of this phase is to evaluate the effectiveness of different predictive modelling approaches and determine which model perform best under each fault scenario. The study investigates both classical ML algorithms (such as RF and XGBoost) and sequential DL architectures (such as LSTM and GRU) comparing their predictive accuracy, generalization ability and computational efficiency.

This chapter begins by outlining the overall modelling strategy and data preparation steps, followed by a detailed discussion of the selected models and training procedures. The performance of each model is then assessed using standard evaluation metrics and results are compared across datasets to dray key insights. The findings from this comparative modelling phase provide valuable direction for selecting optimal models in future prognostic health management systems.

5.1 Modelling Strategy

The adopted fault-specific approach was based on the inherent differences in the operating conditions and fault modes represented by each dataset. FD001 and FD003 simulate engine degradation under a single operating condition (sea level), while FD002 and FD004 involve multiple operational settings, which introduce additional variability and complexity in sensor readings. Furthermore, FD001 and FD002 are characterized by a single fault mode (only HPC degradation) whereas FD003 and FD004 contain dual fault modes involving both HPC and Fan degradation. These distinctions significantly affect the progression and manifestation of faults, leading to unique degradation patterns in each dataset.

Given these differences, training a single, general-purpose model across all datasets would not be effective, as the model might struggle to generalize across varying fault behaviours and operational conditions. As we know that the degradation trends in FD003, which involve complex interactions between HPC and fan components, differ substantially from the simpler more linear fault patterns as in FD001. Therefore, using a single model would either overfit to simpler datasets or underperform in the presence of more complex, multi-modal degradation scenarios.

The fault-specific modelling approach allows each model to learn the distinct degradation characteristics and sensor-response patterns associated with a particular fault condition. This improves the accuracy of RUL prediction, enhances model interpretability and simplifies the model tuning process. Additionally, it supports a more modular and scalable framework for future prognostics systems, where each model can be optimized and deployed independently for different engine fault modes.

By isolating fault types and operational conditions in this way, the study ensures that each model is best suited to capture the dynamics of the data it is trained on, ultimately leading to more robust, reliable and context-aware RUL predictions.

5.2 Data Preprocessing and Feature Engineering

Before model training and evaluation, significant effort was directed towards preparing the data to ensure optimal input quality for both ML and DL models. The C-MAPSS datasets are multivariate time-series data containing operational settings and sensor readings across engine units and cycles. The preprocessing and feature engineering tasks were designed to handle noise, scale inconsistencies and to extract temporal patterns critical for RUL prediction.

5.2.1 Data Cleaning and Structuring

Raw training and test datasets were loaded for each fault mode. These included unit number, cycle number, three operational settings and 21 sensor measurements. Columns containing only NaN values were dropped to clean the datasets. Since FD001 and FD003 involves only single operating condition there are constant features which doesn't offer any variability to the modelling process. The constant features include the three operational settings and the sensor measurement columns 1, 5, 10, 16 and 19 were dropped for both the fault modes.

- For the training data, RUL was computed by subtracting the current cycle from the last cycle observed for each engine unit.
- For the ML models, the last cycle of each engine unit was extracted from the test data and only those instances was used for RUL prediction and compared with the given actual RULs.

- For the DL models, the actual RUL was computed for each engine cycle for the test units, this is done by adding the given RUL to the last observed cycle of each test unit and then subtracting it with the current cycle.

5.2.2 Feature Engineering

To enhance the model's ability to learn from the temporal and dynamic nature of degradation, the following feature transformations were applied to all the relevant sensor readings:

- **Exponential Weighted Moving Average (EWMA):** Captures short-term memory and lag trends using a span of 10 cycles, providing smoother transitions in sensor behaviour and highlighting recent changes in condition.
- **First-Order Differences:** Highlights changes between consecutive sensor readings, helping the model to detect sudden shifts that may signal the onset of degradation.
- **Rolling Mean and Standard Deviation (window size=10):** Extracts local statistical summaries of the sensor values. The rolling mean captures gradual changes, while rolling standard deviation reflects the variability and noise in the signal. These are particularly useful in identifying phases of steady-state operation versus deterioration.

These features were computed for each sensor variable across the training and test sets. Missing values introduced due to rolling operations were handled using forward and backward fill operations to ensure continuity in the data.

5.2.3 Feature Scaling

Since distance-based algorithms like RF, XGB and neural networks are sensitive to feature scaling, MinMaxScaler was applied to both the operational settings and the engineered sensor measurements. This scaled all values into the range [0,1], ensuring that no variable dominated due to its original scale.

5.2.4 Handling Time-Series structure for Deep Learning

Deep learning models LSTM, GRU and Conv1D require input in 3D format: (samples, steps, features). To convert the data:

- A sliding window approach with a time step of 150 cycles was used to form input sequences.
- For each window, the model uses the past 150 timesteps to predict the RUL at the next cycle.

- This approach captures the historical degradation trends, making it well-suited for sequence modelling networks like LSTM and GRU.

The dataset was finally split into training and validation sets using a 90-10 ratio while preserving the time-series order (i.e., no shuffling) to maintain temporal dependencies.

This preprocessing pipeline ensured that the ML models received a rich set of engineered features that capture both point-wise and temporal dynamics, while the DL models were equipped with sequential data suitable for learning long-term dependencies in degradation. The same methodology was applied consistently across all fault conditions (FD001 to FD004), with minor modifications based on the number of operational settings or the complexity of fault modes. This consistency supports fair performance comparison during the comparative modelling phase which described in the next sections.

5.3 Model Selection and Performance

The selection of models for this study was motivated by the need to explore both classical ML techniques and advanced DL architectures suitable for time-series based RUL prediction. Given the sequential nature of the C-MAPSS dataset, which consists of multivariate sensor readings over time for each engine unit, models capable of capturing temporal patterns, degradation trends and feature interactions were prioritized.

To ensure a diverse yet effective modelling approach, a combination of tree-based ensemble ML models and DL models with different architectural characteristics was selected. These models are widely recognized in PHM domain and provide a balance between interpretability, computational efficiency and predictive power.

The following ML models were chosen for their proven capabilities in handling structured tabular data and non-linear relationships:

- RF is Selected for its simplicity, robustness and effectiveness in modelling tabular time-series data without requiring extensive hyperparameter tuning. RF is capable of handling high-dimensional data and automatically captures feature interactions.
- XGB is a powerful gradient boosting framework known for its scalability and performance. It was chosen to complement RF by offering a more refined boosting-based learning approach that improves on error residuals iteratively.
- LGBM is chosen for its high training efficiency and low memory footprint. LGBM offers fast training and is well-suited for large datasets, making it a competitive alternative to XGB in many predictive maintenance tasks.

To capture long-term dependencies and complex time-series dynamics, the following DL models were selected:

- LSTM is a widely used RNN variant capable of learning from sequential data with long-range temporal dependencies. It was chosen due to its strong performance in degradation modelling and predictive maintenance applications.
- GRU, a computationally efficient alternative to LSTM, was included to evaluate whether a simpler recurrent architecture could achieve comparable predictive performance while reducing training time.
- Conv1D was selected to explore the potential of CNN architectures in extracting local temporal features. Its ability to process sequential input with fewer parameters and faster convergence makes it a valuable addition to the modelling suite.

These models allow for a comprehensive evaluation of both learning paradigms, traditional tree-based ML methods and DL sequence models. This comparison facilitates the identification of the most suitable modelling approach under varying fault complexities and operational conditions which is evaluated in the subsequent comparative analysis.

5.3.1 Machine Learning Models

In this section, three ML algorithms were implemented and evaluated for RUL prediction: RF, XGB and LGBM. These ensemble-based models were selected due to their proven effectiveness in handling structured, high-dimensional sensor data with non linear relationships, characteristics typical of the C-MAPSS dataset. Each model was trained and validated using optimized hyperparameters and their performance was assessed based on accuracy metrics (RMSE and R^2) and computational efficiency.

Random Forest (RF): Random Forest is an ensemble method that constructs a large number of decision trees during training and outputs the mean prediction of the individual trees. It reduces overfitting by averaging multiple models and leverages randomness to improve generalization. Below are the hyperparameters that were used:

- n_estimators – number of decision trees in the forest.
- max_depth – maximum depth of each tree.
- min_samples_split and min_samples_leaf – node splitting control.
- random_state – for reproducibility.
- n_jobs – utilized all available cores for faster computation.

The key strengths of this model is its high effectiveness when working with tabular and structured datasets, making it ideal for many real-world applications. It can automatically capture complex non-linear relationships and feature interactions without the need for extensive preprocessing or feature engineering. Additionally, the model is robust to outliers and noise, ensuring reliable performance even with imperfect data. Its structure allows for ease of interpretation and parallelization, making it efficient and scalable for large datasets.

The figure below illustrates the Actual vs Predicted RUL for all four fault modes (FD001 – FD004) using RF. The red dashed line represents the ideal prediction line where predicted RUL equals the actual RUL.

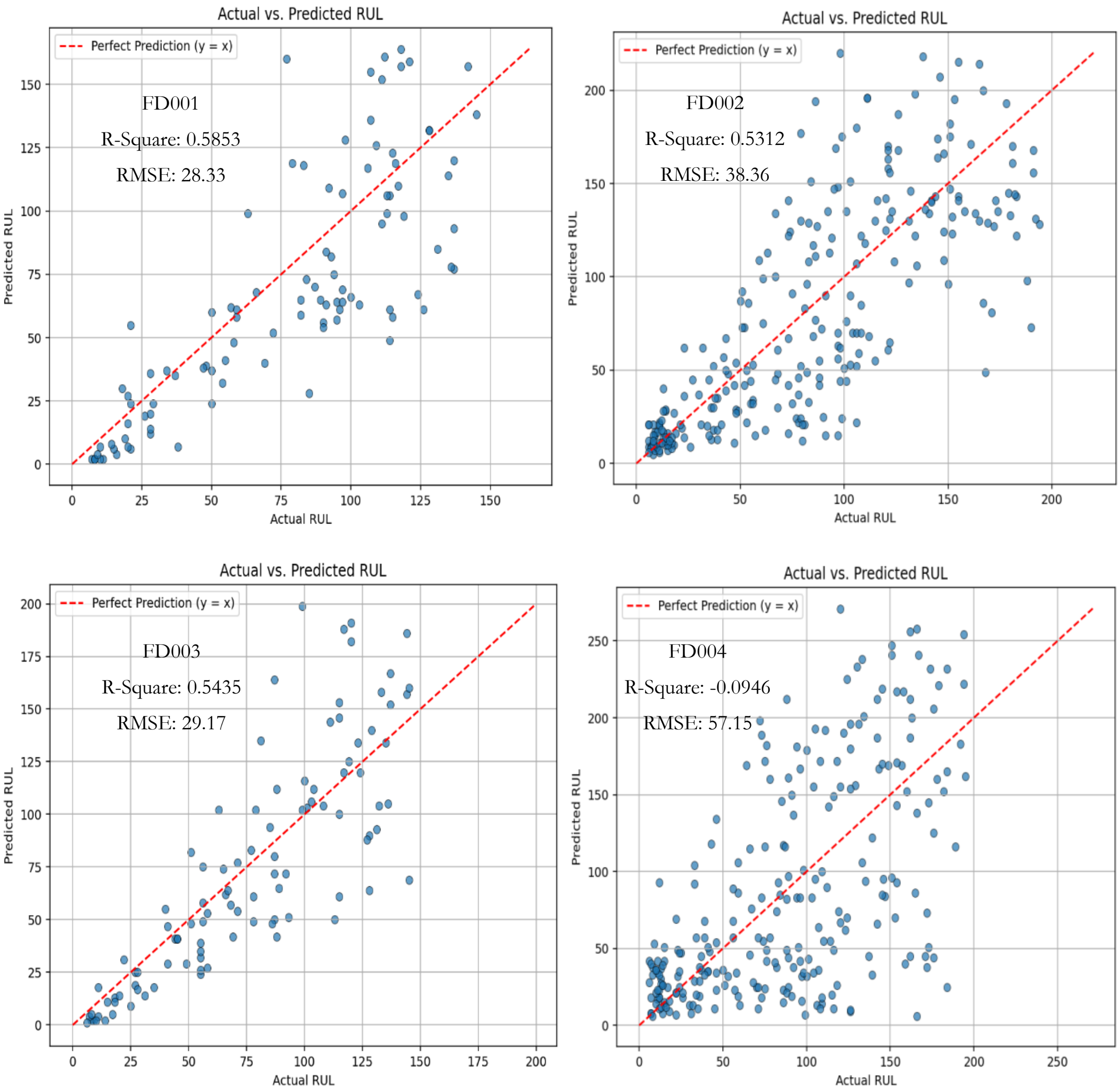


Figure 5-1: RF Actual vs. Predicted RULs (FD001 – FD004)

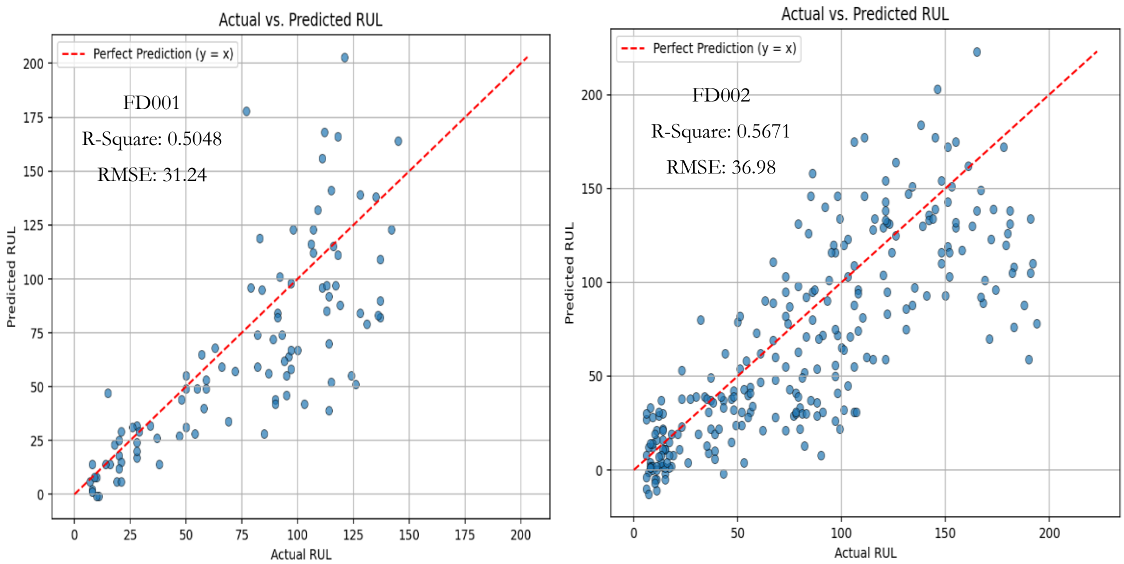
The model performs reasonably on FD001 and FD003, achieving R^2 scores of 0.5853 and 0.5435, respectively with moderate RMSE values. However, performance drops for FD002 due to multiple operating conditions and significantly for FD004, which involves multiple fault types reflected by a negative R^2 (-0.0946) and high RMSE (57.15). Overall RF struggles with generalization under complex, multi-condition datasets.

Extreme Gradient Boosting (XGB):

XGBoost is a gradient boosting framework that builds trees sequentially by correcting the errors of previous trees. It employs regularized learning objective to prevent overfitting and is known for its speed and accuracy. The hyperparameters that were used are given below:

- n_estimators – number of boosting rounds.
- learning_rate – controls step size at each iteration.
- max_depth=6 – limits tree complexity.
- subsample, colsample_bytree – introduces randomness for generalization.
- objective='reg:squarederror' – loss function for regression.

The key strengths lies in its built-in regularization techniques, including both L1 (Lasso) and L2 (Ridge) regularization, which help prevent overfitting and improve the model's generalization ability. It also handles missing values natively, making it robust and reducing the need for extensive data preprocessing. Additionally, it supports parallel computation, significantly accelerating the training process and making it an excellent choice for time sensitive applications. The performance of XGB model for all the four datasets (FD001 – FD004) is given in figure 5-2.



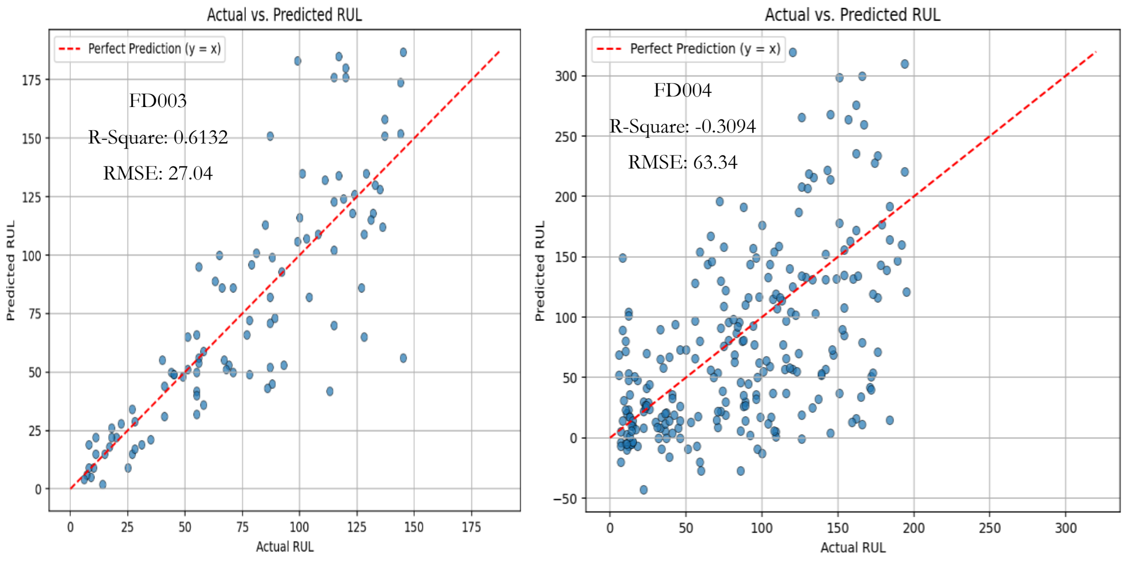


Figure 5-2: XGB Actual vs. Predicted RULs (FD001 – FD004)

XGB outperformed RF across all datasets, particularly excelling in FD003 (R^2 = 0.61, RMSE = 27.04) and FD002 (R^2 0.57, RMSE = 36.98), showing a better ability to learn from complex patterns. FD001 performance was also solid (R^2 = 0.50), but FD004 continued to pose difficulties, yielding a negative R^2 and a highest RMSE (63.34), signifying weak generalisation in the most challenging scenario.

Light Gradient Boosting Machine (LGBM):

LGBM is another gradient boosting framework optimized for speed and memory efficiency. It uses a histogram-based approach and leaf-wise tree growth to achieve faster training and better accuracy with large-scale data. Below are the hyperparameters that were used:

- learning_rate – controls the pace of learning.
- n_estimators – number of boosting rounds.
- num_leaves – maximum number of leaves in one tree (controls model complexity).

It excels in both computation and memory usage. One of its standout features is its ability to support categorical features directly, eliminating the need for one-hot encoding and simplifying the preprocessing pipeline. LGBM maintains high predictive accuracy while significantly reducing training time, making it ideal for large-scale machine learning tasks. Additionally, it is capable of handling datasets with high dimensionality, making it a robust and scalable solution for complex data-driven problems. The performance of LGBM for all the four fault modes (FD001 – FD004) is given below.

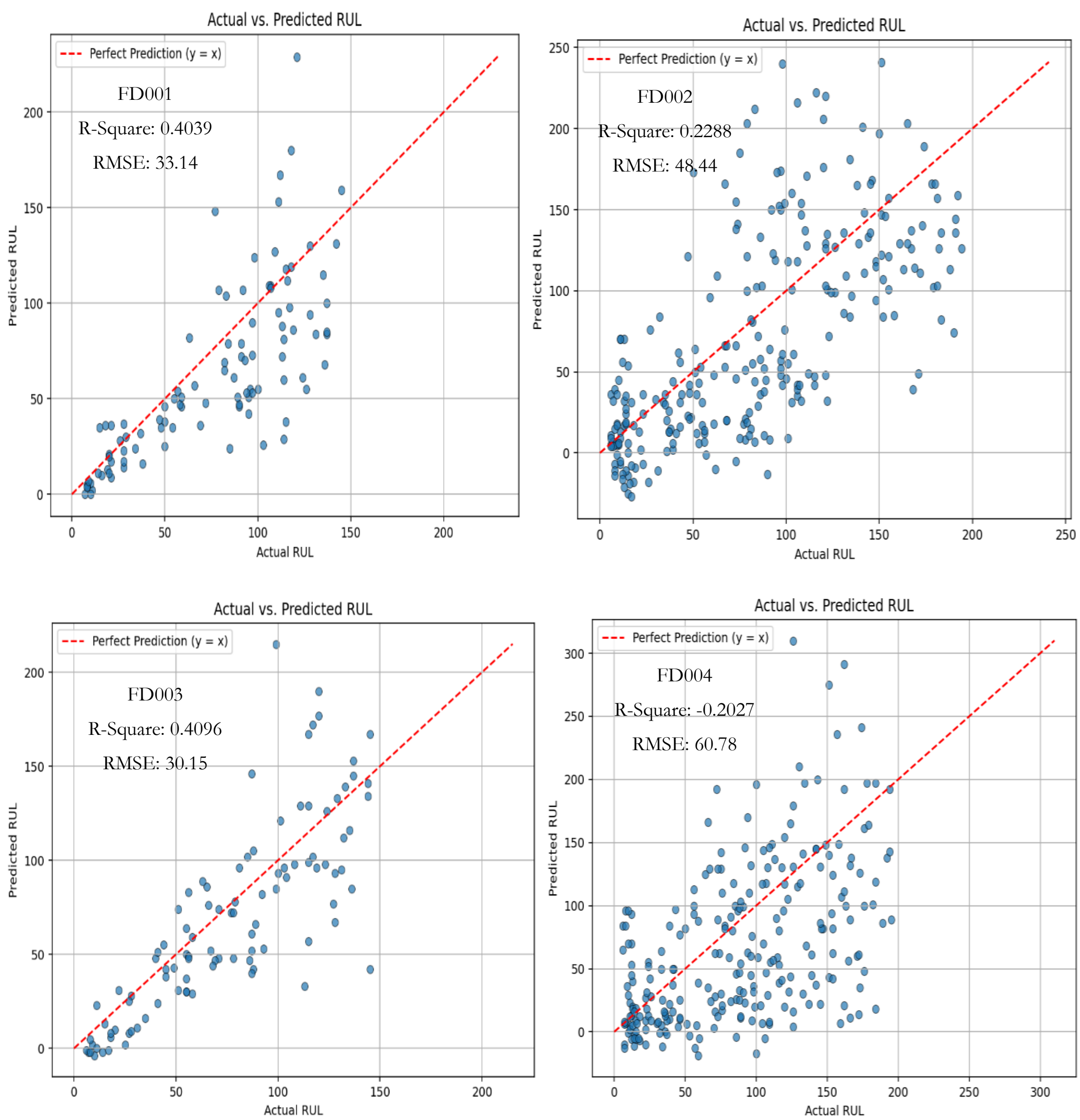


Figure 5-3: LGBM Actual vs. Predicted RULs (FD001 – FD004)

LGBM's performance was generally inferior to XGB and RF, especially in the multi-condition datasets. While FD001 and FD003 produced acceptable results ($R^2 \approx 0.40$), both FD002 ($R^2 = 0.22$) and FD004 ($R^2 = -0.20$) reflected poor predictive capabilities. The higher RMSEs across all datasets point to less robust learning, possibly due to sensitivity to hyperparameters or overfitting in certain cases.

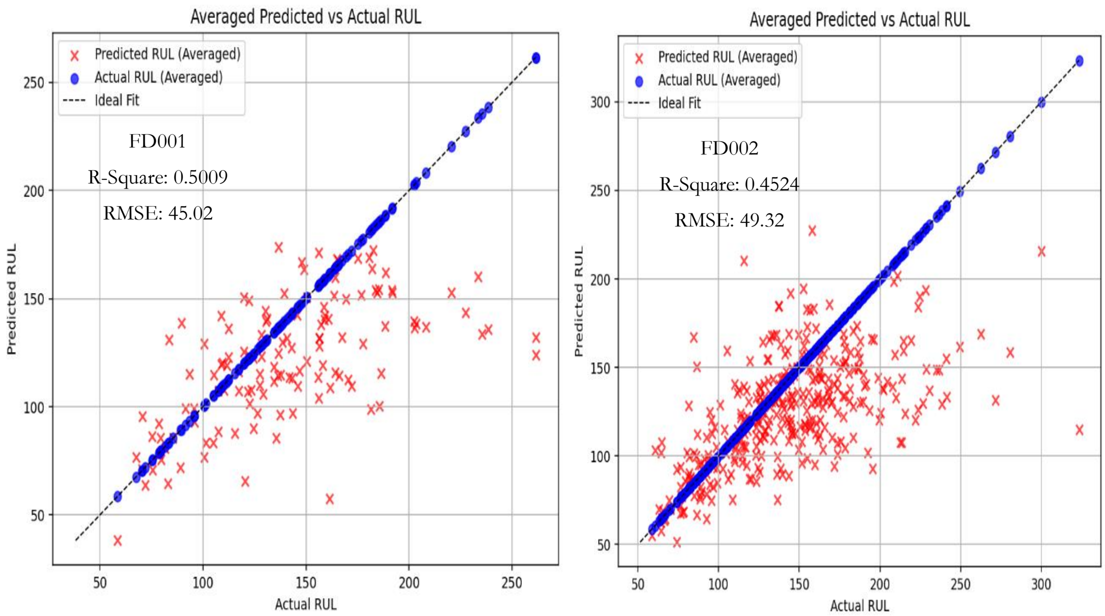
5.3.2 Deep Learning Models

In this section, deep learning architectures were leveraged to model the sequential degradation behaviour observed in aircraft engine sensor data. Three models LSTM, GRU and Conv1D were chosen for their effectiveness in capturing temporal dependencies and feature patterns in time-series data.

Long Short-Term Memory (LSTM):

LSTM networks, a specialized type of recurrent neural network (RNN), are well-suited for learning from sequential data where long-range dependencies are critical. In the context of RUL prediction, they help in modelling how sensor signals evolve over time, making them ideal for tracking the degradation trajectory of engine components.

The implemented LSTM architecture used a stacked structure, allowing the model to learn both short-term and long-term patterns. Regularization techniques such as dropout and L2 penalties were employed to mitigate overfitting. The models were trained with adaptive learning strategies, incorporating early stopping and learning rate scheduling for stability and efficiency. Evaluation was performed using RMSE and R^2 metrics, alongside visual inspection through predicted vs. actual RUL plots. Figure 5-4 shows the performance of LSTM for all the four fault modes.



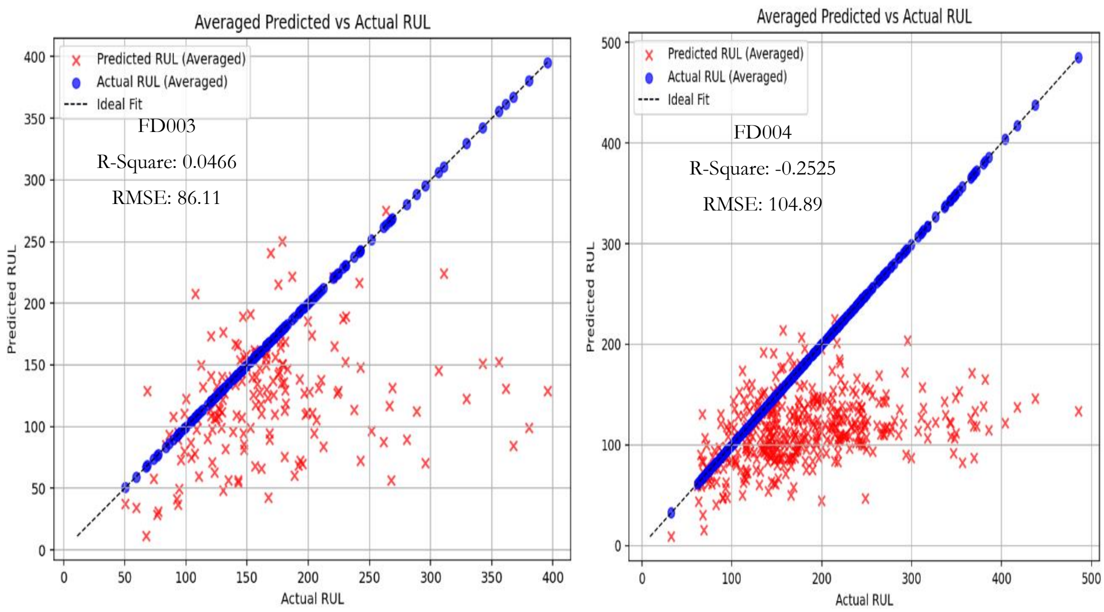


Figure 5-4: LSTM Actual vs. Predicted RULs (FD001 – FD004)

- The LSTM model demonstrated good predictive capability for datasets with single fault conditions (FD001 and FD002).
- Performance dropped significantly for FD003 and FD004, where operating fault modes are more complex which deals with both HPC and Fan degradation.
- The negative R^2 in FD004 indicates poor generalization to complex patterns, possibly due to insufficient sequence learning or overfitting.

Gated Recurrent Unit (GRU):

GRUs are a lighter alternative to LSTM networks, offering similar capabilities in learning sequential patterns but with fewer parameters and a simpler architecture. This makes GRUs computationally more efficient while maintaining strong performance in predictive tasks. The GRU model followed a similar training and evaluation procedure as the LSTM model. It demonstrated reliable generalization, especially in fault modes with moderate degradation progression and was able to capture essential temporal features from the input sequence. The visual plots showing the performance of the GRU model is given below.

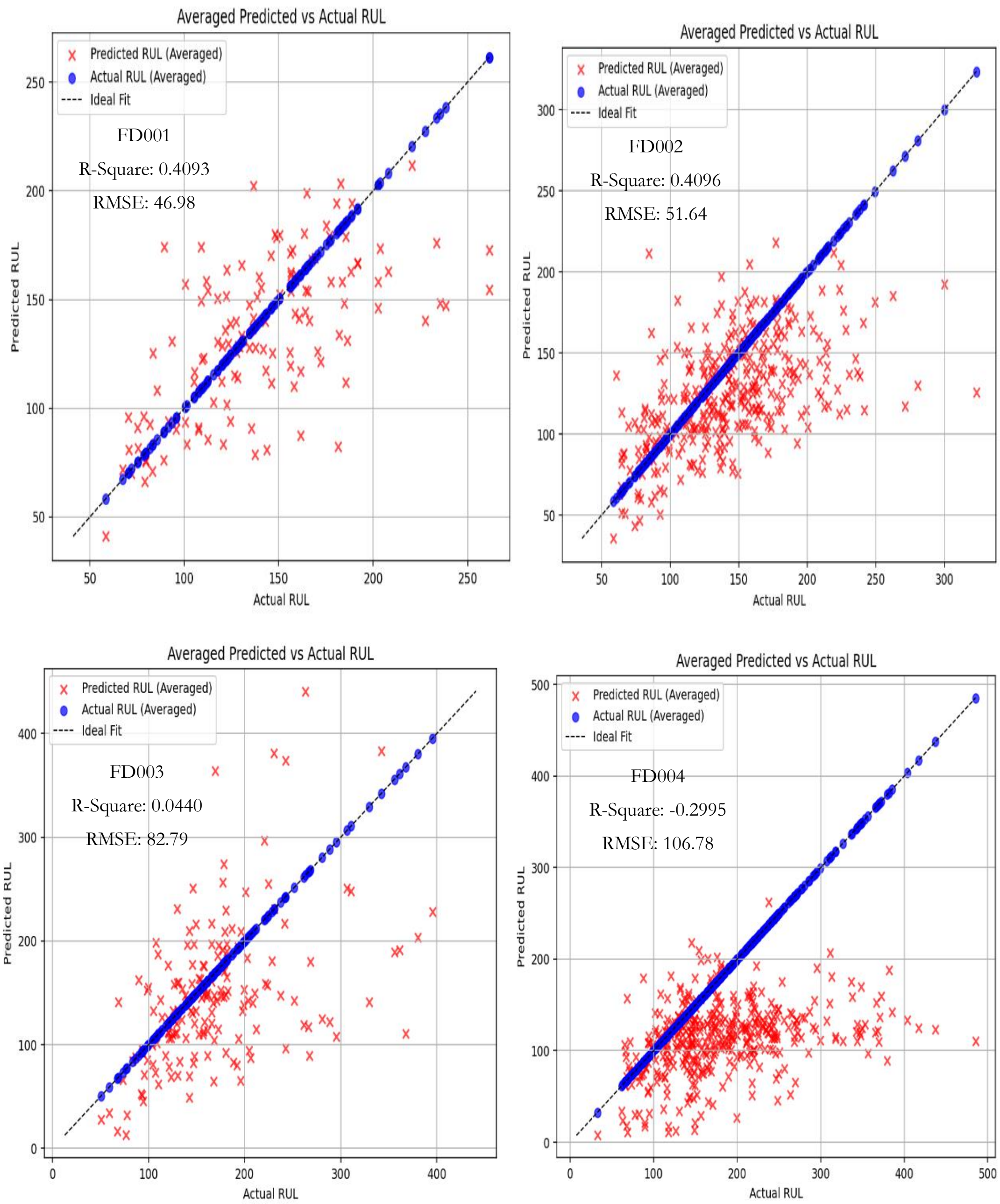


Figure 5-5: GRU Actual vs. Predicted RULs (FD001 – FD004)

- GRU achieved reasonable results in FD001 and FD002, showing its strength in modelling sequences with fewer parameters than LSTM.
- For FD003 and FD004, GRU struggled to maintain performance. The very low R^2 values suggest a limited ability to capture complex degradation trajectories.
- Nevertheless, GRU's efficiency and simplicity make it a good candidate for real-time deployment where computational resources are constrained.

1D Convolutional Neural Network (Conv1D)

Conv1D models are effective for extracting localized temporal patterns in time-series data. Unlike RNN based models, Conv1D operates using convolutional filters that detect short-terms trends across the input window. This makes it particularly useful for identifying abrupt sensor variations or fault onsets.

In this study, Conv1D was used as a standalone temporal model where feature extraction was performed via convolutional layers before regression through dense layers. This architecture is not only computationally efficient but also exhibits strong performance in scenarios where degradation signals are prominent and structured. The model was evaluated using the same performance metrics and showed consistent results in tracking RUL (shown in figure 5-6).

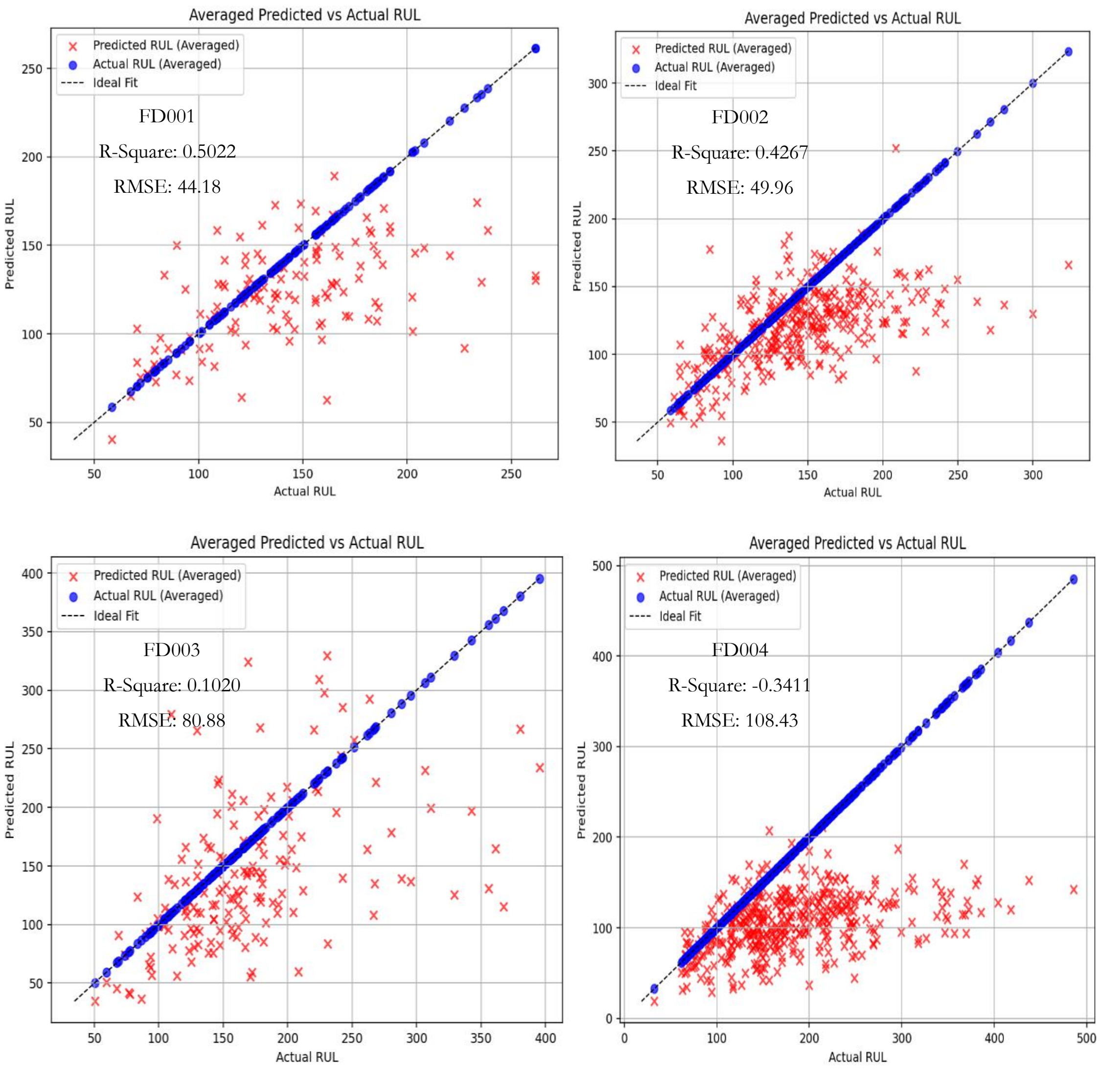


Figure 5-6: Conv1D Actual vs. Predicted RULs (FD001 – FD004)

- The Conv1D model captured local temporal dependencies well, performing similarly to LSTM in simpler fault conditions.
- However, like the recurrent models, it failed to generalize effectively in datasets with multiple operating conditions and fault modes.
- The high RMSE in FD004 suggests that convolutional filters alone may not be sufficient to handle complex sequences in degradation modelling.

Chapter 6 – Key Insights and Future Work

6.1 Performance Comparison Analysis of ML models

The comparison analysis of RMSE and R^2 across different ML models and fault modes shown in figure 6-1 and figure 6-2 reveals several key insights. Overall, XGB and RF consistently outperform LGBM in terms of both prediction accuracy and model fit across most fault conditions. Among the fault modes, FD003 exhibits the good predictive performance for all models, while FD004 presents the greatest challenge, with all models showing reduced accuracy and poor R^2 values. This suggests that simpler fault modes are easier to model, whereas more complex scenarios require more sophisticated approaches. Additionally, XGB demonstrates slightly better generalization across multiple fault types, indicating its robustness in varying conditions.

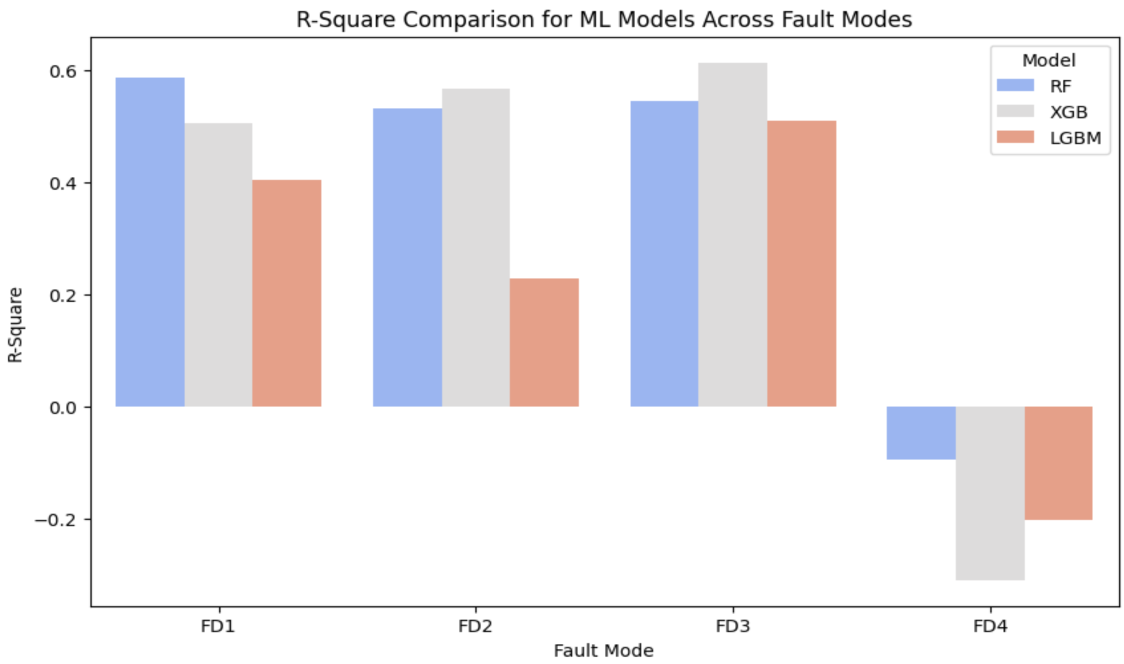


Figure 6-1: R^2 Comparison for ML Models Across Fault Modes

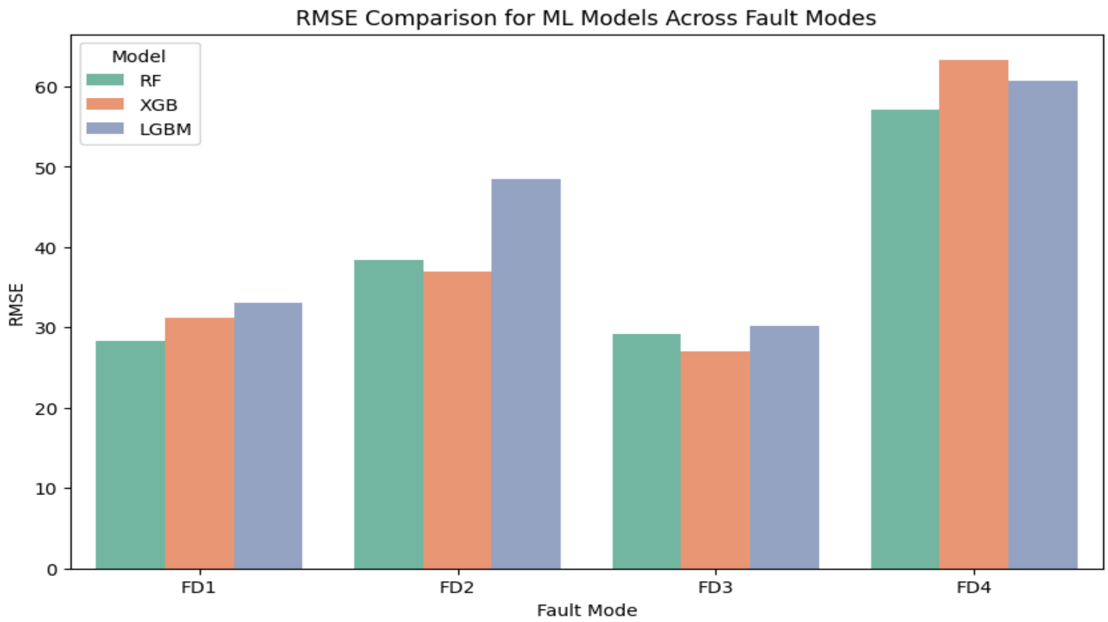


Figure 6-2: RMSE Comparison for ML Models Across Fault Modes

6.2 Performance Comparison Analysis of DL models

The RMSE and R^2 comparison plots (Figure 6-3 and Figure 6-4) illustrate the performance of the three deep learning models across different fault modes. Overall, model performance varies depending on the fault condition. For RMSE, all models exhibit relatively low error in FD001 and FD002, with increased prediction errors in FD003 and FD004. Among the models, Conv1D tends to show slightly better consistency across fault modes in terms of lower RMSE, particularly under complex degradation patterns.

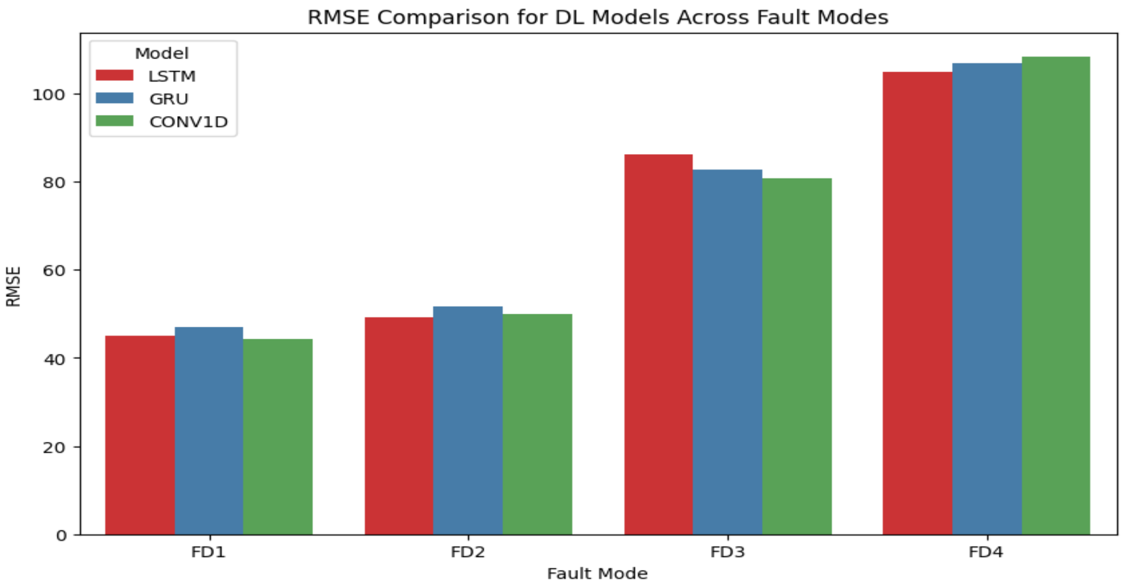


Figure 6-3: RMSE Comparison for DL Models Across Fault Modes

The R^2 plot reveals that all models perform reasonably well for FD001 and FD002, indicating good predictive power. However, a noticeable decline in R^2 is observed for FD003 and all models perform poorly on FD004 showing negative R^2 is observed for FD003 and all models perform poorly on FD004, showing negative R^2 values. This suggests challenges in accurately predicting the RUL under the FD4 fault mode, likely due to increased complexity or noise in the data. These insights highlight that while deep learning models can effectively handle simpler fault modes, their performance may degrade in more complex scenarios, emphasizing the need for further tuning or hybrid approaches for robust RUL prediction.

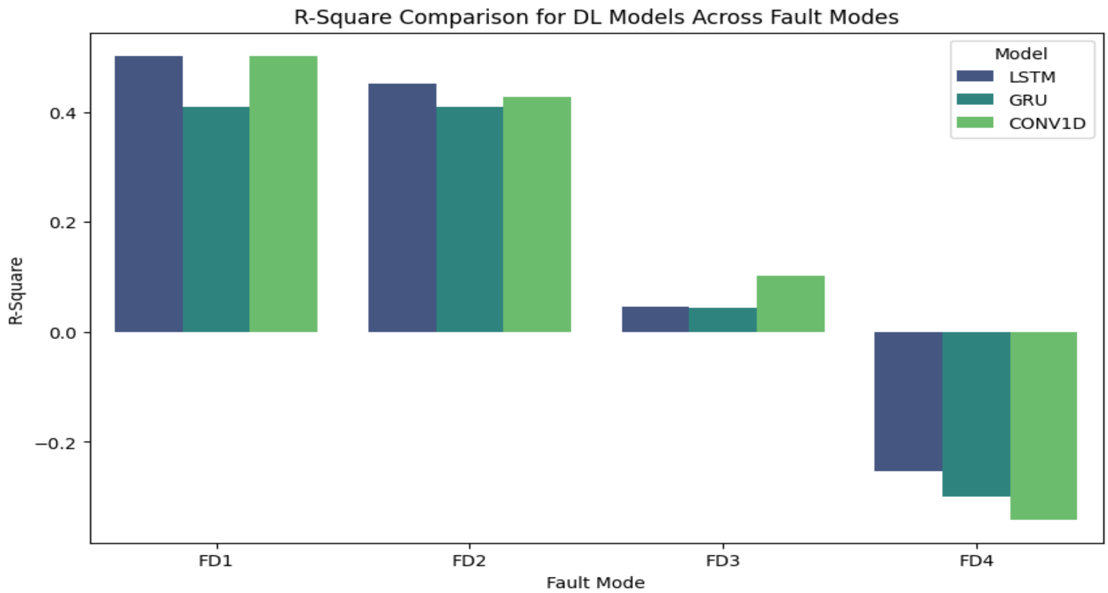


Figure 6-4: R^2 Comparison for DL Models Across Fault Modes

6.3 Overall Key Insights

The 3D plots shown in figure 6-5 provide a visual representation of the trade-offs between model accuracy (R^2), error (RMSE) and computation time for both ML and DL models across fault modes.

In the ML models, LGBM consistently balances good performance with relatively low computation time, making it suitable for time-sensitive applications. RF and XGB also demonstrate reasonable performance but show higher computation time, especially under more complex fault scenarios.

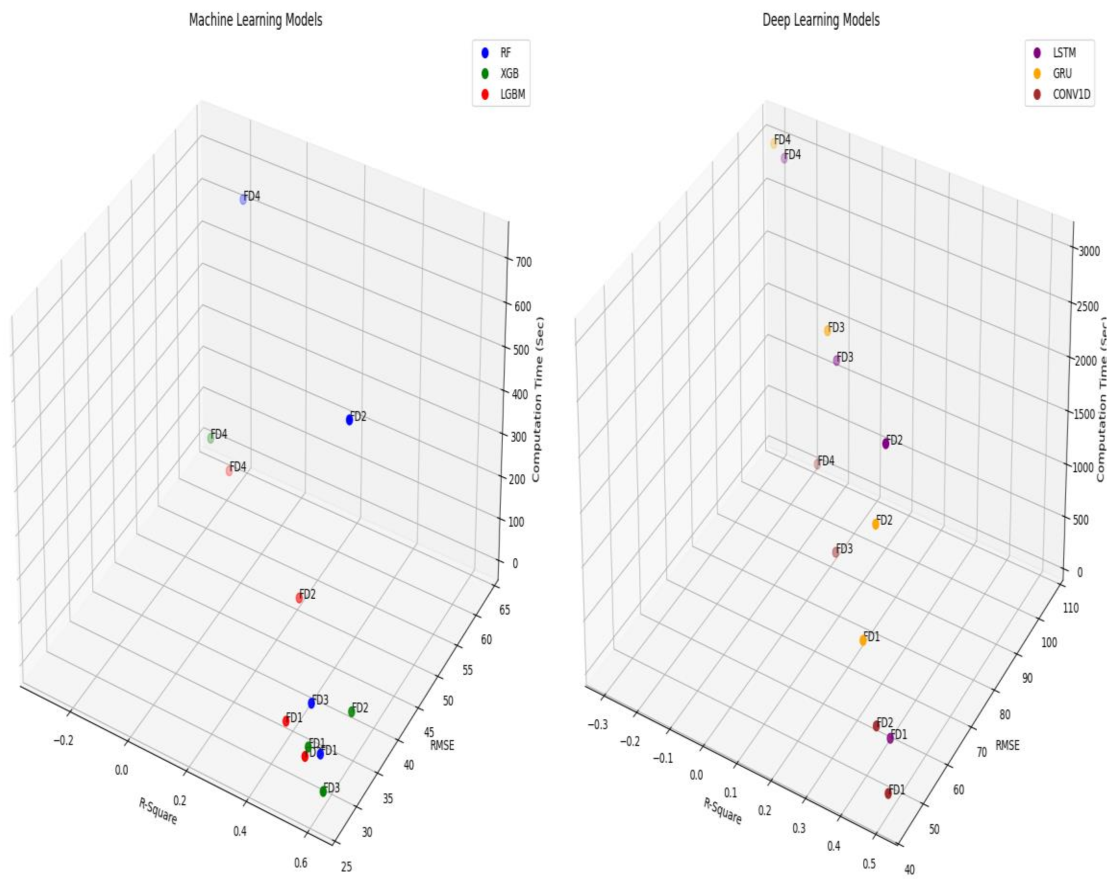


Figure 6-4: Trade-offs between R², RMSE and computation time

DL models, while generally achieving higher RMSE and lower R^2 in certain fault modes, require significantly more computation time. Among them, Conv1D appears to be the most efficient in terms of computation time, while LSTM and GRU show longer training durations, particularly under more complex fault modes. The overall summary of insights from the model performance comparison analysis is listed below:

- None of the models performed well under FD004 (Multiple Fault and Multiple Operating Conditions), indicating the increased complexity and difficulty in modelling real-world degradation patterns under compounded scenarios.
- Among the ML models, LGBM exhibited slightly lower performance in terms of RMSE and R^2 compared to RF and XGB, but offered faster computation time, making it a viable choice for real-time or resource-constrained environments.
- RF and XGB achieved better accuracy across simpler fault modes, but at the cost of increased computation time, especially as fault complexity increased.

- The DL models struggled to handle multiple fault conditions effectively, highlighting potential challenges in learning complex fault interactions from time-series data without additional feature engineering or architecture enhancements.
- DL models are computationally expensive (require longer training times), whereas Conv1D stood out with a good balance between performance and training time, indicating its potential for applications requiring efficient temporal feature extraction without heavy computational overhead.
- Overall, simpler fault modes (FD001, FD002) where handled reasonably well by most models, but model robustness significantly declined as fault complexity increased.

6.4 Future Work

To further enhance the accuracy and generalization capability of the RUL prediction models, several advanced data preprocessing and feature engineering techniques will be explored:

6.4.1 Advanced Feature Engineering

Incorporating sophisticated feature extraction techniques is a key avenue for improving the predictive power of the models. Methods such as PFE, FFT and WT will be considered:

- PFE enables the capture of non-linear relationships by generating interaction terms and higher-order components from the original features. This can help uncover complex patterns that basic linear models might miss.
- FFT will be used to analyse the frequency domain characteristics of sensor signals, which can be particularly useful for detecting hidden cyclic trends or periodic degradation behaviour in engine systems.
- WT allows for time-frequency analysis, making it possible to extract localized signal features at different scales. This can be valuable for identifying transient events and capturing non-stationary patterns in time-series sensor data.

Although these techniques offer significant benefits in terms of richer feature representations, they also involve higher computational costs and may require careful tuning to balance complexity and performance.

6.4.2 Dimensionality Reduction

To manage the high dimensionality resulting from advanced feature engineering and to minimize the risk of overfitting, PCA will be employed:

- PCA reduces computational complexity by transforming the feature space into a set of linearly uncorrelated components while preserving most of the variance in the data. This not only improves training efficiency but also aids in generalizing the model better on unseen data.
- By selecting only the most significant principal components, noise and redundancy in the feature set can be effectively minimized, leading to a more robust learning process.

6.4.3 Feature Selection Methods

Identifying and selecting the most relevant features plays a critical role in improving model performance and interpretability. The following feature selection strategies will be explored:

- GA: These are stochastic optimization techniques inspired by natural selection. GA can be employed to iteratively search for an optimal subset of features by evaluating combinations based on model performance.
- RFE: This is a wrapper-based method that recursively removes the least important features and builds the model using the remaining ones. It helps in selecting features that contribute the most to the prediction task.
- LASSO: Leveraging L1 regularization, LASSO regression shrinks less important feature coefficients to zero, effectively acting as a feature selector while also addressing multi collinearity in the dataset.
- FIRF: This technique involves ranking features based on importance scores obtained from models like RF or GB and selecting the top contributors. This not only enhances efficiency but also helps in model interpretability.

By integrating these feature selection techniques, the goal is to refine the input space to include the most informative and impactful features, which is expected to significantly enhance both computational efficiency and model accuracy.

References

[1] Adryan Fitra Azyus, Sastra Kusuma Wijaya and Mohd Naved (2023) 'Prediction of remaining useful life using the CNN-GRU network: A study on maintenance management', *Software Impacts*, 17, pp. 100535–100535. Available at: https://doi.org/10.1016/j.simpa.2023.100535

[2] Alomari, Y.M., Mátyás Andó and Baptista, M. (2023) 'Advancing aircraft engine RUL predictions: an interpretable integrated approach of feature engineering and aggregated feature importance', *Scientific Reports*, 13(1). Available at: https://doi.org/10.1038/s41598-023-40315-1

[3] Amat, J. (2019) *Deep learning Recurrent Neural Networks - Skforecast Docs*, *Skforecast.org*. Available at: https://skforecast.org/0.13.0/user_guides/forecasting-with-deep-learning-rnn-lstm

[4] Cho, A.D., Carrasco, R.A. and Ruz, G.A. (2022) 'A RUL Estimation System from Clustered Run-to-Failure Degradation Signals', *Sensors (Basel)*, 22(14), pp. 5323–5323. Available at: https://doi.org/10.3390/s22145323

[5] Gordon, J. (2023) *Practical Guide for Feature Engineering of Time Series Data*, *dotData*. Available at: https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/

[6] Huthaifa Al-Khazraji *et al.* (2022) 'Aircraft Engines Remaining Useful Life Prediction Based on A Hybrid Model of Autoencoder and Deep Belief Network', *IEEE Access*, 10, pp. 82156–82163. Available at: https://doi.org/10.1109/access.2022.3188681

[7] Karakose, M. (2016) 'Sensor Based Intelligent Systems for Detection and Diagnosis', *Journal of Sensors*, 2016, pp. 1–1. Available at: https://doi.org/10.1155/2016/5269174

[8] Koehrsen, W. (2020) *Random Forest Simple Explanation*, *Medium*. Available at: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

[9] Lazzeri, F. (2021) *Introduction to feature engineering for time series forecasting*, *Data Science at Microsoft*. Available at: https://medium.com/data-science-at-microsoft/introduction-to-

feature-engineering-for-time-series-forecasting-620aa55fcab0

[10] Lindström, K. (2021) *Fault Clustering With Unsupervised Learning Using a Modified Gaussian Mixture Model and Expectation Maximization*, *DIVA*. Available at: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1566435&dswid=-9288 (Accessed: 6 March 2025)

[11] Mohamed, S. *et al.* (2020) 'Towards Distribution Clustering-Based Deep LSTM Models for RUL Prediction', *Prognostics and Health Management Conference - PHM2020*. [Preprint]. Available at: https://doi.org/10.1109/PHM-Besancon49106.2020.00049.

[12] Mousaab Zakhrouf *et al.* (2023) 'Novel insights for streamflow forecasting based on deep learning models combined the evolutionary optimization algorithm', *Physical Geography*, 44(1), pp. 31–54. Available at: https://doi.org/10.1080/02723646.2021.1943126

[13] Rahul Holla (2024) *Advanced Feature Engineering for Time Series Data - Rahul Holla - Medium*, *Medium*. Available at: https://medium.com/@rahulholla1/advanced-feature-engineering-for-time-series-data-5f00e3a8ad29

[14] Safavi, V. *et al.* (2024) 'Battery Remaining Useful Life Prediction Using Machine Learning Models: A Comparative Study', *Information*, 15(3), p. 124. Available at: https://doi.org/10.3390/info15030124

[15] Saxena, A. *et al.* (2008) 'Damage propagation modeling for aircraft engine run-to-failure simulation', *2008 International Conference on Prognostics and Health Management* [Preprint]. Available at: https://doi.org/10.1109/phm.2008.4711414

[16] Shenfield, A. and Howarth, M. (2020) 'A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults', *Sensors*, 20(18), p. 5112. Available at: https://doi.org/10.3390/s20185112

[17] Trapani, N. and Longo, L. (2023) 'Fault Detection and Diagnosis Methods for Sensors Systems: a Scientific Literature Review', *IFAC-PapersOnLine*, 56(2), pp. 1253–1263. Available at: https://doi.org/10.1016/j.ifacol.2023.10.1749

[18] Vasso Reppa, Polycarpou, M.M. and Panayiotou, C.G. (2016) 'Sensor Fault Diagnosis',

Foundations and Trends® in Systems and Control, 3(1-2), pp. 1–248. Available at: https://doi.org/10.1561/2600000007

[19] Xiong Xinxin *et al.* (2015) 'Remaining Useful Life Prognostics of Aircraft Engines Based on Damage Propagation Modeling and Data Analysis'. Available at: https://doi.org/10.1109/iscid.2015.206

[20] Yin, L., Ma, P. and Deng, Z. (2021) 'JLGBMLoc—A Novel High-Precision Indoor Localization Method Based on LightGBM', *Sensors*, 21(8), p. 2722. Available at: https://doi.org/10.3390/s21082722

[21] Yoo, Y. (2020) 'Data-driven fault detection process using correlation based clustering', *Computers in Industry*, 122, p. 103279. Available at: https://doi.org/10.1016/j.compind.2020.103279

[22] Zainab Hasan Ali and Burhan, A.M. (2023) 'Hybrid machine learning approach for construction cost estimation: an evaluation of extreme gradient boosting model', *Asian Journal of Civil Engineering*, 24(7), pp. 2427–2442. Available at: https://doi.org/10.1007/s42107-023-00651-z