C O V E N T R Y

U N I V E R S I T Y

Faculty of Engineering, Environment and Computing

School of Computing, Electronics and Mathematics

MSc Data Science

7150CEM

Data Science Project

Optimizing Predictive Health Management: A GRU-LSTM Hybrid Model for Remaining Useful Life Prediction in Aviation

Author: Haadhi Mohammed

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science

Academic Year: 2024/25

Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of copyright

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below:

Signed: Haadhi Mohammed Date: 13 December 2024

First Name:	Haadhi
Last Name:	Mohammed
Student ID Number	
Ethics Application Number	
1st Supervisor Name	
2nd Supervisor Name	

Abstract

The dependability and safety of turbofan engines are essential in aviation, requiring effective predictive maintenance procedures. The prediction of Remaining Useful Life (RUL) is an essential element of Prognostics and Health Management (PHM), facilitating the forecast of future problems and mitigating operational risks. Using its ability to capture both short term and long term dependencies in sequential data, this study investigates the use of a hybrid Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) model for precise RUL prediction.

The research use the CMAPSS dataset, a standard dataset for engine health assessment, which includes sensor data and operating parameters from turbofan engines. To improve data quality, a thorough preparation pipeline was put in place, which included dimensionality reduction, feature selection, and scaling with RobustScaler. The hybrid GRU-LSTM architecture has four GRU layers and one LSTM layer, enhanced with a Scaled Dot-Product Attention mechanism to emphasise essential time steps in the sequence.

The model utilised the Adam optimiser for training and was assessed using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared. The validation findings indicate the efficacy of the proposed model, with a Validation RMSE of 24.62 and a validation MAE of 18.21. These findings demonstrate the potential of GRU-LSTM hybrid models to enhance safety in aviation applications, decrease downtime, and advance predictive maintenance systems.

Table of Contents

List of Figure

List of Tables

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr Anup Pandey, for his invaluable guidance, constructive feedback, and constant encouragement throughout the duration of my research. His expertise and insights have been instrumental in shaping this project and improving its overall quality. I sincerely appreciate the time and effort he dedicated to reviewing my progress and providing prompt and thorough feedback at every stage.

I am equally thankful to my second supervisor, Dr Mark Johnston, for his support and insightful suggestions throughout the course duration, which have greatly enhanced the depth and scope of this research.

I am also grateful to my family and friends for their unwavering support, understanding, and encouragement throughout this academic journey. Their belief in my abilities has been a constant source of motivation and strength during the challenging phases of this research.

Additionally, I extend my heartfelt thanks to all the faculty members who have taught me during this course. Their dedication and knowledge have significantly contributed to my academic growth and the successful completion of this project.

Finally, I would like to thank my colleagues and peers for their helpful discussions and encouragement during this journey. Their insights and camaraderie have made this experience both enriching and memorable.

1. Introduction

The aviation sector emphasises dependability and safety, rendering predictive maintenance a crucial component of operations. Turbofan engines, essential to contemporary aviation, need meticulous oversight to reduce the likelihood of mechanical failure. Currently, Prognostics and Health Management (PHM) (D. Wang et al., 2017) is extensively utilised in the aviation sector due to advancements in the industrial domain (*A Prognostics and Health Management Roadmap for Information and Electronics Rich Systems*, n.d.) (Lau & Fong, 2011). Remaining Useful Life (RUL) pertains to the forecasting of impending system failures derived from presently accessible monitoring data, constituting the essence of Prognostics and Health Management (J. Chen et al., 2019). This approach seeks to precisely forecast the remaining operational duration of a system prior to its failure, serving as a fundamental basis for developing a scientific and rational maintenance strategy (S. Sun et al., 2024).

In many instances, real-world data exhibit fault signatures indicative of a developing defect, although there is minimal data documenting fault progression until failure occurs. Acquiring genuine system fault progression data is generally time consuming and costly. This study use the Commercial Modular Aero Propulsion System Simulation (C-MAPSS) dataset, a simulation of turbofan jet engines given by NASA's Prognostics Centre of Excellence (Saxena et al., 2008).

This research project seeks to assess the efficacy of a hybrid deep learning architecture that integrates Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) layers with attention mechanisms to accurately predict the Remaining Useful Life of turbofan engines, utilising sequential data and preprocessing techniques to tackle challenges in temporal modelling and feature selection.

1.1 Background to the Project

The turbofan engine, a crucial component of the aircraft, is a complex and complicated system; its safety and dependability are paramount. An unforeseen engine failure prior to its overhaul might result in a catastrophic incident, potentially incurring millions in lost lives, environmental damage, and expensive repairs (Saxena et al., 2008). Failure prediction generally use the engine's current characteristics to precisely estimate the engine's Remaining Useful Life (RUL) and inform maintenance choices based on these estimations (X. Wang et al., 2023). The Remaining Useful Life is defined as the duration between the device's present cycle and that point it ceases to operate effectively. RUL prediction technology has the capacity to significantly reduce aircraft accidents resulting from engine failures. Simultaneously, it can improve maintenance dependability (Mohd Saufi & Hassan, 2021) and decrease maintenance expenses (Ahn et al., 2021).

The techniques for forecasting system longevity may be generally classified into two categories: physics-based modelling approaches and data-driven methods. Physics-based techniques emphasise the identification of failure processes and depend on specialised physical knowledge on damage propagation, which is often intricate and challenging to acquire (Fan et al., 2011). Data-driven methodologies utilise statistical learning, machine learning, and deep learning to estimate RUL based on gathered run-to-failure data from machinery using multiple online monitoring sensors (X. Wang et al., 2023). In the age of big data, the development of deep learning-based, data-driven approaches for predicting remaining lifetime has become a prominent strategy and research priority (Li et al., 2022).

The growing dependence on turbofan engines in contemporary aviation underscores the essential requirement for precise and effective predictive maintenance procedures. This project aims to tackle the complexity RUL prediction by utilising developments in data-driven approaches, especially deep learning. This study seeks to deliver unique solutions by integrating sophisticated neural network topologies with attention mechanisms, therefore enhancing prediction accuracy and improving the efficiency and reliability of maintenance operations. This method highlights the revolutionary capacity of artificial intelligence in enhancing safety standards and operational excellence within the aviation industry.

1.2 Project Objectives

This research effort aimed to assess the viability and effectiveness of a GRU-LSTM hybrid model with attention mechanisms for estimating the Remaining Useful Life (RUL) of turbofan engines. The research was directed by the subsequent principal enquiries:

1. Can a GRU-LSTM hybrid model proficiently forecast RUL by using sequential data and encapsulating intricate temporal dependencies?
2. How can scaling, activation functions and attention mechanisms enhance model accuracy and interpretability?
3. Can exploratory data analysis yield essential insights into sensor patterns and operational abnormalities to improve model training and evaluation?

The project sought to enhance predictive maintenance procedures in the aviation sector by focussing on comprehensive data preparation, feature selection, and novel model architectures.

1.3 Report Structure

This report is organized into 12 main sections, each addressing specific aspects of the research project. Additional information and supporting materials are provided in the Appendices for reference.

- **Section 1**: Introduction – Provides an overview of the project, its background, and defined objectives.
- **Section 2**: Literature Review – Summarizes existing research in predictive maintenance, RUL prediction methodologies, and the advancements in deep learning techniques applicable to this domain.
- **Section 3**: Methodology – Details the approach and rationale adopted for this research, including the design and implementation of the GRU-LSTM hybrid model.
- **Section 4**: Requirement Analysis – Explains the dataset selection process, preprocessing requirements, and criteria for feature finalization.
- **Section 5**: Dataset Overview – Offers a comprehensive overview of the C-MAPSS dataset, highlighting its attributes, structure, and relevance to the research objectives.
- **Section 6**: High-Level Design – Describes the architectural design and software framework used to develop the GRU-LSTM hybrid model, incorporating attention mechanisms.

- **Section 7**: Techniques and Algorithms – Discusses the techniques applied in this research, including feature selection, attention mechanisms, and hybrid model design.
- **Section 8**: Implementation – Covers the detailed implementation of the proposed methodology, including data preprocessing, model training, and optimization.
- **Section 9**: Results and Discussion – Presents the findings from exploratory data analysis (EDA) and model evaluation, comparing results across different configurations.
- **Section 10**: Project Management – Explains the project timeline, risk management, and quality control processes adopted to ensure timely and effective execution.
- **Section 11**: Conclusion and Future Work – Summarizes the research outcomes, evaluates the project's impact, and suggests recommendations for future studies.
- **Section 12**: Student Reflections – Shares personal insights and reflections on the challenges, successes, and lessons learned throughout the research project.

2. Literature Review

In recent years, data-driven methods have gained significant attention for predicting the Remaining Useful Life (RUL) of complex systems (Si et al., 2011). Various studies have explored innovative approaches to enhance RUL estimation. Ahmad et al. (Ahmad et al., 2019) utilized dynamic regression models to predict the RUL of rolling element bearings, while Hu et al. (Hu et al., 2018) proposed a Wiener process-based method for RUL prediction in wind turbine bearings. Similarly, Huang et al. (Huang et al., 2017) introduced an adaptive skew-Wiener process model for more accurate predictions. Zhang et al. (Zhang et al., 2018) conducted a comprehensive review of Wiener-process-based methods, focusing on degradation data analysis and RUL estimation. Le et al. (K. Le Son et al., 2016) employed a noisy gamma deterioration process to estimate RUL, whereas Ling et al. (Ling et al., 2019) applied Bayesian and likelihood inference within two-phase degradation models under a gamma process framework.

Other researchers have combined data-driven approaches with advanced filtering techniques. Baptista et al. (Baptista et al., 2019) proposed a hybrid method combining data-driven models with a Kalman filter, and Son et al. (J. Son et al., 2016) predicted RUL using constrained Kalman filters to process noisy condition monitoring signals. Duong and Raghavan (Duong & Raghavan, 2018) presented a heuristic Kalman-optimized particle filter approach for RUL prediction. Hidden Markov models (HMMs) have also been widely explored in this field. Liu et al. (Q. Liu et al., 2015) developed an adaptive hidden semi-Markov model (HSMM) for multi-sensor health prognosis, while Z. Chen et al. (Z. Chen et al., 2019) employed an HMM with auto-correlated observations for RUL estimation and optimal maintenance policy formulation. G. Chen et al. (G. Chen et al., 2017) extended these methods by proposing a non-homogeneous hidden semi-Markov model (NHSMM) to address nonlinear multi-state deterioration in RUL prediction.

Despite their widespread application, these methods face notable limitations. As J. Chen et al. (J. Chen et al., 2019) observed, the deterioration processes of equipment are often nonlinear and multifaceted due to complex structures and variable working conditions, which pose challenges to traditional modelling techniques. This underscores the importance of exploring more robust and flexible methods, such as deep learning, to improve RUL prediction accuracy and reliability.

The RUL prediction methods based on statistical learning have shown promising results for specific system forecasting challenges. However, achieving high accuracy becomes increasingly difficult when dealing with longer time series and complex systems, often leading to forecast results that fail to meet prediction expectations (X. Wang et al., 2023). With the rapid development and widespread application of machine learning and deep learning, many researchers have advocated for adopting data-driven methods for RUL prediction. Advances in sensor technology, computer hardware, and big data processing have further enabled deep network structures to become more scalable and resilient, enhancing their applicability in complex tasks.

Deep Learning

Deep learning, a branch of machine learning, has emerged as a powerful approach to handle highly non-linear and variable data in its raw form without requiring extensive human intervention (Lecun et al., 2015). Characterized by its deep hierarchical structure, deep learning leverages multiple processing layers to automatically extract high-level representations from large-scale data, ultimately improving prediction and classification accuracy. Algorithms such as Long Short-Term Memory (LSTM), Deep Neural Networks (DNN), Auto-Encoders (AE), Deep Belief Networks (DBN), and Convolutional Neural Networks (CNN) have demonstrated superior performance over conventional machine learning techniques in various applications.

The success of these deep learning architectures spans numerous fields, including computer vision (Voulodimos et al., 2018), natural language processing (Collobert & Weston, n.d.), and machine health monitoring (Zhao et al., 2019). In the context of RUL prediction, the primary challenge lies in capturing the degradation characteristics inherent in time-series data. The accuracy of RUL prediction largely depends on the model's ability to effectively assimilate temporal information (Ye & Dai, 2018). However, existing life prediction methods based on deep learning often focus on individual time-series networks, which limits their capacity to extract and utilize comprehensive information from the data (S. Sun et al., 2024). This underscores the need for further innovation in deep learning architectures to address these limitations and improve the robustness of RUL prediction models.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep feedforward neural networks that utilize convolutional computation to perform multiple nonlinear transformations, enabling the capture of intricate dynamic features in time-series data within high-dimensional spaces. CNNs employ alternating stacks of convolutional and pooling layers to extract abstract spatial features, with backpropagation determining the convolutional kernel parameters. This hierarchical structure allows CNNs to traverse multiple hidden layers for comprehensive feature extraction. Ren et al. (Ren et al., 2021) proposed the Auto-CNN-LSTM method for RUL prediction of lithium-ion batteries, combining CNN and LSTM with an autoencoder to address the challenges of limited degradation data and instability caused by external factors. Similarly, Wang et al. (C. Wang et al., 2021) introduced a method combining a deep convolutional autoencoder (DCAE) and CNN for RUL prediction of rolling bearings by constructing a health indicator (HI) with DCAE and self-organizing map (SOM) networks. However, CNNs exhibit limitations in feature extraction, particularly for long-term prediction of high-reliability equipment involving complex temporal correlations (S. Sun et al., 2024). While CNNs are adept at capturing local spatial features, their effectiveness diminishes when handling sequential information in temporal data.

Recurrent Neural Networks (RNNs)

To address the challenges of time-series data modelling, Recurrent Neural Networks (RNNs) introduced the concept of a hidden state, enabling feature extraction from sequential data and subsequent conversion to output. RNNs' sequence-driven nature, revealed through their chaining feature, makes them well-suited for time-series data. Guo et al. (Guo et al., 2017) proposed an RNN-based health indicator (RNN-HI) for bearing RUL prediction, addressing unequal feature contributions and varying failure thresholds. Yu et al. (Yu et al., 2019) further demonstrated the effectiveness of a sensor-based RUL estimation method combining a bidirectional RNN autoencoder and similarity-based curve matching, achieving competitive results on turbofan engine and milling datasets. However, RNNs are limited by their short-term memory capacity, struggling to transfer information effectively over long sequences due to challenges like gradient vanishing and explosion during backpropagation.

Long Short-Term Memory (LSTM)

To overcome the above mentioned limitations, Long Short-Term Memory (LSTM) networks were developed, providing the ability to capture long-range dependencies and different time scales. Zheng et al. (Zheng et al., 2017) proposed an engine RUL prediction method based on deep LSTM, comprising two LSTM layers combined with two Feed-forward Neural Network (FNN) layers and an output layer, achieving superior performance compared to CNN-based models. Hsu et al. (Hsu & Jiang, 2018) utilized LSTMs to address RUL prediction for turbine engines, effectively extracting temporal dependencies from historical data. Liao et al. (Liao et al., 2018) employed LSTM with a bootstrap procedure for uncertainty estimation, achieving higher accuracy compared to CNN and LSTM approaches discussed in earlier studies (Tian, 2016); (Zheng et al., 2017) respectively. Additionally, Yuan et al. (Yuan et al., 2016) proposed an LSTM for fault location and RUL estimation of aeroengines under complex operations, hybrid failures, and strong noise conditions.

More recent advancements include Wu et al. (Wu et al., 2020), who introduced a novel deep LSTM approach to discover hidden long-term dependencies in sensor time-series signals for RUL prediction. A variant of LSTM, the Bi-directional LSTM, has also been utilized (J. Wang et al., 2019) for aircraft engine RUL estimation. This architecture captures bidirectional temporal dependencies by learning long-range information from both forward and backward contexts of the input sequence, enhancing its ability to process sequential data effectively.

Gated Recurrent Unit (GRU)

Another significant variant of RNNs recently utilized for RUL estimation is the Gated Recurrent Unit (GRU), an enhanced LSTM model with fewer parameters. J. Chen et al. (J. Chen et al., 2019) proposed a novel approach for RUL estimation of nonlinear degradation processes by combining Kernel PCA (KPCA) for dimensionality reduction and nonlinear feature extraction with GRU for addressing long-term dependency issues. This method allows recurrent units to adaptively extract dependencies of different time scales. Song et al. (Song et al., n.d.) developed a GRU-based recurrent neural network (RNN) for battery RUL prediction, highlighting its computational efficiency and ability to address long-term relationship challenges. Zhou et al. (Zhou et al., 2022) introduced a Kullback-Leibler divergence-based health indicator (KLD-HI) combined with a reinforced memory gated recurrent unit (RMGRU) network for bearing RUL prediction. Similarly, Cao et al. (Cao et al., 2021) proposed a transfer learning method based on bidirectional Gated Recurrent Unit (TBiGRU) to enhance RUL prediction accuracy for bearings under varying working conditions. S. Wang et al. (S. Wang et al., 2019) presented a method integrating Hidden Markov Model (HMM) with an improved GRU network enhanced through Moth Flame Optimization (MGRU) for evaluating health conditions and predicting the RUL of slewing bearings.

Hybrid Models

Hybrid models have also emerged as a promising approach in RUL prediction. Jafari and Byun (Jafari & Byun, 2023) developed a hybrid deep learning model integrating CNN and GRU architectures for lithium-ion battery RUL prediction. By processing data through parallel CNN layers, the model outperformed traditional architectures like LSTM, GRU, and CNN-LSTM on NASA datasets. Liu et al. (Y. Liu et al., 2019) proposed an ensemble RUL prediction method combining Bayesian Model Averaging (BMA) with Long Short-Term Memory Networks (LSTMNs), which effectively improved performance and quantified uncertainty through an online iterated training strategy. This approach demonstrated reliable predictions on lithium-ion battery datasets. In a different domain, Girsang and Stanley (Girsang & Stanley, 2023) developed a hybrid LSTM-GRU model for cryptocurrency price prediction, incorporating historical price data with social media sentiment analysis using the FinBERT model, achieving superior predictive accuracy for Ethereum and Solana. Ragb et al. (Ragb et al., n.d.) proposed a GRU-LSTM hybrid regression model for predicting house prices using the Boston housing dataset. The model achieved better accuracy and lower RMSE than individual networks and other state-of-the-art methods.

Scaled Dot-Product Attention Mechanism

In this research, the scaled dot-product attention mechanism, as proposed by Vaswani et al. (Vaswani et al., 2017), was employed due to its efficiency and robustness in handling sequential data, particularly for predicting the Remaining Useful Life (RUL) of jet engines. This mechanism computes dot products of queries and keys, scales them by $1/\sqrt{d_k}$, and applies a SoftMax function to derive attention weights, which are subsequently used to compute a weighted sum of the values. Compared to additive attention, scaled dot-product attention is computationally more efficient and memory-efficient due to optimized matrix multiplication (Vaswani et al., 2017). The inclusion of the scaling factor addresses gradient saturation issues common in high-dimensional data, ensuring numerical stability and effective learning. These advantages make it an ideal choice for integration into the GRU-LSTM hybrid model, enabling the model to focus on critical temporal features without introducing significant computational overhead. Du et al. (Du et al., 2020) demonstrated the effectiveness of scaled dot-product attention in a hierarchical attention-based transfer learning model for biomedical question answering. Utilizing BERT for semantic representation and scaled dot-product attention for interaction clues, their model achieved state-of-the-art performance on the BioASQ Task B dataset, surpassing previous solutions without relying on handcrafted features.

Swish Activation Function

Additionally, the Swish activation function, as proposed by Ramachandran et al. (Ramachandran et al., 2017), was integrated into this research model due to its unique properties and effectiveness in enhancing deep learning performance. Swish surpasses traditional activation functions like ReLU by being smooth, non-monotonic, and bounded below while unbounded above. These attributes allow Swish to enable more nuanced gradient propagation during training, which is especially advantageous for deeper architectures. Their study has demonstrated that replacing ReLU with Swish consistently improves performance across various architectures, even under hyperparameters optimized for ReLU. For instance, in Mobile NASNet-A, Swish yielded a 0.9% increase in classification accuracy on ImageNet. Within the GRU-LSTM hybrid model, Swish plays a vital role in capturing complex temporal dependencies and enhancing predictive accuracy without necessitating significant architectural modifications.

Dropout Regularization

To mitigate the risk of overfitting, particularly in few-shot datasets, dropout was employed as a regularization technique. As highlighted by W. Sun et al. (W. Sun et al., 2016), dropout prevents complex co-adaptation of neurons to training data by randomly deactivating a subset of neurons during each training iteration. This approach reduces the likelihood of overfitting and ensures that the model generalizes well to unseen data. By incorporating dropout, the GRU-LSTM hybrid model effectively addresses the challenges posed by limited training data, maintaining its robustness and predictive reliability.

Robust Scaler

Robust Scaler normalization has been successfully applied in various fields to enhance the robustness and accuracy of machine learning models. (Khoirunnisa & Ramadhan, 2023) proposed a malaria detection system utilizing a Random Forest model combined with Robust Scaler normalization and K-fold cross-validation to improve the model's performance and robustness against outliers. This approach involved experimenting with optimal hyperparameters and achieved an impressive accuracy of 82%, showcasing the effectiveness of machine learning in healthcare decision-making. Similarly, (Reddy et al., 2021) developed a method for diagnosing Parkinson's disease, incorporating sequential feature selection (SFS) and a range of classifier algorithms, including Random Forest, Logistic Regression, Support Vector Machines (SVM), Gaussian Naive Bayes, and K-Nearest Neighbors (KNN). By leveraging the AdaBoost classifier along with Randomized Search CV and Robust Scaler, the study achieved 100% accuracy and an AUC-ROC score of 1.0 using ten acoustic features. These results demonstrate the superior capability of Robust Scaler in handling data preprocessing, particularly in scenarios where outliers significantly influence the dataset. The incorporation of this technique in diverse domains underscores its effectiveness in improving model performance and reliability.

The advancements in predictive maintenance and Remaining Useful Life (RUL) estimation have resulted from a shift from traditional physics-based and statistical methods to data-driven approaches utilising deep learning architectures. Techniques including CNNs, RNNs, LSTMs, and GRUs have shown considerable advancements in managing intricate temporal dependencies within sequential data. The incorporation of hybrid models and attention mechanisms has improved predictive accuracy and interpretability, while advancements such as the Swish activation function, dropout regularisation, and RobustScaler have optimised model performance. Despite these advancements, challenges including noisy data, feature inconsistencies, and the necessity for robust architectures remain prevalent. This study aims to fill existing gaps by creating a GRU-LSTM hybrid model utilising scaled dot-product attention to enhance Remaining Useful Life (RUL) prediction for turbofan engines, thereby contributing to the expanding literature on predictive maintenance.

3. Methodology

3.1 Software Development Life Cycle (SDLC) Model

This study employs an Agile Software Development Life Cycle (SDLC) model because of the project's iterative and flexible characteristics. In contrast to the other approach, Agile facilitates ongoing enhancement and incremental development, rendering it suitable for projects that need regular experimentation and refining. The Agile approach prioritises adaptability, cooperation, and responsiveness to alterations, which were essential for tackling the obstacles faced throughout this research.

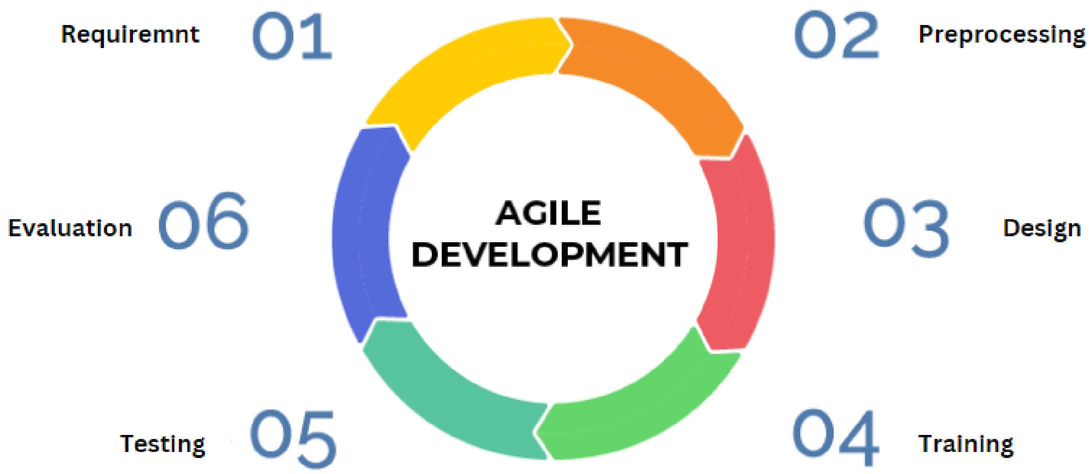

Figure 1: Agile Development (create by me)

3.2 Agile SDLC

Adaptability to Evolving Specifications:

The project originally intended to design and assess a Temporal Convolutional Network (TCN) integrated with an attention mechanism for Remaining Useful Life (RUL) forecasting. Nevertheless, because to implementation difficulties and resource limitations, the approach transitioned to a GRU-LSTM hybrid model. This change was facilitated by the Agile approach, which permitted incremental modifications to the project objectives and procedures.

Incremental Development:

The research encompassed several rounds of experimentation, including the adjustment of dropout rates, evaluation of diverse regularisation strategies, and optimisation of the scaling process. The iterative methodology of Agile enabled methodical enhancements at every phase, guaranteeing optimal results.

Sequential but Iterative Process:

While the project adhered to a distinct sequence of preprocessing, model construction, training, and testing, the Agile methodology facilitated incremental enhancements within each step. Preprocessing stages were optimised to enhance feature selection, and hyperparameter adjustment during training was conducted repeatedly based on evaluation metrics.

3.3 Implementation in Research

Planning and Requirement Analysis:

The project scope was established to anticipate the Remaining Useful Life (RUL) of turbofan engines utilising deep learning algorithms. An evaluation of current methodologies recognised the TCN with attention mechanisms as a viable design.

Due to implementation issues with the TCN, the criteria were amended to concentrate on a GRU-LSTM hybrid model, drawing inspiration from its effective use in Forex price prediction.

Data Preprocessing:

The CMAPSS dataset underwent repeated pre-processing to mitigate noise, standardise characteristics, and identify pertinent sensors for Remaining Useful Life projection.

The procedure involved eliminating low-correlation features and utilising a RobustScaler for normalisation. Modifications to preprocessing were implemented in response to model performance during preliminary testing.

Model Design:

The hybrid GRU-LSTM architecture was developed progressively. Each layer was incrementally included and evaluated to determine its impact on overall performance. The architecture used GRU layers for computing efficiency and an LSTM layer to capture long-term relationships. Dropout regularisation, kernel regularisation, and a Scaled Dot-Product Attention technique were integrated sequentially to improve robustness.

Model Training:

The training technique included an iterative methodology, involving attempts to refine dropout rates, learning rates, and other hyperparameters. Performance was assessed using metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.

Testing and Evaluation:

The model was evaluated on the unobserved segment of the dataset, and predictions were compared with actual RUL values. To evaluate how accurate the predictions were, visualisation techniques were used. Repetitive enhancements throughout this phase guaranteed conformity with the project goals.

3.4 Conclusion

The Agile SDLC methodology offered the requisite adaptability and framework to address the changing requirements and iterative improvements of this project. The incremental and adaptable structure was crucial in addressing problems, including the transition from TCN to GRU-LSTM, and in guaranteeing the project's successful conclusion. This technique promoted methodical growth and underscored the need of adaptation in attaining research objectives.

4. Requirements

This section delineates the methodologies employed to finalise the project requirements. Given that the Agile technique was used for software development, as outlined in Section 3, the requirements were progressively improved and revised throughout the project to guarantee conformity with the research objectives.

4.1 Dataset Selection

To achieve the aims of this research, as outlined in Section 1.2, the selection of the right dataset was essential. To allow precise Remaining Useful Life (RUL) prediction, the dataset has to include enough operating data and sensor readings for turbofan engines.

The CMAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset was chosen following thorough investigation. This dataset, which has four subsets that correspond to various operational situations and failure scenarios, is extensively utilised in the field of aircraft predictive maintenance. The FD001 subset, which includes data for a single operating state and one fault mode, was selected for its appropriateness in training and verifying the GRU-LSTM hybrid model. The dataset comprises 21 sensor measures and operational parameters, containing roughly 20,631 training samples and 13,096 test samples.

4.2 Gap Analysis

A gap analysis was performed to identify opportunities for innovation and enhancement in the context of the increasing application of deep learning models in predictive maintenance. This entailed a comprehensive examination of existing research publications, concentrating on RNN, GRU, LSTM, and hybrid architectures for time-series forecasting. Although GRU-LSTM models have been utilised in domains such as foreign exchange price forecasting and real estate valuation, their use in Remaining Useful Life prediction of aircraft engines is yet inadequately investigated.

This research provided insights that shaped the design and implementation needs for the project, highlighting the necessity for iterative model revisions, attention mechanisms, and effective preprocessing strategies to manage high-dimensional data.

4.3 System Requirements

Training and evaluating deep learning models can be computationally intensive. To ensure smooth development and testing, a system with the following configuration was used:

Table 1: System Requirements

System Configuration	Details
System Type	x64-based Processor
Operating System	Windows 11 Home
Processor	AMD Ryzen 5 4600H with Radeon Graphics @ 3.00 GHz
RAM	8.00 GB
GPU	NVIDIA GeForce GTX 1650 Ti

This configuration provided sufficient computational resources for training GRU-LSTM hybrid model.

4.4 Functionality Requirements

The following functionality requirements were identified for the successful implementation of this research:

1. Preprocess the CMAPSS dataset:

- Add Remaining Useful Life (RUL) as the target variable.
- Normalize data using RobustScaler.
- Select relevant features by analysing correlations with RUL.

2. Perform exploratory data analysis (EDA) to identify correlations in sensor readings.

3. Split the dataset into training and testing subsets in an 70%-30% ratio.

4. Design and implement a GRU-LSTM hybrid model:

- Include multiple GRU and LSTM layers for capturing short-term and long-term dependencies.
- Integrate a Scaled Dot-Product Attention mechanism.
- Add dropout regularization to prevent overfitting.

5. Train the model iteratively:

- Tune hyperparameters such as dropout rates, learning rates, and regularization techniques.

6. Evaluate model performance:

- Use metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.

7. Visualize results:

- Compare predicted and actual RUL values using plots.

4.5 Non-Functionality Requirements

The following non-functionality requirements were established to ensure efficiency, scalability, and maintainability throughout the project:

1. Modular Design:

The codebase was organised into separate modules for preprocessing, model architecture, training, and assessment. This modular architecture guarantees efficient debugging, enables future improvements, and encourages code reutilization.

2. Efficiency on Machines with Limited Resources:

The program was engineered to function effectively on platforms with moderate processing capabilities. Methods like feature selection, RobustScaler normalisation, and dropout regularisation were employed to enhance resource efficiency while preserving performance.

3. Scalability:

The framework is scalable to support larger datasets or additional functionalities. For example:

- Adding more preprocessing methods (e.g., advanced feature extraction).
- Experimenting with additional model architectures, such as Transformer-based approaches or hybrid variations.
- Incorporating more complex evaluation metrics or visualization techniques.

This research study use the publicly accessible C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset, a standard dataset extensively utilised for predictive maintenance and health management research. The dataset was initially created and published by NASA, simulating a realistic big commercial turbofan engine. The C-MAPSS dataset offers operational data and sensor readings for monitoring and forecasting damage development in aviation engines, essential for assessing the Remaining Useful Life (RUL) of turbomachinery components.

5. Dataset Description

The C-MAPSS dataset comprises data produced by a high-fidelity simulation model of a 90,000 lb thrust-class turbofan engine. It functions inside a closed-loop system, integrating authentic engine control mechanisms and environmental parameters. The dataset has various subsets that illustrate distinct operating settings and breakdown mechanisms. This research picked the FD001 subset for its simplicity and appropriateness in training and verifying the GRU-LSTM hybrid model.

The FD001 subset comprises data for a single operating state and one failure mode, including 20,631 samples in the training set and 13,096 samples in the test set. Every sample comprises:

- Operational Parameters: Data on the engine's operational context, including altitude, and throttle configurations.
- Sensor Measurements: Time-series data from 21 sensors tracking critical engine characteristics, including fan speed, high-pressure turbine temperature, and core speed.

The collection is organised such that each row corresponds to a single operational cycle of a particular engine unit, with a distinct Unit Number allocated to identify each engine. The last operating cycle in the dataset signifies the failure point for each engine.

Table 2: Dataset Parameters (Saxena et al., 2008)

Symbol	Description	Units
T2	Total temperature at fan inlet	ºR
T24	Total temperature at LPC outlet	ºR
T30	Total temperature at HPC outlet	ºR
T50	Total temperature at LPT outlet	ºR
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

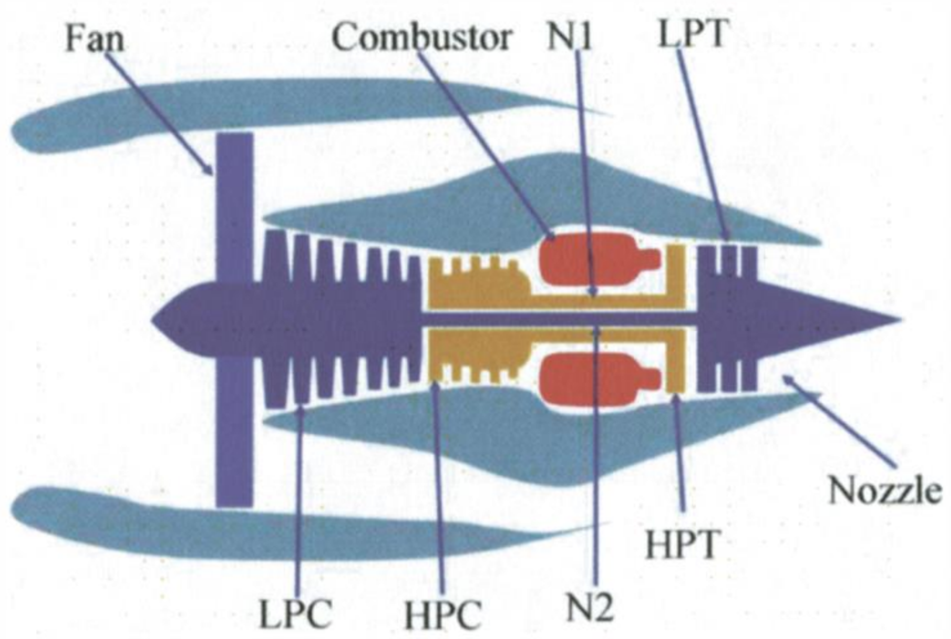

Figure 2: Simplified Diagram of Engine Simulated in C-MAPSS (Saxena et al., 2008)

The dataset was produced with the C-MAPSS simulator, which replicates the physics of turbofan engines across diverse operational profiles and failure scenarios. The simulator features adjustable input parameters, including fuel flow, climatic conditions, and degradation rates for components such as the fan, Low-Pressure Compressor

(LPC), High-Pressure Compressor (HPC), High-Pressure Turbine (HPT), and Low-Pressure Turbine (LPT). The output variables comprise sensor response surfaces and operability margins.

The C-MAPSS dataset is especially significant for Remaining Useful Life (RUL) prediction as it offers run-to-failure data, wherein each engine is observed until it fails. This facilitates the modelling of damage evolution over time and the creation of resilient prognostic algorithms.

6. Design

Using a hybrid GRU-LSTM model, the research project's design aims to create a reliable and effective framework for estimating the Remaining Useful Life (RUL) of turbofan engines. The design process consists of several interrelated phases, each enhancing the overall system. This section delineates the principal elements of the design and their respective functionalities.

6.1 Data Loading and Exploratory Data Analysis (EDA)

The initial phase of the design entails loading the CMAPSS dataset, particularly the FD001 subset, into memory. This subset comprises time-series data for a specific operating state and fault mode, with 21 sensor readings and operational parameters documented throughout numerous engine cycles.

An Exploratory Data Analysis was performed to enhance comprehension of the dataset and ascertain its appropriateness for Remaining Useful Life (RUL) prediction. Essential tasks in this phase encompass:

- Visualizing sensor data trends and distributions to detect outliers and discrepancies.
- Assessing correlations between sensor readings and the RUL to identify the most pertinent aspects for prediction.
- Assessing the data's integrity and rectifying any possible discrepancies.

6.2 Data Preprocessing

Data preprocessing was a critical component of the design, ensuring the dataset was prepared for model training and evaluation. The following transformations were applied:

1. Target Variable Creation:

A Remaining Useful Life (RUL) column was added to the dataset by calculating the difference between the maximum operational cycle and the current cycle for each engine unit.

2. Feature Selection:

Sensors with low correlation to RUL were removed to reduce dimensionality and enhance model performance.

3. Scaling:

The dataset was normalized using a RobustScaler to handle outliers and ensure consistency in the feature values.

4. Data Splitting:

The dataset was divided into training and testing subsets in an 70%-30% ratio. This split ensured that the model was trained on one portion of the data while being evaluated on unseen samples to test its generalization capabilities.

These preprocessing steps transformed the raw dataset into a clean and structured format, ready for use in model development.

6.3 Model Architecture Design

In order to capitalise on the advantages of both Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks, the GRU-LSTM hybrid model architecture was created. The subsequent essential design components were integrated:

- Four GRU layers (256, 128, 64, and 32 units) were used to effectively capture short-term dependencies in the time-series data.
- LSTM Layer: A solitary LSTM layer including 64 units was incorporated to capture long-term dependencies, essential for forecasting RUL from sequential sensor data.
- A Scaled Dot-Product Attention layer was incorporated into the model to concentrate on the most pertinent time steps, hence improving the interpretability and precision of predictions.
- Dropout Regularisation: Dropout layers with a rate of 0.4 were included following each recurrent layer to reduce overfitting and enhance model generalisation.
- Dense Output Layer: A terminal dense layer was employed to forecast the RUL as a continuous variable.

The architecture was meant to be flexible and scalable, enabling future expansions and experimentation with supplementary layers or methods.

6.4 Model Training and Optimization

The training process was iterative, involving multiple experiments to optimize the model's performance. Key aspects of this phase include:

1. Loss Function and Optimization:

The model was compiled with the Mean Squared Error (MSE) loss function and the Adam optimizer, which dynamically adjusts learning rates to accelerate convergence.

2. Hyperparameter Tuning:

Experiments were conducted to determine optimal values for dropout rates, learning rates, and batch sizes.

3. Training Parameters:

The model was trained with a batch size of 32 over 50 epochs. A validation split of 30% was used to monitor the model's performance during training.

The training phase ensured that the model learned meaningful patterns from the data while minimizing overfitting.

6.5 Model Evaluation and Results

Once the model was trained, its performance was evaluated on the test dataset. Evaluation metrics included:

- Mean Absolute Error (MAE): To measure the average error between predicted and actual RUL values.
- Root Mean Squared Error (RMSE): To assess the magnitude of prediction errors.
- R-squared (R^2): To evaluate the proportion of variance in RUL explained by the model.

The results were visualized by plotting the predicted RUL against the actual RUL values for each engine unit. These visualizations provided insights into the model's accuracy and highlighted areas for potential improvement.

7. Techniques

This section presents a comprehensive overview of the fundamental techniques and concepts utilised in this research to predict the Remaining Useful Life (RUL) of turbofan engines. The methods and algorithms that constitute the foundation of the GRU-LSTM hybrid model and its related processes are outlined in each subsection.

7.1 Feature Scaling

Feature scaling is an essential preprocessing step, especially for deep learning models, as it prevents features with large magnitudes from overshadowing those with lower magnitudes during training. This research utilised the RobustScaler for scaling because of its efficacy in managing outliers.

Both ordinary and robust scalers convert inputs to uniform scales. The distinction is in their methods of scaling raw input data. The RobustScaler modifies features by subtracting the median and scaling the data based on the interquartile range (IQR). This approach is resilient to outliers, as it use the median and interquartile range (IQR) rather than the mean and standard deviation, in contrast to standardisation. The transition is delineated as:

$$Scaled\ Value = \frac{Original\ Value - Input's\ Median}{Input's\ IQR}$$

This approach ensures consistent scaling across features, even in the presence of extreme values, making it particularly suitable for sensor data with varying ranges, as found in the CMAPSS dataset.

7.2 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a category of neural networks engineered to analyse sequential input by preserving information over time intervals. In contrast to conventional feedforward neural networks that process each input individually, recurrent neural networks (RNNs) utilise a memory mechanism via recurrent connections, allowing them to record relationships within sequences. RNNs are very proficient for applications including time-series forecasting, natural language processing, and sensor data analysis.

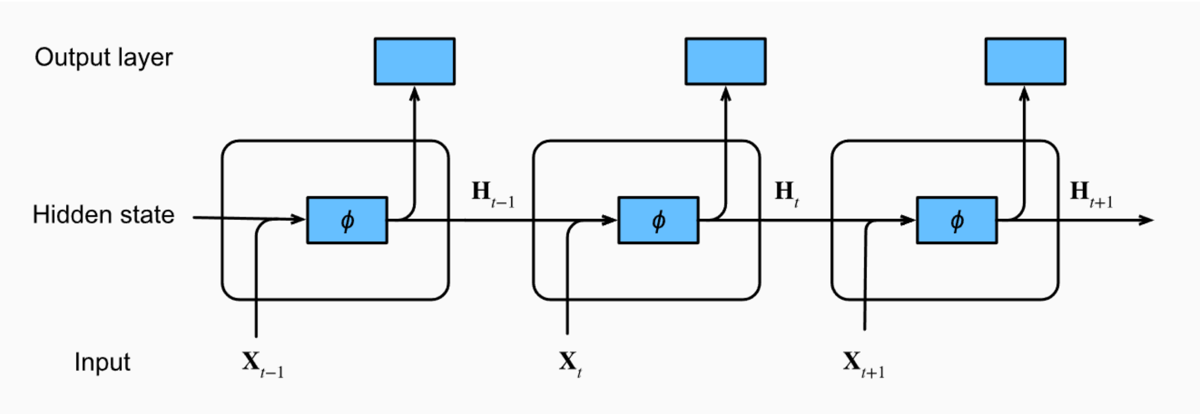

Figure 3: RNN with Hidden State (Dagar, n.d.-c)

At each time step t, the RNN processes the current input x_t and combines it with the hidden state h_{t-1} from the previous time step to produce the current hidden state h_t. The mathematical representation of this process is:

$$h_t = activation(U \cdot x_t + W \cdot h_{t-1} + b)$$

Where:

- h_t: Current state,
- x_t: Input at time t,
- U and W: Weight matrices for the input and hidden state, respectively,
- b: Bias term,
- $activation$: An activation function, such as ReLU or Swish.

The hidden state h_t captures both the current input information and the temporal dependencies from previous inputs, making it a central component of RNNs.

To generate the output, the hidden state h_t is transformed through an output layer. The output equation is defined as:

$$Y = O(V \cdot h + C)$$

Where:

- Y: Output of the network,

- h: Hidden state,
- V: Weight matrix mapping the hidden state to the output,
- C: Bias term for the output layer,
- O: Activation function applied to the output layer.

This equation shows how the hidden state, which encodes temporal dependencies, is used to produce predictions or outputs.

Standard RNNs, despite their efficacy, face difficulties in representing long-term dependencies because to the vanishing gradient problem, wherein gradients decrease with time during backpropagation. This constraint hinders the network's capacity to comprehend linkages across extended sequences. Advanced designs, such Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), tackle this problem by specialised gating techniques that facilitate the retention of information over prolonged durations.

This research employs the fundamental concepts of RNNs inside the GRU-LSTM hybrid architecture, wherein the hidden states progress across time to encapsulate sequential patterns in the sensor data. The model subsequently utilises these hidden states to produce precise forecasts for the Remaining Useful Life of turbofan engines.

7.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specific kind of Recurrent Neural Networks (RNNs) developed to address the shortcomings of conventional RNNs, including the vanishing gradient issue. This problem frequently obstructs RNNs from acquiring long-term relationships in sequential data. LSTMs mitigate this issue by including a cell state and various gating methods, allowing the network to selectively preserve, modify, or discard information as required.

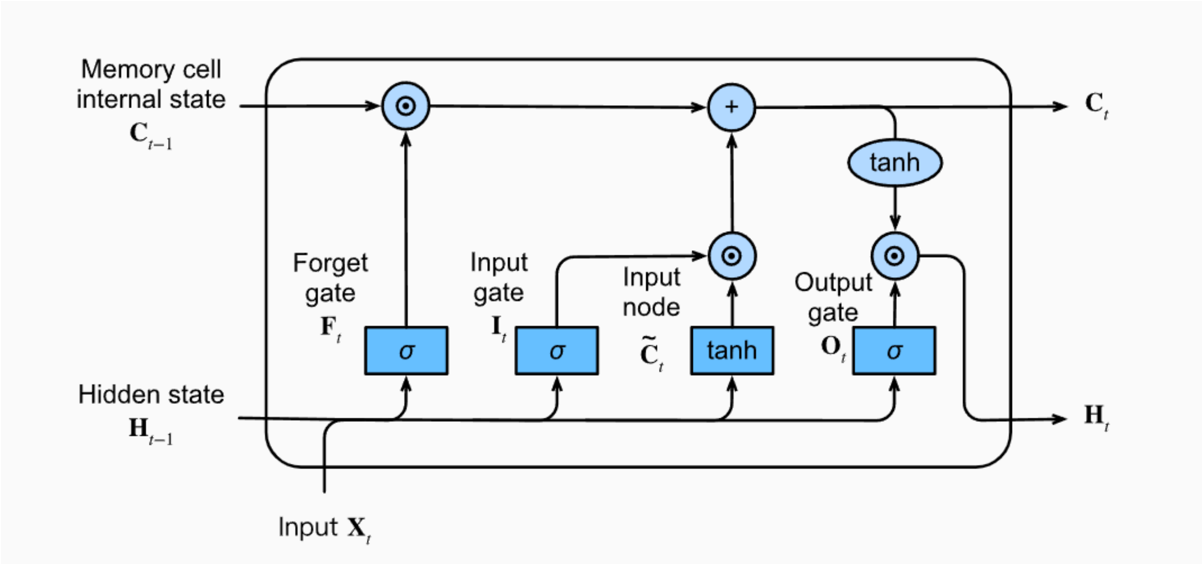

Figure 4: LSTM Model (Dagar, n.d.-b)

An LSTM cell consists of three primary gates - the forget gate, the input gate, and the output gate - that control the flow of information into and out of the cell state. The cell state acts as a memory that carries important information across time steps.

Forget Gate:

The forget gate eliminates information that is no longer pertinent in the cell state. Two inputs x_t (current input) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The output is processed by an activation function that produces a binary result. If the output for a certain cell state is 0, the information is discarded; if the output is 1, the information is preserved for future reference. The equation for the forget gate is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Where:

- W_f: Weight matrix associated with the forget gate.
- $[h_{t-1}, x_t]$: denotes the concatenation of the current input and the previous state.
- b_f: Bias with the forget gate.
- σ: Sigmoid activation function

Input Gate and Candidate Cell State:

The input gate facilitates the incorporation of valuable information into the cell state. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and x_t. Then, a vector is created using $tanh$ function that gives an output from -1 to $+1$, which contains all the possible values from h_{t-1} and x_t. Ultimately, the vector values and the controlled values are multiplied to provide the pertinent information.

- Equation for the input gate is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

- Candidate cell state, which computes potential new information:

$$C^\wedge{}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

We multiply the previous state by f_t, disregarding the information we had previously chosen to ignore. Next, we include $i_t \cdot C^\wedge{}_t$. This signifies the revised candidate values, modified according to the selected adjustments for each state value.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C^\wedge{}_t$$

Output Gate and Hidden State:

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs h_{t-1} and x_t.

The equation for the output gate is:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

At last, the values of the vector and the regulated values are multiplied to be sent as an output or input to the next cell:

$$h_t = o_t \cdot tanh(C_t)$$

In this project, the LSTM layer forms a key component of the GRU-LSTM hybrid model, with 64 units configured to learn long-term dependencies in sequential data. The choice of activation function enhances the model's ability to process complex temporal patterns. Detailed discussion of the activation function used in this research is provided in section 7.5.

7.4 Gated Recurrent Units (GRU)

Gated Recurrent Units (GRUs) are a streamlined version of Long Short-Term Memory (LSTM) networks. They were engineered to mitigate the vanishing gradient issue that constrains conventional Recurrent Neural Networks (RNNs) in simulating long-term relationships inside sequential data. GRUs decrease computational complexity relative to LSTMs by amalgamating the functions of the forget gate and input gate into a singular entity, termed the update gate, while preserving the capacity to choose keep or discard information.

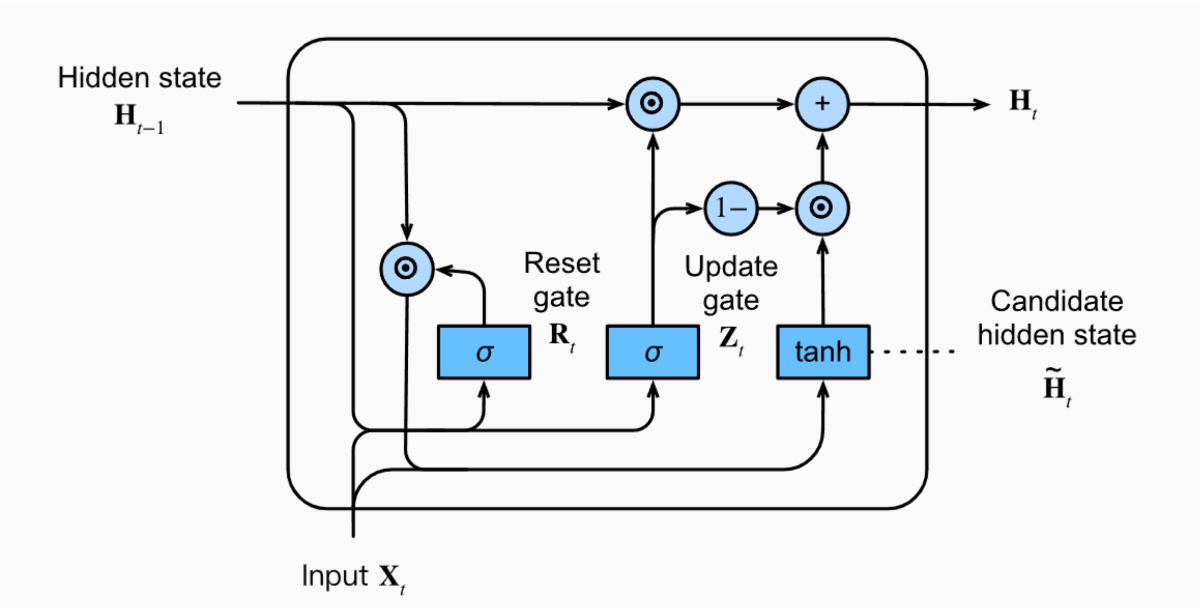

Figure 5: GRU Model (Dagar, n.d.-a)

GRUs are particularly suited for tasks requiring efficient computation and have been shown to perform comparably to LSTMs in many applications.

The GRU cell processes sequential data using the following steps:

Update Gate:

The update gate controls how much of the previous hidden state h_{t-1} should be retained and how much of the current input x_t should influence the new hidden state.

It is computed as:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Where:

- z_t: Update gate activation,
- W_z: Weight matrix for the update gate,
- b_z: Bias term,
- σ: Sigmoid activation function.

Reset Gate:

The reset gate determines how much of the previous hidden state h_{t-1} should be ignored. This allows the GRU to reset its memory when processing a new sequence.

It is calculated as:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

Where:

- r_t: Reset gate activation,
- W_r: Weight matrix for the reset gate,
- b_r: Bias term.

Candidate Hidden State:

The candidate hidden state $h^\wedge{}_t$ represents the potential new information to be added to the hidden state. It uses the reset gate r_t to determine how much of the previous hidden state contributes to the candidate state:

$$h^\wedge{}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h)$$

Where:

- W_h: Weight matrix for the candidate hidden state,
- b_h: Bias term.

Hidden State Update:

The final hidden state h_t is computed by combining the previous hidden state h_{t-1} and the candidate hidden state $h\hat{}_t$ weighted by the update gate z_t:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot h\hat{}_t$$

This equation ensures that the GRU selectively retains relevant information from the past while incorporating new data.

This research utilises GRU layers as the initial component of the GRU-LSTM hybrid architecture, commencing with 256 units designed to capture short- and medium-term relationships in the sensor data. The Swish activation function, as outlined in section 7.5, was utilised to improve learning dynamics and optimise the processing of intricate sequential patterns. The GRU layer collaborates with the LSTM layer to analyse the CMAPSS dataset, hence enhancing the precision of Remaining Useful Life (RUL) forecasts.

7.5 Swish Activation Function

The Swish activation function is a contemporary, non-monotonic activation function that has demonstrated considerable efficacy in deep learning challenges, surpassing conventional functions like Rectified Linear Unit (ReLU) and Sigmoid in several instances. Swish is notably proficient in identifying intricate patterns in data, rendering it an optimal selection for sequential models such as GRU and LSTM, as employed in this study.

The Swish activation function is defined as:

$$Swish(x) = x \cdot \sigma(x)$$

Where:

x: Input to the activation function,

$\sigma(x)$: Sigmoid function, defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The product of x and $\sigma(x)$ allows Swish to exhibit a non-monotonic behaviour, meaning the function is not strictly increasing or decreasing. This non-monotonicity enables the network to explore more nuanced representations of data.

This research utilises the Swish activation function in the GRU and LSTM layers to improve their learning efficacy. It substitutes the conventional activation functions for calculating the hidden state in both layers. Utilising Swish enhances the model's gradient flow and optimisation dynamics.

GRU with Swish:

The hidden state update equation in a GRU is modified to:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot Swish(h^{\wedge}{}_t)$$

Here, Swish is used to compute the candidate hidden state $h^{\wedge}{}_t$, enabling the GRU to model complex short and medium term dependencies more effectively.

LSTM with Swish:

In the LSTM, the hidden state h_t is updated as:

$$h_t = o_t \cdot Swish(C_t)$$

By replacing $tanh$ with $Swish$ in the hidden state computation, the LSTM can better capture long term dependencies and process intricate temporal patterns in the sequential data.

The use of the Swish activation function in the GRU-LSTM hybrid architecture employed in this study is a calculated design choice intended to enhance the model's efficacy on the CMAPSS dataset. Utilising the advanced characteristics of Swish, the network attains enhanced optimisation and precision, especially in forecasting the Remaining Useful Life (RUL) of turbofan engines.

7.6 Dropout Regularization

Regularisation is an essential method in deep learning that mitigates overfitting by enhancing the generalisation capacity of models. Among the several regularisation approaches, dropout has shown to be a straightforward yet very efficient method.

Dropout regularisation is especially advantageous for deep neural networks, as the likelihood of overfitting increases with the extensive number of parameters.

Dropout entails the random exclusion of a percentage of neurones during each training iteration. This inhibits the network from over-relying on particular neurones and compels it to acquire more resilient, dispersed representations.

7.7 Scaled Dot-Product Attention

The Scaled Dot Product The attention mechanism is a potent strategy extensively employed in contemporary deep learning models to capture connections among input characteristics, irrespective of their spatial distance. In contrast to recurrent architectures that handle data sequentially, attention mechanisms enable models to dynamically prioritise significant segments of the input sequence by assessing their significance. This is particularly advantageous for jobs related to temporal or sequential data, such as Remaining Useful Life (RUL) prediction.

The attention mechanism computes a weighted representation of the input sequence by focusing on relevant elements based on learned weights. It operates on three key components: queries (Q), keys (K), and values (V).

- Query (Q): Represents the current input (or feature) for which the attention weights are calculated.
- Key (K): Represents the features against which the query is compared to determine relevance.
- Value (V): Represents the information to be weighted and aggregated based on attention scores.

The computation involves the following steps:

Dot-Product Similarity:

The similarity between the query Q and key K is computed using a dot product

$$Attention\,Score = Q \cdot K^T$$

This measures how closely related the query is to each key.

Scaling:

To stabilize gradients when dealing with high-dimensional vectors, the attention scores are scaled by the square root of the key dimension d_k:

$$Scaled\ Score = \frac{Q \cdot K^T}{\sqrt{d_k}}$$

SoftMax Normalization:

The scaled scores are passed through a SoftMax function to convert them into probabilities, ensuring they sum to 1:

$$Attention\ Weights = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)$$

Weighted Summation:

The attention weights are applied to the value V vectors to Compute the final output:

$$Output = Attention\ Weights \cdot V$$

8. Implementation

This section delineates the software implementation of the GRU-LSTM hybrid model for predicting the Remaining Useful Life (RUL) of turbofan engines. All project code is developed in Python, utilising its comprehensive machine learning and deep learning packages. The TensorFlow and Keras libraries were employed to create and train the neural networks, whilst Scikit-learn and Pandas were utilised for data preparation and analysis. The libraries Matplotlib and Seaborn were employed for data visualisation.

8.1 Data Preprocessing

Data preparation is an essential phase in any machine learning endeavour, especially for sequential data such as the CMAPSS dataset utilised in this study. It entails converting unprocessed data into an organised and refined format appropriate for model training and assessment. To guarantee compatibility with the GRU-LSTM hybrid model, the preprocessing for this study concentrated on target variable calculation, feature selection, scaling, and reshaping.

Loading and Structuring the Data

The CMAPSS dataset was divided into three files: training data, testing data, and RUL values for the test engines. Each dataset was loaded into Pandas DataFrames for ease of manipulation. The columns in the training and testing datasets included:

- Index Columns: Engine unit numbers and time cycles.
- Operational Settings: Parameters related to engine operation.
- Sensor Measurements: Data from 21 sensors measuring various engine parameters.

Computing the Target Variable (RUL)

The Remaining Useful Life (RUL) was calculated for each time cycle of every engine in the training dataset. The calculation used the following formula:

$$RUL = Max\ Cycle - Current\ Cycle$$

Where:

- Max Cycle: The maximum time cycle for each engine.
- Current Cycle: The cycle number of the engine at any given point.

This process involved:

1. Grouping the data by engine unit number to determine the maximum cycle for each engine.
2. Subtracting the current cycle from the maximum cycle to compute the RUL for each data point.
3. Adding the computed RUL as a new column in the training dataset.

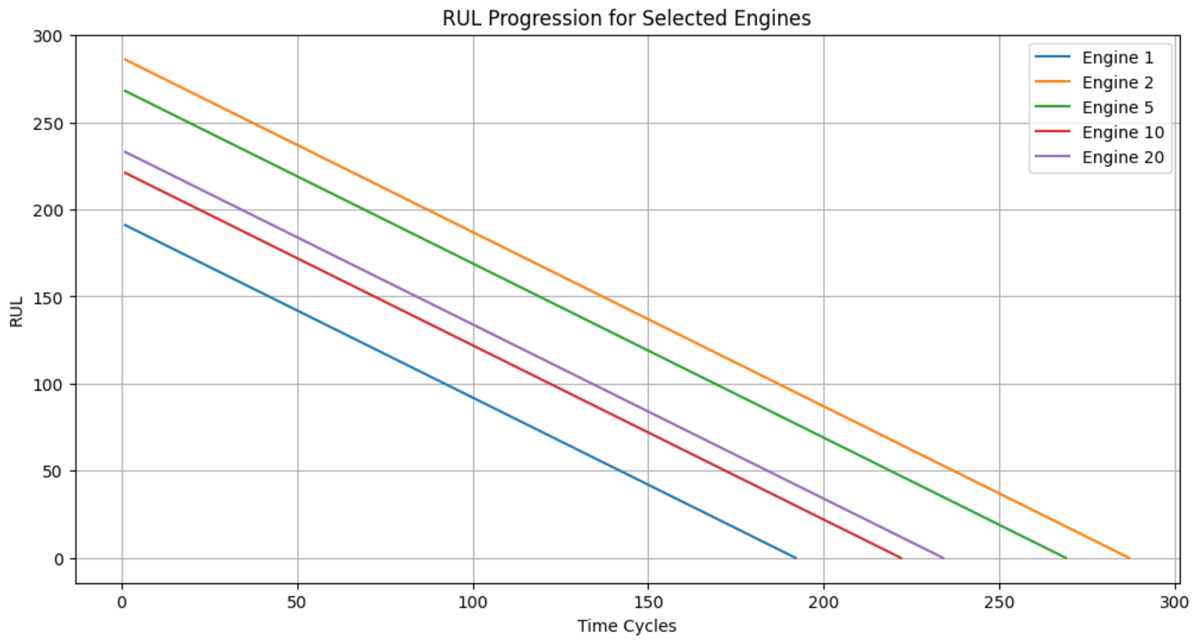

Figure 6: RUL Progression for Selected Engines

Figure 6 elucidates the evolution of RUL values throughout time cycles for specific engines within the dataset, hence enhancing the understanding of the RUL calculation.

This linear decline illustrates the anticipated deterioration of the engines over time, with each line denoting a distinct engine. The chart clearly illustrates the constant decline of RUL from the commencement to the conclusion of an engine's operating lifespan. This visualisation confirms the precision of the RUL calculation and offers insight into the uniformity of engine performance across various units.

Correlation Analysis and Feature Selection

To optimize the model's performance, irrelevant features were identified and removed. A correlation analysis was performed between each column and the target variable (RUL). Features with negligible correlation were excluded to reduce noise and computational complexity. Specifically:

- Sensors s_1, s_5, s_10, s_16, s_18, and s_19 were removed as they showed minimal correlation with RUL.
- Unit_number: This was dropped because it only acted as an identifier and had no predictive relevance.
- Operational Settings: The parameters setting_1, setting_2, and setting_3 were excluded as their correlation with RUL was found to be negligible.

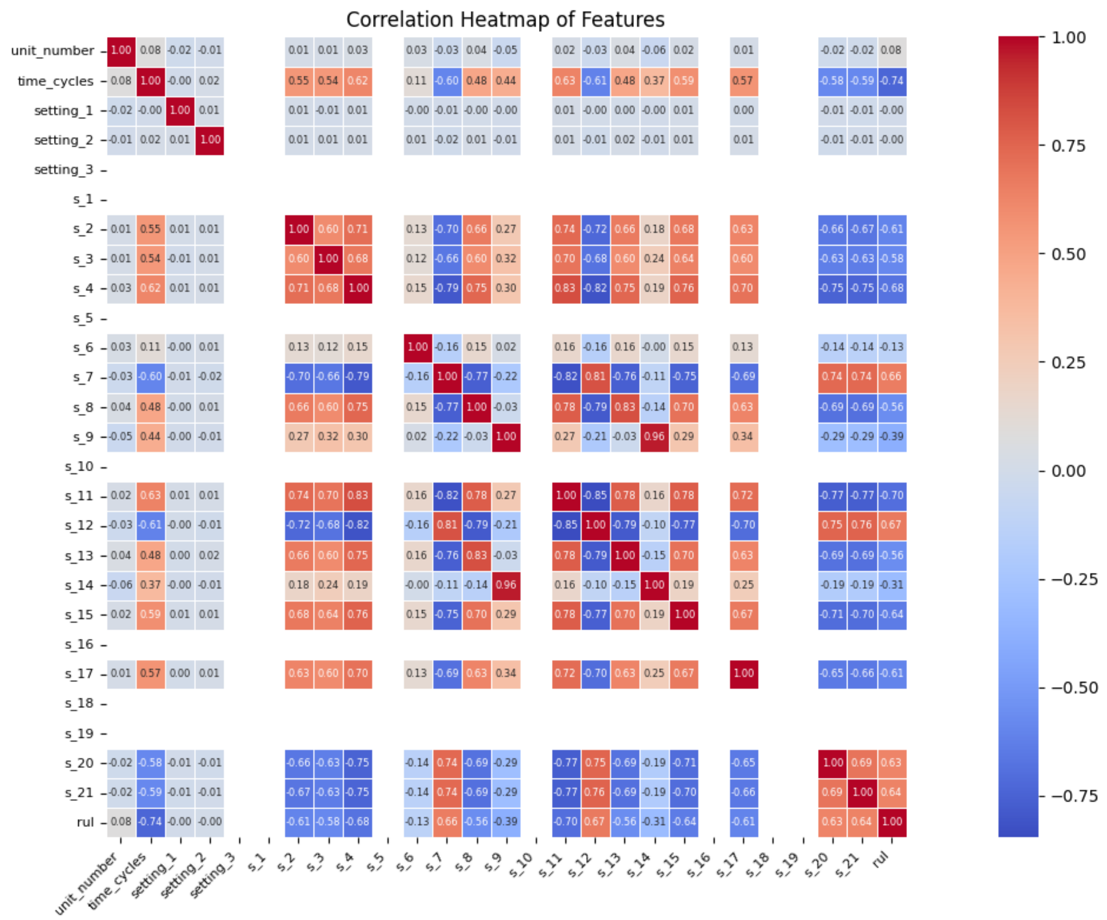

Figure 7: Correlation Heatmap of Features

By removing these features, the dataset was streamlined to include only those columns that provided meaningful input to the model.

Feature Scaling

Feature scaling is a vital preprocessing step in machine learning, particularly for deep learning models like GRU and LSTM, which are sensitive to the magnitude of input features. Without scaling, features with larger magnitudes could dominate the learning process, leading to biased models and slower convergence.

For this research, a RobustScaler was employed to normalize the sensor readings, as the dataset contained significant variations and potential outliers in sensor data. The Robust Scaler works by:

- Subtracting the median of each feature to centre the data.
- Scaling the data using the interquartile range (IQR) to ensure robust handling of outliers.

This method was selected because to its effectiveness in reducing the impact of outliers, which is especially crucial when dealing with real-world sensor data susceptible to measurement noise. Unlike Min-Max Scaling or Standard Scaling, which may exhibit excessive sensitivity to outliers, the Robust Scaler guarantees consistent and dependable normalisation for this application.

The scaling was implemented independently for the training, validation, and test datasets to avert data leakage. The solution utilised Python's scikit-learn module, guaranteeing uniformity across all data subsets.

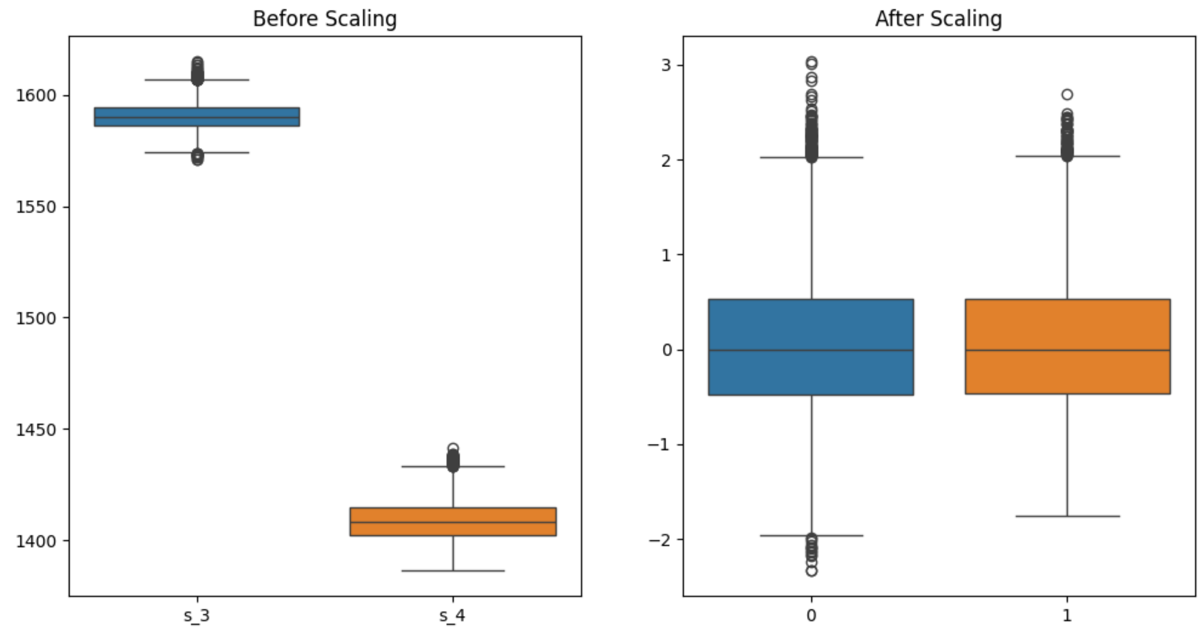

Figure 8: Effect of Robust Scaling on Sensor Data

The distributions of two randomly selected sensor readings, s_3 and s_4, are compared before and after using the RobustScaler in Figure 8 to further highlight the effects of feature scaling. The left panel of the image displays the unprocessed sensor results, emphasising notable discrepancies in magnitude and the occurrence of outliers. These fluctuations may dominate the learning process and skew the model if unscaled.

The identical sensor results are presented in the right panel after to scaling. The RobustScaler has effectively normalised the data by centring it at zero and scaling it within a uniform range. This transformation guarantees that both features contribute equally throughout model training, regardless of their initial magnitudes. Additionally, the scaling method reduces the influence of outliers, as seen by the more consistent boxplot distributions. This visualisation illustrates how feature scaling improves the dataset's appropriateness for deep learning models, hence enhancing the stability and accuracy of the GRU-LSTM hybrid model.

Splitting the Data

The pre-processed dataset was partitioned into three subsets: training, validation, and testing, to enhance model development and assessment.

- Approximately 70% of the dataset was designated for training the GRU-LSTM hybrid model. This subset facilitated the model's acquisition of patterns, temporal dependencies, and correlations between sensor data and the RUL target variable.
- Validation Data: Of the training data, 30% was allocated for validation throughout the training phase. This validation sample, including around 21% of the total dataset, was crucial for assessing the model's performance on unfamiliar material during training. It facilitated the optimisation of hyperparameters and assisted in reducing overfitting.
- Test Data: The residual 30% of the dataset was allocated for conclusive testing. This subset was omitted from both training and validation procedures, guaranteeing an impartial assessment of the model's efficacy on completely novel data. Only the final recorded cycle of each engine was utilised for the test data, according to the RUL prediction criteria for engines not included in the training set.

By partitioning the data in this systematic fashion, the model's efficacy could be precisely evaluated at each phase, so assuring reliable predictions and applicability to real-world contexts.

Reshaping for Sequential Input

Recurrent neural networks (RNNs) such as GRU and LSTM are designed to process sequential data by capturing temporal patterns and dependencies. These models require the input data to be in a three-dimensional format:

$$Input\ Shape = (samples, timesteps, features)$$

- Samples: The number of independent sequences (e.g., each engine's data across multiple cycles).
- Timesteps: The number of sequential steps processed at once. In this research, a single timestep is used for simplicity.
- Features: The number of attributes for each timestep (e.g., sensor readings after feature selection).

In this research, the pre-processed data was restructured to a format of (samples, timesteps, features). This format guarantees compliance with the sequential characteristics of GRU and LSTM layers, allowing the model to proficiently learn from temporal relationships within the input.

Single-step sequences were used to decrease architectural complexity and concentrate on learning short-term connections between characteristics and Remaining Useful Life estimates. Future research may investigate multi-step sequences to represent extended temporal interdependence.

The data was restructured with NumPy's reshape function to conform to the necessary format before to inputting it into the GRU-LSTM hybrid model.

8.2 Model Architecture

The GRU-LSTM hybrid model was meticulously designed to leverage the strengths of both GRU and LSTM layers for sequential data processing. The architecture comprised the following key components:

Gated Recurrent Unit (GRU) Layers:

Four GRU layers were integrated, with the number of units systematically diminishing from 256 to 32. This incremental decrease enabled the model to extract temporal information across various levels of abstraction. The GRU layers were selected for their computational efficiency and capacity to grasp short-term relationships in sequential input. Every GRU layer employed the Swish activation function, facilitating improved learning through the provision of smoother gradients. Dropout regularisation was used following each layer to reduce overfitting and promote generalisation.

Long Short-Term Memory (LSTM) Layer:

A LSTM layer with 64 units was used to capture long-term dependencies in the sensor data. The LSTM layer employed the Swish activation function to enhance the model's optimisation efficiency. The hybrid design, by integrating GRU and LSTM layers, efficiently managed both short-term and long-term temporal associations in the input sequences, hence improving predicted accuracy.

Scaled Dot-Product Attention Mechanism:

A scaled dot-product attention mechanism was used to enhance the model's emphasis on crucial time steps. This layer dynamically modified weights for each input time step, enabling the model to prioritise pertinent temporal information while diminishing the significance of irrelevant ones. The attention mechanism markedly enhanced both interpretability and accuracy by allowing the model to focus on the most informative parts of the input sequences.

Output Layer:

A dense output layer with a single neuron was added at the end of the architecture. This layer predicted the RUL of the engine for each input sequence, providing a continuous value representing the number of cycles remaining before failure.

Regularization Techniques:

To augment the model's resilience, dropout regularisation was systematically implemented following each GRU and LSTM layer. Dropout mitigated overfitting by randomly deactivating neurones throughout the training process. To further avoid overfitting and guarantee consistent convergence, regularisation was added to the GRU and LSTM layers to penalise high weights.

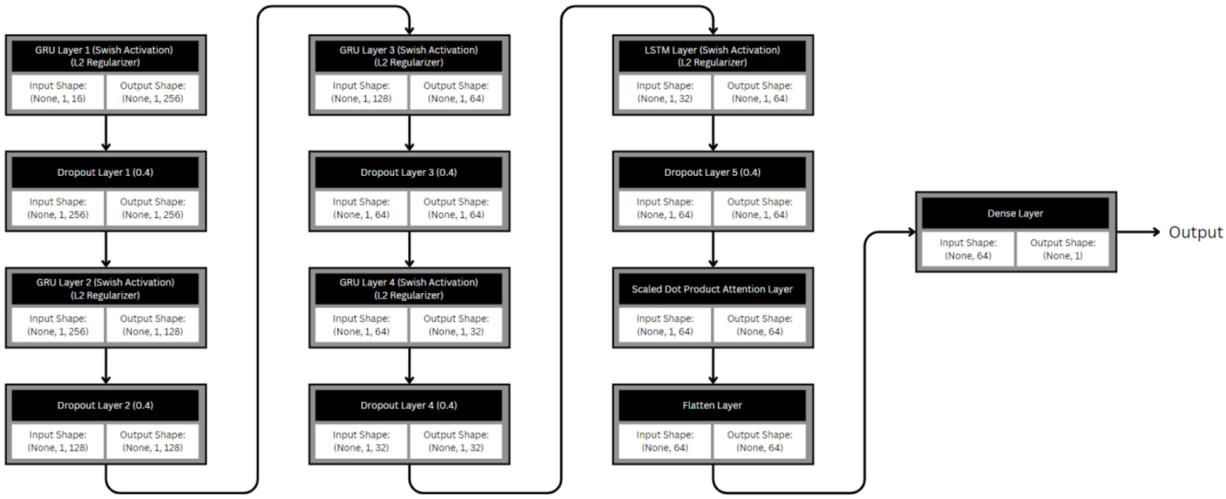

Figure 9: GRU-LSTM Hybrid Model Architecture

8.3 Training and Optimization

The GRU-LSTM hybrid model was created via the Adam optimiser with a learning rate of 0.001, chosen for its adaptable learning proficiency and effective optimisation in deep learning architectures. To measure prediction mistakes and direct the model's learning process, the Mean Squared Error (MSE) was used as the loss function.

Key Training Parameters:

- Batch Size: Set to 32, balancing computational efficiency and convergence speed during training.
- Epochs: The model was trained for 50 iterations to allow sufficient learning while avoiding overfitting.
- Validation Split: 30% of the training data was reserved for validation, enabling performance monitoring on unseen data during training.

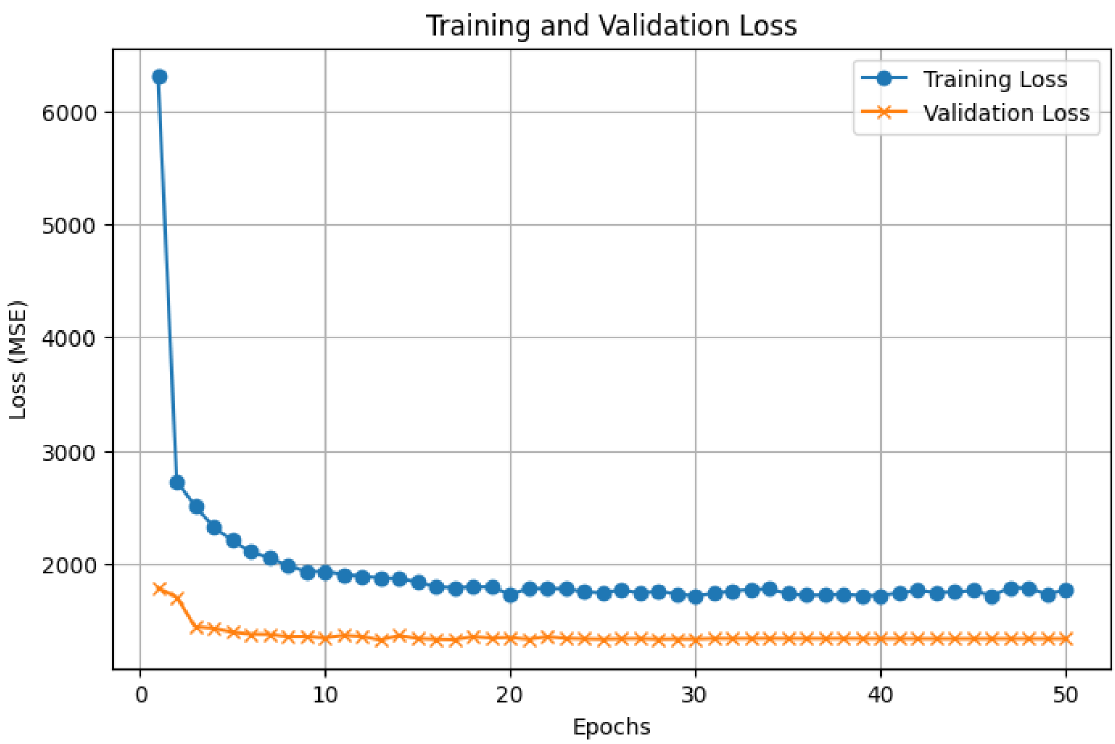

Figure 10: Training and Validation Loss

Figure 10 illustrates the training and validation loss curves throughout 50 epochs to elucidate the training process. The loss is quantified using Mean Squared Error (MSE), which acts as the optimisation objective for the GRU-LSTM hybrid model. The training loss curve (blue line) exhibits a steep reduction in the early epochs, signifying that the model rapidly acquires patterns from the data. Following around 10 epochs, the curve starts stabilisation, indicating convergence.

The validation loss curve (orange line) exhibits a same tendency, but with a continually decreased amplitude relative to the training loss. This signifies successful generalisation throughout the training procedure. The plateau in validation loss after 10 epochs indicates the effect of the ReduceLROnPlateau learning rate scheduler, which optimises the learning process to enhance convergence while preventing overfitting. The consistent and parallel trajectories of the two curves indicate that the model attains an effective equilibrium between bias and variance, hence assuring strong performance on novel data.

8.4 Metrics

The evaluation of the GRU-LSTM hybrid model for Remaining Useful Life (RUL) prediction relied on several performance metrics to quantify the accuracy and robustness of the model. These metrics provided insights into the model's ability to generalize and accurately predict RUL based on the input data.

Mean Squared Error (MSE)

MSE was used as the loss function during model compilation to optimize the weights of the neural network. It measures the average squared difference between predicted and actual RUL values. Smaller MSE values indicate better predictive accuracy, with the formula defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y^\wedge{}_i)^2$$

Where:

- y_i: Actual RUL value for the i-th observation.
- $y^\wedge{}_i$: Predicted RUL value for the i-th observation.
- n: Total number of observations.

MSE is particularly useful as it penalizes larger errors more heavily, ensuring that the model focuses on minimizing significant deviations.

Mean Absolute Error (MAE)

MAE was included as a metric during the model compilation to provide an additional measure of performance. Unlike MSE, MAE calculates the average magnitude of errors in the predictions, without squaring them. The formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y^\wedge{}_i|$$

MAE is a more interpretable metric as it represents the average absolute error in the same units as the target variable, RUL.

Root Mean Squared Error (RMSE)

RMSE was calculated post-training to provide a complementary metric for error evaluation. It is derived as the square root of the MSE, making it more interpretable as it represents the average magnitude of error in the same units as the target variable. The formula is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE penalizes large errors more than MAE, making it a suitable choice for assessing the impact of significant prediction errors.

R-Squared (R^2)

R^2 was used to measure how well the model explains the variability in the RUL. It is a statistical measure that indicates the proportion of variance in the target variable that can be explained by the input features. The formula is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where:

- $\bar{y}$: Mean of actual RUL values.

An R^2 value close to 1 indicates that the model explains most of the variance in the target variable.

Summary of Metrics

The combination of MSE, MAE, RMSE, and R^2 provided a comprehensive evaluation of the GRU-LSTM model's performance. While MSE and MAE were monitored during training and validation, RMSE and R^2 offered additional insights into the model's accuracy and generalizability. These metrics collectively ensured a thorough evaluation of the model's ability to predict RUL effectively.

9. Results

This section presents a thorough assessment of the GRU-LSTM hybrid model's efficacy in forecasting the Remaining Useful Life (RUL) of turbofan engines. The findings are categorised into three primary elements: exploratory data analysis, a comparison of projected and actual RUL values for the test dataset, and a comprehensive evaluation of the model's performance utilising several assessment criteria. The evaluations together illustrate the model's capacity to detect temporal relationships in the data and precisely forecast RUL, underscoring its efficacy and dependability for predictive maintenance applications.

9.1 Exploratory Data Analysis (EDA)

To understand the characteristics and distributions within the CMAPSS dataset, exploratory data analysis was conducted before model training.

Maximum Time Cycles Distribution

Figure 11 illustrates a histogram depicting the distribution of maximum time cycles for individual engines within the dataset, along by an overlay Kernel Density Estimate (KDE) curve to offer a more refined depiction of the data's overarching pattern. This visualisation emphasises the varied operating lifespans of engines, providing insights into the dataset's diversity.

The majority of engines have maximum time cycles concentrated between 150 and 250 cycles, with the highest frequency occurring at the 200-cycle threshold. This central tendency indicates that the majority of engines in the dataset adhere to a very uniform operating lifespan prior to reaching the conclusion of their useful life or necessitating substantial repair. The distribution is asymmetrical, since the histogram displays a pronounced skew. A diminished fraction of engines attain maximum cycles over 250, with the distribution tapering off at elevated levels.

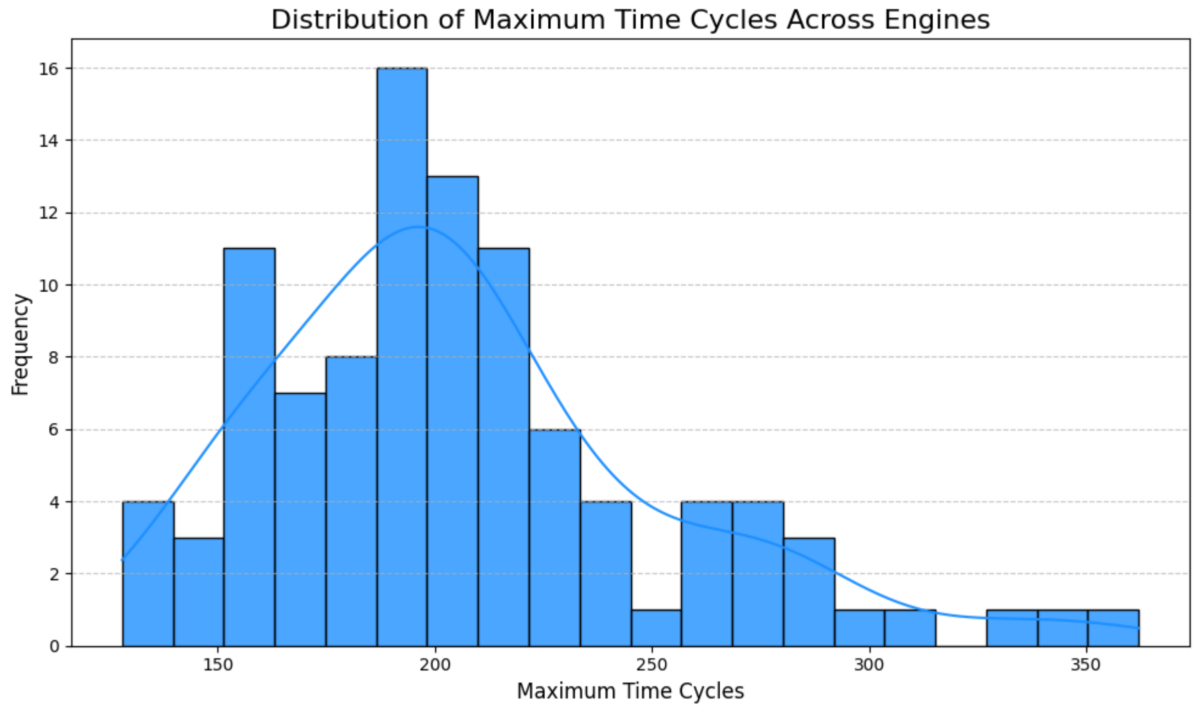

Figure 11: Distribution of Maximum Time Cycle Across Engines

Several engines have outstanding performance, with maximum cycles of 350 or greater. These outliers presumably signify engines functioning under ideal conditions, benefiting from enhanced maintenance regimens or encountering less rigorous usage patterns. Engines having a maximum of less than 150 cycles may signify accelerated deterioration owing to more severe operating conditions or mechanical defects.

The large range of maximum time cycles, from around 100 to over 350 cycles, underscores considerable heterogeneity in engine lifespans. This variety enhances the dataset but also presents obstacles for predictive models. Engines with exceptionally high or low maximum cycles may serve as outliers that might distort the training process of the GRU-LSTM hybrid model. These outliers may elevate the model's loss if it fails to generalise well across such extremes, underscoring the necessity for rigorous preprocessing and regularisation methods.

The distribution of maximum time cycles indicates significant trends within the dataset, with the majority of engines adhering to a predictable lifetime and a few exhibiting distinct performance traits. This variation highlights the necessity of utilising sophisticated modelling strategies, including attention mechanisms and dropout regularisation, to guarantee that the model can proficiently train and generalise throughout the whole spectrum of engine lifespans.

Analysis of Sensor Trends for Engines with Long and Short Lifespans

This section explores the temporal trends of two critical sensors, "HPC Outlet Static Pressure" (sensor s_11) and "Ratio of Fuel Flow to Ps30 (pps/psia)" (sensor s_12), both selected for their high correlation with Remaining Useful Life (RUL). By analysing the behaviour of these sensors over time, we gain insights into engine performance and the factors influencing RUL. Figures 12 and 13 illustrate the smoothed trends for these sensors, highlighting differences between engines with the longest lifespans (Engines 69 and 92) and the shortest lifespans (Engines 91 and 36). A rolling average was applied to reduce noise, enabling a clearer visualization of meaningful patterns.

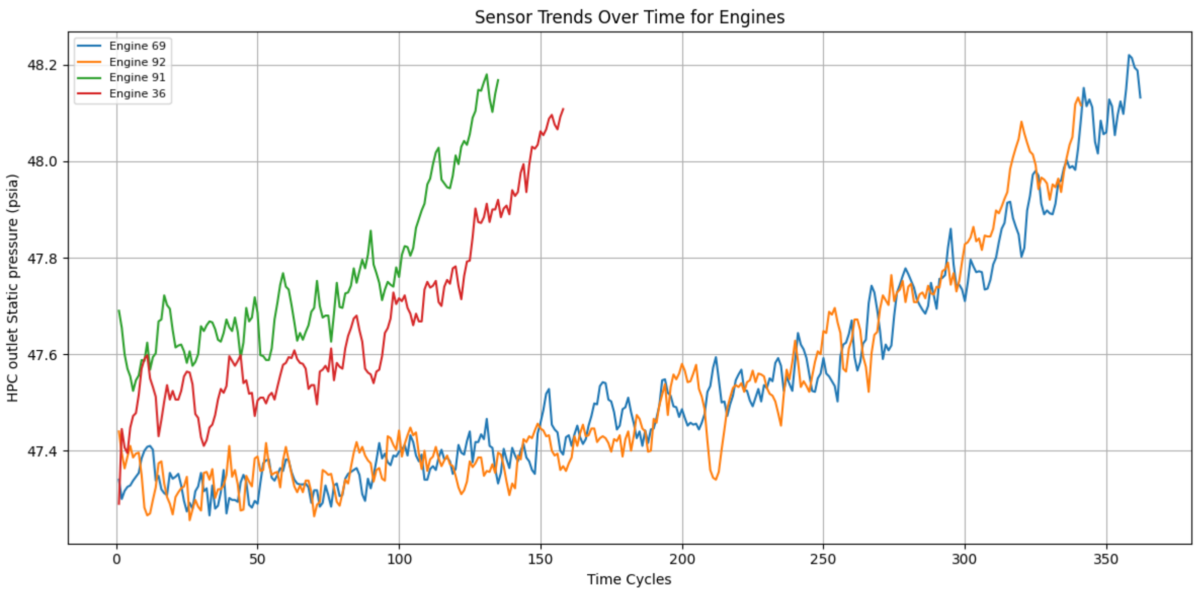

Figure 12: Sensor Trends Over Time for Engines (sensor s_11)

Figure 12 presents the trends for HPC Outlet Static Pressure (sensor s_11), which exhibits distinct behaviours between long and short lived engines. Engines with longer lifespans, such as Engine 69 and Engine 92, demonstrate a consistent and gradual increase in HPC Outlet Static Pressure across their operational cycles. Early and mid-stages of these engines show relatively stable pressure levels, with only minor fluctuations. Toward the end of their lifecycles, a steady rise in pressure becomes apparent, likely reflecting natural wear on the engine components. This smooth and predictable progression suggests robust operational performance and effective maintenance practices.

Conversely, the trends for engines with reduced lifespans, such as Engine 91 and Engine 36, exhibit anomalies and unpredictable patterns. Notable variations in HPC Outlet Static Pressure are detected even during the first phases of their lifecycles, indicating possible performance instabilities or irregularities. Moreover, a more pronounced rise in pressure transpires sooner in the lifetime, signifying expedited wear or deterioration. These patterns indicate fundamental problems, such insufficient maintenance or intrinsic design defects, that lead to the engines' diminished longevity. The substantial disparity between the consistent tendencies of long-lived engines and the unpredictable behaviour of short-lived engines underscores the essential function of sensor s_11 in detecting temporal patterns pertinent to Remaining Useful Life prediction.

Likewise, Figure 13 examines the trends for the Ratio of Fuel Flow to Ps30 (sensor s_12), which is the second most associated characteristic to RUL. In engines with extended lifespans, such Engine 69 and Engine 92, the sensor data demonstrate a slow and consistent decrease with time, accompanied by negligible fluctuation. This steadiness indicates effective fuel usage and reliable operational performance, which may enhance the engines' longevity. The observed trends suggest a properly maintained engine that experiences natural wear without substantial interruptions.

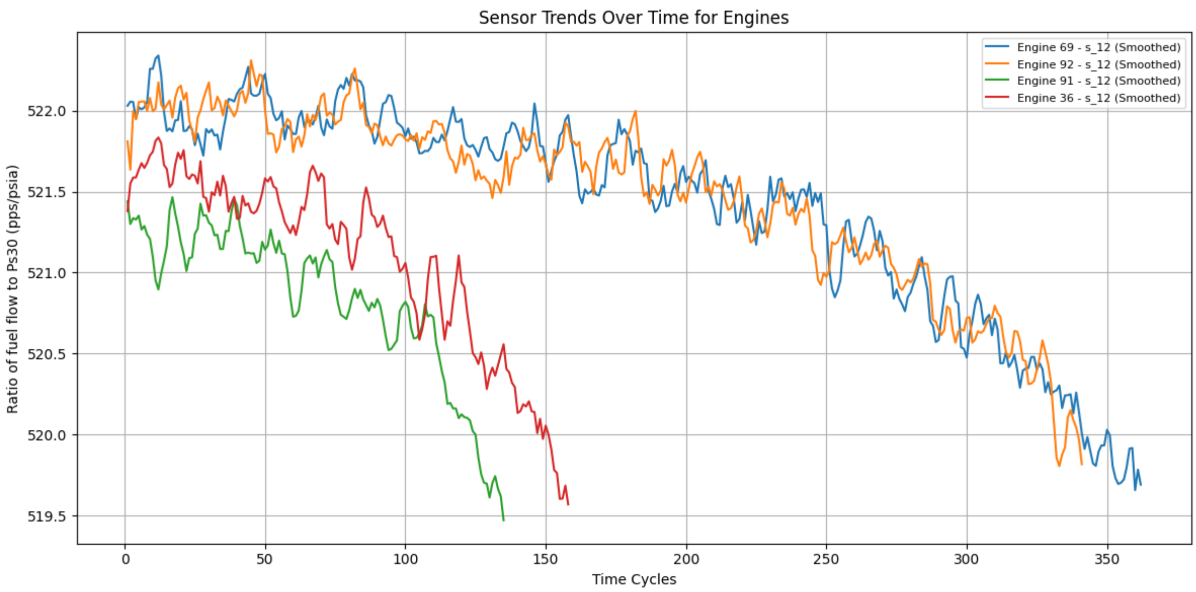

Figure 13: Sensor Trends Over Time for Engines (sensor s_11)

On the contrary, engines with reduced lifespans, including Engine 91 and Engine 36, exhibit sharper decreases in the fuel-to-pressure ratio, along with significant variations.

These irregular tendencies manifest early in the lifespan, signifying ineffective fuel use and indications of stress or malfunction in the engine's performance. The significant and fast decline in sensor data indicates possible problems, such inadequate maintenance or operating strain, that accelerate engine deterioration. The substantial disparities between long-lived and short-lived engines underscore the significance of sensor s_12 in detecting operational inefficiencies and facilitating Remaining Useful Life (RUL) prediction.

The studies of Figures 12 and 13 underscore the significance of temporal patterns in comprehending engine performance. The consistent and reliable performance of sensors in long-lived engines starkly contrasts with the unpredictable patterns seen in short-lived engines, offering essential information regarding engine health and deterioration. These patterns highlight the significance of using highly correlated indicators, such as HPC Outlet Static Pressure and Ratio of Fuel Flow to Ps30, for successful predictive modelling. By integrating these insights, the GRU-LSTM hybrid model is enhanced in its ability to identify significant patterns and produce precise RUL predictions.

9.2 Model Performance

The GRU-LSTM hybrid model's performance was meticulously assessed using novel test data to evaluate its prediction accuracy for Remaining Useful Life (RUL) estimate. The assessment utilised many measures to deliver a thorough comprehension of the model's precision, error distribution, and generalisation capacity.

Table 3: Hybrid Model Performance

METRIC	VALUE
Validation RMSE	24.62
Validation MAE	18.22
Test RMSE	35.21
Test R^2	0.729

The model's performance was evaluated on both validation and test datasets to ensure robustness and generalizability. The key results are as follows:

- Validation RMSE: The model achieved a validation RMSE of 24.62, indicating a moderate level of prediction error during training on unseen validation data. This value demonstrates the model's ability to generalize reasonably well within the training phase.
- Validation MAE: A validation MAE of 18.22 was observed, reflecting the average absolute error in predictions during training. The smaller MAE compared to RMSE suggests that the majority of prediction errors are within a reasonable range, with fewer extreme deviations.
- Test RMSE: On the test data, the model achieved an RMSE of 35.21. While this value is higher than the validation RMSE, it is expected, as the test data represents entirely unseen scenarios. The RMSE value indicates the model's average error when deployed in real-world conditions, reflecting its ability to generalize to new engines.
- Test R^2: The model achieved an R^2 value of 0.729 on the test set, indicating that approximately 73% of the variance in the actual RUL values was explained by the model. This is a strong indicator of the model's ability to capture the underlying patterns in the data and make accurate predictions. However, the remaining 27% unexplained variance highlights areas for potential improvement in feature selection, model architecture, or data preprocessing.

9.3 Comparison of Predicted and Actual RUL Values

A visual comparison of the predicted and actual RUL values for the engines in the test dataset was performed to better assess the model's performance. Figure 14 depicts the anticipated and actual Remaining Useful Life (RUL) values for 100 test units. The blue dashed line denotes the expected Remaining Useful Life (RUL) values, whereas the orange solid line signifies the actual RUL values.

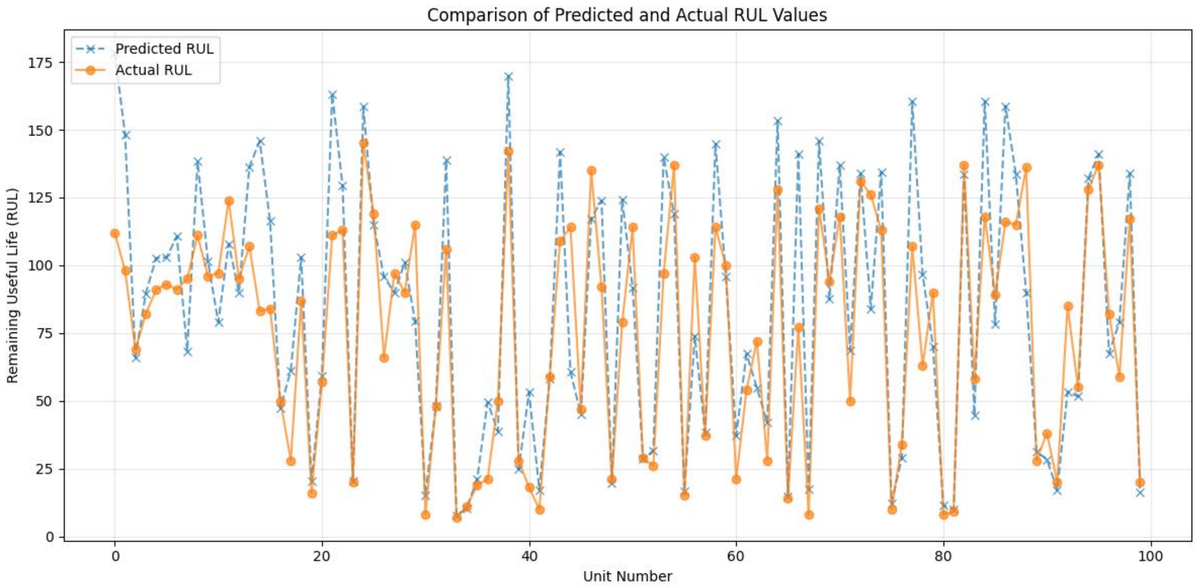

Figure 14: Comparison of Predicted v/s Actual RUL Values

This plot offers essential insights on the model's capacity to estimate the actual RUL values. For several units, the anticipated values nearly align with the actual values, demonstrating the model's efficacy in capturing the fundamental trends in the data. Nonetheless, certain cases demonstrate considerable discrepancies, especially for engines with severe Remaining Useful Life values. These discrepancies signify regions where the model's forecast precision may be improved.

The correlation between the anticipated and real RUL values illustrates the ability of the GRU-LSTM hybrid model in assimilating intricate temporal patterns. The inconsistencies in specific instances, however, highlight the difficulties presented by data noise, outliers, or operating factors not accounted for during training.

9.4 Interpretation of Results

The findings highlight the efficacy of the GRU-LSTM hybrid model in forecasting Remaining Useful Life (RUL) with a notable equilibrium of precision and generalisability. The comparatively low RMSE and MAE values on the validation set demonstrate the model's capacity to discern significant patterns and relationships within the data, enabling it to provide precise predictions on previously unobserved validation data during training. These measures underscore the model's resilience in generalising throughout the training phase.

The elevated RMSE noted in the test dataset aligns with predictive modelling assumptions, as test data frequently encompasses scenarios, patterns, or variations not in the training dataset. This rise illustrates the model's flexibility by reflecting the difficulties of projecting to completely unknown data while staying within a reasonable range.

The strong R^2 value of 0.729 further validates the model's effectiveness, as it explains approximately 73% of the variance in the actual RUL values. This result highlights the model's capacity to capture the complex temporal dependencies, and nonlinear relationships present in the sensor data, which are critical for accurate RUL predictions.

However, the unexplained variance (27%) suggests areas for improvement. These could include:

- Noise or inconsistencies in the input data, which may obscure underlying patterns.
- Insufficient or suboptimal feature selection, potentially excluding variables that could enhance predictive accuracy.
- Limitations in the current model architecture, such as the need for additional layers or attention mechanisms to capture more intricate relationships.

Despite these limitations, the model demonstrates strong foundational capabilities, providing actionable insights into engine health and RUL. This performance suggests that the GRU-LSTM hybrid approach is a viable solution for predictive maintenance tasks in similar domains, with opportunities for refinement and enhancement in future iterations.

9.5 Comparative Analysis of Baseline and Hybrid Models

To evaluate the improvements achieved by the GRU-LSTM hybrid model, a baseline model was implemented using a single GRU layer, ReLU activation function, MinMaxScaler for normalization, and no attention mechanism. The baseline model's performance on the validation and test datasets is summarized in Table 4.

Table 4: Comparative Model Performance

METRIC	Baseline Model	Hybrid Model
Validation RMSE	31.34	24.62
Validation MAE	22.87	18.22
Test RMSE	42.41	35.21
Test R^2	0.607	0.729

A comparison of the baseline and GRU-LSTM hybrid models demonstrates the impact of the enhanced architecture and techniques used in the hybrid model:

Validation Performance: The GRU-LSTM hybrid model achieved a lower Validation RMSE (24.62 vs. 31.35) and Validation MAE (18.22 vs. 22.88), indicating better generalization during training.

Test Performance: On the test set, the hybrid model achieved a Test RMSE of 35.21 compared to the baseline's 42.41, and a higher Test R^2 of 0.729 compared to 0.607. These results show the hybrid model's improved ability to predict RUL accurately and generalize to unseen data.

The comparative analysis underscores the significant improvements introduced by the GRU-LSTM hybrid model, demonstrating its effectiveness in capturing complex temporal dependencies and improving predictive performance over the baseline.

10. Project Management

Efficient project management is crucial for the effective execution of any research endeavour, harmonising time, resources, and quality to attain specified goals. This section describes the project's management, covering scheduling, quality assurance, risk management, and professional concerns. Particular emphasis is placed on modifications implemented during the project to tackle unexpected obstacles, illustrating the iterative and adaptable character of the research process.

10.1 Project Schedule

The original project timeline was structured around a clear work breakdown, with deadlines defined to guide progress.

The initial timeline was as follows:

- Week 1: Literature review of Temporal Convolutional Networks (TCN), attention mechanisms, and Remaining Useful Life (RUL) prediction models.
- Week 2: Data preparation, including preprocessing the C-MAPSS dataset, normalization, and feature extraction.
- Week 3: Development of a baseline TCN model with extensions incorporating self-attention and multi-head attention mechanisms.
- Week 4: Model training and validation using metrics such as RMSE, MAE, and R^2.
- Week 5: Comparative analysis of different attention mechanisms in TCN.
- Week 6: Results and visualization
- Week 7: Final report writing, focusing on methodology, results, and conclusions.

However, computational constraints encountered during the development of the TCN model necessitated a shift in focus. Training the TCN model proved resource-intensive and time-consuming, requiring the exploration of alternative solutions. After discussions with the project supervisor, the plan was revised to focus on a GRU-LSTM hybrid model, which aligned better with available resources while still addressing the research objectives.

The revised timeline reflected these changes:

- Week 3-5: Development, training, and validation of the GRU-LSTM hybrid model, integrating scaled dot-product attention mechanisms for improved predictions.
- Final Report Writing: Report writing in parallel with implementation, as advised by the project supervisor. This approach allowed reflections and insights from implementation to directly inform the report, streamlining the final write-up process.

A detailed Gantt chart illustrating the project timeline is provided in Appendix A for reference.

10.2 Risk Management

Risk management played a crucial role in mitigating potential disruptions to the project. A risk register was maintained to identify and address issues proactively. The following table summarizes the major risks, their impacts, and the mitigation strategies:

Table 5: Identified Risk and Mitigation Plan

Sl No.	Identified Risk	Impact	Risk Severity	Mitigation Plan	Materialized
1	Model Adaptation Challenges	Increased workload due to shift from TCN to GRU-LSTM.	High	Conducted literature review and incremental changes	Yes
2	Health issues due to unforeseen circumstances	Delay in project execution and submission	High	Request an extension if health issues arise	No
3	Time Management Challenges	Difficulty balancing coding and report writing.	Medium	Adjusted timeline and prioritized high-impact tasks.	Yes

10.3 Quality Management

Quality was maintained throughout the project by adopting modular design principles and seeking regular feedback. Each functional block (e.g., data preprocessing, model design, and evaluation) was implemented as an independent module, ensuring scalability and ease of debugging.

Code quality was monitored through adherence to established design practices, and results were consistently validated against multiple metrics to ensure robustness. Supervisor feedback was sought at critical stages, aligning the project outcomes with academic and research standards.

Timelines and deliverables were reviewed weekly to maintain alignment with project goals. Adjustments were made dynamically, including the reallocation of time to address unexpected challenges, such as the shift from TCN to GRU-LSTM models.

10.4 Social, Legal, Ethical, and Professional Considerations

During the study, ethical, legal, social, and professional aspects were meticulously addressed to guarantee responsible research procedures and compliance with institutional norms.

The dataset utilised in this project was obtained from the publicly accessible C-MAPSS repository, specifically designed for academic and research applications. The dataset lacks personal or sensitive information, eliminating privacy issues and dangers of exploitation. Moreover, all materials, including literature, datasets, and procedures, were accurately mentioned and attributed, according to academic integrity norms and preventing plagiarism.

Ethical issues were emphasised during the implementation and assessment stages, ensuring that the procedures employed adhered to the ideals of transparency and repeatability. The initiative adhered to the stipulations detailed in the ethical approval application, assuring conformity with the university's norms.

From a professional standpoint, consistent input from the supervisor was integrated to enhance the project's design and implementation. Only licensed software tools, including Python libraries and machine learning frameworks (TensorFlow, Keras, and

Scikit-learn), were employed to ensure adherence to legal obligations. Furthermore, meticulous attention was devoted to the thorough evaluation of the models, so minimising bias and assuring equitable forecasts.

This section underscores the significance of complying with ethical AI standards, encompassing appropriate data utilisation, transparent model assessment, and upholding academic and professional integrity throughout the research process.

11 Conclusion

To summarize the research project findings, the following key points are noted:

- The GRU-LSTM hybrid model, enhanced with an attention mechanism, successfully predicted the Remaining Useful Life (RUL) of turbofan engines, leveraging the strengths of GRU and LSTM layers to capture complex temporal dependencies.
- Feature selection and scaling using the Robust Scaler improved the model's performance by minimizing noise and reducing computational complexity without compromising accuracy.
- Exploratory Data Analysis provided valuable insights into sensor trends and operational anomalies, validating the preprocessing steps and reinforcing the foundation for effective model training.
- The integration of scaled dot-product attention and the Swish activation function enhanced the model’s ability to focus on relevant time steps and improve prediction performance.

11.1 Achievements

The project effectively addressed the primary research objectives, showcasing the capabilities of the GRU-LSTM hybrid model and highlighting the importance of comprehensive data analysis and preprocessing. Key achievements include:

- **Precise Remaining Useful Life Prediction Employing a GRU-LSTM Hybrid Model:** The goal, "Can a GRU-LSTM hybrid model effectively predict RUL for turbofan engines?" was successfully accomplished. The hybrid model effectively predicted RUL by utilising the complementing advantages of GRU and LSTM layers. This design adeptly captures intricate temporal correlations and patterns in the sequential data, resulting in strong predictions.
- **Impact of Scaling, Activation Functions, and Attention Mechanisms on Model Accuracy and Interpretability:** The second objective was thoroughly explored, demonstrating that RobustScaler effectively handled outliers and improved performance over MinMaxScaler. Swish activation enabled smoother gradient propagation compared to ReLU, and the scaled dot-product attention

mechanism improved interpretability by dynamically focusing on critical time steps. Together, these advancements reduced errors and emphasized the value of refined techniques in model design.

- **Enhanced Comprehension by Exploratory Data Analysis:** A significant achievement was the effective application of exploratory data analysis (EDA). The EDA process validated preprocessing steps and revealed critical patterns in sensor data, such as temporal behaviours of sensors highly correlated with RUL. The analysis also identified operational anomalies and differentiated sensor trends across engines with varying lifespans, providing valuable insights into engine health and strengthening the foundation for effective model training.

These achievements collectively highlight the project’s success in addressing its objectives and underscore the importance of robust preprocessing, feature selection, and model design in predictive maintenance for complex systems like turbofan engines.

11.2 Limitations and Future Work

While the research successfully met its objectives, several limitations were identified, providing avenues for future exploration:

1. **Computational Constraints:**
   The shift from the initially planned Temporal Convolutional Network (TCN) model to the GRU-LSTM hybrid model was driven by computational limitations. Although the GRU-LSTM hybrid architecture demonstrated strong performance, revisiting TCNs with access to enhanced computational resources could reveal additional insights and further improve prediction accuracy.
2. **Impact of Outliers:**
   The dataset contained outliers in certain sensors, which influenced the model's predictive accuracy. Robust outlier detection and handling techniques, such as advanced anomaly detection algorithms or preprocessing strategies, could mitigate these effects and lead to more reliable predictions in future work.
3. **Model Accuracy:**

While the GRU-LSTM model achieved promising results, the test RMSE of 35.21 indicates room for improvement. Incorporating additional features, optimizing hyperparameters, or adopting advanced techniques such as transfer learning, or automated machine learning (AutoML) could enhance the model’s performance.

4. **Input and Model Enhancements:**
   Future research could explore the use of multi-step sequence input data to capture long-term dependencies more effectively. Additionally, investigating more advanced attention mechanisms, such as multi-head attention, or employing ensemble learning approaches could further boost prediction accuracy and model robustness.
5. **Generalizability:**
   The current model was tailored to the CMAPSS dataset, which focuses on turbofan engines. Expanding the research to include diverse datasets or other predictive maintenance scenarios could validate the model's generalizability and adaptability across different domains.

By addressing these limitations, future research can build upon the findings of this study to further refine predictive maintenance models, enhance their accuracy, and expand their applicability to broader contexts.

12. Student Reflection

Initiating this study effort has been an illuminating and transformational experience, both intellectually and personally. Initially, my major objective was to create a comprehensive predictive maintenance model utilising sophisticated machine learning methodologies. Throughout my journey, I had several problems and chances that significantly influenced my comprehension of data science, deep learning, and research methodology.

A pivotal moment was the transition from the Temporal Convolutional Network (TCN) model to the GRU-LSTM hybrid architecture. This transition, prompted by computing limitations, was initially discouraging, as the TCN model had been the focal point of my original strategy. This experience imparted a crucial lesson in flexibility and problem-solving. The GRU-LSTM hybrid model fulfilled my expectations, demonstrating its effectiveness in collecting temporal patterns and providing precise predictions. The event highlighted the significance of adaptability and the capacity to change direction efficiently when confronted with unexpected challenges.

The cyclical process of data preparation, model development, and assessment provided a greater understanding of the intricacies involved in machine learning endeavours. Exploratory data analysis specifically uncovered significant insights into sensor patterns and their correlation with Remaining Useful Life (RUL). This phase underscored the need of rigorous data management and emphasised the necessity of extracting relevant insights from raw data prior to initiating model training.

The incorporation of an attention mechanism into the model design proved to be a demanding but gratifying undertaking. The implementation and integration of this feature enhanced the model's interpretability and enabled an exploration of the complexities of neural network construction. Through this approach, I acquired a deeper understanding of the operation of attention processes and their capacity to improve deep learning models.

Time management was another essential element of this endeavour. Managing the requirements of data preparation, model experimentation, and report composition necessitated a methodical approach. Consistently revising my project timetable, soliciting feedback from my supervisor, and realigning priorities guaranteed that I remained on course, despite certain portions of the project necessitating extra time and work.

This project has considerably improved my technical abilities, especially in Python programming, deep learning frameworks such as TensorFlow and Keras, and data visualisation tools. Moreover, it has cultivated tenacity and critical thinking, attributes that will certainly aid me in future pursuits.

Finally, contemplating the study's limits and pinpointing possibilities for enhancement have yielded significant insights into the iterative essence of research. The method has demonstrated that each barrier and constraint presents a chance for development and education. It has underscored the necessity of upholding a stringent, ethical, and professional demeanour throughout the study process.

This research project has been fundamental to my academic growth, providing me with the tools and confidence to address intricate difficulties in machine learning and predictive analytics. This path has been one of discovery, resilience, and growth, for which I am profoundly thankful for the information and experiences it has provided.

13. References

A Prognostics and Health Management Roadmap for Information and Electronics Rich Systems. (n.d.).

Ahmad, W., Khan, S. A., Islam, M. M. M., & Kim, J. M. (2019). A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. *Reliability Engineering and System Safety*, *184*, 67–76. https://doi.org/10.1016/j.ress.2018.02.003

Ahn, G., Yun, H., Hur, S., & Lim, S. Y. (2021). A time-series data generation method to predict remaining useful life. *Processes*, *9*(7). https://doi.org/10.3390/pr9071115

Baptista, M., Henriques, E. M. P., de Medeiros, I. P. P., Malere, J. P. P., Nascimento, C. L. L., & Prendinger, H. (2019). Remaining useful life estimation in aeronautics: Combining data-driven and Kalman filtering. *Reliability Engineering and System Safety*, *184*, 228–239. https://doi.org/10.1016/j.ress.2018.01.017

Cao, Y., Jia, M., Ding, P., & Ding, Y. (2021). Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. *Measurement: Journal of the International Measurement Confederation*, *178*. https://doi.org/10.1016/j.measurement.2021.109287

Chen, G., Chen, J., Zi, Y., & Miao, H. (2017). Hyper-parameter optimization based nonlinear multistate deterioration modeling for deterioration level assessment and remaining useful life prognostics. *Reliability Engineering and System Safety*, *167*, 517–526. https://doi.org/10.1016/j.ress.2017.06.030

Chen, J., Jing, H., Chang, Y., & Liu, Q. (2019). Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliability Engineering and System Safety*, *185*, 372–382. https://doi.org/10.1016/j.ress.2019.01.006

Chen, Z., Li, Y., Xia, T., & Pan, E. (2019). Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy. *Reliability Engineering and System Safety*, *184*, 123–136. https://doi.org/10.1016/j.ress.2017.09.002

Collobert, R., & Weston, J. (n.d.). *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning*. http://wordnet.princeton.edu

Dagar, A. (n.d.-a). *Gated Recurrent Units (GRU)*. Dive into Deep Learning. Retrieved December 12, 2024, from https://d2l.ai/chapter_recurrent-modern/gru.html

Dagar, A. (n.d.-b). *Long Short-Term Memory (LSTM)*. Dive into Deep Learning. Retrieved December 12, 2024, from https://d2l.ai/chapter_recurrent-modern/lstm.html

Dagar, A. (n.d.-c). *Recurrent Neural Networks*. Dive into Deep Learning. Retrieved December 12, 2024, from https://d2l.ai/chapter_recurrent-neural-networks/rnn.html

Du, Y., Pei, B., Zhao, X., & Ji, J. (2020). Deep scaled dot-product attention based domain adaptation model for biomedical question answering. *Methods*, *173*, 69–74. https://doi.org/10.1016/j.ymeth.2019.06.024

Duong, P. L. T., & Raghavan, N. (2018). Heuristic Kalman optimized particle filter for remaining useful life prediction of lithium-ion battery. *Microelectronics Reliability*, *81*, 232–243. https://doi.org/10.1016/j.microrel.2017.12.028

Fan, J., Yung, K. C., & Pecht, M. (2011). Physics-of-failure-based prognostics and health management for high-power white light-emitting diode lighting. *IEEE Transactions on Device and Materials Reliability*, *11*(3), 407–416. https://doi.org/10.1109/TDMR.2011.2157695

Girsang, A. S., & Stanley. (2023). Hybrid LSTM and GRU for Cryptocurrency Price Forecasting Based on Social Network Sentiment Analysis Using FinBERT. *IEEE Access*, *11*, 120530–120540. https://doi.org/10.1109/ACCESS.2023.3324535

Guo, L., Li, N., Jia, F., Lei, Y., & Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, *240*, 98–109. https://doi.org/10.1016/j.neucom.2017.02.045

Hsu, C.-S., & Jiang, J.-R. (2018). *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE.

Hu, Y., Li, H., Shi, P., Chai, Z., Wang, K., Xie, X., & Chen, Z. (2018). A prediction method for the real-time remaining useful life of wind turbine bearings based on the Wiener process. *Renewable Energy*, *127*, 452–460. https://doi.org/10.1016/j.renene.2018.04.033

Huang, Z., Xu, Z., Ke, X., Wang, W., & Sun, Y. (2017). Remaining useful life prediction for an adaptive skew-Wiener process model. *Mechanical Systems and Signal Processing*, *87*, 294–306. https://doi.org/10.1016/j.ymssp.2016.10.027

Jafari, S., & Byun, Y. C. (2023). A CNN-GRU Approach to the Accurate Prediction of Batteries' Remaining Useful Life from Charging Profiles. *Computers*, *12*(11). https://doi.org/10.3390/computers12110219

Khoirunnisa, A., & Ramadhan, N. G. (2023). Improving malaria prediction with ensemble learning and robust scaler: An integrated approach for enhanced accuracy. *JURNAL INFOTEL*, *15*(4), 326–334. https://doi.org/10.20895/infotel.v15i4.1056

Lau, D., & Fong, B. (2011). Special issue on prognostics and health management. In *Microelectronics Reliability* (Vol. 51, Issue 2, pp. 253–254). https://doi.org/10.1016/j.microrel.2010.10.004

Le Son, K., Fouladirad, M., & Barros, A. (2016). Remaining useful lifetime estimation and noisy gamma deterioration process. *Reliability Engineering and System Safety*, *149*, 76–87. https://doi.org/10.1016/j.ress.2015.12.016

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. https://doi.org/10.1038/nature14539

Li, T., Si, X., Pei, H., & Sun, L. (2022). Data-model interactive prognosis for multi-sensor monitored stochastic degrading devices. *Mechanical Systems and Signal Processing*, *167*. https://doi.org/10.1016/j.ymssp.2021.108526

Liao, Y., Zhang, L., & Liu, C. (2018, August 27). Uncertainty Prediction of Remaining Useful Life Using Long Short-Term Memory Network Based on Bootstrap Method. *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*. https://doi.org/10.1109/ICPHM.2018.8448804

Ling, M. H., Ng, H. K. T., & Tsui, K. L. (2019). Bayesian and likelihood inferences on remaining useful life in two-phase degradation models under gamma process. *Reliability Engineering and System Safety*, *184*, 77–85. https://doi.org/10.1016/j.ress.2017.11.017

Liu, Q., Dong, M., Lv, W., Geng, X., & Li, Y. (2015). A novel method using adaptive hidden semi-Markov model for multi-sensor monitoring equipment health prognosis. *Mechanical Systems and Signal Processing*, *64–65*, 217–232. https://doi.org/10.1016/j.ymssp.2015.03.029

Liu, Y., Zhao, G., & Peng, X. (2019). Deep learning prognostics for lithium-ion battery based on ensembled long short-term memory networks. *IEEE Access*, *7*, 155130–155142. https://doi.org/10.1109/ACCESS.2019.2937798

Mohd Saufi, M. S. R., & Hassan, K. A. (2021). Remaining useful life prediction using an integrated Laplacian-LSTM network on machinery components. *Applied Soft Computing*, *112*. https://doi.org/10.1016/j.asoc.2021.107817

Ragb, H., Muntaser, A., Jera, E., Saide, A., & Elwarfalli, I. (n.d.). *Hybrid GRU-LSTM Recurrent Neural Network-Based Model for Real Estate Price Prediction*.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for Activation Functions*. http://arxiv.org/abs/1710.05941

Reddy, K. V. A., Ambati, S. R., Rithik Reddy, Y. S., & Reddy, A. N. (2021, October 9). AdaBoost for Parkinson's Disease Detection using Robust Scaler and SFS from Acoustic Features. *Proceedings - 1st International Conference on Smart Technologies Communication and Robotics, STCR 2021*. https://doi.org/10.1109/STCR51658.2021.9588906

Ren, L., Dong, J., Wang, X., Meng, Z., Zhao, L., & Deen, M. J. (2021). A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life. *IEEE Transactions on Industrial Informatics*, *17*(5), 3478–3487. https://doi.org/10.1109/TII.2020.3008223

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). *2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation*.

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation - A review on the statistical data driven approaches. In *European Journal of Operational Research* (Vol. 213, Issue 1, pp. 1–14). Elsevier B.V. https://doi.org/10.1016/j.ejor.2010.11.018

Son, J., Zhou, S., Sankavaram, C., Du, X., & Zhang, Y. (2016). Remaining useful life prediction based on noisy condition monitoring signals using constrained Kalman filter. *Reliability Engineering and System Safety*, *152*, 38–50. https://doi.org/10.1016/j.ress.2016.02.006

Song, Y., Li, L., Peng, Y., & Liu, D. (n.d.). *Lithium-ion battery remaining useful life prediction based on GRU-RNN*.

Sun, S., Wang, J., Xiao, Y., Peng, J., & Zhou, X. (2024). Few-shot RUL prediction for engines based on CNN-GRU model. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-66377-3

Sun, W., Shao, S., Zhao, R., Yan, R., Zhang, X., & Chen, X. (2016). A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement: Journal of the International Measurement Confederation*, *89*, 171–178. https://doi.org/10.1016/j.measurement.2016.04.007

Tian, Tianhai. (2016). *Proceedings, 2016 IEEE International Conference on Bioinformatics and Biomedicine : Dec 15-18, 2016, Shenzhen, China*. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. In *Computational Intelligence and Neuroscience* (Vol. 2018). Hindawi Limited. https://doi.org/10.1155/2018/7068349

Wang, C., Jiang, W., Yang, X., & Zhang, S. (2021). Rul prediction of rolling bearings based on a dcae and cnn. *Applied Sciences (Switzerland)*, *11*(23). https://doi.org/10.3390/app112311516

Wang, D., Tsui, K. L., & Miao, Q. (2017). Prognostics and Health Management: A Review of Vibration Based Bearing and Gear Health Indicators. *IEEE Access*, *6*, 665–676. https://doi.org/10.1109/ACCESS.2017.2774261

Wang, J., Wen, G., Yang, S., & Liu, Y. (2019). Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network. *Proceedings - 2018 Prognostics and System Health Management Conference, PHM-Chongqing 2018*, 1037–1042. https://doi.org/10.1109/PHM-Chongqing.2018.00184

Wang, S., Chen, J., Wang, H., & Zhang, D. (2019). Degradation evaluation of slewing bearing using HMM and improved GRU. *Measurement: Journal of the International Measurement Confederation*, *146*, 385–395. https://doi.org/10.1016/j.measurement.2019.06.038

Wang, X., Li, Y., Xu, Y., Liu, X., Zheng, T., & Zheng, B. (2023). Remaining Useful Life Prediction for Aero-Engines Using a Time-Enhanced Multi-Head Self-Attention Model. *Aerospace*, *10*(1). https://doi.org/10.3390/aerospace10010080

Wu, J., Hu, K., Cheng, Y., Zhu, H., Shao, X., & Wang, Y. (2020). Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. *ISA Transactions*, *97*, 241–250. https://doi.org/10.1016/j.isatra.2019.07.004

Ye, R., & Dai, Q. (2018). A novel transfer learning framework for time series forecasting. *Knowledge-Based Systems*, *156*, 74–99. https://doi.org/10.1016/j.knosys.2018.05.021

Yu, W., Kim, I. Y., & Mechefske, C. (2019). Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing*, *129*, 764–780. https://doi.org/10.1016/j.ymssp.2019.05.005

Yuan, M., Wu, Y., & Lin, L. (2016). *IEEE/CSAA AUS 2016 : 2016 IEEE/CSAA International Conference on Aircraft Utility Systems : October 10-12, 2016, Beijing, China*. IEEE.

Zhang, Z., Si, X., Hu, C., & Lei, Y. (2018). Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods. In *European Journal of Operational Research* (Vol. 271, Issue 3, pp. 775–796). Elsevier B.V. https://doi.org/10.1016/j.ejor.2018.02.033

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. In *Mechanical Systems and Signal Processing* (Vol. 115, pp. 213–237). Academic Press. https://doi.org/10.1016/j.ymssp.2018.05.050

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE.

Zhou, J., Qin, Y., Chen, D., Liu, F., & Qian, Q. (2022). Remaining useful life prediction of bearings by a new reinforced memory GRU network. *Advanced Engineering Informatics*, *53*. https://doi.org/10.1016/j.aei.2022.101682

APPENDIX A: Gantt Chart (Timeline)

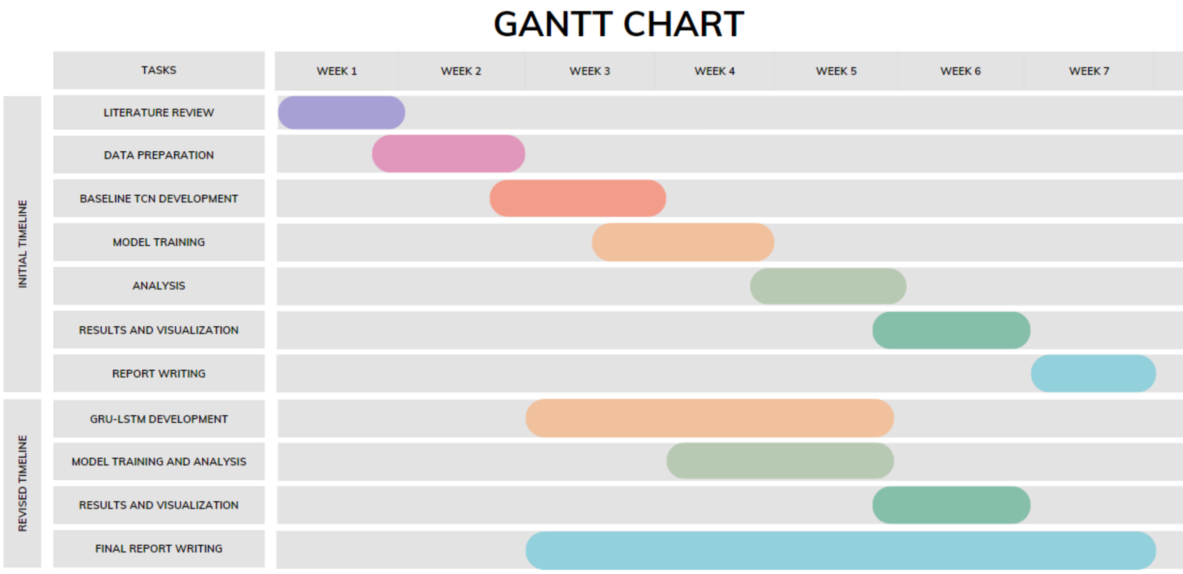

Figure 15: Gantt Chart (Project Timeline)

APPENDIX B: Certificate of Ethical Approval