ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2025.2.001

Remaining Useful Life Prediction for Turbofan Engines Based on Deep Learning Insights from the CMAPSS Dataset

Samiha M. Elsherif, Bassel Hafiz, M. A. Makhlouf, Osama Farouk

Information Systems Department, Faculty of Computers and Informatics, Suez Canal University, Ismailia, 41522, Egypt

Samiha_ahmed@ci.suez.edu.eg, Samiha.ahmed910@gmail.com

Abstract

Predicting the Remaining Useful Life (RUL) of turbofan engines is a vital component in the field of prognostics and health management (PHM). Accurate estimation of RUL enables timely maintenance and prevents sudden failures, which is essential for ensuring safety and reducing operational costs. This study introduces a novel hybrid deep learning model referred to as RCBLA (Residual Convolutional Bidirectional Long Short-Term Memory with Attention), designed specifically to handle the temporal and sequential characteristics of engine sensor data. The model architecture combines residual convolutional layers to enhance feature extraction, Temporal Convolutional Networks (TCN) for capturing long-range dependencies, Bidirectional LSTM layers for understanding both past and future contexts, and an attention mechanism to focus on the most critical time steps. Layer normalization is applied to improve convergence and training stability. The proposed model is evaluated using the FD001 and FD003 subsets of the C-MAPSS dataset. A piecewise linear degradation model is adopted, and only sensor measurements are used as input features. A custom learning rate scheduler is applied to optimize model convergence, and a dropout rate of 0.7 and 0.5 is used to prevent overfitting. The results demonstrate the model's strong predictive capabilities, achieving RMSE values of 14.389 and 12.46 for FD001 and FD003, respectively. The scoring function yields 279.23 and 282.56 for FD001 and FD003, respectively. These outcomes highlight the model's robustness and suitability for real-world RUL prediction, offering a reliable framework for improving PHM systems in aerospace applications.

Keywords: *Remaining Useful Lifetime, Predictive Maintenance, Residual Convolutional Network, BiLSTM, Temporal Convolutional Network (TCN), Attention Mechanism.*

Received: 10 September 2025
Accepted: 30 December 2025
Online: 30 December 2025
Published: 30 December 2025

1 Introduction

The maintenance of large-scale systems such as aircraft presents a substantial challenge due to its significant impact on operational expenses. For instance, aircraft maintenance can consume over 10% of an airline's total operating costs, with average annual expenditures reaching several million dollars per aircraft [21]. Consequently, minimizing maintenance costs has become a strategic priority in the aerospace and industrial sectors. The rise of Internet of Things (IoT) technologies and the increasing availability of sensor data have paved the way for more intelligent and cost-effective maintenance strategies[27].

Predictive maintenance, driven by real-time monitoring and data-driven insights, seeks to anticipate equipment failure before it occurs. This strategy hinges on accurately estimating the Remaining Useful Life (RUL) of critical components, which enables maintenance actions to be scheduled at the optimal time. Effective RUL prediction not only reduces unplanned downtimes but also improves the overall reliability and safety of complex systems. Despite the abundance of sensor information, estimating RUL remains a complex problem due to the dynamic and uncertain conditions under which industrial equipment operates [30].

In parallel, advances in artificial intelligence (AI) have significantly transformed industrial diagnostics and decision-making processes. One prominent area of progress is in Prognostics and Health Management (PHM), which integrates condition monitoring and predictive techniques to assess the future performance of systems [18]. PHM has become increasingly vital in domains where failures are costly or potentially catastrophic, such as aerospace and energy. By leveraging historical operational data and degradation patterns, PHM enhances system dependability and enables proactive maintenance planning, reducing both risk and cost [2].

In this study, we address the critical task of predicting the Remaining Useful Life (RUL) of industrial systems, a fundamental aspect of Prognostics and Health Management (PHM) [25]. Accurate RUL estimation plays a significant role in extending the lifecycle of mechanical components, promoting sustainable reuse, reducing operational energy consumption, and mitigating environmental impact. However, conventional prognostic techniques, particularly model-based approaches, often fall short when dealing with complex, high-dimensional, and nonlinear time-series data.

Current RUL estimation strategies for turbofan engines typically fall into two main categories: physics-based modeling and data-driven techniques [11]. Physics-based methods require a comprehensive understanding of system dynamics and detailed operational parameters to construct accurate analytical models. However, due to the highly intricate internal structure and the tightly integrated components of turbofan engines, it is rarely feasible to derive precise physical models without extensive domain knowledge. As a result, data-driven methods have gained prominence, relying on historical sensor measurements to learn degradation patterns and predict system failure without needing explicit physical modeling [23].

With the advent of deep learning, significant progress has been made in analyzing sequential and nonlinear data. Deep learning architectures [10], characterized by their multilayer hierarchical structures, enable automatic extraction of latent features, thereby enhancing prediction accuracy with minimal human intervention. Since its emergence, deep learning has found widespread application across domains such as image analysis, speech processing, and fault detection. More recently, it has shown substantial promise in the RUL prediction of critical systems—including batteries, rotating machinery, and jet engines—through models like one-dimensional Convolutional Neural Networks (1D-CNNs), Long Short-Term Memory (LSTM) networks, deep transfer learning frameworks, and other hybrid architectures that enhance both temporal modeling and generalization capabilities [15] [28] [14].

To address the challenges of accurately modeling complex degradation patterns, this work introduces a hybrid deep learning architecture that combines residual convolutional encoding, temporal convolutional layers, and bidirectional LSTM units with a learnable attention mechanism. The model is specifically designed to extract both spatial and temporal dependencies from multivariate sensor signals. Initially, sensor sequences are passed through a Residual CNN encoder to capture local degradation features, followed by a temporal convolutional layer with causal and dilated structure to enhance the model's capacity for long-range temporal dependencies. A bidirectional LSTM layer is then employed to model sequence dynamics in both forward and backward directions, further stabilized by layer normalization. To emphasize informative time steps, a learnable attention mechanism is incorporated to adaptively weigh feature contributions before regression. The model concludes with fully connected layers to estimate the remaining useful life. This architecture is supported by effective preprocessing steps and leverages a piecewise degradation assumption, where the system's behavior is captured in discrete stages using sliding windows. The proposed model demonstrates strong potential for handling nonlinear degradation trajectories and achieving robust RUL prediction in complex operating environments. The model is trained using the Adam optimization algorithm with a carefully selected learning rate. The loss function is based on the Mean Squared Error (MSE), while Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are monitored throughout the training process to assess prediction accuracy. The model's hyperparameters were optimized through extensive experimentation. The results demonstrate that the proposed architecture significantly enhances the accuracy and reliability of remaining useful life estimation.

The rest of this paper is structured as follows: Section 2 provides a comprehensive review of related work. Section 3 outlines the proposed methodology in detail. Section 4 presents the experimental setup and results. Section 5 highlights the impact of removing each architectural component from the RCBLA model and referencing specific figures and metrics. Finally, the conclusions and potential future directions are discussed in Section 6.

2 Literature review

Deep learning has become a central approach in predictive maintenance, offering powerful capabilities for modeling complex and nonlinear degradation behaviors of turbofan engines. Unlike traditional statistical or machine learning methods, deep neural networks can automatically extract meaningful features from high-dimensional sensor data, capturing hidden temporal and spatial dependencies that directly influence Remaining Useful Life (RUL) estimation. This ability enables more accurate predictions of engine health status, supports early detection of faults, and enhances decision-making for maintenance scheduling. Consequently, deep learning provides a robust foundation for advancing predictive maintenance strategies in aviation applications.

The introduction of the Relevance Vector Machine (RVM) [4] provided a probabilistic framework for sparse regression and classification, addressing limitations of traditional kernel methods by producing compact models with fewer basis functions while maintaining predictive accuracy. This advancement highlighted the potential of Bayesian inference in enhancing generalization and interpretability. Complementing this, the development of LIBSVM offered a practical and efficient software implementation of Support Vector Machines (SVMs), enabling widespread application of SVM algorithms across various domains. Together, these contributions represent significant milestones in machine learning research: RVM established a foundation for probabilistic sparse learning, while LIBSVM facilitated the practical deployment and benchmarking of SVM-based methods, thereby accelerating their adoption in real-world predictive tasks [26].

The study in [22] investigated the use of a deep convolutional neural network (CNN) as a regression tool to estimate the Remaining Useful Life (RUL) of turbofan engines. Leveraging the CNN's capability to automatically extract hierarchical features from raw sensor data, the model circumvents the need for manual feature engineering and improves robustness under complex operational conditions. Experimental evaluations on the CMAPSS dataset reveal that the proposed CNN model consistently outperforms traditional predictive models—including statistical and classical regression approaches—by delivering higher accuracy and greater robustness against sensor noise and varying degradation scenarios. These results demonstrate the effectiveness of deep convolutional architectures in capturing meaningful degradation patterns for reliable RUL prediction. The effectiveness of the proposed model was assessed using Root Mean Square Error (RMSE) and a dedicated scoring function. For the FD001 dataset, the model achieved an RMSE of 18.45 with a corresponding S-score of 1299, while for the FD003 dataset, the RMSE reached 19.82 and the S-score was 1600.

The work in [35] explored the application of Long Short-Term Memory (LSTM) networks for the estimation of Remaining Useful Life (RUL) in complex systems. The model is specifically designed to capture sequential dependencies in sensor data, which are critical for reflecting progressive degradation patterns. By effectively learning long-range temporal correlations, the LSTM-based approach demonstrated superior prediction performance compared to conventional machine learning methods. The results confirmed its ability to generalize across varying operating conditions, providing accurate RUL forecasts and supporting its suitability for real-world prognostics and health management applications. The effectiveness of the proposed model was assessed using Root Mean Square Error (RMSE) and a dedicated scoring function. For the FD001 dataset, the model achieved an RMSE of 16.14 with a corresponding S-score of 338, while for the FD003 dataset, the RMSE reached 16.18 and the S-score was 852. Authors in [13] introduced a hybrid deep learning framework that integrates Convolutional Neural Networks (CNNs) with LSTM layers to improve the prediction of RUL. The CNN component enables efficient extraction of spatial and local features from multivariate sensor data, while the LSTM captures temporal dynamics across degradation sequences. The hybrid architecture showed significant improvements in prediction accuracy and stability over standalone models. Experimental results highlighted its robustness to noise and its effectiveness in dealing with variable degradation rates, making it a reliable approach for predictive maintenance in turbofan engines and similar industrial systems. The model was evaluated using RMSE and a scoring function, achieving a RMSE of 16.16 and a score of 303 for FD001.

Recent research has explored advanced deep learning architectures for enhancing the prediction of Remaining Useful Life (RUL) in complex systems. Study in [32] proposed a bidirectional recurrent neural network combined with an autoencoder framework, where the bidirectional structure captured both past and future temporal dependencies while the autoencoder extracted compact representations from high-dimensional sensor data. This design improved robustness against noise and yielded more stable RUL estimates. Complementarily, another study [6] introduced an attention-based deep learning model that adaptively emphasized the most relevant sensor signals and time steps for degradation progression. The integration of attention allowed the model to prioritize informative features and better capture long-term dependencies, leading to enhanced prediction accuracy under varying operating conditions. Together, these works demonstrate how integrating temporal modeling with feature compression and attention mechanisms can significantly improve the reliability and interpretability of RUL prognostics in practical applications.

Scholars in [16] proposed a deep convolutional neural network (CNN)-based framework for Remaining Useful Life (RUL) estimation in the context of prognostics. The study emphasized the ability of CNNs to automatically extract hierarchical features from raw multivariate sensor signals, reducing the need for extensive manual feature engineering. By applying convolutional layers with temporal filters, the model captured local degradation patterns and long-term dependencies that are essential for accurate RUL prediction.The approach was validated on the NASA C-MAPSS dataset, where it achieved significant improvements over traditional data-driven methods. The reported results showed Root Mean Square Error (RMSE) values of 12.61 on FD001 and 12.64 on FD003, with corresponding scoring function values of 273.70 and 284.10, respectively. These outcomes highlighted the model's capability to generalize across different operating conditions and fault modes, while maintaining strong predictive accuracy. Overall, the work demonstrated that CNN-based architectures can effectively handle complex sensor data and provide robust prognostic insights for predictive maintenance applications.

A variational encoding framework combined with a novel latent space regularization strategy was introduced in [7]. The model's effectiveness was evaluated using Root Mean Square Error (RMSE) and a scoring function. The achieved RMSE values were 15.81 for FD001 and 14.88 for FD003, while the corresponding S-scores were 326 and 722, respectively. In [8], a hybrid prediction framework combining CNN, LSTM, and Attention mechanisms was introduced. The CNN component was employed to extract local patterns from aero-engine sensor data, while the LSTM-Attention module captured temporal dependencies and emphasized critical features for forecasting engine life. Comparative analyses, including ablation studies, were performed against other RUL prediction approaches. The proposed model demonstrated strong predictive capability across four datasets, achieving RMSE values of 15.977, 14.452, 13.907, and 16.637.

Recent work in [3] has explored hybrid deep learning approaches, such as combining Convolutional Neural Networks with Gated Recurrent Units (CNN-GRU), to enhance feature extraction and capture temporal dependencies in sensor data. By applying preprocessing strategies like windowing and reshaping, these models have been shown to effectively identify degradation trends in the C-MAPSS dataset. Evaluations using RMSE and MSE metrics demonstrated that CNN-GRU architectures outperform conventional machine learning methods, highlighting their robustness and adaptability to different operational conditions. Such approaches underline the growing potential of hybrid deep learning models in advancing prognostics and health management systems. The reported results showed Root Mean Square Error (RMSE) values of 16.29 on FD001 and 23.7 on FD003. To address the limited spatial feature extraction capability in traditional graph-based approaches, a spatio-temporal self-attention network (STCAN) has been developed for RUL prediction of multi-sensor equipment [31]. The method combines graph convolution to capture spatial correlations with self-attention for global and local feature representation. In addition, dilated convolution is incorporated to enhance multi-scale temporal modeling and alleviate long-term dependency challenges. Experimental evaluations indicate that this framework provides improved accuracy compared to conventional techniques.

A CAELSTM hybrid model [9] combining convolutional autoencoders, LSTM, and attention mechanisms has been applied to RUL prediction using the CMAPSS FD001 and FD003 subsets. The framework effectively captures key temporal–spatial features and outperforms baseline approaches, achieving RMSE values of 14.44 and 13.40, MAE of 10.49 and 10.68, and scoring metrics of 282.38 and 264.47, respectively. These results demonstrate its potential for reliable prognostics and predictive maintenance in aerospace applications.

3 Proposed Methodology

The RCBLA (Residual Convolutional Bidirectional Long Short-Term Memory with Attention) model is designed to analyze time series data, particularly in the context of predicting the Remaining Useful Life (RUL) of machinery such as turbofan engines. This model integrates multiple deep learning architectures: Residual Convolutional Neural Networks (ResCNN), Temporal Convolutional Networks (TCN), Bidirectional Long Short-Term Memory (BiLSTM), Learnable Attention Mechanisms, and Layer Normalization to predict the Remaining Useful Life (RUL) of turbofan engines from sensor readings. The architecture begins with a ResCNN encoder to extract local degradation features while mitigating vanishing gradient issues through skip connections. The TCN layer further captures temporal dependencies across multiple receptive fields via dilated convolutions. The BiLSTM module then models sequential dependencies in both forward and backward directions, ensuring that contextual information from the entire input sequence is retained. Finally, the learnable attention mechanism adaptively focuses on the most relevant time steps for the prediction task, while the regression head generates the final RUL estimate. A custom learning rate scheduler is employed during training to optimize convergence. This scheduler dynamically adjusts the learning rate according to the training phase, enabling faster convergence in early epochs and more stable fine-tuning in later epochs, which helps avoid overfitting and local minima.

- **Residual CNN Encoder**

  The ResCNN encoder [5] [19] applies successive 1D convolutional layers with skip connections to capture localized patterns in the sensor data. The skip connection in the residual block ensures that the gradient can flow directly to earlier layers, alleviating the vanishing gradient problem and improving feature reuse. Let x be the input to the residual block, and $F(x; W)$ represent the transformation applied by two successive convolutional layers with learnable weights W. The mathematical formulation is represented as follows:

  $$y = \sigma\left(x + \mathcal{F}(x; W)\right) \tag{1}$$

  Where:

  - $\sigma(.)$: is the ReLU activation function.
  - $F(x; W)$=Conv1D(Conv1D(x))
  - $+$: is the skip connection.

- **Temporal Convolutional Network (TCN) Layer**

  The TCN [34] [17] uses dilated causal convolutions to model long-range dependencies without recurrent operations. The dilation rate d determines the spacing between convolutional filter elements, exponentially expanding the receptive field. The mathematical formulation is represented as follows:

  For an input sequence x and a filter f of size k

  $$y(t) = \sum_{i=0}^{k-1} f(i) \cdot x_{t-d\cdot i} \tag{2}$$

  Where:

  - d: is the dilation factor.
  - $y(t)$: is the output at time step t.

- **Bidirectional LSTM Layer**

  The BiLSTM [24] processes the input sequence in both forward and backward directions to capture dependencies from past and future time steps simultaneously. The mathematical formulation is represented as follows:

  For a given time step t

$$\overrightarrow{h_t} = \mathrm{LSTM}_f(x_t, \overrightarrow{h_{t-1}}) \tag{3}$$

$$\overleftarrow{h_t} = \mathrm{LSTM}_b(x_t, \overleftarrow{h_{t+1}}) \tag{4}$$

  The final hidden state is the concatenation

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{5}$$

- **Learnable Attention Mechanism**

  The learnable attention layer assigns different weights to each time step, allowing the model to emphasize the most relevant periods in the input sequence for RUL estimation. The mathematical formulation is represented as follows:

  h_t Given hidden states

$$e_t = \tanh(W_a h_t + b_a) \tag{6}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{7}$$

  The context vector is

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{8}$$

  Where:

  - W_a, b_a: are learnable parameters.
  - α_t: is the normalized attention weight.

- **Regression Output Layer**

  A fully connected dense layer followed by a linear activation outputs the RUL prediction. The mathematical formulation is represented as follows:

$$\hat{y} = W_o c + b_o \tag{9}$$

  Where W_o and b_o are learnable parameters

- **Loss Function and Optimization Strategy**

  In the proposed architecture, the Huber loss function [29] was adopted instead of the conventional Mean Squared Error (MSE) to enhance the model's robustness against noisy measurements and occasional outliers present in the CMAPSS dataset. Unlike MSE, which applies a purely quadratic penalty and is therefore highly sensitive to large deviations, the Huber loss integrates the strengths of both MSE and Mean Absolute Error (MAE). For residuals within a predefined threshold δ, it behaves quadratically like MSE, enabling precise gradient updates and preserving the model's sensitivity to small fluctuations in the Remaining Useful Life (RUL) signal. When the error magnitude exceeds δ, the loss transitions to a linear penalty similar to MAE, thereby reducing the impact of extreme deviations on the optimization process [20].

  This dual nature makes the Huber loss well-suited for turbofan engine RUL estimation, where sensor readings may occasionally include spikes or irregular patterns caused by mechanical disturbances, environmental variability, or sensor drift. By moderating the influence of such anomalies, the training process achieves more stable convergence and improved generalization across both FD001 and FD003 subsets. As a result, the model is better able to identify underlying degradation trends without being disproportionately influenced by rare measurement outliers.

  Optimization was performed using the Adam algorithm [12], a momentum-based adaptive optimizer that adjusts individual parameter updates according to the estimated first and second moments of the gradients. This mechanism accelerates convergence while maintaining stability during training, ensuring that the model efficiently learns complex temporal–spatial patterns from the multivariate sensor signals.

  The Huber loss $L_\delta(y, \hat{y})$ for a prediction $\hat{y}$ and true target y is defined as:

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |y - \hat{y}| \le \delta \\ \delta \cdot |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{10}$$

  where $\delta > 0$ is a tunable threshold parameter. For residuals $|y - \hat{y}|$ smaller than δ, the loss behaves quadratically, similar to the Mean Squared Error, ensuring that small deviations are penalized strongly. For residuals larger than δ, the loss becomes linear, reducing the impact of extreme errors and making the model less sensitive to outliers.

  The optimization is performed using the Adam algorithm, which adaptively updates each parameter θ_t at iteration t using estimates of the first moment m_t (mean of gradients) and the second moment v_t (uncentered variance of gradients):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{11}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{12}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{13}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{14}$$

  where g_t is the gradient at iteration t, α is the learning rate, β_1 and β_2 are decay rates for the moving averages, and ϵ is a small constant to prevent division by zero. Adam's adaptive scaling and bias correction improve convergence speed and stability, particularly in models with high-dimensional parameter spaces and non-stationary gradient distributions.

- **Learning Rate Scheduler** A custom learning rate scheduler dynamically adjusts the learning rate during training to balance convergence speed and stability. The learning rate η at epoch e is defined as:

$$\eta(e) = \begin{cases} \eta_0, & e < E_1 \\ \eta_1, & E_1 \leq e < E_2 \\ \eta_2, & e \geq E_2 \end{cases} \tag{15}$$

  This approach accelerates learning in early stages and refines convergence in later stages.

The proposed RCBLA architecture has been specifically designed to predict the Remaining Useful Life (RUL) of turbofan engines using only raw sensor measurements, without incorporating any operational settings or external degradation mode parameters. This design choice highlights the model's ability to autonomously learn degradation-related patterns directly from high-dimensional sensor data, reflecting a realistic and data-driven approach to predictive maintenance scenarios.

The architecture combines four key deep learning components in a unified and complementary framework. First, Residual Convolutional Blocks (RCB) enable the extraction of localized degradation features while preserving gradient stability in deep structures. This facilitates the learning of complex spatial patterns within sensor time-series data, even in the presence of noise and varying signal magnitudes. Second, the Temporal Convolutional Network (TCN) component efficiently captures both short- and long-term temporal dependencies, allowing the model to identify gradual degradation trends as well as sudden anomalies in engine performance. Third, Bidirectional LSTM layers incorporate information from both past and future time steps, enabling the detection of degradation patterns that may not be apparent in a single temporal direction. Finally, the attention mechanism assigns adaptive importance weights to sensor channels and time steps, ensuring that the model focuses on the most informative signals while suppressing irrelevant or noisy inputs.

From a generalization perspective, the integration of Layer Normalization and Dropout ensures robustness against overfitting, enabling the model to maintain high predictive accuracy across different engines and operating conditions without retraining. The residual connections further enhance transferability by allowing deeper feature hierarchies to be learned without the vanishing gradient problem.

In predictive maintenance applications, this combination of convolutional, recurrent, and attention-based mechanisms is particularly valuable. The model can accurately forecast RUL well before critical failure, enabling maintenance teams to take proactive actions. This reduces unplanned downtime, optimizes resource allocation, and extends the overall service life of the engine. Furthermore, by eliminating reliance on operational settings, the architecture is more generalizable and can be deployed in environments where only sensor readings are available, making it adaptable to a wider range of industrial scenarios.

Overall, the RCBLA architecture not only demonstrates strong predictive performance but also embodies a high degree of adaptability, scalability, and reliability, making it a practical and powerful solution for real-world early fault detection and RUL prediction tasks in complex machinery. A comprehensive overview of the proposed approach is presented in Table 1 and illustrated in Figure 1 ,while the RCBLA model is formally described in Algorithm 1.

Algorithm 1 Proposed RCBLA Hybrid Model for RUL Prediction

Require: Hyper-parameters of the trained model (window length, batch size, dropout rate, learning rate, number of epochs, etc.), training set.

Output: Trained model for RUL prediction

Data Preprocessing: perform Feature selection from sensor data, normalize features, piecewise degradation model, time window processing

Build: RCBLA hybrid Model

initialize: The parameters of RCBLA hybrid Model initialization

Model Training:

for epoch=1,2,...,30: **do**

  step 1 **Build Residual CNN Block:** Extract local spatial features and reduce noise from input sequences using residual connections.

  step 2 **Build Temporal Convolutional Network (TCN):** Capture multi-scale temporal dependencies using dilated causal convolutions.

  step 3 **Build BiLSTM Block:** Learn bidirectional temporal patterns.

  step 4 **Build Learnable Attention:** Assign importance weights to features.

  step 5 **Apply Layer Normalization:** Normalize intermediate activations for stability.

  step 6 **Add Fully Connected Layers:** Map extracted features to the RUL output space. It generates a final prediction. The proposed model is trained using loss function with the custom learning rate scheduler, Step 1 - Step 6 represent the process of model training

  Step 7 **Evaluation:** The model is compiled using the Adam optimizer. Mean Absolute Error and Root Mean Squared Error metrics are tracked during training for performance monitoring.

Table 1: The architecture of RCBLA model

	Layers	Parameters
Residual CNN Encoder	Conv1D	Filters=64, kernel=3, padding=Same
	MaxPooling1D	pool=2, padding=same
	residual block	Filters=64, kernel=3
	LayerNormalization	
	Conv1D	Filters=128, kernel=3, Activation=ReLU, padding=Same
	MaxPooling1D	pool=2, padding=same
	residual block	Filters=64, kernel=3
	LayerNormalization	
Temporal Convolutional Network Layer	Conv1D	Filters=128, kernel=3, dilation=2, padding=causal, activation=relu
	LayerNormalization	
Bidirectional LSTM Layer	Bidirectional	LSTM (32, return_sequences=True)
Learnable Attention	attention block Function	
Dense Regression Head	Dense Layer	Filters=64, activation=relu
	Dropout Layer	
Output Layer	Dense Layer	1, activation=linear
Final Model Compilation	Optimizer	Adam
	Learning Rate	0.001 (FD001) and 0.005 (FD003)
	Loss Function	Huber loss
	Metrics	Mean Absolute Error, Root Mean Squared Error

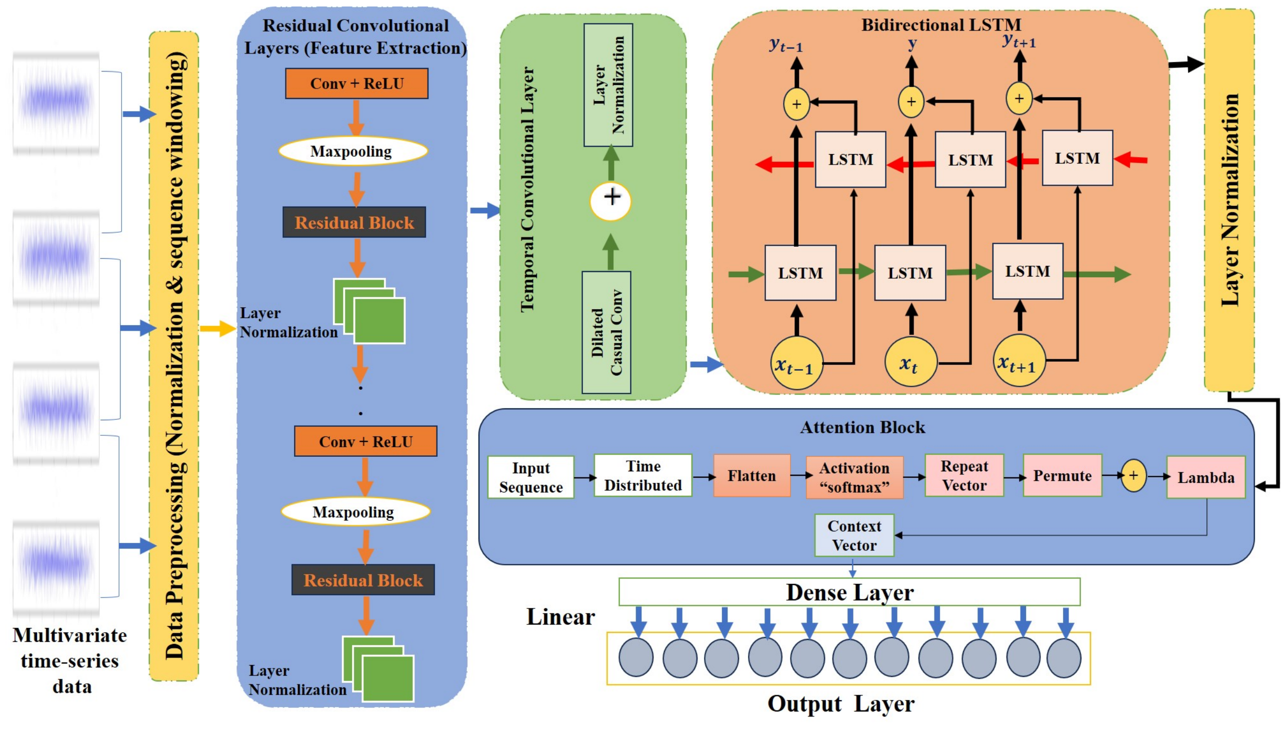

Figure 1: RCBLA Model

4 Experimental verification

4.1 Data description

This study utilized the NASA C-MAPSS (Commercial Modular Aero Propulsion System Simulation) tool, which is designed for simulating turbofan engines, to generate the data. C-MAPSS can simulate a variety of operational, environmental, and control scenarios by modifying the input parameters [1]. Figure 2 presents the diagram of the C-MAPSS simulation engine [9]. C-MAPSS provides run-to-failure statistics for 249 engines simulated under six operational parameters. To enhance realism, these engines incorporate manufacturing variations and differing initial wear levels, with initial wear resulting from varying module efficiencies. Each engine experiences a defect due to one of two failure modes: High Pressure Compressor (HPC) Degradation or Fan Degradation. Initially, the engines operate normally; however, a defect is introduced at some point, leading to eventual engine failure [1].

A total of 21 sensors are installed on different engine modules to monitor the health of the engines. Table 2 presents a detailed list of these sensors. In addition to the 21 sensor readings, three additional parameters are recorded to represent the operational states of the engine. Table 3 lists these operational parameters. Data from all sensors and operational characteristics are captured at each engine cycle [9].

Table 4 provides an overview of the C-MAPSS dataset, which consists of four sub-datasets: FD001, FD002, FD003, and FD004, each characterized by distinct conditions and failure modes. Each sub-dataset is divided into training and testing datasets. The C-MAPSS dataset contains 26 columns per row, with the first five columns detailing the engine unit number, the degradation time step for each engine, and engine operational settings. The remaining 21 columns represent multivariate time series data from the sensors during the same operating cycle. As outlined in Table 2, the sensor data includes metrics such as temperature, pressure, and speed, with further details available in [23] [9]. Additionally, these sensor readings depict the progression of the engine from a normal state through initial wear to gradual failure. This study employs the proposed model to analyze the training dataset and estimate the Remaining Useful Life (RUL) for the engines in the test dataset[9].

In this study, the FD001 and FD003 datasets were utilized to analyze the performance and predict the Remaining Useful Life (RUL) of turbofan engines. These datasets provide a comprehensive framework for evaluating engine degradation under various operational conditions.

FD001 contains one operating condition and one fault mode, whereas FD003 contains one operating condition and two fault modes. It consists of multiple engine cycles, with each cycle representing a unique operational scenario. Figure 30 in Appendix A illustrates the distribution of the number of cycles across different engines in the FD001 dataset. Each bar represents the total number of operational cycles for individual engines, with the y-axis indicating the engine identifiers and the x-axis representing the corresponding number of cycles. It is evident that the majority of engines have a relatively uniform distribution of cycles, with some engines exhibiting a higher number of cycles, reaching up to 525. This distribution is crucial for understanding the operational lifespan of the engines and the extent of data available for analysis.

The FD003 dataset shares similar characteristics with FD001, but it is distinguished by its unique operational conditions and failure modes. Figure 31 in Appendix A depicts the number of cycles for different engines in the FD003 dataset. Similar to the visualization for FD001, this figure shows the distribution of engine cycles, with the highest recorded cycles reaching 362. This analysis is important for evaluating the dataset's robustness and ensuring that a sufficient number of cycles are available for training and validating predictive models.

4.2 Performance metrics

The RUL prediction model for the system leverages data from multiple sensors to forecast the degradation of aircraft engines, with the predicted RUL as the resulting output. To evaluate the performance of the employed techniques, two metrics are utilized: Root Mean Square Error (RMSE), and a scoring function.

1. Root Mean Square Error (RMSE)[33] This metric is commonly employed to evaluate the performance of regression models. RMSE measures the average magnitude of errors between the predicted values and the actual values within a dataset. A straightforward explanation of RMSE is as follows [9]:

$$RMSE_{RUL} = \sqrt{(1/N)\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \qquad (16)$$

   where:

   - N: denotes the total number of samples in the test dataset.
   - Y_i: signifies the actual value of the engine life.
   - $\hat{Y}_i$: represents the predicted value of the engine life.

2. Scoring Function [22]: This is typically defined as a function that evaluates the performance of a model on a specific dataset. Scoring functions are utilized to assess a model's capability to predict or classify data accurately. They represent the average value of the test set fraction, where

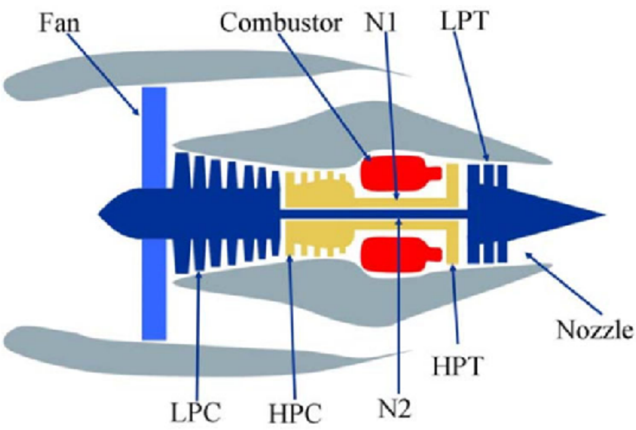

Figure 2: Simplified Simulated engine diagram in C-MAPSS (Fig 2 in [23]).

Table 2: The description of sensors and their units

Sensor number	Symbol	Description	Unit of measure
S1	T2	Total temperature at fan inlet	degrees Rankine (°R)
S2	T24	Total temperature at LPC outlet	degrees Rankine (°R)
S3	T30	Total temperature at HPC outlet	degrees Rankine (°R)
S4	T50	Total temperature at LPT outlet	degrees Rankine (°R)
S5	P2	Pressure at fan inlet	Pounds Per Square Inch Absolute (PSIA)
S6	P15	Total pressure in bypass-duct	Pounds Per Square Inch Absolute (PSIA)
S7	P30	Total pressure at HPC outlet	Pounds Per Square Inch Absolute (PSIA)
S8	Nf	Physical fan speed	Revolution Per Minute (rpm)
S9	Nc	Physical core speed	Revolution Per Minute (rpm)
S10	epr	Engine pressure ratio (P50/P2)	Nil
S11	Ps30	Static pressure at HPC outlet	Pounds Per Square Inch Absolute (PSIA)
S12	phi	Ratio of fuel flow to Ps30	pps/psi
S13	NRF	Corrected fan speed	Revolution Per Minute (rpm)
S14	NRc	Corrected core speed	Revolution Per Minute (rpm)
S15	BPR	Bypass Ratio	Nil
S16	farB	Burner Fuel-air ratio	Nil
S17	htBleed	Bleed Enthalpy	Nil
S18	NF_dmd	demanded fan speed	Revolution Per Minute (rpm)
S19	PCNfR_dmd	demanded corrected fan speed	Revolution Per Minute (rpm)
S20	W31 HPT	HPT coolant bleed	lbm/s
S21	W32 LPT	LPT coolant bleed	lbm/s

Table 3: Operational Parameters

Operational Parameter	Description
Tr	Throttle Resolver Angle (TRA)
Al	Altitude
MN	Match number

a lower score indicates better performance. The formula for calculation is as follows [9]:

$$Score = \begin{cases} \sum_{i=1}^{N}(e^{\frac{-dif_i}{13}} - 1), & \text{if } dif_i < 0 \\ \sum_{i=1}^{N}(e^{\frac{dif_i}{10}} - 1), & \text{if } dif_i \geq 0 \end{cases} \quad (17)$$

When utilizing deep learning models to predict Remaining Useful Life (RUL), Eq.17 indicates that the scoring function imposes greater penalties on later predictions, whereas RMSE treats both later and earlier predictions with equal weight. Later predictions are associated with a higher risk of serious accidents. Figure 3 compares RMSE and the scoring function, illustrating that the penalty for the final prediction is more pronounced than that of RMSE, highlighting the importance of preventing engine failures. The linear

Table 4: C-MAPSS datasets description

Datasets	FD001	FD002	FD003	FD004
No. of engines for training	100	260	100	249
No. of engines for testing	100	259	100	248
Operation conditions	1	6	1	6
Fault mode number	1	1	2	2
Training sample	20631	53759	24720	61249
Test sample	13096	33991	16596	41214

relationship between RMSE and error values suggests a clear physical interpretation. The model's predictive capability was assessed using both RMSE and the scoring function, both of which serve as important indicators [9].

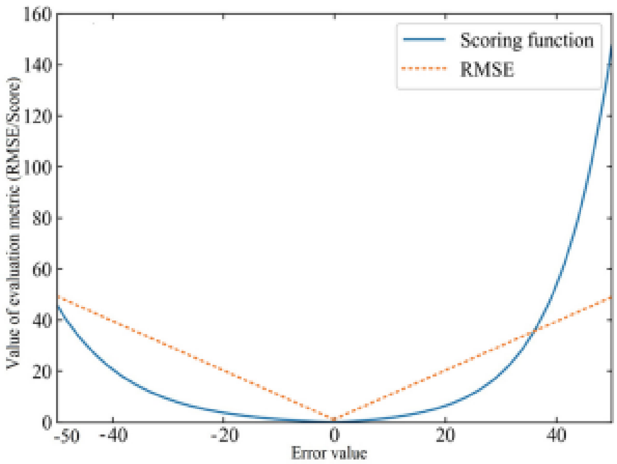

Figure 3: RMSE versus scoring function (Fig.7 in [18])

4.3 Data Preprocessing

4.3.1 Sensor selection

In this study, the proposed approach is applied to both the FD001 and FD003 subsets of the CMAPSS dataset. However, the exploratory analysis and visual illustrations presented in this section correspond exclusively to the FD001 training data. The first two columns of the dataset represent the engine ID and the operational cycle number, which are excluded from model training as they do not contribute to the learning of degradation patterns. Similarly, columns 3 to 5 contain operational setting parameters, which are not utilized in the proposed method. While other studies may integrate these parameters into the feature space, our framework is designed to leverage only the raw sensor measurements recorded in columns 6 to 26.

Figure 4 displays the boxplots of all sensor measurements for the FD001 training set. The plot reveals that sensors 1, 5, 10, 16, 18, and 19 produce constant readings throughout all operational cycles, while sensor 6 exhibits extremely limited variation, with most values concentrated around 21.61. Such features are considered uninformative for prognostics tasks, as they do not capture degradation-related changes. Additionally, constant-valued features present normalization issues due to their zero variance, leading to division-by-zero errors during scaling. Consequently, all constant or near-constant sensor channels are removed from the dataset before training.

Figure 5 illustrates the temporal trends of the remaining sensor signals for a selection of engines in the FD001 subset. These plots clearly highlight distinctive degradation trajectories over time, indicating that the selected sensors carry significant information relevant to Remaining Useful Life (RUL) prediction. Although these visualizations are provided only for FD001, the same preprocessing strategy is consistently applied to FD003 to ensure comparable feature quality across datasets. By focusing exclusively on informative sensor signals, the proposed model enhances its ability to detect early signs of fault progression and improve the accuracy of RUL estimation.

Figure 6 presents the standard deviation of each sensor across all engines in the FD001 dataset. Standard deviation is a statistical measure that quantifies the amount of variation or dispersion in a set of values. In this context, it indicates how much the sensor readings deviate from the mean value. The sensor with the highest standard deviation is Sensor 17, with a value of 22.083. This suggests that the readings from this sensor exhibit significant variability, which may indicate fluctuating operational conditions or sensor reliability issues. Sensor 1 shows a much lower standard deviation of 0.071, reflecting more consistent measurements throughout the operational cycles. Such consistency can be beneficial for predictive modeling, as it implies reliable data. High variability in certain sensors may warrant further investigation to understand the underlying causes, whether they stem from mechanical issues, environmental factors, or sensor malfunctions. Conversely, sensors with lower standard deviations can be prioritized for analysis in predictive maintenance, as their stable readings may provide clearer insights into engine health.

Figure 7 displays the standard deviation of each sensor from the FD003 dataset. Similar to the previous figure, this visualization helps to assess the variability of sensor readings across the engines in this dataset. The standard deviations of the sensors in FD003 vary from 0.173 to 0.343. Sensor 8 has the highest standard deviation of 0.343, indicating that its readings

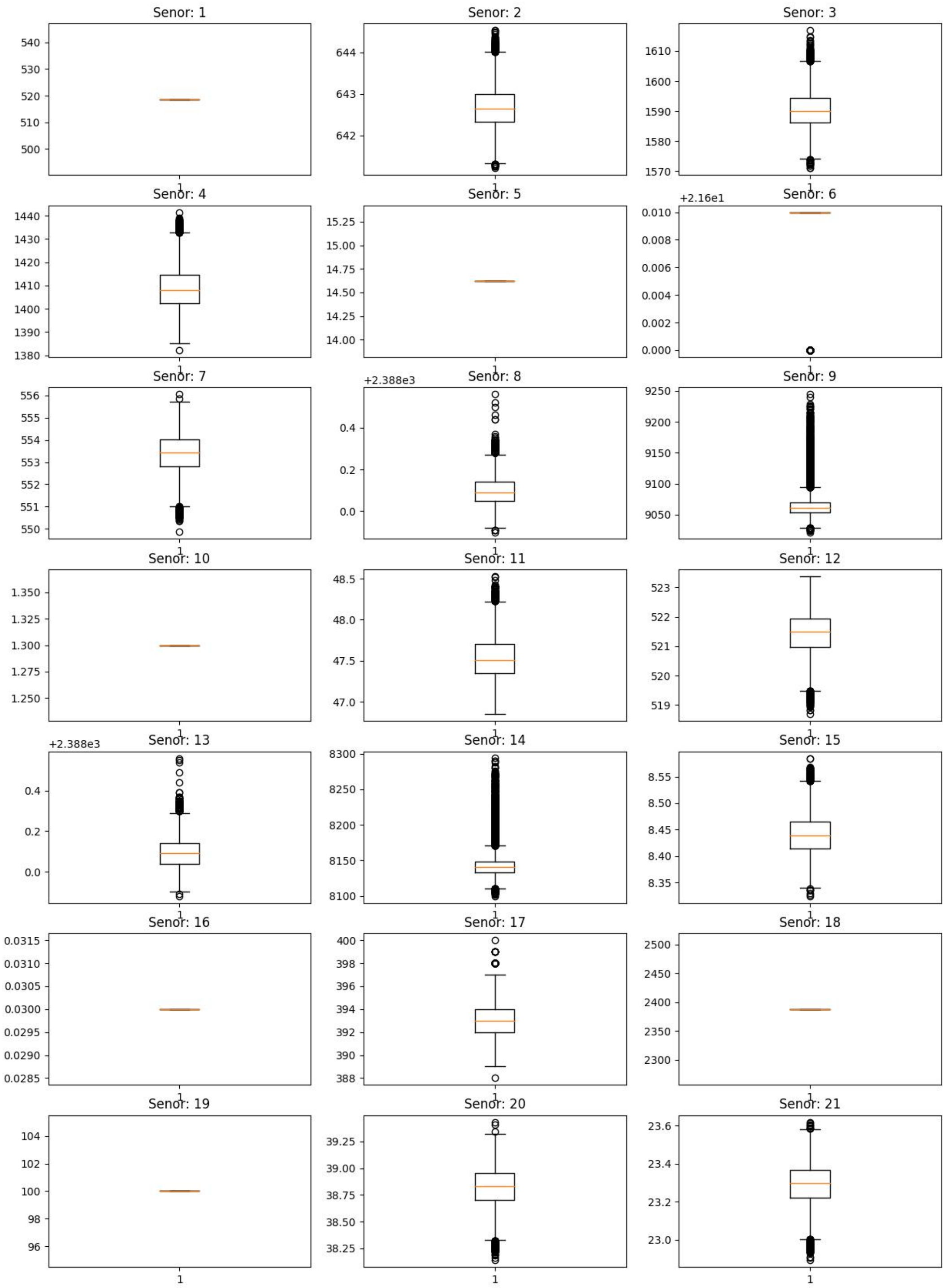

Figure 4: Boxplot of Sensor measurements in FD001 dataset

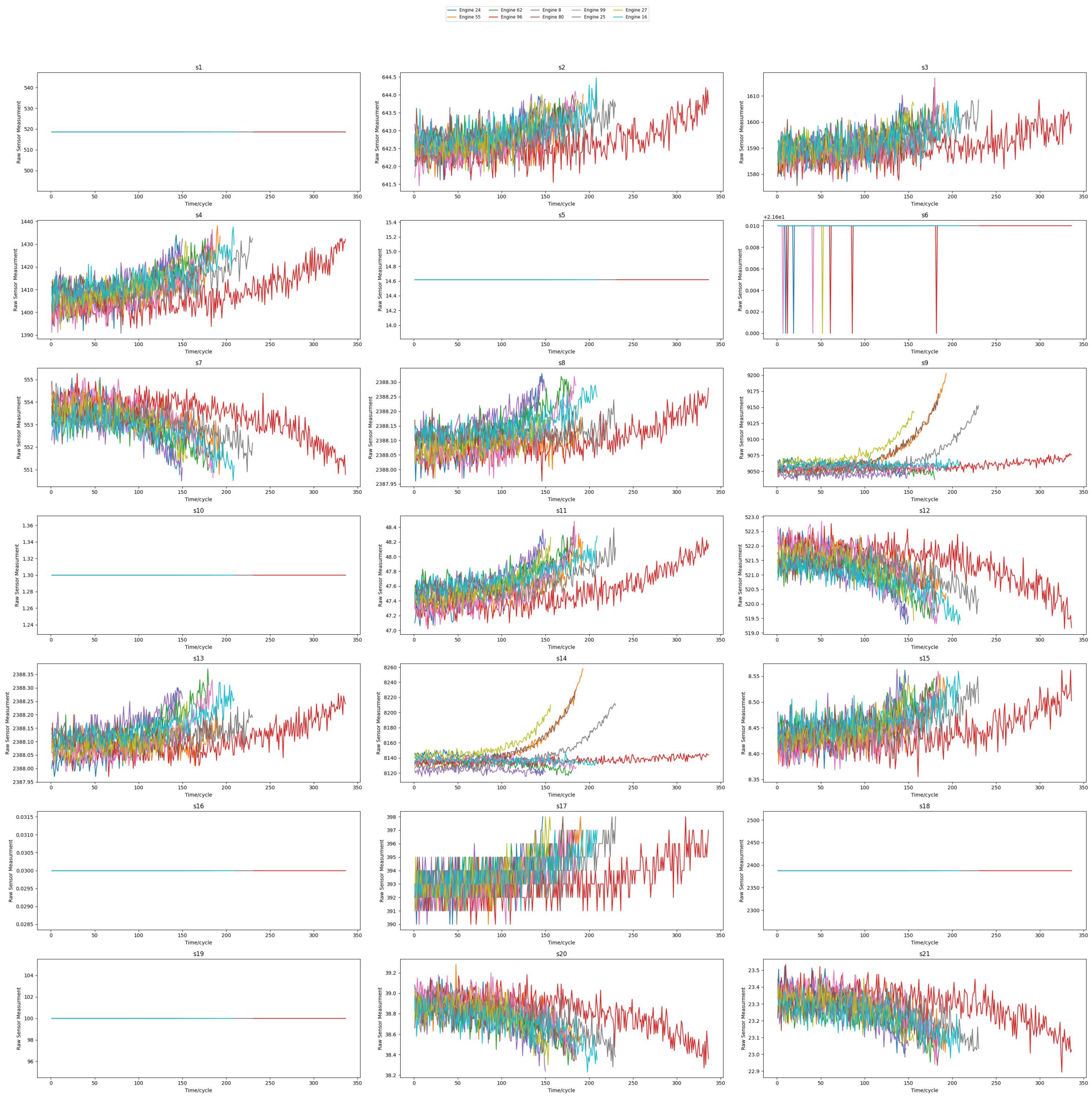

Figure 5: Sensor degradation measurements over Time for randomly selected engines in FD001 dataset.

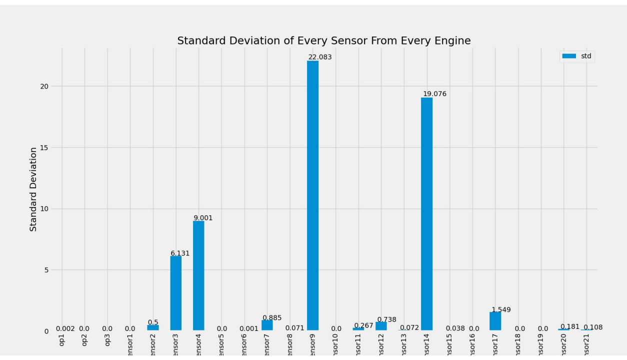

Figure 6: Standard deviation of each sensors from FD001 dataset.

are less stable compared to others, while Sensor 4 has the lowest standard deviation at 0.245. The relatively moderate range of standard deviations across the sensors suggests that while there are variations, they are not as extreme as those observed in FD001. The consistent behavior of many sensors in FD003 can aid in the development of more reliable predictive models. Understanding which sensors exhibit higher variability can help direct maintenance efforts and improve sensor calibration processes. Identifying and addressing the causes of variability in sensors can enhance the overall accuracy of engine health monitoring and RUL predictions.

The analysis of standard deviations in both the FD001 and FD003 datasets provides essential insights into the reliability and variability of sensor readings.

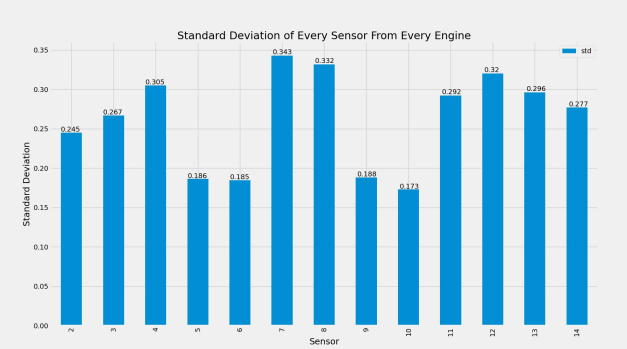

Figure 7: Standard deviation of each sensors from FD003 dataset.

By understanding these variations, researchers can make informed decisions about sensor prioritization, data interpretation, and predictive maintenance strategies, ultimately leading to improved engine performance and longevity.

4.3.2 Data Normalization

Data normalization is a fundamental preprocessing step in deep learning workflows, as it directly influences the stability and convergence of the training process. To maintain a consistent scale across all features, the measurements obtained from multiple sensors must be standardized [16]. In this study, the StandardScaler (Z-score) method is employed to normalize the input data. This transformation adjusts each feature to have zero mean and unit variance, ensuring comparability across different sensor dimensions. The Z-score formula is expressed as:

$$Z = \frac{x - \mu}{\sigma} \tag{18}$$

where:

- z:represents the standardized data.
- x: represents the original value of the feature.
- μ: is the mean value of the feature.
- σ: standard deviation of the feature.

Normalization is particularly important in Remaining Useful Life (RUL) prediction tasks, as the raw sensor measurements often vary widely in magnitude and distribution. Without normalization, features with larger numerical ranges may dominate the learning process, potentially leading to biased model parameters and degraded prediction accuracy. By bringing all sensor data to a uniform statistical scale, the model can effectively learn subtle degradation patterns without being skewed by scale differences. This preprocessing step also enhances the efficiency of gradient-based optimization methods, leading to faster convergence and improved generalization performance on unseen data.

The correlation analysis of sensors in the FD001 dataset is illustrated in Figure 8. This figure provides a visual representation of the relationships among the various sensors, allowing for an examination of how sensor readings correlate with one another. The correlation coefficients range from -1 to 1, where values closer to 1 indicate a strong positive correlation, values near -1 indicate a strong negative correlation, and values around 0 suggest no correlation. The heatmap reveals distinct patterns of correlation among the sensors, highlighting which sensors tend to exhibit similar behaviors during operation. For instance, certain sensors may show strong positive correlations, suggesting they respond similarly to changes in engine conditions or operational parameters. Understanding these correlations can be critical for identifying redundant sensors or determining which sensors contribute most significantly to predictive modeling.

Figure 9 presents a correlation analysis that includes all sensors, without excluding those identified during the data selection process. This comprehensive view allows for a broader understanding of the relationships among all sensors in the dataset. In this figure, the correlations among the included sensors can be further assessed, revealing additional insights into how each sensor's readings interact with one another. The inclusion of previously excluded sensors provides a more complete picture of the dataset.

The correlation analyses presented in Figures 8 and 9 are essential for understanding the interdependencies among sensors in the FD001 dataset. By identifying relationships among sensor readings, we can make informed decisions regarding sensor selection, data interpretation, and model development, ultimately enhancing the effectiveness of predictive maintenance strategies.

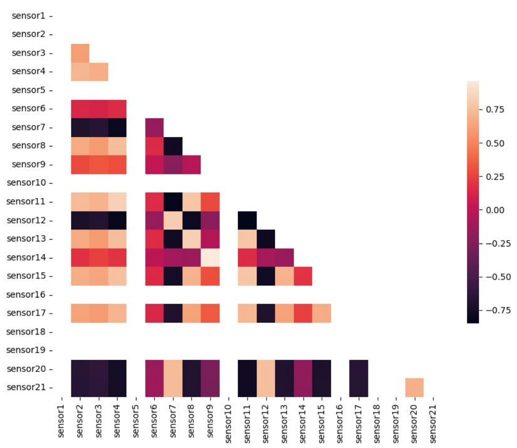

Figure 8: Heatmap of all sensors in dataset FD001.

The time series plots presented for the FD001 dataset illustrate the readings from various sensors

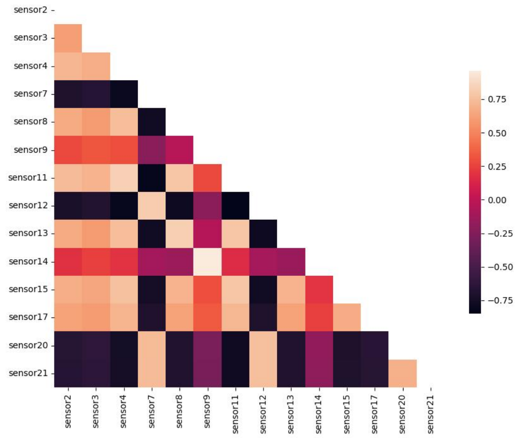

Figure 9: Heatmap of all sensors without sensor 1, 5, 6, 10, 16, 18 and 19.

across multiple operational cycles. Each subplot represents a different sensor, enabling a visual examination of how sensor readings fluctuate over time. The sensor data signals prior to standardization are illustrated in Figure 10, while the corresponding standardized signals are presented in Figure 11.

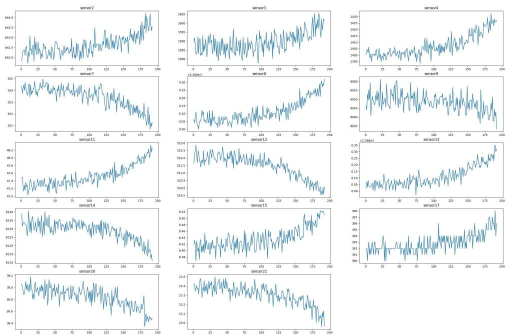

Figure 10: Raw sensor data before normalization.

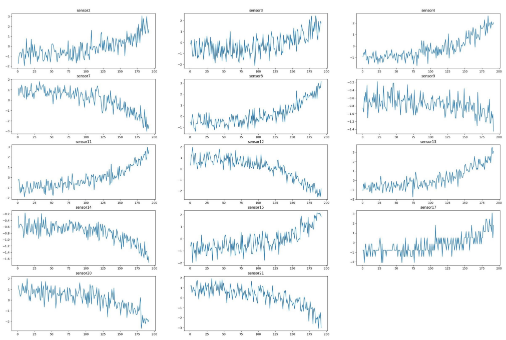

Figure 11: Standardized sensor data.

4.3.3 Piecewise degradation model

Accurate estimation of Remaining Useful Life (RUL) requires careful analysis of the turbofan engine's operational characteristics. Under normal operating conditions, the engine's lifespan declines in a measurable and structured manner. For modeling purposes, it can be assumed that no degradation occurs during the initial operating phase, during which the engine maintains a constant RUL. Once the RUL reaches a specific threshold, a progressive decline begins, typically following a linear degradation pattern until complete failure [18].

Piecewise linear degradation models offer greater flexibility in capturing diverse deterioration behaviors compared to exponential models, which impose a fixed rate of decline. This approach can account for abrupt changes in degradation trends, aligning more closely with realistic operating scenarios. In addition, such models improve interpretability by making it easier to distinguish critical operational stages and are less sensitive to sensor noise, thereby reducing overfitting risks. When integrated with Residual Convolutional and Bi-directional Long Short-Term Memory architectures (RCBLA), they can effectively capture localized degradation trends and transition phases, leading to higher predictive accuracy and more informed maintenance scheduling [8].

In the FD001 subset of the CMAPSS dataset, the training set does not directly provide RUL labels; however, these values can be easily derived since the data represents complete run-to-failure histories for all engines. For example, if Engine 1 operates for 192 cycles before failure, its RUL at the first cycle is 191, decreasing by one unit per cycle until it reaches zero at the final cycle. This pattern represents the linear degradation model, where the RUL decreases uniformly from the start of operation to failure [9].

Alongside the purely linear approach, a piecewise linear degradation model is often adopted to better reflect practical maintenance scenarios. In this model, the RUL is initially assigned a fixed "early RUL" value for a certain number of initial cycles, reflecting the stable performance phase of the engine. Once this threshold is reached, the RUL begins to decline linearly until failure. For demonstration, Figure 12 compares both models using Engine 1, where the early RUL is set to 125 cycles [19]. This choice is flexible and typically guided by empirical observations or domain knowledge rather than fixed rules. Figure 13 illustrates that the Remaining Useful Life (RUL) label decreases in a linear fashion over time, ultimately leading to a complete failure.

The piecewise model offers several advantages over the purely linear approach: it captures the initial plateau phase often observed in machinery health trajectories, reduces premature sensitivity to early-cycle noise, and aligns more closely with real-world degradation patterns where engines maintain peak performance before exhibiting measurable wear.

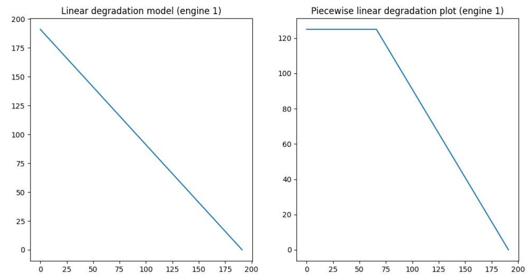

Figure 12: Difference between linear degradation and piecewise linear model [9].

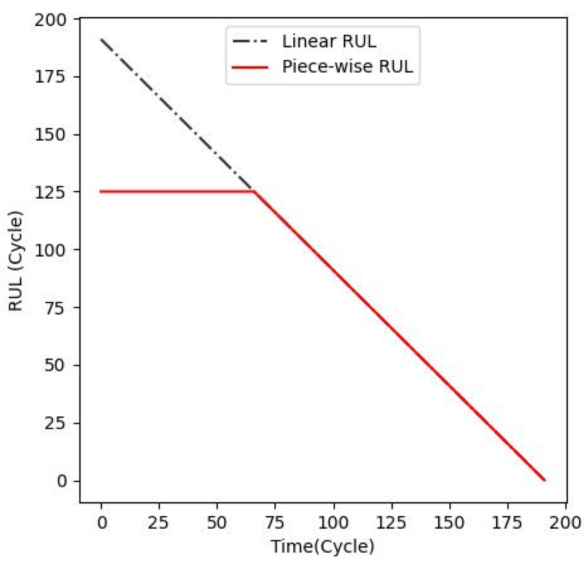

Figure 13: Piecewise linear degradation Model [9].

4.3.4 Time window Processing

Time window processing is an essential technique for structuring time series data [18], enabling the model to capture temporal dependencies by segmenting sensor readings into fixed-length subsequences. As illustrated in Figure 14, a sliding window of length 30 cycles is applied, where the sequence from time step t to $(t + 30 - 1)$ is treated as a training instance. This allows the model to learn degradation patterns by utilizing historical sensor values across multiple channels $(e.g., Sensors 1, 2, 3, and 14)$. By advancing the window one step at a time, the method ensures that short-term and long-term dependencies are effectively preserved.

The benefits of time windowing are twofold. First, it provides the network with localized temporal context, allowing it to recognize early degradation signatures that may not be visible in individual time points. Second, it increases the amount of training data by generating overlapping subsequences, which helps improve model generalization and reduces the risk of overfitting [9].

For the CMAPSS subsets FD001 and FD003, the choice of window length was guided by the minimum engine life cycles in the datasets (128 and 145 for training; 31 and 38 for testing, respectively). Through experimental evaluation of different window sizes (30, 35, 40, and 45), the configuration in Table 5 of 30 cycles demonstrated the best predictive performance. This selection achieves a balance between retaining sufficient temporal information and ensuring compatibility with the shortest engine sequences, making it the most suitable setting for model training.

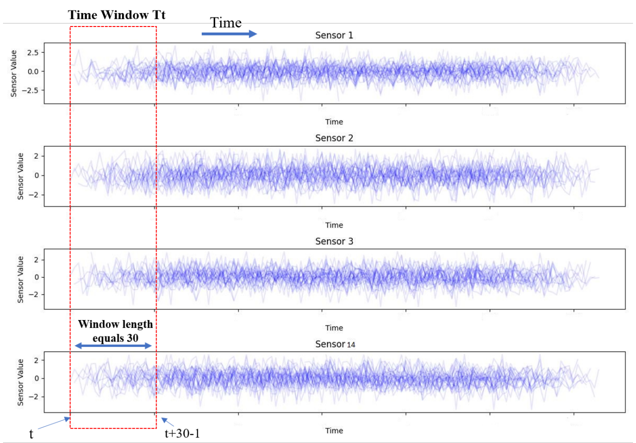

Figure 14: Time window processing [9].

4.4 Experimental analysis and results

4.4.1 Experimental setting and description

All experiments were conducted using TensorFlow version 2.16.1 as the primary deep learning framework, along with supporting libraries including NumPy 1.26.4, Pandas 2.2.3, and Scikit-learn 1.2.2 for data preprocessing and evaluation tasks. The development environment was set up on a machine equipped with an Intel(R) Core(TM) i7 processor, 8 GB of RAM, and running the Windows 10 operating system. Additionally, the training and testing processes were accelerated by utilizing the computational resources of the Kaggle platform, which provides enhanced GPU support and optimized execution for deep learning workloads. Table 6 lists the suggested model's hyper-parameters. In Table 7, we present the RMSE values for the proposed model across the two datasets: 14.38 for FD001 and 12.46 for FD003, along with scores of 279.23 and 282.56, respectively. These results indicate a strong prediction capability, with relatively low prediction error and a significant improvement over earlier experiments. The S-score is particularly important for prognostics applications as it penalizes overestimation more heavily than underestimation, which aligns with safety-critical system requirements.

The proposed hybrid deep learning model—composed of Residual CNN, Temporal Convolutional Networks (TCN), Bidirectional LSTM, attention mechanisms, and Layer Normalization—was evaluated on the FD001 and FD003 subsets of the

Table 5: Time window for two subsets FD001 and FD003 [9].

Subsets	Min life cycle in training datasets	Min life cycle in testing datasets	Time window size
FD001	128	31	30
FD003	145	38	30

Table 6: RCBLA proposed model Hyper-parameters

Parameters	**FD001**	**FD003**
size	128	64
Training rounds	30	30
Activation Function	relu	relu
Activation Function "attention block"	softmax	softmax
Activation Function "output"	linear	linear
Residual CNN Encoder	2	2
TCN Layer	1	1
BiLSTM Layer	1	1
Learning rate	0.001	0.005
convolutional kernel size	3	3
Dropout	0.7	0.5
Optimizer	Adam	Adam
Loss	Huber loss (delta=1.0)	Huber loss (delta=1.0)

Table 7: RCBLA proposed model performance results on FD001 and FD003

Subsets	**FD001**	**FD003**
RMSE	14.38	12.46
Score	279.23	282.56

CMAPSS dataset. These subsets were selected due to their distinct characteristics. The model was trained using the configuration in Table 6, applied consistently across both datasets besides using the learning rate scheduler where the scheduler was designed to enable aggressive learning during early training phases, followed by gradual refinement in later epochs to ensure convergence and avoid overfitting.

For FD001 dataset, The RMSE plot in Figure 15 across training epochs shows a sharp decline in validation error during the initial epochs, followed by stabilization. This indicates rapid initial learning and effective generalization, with no signs of overfitting due to the high dropout regularization. The final validation RMSE plateaued at a value close to 10, indicating robust performance. The training and validation loss curves in Figure 16 reveal consistent minimization across epochs, with the validation loss closely tracking the training loss. This further confirms the model's stability and effectiveness in learning temporal degradation patterns without significant overfitting. The scatter plot comparing predicted RUL values to the true RUL in Figure 17 for each engine instance (Engine ID 1–100) shows that the model successfully tracks the overall trend of degradation across most instances. While some fluctuations are present expected due to high dropout and complex temporal dynamics the predictions remain within an acceptable range. The model avoids large overestimation errors, which is critical for safety-related prognostics. The presented results on FD001 demonstrate that the combination of high dropout regularization and a decaying learning rate strategy leads to effective generalization and improved prediction accuracy. The model maintains low error metrics and a favorable S-score, making it a reliable candidate for predictive maintenance applications where false positives and overestimated RUL values are particularly undesirable.

For FD003, The RMSE curve shown in Figure 18 a consistent decline over training epochs. The validation RMSE closely follows the training curve, indicating proper generalization and minimal overfitting. Convergence was achieved approximately around the 15th epoch, which is consistent with the early learning rate decay phase in the scheduler. This validates the learning rate design, where an aggressive start followed by controlled refinement phases promotes efficient convergence. The regression loss curve (Huber loss) shows in Figure 19 a stable and gradual decrease over time. Notably, the validation loss remains nearly parallel to the training curve, which is a positive indication that the model is not simply memorizing patterns from the training data but is instead learning generalizable degradation trends. The smoothness of the curve also suggests that the optimizer and loss function are working harmoniously to reduce error without introducing instability. The scatter plot compares predicted RUL values against the ground-truth labels in

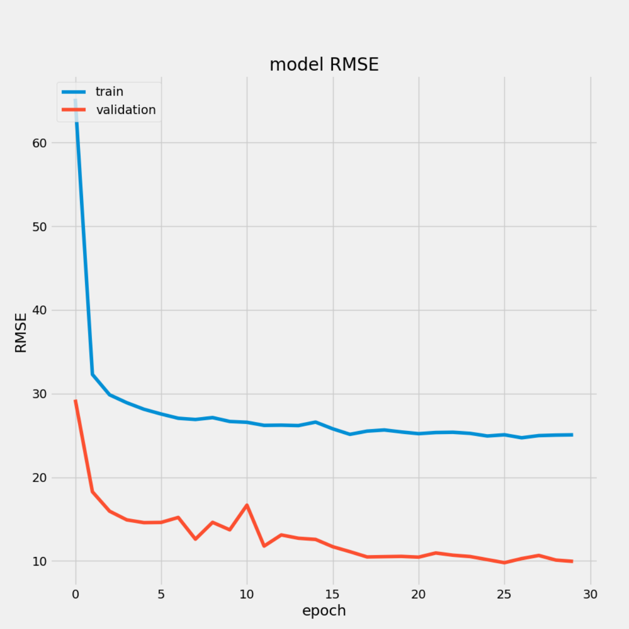

Figure 15: Root Mean Square Error for FD001 dataset.

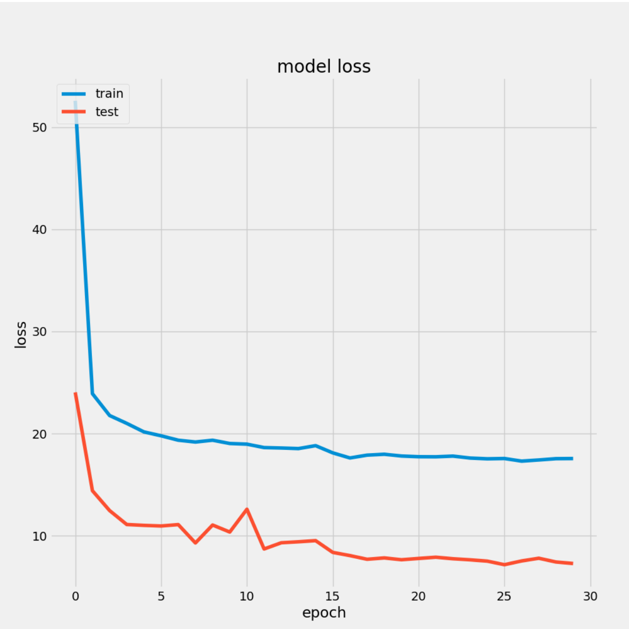

Figure 16: Model Loss for FD001 dataset.

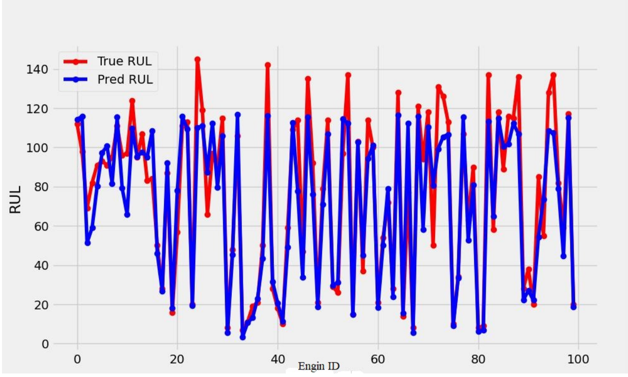

Figure 17: Predicted RUL values (FD001).

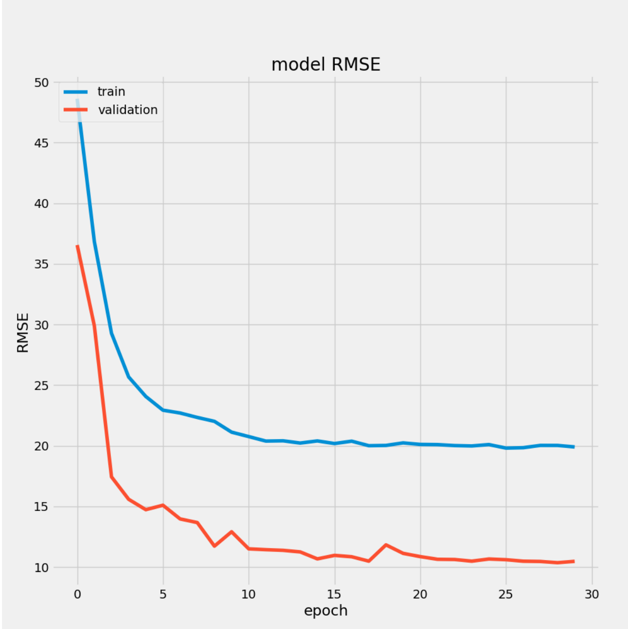

Figure 18: Root Mean Square Error for FD003 dataset.

Figure 20. The data points lie closely along the diagonal, demonstrating high agreement between predicted and actual values. While some scatter is inevitable due to the complexity of the degradation processes, the majority of the predictions fall within an acceptable deviation range. This strong alignment supports the claim that the model learns useful temporal features for accurate RUL estimation.

4.4.2 Comparison with different methods

The proposed RCBLA method is compared with existing state of the art methods including SVR [22], RVR [22], CNN [22], LSTM [35], DCNN [16], CNN+LSTM [13], Bi-LSTM-ED [32], Attention-LSTM [6], VAE-RNN [7], CNN-LSTM-Attention [8], CNN-GRU [3], STCAN [31], and CAELSTM [9]. Table 8 presents the comparative results of various methods for Remaining Useful Life (RUL) prediction on the CMAPSS FD001 and FD003 subsets, using RMSE and score metrics as performance indicators. Early approaches, such as SVR and RVR, demonstrated relatively high error values, with RMSEs exceeding 20 on both datasets. Deep learning-based methods, including CNN, LSTM, and their variants, achieved substantial improvements, reducing RMSE values to the range of 12–18. Hybrid architectures such as CNN-LSTM and Bi-LSTM-ED further enhanced prediction accuracy, achieving scores below 600 on FD003.

More advanced techniques that incorporated attention mechanisms, such as Attention-LSTM and CNN-LSTM-Attention, showed superior capability in capturing long-term dependencies, yielding lower RMSEs of around 15–16 and reduced score values compared to earlier methods. Recently proposed models like VAE-RNN, STCAN, and CAELSTM further advanced the

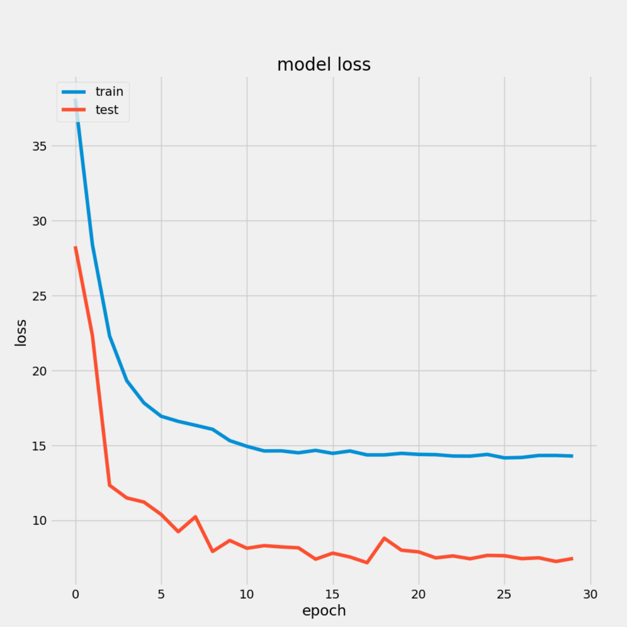

Figure 19: Model Loss for FD003 dataset.

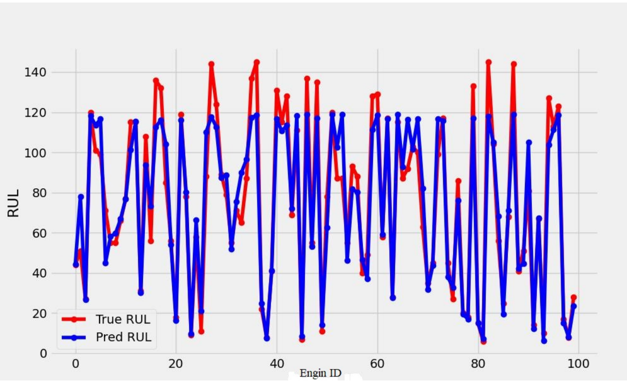

Figure 20: Predicted RUL values (FD003).

state-of-the-art, attaining RMSE values around 13–15 and competitive score results, confirming the effectiveness of combining spatial-temporal modeling with attention mechanisms.

The proposed RCBLA model demonstrates competitive performance compared to these methods. On FD001, it achieves an RMSE of 14.38 and a score of 279.23, while on FD003, it obtains an RMSE of 12.46 and a score of 282.56. These results indicate that RCBLA not only surpasses several earlier approaches but also provides accuracy comparable to or better than more recent advanced architectures. This highlights the robustness and efficiency of the model in learning degradation patterns and accurately predicting RUL in complex aerospace datasets.

5 Ablation Study Results

To assess the contribution of each architectural component within the proposed RCBLA model, we conducted a structured ablation study by selectively removing key modules—namely the Residual CNN encoder, Temporal Convolutional Network (TCN), and Bidirectional LSTM—and evaluating the resulting performance on FD001 and FD003 subsets. Table 9 summarizes the RMSE and S-score metrics across all variants. The full model achieved the lowest RMSE values (14.38 for FD001 and 12.46 for FD003) and competitive S-scores (279.23 and 282.56 respectively), confirming the effectiveness of the hybrid architecture.

Removing the Residual CNN led to the most significant degradation in generalization, especially on FD003, where RMSE increased to 18.54 and the S-score rose sharply to 757.58. The RMSE and MAE curves for FD003 in Figure 21 and Figure 22 show higher error levels and less stable convergence compared to the full model. For FD001, the corresponding MAE curve 23 reveals a slower decline in training error and a wider gap between training and validation MAE, indicating reduced feature extraction capacity without the residual encoder.

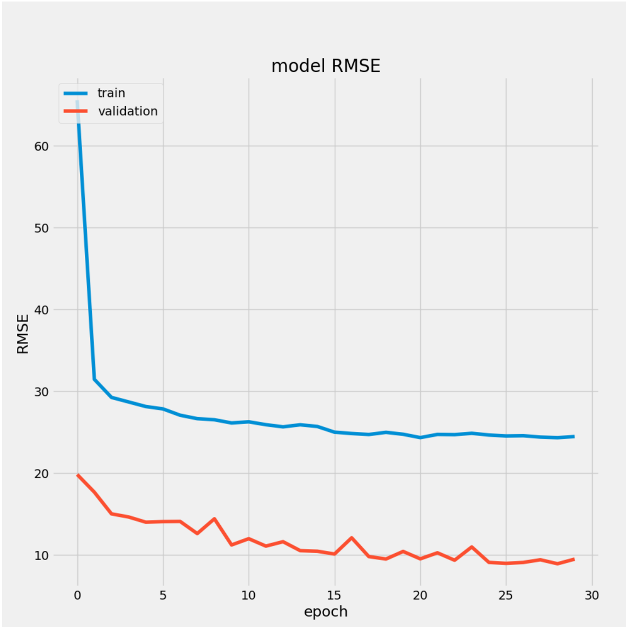

Figure 21: Root Mean Square Error for FD003 dataset without ResCNN.

Removing the Temporal Convolutional Network (TCN) from the RCBLA architecture resulted in a noticeable decline in predictive performance, particularly in capturing long-range temporal dependencies. As shown in Table 9, the RMSE increased to 15.23 on FD001 and 17.38 on FD003, while the S-score rose to 290.98 and 613.58 respectively. These metrics indicate that the absence of TCN limits the model's ability to represent degradation patterns that evolve over time.

The RMSE and MAE curves for FD003 in Figure 24 and Figure 25 further illustrate this effect. The validation error fluctuates more compared to the full model, and convergence is slower, suggesting that the model struggles to generalize without temporal dilation. Similarly, the loss curve in Figure 26 show higher final loss values and less stable training dynamics, reinforcing the role of TCN in optimizing the learning process.

Table 8: RMSE and Score results of different methods. (N/A: values not provided).

		FD001		FD003	
Methods	Year	RMSE	Score	RMSE	Score
SVR [22]	2016	20.96	1380	21.05	1600
RVR [22]	2016	23.80	1500	22.37	1430
CNN [22]	2016	18.45	1299	19.82	1600
LSTM [35]	2017	16.14	338	16.18	852
DCNN [16]	2018	**12.61**	**273.70**	12.64	284.10
CNN+LSTM [13]	2019	16.16	303	17.12	1420
Bi-LSTM-ED [32]	2019	14.74	**273.00**	17.48	574.00
Attention-LSTM [6]	2021	14.53	322.44	N/A	N/A
VAE-RNN [7]	2022	15.81	326	14.88	722
CNN-LSTM-Attention [8]	2024	15.977	N/A	13.907	N/A
CNN-GRU [3]	2025	16.29	N/A	23.71	N/A
STCAN [31]	2025	**14.24**	460.8	15.12	543.7
CAELSTM [9]	2025	14.44	282.38	13.400	**264.47**
RCBLA	-	**14.38**	**279.23**	**12.46**	**282.56**

Table 9: Impact of Component Removal on RUL Prediction Accuracy: RMSE and S-Score Across FD001 and FD003

	FD001		FD003	
	RMSE	**Score**	**RMSE**	**Score**
Full Model RCBLA	**14.38**	**279.23**	**12.46**	**282.56**
Without ResCNN	15.74	331.55	18.54	757.58
Without TCN	15.23	290.98	17.38	613.58
Without BiLSTM	14.41	**266.16**	17.12	590.15

Figure 22: Model MAE for FD003 dataset without ResCNN.

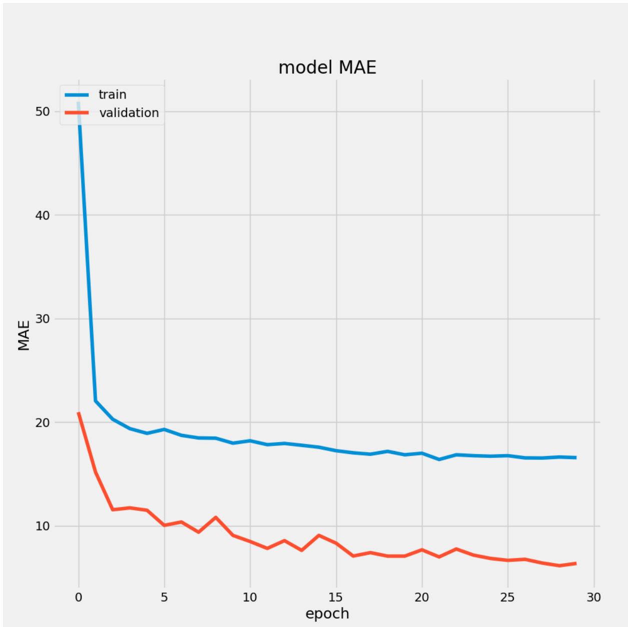

Figure 23: Model MAE for FD001 dataset without ResCNN.

Overall, the results confirm that the TCN layer is essential for modeling sequential sensor data, as it enables the network to capture multi-scale temporal features that are critical for accurate Remaining Useful Life (RUL) prediction.

Removing the Bidirectional LSTM (BiLSTM) from the RCBLA architecture led to mixed performance outcomes across the FD001 and FD003 subsets. As shown in Table 9, the FD001 results remained relatively strong, with an RMSE of 14.41 and the lowest

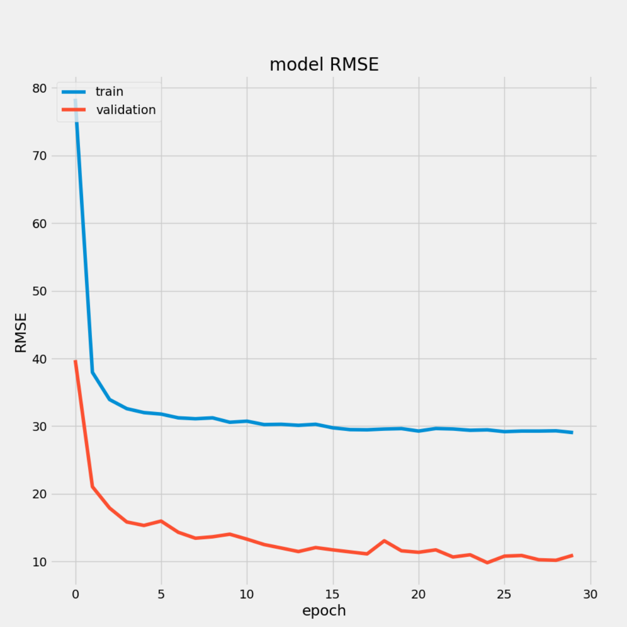

Figure 24: Root Mean Square Error for FD003 dataset without TCN.

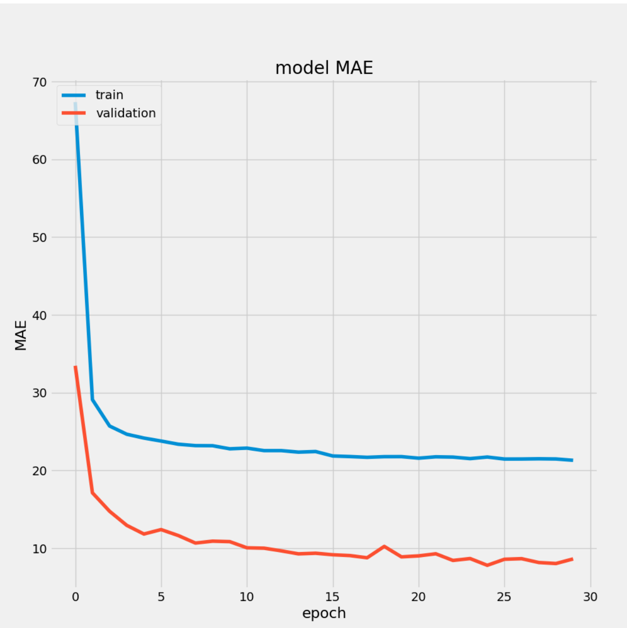

Figure 25: Model MAE for FD003 dataset without TCN.

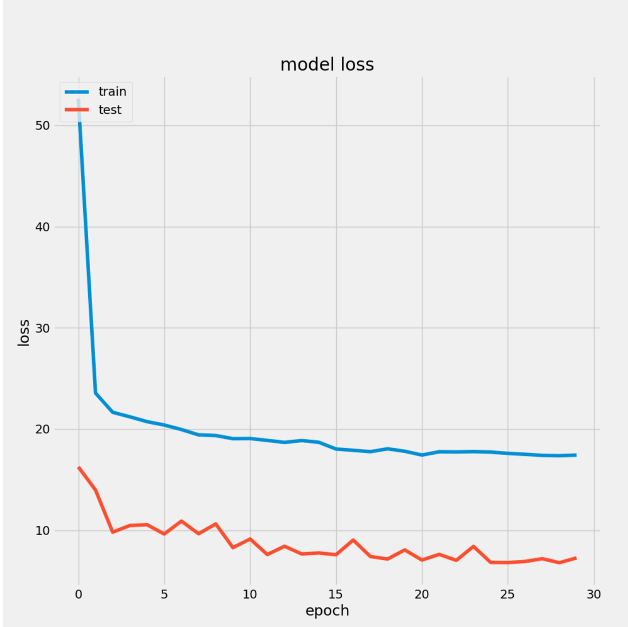

Figure 26: Loss for FD003 dataset without TCN.

S-score of 266.16 among all ablated variants. This suggests that for simpler degradation patterns, the model can maintain reasonable accuracy even without bidirectional temporal modeling.

However, the FD003 subset revealed a clear decline in generalization capability. The RMSE increased to 17.12 and the S-score rose to 590.15, indicating that the absence of BiLSTM hindered the model's ability to capture complex temporal dependencies. This is further supported by the RMSE and MAE curves in Figure 27 and Figure 28, which show slower convergence and persistent gaps between training and validation errors. The validation RMSE stabilizes at a higher level compared to the full model, and the MAE curve reveals fluctuating error behavior, suggesting reduced consistency in predictions.

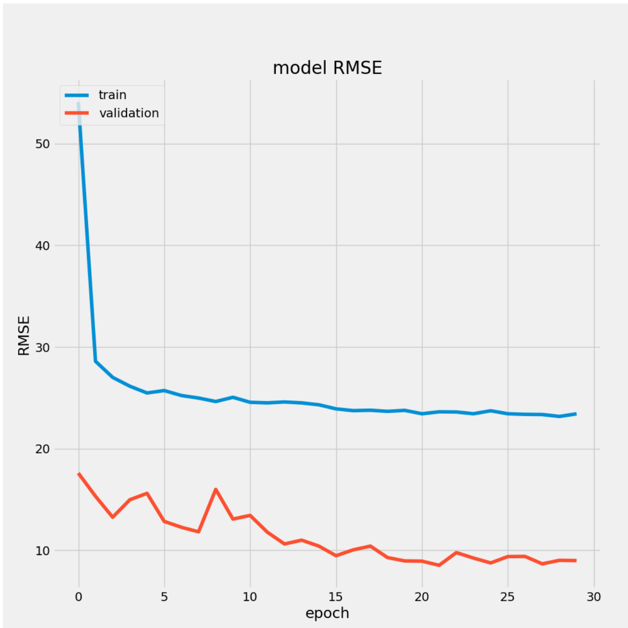

Figure 27: Root Mean Square Error for FD003 dataset without BiLSTM.

The loss curve in Figure 29 reinforces this observation. Although the training loss decreases steadily, the testing loss plateaus early and remains elevated, indicating that the model struggles to generalize beyond the training data. These patterns highlight the role of BiLSTM in enhancing temporal context understanding, especially in datasets with non-linear and multiphase degradation like FD003.

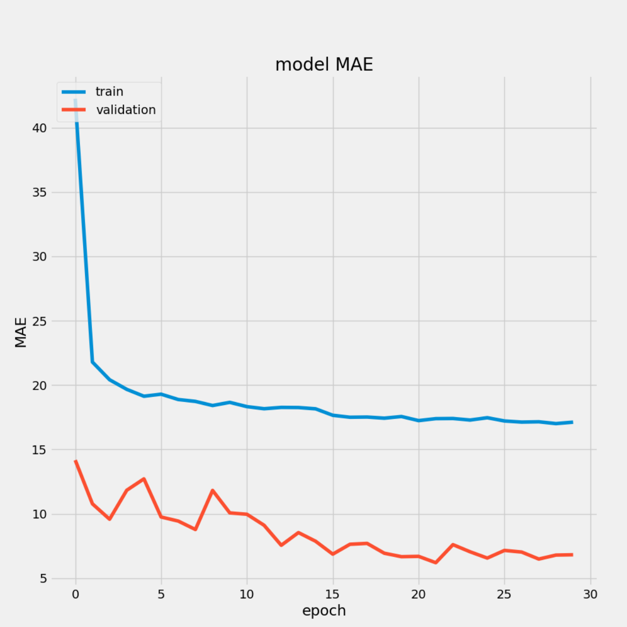

Figure 28: Model MAE for FD003 dataset without BiLSTM.

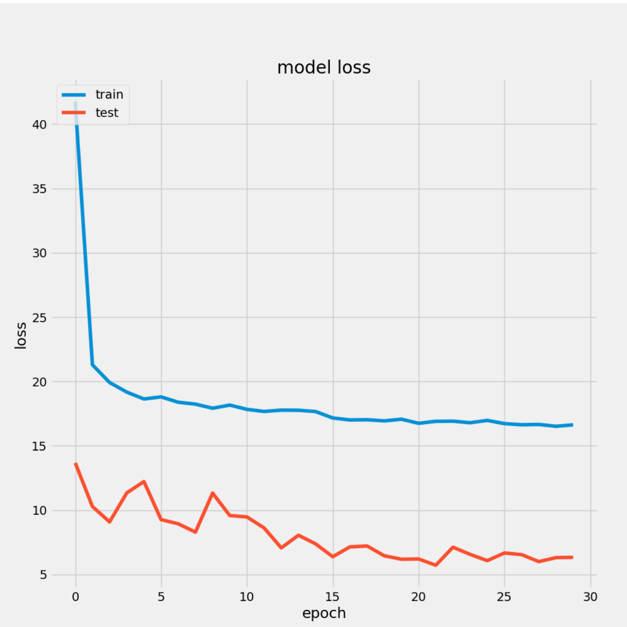

Figure 29: Loss for FD003 dataset without BiLSTM.

In summary, while the BiLSTM component may be less critical for simpler cases (FD001), it proves essential for robust performance in more complex scenarios, confirming its value in capturing bidirectional temporal dynamics.

Finally, this ablation study demonstrates that each architectural component within the RCBLA framework plays a distinct and complementary role in enhancing predictive accuracy and generalization. The observed performance degradation upon removal of individual modules confirms the necessity of their integration, validating the robustness and design rationale of the proposed hybrid architecture.

6 Conclusion

Predicting the Remaining Useful Life (RUL) of turbofan engines plays a crucial role in prognostics and health management (PHM), where accurate estimation enables proactive maintenance planning and ensures operational safety. This study introduced a hybrid deep learning framework, termed RCBLA (Residual Convolutional Bidirectional LSTM with Attention), specifically designed to exploit the sequential and temporal dependencies inherent in aero-engine sensor data. The model integrates residual convolutional layers for effective feature extraction, Temporal Convolutional Networks (TCN) to capture long-range dependencies, Bidirectional LSTMs for contextual learning across both past and future states, and an attention mechanism to prioritize informative time steps. The inclusion of layer normalization further enhanced convergence stability and reduced training variance.

The experimental evaluation on the C-MAPSS dataset, focusing on FD001 and FD003 subsets, confirms the robustness of the proposed architecture. With optimized hyperparameters, including a custom learning rate scheduler and dropout rates of 0.7 and 0.5, RCBLA achieved RMSE values of 14.389 (FD001) and 12.46 (FD003), while the corresponding score values were 279.23 and 282.56, respectively. These results demonstrate that RCBLA outperforms or matches several recent state-of-the-art methods, validating its effectiveness in learning degradation patterns and predicting RUL with high accuracy.

From a practical perspective, the proposed model has promising applications in aerospace PHM systems, where it could be integrated into maintenance platforms to analyze engine sensor data in real time, anticipate component degradation, and optimize maintenance schedules. Such predictive capability would minimize unplanned downtime, reduce costs, and improve operational safety. Furthermore, the approach is extendable to broader IoT-enabled predictive maintenance scenarios, where heterogeneous sensor data from industrial machinery can be processed to enhance reliability and decision-making.

For future research, several directions can be explored. First, extending the model evaluation to the FD002 and FD004 subsets, which involve more complex operating conditions and fault modes, would provide further validation of its generalization capabilities. Second, incorporating data augmentation techniques to simulate diverse degradation trajectories could improve robustness and reduce sensitivity to limited training samples. Third, leveraging domain adaptation strategies would enable better transferability of the model to real-world datasets with different operating conditions. Additionally, uncertainty quantification techniques such as Bayesian deep learning or Monte Carlo Dropout could be integrated to provide confidence intervals for RUL predictions, increasing reliability in safety-critical aerospace applications. Finally, exploring graph-based neural networks to explicitly model the correlations among sensor signals may uncover hidden spatial-temporal dependencies that traditional sequential architectures may overlook.

Overall, the proposed RCBLA framework demonstrates strong potential for advancing predictive maintenance in aerospace systems and lays the groundwork for future enhancements in the broader PHM domain.

7 Conflicts of interest

The authors declare that they have no conflicts of interest.

8 Data Availability

The data set used during the current study was provided by the NASA Ames Prognostics Center of Excellence (PCoE). It is a public data set and can be downloaded from the following website https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/ and follow the following link to download the desired data set used in the current study https://data.nasa.gov/Aeorspace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6. (accessed 15 June 2023)

References

[1] Al Bataineh, A., Mairaj, A., and Kaur, D. Autoencoder based semi-supervised anomaly detection in turbofan engines. *International Journal of Advanced Computer Science and Applications 11*, 11 (2020).

[2] Asif, O., Haider, S. A., Naqvi, S. R., Zaki, J. F., Kwak, K.-S., and Islam, S. R. A deep learning model for remaining useful life prediction of aircraft turbofan engine on c-mapss dataset. *Ieee Access 10* (2022), 95425–95440.

[3] Azyus, A. F., Wijaya, S. K., and Kurniawan, B. Remaining useful life prediction of turbofan engines using cnn-gru: A comparative study on c-mapss dataset. *Journal of Mechanical, Civil and Industrial Engineering 6*, 1 (2025), 19–27.

[4] Chang, C.-C., and Lin, C.-J. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST) 2*, 3 (2011), 1–27.

[5] Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., Zhou, J., and Wang, G. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging 36*, 12 (2017), 2524–2535.

[6] Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., and Li, X. Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics 68*, 3 (2020), 2521–2531.

[7] Costa, N., and Sánchez, L. Variational encoding approach for interpretable assessment of remaining useful life estimation. *Reliability Engineering & System Safety 222* (2022), 108353.

[8] Deng, S., and Zhou, J. Prediction of remaining useful life of aero-engines based on cnn-lstm-attention. *International Journal of Computational Intelligence Systems 17*, 1 (2024), 232.

[9] Elsherif, S. M., Hafiz, B., Makhlouf, M., and Farouk, O. A deep learning-based prognostic approach for predicting turbofan engine degradation and remaining useful life. *Scientific Reports 15*, 1 (2025), 26251.

[10] Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation 18*, 7 (2006), 1527–1554.

[11] Hu, T., Guo, Y., Gu, L., Zhou, Y., Zhang, Z., and Zhou, Z. Remaining useful life prediction of bearings under different working conditions using a deep feature disentanglement based transfer learning method. *Reliability Engineering & System Safety 219* (2022), 108265.

[12] Isbilen, F., Bektas, O., and Konar, M. Deep learning and similarity-based models for predicting turbofan engine remaining useful life: insights from the cmapss dataset. *The Aeronautical Journal 129*, 1337 (2025), 2004–2035.

[13] Kong, Z., Cui, Y., Xia, Z., and Lv, H. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Applied Sciences 9*, 19 (2019), 4156.

[14] Li, J., Jia, Y., Niu, M., Zhu, W., and Meng, F. Remaining useful life prediction of turbofan engines using cnn-lstm-sam approach. *IEEE Sensors Journal 23*, 9 (2023), 10241–10251.

[15] Li, P., Zhang, Z., Grosu, R., Deng, Z., Hou, J., Rong, Y., and Wu, R. An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries. *Renewable and Sustainable Energy Reviews 156* (2022), 111843.

[16] Li, X., Ding, Q., and Sun, J.-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety 172* (2018), 1–11.

[17] Li, Z., Luo, S., Liu, H., Tang, C., and Miao, J. Ttsnet: Transformer–temporal convolutional network–self-attention with feature fusion for prediction of remaining useful life of aircraft engines. *Sensors 25*, 2 (2025), 432.

[18] Lin, R., Wang, H., Xiong, M., Hou, Z., and Che, C. Attention-based gate recurrent unit for remaining useful life prediction in prognostics. *Applied Soft Computing 143* (2023), 110419.

[19] Liu, L., Song, X., and Zhou, Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety 221* (2022), 108330.

[20] Liu, Z., Zheng, X., Xue, A., and Ge, M. Multi-scale temporal-spatial feature-based hybrid deep neural network for remaining useful life prediction of aero-engine. *ACS omega 9*, 48 (2024), 47410–47427.

[21] Markou, C., and Cros, G. Airline maintenance cost executive commentary fy2019 data public version. *Maintenance Cost Technical Group. IATA* (2020).

[22] Sateesh Babu, G., Zhao, P., and Li, X.-L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, April 16-19, 2016, Proceedings, Part I 21* (2016), Springer, pp. 214–228.

[23] Saxena, A., Goebel, K., Simon, D., and Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (2008), IEEE, pp. 1–9.

[24] Sherifi, A. Turbofan engine remaining useful life (rul) prediction based on bi-directional long short-term memory (blstm). *arXiv preprint arXiv:2411.16422* (2024).

[25] Sun, J., Zheng, L., Huang, Y., and Ge, Y. Remaining useful life prediction based on cnn-bgru-sa. In *Journal of Physics: Conference Series* (2022), vol. 2405, IOP Publishing, p. 012007.

[26] Tipping, M. The relevance vector machine. *Advances in neural information processing systems 12* (1999).

[27] Wang, L., Chen, Y., Zhao, X., and Xiang, J. Predictive maintenance scheduling for aircraft engines based on remaining useful life prediction. *IEEE Internet of Things Journal 11*, 13 (2024), 23020–23031.

[28] Wang, T., Guo, D., and Sun, X.-M. Remaining useful life predictions for turbofan engine degradation based on concurrent semi-supervised model. *Neural Computing and Applications 34*, 7 (2022), 5151–5160.

[29] Wang, W., Peng, X., and Tong, T. Robust and efficient mediation analysis via huber loss. *Psychometrika*, 1–23.

[30] Wu, J., Hu, K., Cheng, Y., Zhu, H., Shao, X., and Wang, Y. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. *ISA transactions 97* (2020), 241–250.

[31] Yang, X., Tang, L., and Huang, J. Remaining useful life prediction for multi-sensor mechanical equipment based on self-attention mechanism network incorporating spatio-temporal convolution. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 239*, 2 (2025), 315–329.

[32] Yu, W., Kim, I. Y., and Mechefske, C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing 129* (2019), 764–780.

[33] Zhang, C., Lim, P., Qin, A. K., and Tan, K. C. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE transactions on neural networks and learning systems 28*, 10 (2016), 2306–2318.

[34] Zhang, Q., Liu, Q., and Ye, Q. An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine. *Engineering Applications of Artificial Intelligence 127* (2024), 107241.

[35] Zheng, S., Ristovski, K., Farahat, A., and Gupta, C. Long short-term memory network for remaining useful life estimation. In *2017 IEEE international conference on prognostics and health management (ICPHM)* (2017), IEEE, pp. 88–95.

A Additional Figures

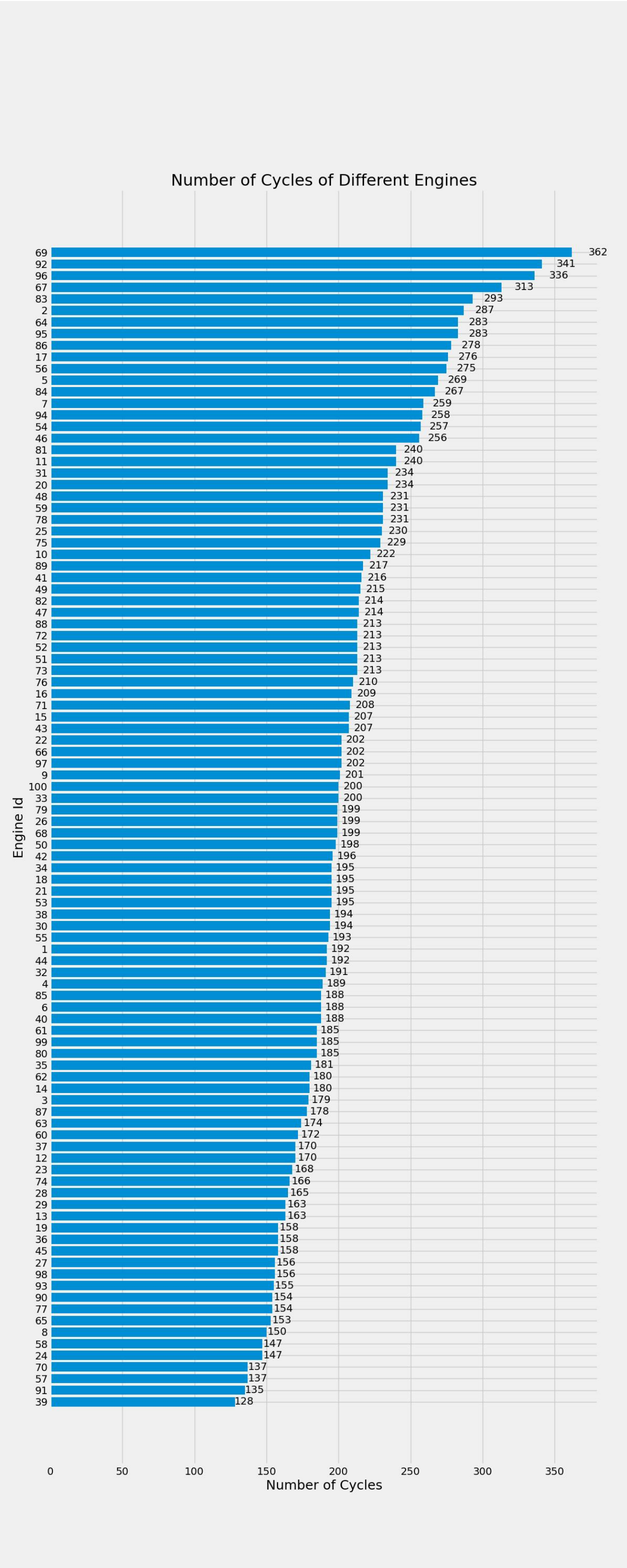

Figure 30: Number of cycles across different engines in FD001.

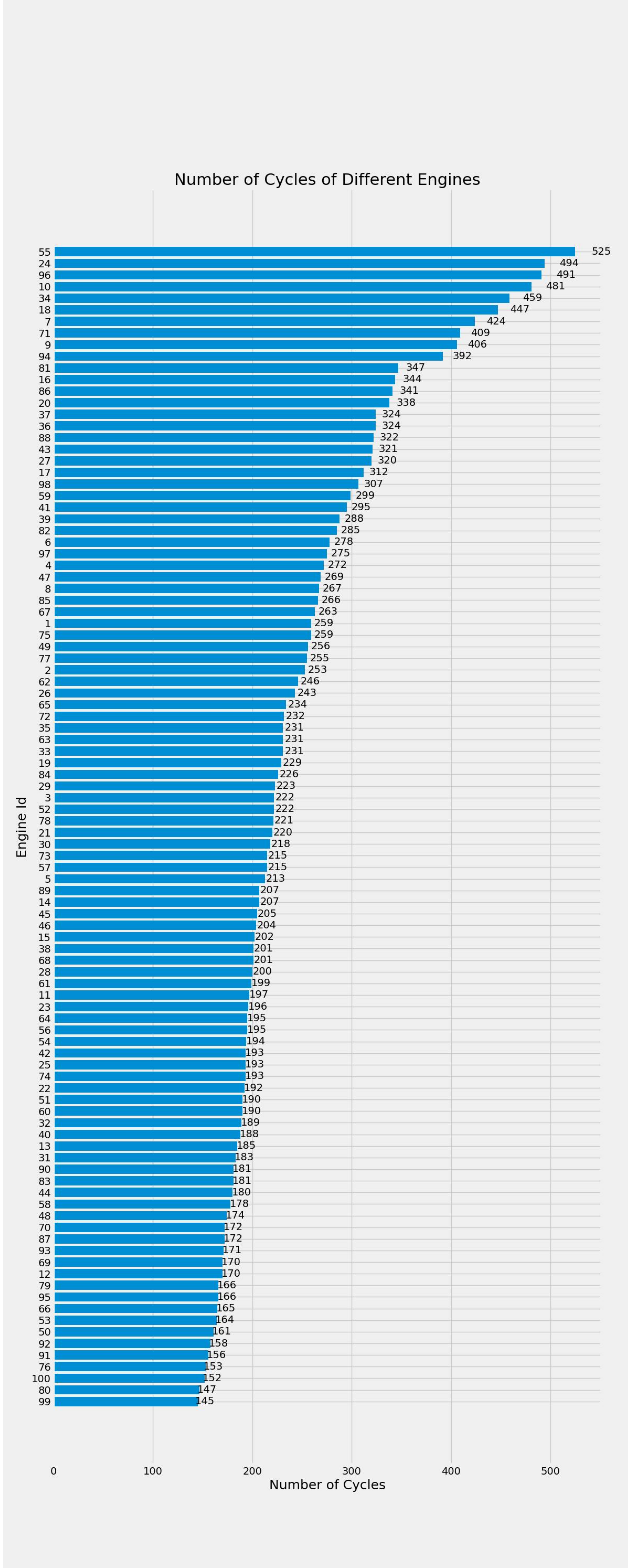

Figure 31: Number of cycles across different engines in FD003.