

LSTM-Based Predictive Maintenance Framework for Turbofan Engines Using C-MAPSS Data

Mohamed Badiy¹, Charaf Hamidi³, Fatima Amounas², Mourade Azrou^{2,*}, Salma Bendaoud²

¹IRIA laboratory, IASS Team, National School of Artificial Intelligence and Digital, Mohammed Premier University of Oujda, Morocco

²IMIA laboratory, MSIA Team, Faculty of Sciences and Techniques, Errachidia, Moulay Ismail University of Meknes, Morocco

³Laboratory of Engineering Sciences, University Ibn Zohr, Faculty of Science Agadir, Agadir, Morocco

Abstract In the modern industrial world, predictive maintenance is an emerging strategy to address the challenges of unexpected equipment failures, high maintenance costs, and production downtime. However, existing studies often focus primarily on prediction accuracy without providing an integrated framework that combines IoT-enabled sensor data with advanced deep learning models for Remaining Useful Life (RUL) estimation. To address this gap, this study proposes an end-to-end LSTM-based predictive maintenance framework for turbofan engines using the C-MAPSS dataset. The proposed approach leverages IoT-generated sensor measurements and Long Short-Term Memory (LSTM) networks to capture temporal degradation patterns and predict engine RUL. The performance of the LSTM model is evaluated against Gated Recurrent Unit (GRU) and traditional Recurrent Neural Network (RNN) models using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Experimental results demonstrate that the proposed LSTM framework achieves superior predictive performance, attaining an MAE of 27.11 and an RMSE of 35.12, outperforming the benchmark recurrent architectures. These results highlight the effectiveness of LSTM networks in modeling complex sequential sensor data and detecting early signs of equipment degradation. The integration of IoT-enabled monitoring with deep learning-based RUL prediction provides a practical solution for real-time predictive maintenance. This approach supports timely decision-making by continuously analyzing sensor data and anticipating equipment failures before they occur.

Keywords Predictive maintenance, Remaining Useful Life, C-MAPSS, Long Short-Term Memory, Deep learning, Turbofan engines, Sensor data.

DOI: 10.19139/soic-2310-5070-4835

1. Introduction

Recent advancements in Industry 4.0 have sparked a revolutionary transformation in traditional manufacturing methods, signifying a major shift away from conventional production models. This new era emphasizes digitalization and automation, integrating cutting-edge technologies such as Big Data, the Internet of Things (IoT), and cyber-physical systems (CPS) [1, 2]. These disruptive innovations empower companies to transition toward smart factories, highly interconnected systems where machines, components, and humans seamlessly collaborate in real-time. This interconnect-edness enables the optimization of production processes, fosters intelligent decision-making, and drives cognitive production systems. A defining feature of this revolution is the exponential growth of data generated across industrial ecosystems [3, 4]. Data streams originate from a variety of sources, including sensor readings, production lines, machine operations, and human inter-actions. This unprecedented influx of information positions data as a cornerstone of value creation, offering opportunities to enhance operational efficiency, improve decision-making, and foster innovation [5]. However, to fully leverage this

*Correspondence to: Mourade Azrou (Email: mo.azrou @umi.ac.ma). Faculty of Sciences and Techniques, Errachidia, Moulay Ismail University of Meknes, Morocco

data-rich environment, organizations must deploy advanced analytical frameworks capable of transforming raw data into actionable insights. This necessitates the adoption of state-of-the-art technologies and methodologies tailored to handle the complexity, scale, and dynamism of industrial data. Among these technologies, Deep Learning (DL), a specialized subfield of Artificial Intelligence (AI) has emerged as a transformative tool, redefining how data is processed and interpreted [6]. Unlike traditional Machine Learning (ML) methods [7, 8, 9], DL architectures can autonomously learn high-level abstractions and intricate patterns directly from raw, high-dimensional data [10, 11]. This ability makes DL particularly well-suited for industrial applications, where datasets are often complex, heterogeneous, and temporal in nature [12, 13]. One of the most promising applications of DL in Industry 4.0 is Predictive Maintenance (PdM), a proactive strategy designed to identify and mitigate potential equipment failures before they occur [14, 15, 16, 17]. PdM leverages both historical and real-time sensor data to predict failures, optimize maintenance schedules, and ensure uninterrupted operations. The benefits of PdM are multifaceted, encompassing cost reduction, extended equipment lifespans, enhanced safety standards, and improved product quality. To comprehend PdM, it is important to first define maintenance, which encompasses all techniques used to restore and preserve a system's ability to perform its required functions. For instance, Figure 1 presents a failure diagram for PdM that factors in both time and equipment condition. Maintenance can be divided into four primary types. The first type involves replacing parts only after they fail. The second type is scheduled periodic replacement, where parts are replaced according to failure history, improving the overall efficiency of maintenance. However, replacing functional parts unnecessarily leads to resource waste. The third type is proactive maintenance, which focuses on eliminating defective components to enhance facility performance. The final type involves analyzing equipment data to detect and address issues before failure occurs. This method, by addressing problems like excessive periodic replacements and inaccurate analyses, offers advantages such as reducing facility costs, improving reliability, and extending their lifespan.

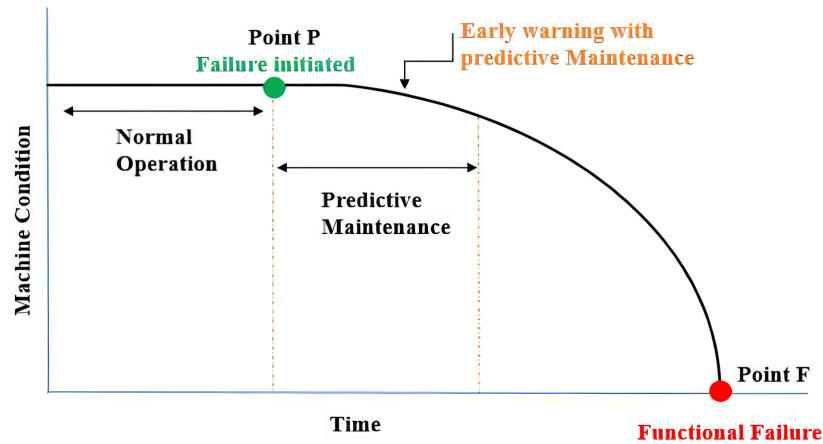


Figure 1. Predictive maintenance over time and Equipment conditions

Despite the widespread adoption of maintenance strategies in industrial environments, traditional approaches such as corrective and preventive maintenance remain inefficient for complex systems operating under dynamic conditions [18, 19, 20, 21, 22]. Corrective maintenance addresses failures only after they occur, often resulting in unplanned downtime, production losses, and increased repair costs. Preventive maintenance, although reducing the risk of unexpected failures through scheduled servicing, frequently leads to the replacement of components that still have useful operational life, thereby increasing maintenance costs and resource consumption [23, 24]. These limitations have accelerated the shift toward predictive maintenance, which exploits machine learning and deep learning algorithms to analyze equipment health and estimate future failures. In the context of PdM, DL frameworks leverage a variety of architectures tailored to specific analytical needs. Recurrent Neural Networks (RNNs), and particularly their advanced variants like Long Short-Term Memory (LSTM) networks, excel in modeling temporal sequences, making them ideal for analyzing time-series data generated by industrial sensors. Convolutional Neural

Networks (CNNs), meanwhile, are used for visual diagnostics, such as analyzing equipment imagery for wear or defects. Additionally, emerging architectures like Transformers have demonstrated potential in multimodal data integration, combining information from diverse sources to provide comprehensive insights into system health. These models surpass traditional methods by autonomously learning hierarchical features, thereby uncovering subtle patterns indicative of degradation or failure. Despite their advantages, the implementation of DL-based PdM systems involves addressing several critical challenges. First, the quality, consistency, and relevance of data are paramount. Industrial datasets must be carefully preprocessed to ensure they are complete and aligned with maintenance objectives. Second, achieving optimal model performance requires balancing predictive accuracy with computational efficiency, particularly in real-time applications where rapid decision-making is essential. Lastly, practical considerations such as hardware infrastructure, data storage capacity, and integration with existing physical assets must be addressed to ensure scalability and cost-effectiveness. Within this framework, the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset serves as a benchmark for developing and validating predictive maintenance models. Designed for prognostics and health management tasks, C-MAPSS provides rich time-series data from aeroengine simulations, encompassing multiple operational conditions and fault scenarios. LSTM networks, in particular, are well-suited for analyzing C-MAPSS data, as they excel in capturing long-term dependencies and modeling complex temporal relationships. By leveraging LSTM-based frameworks, it becomes possible to predict Remaining Useful Life (RUL) with high precision, enabling proactive interventions that minimize downtime and operational disruptions.

This study presents a deep learning-based framework to assist companies, practitioners, and non-expert users in adopting predictive maintenance solutions. The proposed approach primarily exploits LSTM networks to effectively capture and model the complex temporal dependencies inherent in time-series data, particularly those encountered in the C-MAPSS turbofan engine dataset. The proposed framework integrates an efficient preprocessing pipeline, including data normalization, feature selection, and sequence generation, with an optimized LSTM model to enhance prediction accuracy and generalization capability. Furthermore, a comparative evaluation of recurrent deep learning architectures, namely LSTM, GRU, and RNN, is conducted to assess their effectiveness for turbofan engine prognostics. The main contributions that guide this work are:

- To design an effective LSTM-based predictive maintenance framework for RUL estimation using the C-MAPSS dataset and IoT-enabled sensor data.
- To optimize and tune key hyperparameters to improve prediction accuracy and model generalization.
- To conduct a comparative evaluation of deep learning architectures (LSTM, GRU, and RNN) for turbofan engine RUL prediction using the C-MAPSS dataset.
- To assess model accuracy and predictive performance by systematically comparing predicted RUL values with actual ground-truth RUL values.

The rest of this article is structured as follows: Section 2 reviews the related works on RUL prediction, with particular emphasis on LSTM-based and other deep learning approaches for predictive maintenance. Section 3 presents the research methodology and introduces the proposed LSTM-based predictive maintenance framework, covering the C-MAPSS dataset and feature set (section 3.1), the preprocessing pipeline (section 3.2), the data analytics (3.3), the LSTM architecture for RUL prediction (section 3.4), and the evaluation metrics used (section 3.5). Section 4 presents the experimental setup (section 4.1) along with the results and comparative analysis against GRU and RNN models (section 4.2). Section 5 concludes the article by summarizing the main findings, emphasizing the study's contributions, and proposing directions for future research to advance the field further.

2. Related Works

Predictive maintenance is a key technology in smart factories [25, 26, 27]. It is designed to detect and forecast equipment failures and defects through the analysis of collected operational data, thereby enabling more efficient maintenance management and reducing unplanned downtime [28, 29, 30, 31]. Predictive maintenance has made significant progress, driven by advancements in deep learning, LSTM networks, attention mechanisms, and hybrid architectures. Motivated by the growing demand for accurate Remaining Useful Life (RUL) estimation,

numerous studies have investigated LSTM-based architectures for predictive maintenance. For instance, Wu, Rui, and Jie Ma in [32] proposed an enhanced LSTM neural network that integrates uncertainty estimation to improve RUL predictions. By addressing the uncertainty in model outputs, their approach enhances both the accuracy and reliability of RUL estimations. Similarly, Bruneo, Dario, and Fabrizio De Vita in [33] presented a machine learning framework leveraging LSTM networks for predictive maintenance in smart industries. Their work demonstrated that LSTMs can effectively analyze historical system data to forecast RUL, while also introducing a method for optimizing LSTM hyperparameters. As a result, LSTMs have emerged as a viable solution for real-time monitoring and predictive maintenance in industrial applications. In a broader context, Jimenez et al. [34] explored the potential of artificial intelligence in predictive maintenance, emphasizing the industry's continued reliance on traditional maintenance methods. They argued that shifting to AI-driven PM could lead to significant efficiency gains and cost reductions. In support of this perspective, Daniyan et al. [35] demonstrated the application of artificial neural networks (ANNs) in estimating the RUL of wheel bearings, yielding cost savings and improved operational performance. Similarly, another study [36] highlighted the potential of LSTM autoencoders in accurately predicting equipment failures, thereby enabling optimized maintenance schedules that reduce costs and enhance uptime. Expanding on this, Matuszczak et al. [37] investigated the use of deep and machine learning models for predicting the condition of turbofan engine components. Notably, their findings highlighted the superior performance of an ensemble meta-model of neural networks compared to other ML approaches, with enhanced accuracy and reliability. Likewise, Owais Asif et al. [38] introduced a deep learning model that integrates efficient pre-processing steps with LSTM networks to refine RUL predictions for turbofan engines. By leveraging the C-MAPSS dataset, NASA's benchmark for turbofan jet engine degradation, their model achieves superior accuracy, as reflected in lower root mean squared error (RMSE) and score function values. Further extending the integration of AI in PdM, researchers in [39] proposed a hybrid framework that combines time series analysis (TSA) with deep learning techniques such as RNNs and LSTMs. Their findings suggested that TSA effectively uncovered temporal patterns and anomalies, while DL techniques enhanced prediction accuracy and efficiency. Moreover, a comparative study in [40] evaluated the performance of machine learning, deep learning, and hybrid models in fault detection. Interestingly, results indicated that ML models such as LightGBM and XGBoost outperform DL and DHL models in fault detection accuracy. Jaenal et al. [41] recently introduced a significant advancement in predictive maintenance with the development of MachNet, a deep learning architecture tailored for Industry 4.0 environments. Notably, MachNet demonstrates performance on par with or even surpassing state-of-the-art deep learning models like DCNN, BiLSTM, and Li-DAG, highlighting its effectiveness as a powerful solution for industrial predictive maintenance. Building on these advancements, Rath S et al. [42] proposed a deep learning framework for RUL prediction in turbofan engines. Their approach integrates a feature-selection algorithm alongside a logistic regression-based method for detecting change points. Leveraging a BiLSTM network, the model effectively captures both long and short-term dependencies in sensor data, further enhancing predictive maintenance capabilities.

Although some studies emphasize improvements in LSTM architectures and hyperparameter optimization, others focus on advanced deep learning or ensemble models to enhance prediction accuracy. However, comparatively less attention has been devoted to developing integrated frameworks that simultaneously incorporate efficient preprocessing, lightweight feature selection, and comprehensive performance evaluation. To address these challenges, this work introduces a novel framework for RUL estimation in predictive maintenance applications using the C-MAPSS turbofan engine dataset and IoT-enabled sensor data.

3. Research Methodology

In this section, we outline our proposed approach for predicting the Remaining Useful Life (RUL) of industrial equipment. The methodology is structured into six main steps. First, data collection involves gathering relevant data such as sensor readings, operational metrics, and maintenance logs, with the C-MAPSS dataset serving as a common source. Second, data processing is conducted to prepare the raw data, which includes transforming it into usable formats and applying normalization techniques to ensure consistency across features. Third, feature

selection focuses on identifying the most significant variables, such as temperature, pressure, and vibration, to enhance model effectiveness. Fourth, in the model development phase, the dataset is split into training and testing subsets. Predictive models, particularly LSTM networks, are trained on the training data and validated using the testing data to evaluate their generalization performance. Fifth, the forecasted RUL and Comparison step involves using the trained model to predict RUL values, which are then compared to the actual RUL for accuracy assessment. Lastly, performance evaluation is carried out by calculating key metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), aiming to minimize these metrics to ensure precise and reliable RUL predictions. The methodology to predict the Remaining Useful Life is described in Figure 2.

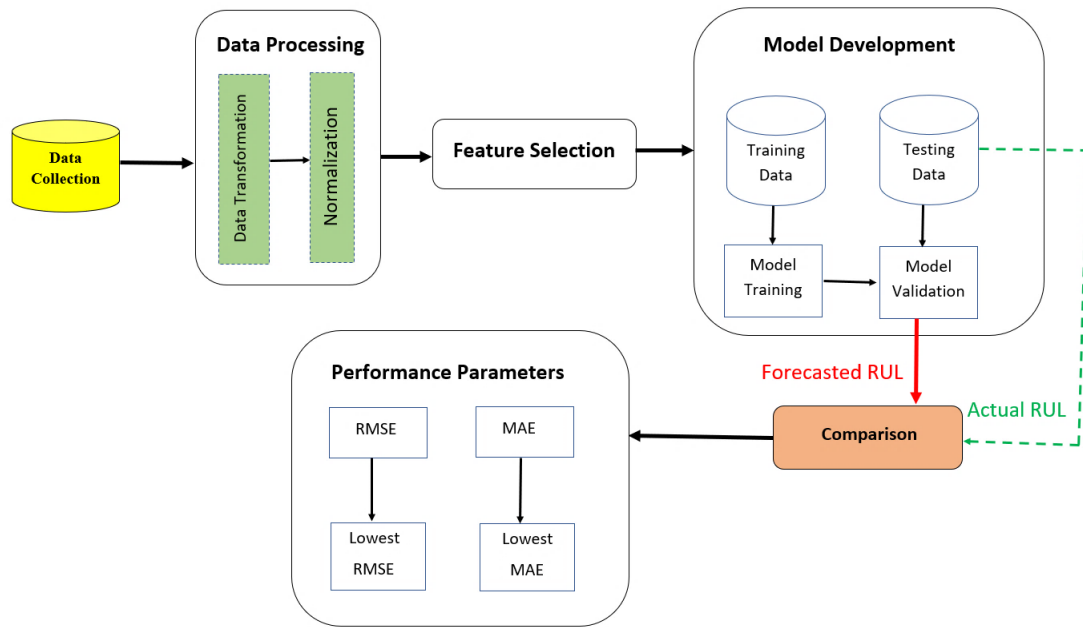


Figure 2. Proposed Approach.

The proposed RUL prediction framework is implemented through five sequential stages, as illustrated in Figure 2 and described in the followings subsections.

3.1. Dataset and Features

Our study employs data from the NASA turbofan dataset [43], which was produced using a turbofan simulation model built on the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) framework [44]. It is an open-access dataset and is widely used in the fields of diagnostics and prognostics. Each simulation run captures the complete run-to-failure operations of an engine, beginning with a brand-new condition, progressing through fault development across multiple flights, and culminating in failure as defined by a set of predefined criteria. Each trajectory represents an engine's cycle records, with every cycle record being a snapshot of data collected during a single operational cycle. In the C-MAPSS dataset, a single cycle is represented as a 24-dimensional feature vector, comprising 3 operational settings and 21 sensor measurements per flight cycle. NASA provides four subsets of this dataset, labeled FD001 to FD004, each reflecting different operational conditions and fault scenarios, such as variations in engine settings and potential failure modes. In this study, we focus on the FD001 subset, conducting all experiments exclusively on this data. A summary of the dataset is presented in Table 1. The FD001 dataset consists of multiple multivariate time series and is split into training and test sets. The training subset comprises 21 sensor measurements, as outlined in Table 2, along with 3 operational parameters, the number of cycles until

Table 1. Summary of the FD001 Dataset

Dataset	FD001
Engine units (training)	100
Engine units (testing)	100
Operational conditions	1
Fault modes	1
Minimum Lifespans (training)	128
Maximum Lifespans (training)	362
Minimum cycles recorded (testing)	31
Maximum cycles recorded (testing)	303

failure, and a unique jet engine identifier. In Table 3, we provide a quick overview of all the predictor variables. The test data shares similar characteristics but concludes before the failure point. The primary objective is to predict the Remaining Useful Life of the engines using the information provided in the training data.

Table 2. Description of 21 Sensor Variables

ID	Symbol	Description	Units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC outlet	°R
3	T30	Total temperature at HPC outlet	°R
4	T50	Total temperature at LPT outlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	Epr	Engine pressure ratio (P50/P2)	-
11	Ps30	Static pressure at HPC outlet	psia
12	Phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass ratio	-
16	farB	Burner fuel–air ratio	-
17	htBleed	Bleed enthalpy	-
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

°R: Rankine temperature; psia: Pounds per square inch absolute; rpm: Revolutions per minute;
pps: Pounds per second; psi: Pounds per square inch; lbm/s: Pound mass per second.

3.2. Data Preprocessing

Preprocessing data is a crucial stage in deep learning that helps to enhance data quality and facilitates the extraction of valuable information from data [45, 46]. Every action used in data preparation is intended to improve the Deep Learning models performance. The details of these preprocessing operations are briefly described in this subsection.

Table 3. Snapshot of Dataset (20,631 rows $\times$ 26 columns)

Statistic	engineID	numCycle	opSetting1	opSetting2	opSetting3	... NRf
count	20631.000	20631.0000	20631.0000	20631.0000	20631.0	20631.00000
mean	51.506568	108.807862	-0.000009	0.000002	100.0	2388.096152
std	29.227633	68.880990	0.002187	0.000293	0.0	0.071919
min	1.0000	1.0000000	-0.008700	-0.000600	100.0	2387.880000
25%	26.0000	52.000000	-0.001500	-0.000200	100.0	2388.040000
50%	52.0000	104.00000	0.0000000	0.000000	100.0	2388.090000
75%	77.00000	156.00000	0.001500	0.000300	100.0	2388.140000
max	100.0000	362.00000	0.008700	0.000600	100.0	2388.560000

The training data have very different scales, which can cause problems as much larger values can have a greater impact on the models than smaller ones. It is therefore necessary to normalize the data. The datasets used in deep learning typically have varying orders of magnitude. Lower performance may result from this scale disparity. To address this issue, data preprocessing techniques such as feature scaling are applied. Feature scaling includes normalization and standardization, both of which help ensure that all features contribute equally to the learning process. Min-Max Scaling is particularly useful when feature values span different scales. By applying this transformation, features are mapped to a fixed range, typically [0,1], reducing variability and minimizing the influence of outliers. Normalization, a common scaling approach, can be effectively achieved using the Min-Max Scaling technique, as described in Algorithm 1.

Algorithm 1 Min-Max Scaling Algorithm

- 1: **Input:** $X = (A_1, \dots, A_n)$, where $1 < n < N$
- 2: **Output:** $X_{\text{normalized}} = (A_1^{\text{norm}}, \dots, A_n^{\text{norm}})$
- 3: **for** $i = 1$ to n **do**
- 4: **if** A_i is non-numeric **then**
- 5: Encode A_i using one-hot encoding
- 6: **end if**
- 7: Compute Min-Max Scaling:

$$A_i^{\text{norm}} = \frac{A_i - \min(A_i)}{\max(A_i) - \min(A_i)}$$

- 8: **end for**
-

After normalization, the dataset is organized into input variables and a target variable. The input data include 17 sensor-derived features extracted from the C-MAPSS dataset, while the target corresponds to the Remaining Useful Life. The dataset contains 20,631 observations, leading to an initial structure of (20631,17) for the input data and (20631,1) for the target values. Since Long Short-Term Memory (LSTM) networks are designed to process sequential data, the dataset is further transformed into a three-dimensional format. A sliding window approach is applied to generate sequences of fixed length. In this study, a sequence length of 20-time steps is selected as a trade-off between capturing temporal dependencies in engine degradation and maintaining computational efficiency. Concretely, for each index i , a sequence is constructed using observations from i to $i+20$. This transformation produces a data structure of size (20611,20,17), where 20611 represents the number of sequences, 20 denotes the number of time steps, and 17 corresponds to the number of features. The target variable is aligned by assigning the RUL value at the end of each sequence, resulting in a structure of (20611,1). To ensure reproducibility and prevent data leakage, the dataset is divided into training and validation subsets (e.g., 80% for training and 20% for validation), while preserving the temporal continuity of each engine unit. This guarantees that sequences from the same engine are not split across different subsets.

3.3. Data analytics

A. Feature selection

Feature selection plays a crucial role in machine learning, particularly in predictive maintenance tasks, as it directly impacts model performance, generalization ability, and computational efficiency. By eliminating irrelevant or redundant variables, the model becomes less prone to overfitting while also reducing training time and complexity. In this study, a statistical standard deviation–based method was employed to evaluate the variability of each feature across the dataset [47]. This approach is based on the assumption that features with extremely low or near-zero variance remain almost constant over time and therefore carry little to no discriminative information for predicting the target variable, namely the Remaining Useful Life. As illustrated in Figure 3, several features exhibit negligible variability. Specifically, the variables `opSetting3`, `T2`, `P2`, `epr`, `farB`, `Nf_dmd`, and `PCNfR_dmd` have standard deviation values equal or very close to zero. This indicates that these parameters do not change significantly across different engine cycles and operating conditions. Consequently, they do not contribute meaningful patterns for the learning process and may instead introduce unnecessary noise or redundancy. Additionally, some features such as `opSetting1` and `opSetting2` show very small but non-zero standard deviation values, suggesting limited variability. However, unlike the previously mentioned features, they are not strictly constant and may still contain marginal information, which justifies their retention at this stage. Based on this analysis, the low-variance features were removed from the dataset to reduce dimensionality and improve data quality. This preprocessing step ensures that the model focuses only on informative and dynamic features, thereby enhancing learning efficiency, improving convergence during training, and contributing to more robust and accurate predictions.

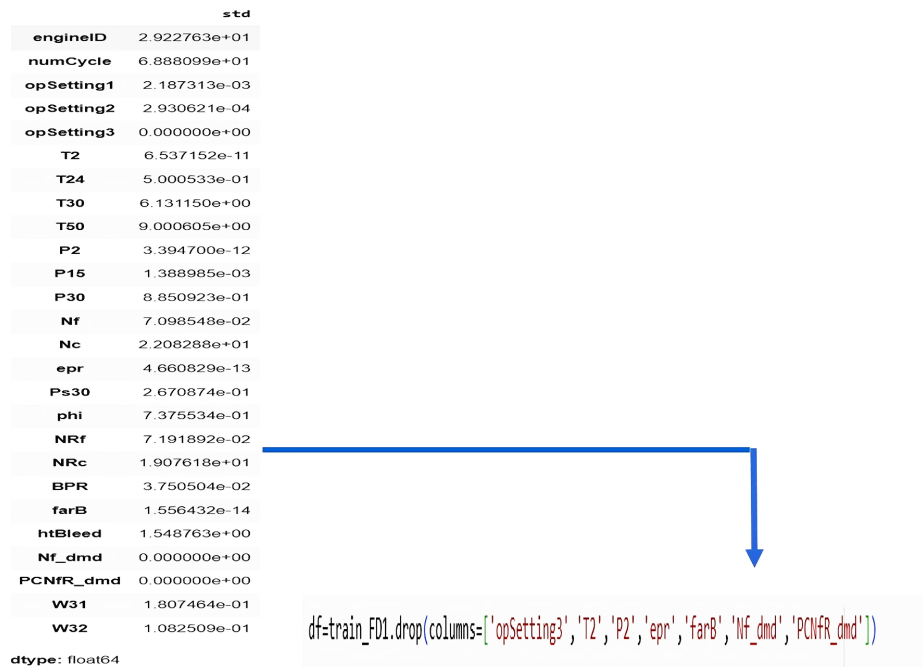


Figure 3. Remove non-useful features.

B. Correlation analysis

Data visualization is an essential step in uncovering hidden knowledge and identifying factors influencing target variables. In our case, where the objective is to analyze the remaining useful life (RUL) of turbofan engines, we began by creating a specific column dedicated to RUL. By subtracting the current number of cycles from the maximum number of cycles for each turbofan engine, we obtain the remaining operating cycles before the

component fails (equation (1)).

$$RUL = NumCycle_{max} - NumCycle_{actuel} \quad (1)$$

Linear correlation is a key statistical method for evaluating the impact of different features on the target variable (RUL). It measures both the strength and direction of the relationship between two variables. By applying this technique to the target column and other variables in the dataset, we can determine which factors have the most significant influence using Equation (2). To visualize the correlation matrix, we employed a heatmap. Figure 4 depicts the relationships between various variables and the strength of their associations. Our analysis of the correlation matrix revealed no signs of high multicollinearity among the variables, which is a fundamental assumption when developing predictive models.

$$r(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2)$$

Where:

r : correlation coefficient

n : number of observations

x_i and y_i : individual values of variables X and Y

$\bar{x}$ and $\bar{y}$: mean values of X and Y , respectively.

As shown in Figure 4, the variables T24, T30, T50, Nf, Ps30, BPR, and htBleed exhibit a negative correlation with RUL, whereas P30, W31, and W32 show a positive correlation. To better understand these relationships, we will visualize some of these features against RUL. Figure 5 illustrates every 10th turbofan engine along with its corresponding RUL and the features T24, T30, T50, Nf, Ps30, BPR, and htBleed, while Figure 6 presents every 10th turbofan engine with its RUL and the features P30, W31, and W32. The plotted figures reveal that, for most turbofan engines, the values of T24, T30, T50, Nf, Ps30, BPR, and htBleed tend to increase as the number of remaining cycles decreases (with the x-axes inverted for clarity). This trend suggests that as the engine nears the end of its operational life, these parameters rise, likely due to increasing thermal loads, declining efficiency, and mechanical wear affecting engine performance.

3.4. LSTM Model Development

The prediction of Remaining Useful Life is inherently a time-series forecasting problem, making LSTM networks an ideal choice. LSTM, an advanced variant of Recurrent Neural Networks (RNNs), is specifically designed to handle sequential data efficiently. Unlike traditional RNNs, which suffer from issues like vanishing gradients, LSTM networks incorporate a cell state and a gating mechanism to retain long-term dependencies, making them well-suited for predictive maintenance tasks.

Our proposed LSTM-based model follows a sequential architecture, as illustrated in Table 4, and consists of the following components:

- Two LSTM layers with 128 and 64 hidden units, respectively, responsible for capturing temporal dependencies in engine sensor data.
- Dropout layers between the LSTM layers and fully connected layers to prevent overfitting and enhance generalization.
- Two fully connected (Dense) layers, with the first layer having 16 neurons and the final output layer having a single neuron to produce the RUL prediction.
- The final regression layer outputs the estimated RUL of the turbofan engine.

Figure 7 illustrates a fundamental LSTM cell, which primarily comprises three control gates: the input gate, forget gate, and output gate. The core function of the LSTM is to transfer information from the previous cell state $C_{(t-1)}$ to the current cell state C_t , while selectively retaining relevant features during this process. The LSTM unit consists of an input gate i_t , a forget gate f_t , and an output gate o_t . The input gate i_t determines the extent to which the input at time t is retained, the forget gate f_t selectively removes information from certain memory

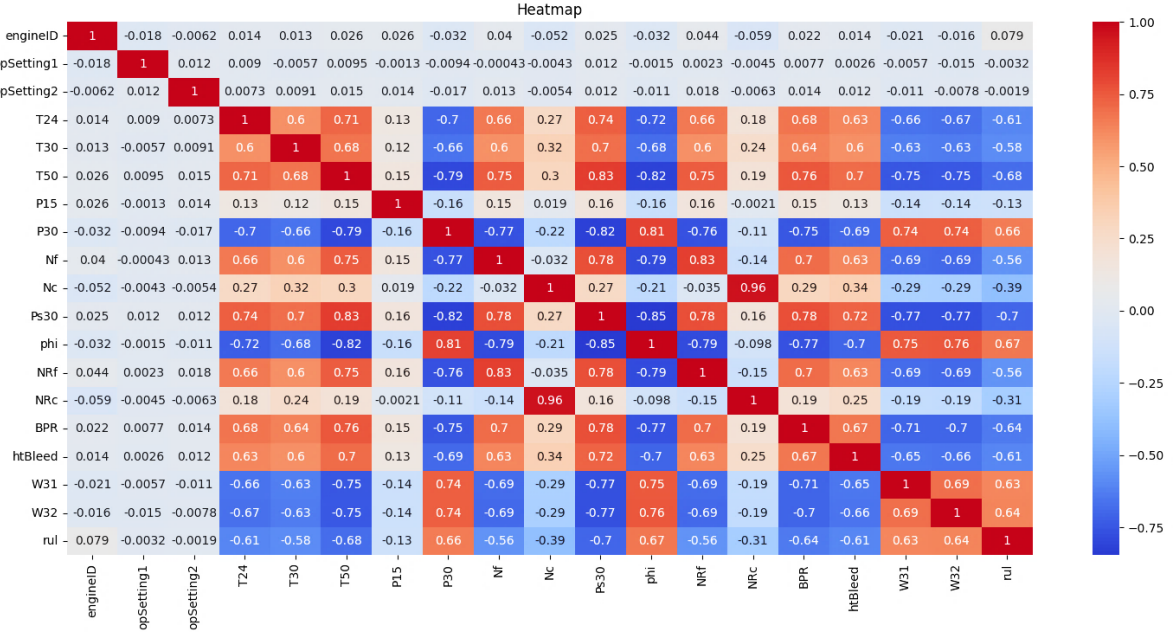


Figure 4. Correlation Matrix.

Table 4. Proposed LSTM Architecture

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 20, 128)	75,264
dropout (Dropout)	(None, 20, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 16)	1,040
dense_1 (Dense)	(None, 1)	17

cells, and the output gate o_t controls the output at time t based on the cell state C_t . Through the coordinated operation of these three gate mechanisms, the LSTM unit can effectively manage information selection, flow, and the preservation of long-term dependencies in the data. The mathematical expressions for these gates are provided in Equation (3).

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \end{cases} \quad (3)$$

Where i_t , f_t , and o_t represent the computed values of the input gate, forget gate, and output gate, respectively. Similarly, W_i , W_f , and W_o denote their corresponding weight matrices. The term $[h_{t-1}, x_t]$ represents the concatenation of the previous hidden state and the current input, while b_i , b_f , and b_o are the respective bias terms. The function $\sigma(\cdot)$ denotes the activation function. After t time steps, the candidate cell state $\tilde{C}$ is determined using Equation (4).

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

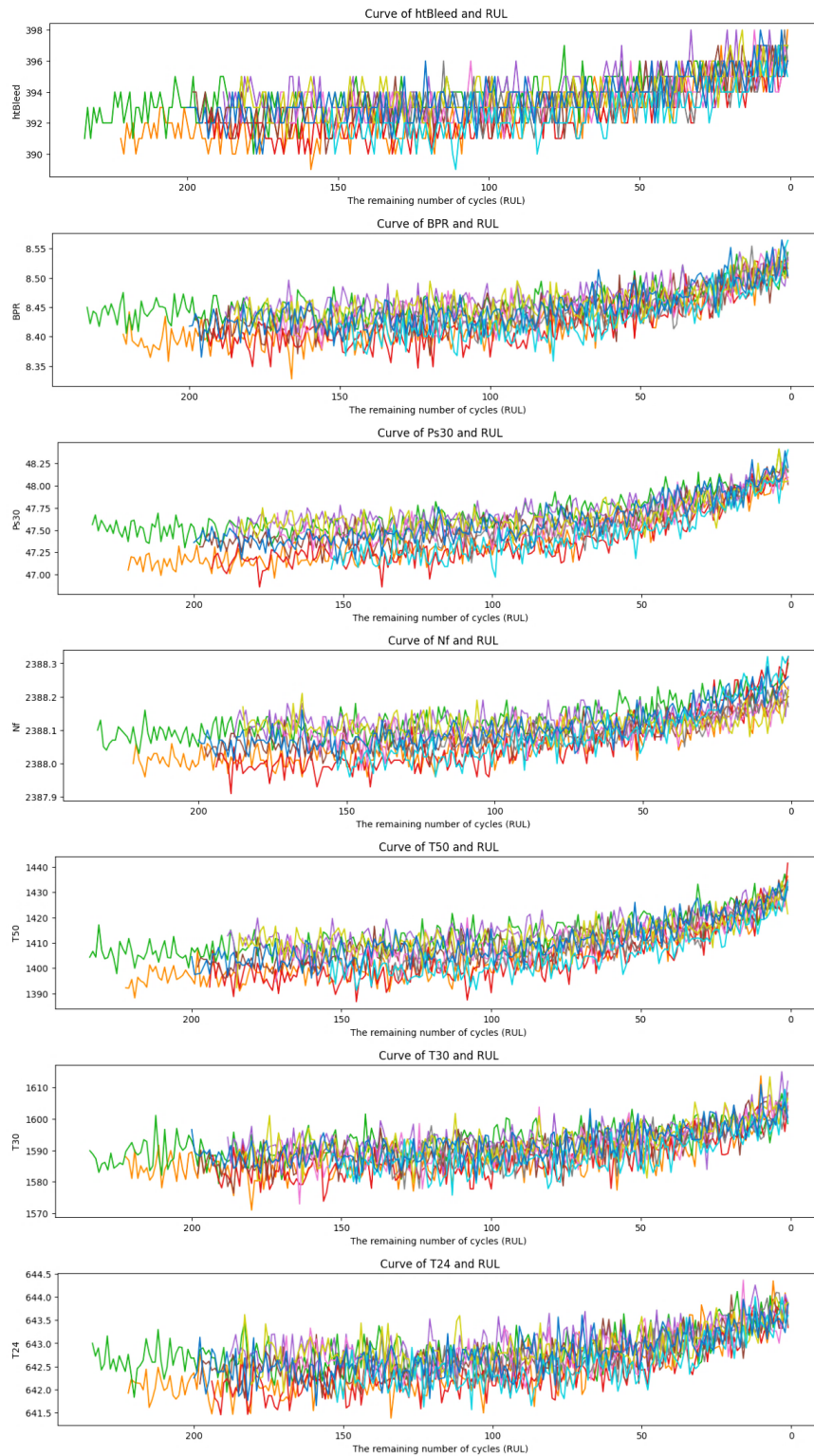


Figure 5. Correlation between RUL and features T24, T30, T50, Nf, Ps30, BPR, and htBleed.

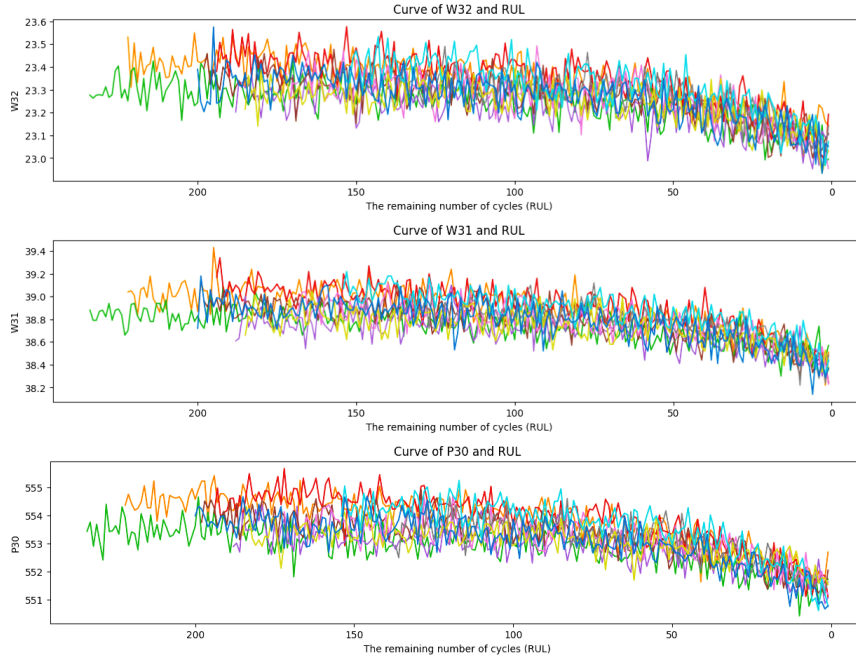


Figure 6. Correlation between RUL and features P30, W31, and W32.

Where $\tilde{C}$ represents the candidate cell state, and $\tanh(\cdot)$ serves as the nonlinear activation function in the LSTM architecture. The parameters W_c and b_c correspond to the updated weight matrix and bias term, respectively. The updated cell state C_t and hidden state h_t are then computed through additional operations, as defined in Equation (5).

$$\begin{cases} C_t = f_t \circ C_{(t-1)} + i_t \circ \tilde{C} \\ h_t = o_t \cdot \tanh(C_t) \end{cases} \quad (5)$$

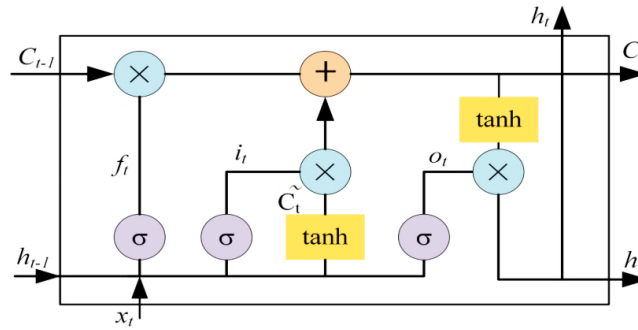


Figure 7. LSTM network structure.

3.5. Evaluation Metrics

After network development, training of LSTM network is carried out, and selection of hyper parameters is made on the basis of root mean square error (RMSE) between actual RUL and predicted RUL for each engine of the turbofan dataset. Our proposed approach is evaluated using two widely used evaluation methods: RMSE and MAE

[?]. Mean absolute error (MAE) is a commonly used evaluation metric for regression and time forecasting tasks. It calculates the mean difference between the actual values y_i and the predicted values $\tilde{y}_i$ of the equipment's remaining Useful life. The errors calculated by MAE are expressed in the same units as the actual and predicted values, which facilitates error interpretation. This metric is calculated as shown in Equation (6).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i| \quad (6)$$

The Root Mean Square Error (RMSE) is similar to the MAE but places greater emphasis on larger errors by squaring the differences. It measures the root-mean-square deviation between the actual values y_i and the predicted values $\tilde{y}_i$ of the equipment's Remaining Useful Life (RUL). This metric is mathematically defined in Equation (7).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2} \quad (7)$$

4. Experiment and Results

4.1. Experiment Environments

The experimental setup for the hardware and software utilized in the paper is displayed in Table 5. The Intel(R) Core(TM) i5-8365U CPU and the Intel UHD Graphics 620 graphics card are part of the hardware setup. Windows is the operating system that powers the applications. We utilized Python 3.11, a popular programming language in the data analysis and machine learning fields. Additionally, we developed and ran our model using the open-source machine learning framework TensorFlow 2.18.0. TensorFlow offers a range of libraries for creating deep learning models.

Table 5. System Specification

Hardware Environment	Software Environment
CPU: Intel(R) Core(TM) i5-8365U	Windows 11, TensorFlow 2.18.0
GPU: Intel UHD Graphics 620	Python 3.11

4.2. Comparative Analysis for DL models

In this section, we compare the performance of three deep learning models on the C-MAPSS dataset. The models under evaluation include Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and our proposed model. The objective is to quantify their predictive performance using key evaluation metrics such as MAE and RMSE. We begin by implementing the RNN and GRU models before analyzing their results and comparing them to the proposed approach.

A. Recurrent Neural Networks (RNN)

The RNN model follows a recurrent architecture to process input sequences of length 20, each containing 18 features. The proposed RNN architecture, illustrated in Table 6, begins with an initial RNN layer comprising 128 units. Next, a Dropout layer with a rate of 20% is applied, randomly setting 20% of the input units to zero during each update to prevent overfitting. This is followed by a second RNN layer with 64 units, accompanied by another Dropout layer with the same 20% rate. Subsequently, a Dense layer is added, fully connecting all nodes to the next layer with 16 neurons and using the ReLU activation function. Another Dropout layer with a 20% rate follows. Finally, an output Dense layer is employed with a ReLU activation function, producing an output of dimension 1. The loss function used for optimization is Mean Squared Error (MSE), and the Adam optimizer is utilized for training.

Table 6. Proposed RNN Architecture

Layer (type)	Output Shape	Param #
simple_rnn_2 (SimpleRNN)	(None, 20, 128)	18,816
dropout_11 (Dropout)	(None, 20, 128)	0
simple_rnn_3 (SimpleRNN)	(None, 64)	12,352
dropout_12 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 16)	1,040
dropout_13 (Dropout)	(None, 16)	0
dense_9 (Dense)	(None, 1)	17

B. Gated Recurrent Units (GRU)

The model architecture and hyperparameters stay largely the same, but this time, instead of an RNN (Recurrent Neural Network), we employ a GRU model with a recurrent structure designed to handle input sequences of length 20, each consisting of 18 features. The architecture of the proposed GRU is shown in Table 7.

Table 7. Proposed GRU Architecture

Layer (type)	Output Shape	Param #
gru_2 (GRU)	(None, 20, 128)	56,832
dropout_9 (Dropout)	(None, 20, 128)	0
gru_3 (GRU)	(None, 64)	37,248
dropout_10 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 16)	1,040
dropout_11 (Dropout)	(None, 16)	0
dense_9 (Dense)	(None, 1)	17

C. Results and Discussion

In this section, we present a detailed analysis of the Remaining Useful Life prediction results obtained using our proposed LSTM-based model. The training process of the LSTM network involves several key hyperparameters that significantly impact model accuracy. These hyperparameters include the learning rate, batch size, number of layers, number of neurons per layer, optimizer choice, and the number of training epochs. Since there is no universal guideline for selecting the optimal values of these parameters, an iterative grid search approach is typically employed to determine the best-performing configuration. By systematically evaluating different parameter combinations, we identify the setup that yields the most accurate predictions. The hyperparameters used in training are summarized in Table 8.

The learning curve of the proposed LSTM model, shown in Figure 8, provides insights into the training dynamics. The figure depicts the loss variation across epochs for both the training and validation datasets. Initially, both training and validation losses decrease significantly, indicating that the model is effectively learning the underlying patterns. However, after approximately 20 epochs, the validation loss stabilizes and begins to fluctuate, while the training loss continues to decrease.

As mentioned earlier, the proposed model is compared with RNN and GRU models. Table 9 presents the performance of the three algorithms RNN, GRU, and LSTM evaluated using two key metrics: MAE and RMSE. Upon examining the results in Table 9, it is evident that the LSTM model achieves a significantly lower MAE (27.11) compared to the GRU (38.44) and RNN (37.92) models. This indicates that LSTM's predictions are, on average, closer to the actual values.

Additionally, the LSTM model also exhibits a lower RMSE (35.12) compared to GRU (46.36) and RNN (46.75), suggesting that LSTM effectively mitigates large prediction errors and produces more stable results. Based on both performance metrics, the LSTM model clearly outperforms the GRU and RNN models on the test dataset.

Table 8. Hyperparameter Configuration

Hyperparameter	Value
Epochs	70
Batch size	250
Loss function	Mean Squared Error (MSE)
Optimization algorithm	Adam
Activation function	ReLU
Kernel initializer	Uniform
Validation set	0.15 (15%)
Learning Rate	0.01
No. of LSTM layers	2
No. of neurons in each LSTM layer	16

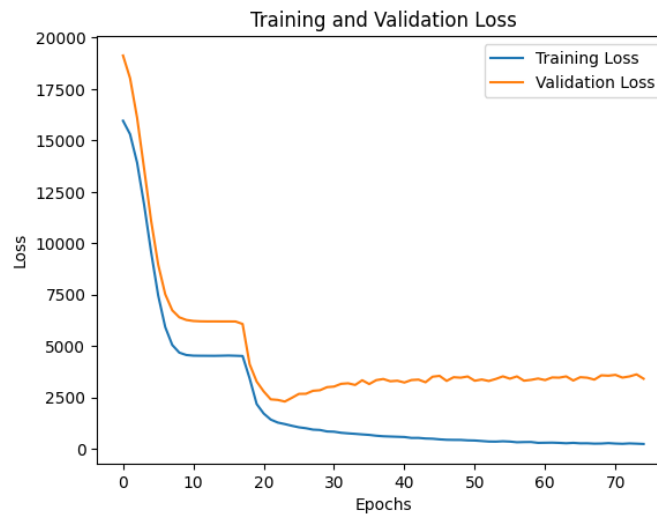


Figure 8. Learning curve of the proposed LSTM.

Table 9. Performance Comparison of Models

Model	MAE	RMSE
RNN	37.92	46.75
GRU	38.44	46.36
Proposed-model	27.11	35.12

This suggests that LSTM provides more accurate and reliable Remaining Useful Life predictions for this dataset. Consequently, for the task of RUL prediction on the turbofan engine dataset, the LSTM algorithm appears to be a more suitable choice than GRU and RNN. For a more in-depth analysis of the performance of the RNN, GRU, and LSTM algorithms, we have plotted comparative graphs showing the turbofan engines against the actual RUL values (real RUL) alongside the predicted RUL values for each model (predicted RUL). The comparison of the RNN, GRU, and LSTM models based on their respective RUL prediction graphs reveals significant differences in their performance.

For a more in-depth analysis of the performance of the RNN, GRU, and LSTM algorithms, we have plotted

comparative graphs showing the turbofan engines against the actual RUL values (real RUL) alongside the predicted RUL values for each model (predicted RUL). The comparison of the RNN, GRU, and LSTM models based on their respective RUL prediction graphs reveals significant differences in their performance.

The superior performance of the LSTM model can be explained by its internal memory structure and gating mechanisms, which enable it to capture long-term temporal dependencies in complex degradation patterns. In contrast, the standard RNN is limited by the vanishing gradient problem, reducing its ability to retain long-term information. Although the GRU partially addresses this limitation through a simplified gating structure, it may still be less effective than LSTM in modeling highly nonlinear and long-duration temporal sequences, such as those found in turbofan engine degradation data. To complement the quantitative evaluation, a qualitative analysis is conducted through graphical comparisons of predicted and actual RUL values (Figures 9, 10, 11). These visualizations show that the LSTM model more closely follows the true degradation trends, producing smoother and more consistent predictions across engine cycles. On the other hand, the RNN and GRU models exhibit larger fluctuations and noticeable deviations, particularly in the later stages of the engine lifecycle, where accurate prediction is critical for maintenance decision-making. Despite the strong performance achieved by the LSTM model, the results remain influenced by several important factors, including the selected sequence length, the feature representation, and the hyperparameter configuration. The adopted setup provides a balance between predictive accuracy and computational efficiency, enabling effective modeling of temporal dependencies. However, the use of a relatively simple variance-based feature selection approach may limit the model's ability to fully exploit the most informative variables.

Figure 9 illustrates the comparison between RNN pre-dicted values and actual values, showing that while the RNN model captures certain trends, it struggles with accuracy, particularly in highly volatile regions. The predicted values (red) often deviate from the actual RUL (blue), especially in the mid-range of engine IDs (e.g., 40–80), indicating that RNN faces challenges in handling sequential dependencies and generalizing across various degradation patterns.

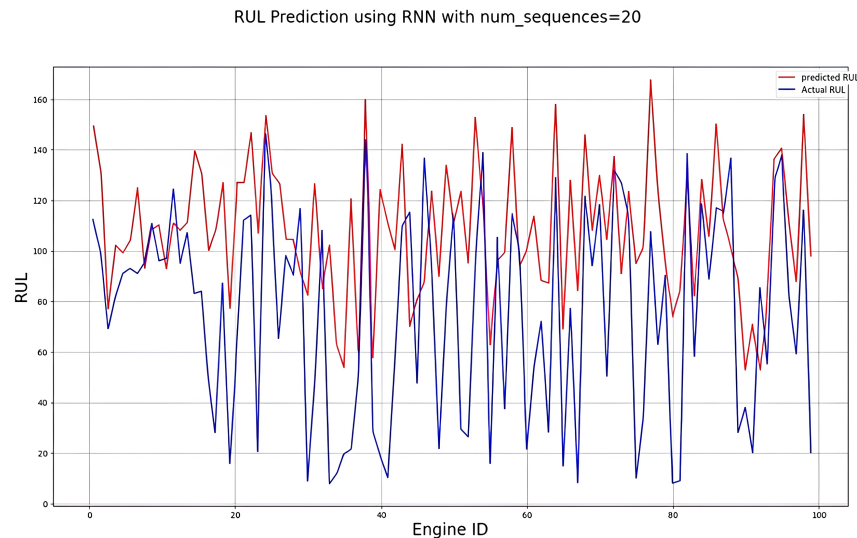


Figure 9. Comparison Between RNN Predicted Values and Actual Values.

Figure 10 illustrates the comparison between GRU predicted values and actual values, demonstrating an improvement over RNN but with a tendency to overestimate the RUL. The red line remains consistently above the blue line in many cases, which could be problematic for maintenance planning, as overpredictions may delay necessary interventions and increase the risk of unexpected failures. However, GRU exhibits greater stability compared to RNN in certain regions.

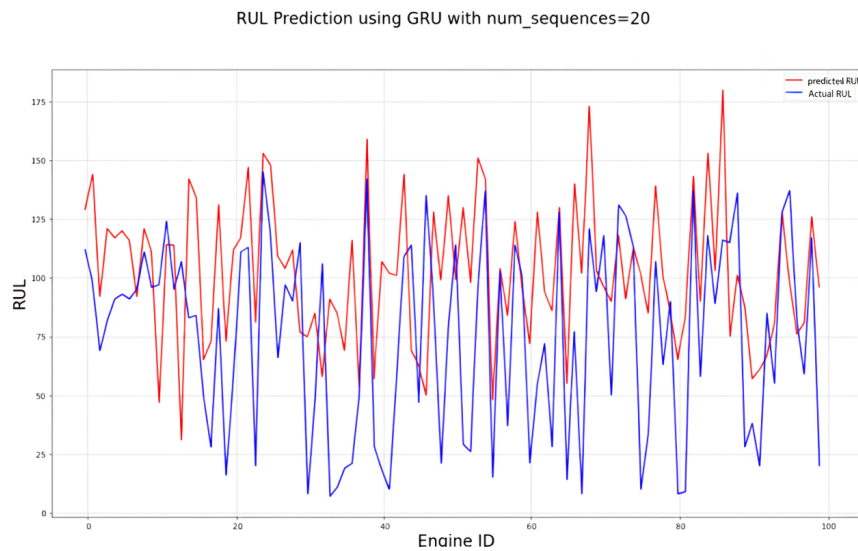


Figure 10. Comparison Between GRU Predicted Values and Actual Values.

Figure 11 illustrates the comparison between LSTM predicted values and actual values, revealing the most accurate and stable performance among the three models. The LSTM model closely follows the actual RUL values, effectively capturing long-term dependencies and reducing prediction errors. While minor discrepancies remain, particularly in regions with sharp variations, LSTM demonstrates superior trend-following behavior, highlighting its robustness in predictive maintenance. These findings confirm LSTM's effectiveness in forecasting Remaining Useful Life, minimizing both underestimation and overestimation errors, and making it the most suitable model for optimizing maintenance strategies and reducing unexpected equipment failures.

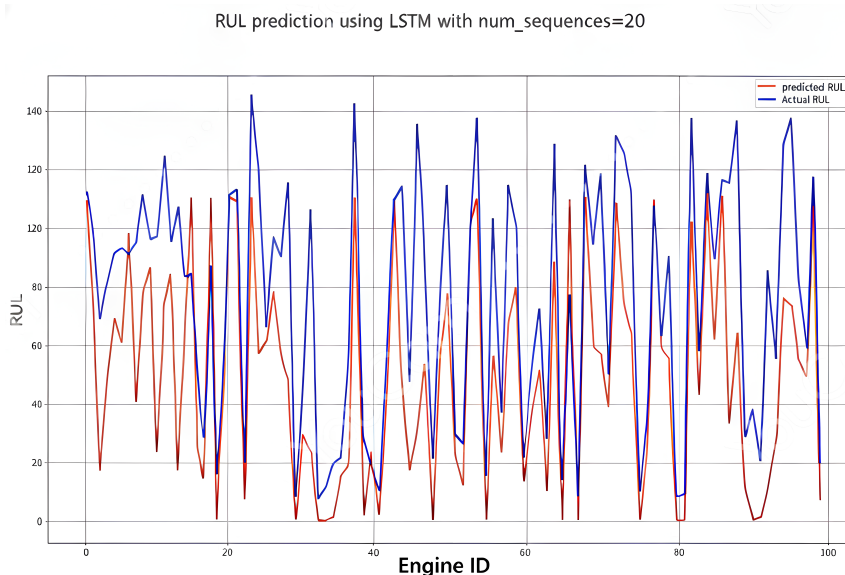


Figure 11. Comparison Between LSTM Predicted Values and Actual Values.

D. Limitations of the Proposed Approach

The main considerations of the current comparative experimental study can be summarized as follows:

- *Scope of benchmark architectures:* The experimental comparison considers three recurrent deep learning architectures, namely RNN, GRU, and LSTM. These models provide a consistent basis for evaluating recurrent sequence modeling for RUL prediction. However, the study does not experimentally include several recent approaches, such as advanced LSTM variants, Transformer-based architectures, hybrid CNN-LSTM models, and ensemble learning methods.
- *Positioning within the state-of-the-art:* The aforementioned advanced architectures were not implemented and evaluated under the same experimental conditions. Therefore, the reported results are intended to establish the performance of the proposed LSTM model relative to the RNN and GRU baselines evaluated using the same dataset representation, sequence length, preprocessing strategy, and evaluation protocol, rather than to claim superiority over all existing state-of-the-art RUL prediction methods.
- *Dependence on experimental configuration:* The performance of the proposed LSTM model may depend on several experimental factors, including the selected sequence length, feature representation, hyperparameter configuration, and training strategy. In particular, the adopted variance-based feature selection approach provides a simple mechanism for reducing input dimensionality but may not capture all features that are potentially informative for modeling long-term degradation behavior.

Future work will extend the evaluation to advanced LSTM and attention-based architectures, CNN-LSTM hybrids, Transformer-based models, and ensemble methods, together with systematic sensitivity and ablation analyses to further assess the robustness and generalizability of the proposed approach.

5. Conclusion

In this paper, we introduced an LSTM-based approach for predicting the Remaining Useful Life of industrial equipment, with a specific focus on aircraft turbojet engines. To evaluate the effectiveness of the proposed model, we performed a comparative study against well-established time-series forecasting architectures, namely RNN and GRU, which are well-suited for handling sequential data relevant to RUL prediction tasks. These models were evaluated using key performance metrics, namely MAE and RMSE. The experimental results show that the LSTM architecture consistently outperforms both RNN and GRU models. In particular, it achieves lower MAE and RMSE values, reflecting higher accuracy in estimating the remaining useful life of the engines. Therefore, the proposed framework establishes a new paradigm for intelligent predictive maintenance, driven by LSTM-based deep learning, bridging temporal sensor data analysis with accurate RUL estimation and data-driven maintenance decision support.

Future Work:

Several promising research directions can be considered to further extend the proposed approach:

- *Advanced Deep Learning Architectures:* Future work will investigate advanced LSTM variants, attention-based recurrent architectures, Transformer-based models, and CNN-LSTM hybrid architectures to determine whether more sophisticated temporal and feature-extraction mechanisms can further improve RUL prediction performance.
- *Incorporating Federated Learning:* Investigating federated learning approaches for predictive maintenance, where data remain distributed across multiple devices, can enhance data privacy while reducing the need for centralized data aggregation and potentially lowering communication costs.
- *Hybrid Models:* Exploring hybrid models that combine LSTM networks with other machine learning techniques, such as Random Forests or Support Vector Machines, may further improve prediction accuracy, robustness, and generalization.

- *Data Augmentation*: Implementing advanced data augmentation techniques can improve the model's ability to generalize, particularly when dealing with noisy, heterogeneous, or limited sensor data.
- *Real-Time Predictive Maintenance*: Extending the current model to support real-time predictive maintenance systems could enable continuous monitoring of sensor measurements and dynamic RUL estimation in operational environments.

REFERENCES

1. M. Soori, B. Arezoo, and R. Dastres, *Internet of Things for Smart Factories in Industry 4.0: A Review*, Internet of Things and Cyber-Physical Systems, vol. 3, pp. 192–204, 2023.
2. A. Valenzuela-Cobos, B. Vera-Cabanilla, L. Castillo-Heredia, and J. Valenzuela-Cobos, *Industry 4.0 in Logistics Management in Latin America: A Bibliometric Review*, Journal of Industrial Engineering and Management, vol. 18, no. 1, pp. 115–129, 2025.
3. A. Schlemitz and V. Mezhyuev, *Approaches for Data Collection and Process Standardization in Smart Manufacturing: Systematic Literature Review*, Journal of Industrial Information Integration, vol. 38, p. 100578, 2024.
4. A. Patil, G. Soni, and A. Prakash, *Data-Driven Approaches for Impending Fault Detection of Industrial Systems: A Review*, International Journal of System Assurance Engineering and Management, vol. 15, no. 4, pp. 1326–1344, 2024.
5. M. Mladineo (ed.), *Industry 4.0—From Smart Factory to Cognitive Cyber-Physical Production System and Cloud Manufacturing*, MDPI – Multidisciplinary Digital Publishing Institute, 2022, 138 p.
6. A. Rai, J. Shastri, and H. Bansal, *Artificial Intelligence Techniques in Predictive Maintenance, Their Applications, Challenges, and Prospects*, in Artificial Intelligence-Enabled Digital Twin for Smart Manufacturing, 2024, pp. 565–579.
7. A. Aminzadeh, S. Sattarpanah Karganroudi, S. Majidi, et al., *A Machine Learning Implementation to Predictive Maintenance and Monitoring of Industrial Compressors*, Sensors, vol. 25, no. 4, p. 1006, 2025.
8. J. Paredes, D. Chávez, and R. Isa-Jara, *A Hybrid Machine Learning Algorithm Approach to Predictive Maintenance Tasks: A Comparison with Machine Learning Algorithms*, Results in Engineering, p. 105137, 2025.
9. I. Elharnaf, K. Achtaich, and S. Tetouani, *Improving Predictive Maintenance Performance Using Machine Learning and Vibration Analysis Algorithms*, Engineering Proceedings, vol. 97, no. 1, p. 45, 2025.
10. W. Li and T. Li, *Comparison of Deep Learning Models for Predictive Maintenance in Industrial Manufacturing Systems Using Sensor Data*, Scientific Reports, vol. 15, no. 1, p. 23545, 2025.
11. Y. Lv, Z. Pan, and Y. Li, *Dynamic Predictive Maintenance Strategy for Multi-Component System Based on LSTM and Hierarchical Clustering*, Quality and Reliability Engineering International, vol. 41, no. 1, pp. 30–50, 2025.
12. J. Nolan and S. Reynolds, *Deep Learning Approaches for Predictive Maintenance in Industrial Systems*, ITS Transactions on Electrical and Electronics Engineering, vol. 13, no. 1, pp. 25–32, 2024.
13. U. Khan, D. S. Cheng, F. Setti, et al., *A comprehensive survey on deep learning-based predictive maintenance*, ACM Transactions on Embedded Computing Systems, vol. 25, no. 2, pp. 1–43, 2026, doi: 10.1145/3732287.
14. S. Arena, E. Florian, F. Sgarbossa, E. Sølvsberg, and I. Zennaro, "A conceptual framework for machine learning algorithm selection for predictive maintenance," *Engineering Applications of Artificial Intelligence*, vol. 133, Art. no. 108340, 2024, doi: 10.1016/j.engappai.2024.108340.
15. X. Li, P. Wang, X. Li, et al., "Generative AI for predictive maintenance for manufacturing: A survey," *Procedia CIRP*, vol. 139, pp. 35–40, 2026, doi: 10.1016/j.procir.2025.09.012.
16. M. Nagy, K. Valaskova, and G. Lazaroiu, "Digitalization-based predictive maintenance systems, Internet of Things sensor networks, and machine intelligence algorithms for Industry 4.0/5.0 manufacturing processes," *Systems*, vol. 14, no. 3, Art. no. 281, 2026, doi: 10.3390/systems14030281.
17. M. Achouch, M. Dimitrova, K. Ziane, S. Sattarpanah Karganroudi, R. Dhoubi, H. Ibrahim, and M. Adda, "On predictive maintenance in Industry 4.0: Overview, models, and challenges," *Applied Sciences*, vol. 12, no. 16, Art. no. 8081, 2022, doi: 10.3390/app12168081.
18. M. Er-Ratby, A. Kobi, Y. Sadraoui, et al., "The impact of predictive maintenance on the performance of industrial enterprises," *SN Computer Science*, vol. 6, no. 1, Art. no. 73, 2025, doi: 10.1007/s42979-024-02841-5.
19. I. Ul Hassan, K. Panduru, and J. Walsh, "Predictive maintenance in Industry 4.0: A review of data processing methods," *Procedia Computer Science*, vol. 257, pp. 896–903, 2025, doi: 10.1016/j.procs.2025.01.123.
20. F. Talssi, F. Jeffali, A. Sadiqi, A. El Barkany, and J. Bouchnaif, "Predictive maintenance using vibration monitoring for rotating machinery faults in the context of Industry 4.0," *International Journal of Advanced Manufacturing Technology*, pp. 1–21, 2026, doi: 10.1007/s00170-026-17802-6.
21. D. A. Alqaraleh, S. S. H. Hajjaj, H. Mohamed, et al., "Predictive maintenance of mining centrifuges using machine learning and deep learning models," *Engineering, Technology & Applied Science Research*, vol. 16, no. 1, pp. 31722–31732, 2026, doi: 10.48084/etasr.15162.
22. P. Y. L. Wong, T. M. Leung, W. Zhang, et al., "Enhancing energy efficiency in road transport systems: A comparative study of Australia, Hong Kong and the UK," *Energies*, vol. 19, no. 1, Art. no. 266, 2026, doi: 10.3390/en19010266.
23. M. Moleđa, B. Małysiak-Mrozek, W. Ding, V. Sunderam, and D. Mrozek, "From corrective to predictive maintenance—A review of maintenance approaches for the power industry," *Sensors*, vol. 23, no. 13, p. 5970, 2023, doi: 10.3390/s23135970.
24. T. Zhu, Y. Ran, X. Zhou, and Y. Wen, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, 2019, doi: 10.48550/arXiv.1912.07383.
25. A. Supramaniam, S. S. S. Ahmad, and Z. I. M. Yusoh, "Predictive maintenance using deep reinforcement learning," in *Proc. 2024 IEEE Int. Conf. Artif. Intell. Eng. Technol. (IICAIET)*, 2024, pp. 671–676, doi: 10.1109/IICAIET60025.2024.00123.

26. T. Akyaz and D. Engin, "Machine learning-based predictive maintenance system for artificial yarn machines," *IEEE Access*, vol. 12, pp. 125446–125461, 2024, doi: 10.1109/ACCESS.2024.3356789.
27. J. Ahn, Y. Lee, N. Kim, C. Park, and J. Jeong, "Federated learning for predictive maintenance and anomaly detection using time series data distribution shifts in manufacturing processes," *Sensors*, vol. 23, no. 17, Art. no. 7331, 2023, doi: 10.3390/s23177331.
28. M. Pech, J. Vrchota, and J. Bednář, "Predictive maintenance and intelligent sensors in smart factory," *Sensors*, vol. 21, no. 4, Art. no. 1470, 2021, doi: 10.3390/s21041470.
29. A. P. Azodo, F. C. T. Mezue, and O. S. Onyekwere, "Leveraging AI for predictive maintenance in energy systems: Enhancing reliability and efficiency," *Australian Journal of Multi-Disciplinary Engineering*, pp. 1–19, 2025.
30. A. Nasser and H. Al-Khazraji, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 721–730, 2022, doi: 10.11591/ijece.v12i1.pp721-730.
31. D. Pagano, "A predictive maintenance model using long short-term memory neural networks and Bayesian inference," *Decision Analytics Journal*, vol. 6, Art. no. 100174, 2023, doi: 10.1016/j.dajour.2023.100174.
32. R. Wu and J. Ma, "An improved LSTM neural network with uncertainty to predict remaining useful life," in *Proc. 2019 CAA Symp. Fault Detection, Supervision and Safety for Techn. Processes (SAFEPROCESS)*, 2019, pp. 274–279, doi: 10.1109/SAFEPROCESS.2019.00056.
33. D. Bruneo and F. De Vita, "On the use of LSTM networks for predictive maintenance in smart industries," in *Proc. 2019 IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2019, pp. 241–248, doi: 10.1109/SMARTCOMP.2019.00054.
34. V. J. Jimenez, N. Bouhmala, and A. H. Gausdal, "Developing a predictive maintenance model for vessel machinery," *Journal of Ocean Engineering and Science*, vol. 5, no. 4, pp. 358–386, 2020, doi: 10.1016/j.joes.2020.06.002.
35. I. Daniyan, K. Mpofu, M. Oyesola, B. Ramatsetse, and A. Adeodu, "Artificial intelligence for predictive maintenance in the railcar learning factories," *Procedia Manufacturing*, vol. 45, pp. 13–18, 2020, doi: 10.1016/j.promfg.2020.04.003.
36. X. Bampoula, G. Siaterlis, N. Nikolakis, and K. Alexopoulos, "A deep learning model for predictive maintenance in cyber-physical production systems using LSTM autoencoders," *Sensors*, vol. 21, no. 3, Art. no. 972, 2021, doi: 10.3390/s21030972.
37. M. Matuszczak, M. Żbikowski, and A. Teodorczyk, "Predictive modelling of turbofan engine components condition using machine and deep learning methods," *Eksplotacja i Niezawodność – Maintenance and Reliability*, vol. 23, no. 2, pp. 359–370, 2021, doi: 10.17531/ein.2021.2.15.
38. O. Asif, S. A. Haider, S. R. Naqvi, J. F. Zaki, K. S. Kwak, and S. R. Islam, "A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset," *IEEE Access*, vol. 10, pp. 95425–95440, 2022, doi: 10.1109/ACCESS.2022.3203456.
39. R. Raparthy, B. Dodda, and S. Maruthi, "Predictive maintenance in IoT devices using time series analysis and deep learning," *Dandao Xuebao (Journal of Ballistics)*, vol. 35, pp. 1–10, 2023.
40. M. Shahin, F. F. Chen, A. Hosseinzadeh, and N. Zand, "Using machine learning and deep learning algorithms for downtime minimization in manufacturing systems: An early failure detection diagnostic service," *International Journal of Advanced Manufacturing Technology*, vol. 128, no. 9, pp. 3857–3883, 2023, doi: 10.1007/s00170-023-11845-2.
41. A. Jaenal, J.-R. Ruiz-Sarmiento, and J. Gonzalez-Jimenez, "MachNet, a general deep learning architecture for predictive maintenance within the Industry 4.0 paradigm," *Engineering Applications of Artificial Intelligence*, vol. 127, Art. no. 107365, 2024, doi: 10.1016/j.engappai.2023.107365.
42. S. Rath, D. Saha, S. Chatterjee, and A. K. Chakraborty, "Remaining useful life prediction of turbofan engine in varied operational conditions considering change point: A novel deep learning approach with optimum features," *Mathematics*, vol. 13, no. 1, Art. no. 130, 2024, doi: 10.3390/math13010130.
43. A. Saxena and K. Goebel, "Turbofan engine degradation simulation dataset," *NASA Ames Prognostics Data Repository*, 2008, pp. 878–887.
44. A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. 2008 Int. Conf. Prognostics and Health Management*, 2008, pp. 1–9.
45. S. Strasser and M. Klettke, "Transparent data preprocessing for machine learning," in *Proc. 2024 Workshop Human-in-the-Loop Data Analytics*, 2024, pp. 1–6.
46. I. Ul Hassan, K. Panduru, and J. Walsh, "Review of data processing methods used in predictive maintenance for next generation heavy machinery," *Data*, vol. 9, no. 5, Art. no. 69, 2024, doi: 10.3390/data9050069.
47. P. Wu, Q. Zhang, G. Wang, F. Yang, and F. Xue, "Dynamic feature selection combining standard deviation and interaction information," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 4, pp. 1407–1426, 2023, doi: 10.1007/s13042-022-01700-1.