








## A Hybrid Deep Learning Framework for Data Quality Enhancement and Uncertainty-Aware Remaining Useful Life Prediction of Aero-Engines

Samira Abderrezek<sup>1,2\*</sup>, Naouel Ouafek<sup>2</sup>, Abdelhabib Bourouis<sup>1</sup>, Eric Bechhoefer<sup>3</sup>, Tarek Berghout<sup>4</sup>

<sup>1</sup> ReLa (CS)<sup>2</sup> Laboratory, University of Oum El Bouaghi, Oum El Bouaghi 04000, Algeria

<sup>2</sup> Ecole Normale Supérieure de Constantine, Constantine 25000, Algeria

<sup>3</sup> GPMS International, Inc., Waterbury 05676, VT, United States of America

<sup>4</sup> Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria

Corresponding Author Email: [abderrezak.samira@ensc.dz](mailto:abderrezak.samira@ensc.dz)

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310214>

### ABSTRACT

**Received:** 16 October 2025

**Revised:** 5 January 2026

**Accepted:** 18 February 2026

**Available online:** 28 February 2026

#### Keywords:

*Remaining Useful Life prediction, deep learning, U-Net, Temporal Convolutional Network, transformer, uncertainty quantification, C-MAPSS, Bayesian optimization*

Accurate Remaining Useful Life (RUL) prediction for aero-engines requires not only strong predictive accuracy but also robust data preparation and reliable uncertainty estimates. In this study, we propose an integrated framework for RUL prediction on the C-MAPSS benchmark that combines data quality enhancement, hybrid temporal modeling, and uncertainty-aware hyperparameter optimization. The preprocessing pipeline includes wavelet denoising, outlier handling, and feature selection guided by prognostic indicators. The prediction model integrates U-Net, Temporal Convolutional Networks (TCN), and multi-head self-attention to capture multi-scale and long-range temporal dependencies. To improve prediction reliability, a customized Bayesian optimization objective jointly considers prediction accuracy and confidence-interval quality. Experiments on the four C-MAPSS subsets yield Root Mean Squared Error (RMSE) values of 7.63, 12.39, 10.73, and 12.67 for FD001–FD004, respectively, which are competitive with previously reported results. The uncertainty-aware optimization also provides improved interval quality compared with RMSE-only optimization. These findings suggest that combining data quality management, hybrid deep learning, and uncertainty-aware optimization is a promising direction for more reliable aero-engine RUL prediction.

## 1. INTRODUCTION

Planning for condition-based and scheduled maintenance relies heavily on the accurate estimation of the Remaining Useful Life (RUL) of components. In the aerospace, manufacturing, and transportation industries, failures can have severe consequences, ranging from financial losses to catastrophic accidents. Accurate RUL prediction enables timely interventions, minimizes the risk of unexpected failures, and optimizes maintenance schedules [1]. However, accurately predicting RUL remains challenging due to the complexity of modern industrial systems and harsh operating conditions, creating demand for advanced techniques that deliver reliable forecasts.

Deep learning models are increasingly adopted for RUL estimation because of their capacity to capture nonlinear relationships and temporal dependencies in sensor data. These models have shown superior performance over conventional approaches in numerous prognostic applications [2]. In particular, Transformer architectures, which rely on self-attention mechanisms, have demonstrated a strong ability to capture long-term dependencies, outperforming LSTM and CNN models [3-5]. This makes them promising for predicting the RUL of turbofan engines by extracting critical features over extended sequences.

Data quality is also crucial in achieving accurate predictions, particularly when utilizing the C-MAPSS dataset, a widely used benchmark in aerospace applications [6]. C-MAPSS provides multidimensional time-series measurements from simulated turbofan engines, reflecting realistic degradation under diverse operating conditions. However, the dataset also contains noise, inconsistencies, and outliers that can degrade model accuracy. Effective data preparation is thus essential. Denoising methods enhance clarity by filtering irrelevant noise, while outlier removal improves reliability and robustness. Careful preprocessing ensures that predictive models learn effectively from informative features, leading to more accurate and stable RUL predictions [7, 8].

Another major challenge is the lack of uncertainty quantification in most existing approaches. Conventional models provide single-point estimates without associated confidence levels, which limits their interpretability and trust in safety-critical contexts. Integrating uncertainty awareness allows models to generate prediction intervals instead of deterministic outputs. This improves reliability, supports risk-aware maintenance decisions, and increases user confidence in model outcomes.

Existing deep learning approaches for RUL prediction, including hybrid LSTM-CNN [9-15], TCN with attention [16], Transformer-based models [17-19], and U-Net variants [20-

22], have demonstrated significant improvements. However, none of these methods jointly incorporates uncertainty quantification with robust data quality enhancement, especially when applied to the C-MAPSS dataset, which involves multivariate operating conditions and noisy sensor data [6-8]. Current approaches also struggle with long-term dependencies, computational complexity, and robustness against noise.

The main contribution of this work is the design of an integrated reliability-oriented framework for aero-engine RUL prediction on C-MAPSS. Rather than presenting preprocessing, model design, and uncertainty handling as isolated steps, the proposed framework links them into a coherent prognostic pipeline. Within this framework, three technical contributions are introduced. First, a data quality enhancement pipeline combines wavelet denoising and dataset-adapted outlier handling to improve the reliability of sensor measurements. Second, a hybrid U-Net-TCN-Transformer model is designed to jointly capture multi-scale temporal patterns, long-range dependencies, and global contextual interactions. Third, an uncertainty-aware Bayesian optimization strategy is used to balance prediction accuracy with confidence interval quality. Together, these components aim to improve not only point-estimate accuracy, but also the reliability and interpretability of RUL prediction.

**1. Data preprocessing and quality enhancement:** We employ discrete wavelet denoising and outlier removal using Z-score, Mahalanobis distance, Interquartile Range (IQR), and Percentile methods, assessing their effectiveness through prognostic indicators such as monotonicity, prognosability, and robustness [7, 8].

**2. Deep sequential learning:** We design a U-Net-based model integrating TCN to capture long-range dependencies and a Transformer block for global context, achieving efficient temporal and spatial feature extraction [16-19].

**3. Uncertainty quantification:** Bayesian optimization is customized to minimize both prediction error and confidence interval width, providing reliable probabilistic predictions [5, 23].

**4. Benchmark validation:** The proposed approach is evaluated on the four C-MAPSS subsets and yields competitive performance relative to previously reported results, while also improving uncertainty-related evaluation [9-22].

This paper is structured as follows: Section 2 reviews existing methods and key concepts in RUL estimation and deep learning. Section 3 covers preprocessing techniques. Section 4 presents our proposed model for practical RUL prediction. Section 5 discusses results, compares our model with other approaches, and analyzes uncertainty performance. Finally, Section 6 concludes and suggests future research directions.

## 2. EXISTING APPROACHES ANALYSIS

Many studies have applied deep learning methods to estimate RUL with the C-MAPSS dataset, each with distinct advantages and limitations.

In the study [9], Long Short-Term Memory (LSTM) networks were combined with classical neural networks in a hybrid deep RNN to capture temporal dependencies directly from sensor data. Dropout and a decaying learning rate were

used to improve learning efficiency. Similarly, the study [10] optimized RNN-LSTM hyperparameters such as hidden layers, learning rate, and batch size using a Genetic Algorithm, significantly enhancing prediction accuracy. To address noisy sensor data, the study [11] introduced an LSTM-FNN bootstrap model (LSTMBS) that generated both deterministic values and prediction intervals, though its computational cost was high. The study [12] further highlighted how piecewise linear degradation and the degradation starting point strongly affect LSTM accuracy. Hybrid models combining CNNs and BiLSTMs were proposed in the study [13] to capture both spatial and temporal dependencies, while the study [14] integrated convolutional autoencoders with BiLSTM, and the study [15] employed CNN, LSTM, and self-attention mechanisms for feature extraction and precise prediction. Temporal Convolutional Networks (TCNs) with attention were shown to be effective in the study [16]. A multi-head self-attention architecture was explored in the study [17], integrating temporal feature enhancement with Transformer encoders, while the study [18] combined CNN and TCN weighted by multi-head attention. Inspired by NLP Transformers, the study [19] designed a self-attention model that handled long dependencies but faced computational efficiency issues on long sequences.

Although the U-Net was originally developed for biomedical segmentation, recent works demonstrate its potential in prognostics. For instance, the study [20] integrated CNN with U-Net to estimate Li-ion battery State of Charge, achieving robustness under variable conditions but struggling with noisy aerospace data and lacking uncertainty quantification. A multiscale 1D U-Net GAN was used in the study [21] for generalized feature extraction across multiple datasets, but robustness against noise remained a challenge. The work [22] proposed a U-Net with ConvLSTM to model spatio-temporal features; however, sequential computations made it computationally expensive, leading the authors to introduce a lighter ConvJANET alternative at the cost of accuracy.

In summary, prior research demonstrates valuable progress in applying LSTM, CNN, TCN, Transformer, and U-Net models for RUL prediction. However, these directions are seldom combined within a single end-to-end framework combining robust data quality enhancement with uncertainty awareness when applied to complex multivariate datasets like C-MAPSS. This gap motivates our proposed framework, which integrates preprocessing, hybrid temporal architectures, and uncertainty quantification to advance both accuracy and reliability in aero-engine prognostics.

Research gap summary. Existing studies on C-MAPSS mainly improve RUL prediction along one of three separate directions: advanced sequential architectures, data preprocessing, or uncertainty estimation. However, these directions are rarely investigated together within a unified framework. In particular, prior work does not clearly establish how data quality enhancement interacts with hybrid temporal architectures, nor does it sufficiently integrate uncertainty-aware optimization into the end-to-end prognostic pipeline. This gap motivates the present study, which treats data preparation, temporal representation learning, and uncertainty-aware model selection as interdependent components of a single reliability-oriented framework.

### 3. MATERIALS

This section presents the C-MAPSS dataset, a benchmark for evaluating RUL prediction models, citing its structure and multivariate sensor information [24]. It emphasizes the necessity of accurate RUL estimates to enhance reliability, optimize maintenance effectiveness, and overcome challenges resulting from complex operating conditions and uncertainty.

#### 3.1 Description of the experimental dataset

The C-MAPSS dataset [24], developed by NASA in

MATLAB, simulates turbofan engine degradation under varying operating conditions. It consists of four subsets (FD001–FD004) containing complete run-to-failure trajectories, divided into training and testing sets. Each engine is described by 21 sensor measurements and 14 input parameters (see Tables 1 and 2), capturing critical parameters such as temperature, pressure, rotational speed, and fuel-to-air ratio. The prediction task is to estimate RUL by comparing predicted values with the ground truth provided for the test set. Figure 1 illustrates the turbofan layout used to contextualize the 21 sensors listed in Table 2.

**Table 1.** Composition of the C-MAPSS dataset (data from [24])

Subset	Training Engines	Test Engines	Operating Conditions	Failure Modes	Training Samples	Test Samples
FD001	100	100	1	1	20,632	13,097
FD002	260	259	6	1	53,760	33,992
FD003	100	100	1	2	24,721	16,597
FD004	249	248	6	2	61,250	41,215

**Table 2.** C-MAPSS sensor measurements used to characterize system response (adapted from [24])

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

#### 3.2 Preprocessing step

To ensure high-quality input data for the RUL prediction model, we preprocess by calculating RUL, selecting relevant sensor data, normalizing features, denoising, and removing outliers, as shown in Figure 2. These steps enhance data quality for more accurate analyses and predictions.

##### 3.2.1 Labeling process and feature selection

The RUL is calculated as shown in Eq. (1):

$$RUL_{train} = Cycle_{max} - Cycle_{now} \quad (1)$$

Here,  $Cycle_{max}$  is the maximum operating cycles for each engine, and  $Cycle_{now}$  is the engine's current cycle count.

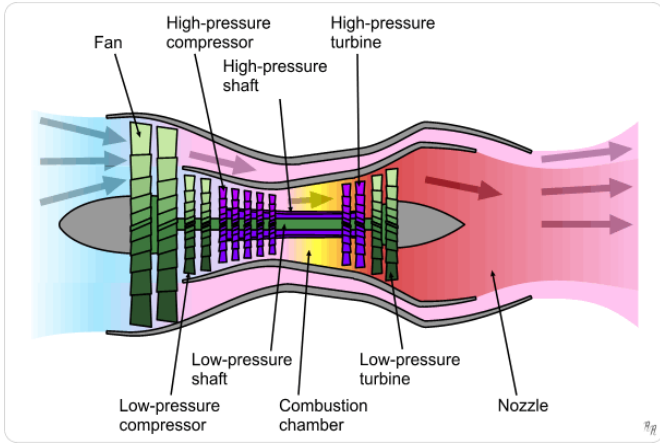
Not all sensors contribute to degradation trends. For instance, in FD001, T2, P2, P15, EPR, FarB, Nfdmd, and PCNfRdmd remain constant and were removed. For FD003, only 16 sensors were selected based on the study [12], while all sensors were retained in FD002 and FD004.

### 3.2.2 Data normalization

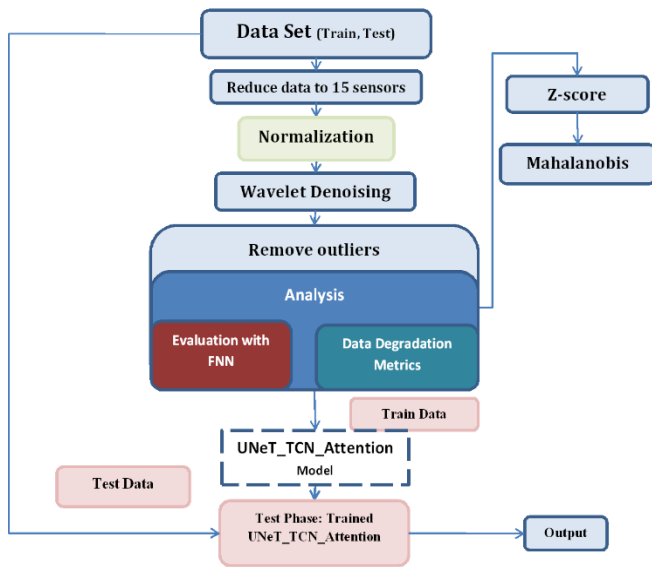
To eliminate scale differences, min–max normalization was applied as shown in Eq. (2):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

where,  $x$  is the raw value, and  $x'$  the normalized value. This ensures comparability between features with different units.



**Figure 1.** Illustration of a turbofan engine [25]



**Figure 2.** Diagram of the preprocessing step

### 3.2.3 Denoising

Signal denoising is essential to enhance quality and remove irrelevant noise. This study employs the Discrete Wavelet Transform (DWT) with the BayesShrink filter for effective denoising of multivariate time series, while preserving key signal features [26-28].

DWT represents a signal in the frequency domain by decomposing it into various frequency sub-bands at different resolution levels. For a time series signal  $x(t)$ , the wavelet transform yields approximation coefficients ( $c_{A,k}$ ) and detail coefficients ( $c_{D,j,k}$ ) through convolution with wavelet functions  $\psi(t)$  at different scales. The signal can be expressed as a combination of these coefficients as shown in Eq. (3), where  $\phi_{j_0,k}(t)$  is the scale function and  $\psi_{j,k}(t)$  is the mother wavelet function.

$$x(t) = \sum_k c_{A,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{J-1} \sum_k c_{D,j,k} \psi_{j,k}(t) \quad (3)$$

### 3.2.4 Outlier removal

In the preprocessing phase, identifying and eliminating outliers is crucial for enhancing the performance and reliability of the prediction model. Outliers can disrupt the learning process, leading to erroneous predictions. Several outlier detection methods were evaluated and compared:

a) **Z-Score method:** This method measures the number of standard deviations that separate an observation from the mean [29].

First, the process calculates the upper and lower bounds for each column, as described in Eqs. (4) and (5).

$$upL = df[i].mean() + 3df[i].std() \quad (4)$$

$$lowL = df[i].mean() - 3df[i].std() \quad (5)$$

Here,  $df[i]$  represents the data values in column or feature  $i$  of the DataFrame. The function  $df[i].mean()$  computes the arithmetic mean (average) of the values in  $df[i]$ , while  $df[i].std()$  calculates the standard deviation of the values in  $df[i]$ , which measures the spread or dispersion of the data.

The upper limit ( $upL$ ) is defined as three standard deviations above the mean of  $df[i]$ , while the lower limit ( $lowL$ ) is set as three standard deviations below the mean of  $df[i]$ . These thresholds are determined by adding and subtracting three times the standard deviation from the mean.

Next, the method counts the number of values that exceed these limits. For each detected outlier, the code replaces the outlier with the corresponding limit, a process known as capping. This approach preserves the data structure and reduces the impact of outliers, thereby improving overall data quality.

These limits are used to identify outliers based on the z-score method, assuming the data follows a normal distribution. This approach is effective for data following a normal distribution, which explains the results obtained on the FD001 and FD003 databases which follow a normal distribution. On the other hand, for FD002 and FD004, the method did not detect outliers because these datasets do not follow a normal distribution as shown in Table 3.

b) **Interquartile Range:** This non-parametric method consists of calculating the IQR (the difference between the 75th and 25th percentile) Observations below  $Q1 - 1.5 \times IQR$  or above  $Q3 + 1.5 \times IQR$  are treated as outliers [30, 31]. The IQR method is robust against non-normal distributions.

c) **Percentile:** This method is used to detect outliers in a data set based on extreme percentiles. In our case, bounds are set at the 1st and 99th percentiles.

d) **Mahalanobis:** This multivariate technique measures the distance between a point and the mean of the data set, considering correlations between variables. Points with high Mahalanobis distance are considered outliers, making this method particularly suitable for detecting multivariate outliers in subsets with correlated sensor behavior, such as FD002 and FD004.

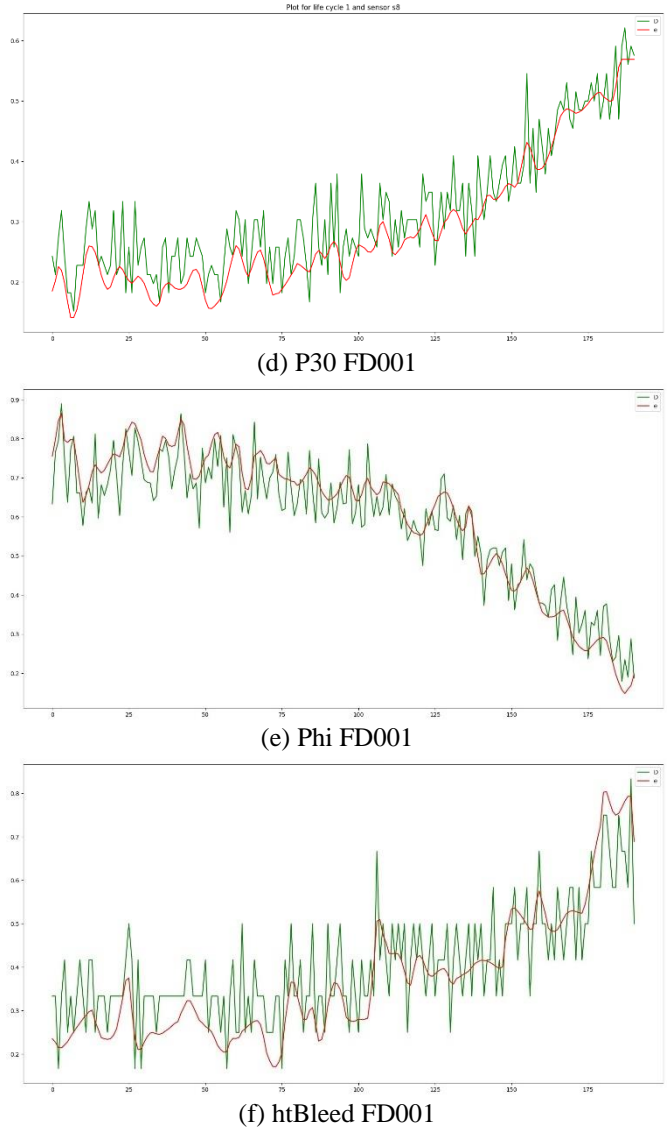
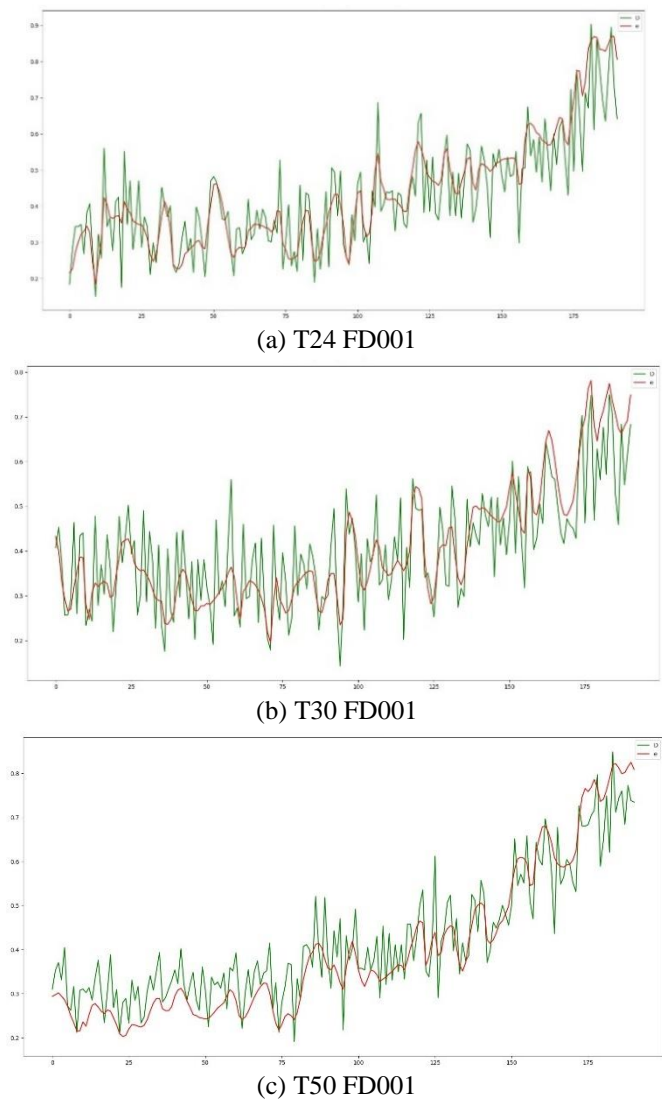
The outlier-handling strategy was not selected uniformly across all subsets, because the statistical characteristics of the

C-MAPSS subsets differ. Z-score filtering was retained when the sensor distributions were sufficiently close to normality and when it improved predictive indicators without excessive information loss. Mahalanobis distance was preferred when multivariate correlation structure was more informative than marginal thresholds, particularly for subsets with more complex operating conditions. In particular, Mahalanobis distance accounts for covariance between sensor variables, making it more suitable for multivariate correlated data. The final selection of preprocessing methods was therefore guided by a combination of statistical assumptions, outlier counts, prognostic indicators, and predictive validation results. Based on this selection strategy, the retained preprocessing choices for each C-MAPSS subset are summarized in Tables 3-6 and illustrated in Figure 3.

**Table 3.** Z-score results on FD001 to FD004

Dataset	Number of Detected Outliers
FD001	6342
FD002	0
FD003	3929
FD004	0

For Mahalanobis-based detection, the covariance structure was estimated from the normalized training data, so that correlations between sensor channels could be explicitly taken into account during outlier identification.



**Figure 3.** FD001 sensor signals before and after outlier removal. Green: original data, Red: after outlier handling

Table 6 shows the number of outliers using Mahalanobis and Figure 3 shows that outlier removal smooths T24/T30/T50 and stabilizes  $\phi$  and htBleed on FD001.

### 3.3 Data quality assessment

This section highlights the key steps taken into account during the assessment of prepared data quality. This involves two key steps. First, we evaluated the performance of the FNN to assess the impact of preprocessing methods on model performance. Second, we examined several metrics of data degradation, including monotonicity, trendability, and robustness, to ensure that the prepared data would be optimal for subsequent analysis and modeling steps.

#### 3.3.1 Evaluation of data preprocessing methods using a feedforward neural network

To obtain an initial and computationally efficient comparison of preprocessing strategies, we used a feedforward neural network (FNN) as a preliminary evaluation model. This choice was intended for relative comparison only, rather than as a substitute for validation on the final architecture. Because testing all preprocessing combinations directly on the proposed U-Net-TCN-Transformer model would be

computationally expensive, the FNN served as a lightweight screening model to identify the most promising preprocessing configurations. The compared methods were assessed using loss, Root Mean Squared Error (RMSE), and R<sup>2</sup>, which allowed for a preliminary estimate of their impact on data quality and predictive behavior.

**Table 4.** Interquartile Range (IQR) results on FD001 to FD004

Dataset	Number of Detected Outliers
FD001	28745
FD002	48169
FD003	49594
FD004	54892

**Table 5.** Percentile results on FD001 to FD004

Dataset	Number of Detected Outliers
FD001	5156
FD002	14400
FD003	4535
FD004	17876

**Table 6.** Outlier counts using Mahalanobis on FD001 to FD004

Dataset	Number of Detected Outliers
FD001	31
FD002	52
FD003	0
FD004	96

Table 7 presents the FNN’s performance on the FD001 dataset using various preprocessing methods, focusing on training data only. The original data serves as a baseline with a loss of 157.4487, RMSE of 9.9213, and R<sup>2</sup> of 0.9153, indicating noise in the input data. Wavelet denoising significantly enhances performance, achieving a loss of 39.9096, RMSE of 5.8291, and R<sup>2</sup> of 0.9669. Z-score, IQR, and percentiles also show notable improvements, while Mahalanobis distance offers intermediate results. The combined method using all preprocessing techniques shows substantial improvement with a loss of 50.9332, RMSE of 6.2539, and R<sup>2</sup> of 0.9623. After analyzing the results, we retained three methods: Denoising Wavelet, Zscore, and Mahalanobis. These methods provided the best performance, particularly Denoising Wavelet. The others, as shown in tables 4 and 5, removed too many outliers, leading to a loss of information and poorer predictive results compared to tables 3 and 8. These results should be interpreted as an initial selection step. A direct ablation on the final U-Net-TCN-Transformer architecture would further strengthen the evidence regarding the contribution of preprocessing to end-task performance.

**Table 7.** Impact of preprocessing methods on feedforward neural network performance (FD001)

Method	Loss	RMSE	R <sup>2</sup>
Original data	157.4487	9.9213	0.9153
Denoising wavelet	39.9096	5.8291	0.9669
Z-score	77.7301	7.0997	0.9457
IQR	115.6563	8.4137	0.9326
Percentile	110.9067	8.0539	0.9351
Mahalanobis	97.9172	7.7558	0.9398
All methods	50.9332	6.2539	0.9623

### 3.3.2 Data degradation metrics

It is crucial to identify a suitable prognostic parameter for evaluating data quality. Key characteristics to consider are monotonicity, prognosability, and trendability [32]. Monotonicity, which signifies a consistent positive or negative trend, is essential for monitoring degradation, particularly in systems that do not self-repair. While some components like batteries may temporarily self-repair when unused, the overall system typically shows a monotonic trend. Quality indicators for processed data are then evaluated following outlier detection methods.

#### a) Monotonicity:

Eq. (6) defines monotonicity, where  $n$  is the number of data samples and  $\#pos$  and  $\#neg$  are the counts of positive and negative derivatives, respectively. On a scale of 0 to 1, monotonicity is measured by the difference between these counts. A value near 1 indicates that  $F$  is strongly monotonic, exhibiting steady increases or decreases with few fluctuations in the opposite direction, whereas a value of 0 indicates that  $F$  is completely non-monotonic [32, 33].

$$mon(F) = \text{mean} \left( \left| \frac{\#pos \, d/dt - \#neg \, d/dt}{n - 1} \right| \right) \quad (6)$$

#### b) Trendability:

As shown in Eq. (7), the trendability metric, which ranges from 0 to 1, represents the linear correlation of  $F$ . The correlation coefficients in  $corrcoef(F)$  have absolute values between -1 and 1, with  $\min(|corrcoef(F)|)$  showing the weakest trend. Values near 1 indicate strong linear trendability, whereas values near 0 indicate weak trendability [32, 33].

$$tren(F) = \min(|corrcoef(F)|) \quad (7)$$

#### c) Robustness:

Robustness measures the degree of fluctuation in a feature vector, with values ranging from [0,1] [32]. The robustness is given in Eq. (8), where  $F$  represents the original feature vector, and  $\hat{F}$  is its trend component computed using the five-point triple smoothing method. The residuals,  $res = F - \hat{F}$ , quantify deviations from the trend. Here,  $n$  denotes the number of data points in  $F$ , normalizing the summation to compute the mean value across all elements. A robustness value close to 1 indicates strong robustness.

$$rob(F) = \text{mean} \left( \frac{1}{n} \sum_i \exp \left( - \left| \frac{res_i}{F_i} \right| \right) \right) \quad (8)$$

#### d) Prognosability:

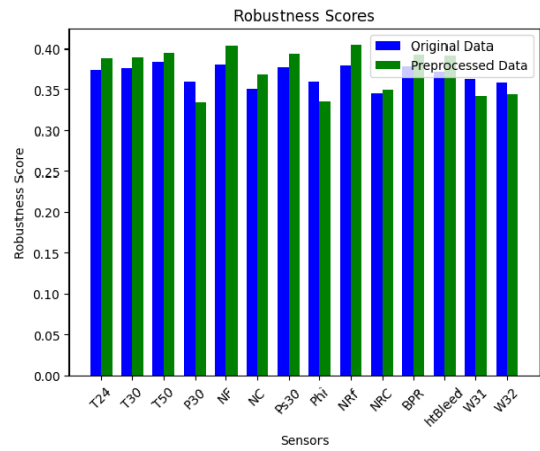
The ability to distinguish between functioning and malfunctioning equipment is known as prognosability, and it is measured by the health indicator's (HI) dispersion on a scale of 0 to 1. Better prognostic performance is indicated by higher values closer to 1. Eq. (9), in which  $F_0$  is the initial HI value and  $F_f$  is the value at failure, can be used to express it [32, 33].

$$pro(F) = \exp \left( - \frac{\text{std}(F_f)}{|\text{mean}(F_0) - \text{mean}(F_f)|} \right) \quad (9)$$

### 3.3.3 Detailed explanation of the outcome following preprocessing

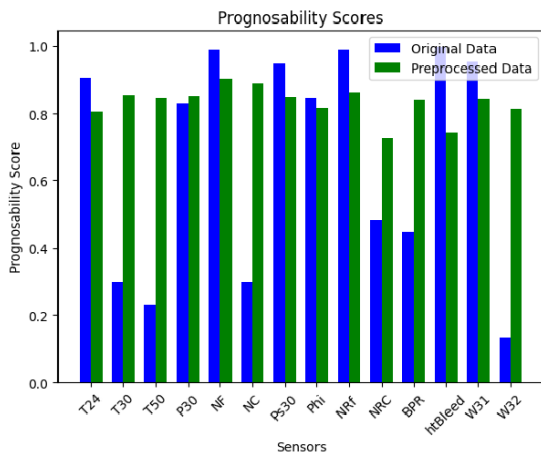
The preprocessing stage led to clearer degradation patterns

across the considered health indicators. As shown in Figure 4, the most substantial improvements were observed for prognosability and trendability, whose mean values increased while their dispersion decreased markedly after preprocessing. This indicates that the cleaned data better support the discrimination of degradation trajectories and the consistency of trend evolution across engines. Robustness showed a slight improvement, suggesting that the preprocessing pipeline reduced noise without destabilizing the feature behavior. Monotonicity remained broadly stable, with only a small decrease in average value, which may reflect the removal of local fluctuations together with part of the short-term variability. Overall, these results indicate that the proposed preprocessing pipeline improves the interpretability and prognostic usefulness of the C-MAPSS signals, especially in terms of trend consistency and failure separability.

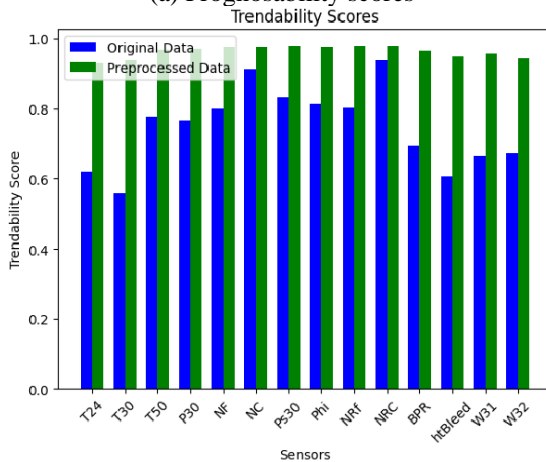


(d) Monotonicity scores

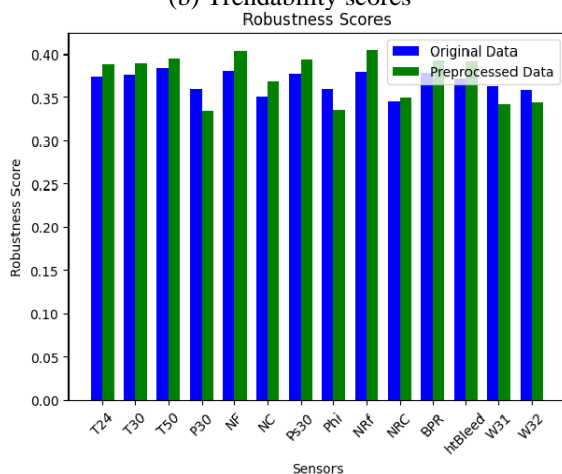
Figure 4. Comparison of prognostic quality metrics for the FD001 dataset before and after preprocessing



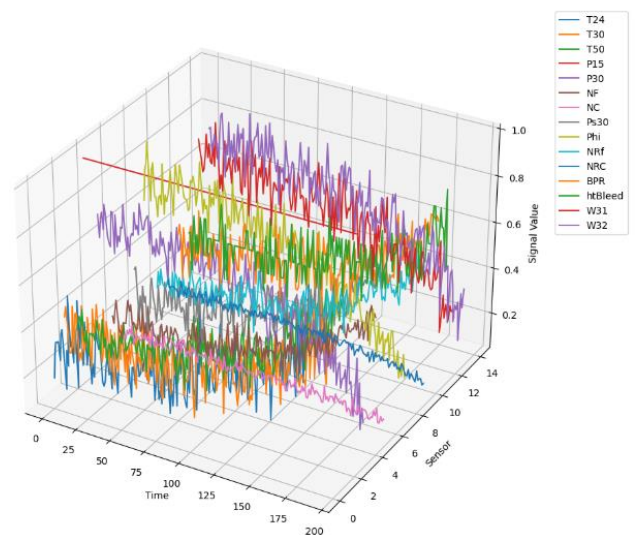
(a) Prognosability scores



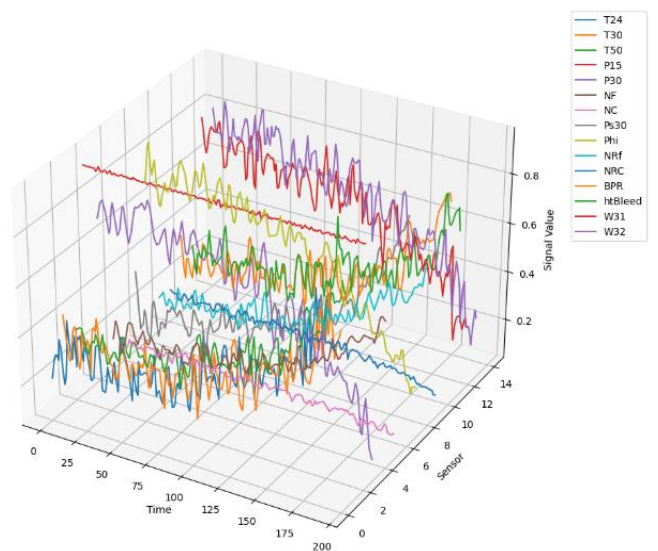
(b) Trendability scores



(c) Robustness scores



(a) Before deleting outliers



(b) After deleting outliers

Figure 5. 3D plot of FD001 sensor signals over cycles before and after deleting outliers

To further support these findings, Figure 5 shows the 3D representation of FD001 sensor signals over cycles before and

after preprocessing. The processed data reflect the application of the proposed preprocessing pipeline. The results show a reduction in irregular variations and an overall improvement in the smoothness of the sensor trajectories.

#### 4. METHODS

The methodology used for RUL prediction employs a U-Net architecture that integrates TCN and Transformers, preceded by a data preprocessing phase, as shown in Figure 6.

##### 4.1 U-Net TCN network overview

The U-Net-TCN architecture is designed to predict the RUL

in multivariate time series data. It leverages the hierarchical feature extraction capabilities of U-Net and the temporal modeling strengths of Temporal Convolutional Networks (TCNs). The model captures multi-scale temporal features, enabling robust predictions through its dense output layer. The encoder extracts essential features from the input time series using multiple TCN blocks. Each TCN block employs dilated 1D convolutions (Eq. (10)) to efficiently capture long-term dependencies. The dilation rates increase progressively, allowing the model to extract information at different time scales. Each TCN block includes batch normalization for stable training, ReLU activation (Eq. (11)) for non-linearity, and dropout (Eq. (12)) for regularization.

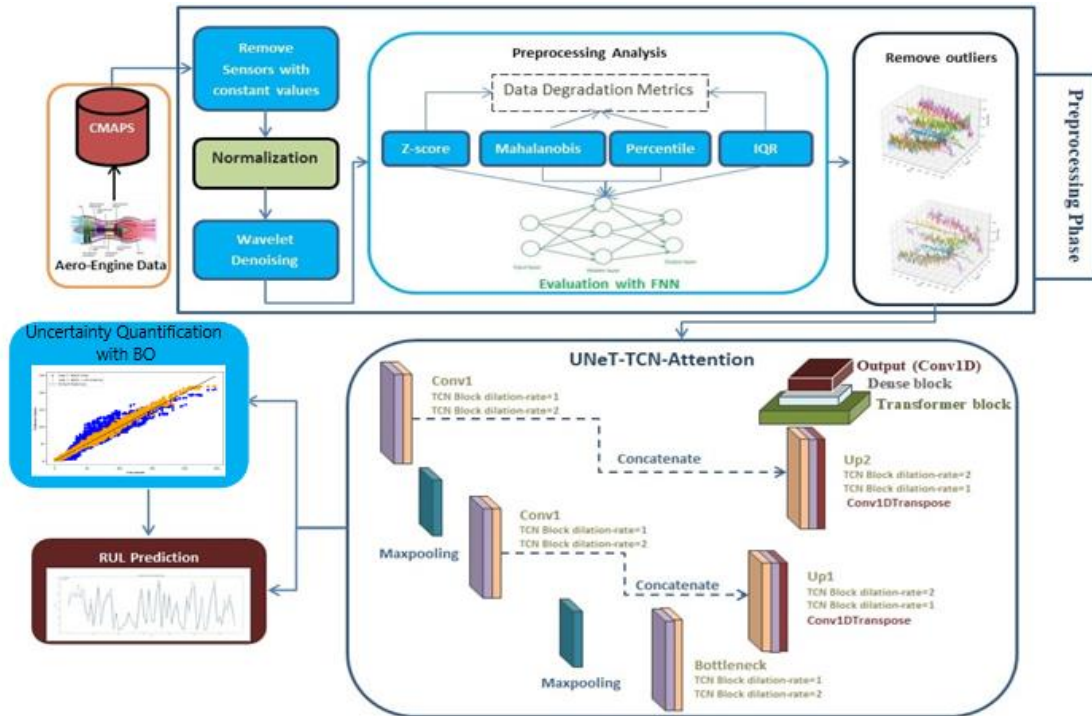


Figure 6. Pipeline architecture of the proposed method

The bottleneck reduces the dimensionality of the encoder’s high-dimensional representations, condensing them into a latent space. This step improves generalization and mitigates sensitivity to noise. Wider TCN filters are used here to capture higher-level abstractions of temporal features. The decoder reconstructs the temporal resolution of the reduced features using transposed convolutions. Skip connections between corresponding encoder and decoder layers preserve fine-grained details, ensuring accurate reconstruction. TCN blocks are integrated after each transposed convolution to refine the recovered features.

$$y[n] = \sum_{k=0}^{K-1} x[n - d \cdot k] \cdot w[k] \quad (10)$$

$$y = \max(0, x) \quad (11)$$

$$y = \begin{cases} 0 & \text{with probability } p, \\ \frac{x}{1-p} & \text{with probability } 1-p \end{cases} \quad (12)$$

##### 4.2 Attention integration

The Transformer block is integrated into the model to enhance its ability to capture global context. This block uses a multi-head self-attention mechanism (Eq. (13)) to dynamically focus on the most relevant parts of the input sequence, enabling the model to prioritize critical features and learn dependencies across time steps. The attention scores are calculated using a *softmax* function applied to the scaled dot-product of queries and keys, followed by a weighted sum of values. This mechanism enables the model to dynamically adjust its focus according to the relevance of different time steps.

Layer normalization (Eq. (14)) stabilizes training by standardizing outputs of the attention and dense layers. The Transformer block consists of multi-headed self-attention layers for sequence interaction, feedforward dense layers to add depth and non-linearity, and normalization for stable gradient flow. This integration ensures that the model captures both fine-grained temporal patterns and global contextual relationships, making it highly effective for RUL prediction.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

$$\hat{x} = \frac{x - \mu}{\sigma} \quad (14)$$

### 4.3 Model architecture

The U-Net-TCN-Attention model, as illustrated in Figure 7, is a unified architecture optimized for multivariate time series prediction. It combines the strengths of U-Net, TCNs, and Transformers into a cohesive framework. The encoder extracts multi-scale features, the bottleneck condenses them, and the decoder reconstructs the temporal resolution using skip connections and TCN blocks. Positioned after the decoder, the Transformer block, shown in Figure 8, uses self-attention to capture global dependencies, ensuring the model prioritizes relevant temporal patterns.

A dense block with 128 units and ReLU activation refines the features for the final prediction. L2 regularization is applied to prevent overfitting (Eq. (15)). The output layer is a 1D convolutional layer with linear activation, optimized for regression tasks. The model is trained using the mean squared error (MSE) loss function and evaluated using mean absolute error (MAE) and RMSE metrics. This ensures robust performance for the time-series forecasting tasks.

$$y = W \cdot x + b \quad (15)$$

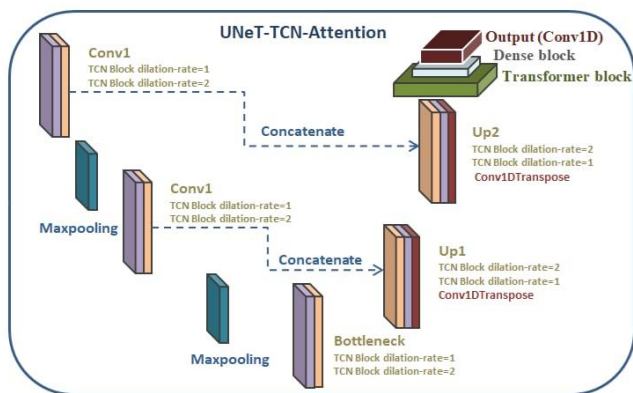


Figure 7. U-Net-TCN-Transformer architecture

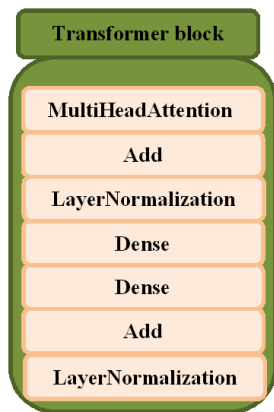


Figure 8. Transformer block diagram

### 4.4 U-Net-TCN-Attention motivations

The hybrid U-Net-TCN-Transformer architecture is

specifically engineered to address the heterogeneous nature of aero-engine degradation. The U-Net module serves as a hierarchical feature extractor, capturing multi-resolution patterns that traditional networks might miss. These features are subsequently processed by TCN layers, which utilize dilated convolutions to extend the receptive field, thereby capturing long-term temporal dependencies without the vanishing gradient issues of RNNs. Finally, the Transformer block applies a multi-head self-attention mechanism to weight the global significance of specific operational cycles. This tripartite integration ensures that the model is sensitive to both localized sensor fluctuations and long-term health trends.

### 4.5 Bayesian optimization under uncertainty quantification (UQ)

In fields such as maintenance, decision-making depends heavily on understanding the uncertainty surrounding a model's predictions. Ignoring uncertainty can lead to overconfidence in predictions, which can be especially risky in high-stakes situations. Uncertainty Quantification is widely used in many real-world applications [34], and it is generally categorized into two types, aleatoric and epistemic uncertainties [23]. In this paper, we aim to improve the performance and reliability of the U-Net-TCN-Attention model by using an uncertainty-aware optimization strategy to account for epistemic uncertainty. The proposed method combines high predictive accuracy with uncertainty measures to gain model reliability in high-stakes applications that rely critically on confidence in prediction. By contrasting the optimization method that will be applied and this method, we demonstrate how integrating uncertainty within the optimization adds value.

#### 4.5.1 Bayesian optimization

Hyperparameter tuning methods, such as grid search and random search, are commonly used but may fail to efficiently explore complex models, like U-Net-TCN-Attention. To address this, we use Bayesian Optimization (BO), which adaptively refines hyperparameters based on past evaluations. Unlike traditional approaches, BO leverages a probabilistic model, often Gaussian Processes, to approximate the objective function. This allows the algorithm to identify promising configurations by balancing exploration and exploitation through an acquisition function. In this analysis, BO optimizes several hyperparameters for the UNet-TCN model. These include model width, which controls the number of convolutional filters and directly impacts the model's capacity; kernel size, which determines the receptive field of the convolutional layers and affects feature extraction; dropout rate, a regularization technique used to prevent overfitting; and learning rate, which ensures that the optimization process remains stable and effective. The primary goal is to minimize the RMSE on validation data, thereby enhancing predictive accuracy efficiently. However, it is important to note that while BO can improve accuracy, it does not take into account the model's confidence in its predictions, which can be a significant limitation in certain applications.

#### 4.5.2 Objective function and uncertainty quantification

The goal of uncertainty-aware optimization is to create models that are not only accurate but also reliable and interpretable. To this end, we incorporate uncertainty quantification via a Z-score confidence interval into the model

optimization process. Z-score confidence interval allows us to assess the interval with the average width, we define a customized objective function, which balances both predictive accuracy and uncertainty quantification measures. Predictive Accuracy, to be minimized, is represented by the RMSE between predictions and actual values. Uncertainty Quantification is assessed based on the average width of the CIs, where narrower intervals mean higher confidence. Prediction Reliability: Coverage refers to the proportion of actual values falling within the predicted confidence intervals. Now, our customized objective function is represented according to Eq. (16).

$$ObjS = RMSE + \alpha \cdot Average\ CI_w + \beta \cdot (1 - C) \quad (16)$$

where,  $\alpha$  and  $\beta$  are weights that balance precision, uncertainty, and reliability. Average  $CI_w$  is the average CI Width,  $ObjS$  presents the Objective Score, and  $C$  is the coverage. By optimizing this function, we can identify a configuration of hyperparameters that minimizes error while effectively calibrating uncertainty. This is motivated by the growing awareness that uncertainty-aware models are crucial for understanding the reliability of predictions and therefore making better decisions. Bayesian Optimization is used to efficiently optimize this customized objective function in a manner that can balance accuracy and uncertainty without excessive computational overhead.

In this study, confidence intervals were computed at the 95% level. Interval quality was assessed using two complementary criteria: average CI width and empirical coverage, defined as the proportion of true RUL values falling inside the predicted intervals over the evaluation set. The

uncertainty-aware optimization was therefore designed to favor configurations that maintain strong predictive accuracy while avoiding poorly calibrated or excessively wide intervals.

#### 4.6 Implementation details

For all C-MAPSS subsets (FD001–FD004), input time-series sequences were generated using a sliding-window approach with a window length of 110 cycles and a stride of 1 cycle, capturing both short- and long-term degradation patterns while maximizing training samples.

RUL targets were calculated as the difference between each engine’s maximum cycle and its current cycle under a linear degradation assumption. No RUL capping was applied to preserve the full degradation trajectory and avoid early-life saturation effects.

An engine-wise split assigned 20% of engines to the validation set, preventing data leakage, with a fixed random seed ensuring reproducibility.

The UNet-TCN-Transformer model was trained with the Adam optimizer (learning rate 0.001), batch size 32, and up to 100 epochs. Early stopping with a patience of 10 epochs based on validation RMSE prevented overfitting. Dropout was applied after the main layers for regularization, with rates tuned per subset: 0.1 (FD001, FD003), 0.01 (FD002), and 0.001 (FD004).

The TCN encoder used temporal convolutional blocks with dilation rates of 1, 2, 4, 8, and 16, while the Transformer module employed one encoder layer, four attention heads, an embedding dimension of 128, and positional encoding to retain temporal order.

**Table 8.** Hyperparameter values of the U-Net-TCN-Transformer model

	FD001	FD002	FD003	FD004	
<b>Sequence Length (Length)</b>	110	110	110	110	The input sequence length.
<b>Number of Input Channels (Num Channel)</b>	18	25	20	25	Number of features or sensors used as input features for the model.
<b>Model Width (Model Width)</b>	60	61	60	61	The number of filters in the first convolutional layer.
<b>Kernel Size</b>	3	3*	3	3	The size of the convolutional kernel used in TCN layers.
<b>Dropout Rate</b>	0.1	0.01	0.1	0.001	The dropout rate is applied after the main layers to regularize the model and prevent overfitting.
<b>L2 Lambda</b>	0.1	0.1	0.1	0.1	Coefficient for L2 regularization penalty, helping to constrain weight values and prevent overfitting.

Experiments were implemented in Python using TensorFlow/Keras on a GPU workstation. Each experiment was repeated five times with different random seeds, and the reported results correspond to the average performance across runs. Table 8 summarizes the hyperparameters for each subset, ensuring optimal performance with a consistent training strategy.

## 5. RESULTS

### 5.1 Model architecture and configuration

There is no universal set of optimal parameters for the entire data set. The best parameter values vary for each subunit [12].

Table 8 displays the settings for the U-Net-TCN-Attention model instance. This includes several key parameters such as the input sequence length, the number of features, model width,

kernel size, and dropout rate. The important parameters are as follows:

### 5.2 Evaluation metrics for model performance

The performance of the proposed model was evaluated using MSE, RMSE, and the coefficient of determination ( $R^2$ ) [35, 36], defined in Eqs. (17)-(19), respectively.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (17)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (18)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n \sum_{i=1}^n (y_i - \bar{y})^2} \quad (19)$$

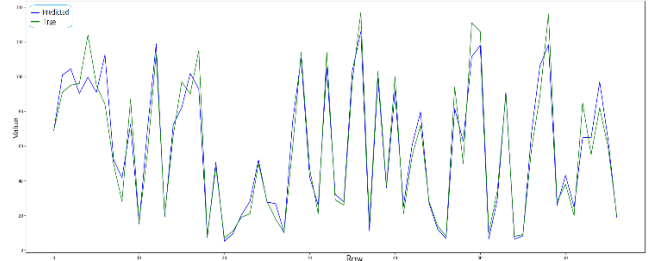
where,  $\hat{y}_i$  is the predicted RUL,  $y_i$  the true RUL,  $\bar{y}$  the mean of true values, and  $N$  the number of test samples. MSE reflects average squared error, RMSE emphasizes larger deviations, and  $R^2$  evaluates the proportion of variance explained by the model.

### 5.3 Results and discussion

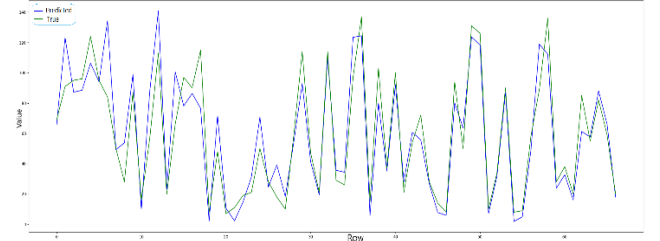
The proposed model achieved strong predictive performance across the four C-MAPSS subsets. For FD001, it obtained an RMSE of 7.63, an MAE of 5.29, and an  $R^2$  of 0.9531. On FD002, which involves more complex operating conditions, the RMSE was 12.39. For FD003 and FD004, the model achieved RMSE values of 10.73 and 12.67, respectively. Relative to several previously reported studies [16, 22, 37], these results are numerically competitive and are better than a number of published baselines on the same subsets. However, such cross-paper comparisons should be interpreted with caution, since preprocessing choices, RUL truncation strategies, training procedures, and evaluation settings may differ across studies [16, 37].

A plausible explanation for these results is that the proposed architecture combines complementary modeling strengths: U-Net supports multi-scale feature extraction, TCN captures long-range temporal dependencies through dilated convolutions, and self-attention highlights globally relevant temporal patterns. Figures 9 and 10 show the prediction results, suggesting that the model captures degradation trends effectively on C-MAPSS. Table 9 situates the proposed model relative to previously published results on C-MAPSS and shows that its RMSE values are competitive across the four subsets. As reported in Table 10, the proposed model yields higher  $R^2$  values than [16] on FD001, FD003, and FD004, while remaining close on FD002.

In summary, the proposed U-Net-TCN-Attention model delivers competitive and consistent results across the four C-MAPSS subsets. These findings suggest that the combination of temporal convolutions, skip-connected multi-scale encoding, and attention-based context modeling is effective for aero-engine RUL prediction on C-MAPSS.

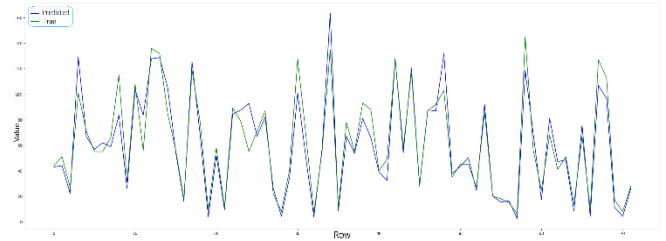


(a) FD001

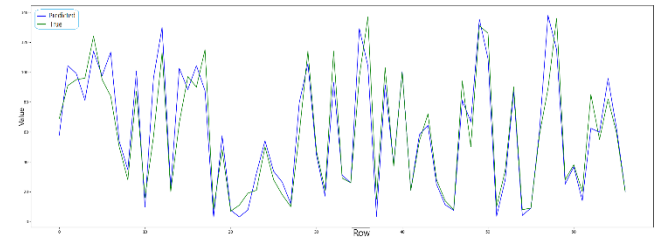


(b) FD002

**Figure 9.** Predicted vs. True RUL for FD001 and FD002



(a) FD003



(b) FD004

**Figure 10.** Predicted vs. True RUL for FD003 and FD004

**Table 9.** Comparison of Root Mean Squared Error (RMSE) across C-MAPSS subsets

Model	Year	FD001 (RMSE, %)	FD002 (RMSE, %)	FD003 (RMSE, %)	FD004 (RMSE, %)
[35]	2016	18.45 (58.64%)	30.29 (59.11%)	19.81 (45.82%)	29.15 (56.54%)
[36]	2017	16.14 (52.71%)	24.49 (49.42%)	16.18 (33.68%)	28.17 (55.03%)
[15]	2024	15.97 (52.21%)	14.45 (14.31%)	13.90 (22.82%)	16.63 (23.83%)
[37]	2021	14.90 (48.77%)	15.19 (18.46%)	16.71 (35.80%)	19.74 (35.83%)
[11]	2018	14.89 (48.73%)	26.86 (53.87%)	15.11 (28.98%)	27.11 (53.27%)
[22]	2022	11.47 (33.45%)	16.15 (23.29%)	10.74 (0.08%)	18.90 (33.00%)
[38]	2023	10.98 (30.50%)	16.12 (23.17%)	11.14 (3.67%)	18.15 (30.23%)
[16]	2021	10.60 (28.00%)	14.55 (14.89%)	11.71 (8.35%)	17.23 (26.49%)
[17]	2023	10.35 (26.26%)	15.82 (21.73%)	11.34 (5.36%)	17.35 (27.01%)
[39]	2021	8.68 (12.06%)	/	9.69 (-10.77%)	/
[12]	2022	7.78 (1.88%)	17.64 (29.80%)	8.03 (-33.60%)	17.63 (28.16%)
Proposed	2025	7.63	12.39	10.73	12.67

**Table 10.**  $R^2$  score comparison with a recent baseline

Model / Dataset	FD001	FD002	FD003	FD004
[16]	0.93	0.88	0.91	0.83
Proposed (U-Net-TCN-Attention)	0.9531	0.8654	0.9225	0.8702

**Table 11.** Ablation study results on FD001

Configuration	RMSE (mean $\pm$ std)	MAE (mean $\pm$ std)	R <sup>2</sup> (mean $\pm$ std)	Score (mean $\pm$ std)
Raw Data	8.2120 $\pm$ 1.4759	7.4102 $\pm$ 1.4706	0.9358 $\pm$ 0.0236	4335.71 $\pm$ 1245.09
Denosing only	7.1179 $\pm$ 1.3081	6.2005 $\pm$ 1.4098	0.9517 $\pm$ 0.0157	3402.38 $\pm$ 898.33
Denosing + Outlier	7.1776 $\pm$ 2.0390	5.1903 $\pm$ 1.7017	0.9491 $\pm$ 0.0259	4856.75 $\pm$ 3576.62
Full Pipeline	6.8608 $\pm$ 1.7979	5.6055 $\pm$ 1.7319	0.9539 $\pm$ 0.0233	3286.60 $\pm$ 1480.55

**Table 12.** Statistical significance of preprocessing configurations on FD001 using paired t-tests

Configuration	p-Value	Significant	Improvement	Valid Runs
Denosing only	0.0974	X	+13.3%	5/5
Denosing + Outlier	0.5025	X	+12.6%	5/5
Full Pipeline	0.1925	X	+16.5%	5/5

**Table 13.** Ablation study results on FD004

Configuration	RMSE (mean $\pm$ std)	MAE (mean $\pm$ std)	R <sup>2</sup> (mean $\pm$ std)	Score (mean $\pm$ std)
Raw Data	12.1940 $\pm$ 3.7390	7.6250 $\pm$ 2.4364	0.9651 $\pm$ 0.0177	110812.12 $\pm$ 87044.59
Denosing only	7.6609 $\pm$ 4.7194	4.7058 $\pm$ 3.0880	0.9833 $\pm$ 0.0164	36855.99 $\pm$ 42168.95
Denosing + Outlier	10.0192 $\pm$ 2.1063	6.3558 $\pm$ 1.6863	0.9773 $\pm$ 0.0083	32441.73 $\pm$ 14510.38
Full Pipeline	8.8577 $\pm$ 2.9039	5.2190 $\pm$ 1.7445	0.9814 $\pm$ 0.0115	143326.48 $\pm$ 274782.53

**Table 14.** Statistical significance of preprocessing configurations on FD004 using paired t-tests

Configuration	p-Value	Significant	Improvement	Valid Runs
Denosing only	0.1014	X	+37.2%	5/5
Denosing + Outlier	0.3211	X	+17.8%	5/5
Full Pipeline	0.2550	X	+27.4%	5/5

#### 5.4 Ablation study of the preprocessing pipeline

To further investigate the impact of the preprocessing pipeline on overall predictive performance, an ablation study was conducted. The objective was to isolate and quantify the contribution of each preprocessing step, namely denosing and outlier handling, as well as their combined effect. Unlike the preliminary evaluation performed using a feedforward neural network, this study uses the final U-Net-TCN-Transformer architecture, providing a more reliable assessment of their impact on the final prediction task.

The ablation experiments were conducted on two representative subsets of the C-MAPSS dataset, namely FD001 and FD004, which were selected due to their different levels of complexity. FD001 corresponds to a single operating condition with relatively simple degradation patterns, whereas FD004 includes multiple operating conditions and fault modes, making it significantly more complex. This selection allows evaluating the robustness of the preprocessing pipeline under both simple and complex operating scenarios. In all ablation experiments, the RUL target was modeled using a linear, non-capped formulation to preserve the full degradation trajectory and avoid introducing artificial saturation effects.

The ablation results for FD001 are presented in Table 11. The baseline model trained on raw data achieves an RMSE of 8.21 and an R<sup>2</sup> of 0.9358. Applying wavelet-based denosing reduces the RMSE to 7.12 and increases R<sup>2</sup> to 0.9517, indicating that noise reduction improves the model's ability to capture degradation trends. The combination of denosing and outlier handling produces mixed results, slightly improving MAE but not consistently outperforming denosing alone. The full preprocessing pipeline achieves the best overall performance, with an RMSE of 6.86 and an R<sup>2</sup> of 0.9539, corresponding to an improvement of approximately 16.5% compared to the raw data baseline. However, statistical

analysis using paired t-tests shows that these improvements are not statistically significant at conventional confidence levels ( $p > 0.05$ ), indicating that performance variability across runs remains non-negligible, as reported in Table 12.

The ablation results for FD004 are presented in Table 13. In this more complex scenario, the baseline RMSE is 12.19 with an R<sup>2</sup> of 0.9651. Denosing alone significantly improves predictive accuracy, reducing RMSE to 7.66 and increasing R<sup>2</sup> to 0.9833, highlighting the importance of noise reduction in complex operating conditions. The combination of denosing and outlier removal does not consistently improve performance, suggesting that outlier removal may sometimes eliminate informative variations. The full preprocessing pipeline achieves an RMSE of 8.86 and an R<sup>2</sup> of 0.9814, corresponding to an improvement of 27.4% compared to the baseline. Nevertheless, as represented in Table 14, statistical tests again indicate that these improvements are not statistically significant ( $p > 0.05$ ), which may be explained by the higher variability of FD004 due to multiple operating conditions and more complex degradation dynamics.

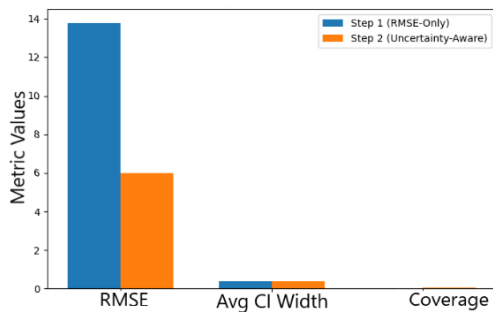
Overall, the ablation study provides several important insights. First, denosing consistently contributes to performance improvement across both datasets, confirming its importance for signal quality enhancement. Second, the impact of outlier removal is dataset-dependent and does not always lead to systematic improvements, particularly in complex scenarios where extreme values may contain useful degradation information. Third, the full preprocessing pipeline generally achieves the best average performance, although the improvements are not always statistically significant. These results support the integration of preprocessing in the proposed framework while indicating that the main performance gains arise from the combination of data quality enhancement and the hybrid U-Net-TCN-Transformer architecture.

### 5.5 Uncertainty analysis

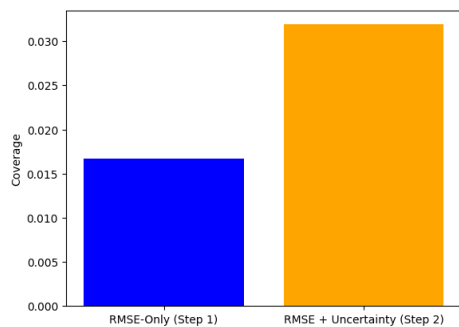
This work is inspired by the use of confidence intervals in many research studies [40, 41]. The two optimized U-Net-TCN models are evaluated based on three crucial aspects. First, their accuracy is assessed by measuring the RMSE on the validation set. Next, the models' uncertainty calibration is examined, focusing on the average Confidence Interval (CI) width and coverage. Lastly, the reliability of the models is determined by how well the predicted uncertainties align with the actual errors. The results, illustrated in Figures 11(a), 11(b) and 12, highlight the advantages of uncertainty optimization over standard approaches that prioritize accuracy alone. Confidence intervals and coverage metrics are used to quantify uncertainty, while the proposed optimization framework leverages these estimates to enable uncertainty-aware decision-making.

By explicitly modeling uncertainty during the optimization process, this U-Net-TCN model not only delivers accurate results but also provides predictions with an associated confidence level. This enables models to be more robust and interpretable, which is particularly valuable in real-world applications where understanding confidence in predictions is as crucial as achieving high performance.

Figure 12 compares predicted vs. true RUL for BO vs. UABO, complementing the CI and coverage views.



(a) predicted vs true RUL

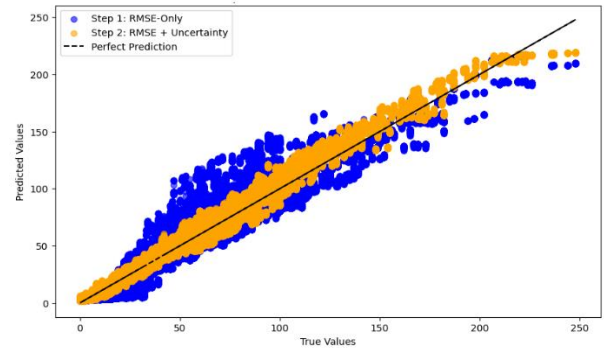


(b) empirical coverage

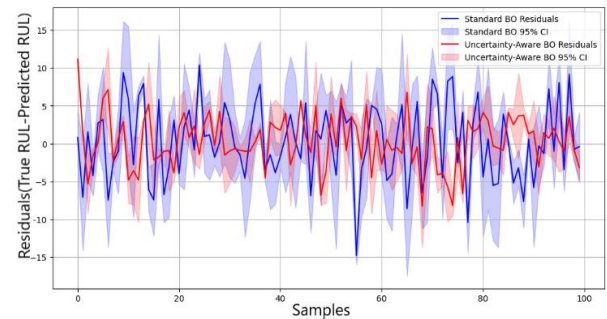
**Figure 11.** Comparison of standard Bayesian Optimization (BO) and uncertainty-aware optimization (UABO): (a) predicted vs true Remaining Useful Life (RUL), (b) empirical coverage

The uncertainty-aware optimization produced a more favorable trade-off between predictive accuracy and interval quality than RMSE-only optimization. As shown in Figures 11-13, the uncertainty-aware configuration reduced residual dispersion, improved empirical coverage, and yielded intervals that were better aligned with the evolution of

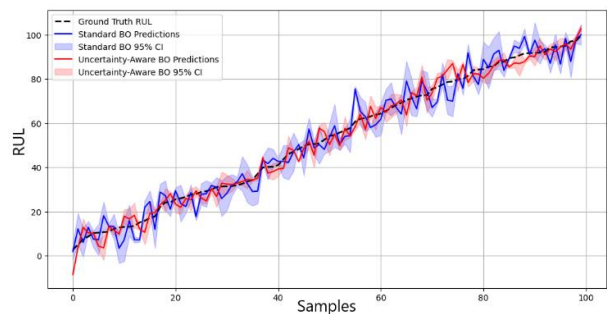
prediction difficulty. In particular, intervals remained relatively tight in stable regimes while widening near more uncertain degradation phases. This behavior is desirable for prognostics because it limits overconfident predictions while preserving decision-relevant precision. Overall, these results suggest that UABO improves not only point-estimate performance, but also the reliability and interpretability of the predicted RUL intervals.



**Figure 12.** Comparison of true and predicted values for Bayesian and uncertainty Bayesian methods



(a) Confidence Intervals of residuals



(b) Confidence Intervals of RUL

**Figure 13.** Confidence interval comparison for Bayesian Optimisation (BO) and uncertainty-aware Bayesian Optimisation (UABO)

Table 15 summarizes these improvements quantitatively, reporting RMSE, average confidence interval width, and coverage for both optimization strategies on FD001 and FD002.

Additionally, at lower RUL values (e.g., True RUL 10), Uncertainty Aware BO avoids unrealistic predictions (e.g., 8 with a CI of 5–11), contrasting with Standard BO's nonsensical -5 prediction and overly broad CI (-15 to 20). These results collectively highlight that Uncertainty-Aware BO reduces residual errors, tightens confidence intervals, avoids invalid predictions, and improves coverage of True

RUL values, thereby enhancing reliability and precision in RUL prediction compared to Standard BO. In conclusion, while the improvements in RMSE and coverage between Step 1 and Step 2 may seem modest, the figures focused on BO demonstrate the significant advantages of uncertainty-aware modeling. Uncertainty-aware BO reduces residuals (for example,  $-3$  compared to  $-12$  at Sample 50), tightens confidence intervals (CIs), with a width of 6 units versus 20 units, eliminates invalid predictions (such as negative remaining useful life values), and enhances calibration (95% coverage compared to 80%).

**Table 15.** Quantitative comparison of standard Bayesian Optimization (BO) and uncertainty-aware Bayesian Optimization (UABO)

Subset	Method	RMSE	Avg CI Width	Coverage (95%)
FD001	BO (RMSE-only)	7.12	4.23	0.82
	UABO (Proposed)	6.86	3.15	0.94
FD002	BO (RMSE-only)	12.58	5.89	0.79
	UABO (Proposed)	12.39	4.52	0.92

Beyond point-estimate accuracy, the calibration of the uncertainty intervals is a key indicator of system reliability. As illustrated in Figure 13, the uncertainty-aware optimization effectively constrains the 95% CIs during stable operation phases, while appropriately widening them near failure points. This behavior prevents the generation of physically implausible RUL values (e.g., negative cycles) and provides a quantified 'safety margin' for maintenance engineers, a feature often absent in deterministic deep learning models.

These improvements are crucial for applications like predictive maintenance, where unreliable intervals (for instance, Standard BO's range of  $-15$  to  $20$ ) can lead to costly misinterpretations. The results highlight that incorporating uncertainty quantification can contribute significantly to the reliability and actionability of predictive systems in such contexts.

## 6. CONCLUSIONS

This study proposed an integrated framework for RUL prediction of aero-engines on C-MAPSS, combining data quality enhancement, a hybrid U-Net-TCN-Transformer architecture, and uncertainty-aware Bayesian optimization. These components are organized as a three-layer pipeline, where data quality enhancement feeds into hybrid temporal modeling, which in turn supplies uncertainty-aware optimization, forming a coherent prognostic chain.

The preprocessing pipeline improved signal usability through denoising, outlier handling, and feature selection guided by prognostic indicators, while the hybrid architecture captured both local temporal patterns and broader contextual dependencies. Experimental results across all four C-MAPSS subsets demonstrated competitive predictive performance, and uncertainty-aware optimization enhanced the quality and interpretability of prediction intervals. Despite these encouraging results, the study is limited to benchmark datasets and cross-paper comparisons, which may be influenced by

differing experimental conditions.

An ablation study highlighted the contribution of the preprocessing pipeline: denoising consistently improved accuracy, while the impact of outlier removal was dataset-dependent. While the full preprocessing pipeline generally improved average performance, these gains were not always statistically significant, suggesting that the primary contribution to predictive accuracy arises from the hybrid U-Net-TCN-Transformer architecture.

Future work will focus on three directions: first, conducting a comprehensive ablation of the U-Net-TCN-Transformer architecture to quantify the contribution of each component; second, improving uncertainty calibration via alternative interval estimation methods and repeated-run analyses; and third, developing adaptive preprocessing strategies where parameters such as denoising strength, outlier thresholds, and window sizes are dynamically adjusted or optimized through Bayesian methods.

The proposed framework delivers robust, accurate, and uncertainty-aware RUL predictions, highlighting the value of combining data quality enhancement with hybrid deep learning for real-world maintenance decision support.

## ACKNOWLEDGEMENT

This work is fully funded by the General Direction of Scientific Research and Technological Development, Algerian Ministry of Higher Education and Scientific Research DGRSDT/MESRS.

## REFERENCES

- [1] Berghout, T., Benbouzid, M. (2022). A systematic guide for predicting remaining useful life with machine learning. *Electronics*, 11(7): 1125. <https://doi.org/10.3390/electronics11071125>
- [2] Lei, Y., Li, N., Guo, L., Li, N., Yan, T., Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104: 799-834. <https://doi.org/10.1016/j.ymsp.2017.11.016>
- [3] Jing, T., Zheng, P., Xia, L., Liu, T. (2022). Transformer-based hierarchical latent space VAE for interpretable remaining useful life prediction. *Advanced Engineering Informatics*, 54: 101781. <https://doi.org/10.1016/j.aei.2022.101781>
- [4] Zhang, X., Sun, J., Wang, J., Jin, Y., Wang, L., Liu, Z. (2023). Paoltransformer: Pruningadaptive optimal lightweight transformer model for aero-engine remaining useful life prediction. *Reliability Engineering & System Safety*, 240: 109605. <https://doi.org/10.1016/j.res.2023.109605>
- [5] Xiang, F., Zhang, Y., Zhang, S., Wang, Z., Qiu, L., Choi, J.H. (2024). Bayesian gated transformer model for risk-aware prediction of aero-engine remaining useful life. *Expert Systems with Applications*, 238: 121859. <https://doi.org/10.1016/j.eswa.2023.121859>
- [6] Berghout, T., Amirat, Y., Diallo, D., Lim, W.H., Benbouzid, M. (2024). Improving data quality for prognostic learning systems considering complex degradation patterns. In *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*,

- Chicago, IL, USA, pp. 1-6. <https://doi.org/10.1109/IECON55916.2024.10905515>
- [7] Baptista, M.L., Henriques, E.M. (2022). 1D-DGAN-PHM: A 1-D denoising GAN for prognostics and health management with an application to turbofan. *Applied Soft Computing*, 131: 109785. <https://doi.org/10.1016/j.asoc.2022.109785>
- [8] Lin, Y.H., Chang, L. (2021). An unsupervised noisy sample detection method for deep learning-based health status prediction. *IEEE Transactions on Instrumentation and Measurement*, 71: 1-11. <https://doi.org/10.1109/TIM.2021.3132374>
- [9] Xia, M., Zheng, X., Imran, M., Shoaib, M. (2020). Data-driven prognosis method using hybrid deep recurrent neural network. *Applied Soft Computing*, 93: 106351. <https://doi.org/10.1016/j.asoc.2020.106351>
- [10] Chui, K.T., Gupta, B.B., Vasant, P. (2021). A genetic algorithm optimized RNN-LSTM model for remaining useful life prediction of turbofan engine. *Electronics* 10(3): 285. <https://doi.org/10.3390/electronics10030285>
- [11] Liao, Y., Zhang, L., Liu, C. (2018). Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method. In 2018 IEEE International Conference on Prognostics and Health Management (ICPHM), Seattle, WA, USA, pp. 1-8. <https://doi.org/10.1109/ICPHM.2018.8448804>
- [12] Asif, O., Haider, S.A., Naqvi, S.R., Zaki, J.F., Kwak, K.S., Islam, S.R. (2022). A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset. *IEEE Access*, 10: 95425-95440. <https://doi.org/10.1109/ACCESS.2022.3203406>
- [13] Jiang, Z., Zhao, Y.X., Yu, W. (2025). A remaining useful life prediction method based on CNN-BiLSTM feature transfer in a high-noise environment. *Sound & Vibration*, 59(1): 1685. <https://doi.org/10.59400/sv.v59i1.1685>
- [14] Abderrezek, S., Bourouis, A. (2021). Convolutional autoencoder and bidirectional long short-term memory to estimate remaining useful life for condition based maintenance. In 2021 International Conference on Networking and Advanced Systems (ICNAS), Annaba, Algeria, pp. 1-6. <https://doi.org/10.1109/ICNAS53565.2021.9628958>
- [15] Deng, S., Zhou, J. (2024). Prediction of remaining useful life of aero-engines based on CNN-LSTM-attention. *International Journal of Computational Intelligence Systems*, 17(1): 232. <https://doi.org/10.1007/s44196-024-00639-w>
- [16] Tan, W.M., Teo, T.H. (2021). Remaining useful life prediction using temporal convolution with attention. *Ai*, 2(1): 48-70. <https://doi.org/10.3390/ai2010005>
- [17] Wang, X., Li, Y., Xu, Y., Liu, X., Zheng, T., Zheng, B. (2023). Remaining useful life prediction for aero-engines using a time-enhanced multi-head self-attention model. *Aerospace*, 10(1): 80. <https://doi.org/10.3390/aerospace10010080>
- [18] Nie, L., Xu, S., Zhang, L., Yin, Y., Dong, Z., Zhou, X. (2022). Remaining useful life prediction of aeroengines based on multi-head attention mechanism. *Machines*, 10(7): 552. <https://doi.org/10.3390/machines10070552>
- [19] Wahid, A., Yahya, M., Breslin, J.G., Intizar, M.A. (2023). Self-attention transformer-based architecture for remaining useful life estimation of complex machines. *Procedia Computer Science*, 217: 456-464. <https://doi.org/10.1016/j.procs.2022.12.241>
- [20] Fan, X., Zhang, W., Zhang, C., Chen, A., An, F. (2022). SOC estimation of Li-ion battery using convolutional neural network with U-Net architecture. *Energy*, 256: 124612. <https://doi.org/10.1016/j.energy.2022.124612>
- [21] Suh, S., Lukowicz, P., Lee, Y.O. (2022). Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks. *Knowledge-Based Systems*, 237: 107866. <https://doi.org/10.1016/j.knosys.2021.107866>
- [22] Wang, S., Ji, B., Wang, W., Ma, J., Chen, H.B. (2022). Remaining useful life prediction of aero-engine based on deep convolutional LSTM network. In 2022 6th International Conference on System Reliability and Safety (ICSRS), Venice, Italy, pp. 494-499. <https://doi.org/10.1109/ICSRS56243.2022.10067647>
- [23] Hullermeier, E., Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* 110(3): 457-506. <https://doi.org/10.1007/s10994-021-05946-3>
- [24] Saxena, A., Goebel, K., Simon, D., Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 International Conference on Prognostics and Health Management, pp. 1-9. <https://doi.org/10.1109/PHM.2008.4711414>
- [25] Hong, C.W., Lee, C., Lee, K., Ko, M.S., Kim, D.E., Hur, K. (2020). Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors*, 20(22): 6626. <https://doi.org/10.3390/s20226626>
- [26] Helm, D., Timusk, M. (2023). Wavelet denoising applied to hardware redundant systems for rolling element bearing fault detection. *Journal of Dynamics, Monitoring and Diagnostics*, 102-114. <https://doi.org/10.37965/jdmd.2023.231>
- [27] Paul, A., Kundu, A., Chaki, N., Dutta, D., Jha, C. (2022). Wavelet enabled convolutional autoencoder based deep neural network for hyperspectral image denoising. *Multimedia Tools and Applications*, 81(2): 2529-2555. <https://doi.org/10.1007/s11042-021-11689>
- [28] Halidou, A., Mohamadou, Y., Ari, A.A.A., Zacko, E.J.G. (2023). Review of wavelet denoising algorithms. *Multimedia Tools and Applications*, 82(27): 41539-41569. <https://doi.org/10.1007/s11042-023-15127-0>
- [29] Bektas, O., Jones, J.A., Sankararaman, S., Roychoudhury, I., Goebel, K. (2019). A neural network filtering approach for similarity-based remaining useful life estimation. *The International Journal of Advanced Manufacturing Technology*, 101: 87-103. <https://doi.org/10.1007/s00170-018-2874-0>
- [30] Ahwiadi, M., Wang, W. (2018). An enhanced mutated particle filter technique for system state estimation and battery life prediction. *IEEE Transactions on Instrumentation and Measurement*, 68(3): 923-935. <https://doi.org/10.1109/TIM.2018.2853900>
- [31] Rousseeuw, P.J., Hubert, M. (2011). Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1): 73-79. <https://doi.org/10.1002/widm.2>
- [32] Qiu, G., Gu, Y., Chen, J. (2020). Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model. *Measurement*, 150: 107097. <https://doi.org/10.1016/j.measurement.2019.107097>

[33] Coble, J., Hines, J.W. (2009). Identifying optimal prognostic parameters from data: A genetic algorithms approach. Annual Conference of the PHM Society, 1(1).

[34] Hu, R., Huang, Q., Chang, S., Wang, H., He, J. (2019). The MBPEP: A deep ensemble pruning algorithm providing high quality uncertainty prediction. Applied Intelligence, 49: 2942-2955. <https://doi.org/10.1007/s10489-019-01421-8>

[35] Sateesh Babu, G., Zhao, P., Li, X.L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In Database Systems for Advanced Applications, Lecture Notes in Computer Science. [https://doi.org/10.1007/978-3-319-32025-0\\_14](https://doi.org/10.1007/978-3-319-32025-0_14)

[36] Zheng, S., Ristovski, K., Farahat, A., Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, pp. 88-95. <https://doi.org/10.1109/ICPHM.2017.7998311>

[37] Chen, J., Chen, D., Liu, G. (2021). Using temporal convolution network for remaining useful lifetime prediction. Engineering Reports, 3(3): 12305. <https://doi.org/10.1002/eng2.12305>

[38] Fan, Z., Li, W., Chang, K.C. (2023). A bidirectional long short-term memory autoencoder transformer for remaining useful life estimation. Mathematics, 11(24): 4972. <https://doi.org/10.3390/math11244972>

[39] Ayodeji, A., Wang, W., Su, J., Yuan, J., Liu, X. (2021). An empirical evaluation of attention-based multi-head models for improved turbofan engine remaining useful life prediction. arXiv preprint arXiv:2109.01761. <https://doi.org/10.48550/arXiv.2109.01761>

[40] Berghout, T., Benbouzid, M. (2024). Confidence intervals for uncertainty quantification in sensor data-driven prognosis. Engineering Proceedings, 82(1): 26. <https://doi.org/10.3390/ecsa-11-20501>

[41] Wen, B., Zhao, X., Tang, X., Xiao, M., Zhu, H., Li, J. (2025). A generalized diffusion model for remaining useful life prediction with uncertainty. Complex & Intelligent Systems, 11(2): 140. <https://doi.org/10.1007/s40747-024-01773-w>

## NOMENCLATURE

### Greek symbols

$\psi(t)$  Mother wavelet function

$\varphi(t)$	Scaling (father) wavelet function	
$\Theta$	Dimensionless temperature (if used)	Dimensionless
<b>Subscripts and Superscripts</b>		
<i>now</i>	Current cycle	
<i>max</i>	Maximum life cycle	
<i>pred</i>	Predicted value	
<i>true</i>	Ground truth value	
<i>0</i>	Initial value of the signal	
<i>f</i>	Final value of the signal	
RUL	Remaining Useful Life	Cycles
RMSE	Root Mean Squared Error	Same as predicted variable
MAE	Mean Absolute Error	Same as predicted variable
MSE	Mean Squared Error	Same as predicted variable
$R^2$	Coefficient of Determination	Dimensionless
CI	Confidence Interval	Same as the predicted variable
CI_W	Width of Confidence Interval	Same as the predicted variable
ObjS	Objective score combining accuracy and uncertainty	Dimensionless
$\alpha, \beta$	Weights in the objective function	Dimensionless
<i>x</i>	Original data value	Varies
<i>x'</i>	Normalized data value	Dimensionless
$\mu$	Mean of data	Same as data
$\sigma$	Standard deviation of data	Same as data
upL, lowL	Upper and lower limits (Z-score method)	Same as data
Q1, Q3	First and third quartiles (IQR method)	Same as data
<i>cA,k</i>	Approximation wavelet coefficients	Same as signal
<i>cD,j,k</i>	Detail wavelet coefficients	Same as signal
<i>F</i>	Feature vector	Varies
$\bar{F}$	Trend component of the feature vector	Varies
<i>n</i>	Number of data samples	Dimensionless
#pos, #neg	Number of positive/negative derivatives (monotonicity)	Count