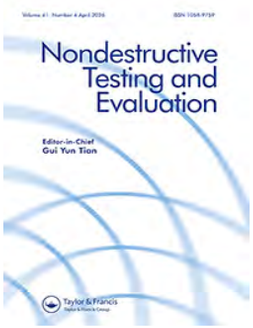

Nondestructive Testing and Evaluation

ISSN: 1058-9759 (Print) 1477-2671 (Online) Journal homepage: www.tandfonline.com/journals/gnte20

Degradation mode identification and remaining useful life prediction via an interpretable CNN-BiLSTM framework

Nima Rezazadeh , Giuseppe Lamanna , Francesco Caputo , Mario De Oliveira & Alessandro De Luca

To cite this article: Nima Rezazadeh , Giuseppe Lamanna , Francesco Caputo , Mario De Oliveira & Alessandro De Luca (19 Apr 2026): Degradation mode identification and remaining useful life prediction via an interpretable CNN-BiLSTM framework, Nondestructive Testing and Evaluation, DOI: 10.1080/10589759.2026.2660753

To link to this article: https://doi.org/10.1080/10589759.2026.2660753

Published online: 19 Apr 2026.

NONDESTRUCTIVE TESTING AND EVALUATION
https://doi.org/10.1080/10589759.2026.2660753

Degradation mode identification and remaining useful life prediction via an interpretable CNN-BiLSTM framework

Nima Rezazadeh[a,b], Giuseppe Lamanna[a], Francesco Caputo[a], Mario De Oliveira[b] and Alessandro De Luca[a]

[a]Department of Engineering, University of Campania "L. Vanvitelli", Aversa, Italy; [b]School of Architecture, Built Environment, Computing and Engineering, Birmingham City University, Birmingham, UK

ABSTRACT

Remaining useful life prediction is essential for condition-based maintenance in safety-critical industries, but existing deep learning approaches often exhibit degraded accuracy under variable operating conditions and provide limited interpretability. This paper presents an explainable CNN-BiLSTM framework incorporating regime-specific normalisation and degradation-aware feature engineering. The methodology employs clustering-based regime identification with condition-specific normalisation to mitigate covariate shift, alongside two complementary feature types: delta-from-baseline features capturing cumulative deviation from healthy states, and first-order temporal differences encoding instantaneous degradation rates. Validation on the NASA C-MAPSS benchmark demonstrates competitive performance under single operating conditions and substantial improvements over existing methods under variable conditions, particularly on the most challenging multi-regime, multi-fault subset. Post-hoc explainability analysis reveals that engineered features dominate model predictions, accounting for over 82% of total importance, whilst raw sensor values contribute minimally. The analysis further identifies three degradation mode clusters with characteristic feature utilisation patterns, indicating that the model learns to recognise multiple degradation signatures. Prediction accuracy improves markedly as engines approach failure, demonstrating highest precision when accurate forecasts are most consequential for maintenance decisions.

ARTICLE HISTORY
Received 23 February 2026
Accepted 11 April 2026

KEYWORDS
Prognostics and health management; condition-based maintenance; feature engineering; operating regime normalisation; turbofan engines; C-MAPSS

1. Introduction

In the era of Industry 4.0 and smart manufacturing, cyber-physical systems generate large volumes of sensor data, and the prediction of remaining useful life (RUL) has become a central topic in prognostics and health management. RUL is commonly defined as the estimated time from the present until an asset can no longer perform its intended function. Reliable RUL estimates support informed operational decisions by translating complex sensor measurements into an interpretable indicator of future availability [1,2].

CONTACT Nima Rezazadeh Nima.Rezazadeh@Unicampania.it

Accurate RUL prediction is particularly important in safety-critical industries, such as aerospace and nuclear power, where unexpected failures can cause severe safety consequences, major financial losses, and environmental harm [3]. Beyond accident prevention, RUL prediction underpins condition-based maintenance by enabling timely interventions, improved planning of maintenance windows, and more efficient inventory management. When implemented effectively, these benefits can reduce unnecessary preventive replacements whilst limiting the risk of disruptive breakdowns [4].

Despite its importance, RUL prediction remains one of the most challenging tasks in predictive maintenance because industrial data rarely satisfy ideal modelling assumptions [5]. Assets operate under varying loads, speeds, and environmental conditions, and they may exhibit interacting fault mechanisms that obscure degradation patterns [6]. These realities often create distribution differences between training datasets and operational deployments. Many existing approaches demonstrate robust performance under single operating conditions but exhibit substantial accuracy degradation when applied to multi-regime scenarios where covariate shift obscures degradation signatures. In addition, high-quality run-to-failure data are difficult to obtain because safety policies and operational constraints typically prevent equipment from being operated until failure [7,8].

As a result, much of the literature relies on benchmark datasets such as C-MAPSS (commercial modular aero-propulsion system simulation) released by NASA (national aeronautics and space administration), which emulate practical complications through high-dimensional multivariate time series with sensor noise and variable conditions [9,10]. Although deep learning (DL) has produced substantial gains, many proposed workflows still face obstacles to industrial uptake, including limited interpretability, insufficient uncertainty characterisation, and high computational cost. Maintenance engineers require transparency regarding which features and temporal patterns drive model predictions to establish operational trust; without such interpretability, even accurate models may face resistance in deployment [11]. These gaps motivate a careful review of existing approaches, their assumptions, and the remaining barriers to dependable RUL deployment in real settings.

The foregoing considerations reveal 3 persistent gaps in the current literature. First, whilst numerous architectures demonstrate impressive performance on single-condition benchmarks, their accuracy frequently degrades under variable operating regimes. Second, the relative contribution of raw sensor measurements versus engineered temporal features remains insufficiently characterised, limiting principled guidance for preprocessing pipeline design. Third, although interpretability is increasingly recognised as essential for industrial adoption, few studies provide systematic analysis of how DL models utilise different feature types or whether distinct degradation modes exist within homogeneous fleets.

The novelty of the present work lies not in a single architectural component but in the integration of operating-condition standardisation, degradation-aware feature construction, and post-hoc interpretability analysis within one prognostic framework. This combined strategy aims to improve robustness under variable conditions and to clarify how different feature groups contribute to RUL prediction. To address the above gaps, this paper presents an explainable CNN-BiLSTM framework with the following contributions:

(1) A regime-normalised preprocessing pipeline employing clustering-based regime identification and condition-specific normalisation to mitigate covariate shift in multi-regime scenarios.
(2) A degradation-aware feature engineering strategy comprising delta-from-baseline features capturing cumulative deviation from healthy states, and first-order temporal difference features encoding instantaneous degradation rates.
(3) Systematic experimental validation demonstrating competitive performance under single operating conditions and substantial improvements under variable conditions on the C-MAPSS benchmark.
(4) A comprehensive post-hoc explainability analysis quantifying feature type importance and identifying distinct degradation mode clusters with characteristic feature utilisation patterns, providing interpretability insights relevant to industrial deployment.

The remainder of this paper is organised as follows. Related works reviews related work on data-driven RUL prediction. Methodology and methods describes the proposed methodology. Case study presents the case study and experimental setup. Results and discussion reports and discusses the results, including the explainability analysis. Conclusion concludes with limitations and future directions.

2. Related works

Early data-driven approaches utilised shallow architectures such as support vector regression (SVR), multilayer perceptron, and relevance vector machine [12]. For example, Abdelghafar et al. [13] proposed an optimised SVR achieving fast convergence for RUL classification, whilst Ramasso [14] introduced RULCLIPPER using computational geometry and case-based reasoning to handle imprecise health indicators. These models depend heavily on manual feature engineering and domain expertise, and their shallow architectures have limited capacity to exploit such features effectively in high-dimensional, multi-regime data. Furthermore, shallow architectures struggle to capture the nonlinear hierarchies, and long-term temporal dependencies present in high-frequency sensor data [15].

Recurrent architectures, particularly long short-term memory (LSTM) and gated recurrent units, became prevalent due to their capacity for modelling long-term dependencies in sequential data [16]. Stacked LSTMs have demonstrated superior performance over traditional hidden Markov models by addressing the vanishing gradient problem [17]. Asif et al. [2] further enhanced LSTM performance by incorporating a piecewise linear degradation model to identify degradation onset points. Bidirectional variants (BiLSTM) extend this capability by capturing context from both past and future time steps within input sequences, enabling richer temporal representations [18]. However, standalone recurrent architectures often treat all time steps as equally important, potentially failing to distinguish between healthy background variation and critical degradation signatures [19].

Convolutional neural networks (CNNs) offer complementary strengths by extracting local spatial patterns from multivariate sensor sequences. Hybrid CNN-LSTM and CNN-BiLSTM architectures have emerged to leverage both spatial feature extraction and

temporal modelling capabilities [16,20]. Zhao et al. [20] demonstrated that such hybrid designs achieve improved accuracy over single-paradigm approaches. Li et al. [21] employed multi-scale convolutional blocks to capture patterns across different time granularities in aero-engine degradation data. These architectures provide a foundation for robust feature learning, though their performance under variable operating conditions and their interpretability remain areas requiring further investigation. In a related direction, You et al. [22] combined a quadratic neural network with BiLSTM for bearing fault diagnosis, demonstrating that hybrid architectures can achieve both computational efficiency and interpretability through visualisation of convolutional kernel responses and hidden states.

Recent state-of-the-art models have explored attention mechanisms and Transformer architectures. Guo et al. [23] proposed the multi-scale Hourglass-Transformer using pyramid self-attention to reduce computational complexity. Lin et al. [24] introduced channel attention and temporal attention-based temporal convolutional network employing dual-attention to identify informative signal combinations. Boujamza and Elhaq [25] integrated attention mechanisms with LSTM to weight temporal contributions adaptively. Whilst these approaches achieve competitive accuracy, they typically introduce additional architectural complexity and remain opaque regarding which specific features or degradation patterns drive their predictions. Complementary work has explored interpretability within Transformer-based fault diagnosis frameworks through attention weight visualisation and feature contribution analysis, though such analyses remain limited to fault classification rather than prognostic RUL estimation [26].

To address distribution shifts between operating conditions, transfer learning approaches have been developed. Yu et al. [1] utilised maximum mean discrepancy to align features from different operating regimes in a shared latent space. Siahpour et al. [27] introduced consistency-based regularisation for incomplete target datasets. Li et al. [28] proposed TFMPINN, a meta-learning framework integrating temporal state modelling with Fourier physics-constrained networks, demonstrating rapid adaptation capabilities for aero-engine RUL prediction under varying operating conditions. Wang et al. [29] proposed the degradation manifold dynamic consistency network, which defines a differentiable embedded degradation manifold to achieve dynamic consistency between state representation and intrinsic evolutionary trends, enabling probabilistic RUL prediction. These methods require substantial prior knowledge of operating conditions and often assume availability of complete run-to-failure cycles in target domains, assumptions frequently violated in practice. An alternative approach involves explicit regime identification and condition-specific normalisation during preprocessing, which can mitigate covariate shift without requiring domain adaptation during training.

The importance of interpretability for industrial adoption has received increasing attention [30]. Li et al. [31] proposed sensor-aware capsule network with uncertainty coverage scoring, providing some transparency regarding sensor contributions. In the broader context of interpretable machinery diagnostics, multimodal information fusion approaches have also been developed to enhance both diagnostic accuracy and model transparency [32], with attention-based mechanisms enabling interpretable weighting of heterogeneous signal modalities [33]. Permutation-based feature importance methods offer model-agnostic insights without architectural modifications [34]. However, systematic analysis of feature type contributions (raw measurements versus engineered

temporal features) and investigation of whether DL models implicitly learn to distinguish degradation modes remain unexplored. This gap limits understanding of model behaviour and hinders the development of principled preprocessing guidelines for prognostic applications.

The preceding review reveals persistent limitations in handling variable operating conditions, characterising feature contributions, and providing interpretability for industrial adoption. The methodology presented in the following section addresses these challenges.

3. Methodology and methods

This section presents the proposed framework for RUL prediction of complex engineering systems. The methodology comprises a multi-stage pipeline incorporating data preprocessing, degradation-aware feature engineering, operating regime normalisation, and a hybrid CNN-BiLSTM DL architecture. An extensive hyperparameter optimisation procedure is conducted to identify optimal model configurations. Figure 1 illustrates the overall framework architecture.

3.1. Data preprocessing module

The preprocessing pipeline transforms raw sensor measurements into regime-normalised representations through variance-based sensor filtering, k-means operating condition clustering, and condition-specific standardisation. A piecewise linear

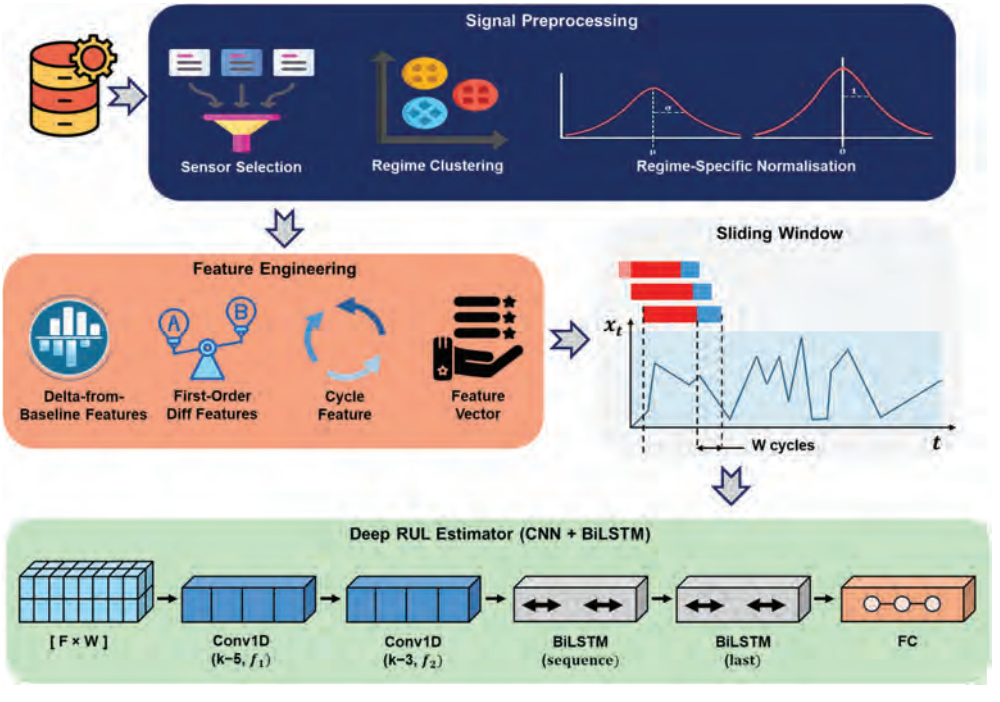

Figure 1. Schematic of the designed CNN-BiLSTM framework.

degradation model is applied to construct target RUL labels that reflect the assumption of an initial healthy operating phase.

3.1.1. Sensor selection

In complex monitored systems, not all sensors necessarily provide informative degradation signatures. Sensors exhibiting constant or near-constant readings throughout the system lifecycle contribute no discriminative information for RUL prediction and may introduce noise into the learning process. A variance-based filtering criterion is therefore applied to identify and retain only those sensors demonstrating meaningful temporal variation. Specifically, for each sensor j, the standard deviation σ_j is computed across all training observations. Sensors satisfying $\sigma_j > \tau$, where τ represents a predefined threshold, are retained for subsequent processing. This filtering step reduces input dimensionality whilst preserving degradation-relevant information.

3.1.2. Target variable construction

The true RUL at each operational cycle is defined as the number of cycles remaining until system failure. For training data, the RUL at cycle t for unit i is computed as:

$$RUL_{i,t} = T_i - t \tag{1}$$

where T_i denotes the total lifecycle duration of unit i. However, empirical evidence and domain knowledge suggest that degradation is typically not detectable during the initial healthy operating phase. Wang et al. [35] demonstrated that segmenting monitoring data into distinct degradation stages through trend detection techniques, combined with dynamic feedback mechanisms for adaptive parameter adjustment, can improve RUL prediction accuracy for rotating machinery. A piecewise linear degradation model is therefore adopted, wherein the RUL is capped at a maximum value $R_{\max}$:

$$RUL_{i,t}^{(\text{capped})} = \min\left(RUL_{i,t}, R_{\max}\right) \tag{2}$$

This formulation assumes that the health state remains constant during the initial operational phase, with measurable degradation commencing only after a threshold point. The specific value of $R_{\max}$ should be determined based on domain knowledge or empirical analysis of the degradation characteristics of the system under consideration.

3.1.3. Operating regime normalisation

Many complex systems operate under multiple distinct operating conditions characterised by varying environmental or operational parameters. Identical sensor readings may indicate different degradation states under different operating regimes, necessitating condition-aware normalisation to prevent cross-regime interference.

A k-means clustering algorithm is applied to the operational settings space to identify k distinct regimes. Let $\mathrm{o}_t \in {}^{\mathrm{m}}$ denote the m-dimensional operational settings vector at cycle t. The clustering results centroid vectors $\{c_1, c_2, \cdots, c_k\}$, and each observation is assigned to its nearest regime:

$$r_t = \arg\min_j \| \mathrm{o}_t - \mathrm{c}_j \|_2 \tag{3}$$

For each regime r, the mean $\mu^{(r)}$ and standard deviation $\sigma^{(r)}$ are computed from the training data. Feature normalisation is then applied on a per-regime basis:

$$x_t^{(\text{norm})} = \frac{\left(x_t - \mu_t^{(r)}\right)}{\sigma_t^{(r)}} \tag{4}$$

For systems operating under a single condition, global z-score normalisation is applied using statistics computed from the entire training set. This regime-aware approach ensures that the normalised features reflect degradation state rather than operating condition variations.

3.2. Degradation-aware feature engineering

To enhance the model's sensitivity to degradation patterns, two complementary feature engineering strategies are employed: baseline deviation features and temporal difference features.

3.2.1. Delta-from-baseline features

The initial operational cycles typically represent the healthy baseline state of each unit. Deviation from this baseline provides a direct measure of accumulated degradation. For each unit i, the baseline feature vector b_i is computed as the mean of the first N_b cycles:

$$b_i = \frac{1}{N_b} \sum_{t=1}^{N_b} x_{i,t} \tag{5}$$

The delta-from-baseline features at each subsequent cycle are then computed as:

$$\Delta x_{i,t} = x_{i,t} - b_i \tag{6}$$

These features capture the cumulative drift from the healthy operating state, enabling the model to learn degradation trajectories relative to each unit's initial condition.

3.2.2. First-order temporal difference features

Whilst delta-from-baseline features capture absolute deviation from baseline, first-order differences capture the instantaneous rate of change between consecutive cycles:

$$\nabla x_{i,t} = x_{i,t} - x_{i,t-1} \tag{7}$$

For the first cycle of each unit ($t = 1$), the difference is set to zero. These features enable the model to detect acceleration or deceleration in the degradation process, which may indicate transitions between degradation phases.

3.2.3. Delta-difference features

The first-order temporal difference operation is applied to the concatenated feature set comprising both raw sensor measurements and delta-from-baseline features. This concludes an additional feature type: first-order differences of the delta-from-baseline features, termed delta-difference features:

$$\nabla\Delta x_{i,t} = \Delta x_{i,t} - \Delta x_{i,t-1} \tag{8}$$

These features encode the rate of change of accumulated degradation, indicating whether deviation from the healthy baseline is accelerating or stabilising. Whereas delta-from-baseline features quantify total degradation magnitude and first-order differences capture instantaneous sensor dynamics, delta-difference features specifically represent degradation trajectory curvature.

3.2.4. Feature vector composition

The final feature vector at each cycle comprises 4 complementary components concatenated in sequence: (i) original sensor readings x, providing absolute measurements; (ii) delta-from-baseline features Δx, capturing cumulative deviation from healthy conditions; (iii) first-order difference features x, encoding instantaneous sensor changes; and (iv) delta-difference features Δx, representing the rate of degradation accumulation. For a system with F retained sensors after variance-based filtering, the resulting feature dimensionality is $4F$ (excluding operational settings and cycle features where applicable). This composite representation provides multiple complementary perspectives on the degradation process.

In summary, the feature engineering pipeline transforms each observation from a single vector of sensor readings into a composite representation encoding 4 complementary views of the degradation process: absolute measurements, cumulative deviation from the healthy baseline, instantaneous rates of change, and the acceleration of degradation accumulation. This enriched input is designed to reduce the representational burden on the subsequent network, allowing the CNN layers to extract local patterns across these feature types and the BiLSTM layers to model their temporal evolution.

3.3. Network architecture

The proposed architecture employs a hybrid CNN-BiLSTM design that leverages the complementary strengths of convolutional and recurrent neural networks; these steps are elaborated in the following sections.

3.3.1. Convolutional feature extraction module

The convolutional module comprises two sequential 1D convolutional layers designed to extract local patterns and spatial features from the multivariate sensor sequences. The first convolutional layer applies f_1 filters with kernel size k_1, followed by batch normalisation and ReLU activation. The second convolutional layer applies f_2 filters with kernel size k_2, similarly followed by batch normalisation and ReLU activation. Same-padding is employed to preserve the temporal dimension. A dropout layer with probability p is applied after the convolutional block to mitigate overfitting.

3.3.2. Bidirectional LSTM temporal modelling module

The temporal modelling module consists of two stacked Bidirectional LSTM layers. Unlike standard LSTMs, BiLSTMs process sequences in both forward and backward directions, enabling the network to capture context from both past and future time steps within each input window. The first BiLSTM layer contains u_1 units and outputs the full

sequence, allowing the subsequent layer to process the complete temporal representation. Layer normalisation is applied after this layer to stabilise training dynamics. The second BiLSTM layer contains u_2 units and outputs only the final hidden state, aggregating the sequential information into a fixed-dimensional representation. Layer normalisation is again applied, followed by dropout regularisation.

The choice of layer normalisation over batch normalisation in the recurrent pathway is motivated by its independence from mini-batch statistics, which is particularly advantageous for sequence modelling where batch composition may vary significantly.

3.3.3. Regression head

The regression head comprises a fully connected layer with h hidden units and ReLU activation, followed by dropout, and a final fully connected layer producing the scalar RUL prediction. The network is trained to minimise the mean squared error between predicted and true RUL values.

3.4. Training strategy

The training procedure combines sliding window sequence generation with engine-wise validation splitting and Adam optimisation, incorporating learning rate scheduling and regularisation techniques to ensure stable convergence and prevent overfitting.

3.4.1. Sliding window sequence generation

Input sequences are generated using a sliding window approach with window length W and stride 1. For each unit trajectory, windows are extracted starting from cycle W onwards, with each window's target RUL corresponding to the RUL at the final cycle of that window. This approach generates multiple training samples per unit whilst preserving temporal ordering within each sample.

For units with fewer than W cycles in the test set, zero-padding is applied at the beginning of the sequence to maintain consistent input dimensions.

3.4.2. Validation strategy

To prevent data leakage and ensure unbiased model selection, validation is performed at the unit level rather than the sample level. Specifically, a fraction v of training units are randomly held out as the validation set. This approach ensures that all sequences from a given unit appear exclusively in either the training or validation set, preventing the model from exploiting inter-sequence correlations within individual unit trajectories.

3.4.3. Optimisation and regularisation

The network is trained using the Adam optimiser with an initial learning rate η_0. A piecewise learning rate schedule reduces the learning rate by a decay factor γ every n epochs. Training proceeds for a maximum number of epochs with early stopping based on validation loss. The model achieving the lowest validation loss is retained for evaluation.

Regularisation is applied through multiple mechanisms: L_2 weight decay, dropout after convolutional and recurrent layers, and gradient clipping to prevent exploding gradients.

3.5. Hyperparameter optimisation

A systematic grid search is conducted over key architectural hyperparameters: sequence length (W), dropout probability (p), number of LSTM units (u_1), and number of convolutional filters (f_1). The search space is defined based on preliminary experiments and computational constraints.

Each configuration is evaluated independently, with model selection based on validation loss and final evaluation performed on the held-out test set. To ensure reproducibility, a fixed random seed is employed throughout the experimental procedure.

3.6. Evaluation metrics

Model performance is assessed using two complementary metrics:

Root Mean Squared Error (RMSE) provides a measure of average prediction accuracy:

$$\text{RMSE} = \sqrt{\left(\frac{1}{N}\right)\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{9}$$

Asymmetric Scoring Function penalises late predictions more heavily than early predictions, reflecting the greater operational risk of underestimating remaining life in safety-critical applications:

$$\text{Score} = \sum_{i=1}^{N} s_i \tag{10}$$

where the individual score s_i for each prediction is defined as:

$$\begin{aligned} s_i &= \exp(-d_i/\alpha) - 1, \text{ If } d_i < 0 \text{ (early prediction)} \\ s_i &= \exp(d_i/\beta) - 1, \text{ If } d_i \geq 0 \text{ (late prediction)} \end{aligned} \tag{11}$$

where $d_i = \hat{y}_i - y_i$ represents the prediction error, and α and β are asymmetry parameters reflecting the domain-specific cost structure. The condition $\alpha > \beta$ ensures that late predictions (which may lead to unexpected failures) incur greater penalties than early predictions.

4. Case study

The empirical validation of the proposed methodology is centred upon the C-MAPSS dataset, a high-fidelity software tool developed by NASA to replicate the degradation dynamics of a 90,000-lb thrust class turbofan engine. This system operates as a complex, non-linear environment where performance deterioration is influenced by an intricate interplay of atmospheric conditions, flight profiles, and internal component wear. A schematic representation of the engine architecture, detailing the connectivity between the fan, high-pressure compressor (HPC), low-pressure compressor (LPC), the combustor, high-pressure turbine (HPT), and low-pressure turbine (LPT), is presented in Figure 2.

This dataset is partitioned into four subsets (FD001 to FD004) to evaluate prognostic performance across varying levels of environmental and mechanical complexity. Each

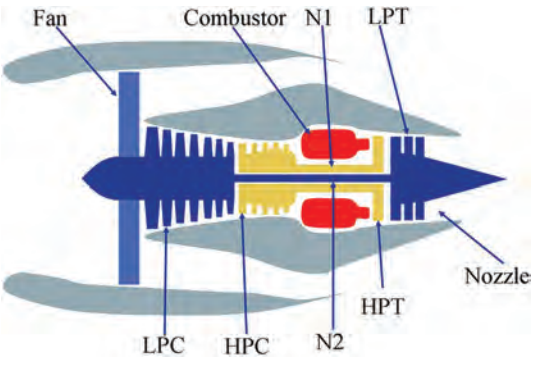

Figure 2. Simplified schematic of the used engine in C-MAPSS dataset [36].

Table 1. Technical specifications of the C-MAPSS sub-datasets.

Sub-dataset	Operational Conditions	Fault Modes	Training Trajectories	Testing Trajectories
FD001	1 (Sea Level)	1 (HPC Degradation)	100	100
FD002	6 (Variable)	1 (HPC Degradation)	260	259
FD003	1 (Sea Level)	2 (HPC, Fan)	100	100
FD004	6 (Variable)	2 (HPC, Fan)	249	248

subset contains multivariate time-series data from multiple engine units, which are assumed to be in a healthy state at the commencement of their respective records. As operational cycles accumulate, these units exhibit degradation until a system failure is declared. The specific technical characteristics of these subsets, including the number of trajectories and the simulated mechanical fault modes, are summarised in Table 1.

In the simulation environment, 21 sensors are deployed across critical engine locations to record performance signatures. These instruments monitor a variety of physical parameters, primarily focusing on gas path temperatures, pressures, and rotational speeds. Each observation in the raw dataset constitutes a snapshot of a single operational cycle, comprising 26 columns: the engine unit identifier, the current cycle number, 3 operational settings (Altitude, Mach number, and Throttle Resolver Angle), and the 21 sensor measurements.

For subsets FD002 and FD004, the engines are subjected to 6 distinct operating regimes based on various combinations of Altitude (0–42000 ft), Mach number (0–0.84), and Sea-level temperature. This diversity in flight conditions results in a significant covariate shift, where the distribution of sensor measurements varies depending on the current regime.

5. Results and discussion

To evaluate the proposed RUL estimation framework, the training set of each C-MAPSS subset (FD001–FD004) was partitioned into training and validation sets at an 85:15 ratio using an engine-wise split, ensuring that entire engine trajectories were assigned exclusively to either training or validation to prevent data leakage. The official C-MAPSS test set, provided separately with corresponding end-of-life RUL values, was strictly held out throughout the entire hyperparameter optimisation process. All model selection

decisions, including grid search ranking and early stopping, were based exclusively on the internal validation loss. The test set was used only once for final performance evaluation after the optimal configuration had been identified. All experiments were executed in MATLAB® 2025b.

5.1. RUL estimation performance

For each dataset, a grid search over 400 hyperparameter combinations was performed. For each combination, the model was trained and evaluated using 10 random seeds to account for training variability. The best model per dataset was selected using the validation RMSE criterion (computed on the validation windows). After selecting the best hyperparameter configuration, final performance was reported on the test set by computing the mean±std across the 10 seeds for RMSE and the C-MAPSS score. The sensor selection threshold was selected as 0.001. Application of this threshold retained 15 sensors for FD001, 21 for FD002, 16 for FD003, and 21 for FD004, reflecting the greater sensor variability introduced by multiple operating regimes in FD002 and FD004.

The grid search varied 4 hyperparameters, while all remaining parameters were held constant. The candidate values for the searched hyperparameters, and the constants used throughout all experiments, are listed in Table 2.

After executing the full framework for each dataset, the most optimum sets of hyperparameters for them were found as specified in Table 3.

The best performing models, selected using validation RMSE, were then evaluated on their corresponding test sets. Performance was reported using RMSE and the asymmetric scoring function. Figure 3 summarises the test set RMSE values across datasets for the 10

Table 2. Hyperparameter values used for RUL estimation.

Hyperparameter	Value	Hyperparameter	Value
Sequence length	20, 30, 40, 50	Regime normalisation	Enabled
Dropout probability	0.15, 0.25, 0.35, 0.45	FC hidden units	128
First BiLSTM units (u_1)	96, 128, 160, 192, 224	Max epochs	80
CNN filters (conv1)	32, 48, 64, 80, 96	Mini-batch size	128
CNN filters (conv2)	2 × conv1filters	Initial learning rate	1×10^{-3}
Training/validation split	85/15 (engine-wise)	L2 regularisation	2×10^{-4}
Baseline cycles	5	Gradient clip threshold	1
RUL cap	125	Seeds per trial	10
Number of operating regimes	6	Sensor variance threshold	1×10^{-3}
Learning rate drop factor	0.5	Learning rate drop period	8 epochs
Early stopping patience	10	Second BiLSTM units	max(32, $\lfloor u_1/2 \rfloor$)
Conv1 kernel size	5	Conv2 kernel size	3

Table 3. Optimal hyperparameter configurations selected for each C-MAPSS dataset (FD001–FD004).

	Value			
Hyperparameter	FD001	FD002	FD003	FD004
Sequence length	30	50	20	50
Dropout probability	0.25	0.35	0.25	0.35
BiLSTM units	160	192	96	224
CNN filters (conv1)	48	48	32	32
CNN filters (conv2)	96	96	64	64

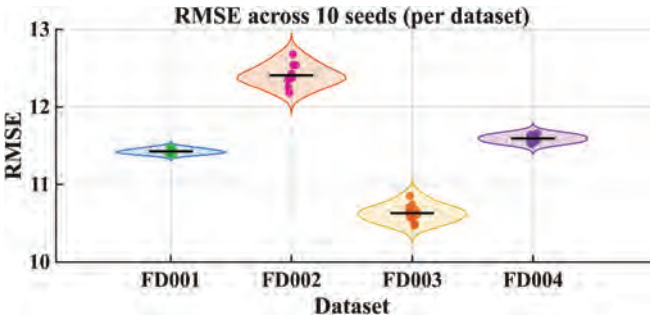

Figure 3. Test set RMSE across FD001 to FD004 for 10 random seeds.

random seeds. The black solid line indicates the mean RMSE across the 10 seeds. For the score metric, the asymmetry parameters were set to $\alpha = 13$ and $\beta = 10$. Using the baseline seed, the achieved scores for FD001, FD002, FD003, and FD004 were 201, 675, 195, and 645, respectively.

Figure 3 shows that test set RMSE is consistent across the 10 seeds, with small standard deviations for FD001, FD003, and FD004. FD002 exhibits the largest variability and the highest mean RMSE, indicating greater sensitivity to initialisation and a more challenging operating regime. Although FD001 and FD004 have similar mean RMSE values, their base seed scores differ, reinforcing that the asymmetric scoring function captures a different risk profile than RMSE alone.

To visualise the framework's RUL prediction performance, plots in Figure 4-a to 4-d present the true and predicted RUL trajectories for units 31, 62, 99, and 22 from the FD001, FD002, FD003, and FD004 test sets, respectively.

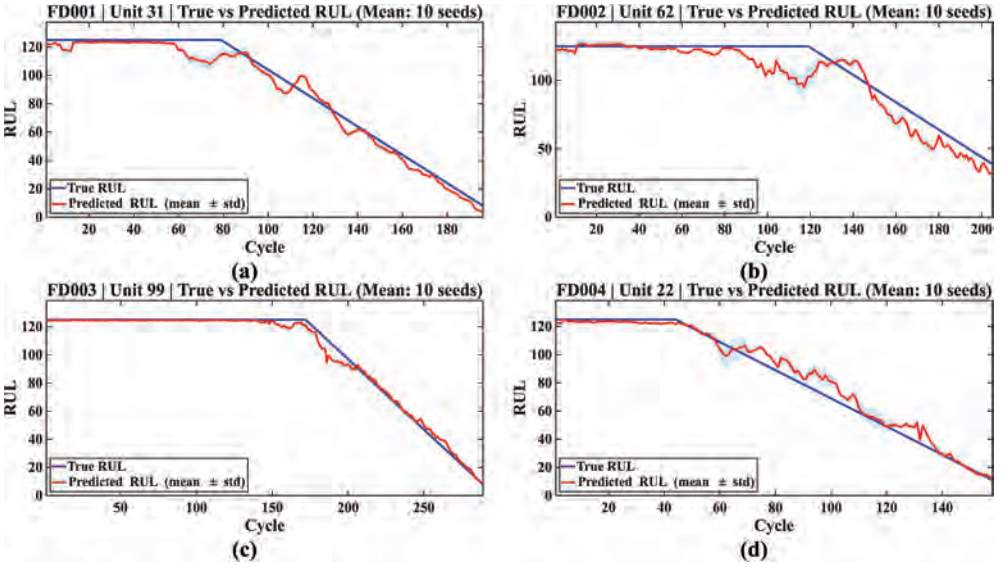

Figure 4. True and predicted RUL trajectories for representative test units: (a) unit 31 (FD001), (b) unit 62 (FD002), (c) unit 99 (FD003), and (d) unit 22 (FD004).

From Figure 4, it can be observed that the proposed CNN-BiLSTM framework effectively captures the degradation trajectory across all 4 sub-datasets. In the initial operational phase, where the RUL is capped at 125 cycles, the model maintains predictions close to the cap value, reflecting the piecewise linear assumption that degradation is not detectable during healthy operation. As the engines progress towards failure, the predicted RUL closely tracks the true RUL with minimal deviation. The shaded uncertainty bands, representing the mean plus or minus one standard deviation across the 10 random seeds, demonstrate consistent predictions with narrow confidence intervals, particularly in the later stages of degradation where accurate predictions are most critical for maintenance decision-making. Notably, all 4 representative units exhibit smooth prediction trajectories without significant oscillations, indicating that the model has learned robust temporal patterns rather than overfitting to noise. The prediction accuracy visibly improves as engines approach end-of-life, which aligns with the expectation that degradation signatures become more pronounced and distinguishable in the final operational phase.

5.2. Post-hoc explainability and degradation mode identification

To address the interpretability limitations inherent in DL models, a comprehensive post-hoc explainability analysis was conducted on the FD001 test set. This analysis aimed to identify which features and sensors contributed most significantly to the model's predictions and to investigate whether distinct degradation patterns exist across the engine fleet. Permutation-based feature importance was employed as the primary explainability method, wherein each feature was systematically permuted across time steps and the resulting change in prediction was measured. This approach provides model-agnostic insights into feature contributions without requiring architectural modifications or access to internal gradients.

The analysis was performed across all 100 test engines in FD001, with 30 permutations per feature to ensure statistical stability. For each engine, feature importance scores were computed and normalised to sum to 100%, enabling direct comparison across engines with different prediction magnitudes. Additionally, time-step importance was computed by permuting all features at each temporal position within the input window, revealing which portions of the historical sequence most influenced the predictions. The global feature importance rankings averaged across all test engines are presented in Figure 5.

As shown in Figure 5, engineered features dominate the model's decision-making process. Specifically, the delta-difference features (denoted delta_diff in figures, corresponding to Δ x as defined in Equation 8), first-order difference features (denoted diff in figures, corresponding to x as defined in Equation 7), and delta-from-baseline features (denoted delta in figures, corresponding to Δ x as defined in Equation 6) are distinguished from raw sensor values throughout the following analysis. Among these, the delta-difference features exhibit the highest importance. The top-ranked feature, Sens7_delta_diff, achieved a mean importance of 2.29% with a standard deviation of 1.05%, followed by Sens2_delta_diff at 2.10% and Sens3_delta_diff at 2.07%. Notably, the top 20 features are composed of delta_diff and diff variants, whilst raw sensor values appear with lower rankings.

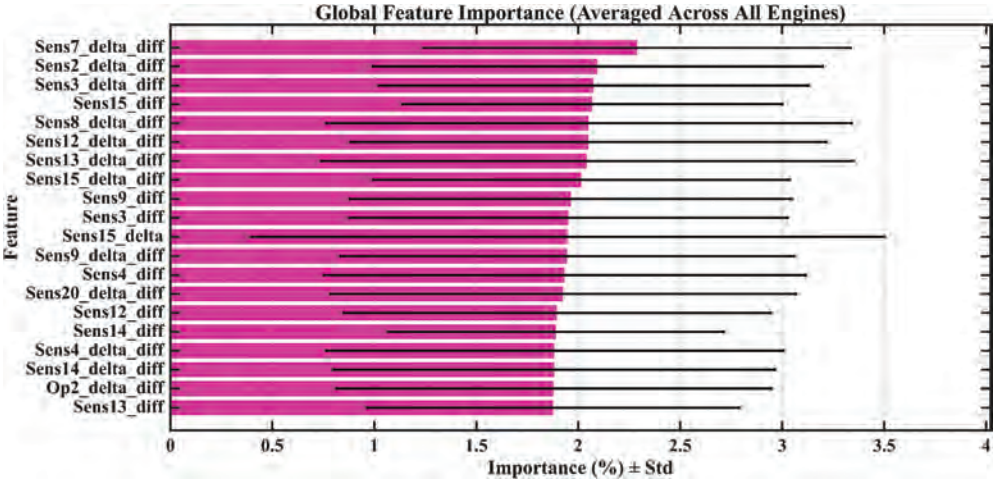

Figure 5. Global feature importance averaged across all 100 test engines in FD001. Error bars indicate ±1 standard deviation across engines, reflecting variability in feature utilisation across the fleet.

A quantitative analysis of feature type contributions revealed that delta_diff features collectively account for 31.5% of total model importance, followed by first-order difference features (diff) at 30%, delta-from-baseline features at 20.6%, and raw sensor values at only 17.9%. This distribution strongly validates the degradation-aware feature engineering approach described in the previous sections, demonstrating that capturing temporal dynamics and deviation from healthy baseline conditions is more informative for RUL prediction than raw sensor measurements alone.

When aggregating importance across all feature variants for each sensor, the analysis identified Sensor 15 (bypass ratio) as the most influential with a total importance of 7.13%, followed by Sensor 7 (high-pressure compressor outlet temperature) at 6.69% and Sensor 3 (high-pressure compressor outlet temperature at an alternative location) at 6.67%. These findings align with domain knowledge of turbofan degradation physics, where thermal stress indicators and combustion efficiency parameters serve as primary signatures of component wear.

To investigate whether the model relies on uniform feature utilisation across all engines or adapts its attention based on degradation characteristics, k-means clustering was applied to the 100-dimensional feature importance vectors. Three distinct clusters emerged, each exhibiting characteristic feature importance profiles. The comparative feature importance patterns across these clusters are illustrated in Figure 6.

Figure 6 reveals substantial differences in feature utilisation across the three clusters. Cluster 1 relies on temperature-related delta_diff features (Sens7, Sens2, Sens8), whilst Cluster 2 shows elevated importance for Sensor 6 variants. Most notably, Cluster 3 exhibits a distinctive pattern wherein raw sensor values (Sens12, Sens7) assume greater importance than engineered features. These divergent profiles suggest that the model has learned to recognise multiple degradation signatures rather than applying a uniform prediction strategy. The detailed characteristics of each cluster are summarised in Table 4.

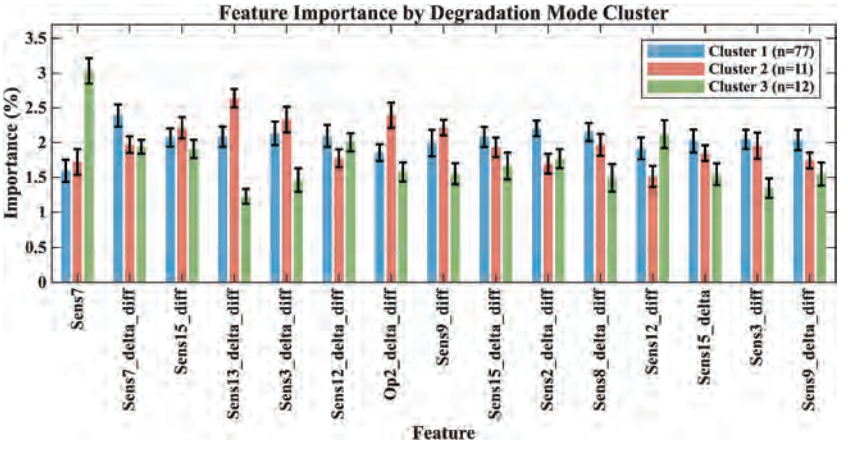

Figure 6. Feature importance profiles for the 3 identified degradation mode clusters, showing the top 15 features ranked by average importance; error bars indicate ±1 standard deviation across 10 independent training runs.

Table 4. Characteristics of the three degradation mode clusters identified in the FD001 test set.

Cluster	Engines	Percentage	Mean RUL (cycles)	Mean Absolute Error (cycles)	Dominant Features
1	77	77%	79.1 ± 35.9	8.8	Sens7_delta_diff, Sens2_delta_diff, Sens8_delta_diff
2	11	11%	108.7 ± 17.3	15.68	Sens6_ diff, Sens6_delta_diff, Sens13_delta_diff
3	12	12%	13.3 ± 6.6	2.28	Sens12, Sens7, Sens11_delta

As presented in Table 4, Cluster 1 comprises 77 engines (77% of the fleet) and represents the predominant degradation mode. Engines in this cluster span a wide range of true RUL values (10 to 125 cycles) and exhibit a mean absolute prediction error of 8.80 cycles, closely matching the overall fleet performance.

Cluster 2 contains 11 engines (11%) with notably higher mean RUL values (108.7 ± 17.3 cycles), indicating engines in earlier degradation stages. This cluster exhibits distinctly different feature utilisation, with Sensor 6 variants dominating rather than Sensors 7 and 2. The mean absolute error for this cluster is 15.68 cycles, the highest among the three groups, suggesting that predicting RUL for engines far from failure presents greater challenges.

Cluster 3 encompasses 12 engines (12%) characterised by low RUL values (13.3 ± 6.6 cycles), representing engines approaching imminent failure. Notably, this cluster exhibits the lowest prediction error at 2.28 cycles and demonstrates a distinctive shift towards raw sensor values rather than engineered features. This finding suggests that as engines approach failure, absolute sensor magnitudes become more diagnostic than relative changes, likely because degradation has progressed sufficiently to produce clearly anomalous readings.

It should be noted that FD001 contains only a single fault mode (HPC degradation), and therefore the 3 clusters represent distinct degradation stages within the same failure mechanism rather than different physical fault types. The correspondence between cluster assignment and mean RUL values (108.7, 79.1, and 13.3 cycles for Clusters 2, 1, and 3, respectively) supports this interpretation. Extension of this analysis to multi-fault subsets, such as FD003, where HPC and fan degradation coexist, could potentially reveal mechanism-specific clusters and is identified as a direction for future investigation.

A principal component analysis (PCA) was conducted to visualise the separation between clusters in reduced dimensional space, as presented in Figure 7.

The left panel of Figure 7 displays engines coloured by their cluster assignment, revealing clear boundaries between the three degradation modes in the principal component space. The right panel colours engines by their true RUL value, demonstrating a gradient pattern wherein engines with similar remaining life tend to occupy proximate regions in the feature importance space. This correspondence further supports the interpretation that the model adapts its attention mechanism based on the degradation stage of each engine, with feature utilisation patterns systematically varying as a function of remaining life.

The relationship between degradation mode and prediction accuracy is examined in Figure 8.

As evident from the box plots in Figure 8, Cluster 3 (near-failure engines) exhibits lower prediction errors with a tighter distribution compared to the other clusters. Cluster 2 (early-life engines) shows the highest errors with the greatest variability, whilst Cluster 1 occupies an intermediate position. This pattern has important practical implications: the model achieves highest accuracy precisely when accurate predictions are most critical for maintenance planning, i.e., in the period immediately preceding failure. The mean

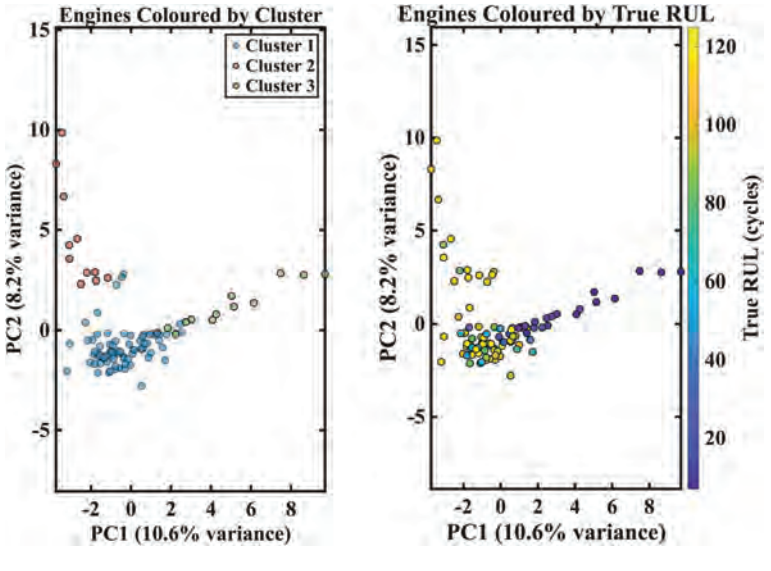

Figure 7. Principal component analysis visualisation of engine feature importance patterns. Left panel: engines coloured by cluster assignment. Right panel: engines coloured by true RUL value.

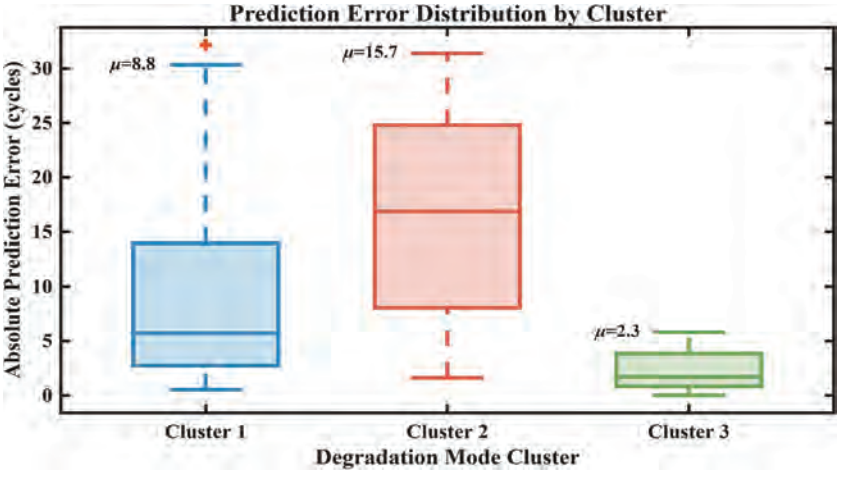

Figure 8. Distribution of absolute prediction errors across the 3 degradation mode clusters.

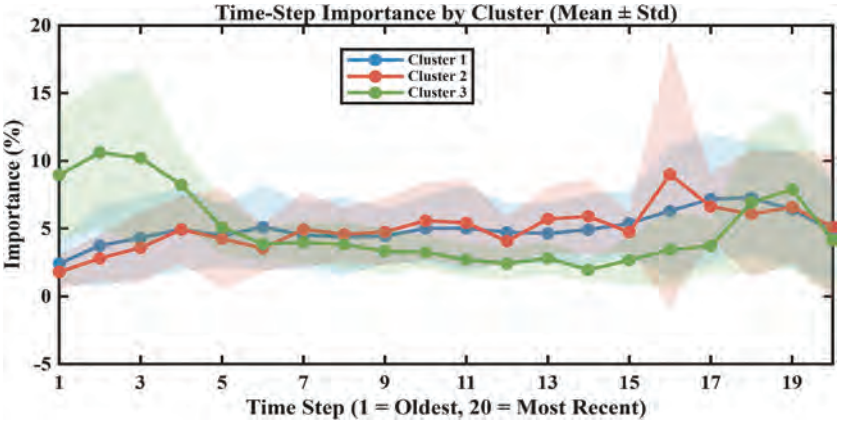

Figure 9. Time-step importance profiles for each degradation mode cluster, showing how temporal attention varies across the 20-cycle input window.

absolute errors annotated above each box plot quantify this relationship, with Cluster 3 achieving 2.28 cycles compared to 15.68 cycles for Cluster 2.

The temporal attention patterns across the three clusters are illustrated in Figure 9.

Figure 9 demonstrates that all three clusters exhibit distributed importance across the input window rather than concentration on the most recent time steps. The final 5 time steps account for approximately 11% of total importance on average, indicating that the model does not disproportionately weight recent observations. This distributed temporal attention suggests that the BiLSTM architecture effectively leverages the full historical context, capturing both gradual degradation trends and short-term fluctuations. The similarity in temporal profiles across clusters indicates that whilst feature importance varies between degradation modes, the temporal weighting strategy remains consistent.

The explainability analysis results in multiple key insights regarding the model's behaviour. First, the dominance of engineered features (82.1% combined importance)

over raw sensor values (17.9%) empirically validates the feature engineering methodology and suggests that similar preprocessing strategies may benefit other prognostic applications. Second, the identification of three distinct degradation modes indicates that the model has learned to recognise multiple failure signatures rather than applying a uniform prediction strategy, enhancing its robustness across heterogeneous operating histories. Third, the inverse relationship between remaining life and prediction accuracy, wherein near-failure engines achieve the lowest errors, represents a desirable characteristic for maintenance applications where precision near the failure threshold directly impacts operational decisions.

5.3. Ablation study

To quantify the contribution of the proposed feature engineering components, an ablation study was conducted by systematically removing each feature type from the full pipeline. Three ablation configurations were evaluated: (1) No Diff, wherein first-order temporal difference features were excluded whilst retaining delta-from-baseline features; (2) No Delta, wherein delta-from-baseline features were removed whilst retaining difference features; and (3) Raw Only, wherein all engineered features were removed, and the model received only the original sensor measurements. The first two configurations isolate the marginal contribution of each feature type, whilst the third establishes the baseline performance achievable without degradation-aware preprocessing. All configurations utilised the optimal hyperparameters identified for each dataset and were evaluated using the mean RMSE across 10 random seeds. The results are presented in Figure 10.

As shown in Figure 10, removing any feature engineering component results in performance degradation across all four datasets, with standard deviations across 10 random seeds remaining below 0.53 RMSE in all configurations, confirming the statistical robustness of these findings. The removal of delta-from-baseline features (No Delta) produces the most substantial RMSE increase, with FD001 exhibiting a 62.1% degradation (from 11.40 ± 0.36 to 18.48 ± 0.53) and FD003 showing a 64.4% increase (from 10.71

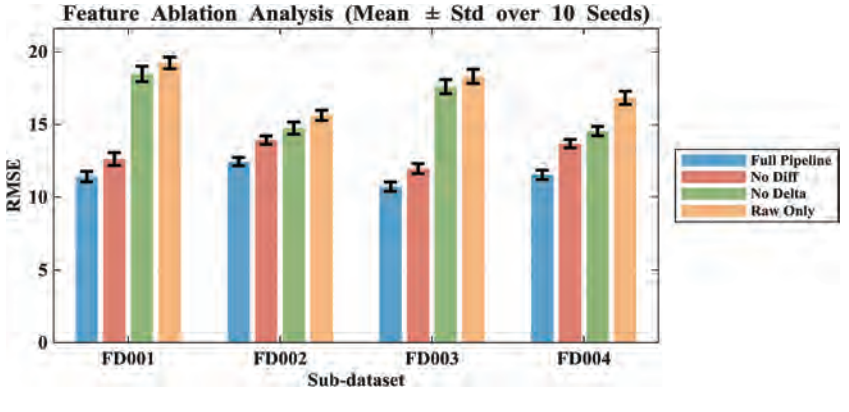

Figure 10. Ablation study results showing RMSE degradation when feature engineering components are removed from the full pipeline.

± 0.32 to 17.61 ± 0.48). The removal of difference features (No Diff) results in moderate performance reduction, with RMSE increases ranging from 10.7% on FD001 (from 11.40 ± 0.36 to 12.62 ± 0.44) to 18.6% on FD004 (from 11.53 ± 0.32 to 13.68 ± 0.29). The Raw Only configuration, retaining no engineered features, produces the largest degradation across all subsets, with RMSE increasing by 68.8% on FD001 (from 11.40 ± 0.36 to 19.24 ± 0.41) and 46.1% on FD004 (from 11.53 ± 0.32 to 16.84 ± 0.45). Notably, the non-overlapping standard deviation intervals between the Full Pipeline and each ablated configuration across all datasets indicate that the observed performance differences are not attributable to weight initialisation variability.

These findings are consistent with the post-hoc explainability analysis presented in the previous section, which identified delta-diff and diff features as accounting for 61.5% of total model importance. The ablation results empirically confirm that delta-from-baseline features capture the most critical degradation information, aligning with the observation that deviation from healthy baseline conditions serves as the primary indicator of accumulated wear. The greater sensitivity to delta-from-baseline feature removal in single operating condition datasets (FD001, FD003) compared to variable condition datasets (FD002, FD004) may reflect the more consistent baseline reference available under uniform operating regimes.

5.4. Comparison study

The effectiveness of the designed RUL estimation framework was compared with related state-of-the-art DL methods reported in the literature. Given the extensive body of work utilising the C-MAPSS benchmark dataset, 10 methods were selected to represent a progressive architectural trajectory relevant to the proposed framework. The selection begins with standalone recurrent networks (LSTM, BiLSTM) as foundational temporal baselines, followed by convolutional architectures (DCNN) to represent spatial feature extraction. Hybrid models (CNN-LSTM, CNN-BiLSTM) serve as direct architectural comparators, whilst attention-enhanced variants (LSTM-Attention, CNN-LSTM-Attention, CNN-BiLSTM-Attention) represent the current trend of incorporating adaptive temporal weighting. Finally, PPL-LSTM and TFIM were included because they emphasise preprocessing and feature-level innovation rather than architectural complexity, making them the most relevant comparators for the feature engineering contribution of this work.

Table 5 presents the comparative results for both RMSE and Score metrics across all four C-MAPSS sub-datasets.

It should be noted that PPL-LSTM is a preprocessed piecewise-linear labelling method combined with LSTM and TFIM stands for the time–feature interaction model. Lower values indicate superior performance for both metrics. The comparative RMSE performance across all methods is visualised in Figure 11 for enhanced clarity.

As illustrated in Table 5 and Figure 11, the proposed framework demonstrates competitive performance on the single operating condition sub-datasets (FD001 and FD003) whilst achieving superior results on the multi-condition sub-datasets (FD002 and FD004). On FD001, the proposed method achieves an RMSE of 11.40, which is competitive with established methods such as TFIM (10.74) and CNN-LSTM (11.56). On

Table 5. Performance comparison between the proposed method and state-of-the-art approaches on the C-MAPSS dataset.

	RMSE				Score				
Method	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004	Reference
LSTM	16.14	24.49	16.18	28.17	338	4450	852	5550	Zheng et al. [37]
BiLSTM	13.65	23.18	13.74	24.86	295	4130	317	5430	Wang et al. [38]
DCNN	12.61	22.36	12.64	23.31	274	10412	284	12466	Li et al. [21]
CNN-LSTM	11.56	17.67	12.98	20.19	247	1743	808	3051	Shi et al. [16]
CNN-BiLSTM	12.58	19.34	12.18	20.03	231	2650	257	3400	Zhao et al. [20]
LSTM-Attention	13.95	17.65	12.72	20.21	320	2102	223	3100	Boujamza and Elhaq [25]
CNN-LSTM-Attention	15.98	14.45	13.91	16.64	–	–	–	–	Deng and Zhou [39]
CNN-BiLSTM-Attention	12.49	20.32	12.34	26.54	267	1576	226	3905	Yu et al. [40]
PPL-LSTM	**7.78**	17.64	**8.3**	17.63	**100**	1443	**104**	2395	Asif et al. [2]
TFIM	10.74	13.32	11.05	14.2	164	729	281	879	Zhevnenko et al. [41]
Proposed Method	11.40	**12.44**	10.71	**11.53**	201	**675**	195	**645**	–

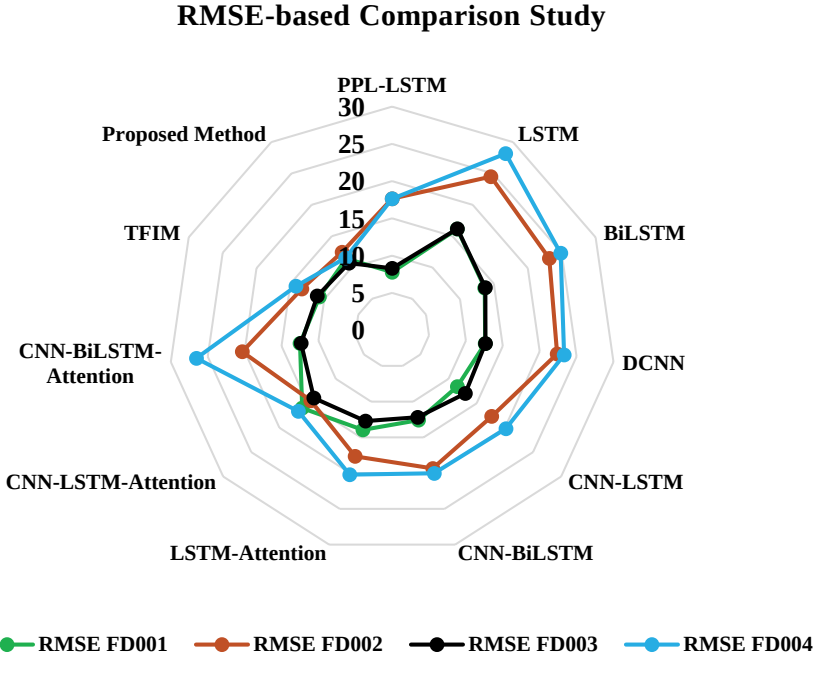

Figure 11. RMSE comparison across state-of-the-art methods and the proposed framework on the C-MAPSS dataset.

FD003, the proposed framework attains the second-best RMSE of 10.71, surpassed only by PPL-LSTM (8.30).

The distinguishing characteristic of the proposed framework emerges when examining performance under variable operating conditions. On FD002, which encompasses 6 distinct operational regimes with a single fault mode, the proposed method achieves an RMSE of 12.44, representing a 6.6% improvement over TFIM (13.32) and a 29.5% improvement over PPL-LSTM (17.64). The performance differential becomes even more pronounced on FD004, which represents the most challenging scenario combining six operating conditions with two fault modes. The proposed framework achieves an

Table 6. Trainable parameters for the optimal model configuration of each C-MAPSS dataset.

Dataset	BiLSTM Units	CNN Filters (f1, f2)	Trainable Parameters
FD001	160	48, 96	639,793
FD002	192	48, 96	877,553
FD003	96	32, 64	248,545
FD004	224	32, 64	1,073,249

RMSE of 11.53, demonstrating an 18.8% improvement over TFIM (14.20) and a 34.6% improvement over PPL-LSTM (17.63). Similarly, on the Score metric, the proposed method achieves 675 on FD002 and 645 on FD004, representing improvements of 7.4% and 26.6% over TFIM, respectively.

The superior performance under variable operating conditions can be attributed to a couple of synergistic design elements. The operating regime normalisation strategy employs k-means clustering to identify distinct operational regimes and applies condition-specific z-score normalisation, effectively mitigating covariate shift induced by varying flight profiles. The degradation-aware feature engineering, comprising delta-from-baseline and first-order temporal difference features, provides stable degradation indicators that remain interpretable across different flight regimes. As demonstrated in the explainability analysis, these engineered features account for over 82% of the model's predictive importance. Additionally, the CNN-BiLSTM architecture provides complementary spatial and temporal feature extraction capabilities well-suited to multi-condition scenarios, whilst the systematic hyperparameter optimisation ensures that architectural parameters are tailored to the specific complexity of each sub-dataset.

Beyond numerical performance, the proposed framework offers interpretability advantages that distinguish it from competing approaches. The post-hoc explainability analysis demonstrates that the framework permits comprehensive feature importance characterisation and identification of degradation mode clusters, which is increasingly recognised as essential for industrial adoption where maintenance engineers require understanding of model behaviour to establish operational trust.

5.5. Computational efficiency

All experiments were performed in MATLAB® 2025b on a workstation equipped with an NVIDIA Quadro P2000 (5 GB GDDR5, 1024 CUDA cores), 192 GB of physical memory, and Microsoft Windows 11 Pro for Workstations. To provide a hardware-independent characterisation of the framework's computational cost, Table 6 reports the total trainable parameters and input feature dimensionality for the optimal configuration of each dataset.

The parameter counts range from approximately 249K (FD003) to 1.07 M (FD004), reflecting differences in both the optimal hyperparameter configurations (Table 3) and the number of sensors retained after variance-based filtering. The higher dimensionality of the multi-regime subsets (FD002, FD004), where all 21 sensors exhibit sufficient variance, contributes to their larger parameter counts. This modest model complexity indicates compatibility with deployment on standard industrial hardware without requiring high-end accelerators.

6. Conclusion

This paper has presented an explainable CNN-BiLSTM framework for Remaining Useful Life prediction that addresses two persistent challenges in prognostics: accuracy degradation under variable operating conditions and insufficient model interpretability.

The regime-normalised preprocessing pipeline, combining clustering-based regime identification with condition-specific z-score normalisation, effectively mitigates covariate shift induced by varying flight profiles. On the multi-condition C-MAPSS subsets, the framework achieves RMSE values of 12.44 (FD002) and 11.53 (FD004), representing improvements of 6.6% and 18.8% over the strongest competing methods. Performance on single-condition subsets remains competitive, with RMSE values of 11.40 (FD001) and 10.71 (FD003).

The degradation-aware feature engineering strategy has demonstrated substantial value through both ablation studies and explainability analysis. Removing delta-from-baseline features increases RMSE by up to 64.4%, confirming their critical role in capturing accumulated wear. The post-hoc analysis establishes that engineered features collectively account for 82.1% of predictive importance, with delta-difference and first-order difference features alone contributing 61.5%, providing evidence-based guidance for preprocessing pipeline design in related prognostic applications.

The explainability framework revealed three different degradation mode clusters within the FD001 test fleet, each exhibiting characteristic feature utilisation patterns. Cluster 1, comprising 77% of engines, relies on temperature-related delta-from-baseline features. Cluster 2, containing 11% of early-degradation engines, exhibits elevated importance for pressure-ratio indicators. Cluster 3, encompassing 12% of near-failure engines, demonstrates a shift towards raw sensor magnitudes. This heterogeneity indicates that the model implicitly learns to distinguish degradation signatures rather than applying a uniform prediction strategy. A particularly significant finding concerns the relationship between degradation stage and prediction accuracy. Near-failure engines achieve mean absolute errors of 2.28 cycles, compared to 15.68 cycles for early-life engines. This inverse relationship represents a desirable characteristic for maintenance applications, as prediction precision is highest when accurate forecasts are most consequential for operational decisions.

A number of limitations warrant acknowledgement. The experimental validation relies exclusively on the C-MAPSS benchmark, a simulation environment that may not fully capture real-world complexities including sensor noise, missing data, and measurement artefacts. The framework lacks explicit uncertainty quantification mechanisms; whilst standard deviations across random seeds provide rudimentary variability measures, this approach does not distinguish between epistemic and aleatory uncertainty, both essential for risk-informed maintenance planning. The post-hoc explainability analysis was conducted only on FD001, and generalisability of degradation mode findings to multi-condition scenarios remains to be established. Additionally, the regime normalisation strategy requires prior specification of operating condition counts, which may be unavailable for novel systems.

These limitations suggest multiple directions for future investigation. Validation on operational industrial datasets exhibiting incomplete run-to-failure trajectories and sensor failures would establish practical applicability beyond benchmark performance.

Integration of principled uncertainty quantification methods would enable calibrated confidence intervals essential for risk-informed maintenance policies. Three candidate approaches are particularly compatible with the proposed architecture. First, MC-Dropout, which is directly applicable since the framework already employs dropout layers, would require only that dropout remain active during inference to generate a distribution of predictions from which confidence bounds can be derived. Second, deep ensembles, which the current multi-seed evaluation partially approximates, could be formalised by training a set of independently initialised models and aggregating their predictions to estimate both mean RUL and predictive variance. Third, evidential deep learning could replace the regression head with a layer parameterising a higher-order distribution, enabling the model to distinguish between epistemic uncertainty (arising from limited training data) and aleatory uncertainty (arising from inherent system variability) within a single forward pass. Extension of the explainability analysis to multi-condition subsets would strengthen claims regarding degradation mode identification under variable operating regimes. In particular, applying this analysis to multi-fault subsets such as FD003 could determine whether the clustering approach identifies physically distinct failure mechanisms rather than degradation stages alone. Further development could address adaptive regime detection through online clustering or change-point algorithms, eliminating requirements for prior specification of operating condition counts. Transfer learning techniques also warrant exploration for scenarios where labelled run-to-failure data are scarce, a common constraint in industrial settings where safety policies preclude operation until failure.

Author contributions

CRediT: **Nima Rezazadeh:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing; **Giuseppe Lamanna:** Investigation, Writing – review & editing; **Francesco Caputo:** Investigation, Supervision, Writing – review & editing; **Mario De Oliveira:** Investigation, Writing – review & editing; **Alessandro De Luca:** Investigation, Supervision, Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research study was carried out in the framework of the project Unione europea – Next Generation EU, as part of Prin 2022 PNRR – D.D. n. 1409 del 14–09-2022 program.

Data availability statement

The used dataset is publicly accessible at https://data.nasa.gov/dataset/cmapss-jet-engine-simulated-data.

References

[1] Yu S, Wu Z, Zhu X, et al. A domain adaptive convolutional LSTM model for prognostic remaining useful life estimation under variant conditions. In: 2019 Prognostics and System Health Management Conference (PHM-Paris). Paris, France: IEEE; 2019. p. 130–137.

[2] Asif O, Haider SA, Naqvi SR, et al. A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset. IEEE Access. 2022;10:95425–95440. doi: 10.1109/ACCESS.2022.3203406

[3] Hossain MI, Bin Helal S, Ray K, et al. Remaining useful life prediction for turbojet engines using deep CNN on raw sensor data. In: 2024 27th International Conference on Computer and Information Technology (ICCIT). Cox's Bazar, Bangladesh: IEEE; 2024. p. 2098–2103.

[4] Lei Y, Li N, Lin J. A new method based on stochastic process models for machine remaining useful life prediction. IEEE Trans Instrum Meas. 2016;65(12):2671–2684. doi: 10.1109/TIM.2016.2601004

[5] Chen Y, Liu C. A sample-efficient transfer learning framework for industrial remaining useful life prediction leveraging large language models. Reliab Eng Syst Saf. 2026;268:111980. doi: 10.1016/j.ress.2025.111980

[6] Rezazadeh N, De Luca A, Perfetto D, et al. Domain-adaptive graph attention semi-supervised network for temperature-resilient SHM of composite plates. Sensors. 2025;25(22):6847. doi: 10.3390/s25226847

[7] Zhang K, Liu Y, Zou Y, et al. Degradation trend feature generation improved rotating machines RUL prognosis method with limited run-to-failure data. Meas Sci Technol. 2023;34(7):075019. doi: 10.1088/1361-6501/accbde

[8] Benhanifia A, Ben CZ, Oliveira PM, et al. Systematic review of predictive maintenance practices in the manufacturing sector. Intell Syst Appl. 2025;26:200501.

[9] Ramasso E, Saxena A. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. Int J Progn Health Manag. 2020;5(2). doi: 10.36001/ijphm.2014.v5i2.2236

[10] Rezazadeh N, Perfetto D, De Luca A, et al. Ensemble learning for estimating remaining useful life: incorporating linear, KNN, and Gaussian process regression. In: Fera M, Caterino M, Macchiaroli R, Pham DT, editors. Advances in Remanufacturing. IWAR 2023. Lecture Notes in Mechanical Engineering. Caserta, Italy: Springer, Cham; 2024. https://doi.org/10.1007/978-3-031-52649-7_16

[11] Fallahy S, Rezazadeh N. Marble-da: masonry analysis with robust, batch-normalised, label-free, explainable domain adaptation for crack detection. J Building Eng [Internet]. 2025;116:114673. doi: 10.1016/j.jobe.2025.114673

[12] Yu W, Sun X, Yu Z, et al. Controllable data augmentation and application of multiaxial fatigue experiments by fatigue conditional generative adversative network model. Int J Fatigue. 2026;202:109216. doi: 10.1016/j.ijfatigue.2025.109216

[13] Abdelghafar S, Khater A, Wagdy A, et al. Aero engines remaining useful life prediction based on enhanced adaptive guided differential evolution. Evol Intel. 2024;17(2):1209–1220. doi: 10.1007/s12065-022-00805-z

[14] Ramasso E. Investigating computational geometry for failure prognostics. Int J Progn Health Manag. 2020;5(1). doi: 10.36001/ijphm.2014.v5i1.2205

[15] Mohammadi Foumani N, Miller L, Tan CW, et al. Deep learning for time series classification and extrinsic regression: a current survey. ACM Comput Surv. 2024;56(9):1–45. doi: 10.1145/3649448

[16] Shi J, Zhong J, Zhang Y, et al. A dual attention LSTM lightweight model based on exponential smoothing for remaining useful life prediction. Reliab Eng Syst Saf. 2024;243:109821. doi: 10.1016/j.ress.2023.109821

[17] Aydemir G, Acar B. Anomaly monitoring improves remaining useful life estimation of industrial machinery. J Manuf Syst. 2020;56:463–469. doi: 10.1016/j.jmsy.2020.06.014

[18] Jiang Y, Vaicaitis A, Dooley J, et al. Efficient neural networks on the edge with FPGAs by optimizing an adaptive activation function. Sensors. 2024;24(6):1829. doi: 10.3390/s24061829
[19] Sihman Bharattej RRVS, Aich M, Alzaidi A, et al. Edge-optimized LSTM framework for blockchain-enabled supply chain monitoring. In: 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT). Bidar, Karnataka, India: IEEE; 2025. p. 1–6.
[20] Zhao C, Huang X, Li Y, et al. A double-channel hybrid deep neural network based on CNN and BiLSTM for remaining useful life prediction. Sensors. 2020;20(24):7109. doi: 10.3390/s20247109
[21] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf [Internet]. 2018;172:1–11. doi: 10.1016/j.ress.2017.11.021
[22] Keshun Y, Puzhou W, Yingkui G. Toward efficient and interpretative rolling bearing fault diagnosis via quadratic neural network with bi-LSTM. IEEE Internet Things J. 2024;11(13):23002–23019. doi: 10.1109/JIOT.2024.3377731
[23] Guo J, Lei S, Du B. MHT: a multiscale hourglass-transformer for remaining useful life prediction of aircraft engine. Eng Appl Artif Intell. 2024;128:107519. doi: 10.1016/j.engappai.2023.107519
[24] Lin L, Wu J, Fu S, et al. Channel attention & temporal attention based temporal convolutional network: a dual attention framework for remaining useful life prediction of the aircraft engines. Adv Eng Inf. 2024;60:102372. doi: 10.1016/j.aei.2024.102372
[25] Boujamza A, Lissane Elhaq S. Attention-based LSTM for remaining useful life estimation of aircraft engines. IFAC-Papersonline. 2022;55(12):450–455. doi: 10.1016/j.ifacol.2022.07.353
[26] Keshun Y, Zengwei L, Ronghua C, et al. A novel rolling bearing fault diagnosis method based on time-series fusion transformer with interpretability analysis. Nondestr Test Evaluation. 2024:1–27. doi: 10.1080/10589759.2024.2425813
[27] Siahpour S, Li X, Lee J. A novel transfer learning approach in remaining useful life prediction for incomplete dataset. IEEE Trans Instrum Meas. 2022;71:1–11. doi: 10.1109/TIM.2022.3162283
[28] Li X, Wang H, Zhu G. Tfmpinn: a meta-learning approach for remaining useful life prediction via time-domain latent encoding and frequency-domain physics-informed neural networks. Nondestr Test Evaluation. 2026:1–31. doi: 10.1080/10589759.2026.2633576
[29] Wang Y, Li M, Liu S, et al. Degradation manifold dynamic consistency network for probabilistic remaining useful life prediction. Nondestr Test Evaluation. 2026:1–27. doi: 10.1080/10589759.2026.2633577
[30] Rezazadeh N, De Luca A, Lamanna G, et al. A novel interpretable domain adaptive framework for robust damage detection in composite structures under environmental variability. Struct Health Monit. 2026. doi: 10.1177/14759217261433879
[31] Li D, Chen J, Huang R, et al. Sensor-aware CapsNet: towards trustworthy multisensory fusion for remaining useful life prediction. J Manuf Syst. 2024;72:26–37. doi: 10.1016/j.jmsy.2023.11.009
[32] You K, Gu Y, Shao H, et al. A liquid-impulse neural network model based on heterogeneous fusion of multimodal information for interpretable rotating machinery fault diagnosis. Mech Syst Signal Process. 2026;246:113923. doi: 10.1016/j.ymssp.2026.113923
[33] Keshun Y, Zengwei L, Yingkui G. A performance-interpretable intelligent fusion of sound and vibration signals for bearing fault diagnosis via dynamic CAME. Nonlinear Dyn. 2024;112(23):20903–20940. doi: 10.1007/s11071-024-10157-1
[34] Fisher A, Rudin C, Dominici F. All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. J Mach Learn Res. 2019;20:20. doi: 10.1080/01621459.1963.10500830

[35] Wang C, Sun J, Meng X, et al. Multi-stage degradation feature with dynamic feedback mechanism for remaining useful life prediction. Nondestr Test Evaluation. 2025:1–34. doi: 10.1080/10589759.2025.2453907
[36] Liu Y, Frederick DK, Decastro JA, et al. User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS): version 2. NASA/TM-2012-217432. Cleveland (OH): National Aeronautics and Space Administration, Glenn Research Center; 2012.
[37] Zheng S, Ristovski K, Farahat A, et al. Long short-term memory network for remaining useful life estimation. In: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, Texas, USA; 2017. p. 88–95.
[38] Wang J, Wen G, Yang S, et al. Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network. In: 2018 Prognostics and System Health Management Conference (PHM-Chongqing). Chongqing, China: IEEE; 2018. p. 1037–1042.
[39] Deng S, Zhou J. Prediction of remaining useful life of aero-engines based on CNN-LSTM-attention. Int J Comput Intell Syst. 2024;17(1):232. doi: 10.1007/s44196-024-00639-w
[40] Yu B, Guo H, Shi J. Remaining useful life prediction based on hybrid CNN-BiLSTM model with dual attention mechanism. Int J Electr Power Energy Syst. 2025;172:111152. doi: 10.1016/j.ijepes.2025.111152
[41] Zhevnenko D, Kazantsev M, Makarov I. Interaction models for remaining useful lifetime estimation. J Ind Inf Integr. 2023;33:100444. doi: 10.1016/j.jii.2023.100444