Turbofan Engine Remaining Useful Life Prediction based on Physics-aware Hybrid Framework and Fatigue Cycles

Slawomir Szrama[1]

[1] *Institute of Powertrains and Aviation, Poznan, Piotrowo 3, 60-965, Poland*
slawomir.szrama@put.poznan.pl

ABSTRACT

Accurate remaining useful life (RUL) prognostics for turbofan engines are critical for advancing aviation safety and optimizing condition-based maintenance. While conventional methods rely on static service intervals or purely data-driven models, this study introduces a physics-aware hybrid framework that synergizes low-cycle fatigue (LCF) dynamics with a multi-architecture neural network. The framework addresses two fundamental limitations in existing approaches: (1) the omission of material fatigue mechanisms in purely data-driven models, and (2) the limited generalizability of simulated training data to real-world operational variability. The proposed model achieves superior degradation tracking by integrating fatigue cycle analytics, derived from engine operational stress profiles, with a hybrid neural architecture comprising Long Short-Term Memory (LSTM) networks for temporal dependencies, Temporal Convolutional Networks (TCN) for local feature extraction, and Multi-Head Self-Attention (MHSA) layers for dynamic weighting of critical sensor inputs. Comprehensive evaluation on both a 12-year real-world fleet dataset (53 F100-PW-229 engines) and the NASA C-MAPSS benchmark demonstrates significant improvements in predictive performance, with RMSE reductions of 29-48% compared to state-of-the-art hybrid models. The results underscore the potential of physics-aware deep learning to revolutionize predictive maintenance practices in aviation.

1. INTRODUCTION

1.1 Background and motivation

Accurate remaining useful life (RUL) prediction for turbofan engines is critical for aviation safety and cost effective condition based maintenance. Engine components degrade over time mainly due to low cycle fatigue (LCF) caused by large throttle transients during start, take off, and shutdown (Li et al., 2025). Data driven deep learning models can learn complex sensor patterns, but they lack explicit knowledge of the underlying fatigue physics, which limits their generalisability and makes them prone to unrealistic extrapolation. Embedding LCF principles, strain energy density, and cycle counting directly into the learning process can create more robust and interpretable prognostics.

1.2 Research gap

Traditional RUL prediction methods exhibit two fundamental limitations. First, purely data-driven models operate as black boxes that learn statistical correlations without incorporating physical knowledge about damage mechanisms. They extrapolate poorly outside training distributions and can produce predictions that violate physical constraints (Thakkar & Chaoui, 2023). Second, most existing studies rely exclusively on simulated datasets such as C-MAPSS (Sharma, 2024), which cannot capture the full complexity of real-world operational variability, including sensor noise, missing data, unrecorded maintenance events, and diverse operating conditions encountered in actual fleets (Szrama & Lodygowski, 2024).

A critical omission in current approaches is the lack of explicit fatigue modeling. Low-cycle fatigue (LCF) resulting from large-amplitude stress cycles during engine start, takeoff thrust transients, and shutdown is the life limiting mechanism for the majority of critical rotating components; indeed, airworthiness regulations explicitly require LCF based life limits for all rotor structural parts whose failure could endanger the aircraft (Li et al., 2025). While recent hybrid frameworks (e.g., CNN-LSTM, autoencoder-BiLSTM) improve feature extraction, they lack explicit mechanisms to model fatigue-induced degradation. Physics-informed approaches such as AttnPINN (Liao et al., 2023) and Cau-BiMamba-LSTM (Li et al., 2025) incorporate general physical constraints or causal relationships but do not compute explicit fatigue damage from operational cycles.

This study bridges these gaps by proposing a dual-domain framework that integrates LCF dynamics with multi-

https://doi.org/10.36001/IJPHM.2026.v17i1.4733

architecture deep learning, validated on both real-world and simulated data.

1.3 Contributions

The main contributions of this work are:

(a) Fatigue-cycle analytics module. A feature extraction module that computes cumulative strain energy density and LCF cycle counts from engine thrust variations, explicitly encoding damage accumulation mechanisms that are absent from purely data-driven models.

(b) Multi-architecture neural network. A hybrid LSTM-TCN-MHSA architecture that captures local transients (TCN), long-term temporal dependencies via a two-layer bidirectional LSTM encoder, and dynamically weights the most fatigue-sensitive sensor channels (MHSA), ensuring that the physics features are optimally fused with operational data.

(c) Comprehensive dual-domain validation. The proposed model is evaluated on a proprietary 12-year fleet dataset (53 F100-PW-229 engines) and the NASA C-MAPSS benchmark, demonstrating real-world applicability while preserving comparability with published results, thereby closing the gap between simulated and operational performance.

(d) Ablation and benchmarking studies. Controlled experiments quantify the contribution of the physics features (53 % RMSE reduction on real-world data) and benchmark the model against 13 state-of-the-art methods, confirming the advantage of embedding explicit fatigue knowledge into deep learning.

2. LITERATURE REVIEW

2.1 Evolution of predictive maintenance methodologies

The evolution of predictive maintenance methodologies over the past few decades reflects the increasing complexity of modern engineering systems. Early approaches to RUL prediction primarily involved physics-based models. These models, grounded in the principles of thermodynamics and material science, attempt to simulate the degradation process by modeling the accumulation of damage due to repetitive stress and thermal cycling. Although these models offer the advantage of interpretability and can be tailored to specific components, they require detailed knowledge of material properties and operating conditions. Moreover, they often rely on simplifying assumptions that limit their applicability in complex, real-world environments.

With the advent of machine learning, researchers began exploring data-driven methods for RUL prediction. Initial efforts in this area focused on traditional statistical techniques and shallow machine learning models such as support vector machines (SVMs), decision trees, and random forests. These methods were used to identify patterns and correlations in historical sensor data, providing early warnings of engine degradation.

Predicting the remaining useful life (RUL) of engines, particularly aircraft engines, is crucial for ensuring safety, optimizing maintenance schedules, and reducing operational costs. Early prediction of the engine RUL expiration is extremely significant for aircraft safety reasons. Various machine learning (ML) and deep learning (DL) techniques have been developed and applied to achieve accurate RUL predictions. The estimation of engine RUL is influenced by a great deal of factors such as multiple operating conditions and nonlinear degradation features (Ren et al., 2022).

2.2 Machine learning and deep learning approaches

Machine learning models (Szrama, 2024; Szrama & Lodygowski, 2024b), particularly deep learning approaches, have shown promising results in RUL predictions. However, challenges persist in accurately estimating RUL, especially in handling complex sensor data and nonlinear degradation trends. Various methods, including hybrid solutions and data-driven approaches, have been proposed to calculate engine remaining useful life, demonstrating the ongoing efforts to improve prognostics and health management for engines.

Machine learning algorithms have been applied in many cases to determine engine RUL and predict its expiration. For instance, Thakkar and Chaoui (2023) presented a prediction framework for the RUL of aircraft engines using machine learning (ML) techniques, specifically a Deep Layer Recurrent Neural Network (DL-RNN) model. The proposed method is compared to other ML algorithms, such as Multilayer-Perceptron (MLP), Cascade Forward Neural Network (CFNN), and Nonlinear Auto Regressive Network with Exogenous Inputs (NARX). Sharma (2024) tested various ML algorithms in an Industry 4.0 environment, proving Random Forest and LightGBM (LGBM) effectiveness in different degradation conditions. Zheng et al. (2022) studied the RUL prediction algorithm for turbofan engines based on the fused deep learning models. A multimodal deep learning approach based on CNN (Convolutional Neural Network) and an enhanced Bi-LSTM (Bidirectional Long Short-Term Memory) network was proposed.

The introduction of deep learning revolutionized the field of predictive maintenance. Recurrent Neural Networks (RNNs) and their variants, particularly Long Short-Term Memory (LSTM) networks, became popular due to their ability to learn complex temporal relationships (Lan et al., 2018; Wang et al., 2018; Li et al., 2018). LSTM networks are designed to overcome the vanishing gradient problem, which often hinders the performance of traditional RNNs when processing long sequences of data. Studies employing LSTM-based models have reported significant improvements in RUL prediction accuracy, particularly when dealing with

noisy and high-dimensional sensor data (Sharma & Bojjagani, 2024; Cheng et al., 2020; Wang et al., 2021; Chen et al., 2020).

In parallel with the development of LSTM networks, Convolutional Neural Networks (CNNs) gained prominence for their powerful feature extraction capabilities. Although CNNs are traditionally associated with image processing tasks, their application to time-series data has also been explored. By applying one-dimensional convolutions, CNNs can extract localized features from sensor data, such as sudden spikes or drops in temperature or pressure. However, while CNNs are effective at identifying short-term anomalies, they are less adept at capturing long-term trends, which are critical for accurate RUL estimation. Temporal Convolutional Networks (TCNs) emerged as a solution to some of these limitations. TCNs use causal convolutions to ensure that the model's predictions are based only on past and present data, thereby preserving the temporal order. Their ability to handle long-range dependencies and maintain stable gradients during training has made them an attractive option for time-series prediction tasks, including RUL estimation.

To enhance the performance of LSTM networks, hybrid models combining LSTM with other neural networks like Convolutional Neural Networks (CNN) have been proposed. These models leverage CNNs for feature extraction and LSTMs for temporal sequence processing, resulting in improved prediction accuracy and robustness (Liu et al., 2024; Wang et al., 2022; Zhang et al., 2020).

Deep learning approaches, such as stacked sparse autoencoders and logistic regression, have been proposed by Ma et al. (2018) to automatically extract performance degradation features from multiple sensors on aircraft engines and predict RUL. Recurrent neural networks (RNNs), including long short-term memory (LSTM) networks and gated recurrent units (GRUs), have been used for RUL prediction, leveraging temporal dependencies in the data to improve prediction accuracy by Sharma et al. (2024). Extracting useful degradation features from multi-sensor data with complex correlations is a key technical problem that has hindered the implementation of degradation assessment for aircraft engines. Deng et al. (2020) proposed aircraft engines' remaining useful life prediction methods with long-short term differential techniques for feature processing and forward differential feature construction. Wang et al. (2023) used random-coefficient regression (RCR) to model the degradation process and estimate the underlying failure threshold of aero-engines.

Recent advances in unsupervised and adaptive methods have further expanded the toolkit for RUL prediction. Lodygowski and Szrama (2025) developed an unsupervised classification framework using autoencoders and Gaussian mixture models, demonstrating that latent representations learned from unlabeled data can effectively capture degradation patterns without requiring labeled failure instances. Their approach provides a foundation for hybrid supervised-unsupervised methods that combine the strengths of both paradigms. Szrama (2025) extended this work by proposing a dynamically adaptive predictive maintenance framework that compares autoencoders, LSTMs, and Gaussian processes, showing that adaptive methods can track evolving degradation patterns more accurately than static models. These studies highlight the value of combining multiple methodological approaches which is the principle extended in our work through the integration of fatigue physics with multi-architecture deep learning.

2.3 Attention mechanisms and advanced architectures

More recently, attention mechanisms have been integrated into deep learning models to enhance their ability to focus on the most relevant parts of the input data. Multi-Head Self-Attention (MHSA) is one such mechanism that has proven effective in various domains, including natural language processing and computer vision. In engine RUL prediction, MHSA enables the model to assign dynamic weights to different sensor inputs, effectively highlighting the most critical signals that indicate degradation.

Although CNNs, RNNs, and attention mechanisms have improved RUL prediction, they are generally data-driven and omit explicit physics. Liao et al. fused self-attention with a physics-informed NN (AttnPINN) and achieved state-of-the-art performance on CMAPSS (Liao et al., 2023), but their model used a single LSTM backbone without fatigue-cycle inputs. Li et al. introduced a BiLSTM-based model with causal discovery (Cau-BiMamba-LSTM) (Li et al., 2025); this model leverages attention on causal features but does not compute physical fatigue damage. In contrast, our hybrid framework uniquely combines multiple network types with LCF-derived damage indicators, as highlighted below.

Another data-driven example of RUL prediction was presented by Peng et al. (2020), where a summation wavelet-extreme learning machine was used to determine engine key performance indicators. Based on them, degradation states were determined by subtractive-maximum entropy fuzzy clustering to calculate the RUL of engines. Furthermore, there are some hybrid methods, combining data-driven approaches with domain knowledge. For instance, Ibrahima and Meriem (2023) introduced the RUL estimation method with Machine Learning techniques, guided by domain knowledge. This hybrid method proposed engine predictive maintenance development and potential for improved RUL predictions.

Having analyzed all the techniques and models, it might be concluded that deep learning neural networks allow for high precision, ensemble neural networks (Wang et al., 2022) provide low RMSE, while with multimodal deep learning, accuracy improvement might be achieved. These methodologies highlight the advancements in RUL

prediction, emphasizing the importance of combining various ML, DL, and physics-based techniques to handle the complexities of engine degradation and improve predictive maintenance strategies.

2.4 Summary of literature review

The literature review reveals that while significant progress has been made in data-driven RUL prediction, several gaps remain. First, most studies rely exclusively on simulated datasets, with limited validation on real-world operational data. Szrama and Lodygowski (2024) highlighted the significant differences between real-life engine operational data and simulated datasets, noting that models validated only on simulations may not generalize to actual operations. Second, explicit modeling of fatigue mechanisms, particularly low-cycle fatigue is largely absent from current deep learning approaches, despite LCF being a dominant failure mode for rotating components. Third, while hybrid architectures combining multiple network types have been explored, the integration of physics-derived fatigue features with multi-head attention mechanisms remains unexplored. This study addresses these gaps by proposing a physics-aware hybrid framework validated on both real-world and simulated data.

3. PROBLEM FORMULATION

3.1 Mathematical Framework

The remaining useful life prediction problem can be formally defined as follows. Let $\mathcal{E} = \{e_1, e_2, \dots, e_N\}$ represent a fleet of N engines of identical type. For each engine e_i, we observe a multivariate time series:

$$\boldsymbol{X}_i = \{\boldsymbol{x}_i(t) \in \mathbb{R}^d \mid t = 1,2, \dots, T_i\} \quad (1)$$

where $\mathbf{x}_i(t)$ is a d-dimensional vector of sensor measurements at time step t, and T_i is the total number of observations for engine i. Each sensor measurement vector includes:

- Operational parameters: altitude, Mach number, throttle setting (Power Lever Angle)
- Physical sensor readings: temperatures (e.g., fan turbine inlet temperature), pressures, rotor speeds (N1, N2), fuel flows, vibration amplitudes
- Derived quantities: efficiency estimates, corrected parameters

For each observation, we also have access to fatigue-related quantities derived from operational history:

$$f_i(t) = LCF_{TAC}(t), SED_{cum}(t), Cyc1(t), Cyc2(t), HPTEff(t) \quad (2)$$

where $f_i(t)$ is fatigue feature vector, $LCF_{TAC}(t)$ is accumulated low-cycle fatigue cycles, $SED_{cum}(t)$ is cumulative strain energy density, $HPTEff(t)$ is High Pressure Turbine efficiency.

The target variable is the remaining useful life at each time step:

$$y_i(t) = T_i^{\text{failure}} - t \quad (3)$$

where T_i^{failure} is the cycle at which the engine i reaches end-of-life (defined by performance threshold or maintenance removal). The goal is to learn a function $\mathcal{F}$ that maps from observed data to RUL:

$$\hat{y}_i(t) = \mathcal{F}(\{\mathbf{x}_i(1:t)\}, \{\mathbf{f}_i(1:t)\}, \boldsymbol{\theta}) \quad (4)$$

where $\boldsymbol{\theta}$ represents model parameters learned from training data.

3.2 Assumptions

The problem formulation relies on the following assumptions:

A1: Monotonic degradation – Underlying physical degradation processes are irreversible, leading to monotonically decreasing RUL.

A2: Markov property – Future degradation depends on current state and accumulated damage history; however, LSTM architectures can capture longer-range dependencies.

A3: Sensor fidelity – Recorded sensor measurements provide sufficient information to infer underlying degradation states.

A4: Consistent failure definition – End-of-life is consistently defined across the fleet (either by performance threshold or maintenance practice).

A5: Fatigue as dominant mechanism – LCF is the primary damage mechanism for critical rotating components, justifying explicit modeling.

4. METHODOLOGY

4.1 Overview

The proposed methodology comprises three main phases: (1) engine performance data processing, including fatigue feature extraction and health indicator construction, (2) neural network design, training, and validation, and (3) RUL estimation and performance evaluation. Figure 1 presents a high-level flowchart of the complete pipeline, from raw data ingestion to final RUL prediction.

Phase 1 covers data preprocessing: multi-criteria feature quality assessment (monotonicity, prognosability, trendability), operating-regime normalisation, prognostic sensor selection, and weighted sensor fusion into a scalar Health Condition Indicator (HCI). Phase 2 defines the hybrid LSTM-TCN-MHSA neural network, which combines causal dilated temporal convolutions for multi-scale feature extraction, a two-stage bidirectional LSTM encoder for

temporal modelling, and a self-attention mechanism for context-aware sequence weighting, followed by a regression output layer. Phase 3 derives the RUL estimate from the predicted HCI trajectory and evaluates predictive accuracy against ground-truth values. Detailed descriptions of each stage are provided in sections below.

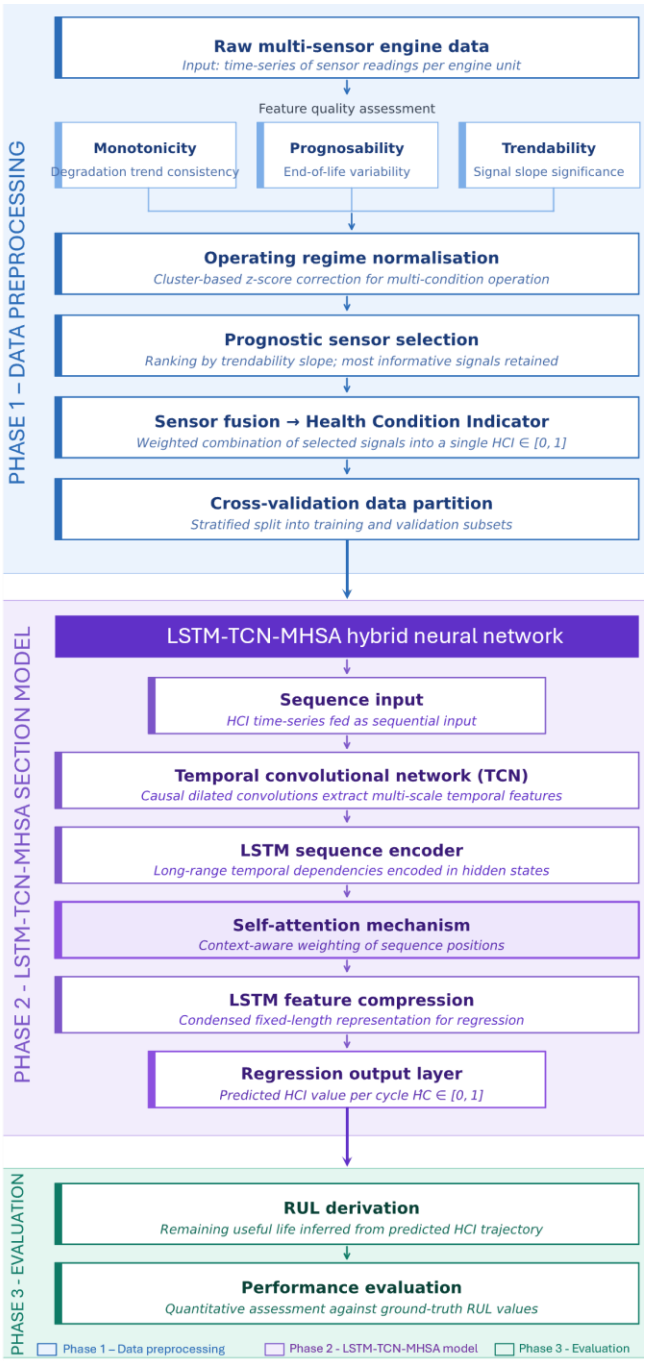

Figure 1. Methodology flowchart showing the complete pipeline from raw engine data to final RUL prediction.

Methodological innovations:

- **Fatigue-aware health indicator:** Accumulated LCF cycles are weighted by mission-specific stress amplitudes derived from Power Lever Angle (PLA) transitions. The health index fuses LCF metrics with sensor-derived features (e.g., HPT efficiency decay) via linear regression, suppressing noise from non-degradative signals.
- **Hybrid neural network architecture:** A TCN front-end processes sensor streams (5 Hz resolution) with dilated causal convolutions to capture short-term anomalies; a bidirectional LSTM models long-term degradation trends; multi-head self-attention dynamically reweights features based on fatigue relevance.
- **Real-world validation:** The model is evaluated on a proprietary 12-year dataset of 53 F100-PW-229 engines (89 parameters, take-off snapshots) and the NASA C-MAPSS benchmark, with robust preprocessing (moving median filtering, outlier removal) to address real-world data fidelity issues.

The following subsections detail each stage.

4.2 Data Loading and Description

The foundation of any predictive maintenance system is high-quality data. For this study, data were collected from a fleet of very advanced turbofan engines operating under diverse conditions. The sensors installed on these engines continuously monitor various parameters that are critical indicators of engine health. One of the important and sometimes crucial decisions engine maintenance tracking personnel must make is to decide which type of data should be applied for engine RUL estimation. There are many various data types recorded during engine operations. They were thoroughly described by Szrama (2019). The engine dataset that was used for the case study analysis is the engine performance data type.

The primary dataset comprises operational records from 53 F100-PW-229 turbofan engines collected over a 12-year operational period (2010-2022). These engines power military aircraft and operate under diverse mission profiles including training, combat exercises, and routine patrols. Data are recorded during the takeoff flight phase, when engines experience maximum thermal and mechanical stress, providing the most informative snapshot of health status.

Data characteristics:

- Total records: 43,472 individual takeoff snapshots
- Engines: 53 unique units (each with 200-1,500 records depending on operational tempo)
- Sensors: 12 physical sensors measuring temperatures, pressures, rotor speeds, fuel flows

- Derived parameters: 86 calculated parameters, including thermodynamic efficiencies, corrected flows, and performance margins
- Fatigue counters: Low-cycle fatigue accumulators (LCF~TAC~, Cyc1, Cyc2, MaxCycTime) computed by the Engine Diagnostic Unit
- Maintenance records: Removal dates and reasons for each engine

The dataset is proprietary and cannot be made publicly available due to military aviation security restrictions. However, detailed preprocessing steps and model implementation are provided to enable replication on similar datasets.

While loading the dataset into the data processing software, all the sensors' recorded parameters were restructured by converting the table of records to cell arrays or numeric arrays. In the following step, operational conditions and sensor data were identified for further analysis. The secondary engine dataset, used solely for validating the engine RUL prediction framework, was the NASA Turbofan Engine Degradation Dataset (CMAPSS). This is a widely used simulated dataset, employed in this study to demonstrate the effectiveness of the designed model and facilitate comparison with other models presented in scientific publications. Unlike the real-life data, C-MAPSS data is run-to-failure data, where engines reach their failure moment in the last recorded set of parameters. The CMAPSS dataset comprises four sub-datasets (FD001, FD002, FD003, and FD004), and the results presented in this paper for CMAPSS are typically the mean performance across these sub-datasets.

Fatigue features for C-MAPSS dataset

The NASA C-MAPSS simulated data contain no pre-computed fatigue counters. We therefore derive surrogate fatigue indicators from the time series of the operational settings (altitude, Mach number, throttle resolver angle, TRA). A full throttle cycle is identified each time the TRA crosses predefined thresholds corresponding to idle and maximum power, analogous to the real-engine logic. The stress amplitude is approximated by the peak-to-peak change in fan speed (when available) or, as a fallback, by the TRA excursion magnitude, and the per-cycle strain energy is computed proportionally to $(\Delta TRA)^2$. These synthetic features (denoted LCF_TAC_sim and SED_cum_sim) are then accumulated and fed to the network in the same manner as the real-world fatigue features. While they lack the fidelity of on-board EDU logs, they encapsulate the dominant throttle-driven damage mechanisms and are shown to improve predictions (Table 2, Section 4).

4.3 Fatigue Feature Extraction

Low-cycle fatigue is the dominant damage mechanism for rotating parts subjected to large throttle transients. The Engine Diagnostic Unit (EDU) of the F100-PW-229 records fatigue cycles classified by the amplitude of the stress reversal. Three cycle types are used in this study:

- **Type I cycles**: Start to maximum power with augmentation to shutdown (full mission profile)
- **Type III cycles**: Idle to maximum power with augmentation and return to idle
- **Type IV cycles**: Cruise power to maximum power without augmentation and return to cruise

Each cycle type inflicts different levels of damage. Type I cycles, involving the full stress range from ambient to maximum augmented power, cause the greatest damage per cycle; Type III cycles cause intermediate damage; Type IV cycles cause the least damage. Note that Type II cycles are omitted from this classification as their definition does not appear in the available literature and they are not utilized as a standard parameter within this specific degradation framework.

Cumulative Fatigue Indicator

We compute total accumulated fatigue cycles using a weighted combination that reflects the relative damage contribution of each cycle type:

$$LCF_{TAC} = MAN_I + 0.25(LCF_{III} - MAN_I) + 0.025(LCF_{IV} - LCF_{III}) \quad (5)$$

where:

- MAN_I = Type I fatigue cycles (full mission cycles)
- LCF_{III} = Type III fatigue cycles (idle-to-max-augmented cycles)
- LCF_{IV} = Type IV fatigue cycles (cruise-to-max cycles)

The weighting factors (0.25, 0.025) are derived from stress amplitude analysis using engine thermodynamic models. Type I cycles receive full weight (implicitly 1.0) as they cause the greatest damage per cycle.

Strain Energy Density

To capture the physics of LCF, we compute fatigue-related quantities from each engine's duty cycles. Specifically:

- Load amplitude ($\Delta\sigma$): We approximate the stress range experienced by rotating components per cycle. Using the measured low-pressure shaft speed (N1) profile for each flight cycle, we count one LCF cycle each time the throttle goes from idle to max and back (or vice versa), following aircraft maintenance practice. The stress amplitude for that cycle is inferred from engine thermodynamic relationships (e.g. pressure maps) or approximated via ΔN1, assuming linear correlation between N1 fluctuation and stress change.
- Loading frequency: The number of LCF cycles per hour of flight (derived from N1 cycle counts) is used as a stress frequency parameter. This distinguishes heavy-use flights (many cycles) from light-use flights.

Using Δσ, and frequency, we compute the cumulative strain energy density (SED) for each engine over time. SED measures the total mechanical energy absorbed by a material due to cyclic loading and is commonly used in fatigue life estimation. In the proposed hybrid framework, a simplified model is used where the strain energy density per cycle is proportional to $(\Delta\sigma)^2$; the per-cycle energy scales with the square of the stress amplitude. The cumulative SED is obtained by summing the per-cycle contributions:

$$SED_{cum}(t) = \sum_{k=1}^{n(t)} \alpha(\Delta N\ 1_k)^2 \quad (6)$$

where $n(t)$ is the number of full LCF cycles recorded up to time t, and α is a scaling constant that depends on material properties and component geometry. Simultaneously, we count the cumulative number of LCF cycles for each time step. These physics-derived features (cumulative SED and cycle count) thus increase monotonically as the engine degrades.

The fatigue feature vector $f_i(t)$ was defined in Eq. (2); its components are extracted as described above. These fatigue features can be extracted from any turbofan engine equipped with a basic Engine Monitoring System that records PLA, shaft speeds, and time-at-condition; therefore, the approach is transferable to other engine types.

4.4 Data Splitting, Normalisation, and Preprocessing

To achieve reliable neural network predictions, the data are split into training and validation sets using a 5-fold cross-validation mechanism, which randomly assigns engines to folds while preserving the temporal order of records within each engine. This ensures that all engines appear in both training and evaluation across different folds, and the reported metrics are averaged over the folds.

Sensor readings are normalised to remove variability due to operational conditions and to align sensor data to a common scale. A normalisation function adjusts sensor data by correcting for biases introduced by different operating regimes (e.g., altitude, temperature, pressure). Specifically, for each observation, the sensor measurements are centred and scaled relative to the cluster mean and standard deviation of the operating condition to which the observation belongs.

Since real-life operational data are contaminated with noisy sensor readings and occasional incorrect values, techniques such as median filtering and wavelet transformation are applied to smooth the data and remove sporadic noise spikes. This step reduces the impact of transient errors that could adversely affect model training. In addition, columns containing data irrelevant for RUL prediction, such as Engine Control Unit serial number, aircraft type, or location are removed.

4.5 Feature Selection and Health Indicator Construction

Not all recorded engine features are equally informative for RUL estimation. A critical step is therefore to identify the most meaningful sensors. Three metrics are used to evaluate the degradation tracking ability of each sensor:

- **Monotonicity:** measures the consistency of the increasing or decreasing trend of a feature over time.
- **Prognosability:** quantifies how well a feature's variance at failure differs from its variance earlier in life.
- **Trendability:** assesses the similarity of degradation trends across different engines.

All sensor channels are ranked by these metrics, and the top eight sensors with the highest engine degradation reflection are selected. The identified sensors numbers 75, 43, 30, 26, 24, 17, 15, and 10 include the LCF accumulators (LCF_TAC, Cyc1, Cyc2), high-pressure turbine efficiency (HPTEff), total cycle time (MaxCycTime), fuel flow (WFMFC), fan turbine inlet temperature (FTIT), and engine operating time (EOT). This selection is confirmed by thermodynamic reasoning: LCF cycles directly reflect fatigue damage; operating time captures cumulative wear; turbine efficiency decay is a well-known degradation symptom, which in turn drives higher fuel flow and hotter turbine inlet temperatures.

Health Indicator Construction

A Health Condition Indicator (HCI) is built by fusing the selected sensors. The RUL for each sample is first transformed into a health condition metric by normalising RUL values to the range [0, 1], where 1 represents the earliest (healthiest) state and 0 the end of useful life. The eight most trendable sensor measurements are then linearly combined into a single HCI using regression weights obtained from a linear degradation model fitted on the training data:

$$HCI = \beta_0 + \sum_{j=1}^{8} w_j \cdot s_j \quad (7)$$

where s_j denotes the j-th selected sensor and w_j its weight. The resulting HCI trajectories are smoothed with a moving average filter to reduce residual noise. Figure 2 shows examples of the HCI curves for the training and validation datasets.

4.6 Hybrid LSTM-TCN-MHSA Neural Network Architecture

A hybrid neural network is designed to ingest sequential engine data and predict RUL. The input at each time step t consists of the normalised sensor vector concatenated with the physics-derived fatigue features (cumulative SED and cycle counts up to t). By feeding the physics features alongside raw sensor data, the network learns how these fatigue indicators influence the degradation trajectory.

The architecture (Fig. 3) comprises the following elements:

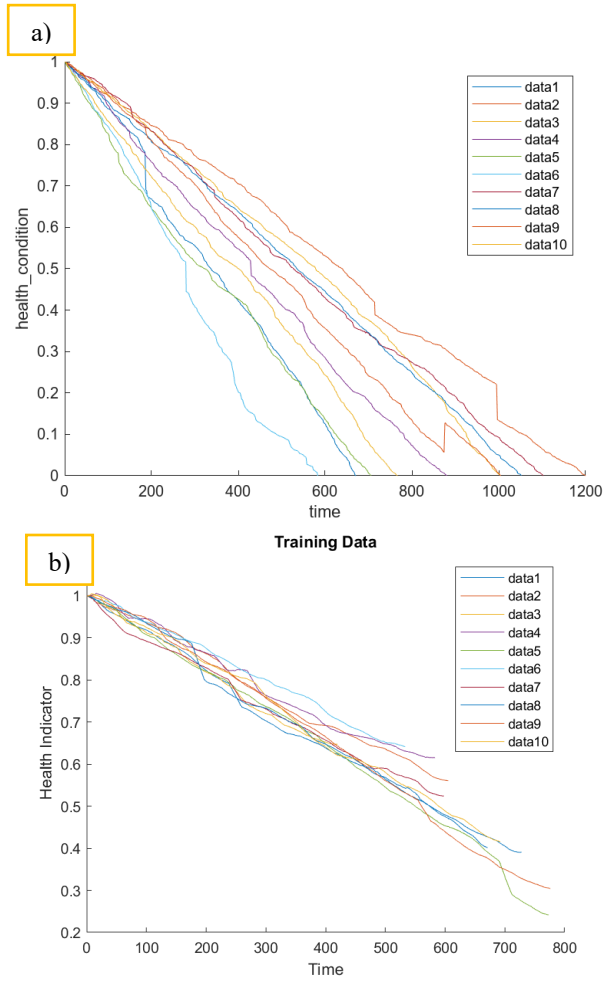

Fig. 2. Engine health indicator for each engine from the validation dataset (a) and Health Indicator for fused training data (b)

- Temporal Convolutional Network (TCN): A stack of three dilated, causal 1D convolutional layers (kernel size = 5, dilation rates [1, 2, 4], 64 filters per layer, ReLU activation) processes the input sequence. Dilated convolutions provide an exponentially large receptive field that grows with the number of layers; for a stack of l layers with dilation rates d_1,d_2,…,d_l, the receptive field is R=1+(k-1)∑_(i=1)^l▒d_i .With the chosen dilations [1, 2, 4] and kernel size k=5, this yields R=29. The TCN excels at capturing short term anomalies and local patterns in the sensor streams (Long et al., 2015; Van Den Oord et al., 2016).
- **Bidirectional LSTM layers:** Two LSTM layers with 256 and 128 hidden units, respectively, model long-range temporal dependencies. The first LSTM returns the full sequence, while the second returns only the last output. Dropout (0.3) is applied between the layers.
- **Multi-Head Self-Attention (MHSA):** A self-attention layer with 4 heads (key, query, value dimensions = 32, 32, 32) dynamically weights the importance of different time steps and sensor channels. The attention mechanism highlights the most degradation-relevant features and enhances interpretability.
- **Dropout and output layer:** A dropout layer (rate = 0.4) provides regularisation, and a fully connected layer with a single neuron produces the RUL prediction.

The TCN and LSTM blocks are complementary: the former efficiently extracts local, fine-grained patterns, while the latter captures the global degradation trend. The MHSA layer re-weights the combined representation, emphasising the most fatigue-sensitive channels.

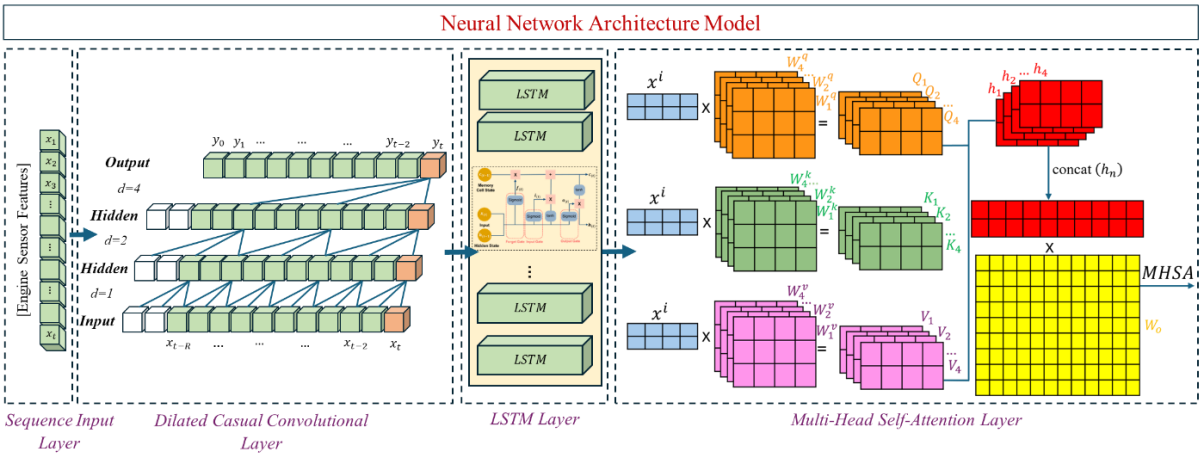

Fig. 3. Physics-aware hybrid LSTM-TCN-MHSA neural network architecture

4.7 Model Training, Validation, and Performance Metrics

Training is performed with the Adam optimiser (initial learning rate = 0.001, reduced on plateau with factor 0.5 and patience 10) and a batch size of 32. The maximum number of epochs is set to 200, with early stopping if the validation loss does not improve for 20 consecutive epochs. Gradient clipping (norm = 1.0) prevents gradient explosion. The complete set of hyperparameters is listed in Table 1.

Component	Parameter	Value
TCN	Number of layers	3
	Kernel size	5
	Dilation rates	[1, 2, 4]
	Number of filters per layer	64
	Activation	ReLU
	Padding	Causal
LSTM	Number of layers	2
	Hidden units per layer (first)	256
	Hidden units per layer (second)	128
	Bidirectional	Yes
	Dropout rate (between layers)	0.3
	Output mode	Sequence (first), Last (second)
Multi-Head Self-Attention	Number of heads	4
	Key dimension	32
	Query dimension	32
	Value dimension	32
	Dropout rate	0.1
Input/Output	Input sequence length	50 time steps
	Input features	Sensor readings + fatigue features
	Output	Single RUL value
Training	Optimizer	Adam
	Initial learning rate	0.001
	Learning rate schedule	Reduce on plateau (factor 0.5, patience 10)
	Batch size	32
	Maximum epochs	200
	Early stopping patience	20 (based on validation loss)
	Gradient clipping norm	1.0
	Validation split	20% of training data
Regularization	L2 weight decay	1×10^{-5}
	Dropout (final layer)	0.4

Table 1. Complete hyperparameter specifications for reproducibility

A 5-fold cross-validation protocol is used on the real fleet data: for each fold, the model is trained on four folds and tested on the held-out engines. On the C-MAPSS dataset, the standard train-test split is followed. In both cases, the test data are never seen during training, and overfitting is mitigated through early stopping, dropout, and L2 weight decay (1×10^{-5}).

The following performance metrics are computed:

- **Root Mean Square Error (RMSE):** $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i)^2}$
- **R-squared (R^2):** $1 - \frac{\sum(\tilde{y}_i - y_i)^2}{\sum(y_i - \overline{y})^2}$
- **Prognostic Score:** an asymmetric penalty function that penalises late predictions (predicted RUL > actual RUL) more heavily than early ones:

$$Score = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} exp\left(-\frac{\tilde{y}_i - y_i}{13}\right) - 1, (\tilde{y}_i - y_i) < 0 \\ \frac{1}{N}\sum_{i=1}^{N} exp\left(\frac{\tilde{y}_i - y_i}{10}\right) - 1, (\tilde{y}_i - y_i) \geq 0 \end{cases} \quad (8)$$

where $\tilde{y}_i$ is the predicted RUL and y_i the true RUL. A correct prediction contributes zero; early (safe-side) predictions are penalised less than late (dangerous) ones. This metric directly reflects the maintenance risk aversion required in aviation.

The network is trained to minimise the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i)^2 \quad (9)$$

Huber loss was also evaluated for robustness to outliers, but MSE provided slightly better convergence for this application.

5. RESULTS AND VISUALIZATION

To compare the proposed neural network architecture and its performance, it was decided to validate the designed engine RUL prediction model with the very popular (and the only one available) engine simulated data provided by NASA. This data set was generated with the C-MAPSS simulator. C-MAPSS stands for Commercial Modular Aero-Propulsion System Simulation. Simulated datasets consist of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine, i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation, which is unknown to the user. These wear and variation are considered normal, i.e., it is not considered a fault condition. Every engine data point was generated for three operational settings that have a substantial effect on engine performance. These settings are the combination of flight altitude, aircraft Mach speed, and throttle position. The data is contaminated with sensor noise.

Impact of physics features

Table 2 summarizes the performance of the physics-aware hybrid LSTM-TCN-MHSA neural network vs. the baseline model for the real-life engine dataset. On the real fleet data, the hybrid model achieved an average RMSE of 0.3336 cycles and a Score of 0.0184, while the non-physics-aware baseline achieved an RMSE of 0.7153 (0.3817 higher) and a Score of 0.0454 (0.027 higher). In numeric terms, including strain energy and cycle count, reduced RMSE by roughly 53% and decreased the cumulative Score by ~59%, relative to the pure data-driven model. This indicates that the physics inputs significantly improve prediction accuracy and reduce dangerous late-estimation errors. The improvement was

consistent across different flight profiles and degradation modes. These results demonstrate that the hybrid model effectively leverages the fatigue information. The severely negative R^2 (−134.2) obtained for the non-physics-aware model on the real-fleet data merits special comment. Because the Health Condition Indicator is normalised to $[0, 1]$, the total variability of the true values is small (total sum of squares ≈ 0.31). The non-physics-aware model, lacking any physical guidance, produces predictions with an RMSE of 0.7153, which is larger than the standard deviation of the target. Consequently, the sum of squared residuals is far larger than the total sum of squares, resulting in an R^2 that is extremely negative. This simply means that the model performs substantially worse than a constant prediction of the mean HCI. In contrast, the physics-aware variant reduces RMSE to 0.3336, bringing the residuals closer to the natural variability and yielding an R^2 of −0.422, still negative but far less extreme. These values underscore the necessity of explicit fatigue features for meaningful degradation tracking on real-world data.

A further ablation was performed by feeding only the physics feature vector $f_i(t)$ (Eq. 2) to the network, completely excluding raw sensor measurements. This "physics-only" variant achieves an RMSE of 0.4821 (real-fleet) and 11.507 (C-MAPSS), which is intermediate between the full physics-aware model and the non-physics-aware baseline (Table 2). The gap confirms that while the explicit fatigue indicators already carry substantial prognostic information, the best performance is attained when sensor data and physics features are fused, allowing the network to exploit complementary degradation signatures.

Engine Dataset	Model Variant	RMSE	Score	R-squared	MAE
Real-life operational data	Physics-aware	0.3336	0.0184	-0.4220	0.1881
	Non-physics-aware	0.7153	0.0454	-1.342E02	0.7095
	Physics features only	0.4821	0.0312	-0.6094	0.3815
C-MAPPS simulated	Physics-aware	9.4708	179.1070	0.5414	37.9907
	Non-physics-aware	14.2185	284.3321	0.3187	51.2843
	Physics features only	11.507	258.67	0.4018	44.21

Table 2. Comparison of the neural network's performance metrics

Error histograms for validation data and test engines of the C-MAPPS data and real-life operational data were presented in Fig. 4.

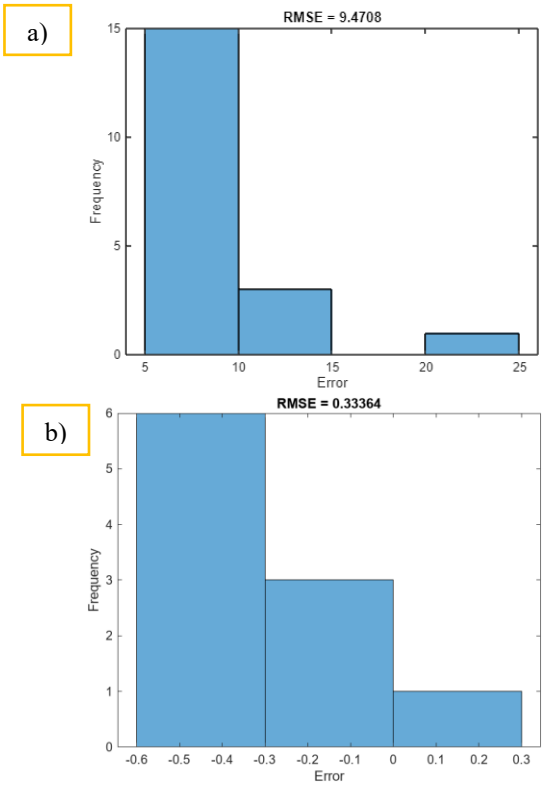

Fig. 4. Error histogram for validation data and test engines of the C-MAPPS data (a) and real-life operational data (b)

The trained model predicts RUL for validation data. Real vs. predicted RUL values are plotted for selected engines as presented in Fig. 5.

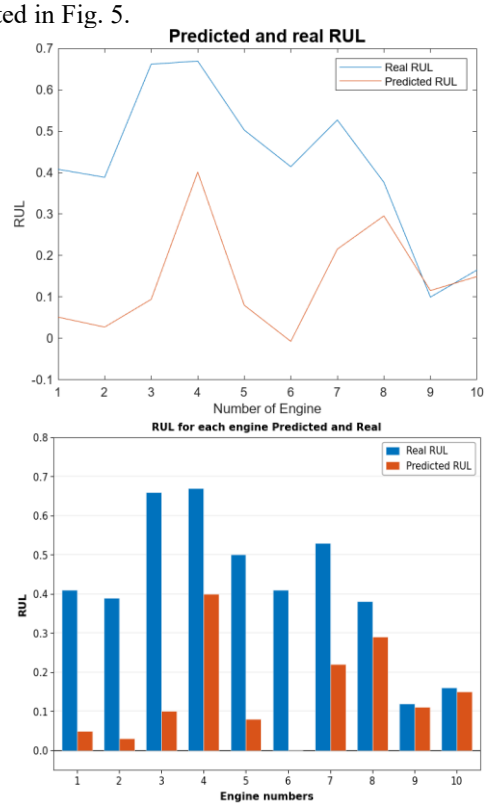

Fig. 5. Predicted and real RUL for validation data and test engines for real-life engine operational data

Real vs. predicted RUL is plotted for individual engines, showing: Actual RUL, Predicted RUL, and Smoothed predicted RUL, providing insights into model accuracy and trends (Fig. 6).

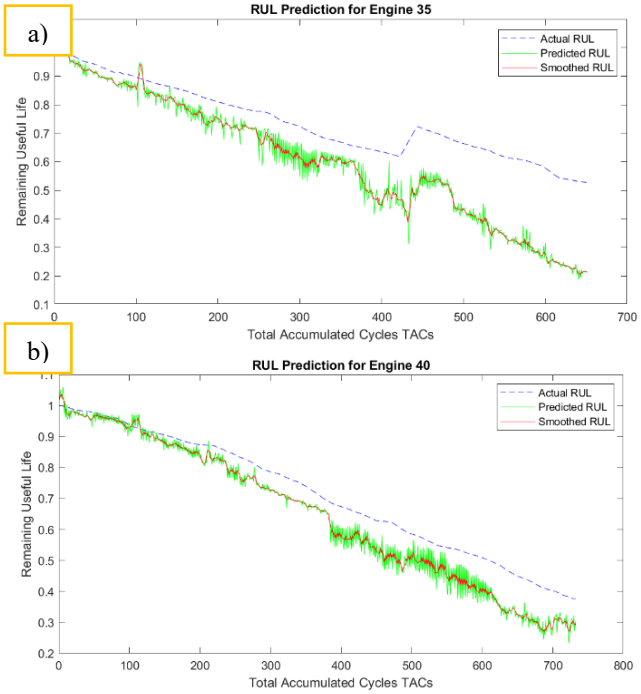

Fig. 6. True vs. predicted RUL for two CMAPSS test engines: engine 35 (a) and engine 40 (b). Dashed lines show the actual remaining cycles until failure, and solid lines show the model's predictions.

LSTM-TCN-MHSA Performance Validation

To validate the designed advanced turbofan engine RUL prediction model, it was decided to compare the achieved results to the C-MAPPS data. The same novel NN architecture was applied for RUL predictions on simulated engine data. Predicted and real RUL for validation data and two selected test engines based on simulated C-MAPPS data are presented in Fig. 7. In Fig. 7(a) (engine 84), the predicted RUL closely tracks the true RUL with minimal deviation, indicating high accuracy. In Fig. 7(b) (engine 96), the model slightly overestimates the RUL towards the end of life, suggesting a modest positive bias in this case. The magnitude of this overestimation is small — on the order of a few cycles — which is fully consistent with the aggregate RMSE of 9.47 reported in Table 2. These examples demonstrate that our model generally follows the degradation trend for simulated engines, with only small end-of-life errors.

a)

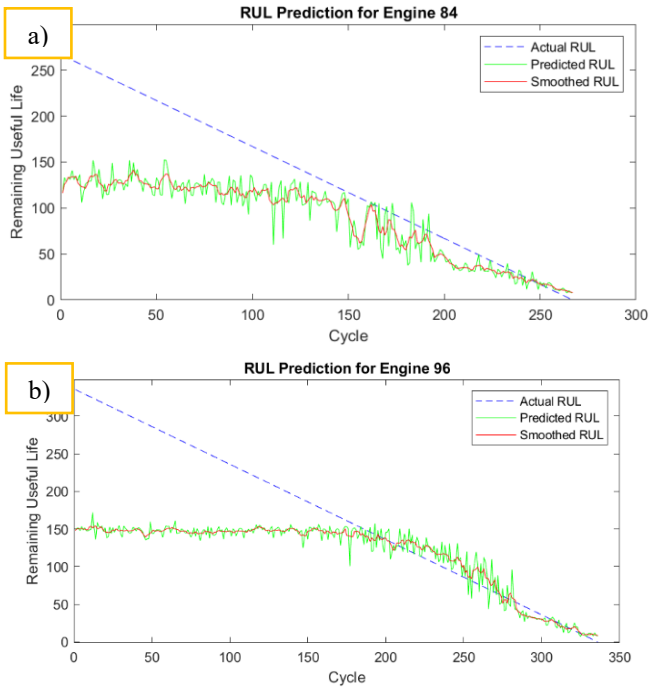

Fig. 7. True vs. predicted RUL for two CMAPSS test engines: engine 84 (a) and engine 96 (b). Dashed lines show the actual remaining cycles until failure, and solid lines show the model's predictions.

The designed hybrid network performance could also be presented by taking advantage of the box chart. This type of diagram displays various data information like the median, the lower and upper quartiles, the interquartile range IQR, and the minimum and maximum values that are not outliers. The horizontal line inside each box is the RUL median.

In Figure 8, engine RUL error results for the LSTM-TCN-MHSA neural network were presented with the box chart. For all the engine RUL error results, the most important statistical parameters were calculated and presented in the box chart figure. The boxplots in Figure 8 show that most C-MAPPS engines have errors within ±3 cycles, with median errors near zero, while for the real-life engines, RUL prediction errors fell mostly within ±0.3 cycles. Engine-level error dispersion is larger for the simulated CMAPSS set (a) than for the real fleet (b), reflecting dataset variability. On the C-MAPSS data, over 80 % of the test engines had an absolute RUL error below 5 cycles. On the real-fleet data, all engines had an absolute error beneath 0.6 cycles. These results confirm the robust performance of the proposed model in both domains.

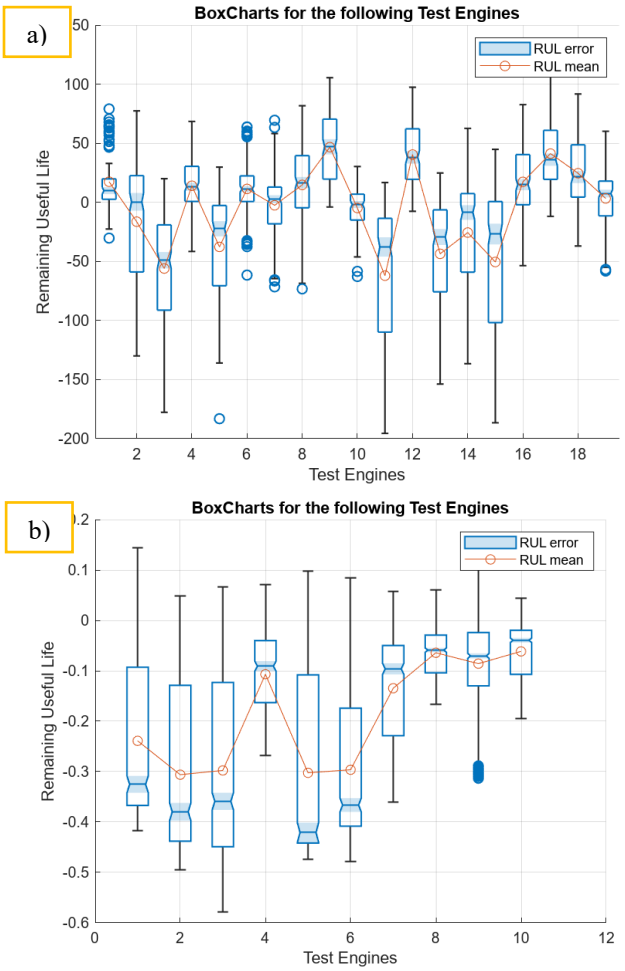

Fig. 8. Distribution of RUL prediction errors (in cycles) for: (a) the CMAPSS test set and (b) the real-world engine test set. Box plots show median, quartiles, and outliers of the error distributions.

Comparison with Other Models

Since the presented case study was conducted for the real-life turbofan engine operational data it was decided to compare the proposed LSTM-TCN-MHSA model to other existing RUL prediction models on the C–MAPSS dataset. Table 3 summarizes the results of the RUL prediction performance comparison between LSTM-TCN-MHSA and 11 existing deep neural networks specifically designed for aero-engine RUL prediction on the C-MAPSS dataset. The bolded results highlight performance results for the proposed model. As far as the RMSE is concerned, the prediction error result is the lowest among the NN architectures compared. The difference ranges from 29-48%. As for the score performance metric, the proposed model achieved the best result in 8 cases. The proposed model outperformed others in 4 cases by up to an order of magnitude. It proves that the proposed LSTM-TCN-MHSA model has a competitive advantage in RUL prediction performance on the C-MAPSS dataset compared to other models. This fact confirms that the proposed NN architecture can comprehensively extract the hidden information from the data as well as detect the most trendable engine operational data features. It results in a definitely better engine health status estimation and more accurate RUL prediction.

RUL Prediction Model	FD001÷FD004 Mean RMSE	FD001÷FD004 Mean Score
DABA (Liu et al., 2022)	15.65	951.00
BiGRU-TSAM (Zhang et al., 2022)	12.56	2.13
AE-MSEN (Xia et al., 2021)	15.58	12.31
MHN (Ren et al., 2022)	16.76	N/A
CNN-Wiener (Wen et al., 2023)	17.47	1606.00
AttnPINN (Liao et al., 2023)	17.21	1313.75
LSTM-CNN (Jiang et al., 2020)	18.26	2629.75
CBHRL (Zhu et al., 2024)	12.75	2.51
KGHM (Li et al., 2022)	13.18	250.99
TCN (Zha & Ye, 2024)	14.99	330.54
IDMFFN (Li et al., 2021)	16.24	1392.12
LSTM-TCN-MHSA	**9.4708**	**179.11**

Table 3. Comparison of RUL Prediction Performance between LSTM-TCN-MHSA and Other Models on C-MAPSS Dataset

Direct comparison with published results requires caution due to differences in preprocessing, train/test splits, and evaluation protocols. However, our C-MAPSS results (RMSE 9.47) compare favorably with recent physics-informed approaches (AttnPINN: 17.21, Cau-BiMamba-LSTM: reported qualitatively) and other hybrid architectures. These comparisons should be interpreted as approximate benchmarks rather than exact head-to-head evaluations, as methodological details may vary across studies. Nevertheless, the consistent improvement across multiple metrics suggests that explicit fatigue modeling adds value beyond existing approaches.

Table 4 demonstrates that the proposed physics-aware model reduces RMSE by 24–54 % relative to the literature architectures and yields the lowest Score, confirming its advantage on real-world fleet data. The non-physics variant underperforms because it lacks the explicit degradation indicators that are critical for this engine type.

Method	RMSE	Score
LSTM (bidirectional)	0.4412	0.0289
TCN	0.5017	0.0356
CNN-LSTM (Liu et al., 2024)	0.5394	0.0402
LSTM-TCN-MHSA (no physics)	0.7153	0.0454
Proposed (physics-aware)	0.3336	0.0184

Table 4. Performance comparison on the proprietary fleet dataset (53 F100-PW-229 engines, 5-fold cross-validation).

6. DISCUSSION AND SUMMARY

A comparison of the hybrid LSTM-TCN-MHSA neural network model performance results is presented in Table 2. One of the most popular performance metrics used in neural

network evaluation is RMSE. For the C-MAPPS data, which was the mean value of the result achieved for the whole collection of data, FD001÷004, RMSE was 9.47. What was unexpected was the fact that for the real-life engine operational data, RMSE was lower by up to an order of magnitude, reaching 0.33. It means that for the proposed physics-aware hybrid framework model, the average prediction error of the engine RUL was lower than for the simulated data. For the C-MAPPS data, the greatest number of engines, 15, fell into the prediction error between 5 and 10 cycles, while for the real-life data, 6 test engines fell into the bin with the prediction error between -0.3 and 0.6. What is extremely important to note is the fact that for one of the test engines (engine No.9), the prediction error was estimated as a positive value. It means that the engine health index calculated for the engine's remaining useful life was predicted as still valid, while in fact, it has already expired. Such a situation in air transport is very dangerous as it might result in engine in-flight failure with the consequences of an aircraft incident or even an accident. However, it is also worth mentioning that the prediction error was so close that, in real aircraft operations, such a situation is acceptable, as engines are usually scheduled for overhaul, and there is always a margin for the scheduled engine maintenance remaining on the engine. The results also indicate that even for the simulated engine fleet, with only one test engine, RUL was predicted to reach a residual number of cycles of 20-25. Such predictions result in significant money savings for airline operators.

Another important NN performance metric is the Score. For the C-MAPSS data, the result of 179 was achieved, which is promising compared to other models (Table 3). For the real-life engine operational data, the Score of the physics-aware model is only 0.0184, reflecting very small prediction errors and a well-balanced error distribution. According to Eq. (8), a Score near zero indicates that neither late nor early errors dominate appreciably; the model's predictions are tightly clustered around the true RUL. The Score values for the non-physics-aware baseline (0.0454) and the physics-only variant (0.0312) are larger, consistent with their higher RMSE. Overall, the very low Score obtained on real-world data confirms the high accuracy and safe-side bias of the hybrid model.

The poor performance of the non-physics-aware LSTM-TCN-MHSA relative to the simpler LSTM and TCN baselines (Table 4) deserves comment. With only 43 engines available for training in each cross-validation fold, the high-capacity hybrid architecture, which includes a three-layer dilated TCN, two bidirectional LSTM layers (256 + 128 units), and a four-head self-attention module, has far more trainable parameters than the stand-alone LSTM or TCN. Without the strong inductive bias provided by the physics-derived fatigue features, this model tends to memorise noise in the sensor signals rather than learning generalisable degradation trends. The simpler baselines, with fewer parameters, are less susceptible to such overfitting and therefore achieve lower RMSE. When the physics features (cumulative SED, LCF cycle counts, HPT efficiency) are included, they act as a powerful domain-informed regulariser: they force the network to focus on physically meaningful, monotonic indicators of damage. This prevents overfitting and allows the full representational capacity of the LSTM-TCN-MHSA architecture to be productively used, giving the best overall performance.

As far as the R-squared performance metric is concerned, for the C-MAPPS data, the result achieved was positive, reaching a 0.5 value, while for the real-life engine data achieved results were negative, close to -0.4. A negative R^2 indicates that the model's predictions are worse than a constant prediction of the mean; this is a legitimate outcome for out-of-sample evaluation, not a violation of any mathematical property. Having analyzed engine data and the results achieved, the reason for this might be the fact that engine degradation data in real-life operations is usually non-linear. As the R2 performance metric is trying to fit the degradation trend as a linear model, in the case of real data, it fails to capture patterns and trends in the recorded engine data.

While the Loss performance metrics are similar in both cases of the engine data, the Mean Absolute Error is significantly different, with an order of magnitude difference. In this case, MAE was lower for the real-life data, achieving 0.188, while for the C-MAPPS, it was about 38. What might be deduced from this is the fact that the proposed LSTM-TCN-MHSA engine RUL prediction model is working very well for the real data, since MAE explains how close engine RUL predictions are to the real values.

Having analyzed the box charts of the engine RUL predictions for the validation datasets of the C-MAPPS and real-life engine operational data, it might be deduced that there are definitely more outliers in the case of the simulated data. In the case of the real-life engine data, the proposed data filtering was definitely working better, removing any noisy data recorded from the sensors.

Discussed results and performance metrics achieved confirmed that the original and novel concept of engine RUL prediction based on a hybrid method combining physics-based and data-driven methods has proven its performance. A combination of the LSTM and TCN neural network layers was able to detect and sense long-term sequential dependencies in the data. Additionally, the application of the Multi-Head-Self-Attention layer mixed into other layers allowed the extraction of the most important engine data features reflecting its degradation the most. The question might be raised as to why implement MHSA into the RUL prediction model. Multi-head self-attention is an advanced technique that enhances the standard self-attention mechanism by running it multiple times in parallel. Each "head" learns different patterns or focuses on various aspects

of the input. It allows the model to address different parts of the sequence simultaneously, capturing both short-term and temporal dependencies.

The improved results confirm that physics-informed features help capture turbofan degradation. By modeling low-cycle fatigue explicitly, we introduce an interpretable indicator of damage that the network alone may struggle to learn from raw sensor data. The strain energy density effectively summarizes the cumulative effect of cyclic stress, while the cycle count directly encodes usage intensity. Incorporating these features reduced prediction variance and notably lowered the late-prediction error rate (as seen in the lower Score).

We justified the focus on low-cycle fatigue since engine disks and blades experience few high-amplitude cycles (thrust changes) that dominate crack growth. High-cycle fatigue from vibrations (which produce many small-amplitude cycles) plays a smaller role in RUL for major components. Therefore, our LCF-based approach aligns with maintenance practices that track throttle cycles as life counters. The identified physics variables ($\Delta\sigma$, cycle frequency) are clearly defined: $\Delta\sigma$ is linked to throttle-induced stress range, and the cycle frequency is measured by counting N1 speed reversals. Each of these enters our fatigue model and thus into the hybrid network as additional inputs. Future work could refine the material parameters by component-specific calibration.

Model validation was carefully controlled. The reported accuracy metrics come from test data, not used in training. On the fleet data, we used cross-validation across engines; on the NASA data, we used its official test split. In both cases, overfitting was mitigated by early stopping and regularization. We note that the hybrid model's advantage persists across all folds and both data domains.

Out of the results achieved in the presented case study, it might be concluded that data analytics, through the use of sophisticated machine learning and deep learning models, feature selection, and data fusion techniques, significantly enhances the accuracy and reliability of RUL predictions for engines. These advancements not only improve maintenance scheduling and safety but also reduce operational costs by preventing unexpected failures.

7. CONCLUSION

This paper has presented a physics-aware hybrid framework for turbofan RUL prediction. The work contributes (a) a fatigue-cycle analytics module that computes cumulative strain energy density and LCF cycle counts from throttle data, (b) a multi-architecture LSTM-TCN-MHSA network that optimally fuses sensor and physics features, (c) comprehensive validation on both a 12-year real-world fleet dataset and the NASA C-MAPSS benchmark, and (d) extensive ablation and benchmarking experiments that demonstrate a 53 % RMSE reduction from explicit fatigue modelling and state-of-the-art performance. Future work will extend the framework to component-specific calibration and online deployment.

REFERENCES

Chen, Z., Zhang, Y., & Zhu, X. (2020). Machine remaining useful life prediction via an Attention-Based Deep Learning approach. IEEE Transactions on Industrial Electronics, 68(3), 2521–2531. https://doi.org/10.1109/tie.2020.2972443

Cheng, Y., Wang, J., & Li, H. (2020). Remaining useful life prognosis based on ensemble Long Short-Term memory neural network. IEEE Transactions on Instrumentation and Measurement, 70, 1–12. https://doi.org/10.1109/tim.2020.3031113

Deng, K., Liu, X., & Zhang, Y. (2020). A remaining useful life prediction method with long-short term feature processing for aircraft engines. Applied Soft Computing, 93, 106344. https://doi.org/10.1016/j.asoc.2020.106344

Ibrahima, B., & Meriem, H. (2023). Towards Hybrid Predictive Maintenance for Aircraft Engine: Embracing an Ontological-Data Approach. 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), 1–6. https://doi.org/10.1109/aiccsa59173.2023.10479328

Jiang, Y., Lyu, Y., Wang, Y., & Pin, W. (2020). Fusion Network Combined with Bidirectional LSTM Network and Multiscale CNN for Remaining Useful Life Estimation. International Conference on Advanced Computational Intelligence (ICACI), 620–627. https://doi.org/10.1109/ICACI49185.2020.9177775

Lan, G., Li, Q., & Cheng, N. (2018). Remaining useful life estimation of Turbofan engine using LSTM neural networks. 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), 1–5. https://doi.org/10.1109/gncc42960.2018.9019107

Li, B., Zhu, J., & Zhao, X. (2025). A hybrid physics informed predictive scheme for predicting low-cycle fatigue life and reliability of aerospace materials under multiaxial loading conditions. Reliability Engineering & System Safety, 257, 110838. https://doi.org/10.1016/j.ress.2025.110838

Li, M., Zhu, L., Luo, M., & Ke, T. (2025). Remaining useful life prediction of airplane engine based on bidirectional MAMBA and causal discovery. Sensors, 25(11), 3429. https://doi.org/10.3390/s25113429

Li, X., Wang, Y., & Liu, Z. (2018). Predicting Remaining Useful Life of Industrial Equipment Based on Multivariable Monitoring Data Analysis. Proceedings 2018 Chinese Automation Congress, CAC 2018, 1861–1866. https://doi.org/10.1109/cac.2018.8623249

Li, X., Zhang, W., & Chen, J. (2021). An integrated deep multiscale feature fusion network for aeroengine remaining useful life prediction with multisensor data.

Knowledge-Based Systems, 235, 107652. https://doi.org/10.1016/j.knosys.2021.107652

Li, Y., Zhao, H., & Sun, L. (2022). Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. Reliability Engineering & System Safety, 229, 108869. https://doi.org/10.1016/j.ress.2022.108869

Liao, X., Chen, S., Wen, P., & Zhao, S. (2023). Remaining useful life with self-attention assisted physics-informed neural network. Advanced Engineering Informatics, 58, 102195. https://doi.org/10.1016/j.aei.2023.102195

Lodygowski, T., & Szrama, S. (2025). Unsupervised Classification and Remaining Useful Life Prediction for Turbofan Engines Using Autoencoders and Gaussian Mixture Models: A Comprehensive Framework for Predictive Maintenance. Applied Sciences, 15(14), 7884. https://doi.org/10.3390/app15147884

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2015.7298965

Liu, L., Song, X., & Zhou, Z. (2022). Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. Reliability Engineering & System Safety, 221, 108330. https://doi.org/10.1016/j.ress.2022.108330

Liu, Y., Chen, H., & Wang, X. (2024). A remaining useful life prediction method of mechanical equipment based on Particle Swarm Optimization-Convolutional neural Network-Bidirectional Long Short-Term memory. Machines, 12(5), 342. https://doi.org/10.3390/machines12050342

Ma, J., Zhang, Y., & Li, H. (2018). Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. Complexity, 2018, 1–13. https://doi.org/10.1155/2018/3813029

Peng, D., Li, X., & Wang, Y. (2020). An SW-ELM based remaining Useful life prognostic approach for aircraft engines. IFAC-PapersOnLine, 53(2), 13601–13606. https://doi.org/10.1016/j.ifacol.2020.12.853

Ren, L., Qin, H., Xie, Z., Li, B., & Xu, K. (2022). Aero-Engine Remaining Useful Life Estimation Based on Multi-Head Networks. IEEE Transactions on Instrumentation & Measurement, 71, 1–10. https://doi.org/10.1109/TIM.2022.3149094

Ren, L.-H., Ye, Z.-F., & Zhao, Y.-P. (2022). Long short-term memory neural network with scoring loss function for aero-engine remaining useful life estimation. Proceedings of the Institution of Mechanical Engineers Part G: Journal of Aerospace Engineering, 237(3), 547–560. https://doi.org/10.1177/09544100221103731

Sharma, N. K., & Bojjagani, S. (2024). Mechanical element's remaining useful life prediction using a hybrid approach of CNN and LSTM. Multimedia Tools and Applications, 83(31), 75927–75953. https://doi.org/10.1007/s11042-024-18546-9

Sharma, R. K. (2024). Framework based on Machine learning approach for prediction of the remaining useful life: A case study of an aviation engine. Journal of Failure Analysis and Prevention, 24(3), 1333–1350. https://doi.org/10.1007/s11668-024-01922-w

Sharma, S., Pandit, A. K., & S, S. (2024). Predicting Aircraft Turbofan Engine Degradation with Recurrent Neural Networks. 2024 IEEE International Conference on Information Technology, Electronics, and Intelligent Communication Systems (ICITEICS), 1–6. https://doi.org/10.1109/iciteics61368.2024.10625502

Szrama, S. (2019). F-16 turbofan engine monitoring system. Silniki Spalinowe/Combustion Engines, 177(2), 23–35. https://doi.org/10.19206/ce-2019-205

Szrama, S. (2024). Turbofan engine health status prediction with neural network pattern recognition and automated feature engineering. Aircraft Engineering and Aerospace Technology, 96(11), 19–26. https://doi.org/10.1108/aeat-04-2024-0111

Szrama, S. (2024b). Turbofan engine health status prediction with artificial neural network. Aviation, 28(4), 225–234. https://doi.org/10.3846/aviation.2024.22554

Szrama, S. (2025). Optimizing aircraft engine longevity: A comparative framework for dynamically adaptive predictive maintenance using autoencoders, LSTMs, and Gaussian processes. Engineering Applications of Artificial Intelligence, 156, 111199. https://doi.org/10.1016/j.engappai.2025.111199

Szrama, S., & Lodygowski, T. (2024). Aircraft Engine Remaining Useful Life Prediction using neural networks and real-life engine operational data. Advances in Engineering Software, 192, 103645. https://doi.org/10.1016/j.advengsoft.2024.103645

Thakkar, U., & Chaoui, H. (2023). Prognostic and health management of an aircraft turbofan engine using machine learning. 2021 IEEE Vehicle Power and Propulsion Conference (VPPC), 1–6. https://doi.org/10.1109/vppc60535.2023.10403231

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: a generative model for raw audio. arXiv. https://doi.org/10.48550/arxiv.1609.03499

Wang, F., Liu, X., & Zhang, Y. (2023). Remaining useful life prediction of aero-engines based on random-coefficient regression model considering random failure threshold. Journal of Systems Engineering and Electronics, 34(2), 530–542. https://doi.org/10.23919/jsee.2023.000042

Wang, L., Chen, H., & Li, J. (2022). Long Short-Term Memory Neural Network with Transfer Learning and Ensemble Learning for Remaining Useful Life Prediction. Sensors, 22(15), 5744. https://doi.org/10.3390/s22155744

Wang, S., Li, X., & Zhang, Y. (2018). A Remaining Useful Life Prediction Model Based on Hybrid Long-Short Sequences for Engines. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 1757–1762. https://doi.org/10.1109/itsc.2018.8569668

Wang, W., Liu, Y., & Chen, Z. (2021). Residual convolution Long Short-Term memory network for machines remaining useful life prediction and uncertainty quantification. Journal of Dynamics Monitoring and Diagnostics, 1(1), 2–8. https://doi.org/10.37965/jdmd.v2i2.43

Wen, L., Gao, L., & Li, X. (2023). A new multi-sensor fusion with hybrid Convolutional Neural Network with Wiener model for remaining useful life estimation. Engineering Applications of Artificial Intelligence, 126, 106934. https://doi.org/10.1016/j.engappai.2023.106934

Xia, T., Liu, Y., & Li, H. (2021). Multiscale similarity ensemble framework for remaining useful life prediction. Measurement, 188, 110565. https://doi.org/10.1016/j.measurement.2021.110565

Zha, W., & Ye, Y. (2024). An aero-engine remaining useful life prediction model based on feature selection and the improved TCN. Franklin Open, 6, 100083. https://doi.org/10.1016/j.fraope.2024.100083

Zhang, H., Li, X., & Wang, Y. (2020). Attention-Based LSTM network for rotatory machine remaining useful life prediction. IEEE Access, 8, 132188–132199. https://doi.org/10.1109/access.2020.3010068

Zhang, J., Jiang, Y., Wu, S., Li, X., & Luo, H. (2022). Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. Reliability Engineering & System Safety, 221, 108297.

Zheng, Y., Wang, L., & Chen, X. (2022). Prediction of remaining useful life using fused deep learning models: a case study of turbofan engines. Journal of Computing and Information Science in Engineering, 22(5). https://doi.org/10.1115/1.4054090

Zhu, Q., Liu, Y., & Wang, H. (2024). Contrastive BILSTM-enabled health representation learning for remaining useful life prediction. Reliability Engineering & System Safety, 249, 110210. https://doi.org/10.1016/j.ress.2024.110210