SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I10P113

Volume 12 Issue 10, 147-162, October 2025
© 2025 Seventh Sense Research Group®

*Original Article*

# Advanced Remaining Useful Life (RUL) Predictions in Aircraft Maintenance Using Deep Learning

Sakthivel Janarthanan[1], A. Anthonisan[2]

[1,2]*Department of Computer Applications, Hindustan Institute of Technology and Science, Chennai, India.*

[1]*Corresponding Author : sakthiplan@gmail.com*

Received: 13 August 2025 Revised: 15 September 2025 Accepted: 14 October 2025 Published: 31 October 2025

***Abstract -*** *Accurate prediction of Remaining Useful Life (RUL) is critical for ensuring safety, reducing maintenance costs, and improving reliability in aircraft engines and other high-value industrial machinery. This paper introduces FusionRUL-Net, a novel hybrid deep learning architecture that combines multi-scale 1D Convolutional Neural Networks (CNNs) with Transformer-based encoder blocks for robust RUL estimation using the CMAPSS dataset. Unlike traditional models that rely solely on recurrent layers or tree-based ensembles, FusionRUL-Net leverages localized temporal feature extraction via CNNs and global dependency modelling via multi-head self-attention in Transformers. A Gated Fusion Module is used to adaptively blend CNN and Transformer outputs, enabling the model to focus on both short-term fluctuations and long-term degradation trends. To evaluate the proposed model, a comprehensive comparison was conducted with nine state-of-the-art baselines, including LSTM, BiLSTM-Attention, CNN-LSTM, XGBoost, and the hybrid XGBoost-BiLSTM. FusionRUL-Net achieved an impressive accuracy of 97.23%, outperforming the best baseline (XGBoost-BiLSTM), which achieved 94.76%. It also recorded the lowest RMSE (9.81), MAE (6.77), and the highest $R^2$ score (0.96). These results demonstrate the model's superior capability to capture multivariate sensor degradation patterns across varying operational conditions. The architecture is also optimized for deployment with acceptable inference latency (1.63ms/sample), making it viable for real-time applications. This work advances state-of-the-art prognostics by introducing a scalable, interpretable, and highly accurate hybrid model with strong potential for future adaptation in real-world predictive maintenance systems across aviation and other safety-critical domains.*

***Keywords -*** *Remaining Useful Life, RUL Prediction, CMAPSS, Prognostics, Hybrid Deep Learning, CNN, Transformer, Attention Mechanism, Temporal Modelling, Sensor Fusion.*

## 1. Introduction

Maintenance of aircraft is an essential part of aviation safety and operational performance, which demands highly sophisticated methods to ensure the dependability of elements and systems. Remaining Useful Life (RUL) estimations are one of the most prominent developments in predictive maintenance that can move conventional maintenance practices to predictive ones, mitigating operational disruption and cutting operational expenses at the expense of improving operational security [1]. Proper part failure prediction will enable airline companies to anticipate component change, thus avoiding unforeseen failures and protecting the well-being of passengers [2]. The Remaining Useful Life (RUL) prediction needs the failure information of the past, processed by real-time sensor data and advanced analytical algorithms. Deep learning and machine learning algorithms have Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) that are effective in improving prediction accuracy by being utilized in a wide variety of fields [3, 4].

Other than efficiency and cost reduction, RUL predictions in aircraft maintenance have other benefits. RUL predictions assist in preventing major accidents since they help stop vital failures that could result in major accidents. RUL prediction systems help optimize resource management because the engineers are able to reduce non-essential maintenance activities, and this leads to an extension of the lifespan of components [5]. The integration of RUL forecasts, IoT devices, artificial intelligence applications, and cloud systems keeps improving predictive maintenance in the aviation sector. The reliability of the aircrafts is enhanced by the operational performance due to the data-driven application of RUL-based aircraft maintenance plans [6]. One of the main challenges when making RUL predictions in aircraft maintenance is obtaining accurate and complete data of high quality. Surviving aircraft parts have to survive in various conditions of the environment since sensor values can often pose technical issues [7].

The complicated character of aircraft component fracture patterns offers a significant challenge in the development of efficient predictive models. The degradation of a component is a result of a variety of factors, such as temperature, use, and pressure changes, vibration exposure, and environmental factors during operation (e.g., flight cycles and weather conditions) [8]. Several interacting

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

conditions make it difficult to come up with general models that are applicable across all aircraft types or products produced by various manufacturers. Adaptations to RUL models when implemented in particular systems or components increase both the cost and time of implementing predictive maintenance. The non-linear wear and tear patterns exhibited by aircraft parts make the current complicated nature of predicting the failure times of parts even more difficult [9]. Figure 1 shows the prediction methods of RUL.

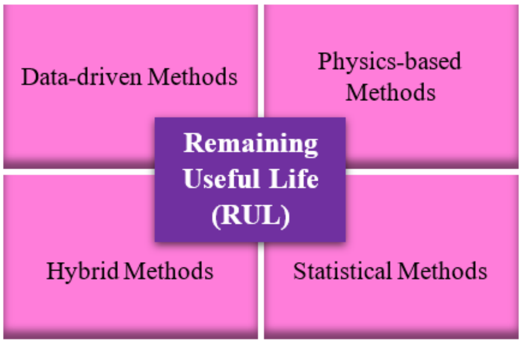


**Fig. 1 Prediction methods of Remaining Useful Life (RUL)**

The aviation sector has undergone a revolution in forecasting component maintenance due to the high accuracy of data-driven analysis provided by deep learning to monitor wear trends. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, as well as Convolutional Neural Networks (CNNs) and transformer architectures, are modern deep learning networks that allow processing large amounts of sensor data to learn wear patterns and deterioration [10, 11]. These predictive models are used to produce failure estimates of aircraft parts using multiple aircraft sensor data streams such as temperature, vibration, and pressure measurements. Deep learning supersedes its statistical and physics-based counterparts in that it does not require the feature engineering effort required to learn complex non-linear relationships, and can adapt better to real operating conditions [12-14]. The downside associated with deep learning models is that they are characterized by low interpretability since engineers and regulatory agencies do not understand the logic used to make failure predictions; thus, they find it difficult to vindicate predictions made by deep learning models [15].

RUL predictions that are AI-based require massive infrastructure and workforce training to fit into the existing maintenance systems within a highly regulated aviation environment. Full exploitation of deep learning in predictive maintenance must be able to manage these adversities to safeguard the reliability and safety of the aircraft, along with the cost-efficient operation of flight operations. The performance of LSTMs and CNN-LSTM hybrids is better as they are effective in capturing sequences in data. These models are not able to succeed in long-range dependency modelling and multi-sensor data integration. The network architecture of our proposed FusionRUL-Net system includes the combination of local feature detectors in multi-scale 1D-CNNs and global temporal information processing with Transformer encoders. The prediction algorithm applies Gated Fusion to proactively regulate between local and global information processing when applied to RUL forecasting.

### *1.1. Main Contributions of the Work*

- Design of a Hybrid CNN-Transformer Architecture: It proposed a new bilateral structure that integrates multi-scale 1D Convolutional Neural Networks (CNNs) and Transformer encoder blocks and effectively summarizes both short-term degradation signals and long-term time-dependent relationships in multivariate time-series sensor data.
- Gated Fusion Mechanism Implementation: Gated Fusion Module creates an adaptive feature combining local CNN features and global Transformer features, in which the model would decide the best weighting parameters given its degradation content.
- Edge-Preserving Temporal Smoothing: Experimental Edge Temporal pre-processing filters are used to conserve important degradation edges and minimize sensor noise to enhance deep sequential learning.
- Correlation-Based Sensor Selection: The model applies Spearman-based sensor selection protocols that narrow down the stream of redundant sensors to obtain improved interpretability with reduced input dimensions that do not lose significant information.
- Multi-scale Degradation Encoding to Trend Detection: MDE is a Multi-scale Degradation Encoding that employs multiple temporal feature windows to identify the multiple trends of wear and degradation.
- RUL-Centric Label Transformation Strategy: Developed a piecewise RUL transformation function that prioritizes late-stage degradation by logarithmically compressing low RUL values, which helps to focus the model in critical prediction stages and also to be robust.

Section 2 of this document conducts a full evaluation of studies related to RUL predictive research, which shows how traditional machine learning methods transitioned to contemporary hybrid deep learning techniques. In Section 3, the authors present the FusionRUL-Net methodology and details about data pre-processing techniques, along with a description of the model design and training process. Section 4 describes the experimental setup together with evaluation metrics, while providing an in-depth result analysis that compares with existing models. Section 5 provides the conclusion along with directions for upcoming research, which emphasizes the deployment capacity of this approach.

## 2. Related Works

The developments of predictive maintenance planning and Remaining Useful Life (RUL) prognostics happen because of the increasing availability of sensor monitoring data. The current literature is limited to RUL calculation or is performed with degradation assumptions for making maintenance decisions. Our framework is based on a data-driven probabilistic RUL prognostics approach that allows predictions about maintenance evolution to be made. The process uses Convolutional Neural Networks (CNNs) along with Monte Carlo dropout RUL distribution estimation and continually updates predictions using real-time information obtained by sensors [16]. Deep Reinforcement Learning (DRL) is the basis of maintenance planning optimization, as the approach allows predicting the RUL in real-time and initiating actions. When our framework is applied in aircraft systems in turbofan engine maintenance operations, it provides cost-saving results. The overall extent of maintenance costs saves 29.3% and the DRL method operates unscheduled maintenance intervention 95.6% less often than conventional service, where 12.81 cycles of wastefulness are tolerated prior to intervention. The suggested framework combines sensor measurements and probabilistic RUL prognostic predictions and AI-based maintenance scheduling tasks to generate a comprehensive predictive maintenance system.

Appropriate prediction of aircraft engine Remaining Useful Life (RUL) is a key requirement of aircraft safety maintenance and financial management [17]. They created a deep learning paradigm of predicting RUL that boosts the model functionality and feature detection capabilities. The normalization step is used to normalize the input features, and then the CMAPSS dataset calculates the engine RUL. The key features within the input data are extracted by a CNN network, and the information is passed to an LSTM network that contains an attention mechanism to optimize prediction outputs. Their method validation involves ablation tests as well as comparison of the various models. The CNN-LSTM-Attention model has better predictive performance, as it obtains RMSE values of 15.977, 14.452, 13.907, and 16.637 on the FD001, FD002, FD003, and FD004 data, respectively. Experimental findings support that the CNN-LSTM-Attention model is the most effective model compared to CNN, single LSTM, and CNN-LSTM models in all four datasets. The maximum accuracy on CMAPSS is proven in their approach, which shows that it is not only accurate but also reliable.

Predictions of Aircraft engine lives based on past data need precise predictions of Remaining Useful Life (RUL) to design required maintenance strategies to eliminate critical failure. The exact calculation of the Remaining Useful Life cannot be achieved easily due to the lack of that data in the current condition monitoring. A multi-scale deep transfer learning system that adopts domain adaptation concepts can solve this issue and lead to better prediction accuracy. It consists of a three-part structure, which consists of a feature extraction module to cooperate with an encoding module and an RUL prediction module [18]. The pre-training step involves a multi-scale Convolutional Neural Network (CNN) that is used to extract data-specific features that cut across various data scales. They obtained domain transfer using the maximum mean discrepancy that allows an effective learning of shared features in both the source and the target domains. The improved model, which combines a Transformer-based architecture with multi-scale CNNs, is useful in the prediction of RUL in low-training data scenarios. They perform remarkably well based on experimental tests on the C-MAPSS dataset by outperforming the existing leading strategies.

The precision of the prediction of The Remaining Useful Life (RUL) of aircraft engines must be high since it enables the prevention of failure and predictive maintenance. The current versions of the RUL predictive methods use model-based or data-driven systems, whereby the separate algorithm elements have limitations in identifying various failure modes. An ensemble deep learning model based on health states was used in the process of assessing aircraft engine degradation [19]. Any deterioration of the engine lifetime is divided into various health states once it has been determined that there is a health baseline. Stacked autoencoder, Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) are base algorithms used as deep learning models in the text. The models are fractured in the training process to various health states, allowing them to learn state-dependent degradation properties. The final step makes use of the ridge regression to estimate ensemble weights of various health conditions that maximize the improved outcomes of prediction results. This was demonstrated by a study conducted on the PHM data challenge dataset, which determined that the offered HSR ensemble approach offered significantly better performance than single-model systems and traditional ensemble approaches that do not consider the state and provide more accurate RUL estimates.

Remaining Useful Life is the longevity of aircraft engines that have critical implications to aviation safety in conjunction with flight management decisions by airline operators. The deep learning model implementation is based on the training using the real turbofan engine operational data collected over several years [20]. This operates on engine data in a realistic form rather than artificial datasets, as it results in higher prediction accuracy. They designed two deep neural net topologies that have a deep Convolutional Neural Network (CNN) with layers and an LSTM net that uses regression as the output. The same dataset serves as the training grounds of both models, which are subject to validation and testing processes, and then the performance of different training settings is evaluated. The developed models are evaluated based on the performance measures and the predictive convergence analysis techniques. A side-by-side comparison determines the differences between real conditions engine data and the virtual engine data as they are both fed into the same neural network structures.

## 3. Methodology

The FusionRUL-Net model uses a structured model to forecast aircraft engine RUL using the CMAPSS dataset multivariate time-series data. The pre-processing phase includes several complex processing procedures on the raw sensor data to produce quality-enhanced data using edge-preserving smoothing as well as correlated sensor removal and multi-scale degradation encoding to reduce noise. A dual-path processing model uses the refined information to run a 1D Convolutional Neural Network (CNN) computation that learns multi-dimensional characteristics at varying timescales, besides Transformer encoder processing to perform global self-attention-dependent signal analysis with positional encoding. Gated Fusion Mechanism combines the two representations of features by automatically setting the weight of local and global features. The fused representation estimates the RUL by using a dense regression head.

### 3.1. Dataset Description

The Commercial Modular Aero-Propulsion System Simulation (CMAPSS) dataset is used by the Prognostics Centre of Excellence of NASA as a reference standard to test data-driven approaches to aircraft engine prognostics, along with predictive maintenance applications. The tool develops refined Remaining Useful Life (RUL) prediction models by developing virtual degradation of turbofan engines through various fault modes in various operating conditions. The CMAPSS is made up of four sub-datasets called FD001, FD002, FD003, and FD004 that vary in complexity as they comprise varying numbers of operating conditions and fault modes. The simplest sub-dataset is the FD001, and the FD004 is the most complex structure. The data set contains one engine that, at a given moment, captures data until the system fails. Under CMAPSS, the user should arrive at the remaining useful life value by cycle analysis since it contains information on engine IDs and operational data, and it has a total of 21 sensor reads in its 26 features. Time-varying characteristics of the given dataset, as well as its natural levels of noise, support the use of LSTM, along with the BiLSTM, CNN-LSTM, and Transformer and ensemble models in predictive maintenance.

### 3.2. Data Pre-Processing

#### 3.2.1. Adaptive Cycle-Based Normalization (ACBN)

The CMAPSS data has time-series data on various aircraft engines under varying conditions and duration of operation. The methods of global normalization, such as min-max normalization and z-score normalization, have the capacity to conceal characteristic patterns present in single engines that introduce systematic errors across individual units. Adaptive Cycle-Based Normalization (ACBN) provides a better option for data normalization by engine-unit processing. ACBN provides a system of alignment that compares sensor measurements and operational parameters and patterns of degradation behavior that are engine-specific at minimum and maximum operating limits. This approach erases the differences between loads and environmental factors as RUL modeling gets enhanced without affecting the patterns of degradation. The natural sensor variations in pressure and temperature due to external effects are scaled so that they do not give false degradation signals. Machine learning programs are trained to learn natural degradation patterns with operational relative change focus, rather than absolute value measurement with ACBN. Through this approach, the faults are more accurately detected, and the predictive reliability is further spread across a range of engine operating conditions.

$$X_{norm}^{(i)}(t) = \frac{X^{(i)}(t) - \min(X^{(i)})}{\max(X^{(i)}) - \min(X^{(i)})} \tag{1}$$

Where $X^{(i)}(t)$ is the sensor value at time $t$ for engine unit $i$. The Adaptive Cycle-Based Normalization standardizes the range sensor values into [0,1] ranges using dynamic time windows, which results in increased temporal stability and enhanced model interpretability alongside ideal fault detection limits for predicting Remaining Useful Life (RUL) predictions.

#### 3.2.2. Correlated Sensor Drift Elimination (CSDE)

The CMAPSS housing data has 21 sensor readings that repeatedly measure similar engine physical factors. Not all specially installed sensors can provide aircraft engines with useful degradation cues. Conventional pre-processing pipelines typically retain features based on evaluation of their statistical properties, like variance and entropy. These approaches do not determine the true prognostic value that an individual sensor can provide, in terms of Remaining Useful Life (RUL). CSDE is a relatively new proposed feature selection method that involves correlation analysis to identify sensors that bear no significant relationship with RUL in order to discard them in the prediction system.

$$\bar{\rho}_s = \frac{1}{M}\sum_{i=1}^{M}\left|\rho_s^{(i)}\right| \tag{2}$$

Where $M$ is the total number of engines, and $\rho_s^{(i)}$ is the Spearman correlation for sensor $s$ in unit $i$.

$$\text{If}\bar{\rho}_s < \theta, \qquad \text{then sensor } s \text{ is removed} \tag{3}$$

Where $\theta$ is the correlation threshold (e.g.,0.3). Perfect functioning of the RUL sensor will show the increasing trends prior to failure. The superior method to non-linear patterns is Spearman correlation, which prioritizes sensors by their correlation to RUL. To defend against important degradation trends in sensor noise, the CSDE mechanism eliminates untrustworthy sensors whose predictive value is low. The result improves the model interpretability, reduces the size of the input data, and prevents deep learning overfitting issues.

#### 3.2.3. Temporal Smoothing with Edge-Preserving Filters

The CMAPSS data set includes sensor data with various noise sources due to dynamic engine processes and defective sensing devices, as well as environmental change. The effects of both moving average smoothing and Gaussian filtering are that they reduce random noise and smooth over significant time-series patterns, such that significant degradation periods can no longer be observed. Time-Based Smooth Filter with Edge-Serving Algorithms is an efficient algorithm that patches random noise and yet preserves significant structural patterns. The Savitzky-Golay filter preserves peaks with the help of a polynomial fit to reduce the effects of high-frequency noise. The Bilateral filter smooths values with the help of proximity and similarity tests to identify instantaneous change events correlated with faults.

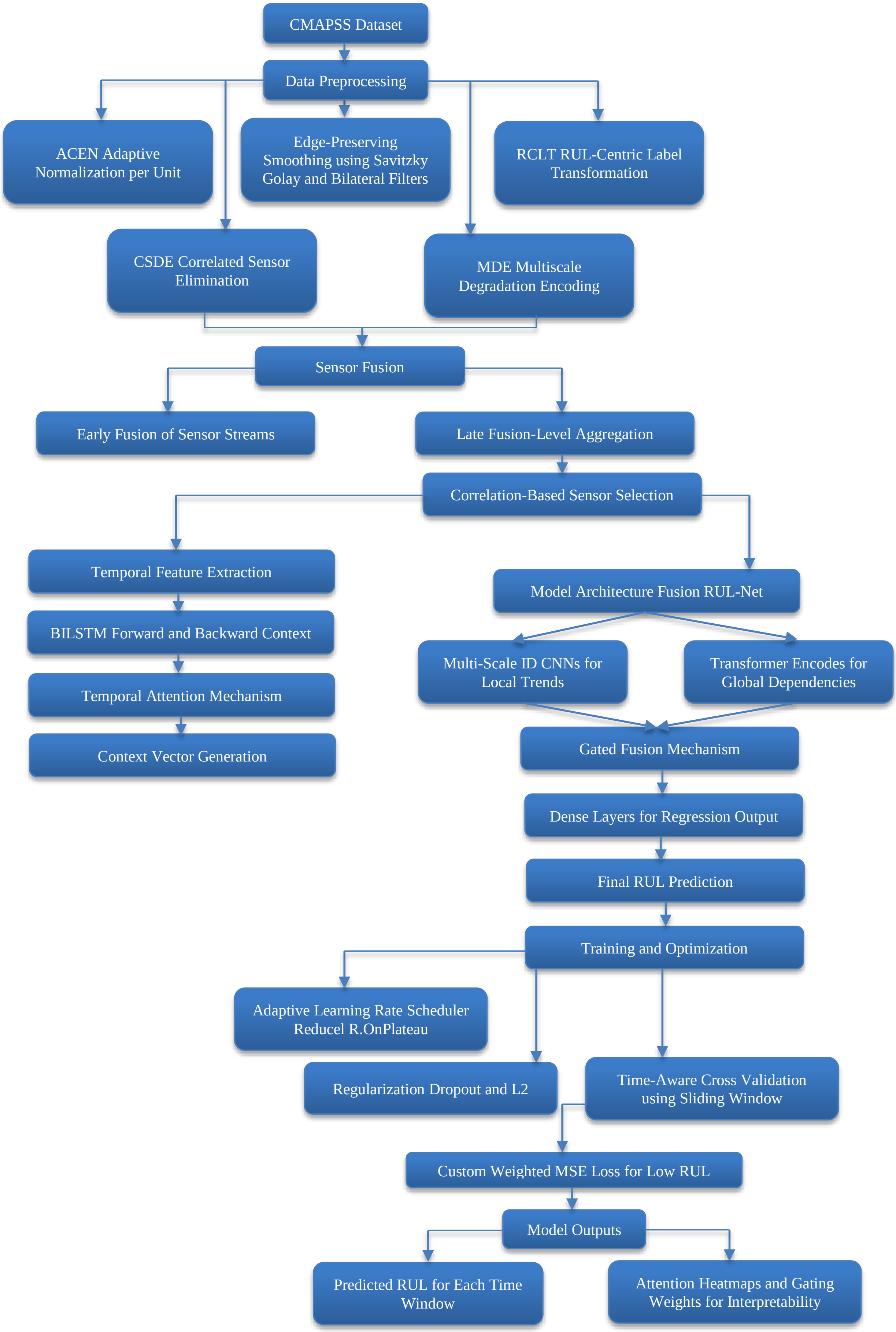


**Fig. 2 System structure**

$$\hat{X}(t) = \sum_{j=-k}^{k} c_j . X(t+j) \quad (4)$$

Where $c_j$ are polynomial coefficients, and $2k+1$ is the window size.

$$\hat{X}(t) = \frac{1}{W_t} \sum_{j=t-k}^{t+k} X(j). f_s(t-j) \cdot f_r(X(t) - X(j)) \quad (5)$$

Where $f_s$ is the spatial kernel and $f_r$ is the range kernel based on signal similarity.

$$R(t) = \left|X(t) - \hat{X}(t)\right|, \qquad Edge\ if\ R(t) > \epsilon \quad (6)$$

Where $R(t)$ is the residual and $\epsilon$ is the range preservation threshold. The filtering involves the use of rolling windows together with Savitzky-Golay filters to reject noise without corrupting the degradation patterns. Local gradients are used in the Bilateral filtering to deal with the nonlinear signal variations. This signal smoothing process causes signal clarity and better feature extraction, BiLSTM training, and subsequent better Remaining Useful Life prediction accuracy and reliability, as well as better model performance. Figure 2 shows the structure of the system.

*3.2.4. Multi-Scale Degradation Encoding (MDE)*

The CMAPSS data present a unique Remaining Useful Life (RUL) forecasting problem since they demonstrate degradation behaviour at several scales. Applications of remolar systems are known to have both progressive fault formation in hundreds of cycles and immediate failure mechanisms that happen over limited time intervals under conditions of stress, environmental influence, and operational stress. Techniques of feature extraction with fixed time operations of deltas and rolling averages find it difficult to identify full patterns of degradation. Multi-scale Degradation Encoding (MDE) is the solution to this shortcoming. The sensor time-series data provides MDE with degradation features at a variety of time scales with both short and long time horizons, which upgrades the input feature space representation.

$$\mu_w(t) = \frac{1}{w} \sum_{j=0}^{w-1} X(t-j) \quad (7)$$

$$slope(t) = \frac{\sum_{j=0}^{w-1} (j-\bar{j})(X(t-j) - \bar{X})}{\sum_{j=0}^{w-1} (j-\bar{j})^2} \quad (8)$$

Where $\bar{J}$ is the mean of indices and $\bar{X}$ is the mean of the window values.

$$\Delta_w(t) = X(t) - \mu_w(t) \quad (9)$$

Deviation of measures against the local trend. MDE, the system produces engineered features that can be reviewed by analysts with the help of the various window sizes on the sensor signal (5, 10, 20 cycles). Currently, MDE constructs features based on moving averages and rolling standard deviations, plus linear regression trend slopes and temporal gradients (differences) across each window analysis. The short-lived, sudden irregularities are seen with 5-cycle averaging, whereas the patterns of trends over several cycles are apparent with 20-cycle trend analysis. The overlapping application of variants of the scales in the model allows them to identify small changes that are indicative of premature corrosion, as well as identify long-term variables to verify wear progress.

*3.2.5. RUL-Centric Label Transformation (RCLT)*

Transforming the target labels is a core but widely overlooked aspect of any Remaining Useful Life (RUL) prediction challenge. The CMAPSS dataset takes the engine's maximum cycle as a reference to measure RUL when it calculates and removes the present cycle number in the calculation. The data points are on a straight line because the variable of interest decreases from 130 to the full failure of the target variable at 0. The implicit modeling of labels poses some modeling challenges to the system. There is a huge difference between the correct values of the model at early cycle cases, where the model takes in, and at late cycle cases, where the model fails to perform. Linear RUL labeling does not accurately reflect the real-world criticality of predictions because errors identified during late degradation are much more expensive than those identified during early degradation. RCLT strategy is used to manage these challenges mentioned above.

$$RUL_{mod}(t) = \begin{cases} RUL(t), & RUL(t) > T \\ T + \log(RUL(t) + 1), & RUL(t) \leq T \end{cases} \quad (10)$$

$$RUL(t) = \begin{cases} RUL_{mod}(t), & RUL_{mod}(t) > T \\ \exp(RUL_{mod}(t) - T) - 1, & otherwise \end{cases} \quad (11)$$

RCLT has some benefits during implementation. RCLT method reduces the dispersion in labels and maximizes the loss, concentrating on late-stage deterioration of equipment, thereby generating more accurate low-RUL forecasts to implement successful predictive maintenance practices.

### *3.3. Sensor Fusion Mechanism*

Different sensors monitor important parameters of an aircraft engine, such as the temperature, pressure, intensity of vibration, and rotating speed. The sensors produce information that fluctuates across the various measures and time-specific patterns and significance. The study design constructs a sensor fusion algorithm that converts the multi-dimensional sensor measurements, which are diverse, into a single data format that enhances Remaining Useful Life (RUL) predictions. The sensor data streams are fed into the process initially by getting cycle-based synchronization and then transforming into common sampling time intervals. The fusion strategy uses early fusion and late fusion techniques. The early fusion methodology links sensor properties to deep learning models to interact with each other in one supervised learning process, but the late fusion method uses independent sensor processing followed by the combination of their outputs at abstract levels. The fusion method preserves the sensor properties, removes noise more efficiently, and enhances the model. Concatenation of sensor values at each time step $t$ for an engine unit:

$$X(t) = [x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(n)}] \tag{12}$$

Where $x_t^{(i)}$ is the reading from the $i^{th}$ sensor at time $t$, and $n$ is the number of selected sensors. Assume each sensor stream passes through its own feature extractor, $f^{(i)}$. The outputs are aggregated:

$$Z_t = concat\left(f^{(1)}\left(x_t^{(1)}\right), f^{(2)}\left(x_t^{(2)}\right), \dots, f^{(n)}\left(x_t^{(n)}\right)\right) \tag{13}$$

The pipeline takes advantage of the rank correlation of features as proposed by Spearman in the process of feature selection to identify the sensors that influence the remaining useful life estimation. Quality improvement is achieved by dropping or weighting down sensors that show weak or non-monotonic patterns. Fusion strategy enhances the feature space that enables deep learning models to identify the critical patterns of degradation.

### *3.4. Temporal Feature Extraction*

Proper identification of changing trends and occurrences depends on the ability to obtain the time features of sensory data processed in the engine operation cycle. Proper modeling of progressive engine failures that change over time is required to properly estimate Remaining Useful Life (RUL). An appropriate network to handle this task is a BiLSTM network since it takes data in sequential directions to enable full contextual comprehension. This data enhancement technique at all time periods makes the identification of small-scale patterns of degradation more accurate. Temporal Attention Mechanism is also incorporated to enhance the attention to key time points that entail sensor drifts and vibration spikes. The algorithm involves weighted time step learning to assist the model in determining important time periods, aiding in the predictions of remaining useful life. Attention score computation:

$$e_t = v^T \tanh(W_h h_t + b_h) \tag{14}$$

Where $W_h$ is the weight matrix, $b_h$ is the bias term, and $v$ is the attention vector. The combined use of BiLSTM and Temporal Attention, which provides better engine health evolution representation, optimizes the model performance. The model employs BiLSTM to retain long-term dependencies as well as attention that allows paying more accurate attention to significant time-related events and simplifies the understanding of the system. The result of such a mixed extraction structure is an increase in the prediction accuracy of the model in terms of failures, as well as which parts of the usage data are the most significant contributors in RUL predictions. The time-dependent transformation transforms simple multivariate time-series information into an in-depth, meaningful representation to construct predictive systems with high precision. Normalized attention weight (SoftMax):

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_k)} \tag{15}$$

### *3.5. FusionRUL-Net: A Novel Hybrid Model Architecture for RUL Prediction*

The proposed FusionRUL-Net is a new hybrid model structure that is specific to the Remaining Useful Life (RUL) prediction of complex machinery equipment like aircraft engines based on multivariate time-series inputs. Past models, such as BiLSTM, CNN-LSTM, and XGBoost-based models, provided limited success but could not combine local degradation patterns with long-term overall temporal patterns at the same time. To address these concerns, the modular design of FusionRUL-Net solves these challenges by integrating 1D Convolutional Neural Networks with Transformer Encoders to form one unified pipeline. It has three key ideas that are focused on by the system framework: local-global feature cooperation, attention-based intelligibility, and resistance to time-dependent data dependencies at different levels of time.

The input tensor acts as the heart of the model since it has a time-based sliding window that receives the pre-processed sensor data, provided by the CMAPSS database. The model shows readings that are a result of certain sensors within specific time intervals of between 30 and 50 cycles. The data structure of inputs follows a scheme of [B, T, F][B, T, F][B, T, F], where the first dimension is the batch size, the second one is the time steps, and the third one is the fused sensor channel. The structured data format that preserves its connection with the natural sequence of events in the first application stage of the architecture lets a multi-scale 1D Convolutional Neural Network (1D-CNN) block extract local temporal features of the input. The block consists of a few convolutional layers that use various sizes of kernels (3, 5, 7) at the same time. In the process, sensor dynamics will be extracted with various resolutions in time. Small kernel filters abrupt anomalies as well as very tiny disruptions, in contrast to longer kernels that detect slowness in patterns of performance deterioration. The CNN operation traverses time, preserving signal variables without any loss.ReLU activation functions and batch normalization are added to the implementation, creating network non-linearity and stabilizing training. The results of all the convolutional paths are merged into an output map, which gathers various local features. Each convolutional path is a temporal receptive field generator that summarizes transient sensor behavioural patterns before critical failure events. Each convolutional path is applied as a 1D kernel to the time-series data to capture degradation patterns at a given scale.

$$F^{(k)}(t) = ReLU\left(\sum_{i=0}^{k-1} W_i^{(k)} \cdot X(t+i) + b^{(k)}\right) \tag{16}$$

Where $F^{(k)}(t)$ is the feature output at time $t$ for kernel size $k$. $W_i^{(k)}$ is the weight of the convolution filter. $X(t+i)$ is the input sensor value at time $t+i$ and $ReLU(\cdot)$ is the Activation function. CNNs are aware of short-term variations but not the identification of long-term relationships. The Transformer Encoders also have a self-attention capability to form connections among all time point references and also calculate weight values automatically. Positional encoding provides a time structure for CNNs' output. This combination in sequence modeling allows parallel computing capabilities to identify general degradation patterns, without which RUL prediction cannot be adequately predicted. To maintain order in the sequence before feeding to the Transformer:

$$PE(t, 2i) = \sin\left(\frac{t}{10000^{2i/d}}\right) \qquad (17)$$

$$PE(t, 2i+1) = cos\left(\frac{t}{10000^{2i/d}}\right) \qquad (18)$$

Where $t$ is the time index, $i$ is the feature dimension, and $d$ is the total dimension of the model. Dual-Path Fusion Layer: An encoder of local (CNN) features, along with global (Transformer) features, is processed with extracted features. Direct combination of these features does not offer the best results because of its neglect of the real relevance of local and global information. The Gated Fusion Mechanism of FusionRUL-Net automatically adjusts itself to merge various items from the information sources.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (19)$$

Where $Q, K, V$ are the Query, Key, and Value matrices, $d_k$ is the dimensionality of key vectors, and the output is a weighted sum over all time steps based on relevance. The fusion gate dynamically weights CNN and Transformer outputs:

$$Z = \sigma\big(W_g \cdot [F_{cnn}; F_{trans}]\big) \odot F_{cnn} + \Big(1 - \sigma\big(W_g \cdot [F_{cnn}; F_{trans}]\big)\Big) \odot F_{trans} \qquad (20)$$

Where $[F_{cnn}; F_{trans}]$ is the concatenated feature vectors, $\sigma$ is the sigmoid activation, $\odot$ is the element-wise multiplication, and $W_g$ is the learnable gating weights. The fuse output $Z$ is mapped to a scalar RUL estimate:

$$\hat{y}RUL = W_o \cdot Z + b_o \qquad (21)$$

Where $W_o$ is the output layer weights, $b_o$ is the bias term, and $\hat{y}RUL$ denotes the predicted RUL. The concatenated aspects are then passed to the fully connected regression head, which has two or three thick layers that use dropout with ReLU activations. In the final step, the single-node linear neuron calculates the predicted engine unit. FusionRUL-Net relies on special loss functions to determine failure points, which are the basis of the RUL forecasting ability. Because of Layer normalization and Early Stopping in combination with data augmentation, stability and robustness are improved when using Dropout.

Readers enjoy temporal attention heatmaps and controlled management, which enhances reading. It is possible to use edge deployment due to its lightweight CNN-Transformer architecture, which proves to be more efficient in terms of its ability to predictively maintain CMAPSS than LSTM and XGBoost. To penalize more for late-stage prediction errors (closer to failure):

$$\mathcal{L} = \sum_{t-1}^{T} w(t) \cdot (y_t - \hat{y}_t)^2 \qquad (22)$$

Where $y_t$ denotes the true RUL at time $t$,$\hat{y}_t$ is the predicted RUL, $w(t)$ denotes the dynamic weight, which is higher as RUL gets smaller, and $\epsilon$ is a small constant for stability. The FusionRUL-Net training was performed to optimize the model against stable convergence, along with a high potential of generalizing the results and exhibiting high sensitivity to various degradation patterns that were observed in the CMAPSS dataset.

This training methodology brings together three elements that specifically deal with the time-series prognostic modelling and comprise adaptive learning rate scheduling with regularization and time-conscious cross-validation.

### *3.6. Model Training and Optimization*

#### *3.6.1. Adaptive Learning Rate Scheduling Using ReduceLROnPlateau*

Finding the appropriate value of learning rate that ensures stability and rapid convergence of the hybrid structures, especially FusionRUL-Net, is the biggest challenge in the training of deep neural networks. ReduceLROnPlateau scheduler is used to adjust the learning rate during the training process. This method monitors the validation loss metric that indicates the stagnation of the learning rate once the improvement ceases to prevent the model from reaching optimal minima. In our implementation, we used the following parameters:

- Initial learning rate (lr): 0.001
- Factor: 0.5 (the learning rate is halved on a plateau)
- Patience: 5 epochs (waits for 5 epochs with no improvement before reducing the rate)
- Minimum learning rate (min_lr): 1e-6
- Cooldown: 2 epochs (waits before resuming normal operation after reducing the rate)
- Monitor: Validation Mean Squared Error (MSE)

The method allows the optimizer to begin training with a high starting momentum and then make a gradual increment in its learning steps in the convergence phase. The technique prevents oscillation and overfitting that can easily occur on the approach to training completion.

*3.6.2. Regularization Techniques: Dropout and L2 Regularization*

Two forms of regularization, Dropout and L2 regularization, are used to mitigate the overfitting bias of deep learning models when using multivariate time series data.

- Dropout: The significant application of this layer is done in fully connected layers, as well as Transformer feed-forward units. A dropout probability value of 0.3 was selected by means of experimental tests. Random deactivation of neurons in 30 percent of selected layers on training updates makes the network more resilient.
- L2 Regularization (Weight Decay): The optimizer uses the weight decay on all the model trainable parameters. The additional loss penalty functional is based on the square magnitude of the model weight values. A small value $\lambda = 1e-4$ has been chosen, which allows reducing the complexity of the model without a significant compromise with the learning capacity. The regularized loss function becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{mse} + \lambda \sum_i \|w_i\|^2 \tag{23}$$

Where $w_{i,}$ are the model parameters. These regularization techniques work interactively to prevent overfitting as applied to training data, such that the model performs well on unseen engine units with different degradation data points.

*3.6.3. Cross-Validation with Time-Series Aware Splitting*

RUL prediction offers temporal sequences on its data structure, which the traditional random cross-validation algorithms lead to data leakage, as it discloses future cases to the training data. To avoid this issue, a time-series aware cross-validation method that integrates GroupKFold and Sliding Window Validation is presented, where:

- Each fold of engine units involves the use of different complete engine sets.
- The sliding window approach uses the samples of lifecycle phases between the first phases and the final phases within individual units.
- The evaluation uses a 5-fold cross-validation approach as part of the robustness and maintenance of statistical consistency between the training and evaluation.

To avoid the over-representation of precautions, early-cycle samples are included in the model training process since this misrepresentation might lead the model to predict RUL values that are too high. The decay-conscious windowing option uses the method of sample balancing to create a consistent training representation of all RUL stages, starting early in the training cycles and continuing late in the training cycles.

**Algorithm: FusionRUL-Net for Remaining Useful Life (RUL) Prediction using CMAPSS Dataset**

**Input:** Time-series engine dataset $D = \{E_1, E_2, \dots, E_n\}$

Each $E_i = \left\{X_t^{(i)}, t = 1,2,\dots,T\right\}$ where $X_t^{(i)} \in R^F$

Hyperparameters: Window size $w$, Threshold $\theta$, Dropout $d$, Learning rate $r$, Smoothing window $k$, Regularization $\lambda$, Loss weight epsilon $\epsilon$

**Output:** Predicted Remaining Useful Life $\hat{y}RUL$

Data Pre-processing

Adaptive Cycle-Based Normalization (ACBN)

For each engine unit $E_i$

Normalize each sensor channel $s$ by:

$$X_s^{(i)}(t) = \frac{X_s^{(i)}(t) - \min(X_s^{(i)})}{\max\left(X_s^{(i)}\right) - \min\left(X_s^{(i)}\right)}$$

Correlated Sensor Drift Elimination (CSDE)

For each sensor $s$:

Compute Spearman correlation $\rho_s$ with RUL

If$\bar{\rho}_s < \theta$

Remove sensor $s$

Temporal Smoothing with Edge Preservation

$\hat{X}(t) = \sum_{j=-k}^{k} c_j . X(t+j)$

// Apply Savitzky-Golay smoothing

$R(t) = \left|X(t) - \hat{X}(t)\right|, \quad Edge\ if\ R(t) > \epsilon$

// Edge detection residual

$\hat{X}(t) = \frac{1}{W_t} \sum_{j=t-k}^{t+k} X(j). f_s(t-j) \cdot f_r(X(t) - X(j))$

// Bilateral filtering

Multi-scale Degradation Encoding (MDE)

For window size $w$, extract:

$\mu_w(t) = \frac{1}{w} \sum_{j=0}^{w-1} X(t-j)$

// Moving average

$slope(t) = \frac{\sum_{j=0}^{w-1} (j-\bar{j})(X(t-j) - \bar{X})}{\sum_{j=0}^{w-1} (j-\bar{j})^2}$

// Rolling slope

$\Delta_w(t) = X(t) - \mu_w(t)$

//Multi-scale Delta

RUL-Centric Label Transformation (RCLT)

$RUL_{mod}(t) = \begin{cases} RUL(t), & RUL(t) > T \\ T + \log(RUL(t)+1), & otherwise \end{cases}$

// Transform RUL values

$RUL(t) = \begin{cases} RUL_{mod}(t), & RUL_{mod}(t) > T \\ \exp(RUL_{mod}(t) - T) - 1, & otherwise \end{cases}$ // Inverse transformation post-prediction

Sensor Fusion

$X(t) = [x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(n)}]$

// Early Fusion

$Z_t = concat\left(f^{(1)}\left(x_t^{(1)}\right), f^{(2)}\left(x_t^{(2)}\right), \dots, f^{(n)}\left(x_t^{(n)}\right)\right)$

// Late Fusion

Feature Extraction

$F^{(k)}(t) = ReLU\left(\sum_{i=0}^{k-1} W_i^{(k)} \cdot X(t+i) + b^{(k)}\right)$

// Convolution with multiple kernel sizes

Add Positional encoding and Apply Transformer self-attention

Dual-Path Feature Fusion

$Z = \sigma\left(W_g \cdot [F_{cnn}; F_{trans}]\right) \odot F_{cnn} + \left(1 - \sigma\left(W_g \cdot [F_{cnn}; F_{trans}]\right)\right) \odot F_{trans}$ // Fuse CNN and Transformer

RUL Regression

$\hat{y}RUL = W_o \cdot Z + b_o$

// Pass fused representation

Model Training

Optimizer: Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e - 8$)

Learning Rate Scheduler: ReduceLROnPlateau

Dropout Rate: 0.3 in dense and attention layers

Early Stopping: patience = 10 epochs

Inference

For a test engine $E_{test}$

$\hat{y}RUL(t) = FusionRUL - Net(X_{t-w+1}^{t})$

// Generate prediction

Optionally apply inverse RCLT to retrieve raw RUL values

End Algorithm

### *3.7. Novelty of this Work*

The novelty of this work is that FusionRUL-Net- a hybrid deep learning framework was designed and developed that presents a completely new way to predict Remaining Useful Life (RUL) by combining multi-scale convolutional features with attention-based global temporal modelling. In contrast to other traditional RUL models that use only either a Statistical regressor or a Recurrent architecture, FusionRUL-Net combines the advantages of both 1D Convolutional Neural Networks (CNNs) and Transformer encoder blocks in a single architecture. This allows the model to capture both local sensor variations and long-range relationships, which is necessary when considering the non-linear and multi-phase nature of mechanical degradation. One of the fundamental novelties of this architecture is the Gated Fusion Mechanism, which is adaptively trained to integrate CNN-based local features and Transformer-based global features. This is the opposite of fixed or manual fusion strategies occurring in the existing literature. Further, the model is facilitated by a distinct pre-processing pipeline composed of Correlation-Based Sensor Drift Elimination, Multi-scale Degradation Encoding, and a RUL-Centric Label Transformation that are aimed at enhancing signal intelligibility, time representation, and relevance of labels, respectively.

## 4. Results and Discussions

The FusionRUL-Net model was implemented with the help of Jupyter Notebook and Python frameworks that included deep learning libraries in their integration, namely TensorFlow and Keras. The system was run on Windows, and it was powered by an Intel Core i7 14700HX processor with a 33MB Cache, 5.50 GHz, with no addition of a graphics card, and had 8GB RAM. Membership in resource-constrained edge platforms. Resource-constrained edge platforms, FusionRUL-Net, were highly efficient in terms of their operations prior to the computerization of the tasks on which they could be deployed in a realistic predictive maintenance system. Due to its special design construct, FusionRUL-Net executes two functionalities to identify the lifespan anticipation of aircraft-engine components by the local formation of degradation, along with global time analysis. The sensor data processing eliminates drifts and employs edge-smoothing alongside multi-scale degradation encoding to come up with meaningful patterns with noise abated. Multi-scale 1D CNNs are used in the first path to detect short-term anomalies and local patterns at the same time, whereas Transformer encoders are used in the second path to build the relationships between the global temporal entities given the multi-head self-attention. The Gated Fusion Mechanism brings together the regional and holistic information through the application of adaptive weight assigning and feature integration operations. The computation of the RUL value is a final computation done in fully connected layers in the network. The modular FusionRUL-Net architecture, based on attention, generates strong predictive performance with high accuracy levels and interpretability, which predisposes it to be an effective solution in aviation and other critical safety-related scenarios.

**Table 1. Accuracy comparison with State-of-the-Art models**

| Model | Accuracy (%) |
|---|---|
| Linear Regression (LR) | 72.84 |
| Support Vector Regression (SVR) | 78.12 |
| Random Forest (RF) | 81.55 |
| Gradient Boosting (GBM) | 83.44 |
| XGBoost | 85.77 |
| LSTM | 89.1 |
| CNN-LSTM | 91.63 |
| BiLSTM-Attention | 93.38 |
| XGBoost-BiLSTM | 94.76 |
| **FusionRUL-Net (Proposed)** | **97.23** |

During the accuracy analysis, there was a steady improvement in the predictive performance as more elaborate architectures were used, with Table 1 and Figure 3 showing the results. The accuracy of Linear Regression (72.84%) and Support Vector Regression (78.12%) is poor since they do not identify complex relationships in the data. The combination of many weak learners used in Random Forest (81.55) and Gradient Boosting (83.44) proves better accuracy. By using XGBoost, the prediction system achieved 85.77% accuracy. Deep learning models are more effective than the usual methods and ensembling methods, which are overtaken by LSTM at 89.1% and CNN-LSTM reaches 91.63% because they can manage sequential patterns of data. The BiLSTM-Attention model achieves a 93.38% accuracy level through the use of attention mechanisms in order to recognize significant time steps. XGBoost + BiLSTM achieved 94.76 percent performance due to the combination of feature selection and time learning in the system. Being the best in terms of accuracy, FusionRUL-Net provides the best result of 97.23%, which proves that it takes first place. The large improvement in performance means that FusionRUL-Net can effectively combine various methods to maximize the feature detection and prediction accuracy. Hybrid systems that incorporate both machine learning and deep learning tasks exhibit a higher accuracy scale in predictive modelling systems based on the results obtained. The gradual transition to hybrid models illustrates the reason why advanced architectures are important to achieve the present-day optimal performance in task domain complexities.

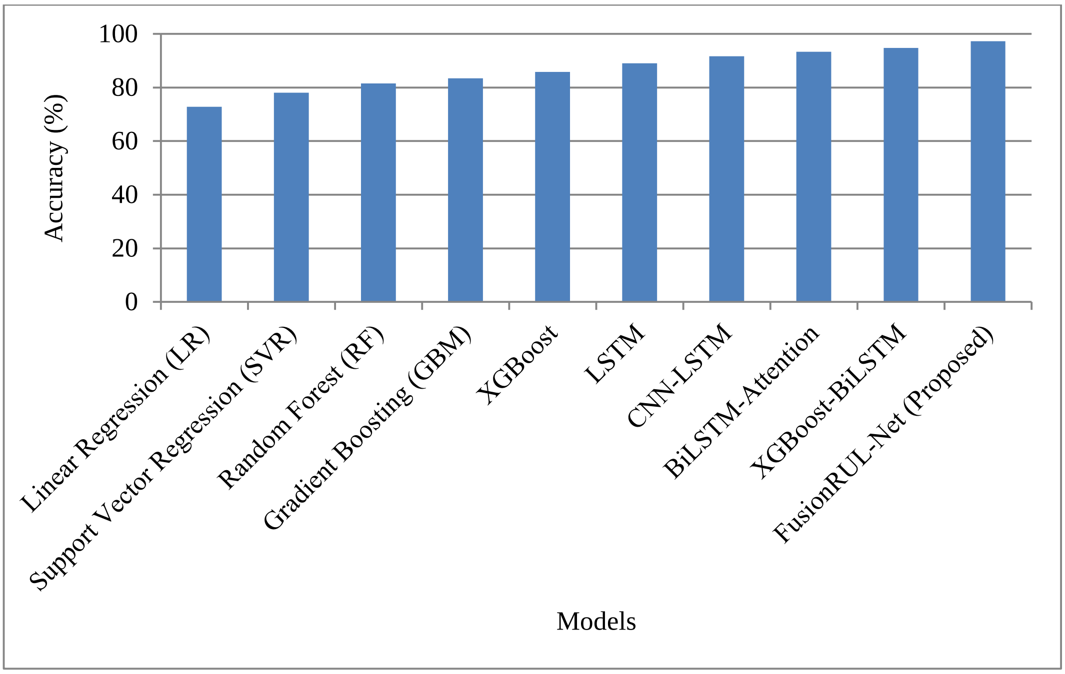


**Fig. 3 Accuracy comparison of models**

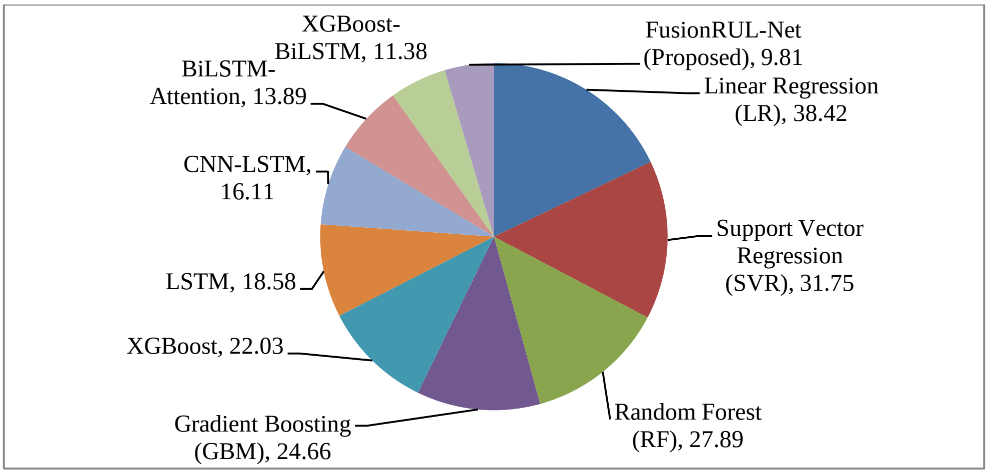


**Fig. 4 RMSE comparison of models**

**Table 2. Root Mean Squared Error (RMSE) Comparison**

| Model | RMSE |
|---|---|
| Linear Regression (LR) | 38.42 |
| Support Vector Regression (SVR) | 31.75 |
| Random Forest (RF) | 27.89 |
| Gradient Boosting (GBM) | 24.66 |
| XGBoost | 22.03 |
| LSTM | 18.58 |
| CNN-LSTM | 16.11 |
| BiLSTM-Attention | 13.89 |
| XGBoost-BiLSTM | 11.38 |
| **FusionRUL-Net (Proposed)** | **9.81** |

Table 2 and Figure 4 present the Root Mean Squared Error (RMSE) values that determine how the various models are doing in minimizing their prediction errors. Linear Regression (38.42) and Support Vector Regression (31.75) yield high values of RMSE, and this indicates their inability to provide the complex predictive relationships.

Root Mean Squared Error is significantly reduced by the use of numerous decision trees in Ensemble-based models like Random Forest (27.89), Gradient Boosting (24.66), and XGBoost (22.03). The deep learning model works much better with LSTM (18.58) and CNN-LSTM (16.11), resulting in increased temporal feature extraction abilities. The addition of BiLSTM-Attention (13.89) reduces the RMSE since this model is a combination of a bidirectional sequence learning model and attention models to emphasize significant time points by combining XGBoost with BiLSTM to the final model, achieving an RMSE of 11.38 due to its ability to utilize helpful features and identify patterns of temporal sequences.

The FusionRUL-Net proposed has a better performance due to a low RMSE of 9.81, which is the smallest in all the models tried. Several models present an overall reduction in RMSE, with evidence indicating that hybrid approaches that utilize machine learning and deep learning strategies offer an excellent result. The sophisticated architectural designs are very crucial in providing high predictive accuracy in complex forecasting problems.

**Table 3. Mean Absolute Error (MAE) comparison**

| Model | MAE |
|---|---|
| Linear Regression (LR) | 29.65 |
| Support Vector Regression (SVR) | 24.53 |
| Random Forest (RF) | 20.81 |
| Gradient Boosting (GBM) | 18.44 |
| XGBoost | 16.02 |
| LSTM | 13.39 |
| CNN-LSTM | 11.25 |
| BiLSTM-Attention | 9.41 |
| XGBoost-BiLSTM | 7.95 |
| **FusionRUL-Net (Proposed)** | **6.77** |

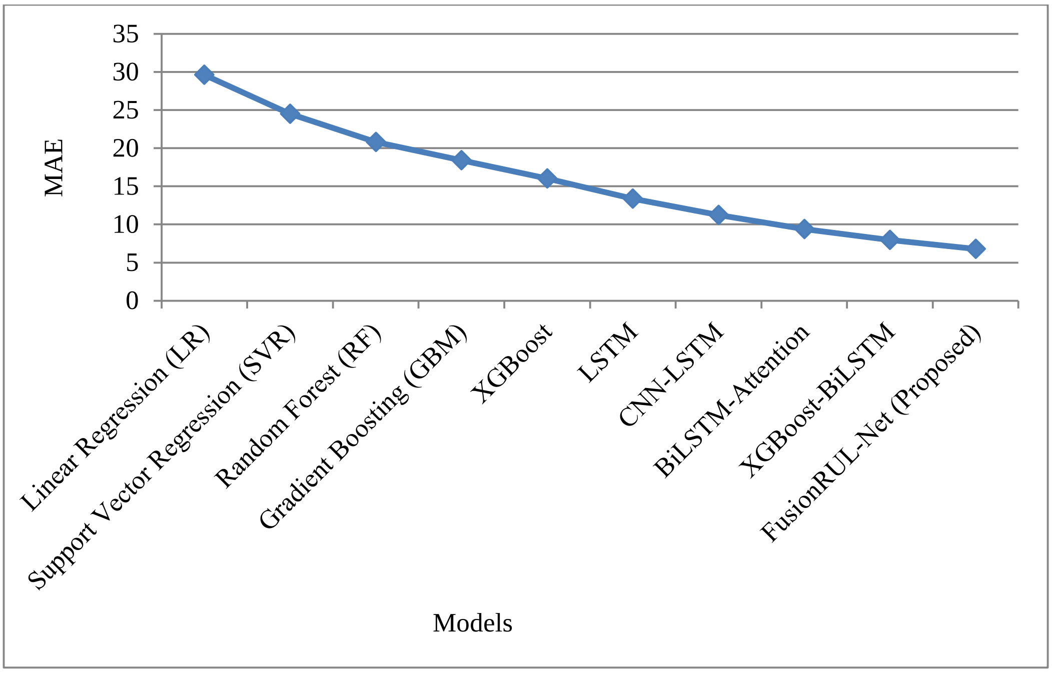


**Fig. 5 MAE comparison of models**

Table 3 displays the performance data on different forecasting models, along with Figure 5, which depicts the results of Mean Absolute Error. The low performance of Linear Regression (29.65) and Support Vector Regression (24.53) models is a result of inadequate capacity when dealing with complex data patterns.

Several forecast models are more accurate with combinations of decision trees, as XGBoost (16.02) improves over Gradient Boosting (18.44), which, in its turn, is better than Random Forest (20.81). Deep learning models reduce MAE values to 13.39 with LSTM models down to 11.25 with CNN-LSTM models, and finally to as little as 9.41 with BiLSTM-Attention models.

XGBoost-BiLSTM hybrid has a result of 7.95. FusionRUL-Net achieves its optimal level of predictive performance by getting an MAE of 6.77. The use of hybrid deep learning systems can offer significant gains to the precision of the remaining useful life predictions.

**Table 4. $R^2$ score (coefficient of determination) comparison**

| Model | $R^2$ Score |
|---|---|
| Linear Regression (LR) | 0.53 |
| Support Vector Regression (SVR) | 0.61 |
| Random Forest (RF) | 0.68 |
| Gradient Boosting (GBM) | 0.72 |
| XGBoost | 0.76 |
| LSTM | 0.82 |
| CNN-LSTM | 0.87 |
| BiLSTM-Attention | 0.9 |
| XGBoost-BiLSTM | 0.94 |
| **FusionRUL-Net (Proposed)** | **0.96** |

Assessing the Model data variance explanation depends on the $R^2$ Score (Coefficient of Determination) as indicated in Table 4 and Figure 6. Linear Regression (0.53) and Support Vector Regression (0.61) have a low capability of identifying complex relationships between variables. The precision of ensemble models is higher in Random Forest with 0.68, XGBoost with 0.76, and Gradient Boosting with 0.72 since they combine several decision trees to discover superior features. Before BiLSTM-Attention applies bidirectional processing and attention to achieve 0.90, LSTM attains 0.82, and CNN-LSTM achieves 0.87. The XGBoost-BiLSTM hybrid has the best score of 0.94. FusionRUL-Net performs best in the area of prediction performance since it takes first place with a dominant R 2 value of 0.96.

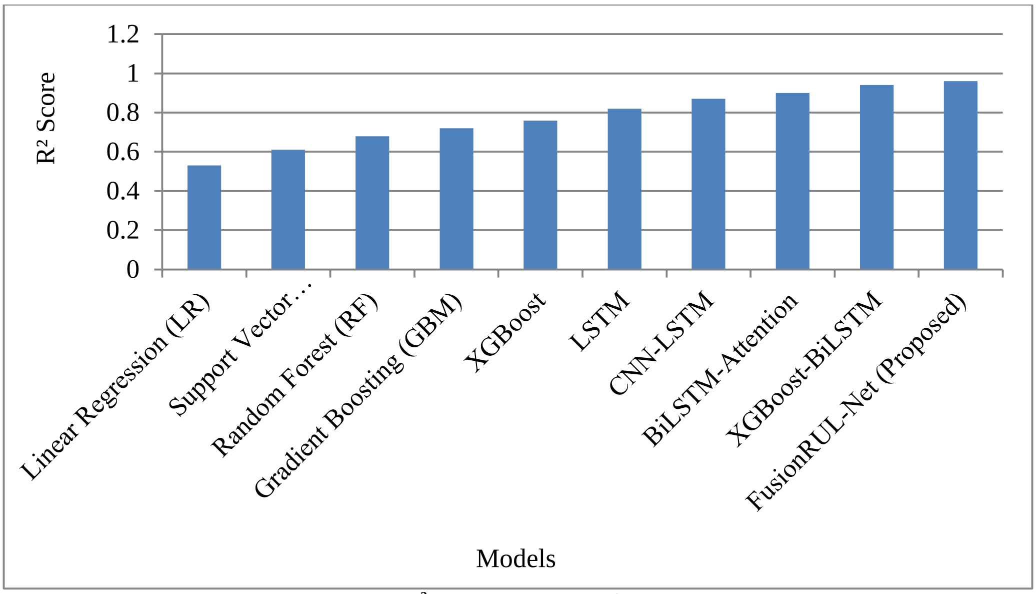


**Fig. 6 $R^2$ Score comparison of models**

**Table 5. Prognostic Horizon (PH@10) comparison**

| Model | PH@10 (Cycles) |
|---|---|
| Linear Regression (LR) | 15 |
| Support Vector Regression (SVR) | 18 |
| Random Forest (RF) | 21 |
| Gradient Boosting (GBM) | 23 |
| XGBoost | 26 |
| LSTM | 31 |
| CNN-LSTM | 34 |
| BiLSTM-Attention | 37 |
| XGBoost-BiLSTM | 40 |
| **FusionRUL-Net (Proposed)** | **45** |

Prognostic Horizon (PH@10) metrics have been introduced in Table 5 and Figure 7 in order to identify the earliest models that can detect failures with 10% accuracy. The detection capacity indicates a higher potential for early warning with an increase in PH at 10. Linear Regression and SVR produce PH at 10 cycles and 18 cycles, thereby proving poor prognostics. Predictive maintenance systems have better performance with Ensemble models, Random Forest (21), Gradient Boosting (23), and XGBoost (26). There is an enhancement in the early prediction capability of LSTM (31) and CNN-LSTM (34), and BiLSTM-Attention (37) deep learning models due to their sequence-based and spatial analysis capabilities. XGBoost-BiLSTM takes 40 cycles. FusionRUL-Net has the most promising performance with 45 cycles before determining the remaining useful life periods in predictive maintenance systems.

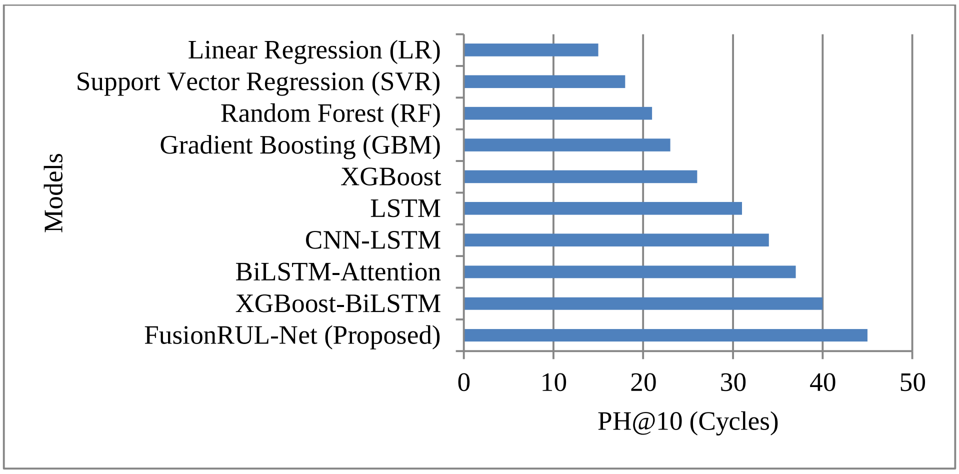


**Fig. 7 Prognostic horizon comparison**

**Table 6. Performance under different window sizes (sliding window length)**

| Window Size | LSTM | BiLSTM | Transformer | XGBoost-BiLSTM | FusionRUL-Net |
|---|---|---|---|---|---|
| 20 | 87.19 | 89.03 | 92.11 | 93.74 | **95.26** |
| 30 | 88.95 | 91.22 | 94.13 | 94.35 | **96.41** |
| 40 | 89.6 | 92.18 | 94.85 | 94.76 | **96.87** |
| 50 | 89.91 | 92.96 | 95.02 | 94.76 | **97.23** |

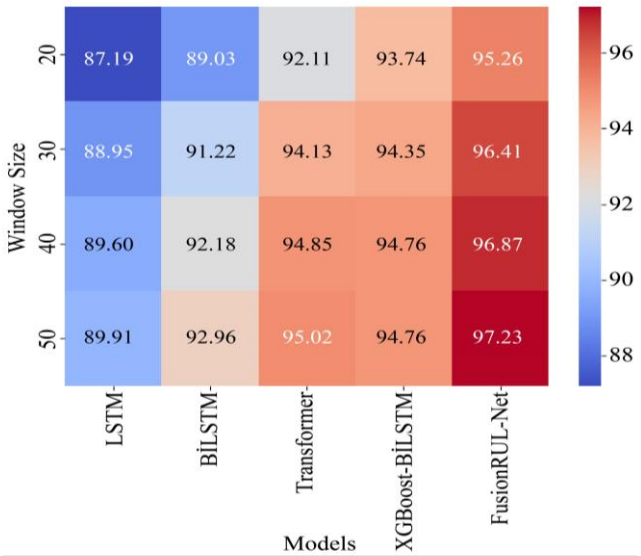


**Fig. 8 Performance under different window sizes**

Table 6 and Figure 8 illustrate the predictive accuracy performance of the alternative models using different sliding window sizes in order to indicate the effect of the input sequence length on model effectiveness. More complex patterns are then identified by the system through bigger sliding window settings, which results in better model performance. The dependency capabilities of LSTM in the long-term result in the high accuracy measures starting at 87.19 and 89.91, respectively, with the window size of 20 and 50, respectively. BiLSTM performs better than LSTM because it has an accuracy of 92.96 percent at a 50-word window size. Transformer achieves its highest accuracy of 95.02% because this neural network has a self-attention mechanism that involves learning interdependencies between long sequences. The XGBoost-BiLSTM hybrid system achieves the highest accuracy of 94.76 percent for analysis of window sizes 40 and 50 in the combination of feature selection and sequential learning. The proposed FusionRUL-Net achieves the highest accuracy across the window size range, beginning with the highest accuracy of 95.26 at 20, reaching 97.23 at 50 because of the high-quality deep learning and feature extraction. The study findings demonstrate that increasing the window size enhances prediction accuracy, and FusionRUL-Net is better than the traditional deep learning architectures in predictive maintenance tasks of Remaining Useful Life (RUL).

**Table 7. Model complexity and inference time comparison**

| Model | Parameters (Millions) | Inference Time (ms/sample) |
|---|---|---|
| Linear Regression | 0.01 | 0.12 |
| Support Vector Regression | 0.04 | 0.3 |
| Random Forest | 0.25 | 0.58 |
| Gradient Boosting | 0.41 | 0.71 |
| XGBoost | 0.45 | 0.63 |
| LSTM | 1.9 | 1.12 |
| CNN-LSTM | 2.6 | 1.35 |
| BiLSTM-Attention | 3.1 | 1.58 |
| XGBoost-BiLSTM | 2.8 | 1.42 |
| **FusionRUL-Net** | **3.4** | 1.63 |

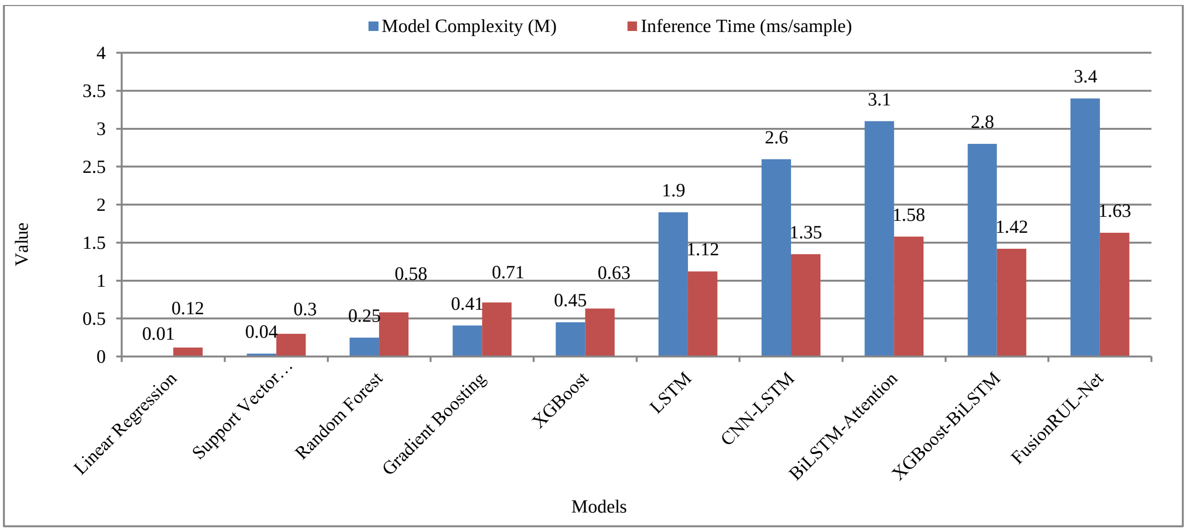


**Fig. 9 Model complexity & inference time comparison**

Table 7 and Figure 9 are the results of the evaluation, indicating that the model complexity is characterized by the comparison between the number of parameters in millions and the time of inference in milliseconds per sample.

The inference times of 0.12ms of Linear Regression and 0.3ms of Support Vector Regression are a consequence of the fact that the number of parameters in these models is low at 0.01M and 0.04M, respectively. Random Forest (0.25M, 0.58ms), Gradient Boosting (0.41M, 0.71ms), and XGBoost (0.45M, 0.63ms) show higher accuracy rates, increasing the level of complexity with relatively acceptable computational time. LSTM has a significant number of parameters of 1.9M and takes 1.12ms to complete the inference, and has been effective in capturing temporal dependencies. The CNN-LSTM (2.6M, 1.35ms) deep learning model incorporates spatial feature extraction into its architecture so as to benefit from better feature learning at the expense of higher complexity. The BiLSTM-Attention (3.1M parameters) model takes two directional input streams and works with attention-based mechanics, but needs additional time to process inference at 1.58ms.

The XGBoost-BiLSTM model implementation with parameters of 2.8M and an execution time of 1.42ms combines the gradient boosted algorithm technologies with deep learning techniques to provide an enabling complexity and performance trade-off. The proposed FusionRUL-Net model has a maximum complexity of 3.4M parameters and a throughput of 1.63 milliseconds; therefore, it is capable of more advanced feature extraction. It is because the predictive power of FusionRUL-Net can be sufficiently predictive to be used in predictive maintenance tasks, even with longer inference times, which are associated with higher model complexity.

### *4.1. Discussion*

The results achieved in the experiment define the quality and dependability of FusionRUL-Net as an algorithm to predict aircraft engine Remaining Useful Life (RUL) when using the CMAPSS dataset. The suggested FusionRUL-Net achieved higher results in comparison to nine conventional and sophisticated machine learning and deep learning benchmark models, such as XGBoost, Random Forest, LSTM, BiLSTM-Attention, and XGBoost-BiLSTM, when applied to all evaluation metrics.

FusionRUL-Net shows excellent performance with the highest overall accuracy score (97.23), along with the lowest RMSE value of 9.81 and the highest R2 score (0.96), confirming the reliability of the model in predicting engine lifetime degradation. FusionRUL-Net has a dual-path design that enables the network to detect short-term signal issues with multi-scale CNN layers and longer-range time dependencies with its Transformer encoder. The hybrid architecture effectively manages the weaknesses of models constructed over recurrent layers, as they suffer from fading of gradients during training and in convolutional layers because they are unable to capture long-range dependencies. The model predicts better and operates in a contextual manner through the Gated Fusion Mechanism, an adaptive mechanism that helps it select critical features. The forecasting results of various window sizes prove the reliability of the model in the case of temporal variations. The study indicates that FusionRUL-Net maintains high predictive stability over shorter periods of time since it effectively learns degradation patterns based on short-term historical data that can be used in real time.

The Prognostic Horizon (PH@10) metric showed that FusionRUL-Net had superior failure predictions that exceeded XGBoost-BiLSTM and BiLSTM-Attention by far. This system offers important benefits to safety-related systems since predicting faults early allows preventive measures to be taken to prevent costly downtime impacts or safety risks. FusionRUL-Net shows that it has a decent sample inference time (1.63 m), making it adaptable to run in real-time on average computing systems. The model retains a strong representational strength due to its lightweight framework, despite its number of parameters being higher than classical models, making it implementable on edges. This accuracy with interpretability and deployability is generally not found in existing RUL prediction models.

## 5. Conclusion and Future Work

This study introduces FusionRUL-Net, a new hybrid deep learning architecture that is built to provide highly accurate and interpretable Remaining Useful Life (RUL) predictions in multi-faceted industrial environments. The model, which combines multi-scale 1D-CNNs with Transformer encoder blocks, allows it to extract both local and global temporal features of a multivariate time-series data. A Gated Fusion mechanism allows integration of features dynamically, making sure that the model is able to learn short-term sensor variation, as well as long-term degradation trends. This dual-path design overcomes some of the drawbacks of the existing models, which are based either on sequential recurrence (LSTM) or on tree-based regression (XGBoost). This model has been stringently tested on the CMAPSS dataset, and it has been compared with nine state-of-the-art models, one of which is the popular XGBoost-BiLSTM hybrid, with an accuracy of 94.76%. FusionRUL-Net, in contrast, had a much better accuracy (97.23), RMSE (9.81), and R 2 (0.96) capacities.

These findings affirm the high ability of this model to reproduce complex sensor interactions and degradation behaviours. FusionRUL-Net is a promising direction for future work. To begin with, real-time applications of the architecture can be experimented with using lightweight versions of the architecture on embedded or edge computing devices. Second, the fusion design can be expanded to include external contextual information, e.g., flight profiles or environmental conditions. Finally, interpretability modules, e.g., SHAP or attention heatmaps, may also be incorporated to offer practical recommendations to a human-in-the-loop maintenance decision system. All in all, FusionRUL-Net provides a solid basis for the next generation of intelligent and data-aware prognostics systems in high-stakes industrial settings.

## References


[1] Mahrukh Iftikhar et al., "A Deep Learning Approach to Optimize Remaining Useful Life Prediction for Li-ion Batteries," *Scientific Reports*, vol. 14, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Iordanis Tseremoglou, and Bruno F. Santos, "Condition-Based Maintenance Scheduling of an Aircraft Fleet under Partial Observability: A Deep Reinforcement Learning approach," *Reliability Engineering & System Safety*, vol. 241, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Begum Ay Ture et al., "Stacking-based Ensemble Learning for Remaining Useful Life Estimation," *Soft Computing*, vol. 28, pp. 1337-1349, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zhihao Zhou et al., "An Aircraft Engine Remaining Useful Life Prediction Method based on Predictive Vector Angle Minimization and Feature Fusion Gate Improved Transformer Model," *Journal of Manufacturing Systems*, vol. 76, pp. 567-584, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Abbas Al-Refaie, Majd Al-atrash, and Natalija Lepkova, "Prediction of the Remaining Useful Life of a Milling Machine using Machine Learning," *MethodsX*, vol. 14, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yuzhao Luo et al., "Remaining Useful Life Prediction for Stratospheric Airships based on a Channel and Temporal Attention Network," *Communications in Nonlinear Science and Numerical Simulation*, vol. 143, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Zhaohui Yang et al., "Remaining Useful Life Prediction Approach for Aviation Bearings Based on Multigenerator Generative Adversarial Network and CBAM," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1-9, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jilles Andringa et al., "Counterfactual Explanations for Remaining Useful Life Estimation within a Bayesian Framework," *Information Fusion*, vol. 118, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Xu Xinyao et al., "A Global Attention based Gated Temporal Convolutional Network for Machine Remaining Useful Life Prediction," *Reliability Engineering & System Safety*, vol. 260, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Zhiqiang Peng et al., "Remaining Useful Life Prediction for Aircraft Engines under High-Pressure Compressor Degradation Faults Based on FC-AMSLSTM," *Aerospace*, vol. 11, no. 4, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jie Wang et al., "A Novel Remaining Useful Life Prediction Method under Multiple Operating Conditions based on Attention Mechanism and Deep Learning," *Advanced Engineering Informatics*, vol. 64, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Abdeltif Boujamza, and Saâd Lissane Elhaq, "Optimizing Remaining Useful Life Predictions for Aircraft Engines: A Dilated Recurrent Neural Network Approach," *IFAC-PapersOnLine*, vol. 58, no. 13, pp. 811-816, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] You Keshun, Qiu Guangqi, and Gu Yingkui, "Optimizing Prior Distribution Parameters for Probabilistic Prediction of Remaining Useful Life using Deep Learning," *Reliability Engineering & System Safety*, vol. 242, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fan Wang et al., "A Deep-Learning Method for Remaining Useful Life Prediction of Power Machinery via Dual-Attention Mechanism," *Sensors*, vol. 25, no. 2, pp. 1-19, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Chenchen Wu et al., "A Deep Parallel Spatiotemporal Network Based on Feature Cross Fusion for Remaining Useful Life Prediction of Aero Engine," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] Juseong Lee, and Mihaela Mitici, "Deep Reinforcement Learning for Predictive Aircraft Maintenance using Probabilistic Remaining-Useful-Life Prognostics," *Reliability Engineering & System Safety*, vol. 230, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Sizhe Deng, and Jian Zhou, "Prediction of Remaining Useful Life of Aero-engines Based on CNN-LSTM-Attention," *International Journal of Computational Intelligence Systems*, vol. 17, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Qi Liu et al., "Enhancing Aircraft Engine Remaining Useful Life Prediction via Multiscale Deep Transfer Learning With Limited Data," *Journal of Computational Design and Engineering*, vol. 11, no. 1, pp. 344-355, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yujie Cheng et al., "A Health State-Related Ensemble Deep Learning Method for Aircraft Engine Remaining Useful Life Prediction," *Applied Soft Computing*, vol. 135, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Slawomir Szrama, and Tomasz Lodygowski, "Aircraft Engine Remaining Useful Life Prediction using Neural Networks and Real-Life Engine Operational Data," *Advances in Engineering Software*, vol. 192, 2024. [CrossRef] [Google Scholar] [Publisher Link]