

Article

Not peer-reviewed version

Bidirectional LSTM Autoencoder Transformer for Remaining Useful Life Estimation

[Zhengyang Fan](#)*, [Wanru Li](#), [Kuo-Chu Chang](#)

Posted Date: 27 November 2023

doi: 10.20944/preprints202311.1705.v1

Keywords: transformer; self-supervised learning; autoencoder; remaining useful life prediction; bidirectional LSTM; turbofan engine



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Bidirectional LSTM Autoencoder Transformer for Remaining Useful Life Estimation

Zhengyang Fan ^{1,*}, Wanru Li ¹ and Kuo-Chu Chang ¹

¹ Department of Systems Engineering and Operations Research, George Mason University; wli15@gmu.edu (W.L.); kchang@gmu.edu (K.C.)

* Correspondence: zfan3@gmu.edu

Abstract: Estimating the Remaining Useful Life (RUL) of aircraft engines holds a pivotal role in enhancing safety, optimizing operations, and promoting sustainability, thus being a crucial component of modern aviation management. Precise RUL predictions offer valuable insights into an engine's condition, enabling informed decisions regarding maintenance and crew scheduling. In this context, we propose a novel RUL prediction approach in this paper, harnessing the power of Bi-directional LSTM and Transformer architectures, known for their success in sequence modeling, such as natural languages. We adopt the encoder part of the full Transformer as the backbone of our framework, integrating it with a self-supervised denoising autoencoder that utilizes Bidirectional LSTM for improved feature extraction. Within our framework, a sequence of multivariate time series sensor measurements serves as the input, initially processed by the Bidirectional LSTM autoencoder to extract essential features. Subsequently, these feature values are fed into our Transformer encoder backbone for RUL prediction. Notably, our approach simultaneously trains the autoencoder and Transformer encoder, different from the naive sequential training method. Through a series of numerical experiments carried out on the C-MAPSS datasets, we demonstrate that the efficacy of our proposed models either surpasses or stands on par with that of other existing methods.

Keywords: transformer; self-supervised learning; autoencoder; remaining useful life prediction; bidirectional LSTM; turbofan engine

1. Introduction

The aviation and surface systems of today are characterized by an ever-increasing level of automation and cutting-edge machinery, equipped with advanced sensors that diligently monitor essential functions of aircraft, ships, and auxiliary systems. In conjunction with the ongoing development of Industry 4.0, a transformative industrial paradigm that harmonizes sensors, software, and intelligent control to enhance manufacturing processes, the field of aircraft maintenance is undergoing a notable transition. This transition involves departing from conventional corrective and preventive maintenance methods towards a data-driven approach known as condition-based predictive maintenance (CBPM).

CBPM is an approach geared towards proactively assessing the health and maintenance needs of critical systems, aiming to avert unscheduled downtime, streamline maintenance processes, and ultimately enhance productivity and profitability [1]. An integral facet of this contemporary maintenance strategy is the task of predicting Remaining Useful Life (RUL). RUL prediction is a prominent challenge that has garnered significant attention from the research community in recent years. It revolves around estimating the time interval between the present moment and the anticipated end of a system's operational service life, serving as a fundamental element in the overall predictive maintenance framework.

Traditional statistical-based methods used for RUL estimation rely on fitting probabilistic models to data. They assume that system component degradation can be characterized by specific parametric functions or stochastic process models. Giantomassi et al. [2] introduced a Hidden Markov Model (HMM) within a Bayesian inference framework for predicting RUL in turbofan

engines. Wiener processes have been effectively used to model component degradation in these engines, as demonstrated in [3], which implemented a two-stage degradation modeling employing nonlinear Wiener processes for improved RUL predictions. Similarly, Yu et al. [4] also utilized a nonlinear Wiener process model, considering sources of uncertainty and providing RUL probability distributions for online and offline scenarios. Additionally, in conjunction with kernel principal component analysis, Feng et al. [5] developed a degradation model based on the Wiener process for turbofan engine RUL prediction. Lv et al. [6] introduced a predictive maintenance strategy tailored for multi-component systems, combining data and model fusion through particle filtering and degradation distribution modeling to enhance RUL predictions. Zhang et al. [7] integrated a non-homogeneous Poisson process and a Weibull proportional hazard model, along with intrinsic failure rate modeling, to predict RUL. While these conventional statistical-based RUL prediction methods have shown effectiveness, they often rely on prior knowledge of the system's underlying physics, posing limitations when such information is incomplete or absent. Furthermore, handling high-dimensional data presents challenges, known as the curse of dimensionality, in these methods. For a more comprehensive discussion on traditional statistical-based approaches for RUL estimation, readers are referred to [8].

As data volume and computing capabilities continue to expand, artificial intelligence and machine learning (AI/ML) have found success in applications across various domains, including cyber security [9-10], geology [11,12], aerospace engineering [13,14], and transportation [15,16]. In parallel, the focus of research on data-driven approaches for RUL estimation is in the process of transitioning from conventional statistical-based probabilistic techniques to AI/ML methods. This transition is attributed to AI/ML's capacity to address the limitations associated with classical statistics-based methodologies.

Typically, AI/ML methods for RUL prediction can be categorized as shallow and deep methods. Shallow methods often rely on Support Vector Machines (SVMs) and tree ensemble algorithms, while deep methods employ complex neural networks like Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN).

In the realm of shallow methods, Ordóñez et al. [17] integrated auto-regressive integrated moving average (ARIMA) time series models with SVMs to predict RUL for turbofan engines, showing superior predictive capabilities compared to the vector auto-regressive moving average model (VARMA). García Nieto et al. [18] introduced a hybrid model that combines SVMs with particle swarm optimization to optimize SVM kernel parameters, enhancing their performance. Meanwhile, Benkedjouh et al. [19] proposed a method that combines nonlinear feature reduction and SVMs, effectively reducing the number of monitoring signal features for wear level estimation and RUL prediction. Their simulation results underscored the method's suitability for monitoring tool wear and enabling proactive maintenance actions. To tackle the challenges posed by nonlinearity, high dimensionality, and complex degradation processes, Wang et al. [13] combined the Random Forest algorithm for feature selection with single exponent smoothing for noise reduction. It further employed a Bayes-optimized Multilayer Perceptron (MLP) model for RUL prediction.

Deep methods have gained substantial attention for RUL prediction, with researchers exploring the capabilities of deep neural networks in this domain. In [21], a fusion model named B-LSTM combines a comprehensive learning system for feature extraction with Long Short-Term Memory (LSTM) to handle temporal information in time-series data. Ensarioğlu et al. [22] introduced an innovative approach that incorporated difference-based feature construction, change-point-detection-based piecewise linear labeling, and a hybrid 1D-CNN-LSTM neural network to predict RUL. In [23–25], LSTM networks are employed and trained in a supervised manner to directly estimate RUL. Li et al. [26] presented a data-driven approach, utilizing CNN with a time window strategy for sample preparation to enhance feature extraction. Yu et al. [27] proposed a two-step method, involving a bidirectional RNN autoencoder followed by similarity-based curve matching for RUL estimation. The autoencoder converted high-dimensional sensor data into low-dimensional representation embeddings to facilitate subsequent similarity matching. Zhao et al. [28] introduced a double-channel hybrid spatial-temporal prediction model, incorporating CNN and a bidirectional

LSTM network to overcome the limitations of using CNN and LSTM individually. Similarly, Peng et al. [29] developed a dual-channel LSTM model that adaptively selects time features and performs initial processing on time feature values, using LSTM to extract time features and first-order time feature information. Wang et al. [30] presented a multi-scale LSTM designed to cover three distinct degradation stages of aircraft engines: constant stage, transition stage, and linear degradation stage. To tackle the issue of dissimilar data distributions in training and testing, Lyu et al. [31] utilized LSTM networks for feature extraction from sensor data and applies multi-representation domain adaptation methods to ensure effective feature alignment between source and target domains. Recently, Deng et al. [32] presented a novel multi-scale dilated CNN that employs a new multi-scale dilation convolution fusion unit to improve the receptive field and operational efficiency when predicting the RUL of essential components in mechanical systems.

While AI/ML methods, particularly those based on RNN and CNN for RUL estimation, consistently outperform classical statistical methods, it's important to acknowledge that there are limitations associated with both RNN-based and CNN-based deep learning approaches. In the case of RNNs, they process historical sensor data in a sequential manner, inherently hindering their capacity for efficient parallel computation. This can result in prolonged training and inference times. Additionally, RNNs struggle to capture long-range dependencies, rendering them less suitable for handling extensive time-series inputs, such as observed sensor data over time. In contrast, CNNs face limitations related to the receptive field at a given time step, which depends on factors like convolutional kernel size and the number of layers. Consequently, their ability to capture long-range information is limited.

In recent years, the Transformer network [33] has gained prominence for its effectiveness in natural language sequence modeling and its successful adaptation to various applications, such as computer vision [34]. The multi-head self-attention mechanism in the Transformer allows it to capture long-range dependencies over time efficiently and in parallel. This capability addresses the limitations of RNN architecture, leading to state-of-the-art results in various sequence modeling tasks. In this study, we propose a Transformer-based model for RUL estimation. In contrast to RNN and CNN-based models, the Transformer architecture stands out for its ability to access any part of historical data without being constrained by temporal distance. It achieves this by leveraging the attention mechanism to simultaneously process a sequence of data, enhancing its potential for capturing long-term dependencies.

To ensure the robustness of the network, it is imperative to handle noise and outliers. The neural network's ability to represent data hinges on the quality of the input source. Therefore, in accurately predicting RUL for aircraft engines, we incorporate a Bidirectional LSTM Denoising AutoEncoder (BiLSTM-DAE). This autoencoder, renowned for its capability to learn representations from noisy data, is employed for data reconstruction.

The main contributions of this work are:

1. We propose a novel BiLSTM-DAE based Transformer architecture for RUL prediction. To the best of our knowledge, this is the first successful attempt in combining Transformer architecture and denoising autoencoder for aircraft engine RUL prediction.
2. We explore the significance of the features extracted from the BiLSTM-DAE in the context of RUL prediction.
3. We conduct a series of experiments using four CMPASS turbofan engine datasets and demonstrate that our model's performance surpasses or is on par with that of existing methods.

The rest of the paper is structured as follows: Section 2 provides a review of related work in Transformer-based RUL estimation. In Section 3, the proposed model architecture is detailed. Section 4 delves into experimental specifics, results, and analysis. Finally, Section 5 concludes the paper.

2. Related Work

Due to the remarkable performance of Transformer networks in sequence modeling, they have found successful applications in predicting the RUL based on time series sensor data. Mo et al. [35] devised a methodology employing the Transformer to capture both short and long interdependencies

within time series sensor data. Furthermore, the authors introduced a gated convolutional unit to enhance the model's capability to assimilate local contextual information at each temporal step. In a recent study, Ai et al. [36] introduced a multilevel fusion Seq2Seq model that integrates a Transformer network architecture along with a multi-scale feature mining mechanism. Using an encoder-decoder structure entirely relying on self-attention and devoid of any RNN or CNN components, Zhang et al. [37] introduced a dual-aspect self-attention model based on the Transformer. Their model incorporates two encoders and demonstrates effectiveness in handling long data sequences, as well as learning to emphasize the most crucial segments of the input. Similarly, Hu et al. [38] devised an adapted Transformer model that integrates LSTM and CNN components to extract degradation-related features from various angles. Their model utilizes the full Transformer as the backbone structure, distinguishing it from most Transformer encoder-based networks. Harnessing domain adaptation for the prediction of RUL, Li et al. [39] introduced an approach that aligns distributions both at the feature level and the semantic level using a Transformer architecture, thereby enhancing model convergence. In a recent study [40], a convolutional Transformer model was introduced, merging the ability to capture the global context with the attention mechanism and the modeling of local dependencies through convolutional operations. This model effectively extracted degradation-related information from raw data, encompassing both local and global perspectives. In Chadha et al. [41], a shared temporal attention mechanism was devised to identify patterns related to RUL that evolve over time. Additionally, the authors introduced a split-feature attention block, enabling the model to focus on features from various sensor channels. This approach aims to capture temporal degradation patterns in individual sensor signals before establishing a shared correlation across the entire feature range. As Many AI/ML models predominantly prioritize prediction accuracy and tend to overlook the aspect of model reliability, which is a critical factor limiting their industrial applicability. In response, a Bayesian Gated-Transformer model designed for reliable RUL prediction was introduced [42], emphasizing the quantification of uncertainty in their predictions. In an effort to effectively capture temporal and feature dependencies, Zhang et al. [42] introduced a trend augmentation module and a time-feature attention module within the conventional Transformer model. They complemented this approach with a bidirectional gated recurrent unit for the extraction of concealed temporal information.

The study conducted in [43] closely aligns with our current work. Specifically, it involved the integration of a denoising autoencoder for preprocessing raw data and subsequently utilized a Transformer network for predicting the RUL of lithium-ion batteries. However, several distinctions set our work apart from theirs:

4. While Chen et al. [43] employed a denoising autoencoder akin to a multi-layer perceptron for signal reconstruction, our model opted for BiLSTM networks to encode features, a choice better suited for preserving temporal information within raw signals.
5. In our model, we introduced a learnable positional encoding, a feature that distinguishes our approach from the study conducted by Chen et al. [43], which employed trigonometric functions to encode a fixed positional structure before the training process.

3. Methodology

In this section, we provide a detailed exposition of the overall architecture of our BiLSTM-DAE-based Transformer model. As depicted in Figure 1, our model is comprised of two integral components: the BiLSTM Autoencoder part and the Transformer part. To facilitate comprehension, we will initially delve into the BiLSTM-DAE part in Subsection 3.1, followed by a detailed introduction to the Transformer part in Subsection 3.2.

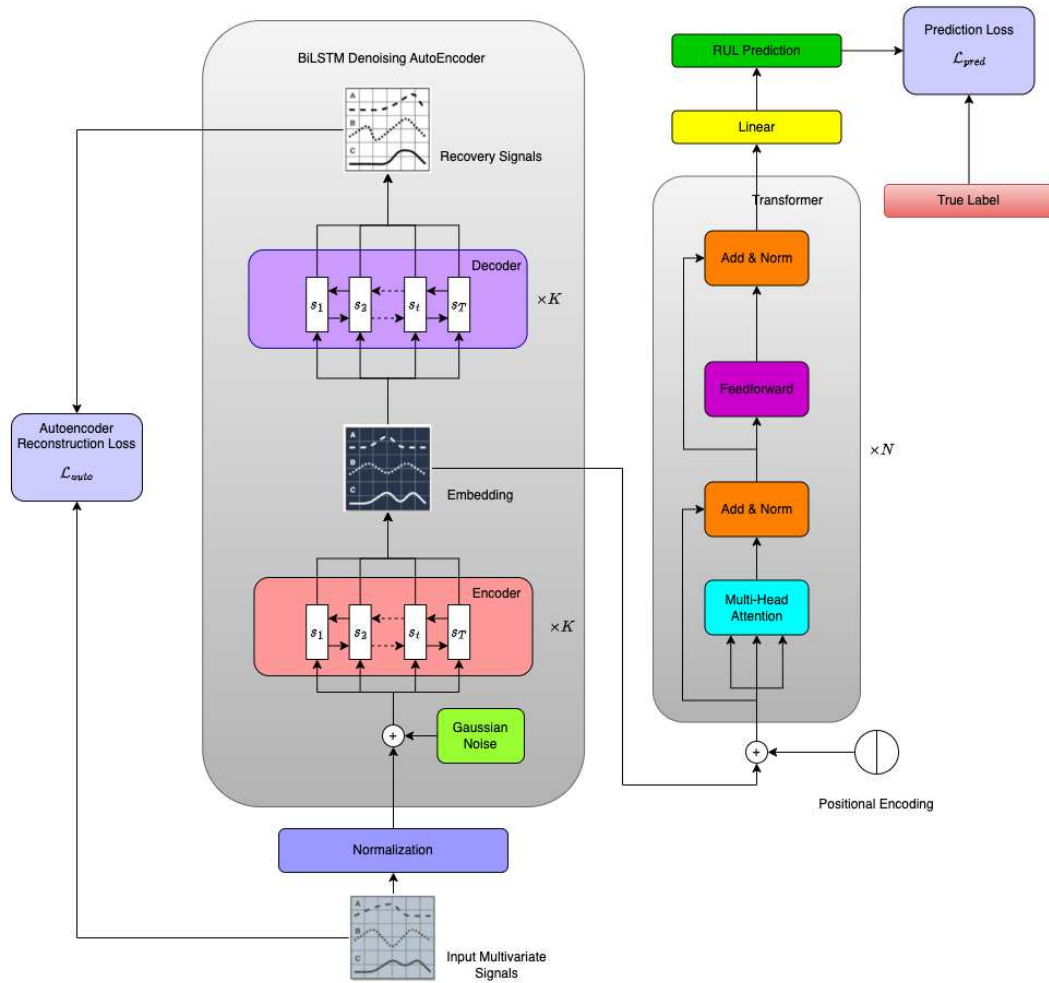


Figure 1. Overall Structure of BiLSTM-DAE Transformer Model.

3.1. Bidirectional LSTM Autoencoder

LSTM, a widely employed architecture in time sequence AI/ML modeling, incorporates gates to regulate information flow in recurrent computations, excelling in the retention of long-term memories. The original LSTM formulation was proposed in Hochreiter and Schmidhuber[44], and the Vanilla LSTM, a popular variant, was introduced in [45], augmenting the LSTM architecture with a forget gate to enhance its capabilities. Figure 2 shows the vanilla LSTM cell.

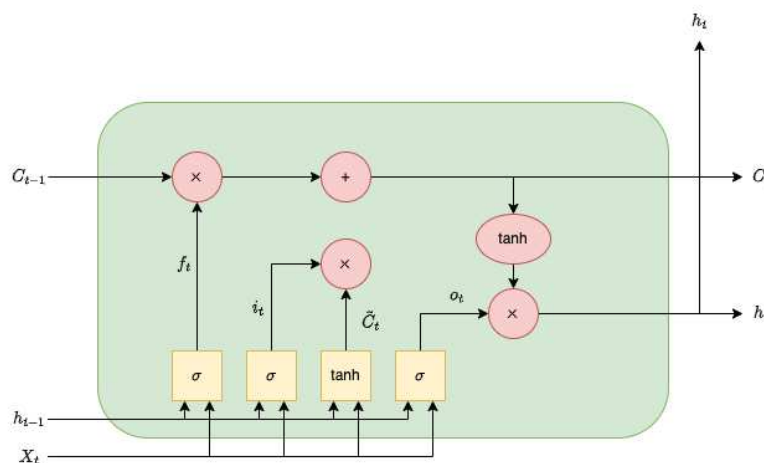


Figure 2. A Vanilla LSTM Cell.

LSTM memory cells encompass various neural networks referred to as gates, which play a crucial role in managing interactions between memory units, determining what data to retain or discard during training. The input and output gates dictate whether the state memory cells can be modified by incoming signals, while the forget gate governs the decision to retain or erase the preceding signal's status. In the cell, the functional relationship for each component is given as follows:

$$\begin{bmatrix} f_t \\ i_t \\ \tilde{C}_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{bmatrix} \left(W_\theta \begin{bmatrix} X_t \\ h_{t-1} \end{bmatrix} + b_\theta \right) \quad (1)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (2)$$

$$h_t = o_t \odot \tanh(c_t) \quad (3)$$

The forget gate (f_t), input gate (i_t), and output gate (o_t) are denoted in the LSTM architecture. The forget gate manages the removal of historical information from C_{t-1} , while the input and output gates govern the updating and outputting of specific information. The symbol σ represents a nonlinear activation function, and the symbol \odot denotes the element-wise product.

BiLSTM, with its bidirectional propagation, processes information in both directions simultaneously, capturing both past and future context. Illustrated in Figure 3, at each time step t , the input is fed into both the forward and backward LSTM networks, and the BiLSTM output is a representation of this bidirectional processing.

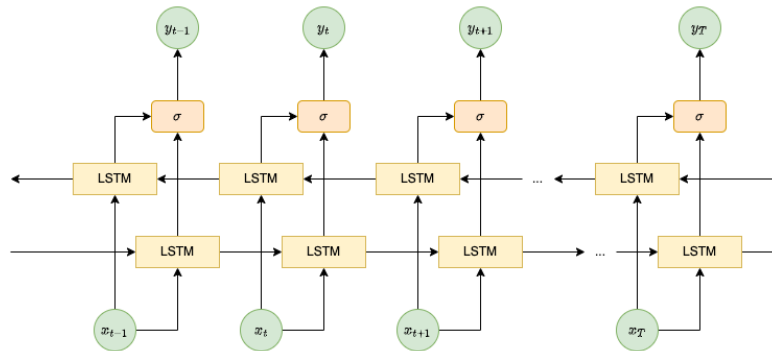


Figure 3. BiLSTM Architecture.

Denoising autoencoder (DAE) is a type of artificial neural network used in machine learning for feature learning and data denoising [46]. It is a variant of the traditional autoencoder, which is designed to learn efficient representations of data by training on the corrupted version of the input data. The architecture of a denoising autoencoder typically consists of an encoder and a decoder. The encoder maps the input data to a lower dimensional representation, and the decoder reconstructs the original data from this representation. During the training process, the network minimizes the reconstruction error, encouraging the model to capture meaningful features while ignoring noise.

Due to the challenges posed by noisy and complex multivariate time series sensor reading data, we propose a BiLSTM-DAE. Time series data, especially in multivariate scenarios, often exhibit fluctuations, irregularities, and diverse patterns that can impact the performance of traditional models. The bidirectional nature of the BiLSTM allows it to capture dependencies in both forward and backward directions, facilitating a more comprehensive understanding of temporal relationships.

Our motivation stems from the need for a robust model capable of handling the intricacies of multivariate time series datasets. The bidirectional architecture enhances the model's ability to capture temporal dependencies, while the denoising autoencoder aspect improves its resilience

against noise and variations in the data. This combined approach aims to provide an effective solution for tasks such as feature extraction, anomaly detection, and prediction within the context of multivariate time series analysis.

The architecture presented in the left part of Figure 1 focuses on our detailed BiLSTM-DAE model. In this illustration, when provided with a sequence of sensor readings for predicting the RUL of aircraft engines, we initially introduce small Gaussian noises to corrupt the input sequence. The inclusion of Gaussian noise in the training process of an autoencoder serves as implicit regularization, preventing the model from overfitting to the noise present in the input sequence. This, in turn, promotes the development of a more robust feature representation. The corrupted data is then directed into a K -layer BiLSTM encoder. The outputs from this BiLSTM encoder are utilized in two distinct ways:

6. They are fed into the Transformer component as input data to predict the RUL for aircraft engines.
7. They are inputted into the K -layer BiLSTM decoder of the BiLSTM-DAE module to reconstruct the original data before corruption. It's noteworthy that although the number of layers for the BiLSTM encoder and decoder can differ, we keep them the same in our experiments.

Finally, a reconstruction loss L_{auto} is computed in a conventional autoencoder model to minimize the disparity between the initial input signal x and the reconstructed/recovered signal \hat{x} . In this study, the ℓ_2 is employed to measure the difference between x and \hat{x} , as expressed in the following equation:

$$L_{auto} = \frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|_2 \quad (4)$$

Here, N denotes the number of sample sequences in our dataset.

3.2. Transformer and Multi-head Attention

As mentioned earlier, the Transformer architecture has emerged recently as a pivotal paradigm in natural language processing and various sequence-to-sequence tasks. The Transformer architecture relies on self-attention mechanisms, enabling it to capture long-range dependencies in input sequences efficiently [29]. Unlike traditional recurrent or convolutional architectures, the Transformer abandons sequential processing, embracing a parallelizable architecture that facilitates accelerated training on parallel hardware.

At the core of the Transformer's success is the self-attention mechanism. Given an input sequence $X = \{x_1, x_2, \dots, x_n\}$, the self-attention mechanism computes a set of attention weights, allowing each element in the sequence to attend to all other elements. This is achieved through the following formulation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

Here, Q, K and V denote the query, key, and value matrices, respectively, and d_k represents the dimensionality of the key vectors. The SoftMax operation normalizes the attention scores, determining the weight each element contributes to the final output.

To enhance the expressive power of self-attention, the Transformer employs a mechanism known as multi-head attention. This involves linearly projecting the input sequences into multiple subspaces, each with its own attention mechanism. The outputs from these multiple heads are then concatenated and linearly transformed to produce the final output. Formally, the multi-head attention is defined as:

$$MultiHead(Q, K, V) = Concat(head^1, head^2, \dots, head^h)W_o \quad (6)$$

where $head^i = Attention(QW_Q^i, KW_K^i, VW_V^i)$, and W_Q^i, W_K^i, W_V^i and W_o are all learnable weight matrices.

This multi-head mechanism enables the model to attend to different aspects of the input sequence simultaneously, capturing diverse patterns and dependencies.

Complementing the self-attention mechanism, the Transformer incorporates a feed-forward layer for capturing complex, non-linear relationships within the encoded representations. Each position in the sequence is processed independently through a feed-forward neural network, composed of two linear transformations with a ReLU activation in between:

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad (7)$$

Here, W_1, b_1, W_2 , and b_2 are learnable parameters. The feed-forward layer introduces additional flexibility, allowing the model to capture intricate patterns and interactions in the data, enhancing its capacity to learn intricate representations.

Finally, as illustrated in the right part of Figure 1, the prediction loss L_{pred} is computed to minimize the disparity between the predicted RUL \hat{y} (output of feed-forward layer) and the true RUL y . In this study, the ℓ_2 is employed to measure the difference between y and \hat{y} , as expressed in the following equation:

$$L_{pred} = \frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|_2 \quad (8)$$

Here, N denotes the number of sample sequences in our dataset.

3.3. Learning

Throughout our training process, our main goal is to acquire meaningful representations of input signals while simultaneously achieving accurate predictions of RUL. The objective function guiding our training encompasses a weighted average of the autoencoder reconstruction loss L_{auto} and prediction loss L_{pred} . This is formally defined as:

$$L = L_{auto} + \alpha \cdot L_{pred} + \lambda \cdot \Omega(\Theta) \quad (9)$$

Here, α serves as the parameter balancing the weights between the two distinct losses, while λ acts as the shrinkage parameter determining regularization strength. The set of parameters in both the BiLSTM-DAE and Transformer networks is denoted by Θ , and the function $\Omega(\cdot)$ represents the regularization function. In our experimental setup, we opt for the ℓ_2 norm to regularize parameters Θ .

4. Experimental Results and Analysis

Our experimentation is carried out on a computing system equipped with an Intel Core i9 3.6 GHz processor, 64 GB of RAM, and an NVIDIA RTX 3080 GPU. The operating system employed is Windows 10, and the programming platform utilized is Python.

In the subsequent subsections of this section, we will commence by introducing the dataset utilized in our study. Following this, we will describe the performance metrics employed for the evaluation of the proposed method. Lastly, we will present our numerical experiments, conduct a comparative analysis of its performance against existing methods, and engage in a comprehensive discussion.

4.1. C-MAPSS Dataset and Preprocessing

The NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset stands as a benchmark in the field, offering a rich source of information for researchers and practitioners engaged in the challenging task of predicting the RUL of turbofan engines [47]. This dataset encompasses simulated sensor data generated by various turbofan engines over time, providing a realistic representation of engine health deterioration. The dataset is structured into four sub-datasets, namely FD001, FD002, FD003, and FD004, each designed to capture different aspects of engine behavior under various operational conditions and fault scenarios.

The FD001 and FD003 datasets focus on a single operational condition, each containing one and two fault types, respectively. In contrast, the FD002 and FD004 datasets incorporate six distinct operational conditions, along with one and two fault types, respectively. These sub-datasets include detailed information such as engine numbers, serial numbers, configuration items, and sensor data from 21 sensors. See Table 1 below for a detailed description of these sensors. The sensor data simulates the progressive degradation of the engine, starting from a healthy state and culminating in failure, under diverse initial conditions. Nonetheless, not every piece of sensor data contains information conducive to the estimation of RUL. Certain sensors, for instance, exhibit constant measurements throughout the entire life cycle. To mitigate computational complexity, we adopt the approach outlined in [35], selectively incorporating data from 14 sensors (sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21) into our training process.

Recognizing the disparate numerical ranges resulting from distinct sensor measurements, we also employ a min-max normalization technique by using the following formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{10}$$

This normalization procedure serves to standardize the input sensor data, ensuring a consistent and comparable range for all data points.

Table 1. C-MAPSS Monitoring Sensor Data Description.

Symbol	Description	Units
T2	Total temperature at fan inlet	R
T24	Total temperature at LPC inlet	R
T30	Total temperature at HPC inlet	R
T50	Total temperature at LPT inlet	R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Ne	Physical core speed	rpm
epr	Engine pressure ratio	-
Ps30	Static pressure at HPC outlet	psia
Phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRe	Corrected core speed	rpm
BPR	Bypass ratio	-
farB	Burner fuel-air ratio	-
htBleed	Bleed Enthalpy	-
Bf-dmd	Demanded fan speed	rpm
PCNfR-dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

The training and testing trajectories within each sub-dataset are designed to assess the model's performance under varying conditions. For instance, FD001 and FD003 consist of 100 training and testing trajectories, while FD002 and FD004 feature 260 and 259 training trajectories and 259 and 248 testing trajectories, respectively. The operational conditions and fault modes further contribute to the dataset's complexity, making it a suitable benchmark for evaluating the effectiveness of predictive models. Table 2 summarizes the key statistics for C-MAPSS dataset.

Table 2. Parameters of the C-MAPSS dataset.

Dataset	FD001	FD002	FD003	FD004
No. of Training Trajectories	100	260	100	249
No. of Testing Trajectories	100	259	100	248
Operating Conditions	1	6	1	6
Fault Modes	1	1	2	2

The training set spans the entire operational lifecycle of the turbofan engine, capturing data from its initial operation to degradation and failure. Conversely, the test set begins at a healthy state and undergoes arbitrary truncation, with the operating time periods leading up to system failure calculated from these truncated data. Additionally, the test set includes the actual RUL values of the test engine, facilitating the assessment of the model's accuracy in predicting the time remaining until failure.

In light of the challenges associated with RUL labeling in run-to-fail sequences, our approach diverges from earlier methodologies, such as the linear decrease assumption in a previous study [48]. This assumption posits that RULs decrease linearly over time, implying a linear development of the system's degradation state. However, this assumption may not accurately reflect the reality of system degradation, especially during the initial stages of its life cycle when degradation is typically negligible.

To address this, we adopt a more nuanced RUL labeling strategy inspired by the truncated linear model [25]. This approach recognizes that the system experiences varying degrees of degradation over different phases of its life cycle. In Figure 4, our labeling procedure is visually represented. Initially, RULs are labeled with a constant value, denoted as RUL_{max} , akin to the horizon dot line. As the system operates over time, it transitions into a phase of linear degradation until it ultimately reaches failure. This truncated linear model provides a more realistic representation of RUL evolution, accommodating the dynamic nature of system degradation. By assigning RUL labels that consider both the initial phase of system operation and subsequent linear degradation periods, our approach captures the nuanced progression of system health over time. This ensures that our RUL labeling aligns more closely with the actual behavior of the turbofan engines in the dataset.

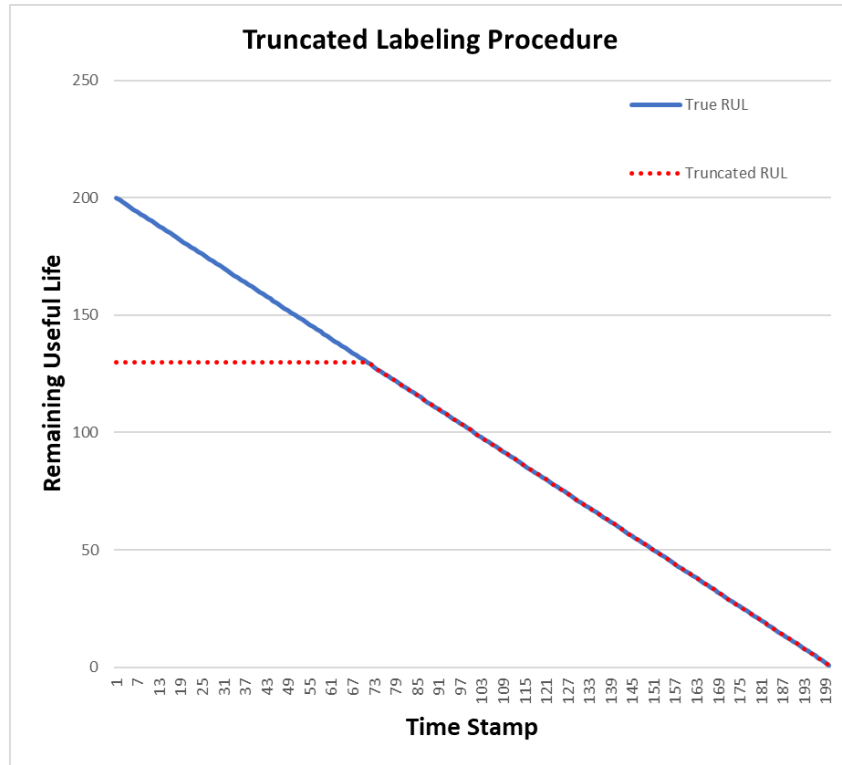


Figure 4. Truncated RUL of C-MAPSS Dataset, $RUL_{max} = 130$.

4.2. Evaluation Metrics

Two evaluation metrics used to evaluate the prediction performance of RUL include root mean squared error (RMSE) and Score, which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (11)$$

$$Score = \begin{cases} \sum_{i=1}^N e^{-\left(\frac{y_i - \hat{y}_i}{13}\right)} - 1, & d < 0 \\ \sum_{i=1}^N e^{-\left(\frac{y_i - \hat{y}_i}{10}\right)} - 1, & d \geq 0 \end{cases} \quad (12)$$

With an increase in the absolute value of $y_i - \hat{y}_i$, both evaluation indices show an upward trend. Notably, when $y_i - \hat{y}_i > 0$, the Score metric introduces distinct penalty weights for model predictions lag and advance. In cases where the predicted value is smaller than the true value, indicating prediction advancement, the penalty coefficient is correspondingly smaller. Conversely, if the prediction lags, the consequences are deemed more severe, leading to a larger penalty coefficient. Anticipating the RUL value earlier facilitates proactive maintenance planning, mitigating potential losses and underscoring the importance of the penalty coefficient's dynamic adjustment based on the direction of prediction deviation.

4.3. RUL Prediction

To verify the performance of the proposed BiLSTM-DAE Transformer model, we compare our model with other existing RUL prediction methods, including multi-layer perceptron (MLP) [49], support vector regression (SVR) [49], CNN [49], LSTM [25], BiLSTM [50], deep belief network ensemble (DBNE) [51], B-LSTM [21], gated convolutional Transformer (GCT) [35], CNN+LSTM [52] and multi-head CNN+LSTM [53]s. Table 3 shows the comparison results.

Table 3. Performance Comparison. The **bold** number represents the best model, while the underscore number represents the second-best model.

Method	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
MLP	37.56	18000	80.03	7800000	37.39	17400	77.37	5620000
SVR	20.96	1380	42.00	590000	21.05	1600	45.35	371000
CNN	18.45	1290	30.29	13600	19.82	1600	29.16	7890
LSTM	16.14	338	24.49	4450	16.18	852	28.17	5550
BiLSTM	13.65	295	23.18	4130	13.74	<u>317</u>	24.86	5430
DBNE	17.27	523	37.28	49800	18.47	574	30.96	12100
B-LSTM	12.45	279	15.36	4250	13.37	356	16.24	5220
GCT	<u>11.27</u>	-	22.81	-	<u>11.42</u>	-	24.86	-
CNN+LSTM	16.16	303	20.44	<u>3440</u>	17.12	1420	23.25	4630
Multi-head CNN+LSTM	12.19	<u>259</u>	19.93	4350	12.85	343	22.89	<u>4340</u>
Proposed Method	10.98	186	<u>16.12</u>	2937	11.14	252	<u>18.15</u>	3840

As depicted in Table 3, our BiLSTM-DAE-Transformer architecture exhibits superior performance on both the FD001 and FD003 datasets, showcasing relative improvements in RMSE compared to previous state-of-the-art models. Specifically, the relative improvements range from 0.28 for FD003 to 0.29 for FD001 in terms of RMSE. Additionally, our model secures the second-best performance on the FD002 and FD004 datasets, emphasizing its effectiveness across different operational conditions. Notably, our model achieves the smallest Score across all four datasets in comparison to existing methods. Particularly noteworthy are the substantial Score improvements for FD001 and FD003, where our model outperforms state-of-the-art models by 28% and 26%, respectively. In contrast to other autoencoder methods that involve building the autoencoder and prediction model separately, our end-to-end approach requires less effort in implementation. This advantage stems from the seamless integration of the autoencoder and prediction model within our architecture. The end-to-end nature of our method streamlines the implementation process, making it more efficient and requiring fewer resources.

To further substantiate the reliability of our model, we conduct a comparative analysis by visually contrasting the estimated RUL with the ground truth RULs, presenting the results in Figure 5. This visual representation offers a comprehensive understanding of the accuracy and effectiveness of our RUL predictions across diverse datasets.

To enhance clarity in our analysis, we organize all test sequences based on their corresponding ground-truth RUL lengths along the horizon axis in ascending order. Upon reviewing Figure 5, it becomes evident that our model consistently produces satisfactory outcomes. Notably, in the case of FD001 and FD003, our model demonstrates accurate RUL predictions in both the early and late stages of the engine life cycle, with relatively smaller variations. Conversely, for the FD002 and FD004 datasets, predictions appear noisier at the early and late stages compared to FD001 and FD003. We attribute this variability to differences in the complexity of working conditions, as characterized by the number of operational conditions and fault modes, and the size of the training dataset.

As indicated in Table 2, FD001 training samples are derived from a single working condition, offering a simpler scenario for training. Similarly, FD003 involves two working conditions. In contrast, FD002 and FD004 present more intricate scenarios with six and twelve working conditions, respectively (comprising six operational conditions and one or two fault modes). This increased complexity in FD002 and FD004 can contribute to the noisier predictions observed, accentuating the impact of working condition intricacies and training dataset size on the model's performance across different datasets.

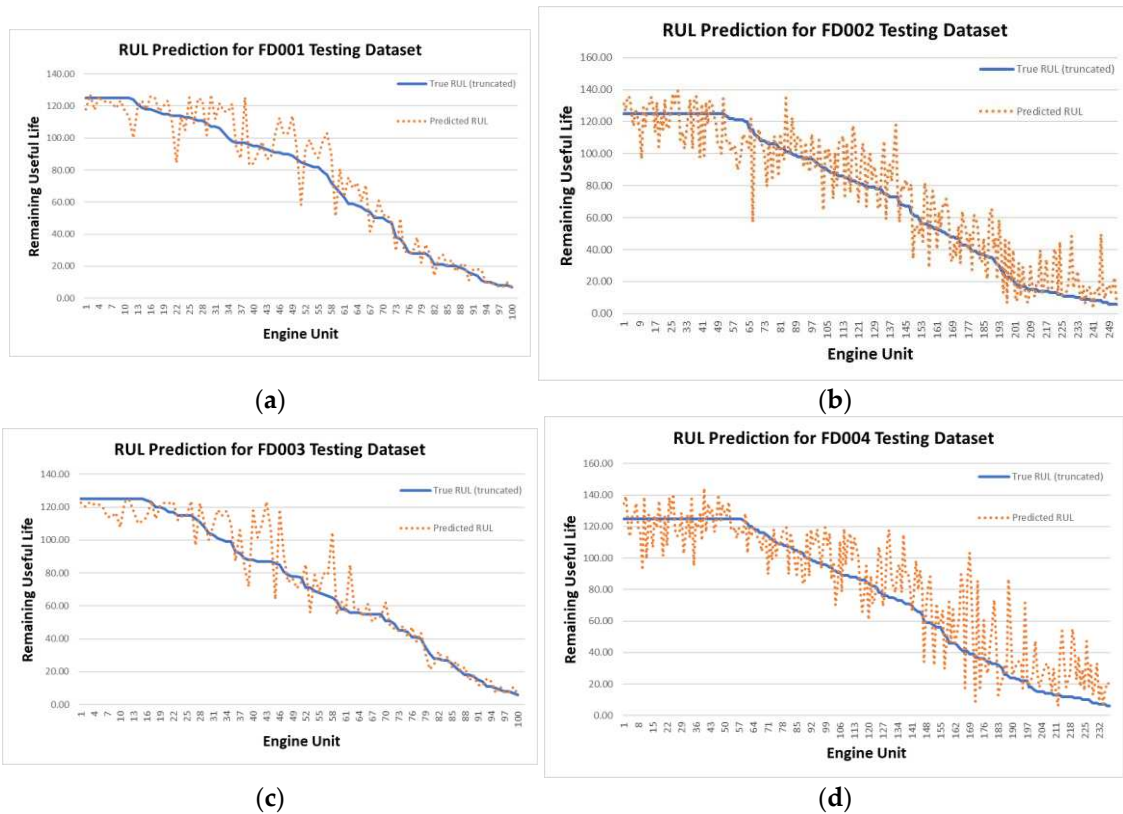


Figure 5. Comparisons between predicted RUL and ground truth RUL for all four C-MAPSS datasets. (a) RUL prediction for FD001; (b) RUL prediction for FD002; (c) RUL prediction for FD003; (d) RUL prediction for FD004.

5. Conclusions

This paper presents a novel and effective approach for RUL prediction in turbofan engines using a hybrid architecture comprising Bidirectional LSTM, Denoising Autoencoder, and Transformer models. Leveraging the C-MAPSS dataset, we demonstrated the superior performance of our BiLSTM-DAE-Transformer architecture compared to state-of-the-art models across multiple datasets. Our truncated RUL strategy for RUL labeling and the incorporation of a selective set of 14 sensors significantly contribute to the success of our model. The end-to-end nature of our approach simplifies implementation compared to alternative Auto-Encoder-based methods, where the autoencoder and prediction model are built separately. The visual validation of our model against ground truth RULs showcases its accuracy and reliability, particularly in the early and late stages of engine life cycles. Notably, our model outperforms existing methods, achieving substantial improvements in Score and RMSE. These results underscore the practical applicability of our proposed approach in turbofan engine prognostics, enabling more accurate maintenance planning and the avoidance of unplanned system shutdowns.

Future work may explore enhancements to accommodate additional complexities and diverse operational scenarios, further refining the precision and versatility of our RUL prediction model.

Supplementary Materials: Not applicable

Author Contributions: Conceptualization, Z.F. and K.C.; writing, manuscript preparation, Z.F. and W.L.; review and editing, W.L. and K.C.; supervision and project management, K.C..All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data of this paper came from the NASA Prognostics Center of Excellence, and the data acquisition website was: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan>

Acknowledgments: Not applicable

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, Z.; Chang, K.; Ji, R.; Chen, G. Data Fusion for Optimal Condition-Based Aircraft Fleet Maintenance with Predictive Analytics. *J. Adv. Inf. Fusion* **2023**, *In Press*.
2. Giantomassi, A.; Ferracuti, F.; Benini, A.; Ippoliti, G.; Longhi, S.; Petrucci, A. Hidden Markov Model for Health Estimation and Prognosis of Turbofan Engines.; American Society of Mechanical Engineers Digital Collection, June 12 2012; pp. 681–689.
3. Lin, J.; Liao, G.; Chen, M.; Yin, H. Two-Phase Degradation Modeling and Remaining Useful Life Prediction Using Nonlinear Wiener Process. *Comput. Ind. Eng.* **2021**, *160*, 107533, doi:10.1016/j.cie.2021.107533.
4. Yu, W.; Tu, W.; Kim, I.Y.; Mechefske, C. A Nonlinear-Drift-Driven Wiener Process Model for Remaining Useful Life Estimation Considering Three Sources of Variability. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107631, doi:10.1016/j.ress.2021.107631.
5. Feng, D.; Xiao, M.; Liu, Y.; Song, H.; Yang, Z.; Zhang, L. A Kernel Principal Component Analysis-Based Degradation Model and Remaining Useful Life Estimation for the Turbofan Engine. *Adv. Mech. Eng.* **2016**, *8*, 1687814016650169, doi:10.1177/1687814016650169.
6. Lv, Y.; Zheng, P.; Yuan, J.; Cao, X. A Predictive Maintenance Strategy for Multi-Component Systems Based on Components' Remaining Useful Life Prediction. *Mathematics* **2023**, *11*, 3884, doi:10.3390/math11183884.
7. Zhang, Y.; Guo, G.; Yang, F.; Zheng, Y.; Zhai, F. Prediction of Tool Remaining Useful Life Based on NHPP-WPHM. *Mathematics* **2023**, *11*, 1837, doi:10.3390/math11081837.
8. Si, X.-S.; Wang, W.; Hu, C.-H.; Zhou, D.-H. Remaining Useful Life Estimation – A Review on the Statistical Data Driven Approaches. *Eur. J. Oper. Res.* **2011**, *213*, 1–14, doi:10.1016/j.ejor.2010.11.018.
9. Li, W.; Lee, J.; Purl, J.; Greitzer, F.; Yousefi, B.; Laskey, K. *Experimental Investigation of Demographic Factors Related to Phishing Susceptibility*; 2020; ISBN 978-0-9981331-3-3.
10. Greitzer, F.L.; Li, W.; Laskey, K.B.; Lee, J.; Purl, J. Experimental Investigation of Technical and Human Factors Related to Phishing Susceptibility. *ACM Trans. Soc. Comput.* **2021**, *4*, 8:1-8:48, doi:10.1145/3461672.
11. Li, W.; Finsa, M.M.; Laskey, K.B.; Houser, P.; Douglas-Bate, R. Groundwater Level Prediction with Machine Learning to Support Sustainable Irrigation in Water Scarcity Regions. *Water* **2023**, *15*, 3473, doi:10.3390/w15193473.
12. Liu, W.; Zou, P.; Jiang, D.; Quan, X.; Dai, H. Computing River Discharge Using Water Surface Elevation Based on Deep Learning Networks. *Water* **2023**, *15*, 3759, doi:10.3390/w15213759.
13. Fan, Z.; Chang, K.; Raz, A.K.; Harvey, A.; Chen, G. Sensor Tasking for Space Situation Awareness: Combining Reinforcement Learning and Causality. In Proceedings of the 2023 IEEE Aerospace Conference; March 2023; pp. 1–9.
14. Salmaso, F.; Trisolini, M.; Colombo, C. A Machine Learning and Feature Engineering Approach for the Prediction of the Uncontrolled Re-Entry of Space Objects. *Aerospace* **2023**, *10*, 297, doi:10.3390/aerospace10030297.
15. Zhou, W. Condition State-Based Decision Making in Evolving Systems: Applications in Asset Management and Delivery. Ph.D., George Mason University: United States -- Virginia, 2023.
16. Ravi, C.; Tigga, A.; Reddy, G.T.; Hakak, S.; Alazab, M. Driver Identification Using Optimized Deep Learning Model in Smart Transportation. *ACM Trans. Internet Technol.* **2022**, *22*, 84:1-84:17, doi:10.1145/3412353.
17. Ordóñez, C.; Sánchez Lasheras, F.; Roca-Pardiñas, J.; Juez, F.J. de C. A Hybrid ARIMA-SVM Model for the Study of the Remaining Useful Life of Aircraft Engines. *J. Comput. Appl. Math.* **2019**, *346*, 184–191, doi:10.1016/j.cam.2018.07.008.
18. García Nieto, P.J.; García-Gonzalo, E.; Sánchez Lasheras, F.; de Cos Juez, F.J. Hybrid PSO-SVM-Based Method for Forecasting of the Remaining Useful Life for Aircraft Engines and Evaluation of Its Reliability. *Reliab. Eng. Syst. Saf.* **2015**, *138*, 219–231, doi:10.1016/j.ress.2015.02.001.
19. Benkedjouh, T.; Medjaher, K.; Zerhouni, N.; Rechak, S. Health Assessment and Life Prediction of Cutting Tools Based on Support Vector Regression. *J. Intell. Manuf.* **2015**, *26*, 213–223, doi:10.1007/s10845-013-0774-6.
20. Wang, H.; Li, D.; Li, D.; Liu, C.; Yang, X.; Zhu, G. Remaining Useful Life Prediction of Aircraft Turbofan Engine Based on Random Forest Feature Selection and Multi-Layer Perceptron. *Appl. Sci.* **2023**, *13*, 7186, doi:10.3390/app13127186.

21. Wang, X.; Huang, T.; Zhu, K.; Zhao, X. LSTM-Based Broad Learning System for Remaining Useful Life Prediction. *Mathematics* **2022**, *10*, 2066, doi:10.3390/math10122066.
22. Ensarioğlu, K.; İnkaya, T.; Emel, E. Remaining Useful Life Estimation of Turbofan Engines with Deep Learning Using Change-Point Detection Based Labeling and Feature Engineering. *Appl. Sci.* **2023**, *13*, 11893, doi:10.3390/app132111893.
23. Yuan, M.; Wu, Y.; Lin, L. Fault Diagnosis and Remaining Useful Life Estimation of Aero Engine Using LSTM Neural Network. In Proceedings of the 2016 IEEE International Conference on Aircraft Utility Systems (AUS); October 2016; pp. 135–140.
24. Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; Liu, Y. Remaining Useful Life Estimation of Engineered Systems Using Vanilla LSTM Neural Networks. *Neurocomputing* **2018**, *275*, 167–179, doi:10.1016/j.neucom.2017.05.063.
25. Zheng, S.; Ristovski, K.; Farahat, A.; Gupta, C. Long Short-Term Memory Network for Remaining Useful Life Estimation. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM); June 2017; pp. 88–95.
26. Li, X.; Ding, Q.; Sun, J.-Q. Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11, doi:10.1016/j.ress.2017.11.021.
27. Yu, W.; Kim, I.Y.; Mechefske, C. Remaining Useful Life Estimation Using a Bidirectional Recurrent Neural Network Based Autoencoder Scheme. *Mech. Syst. Signal Process.* **2019**, *129*, 764–780, doi:10.1016/j.ymssp.2019.05.005.
28. Zhao, C.; Huang, X.; Li, Y.; Yousaf Iqbal, M. A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction. *Sensors* **2020**, *20*, 7109, doi:10.3390/s20247109.
29. Peng, C.; Wu, J.; Wang, Q.; Gui, W.; Tang, Z. Remaining Useful Life Prediction Using Dual-Channel LSTM with Time Feature and Its Difference. *Entropy* **2022**, *24*, 1818, doi:10.3390/e24121818.
30. Wang, Y.; Zhao, Y. Multi-Scale Remaining Useful Life Prediction Using Long Short-Term Memory. *Sustainability* **2022**, *14*, 15667, doi:10.3390/su142315667.
31. Lyu, Y.; Zhang, Q.; Wen, Z.; Chen, A. Remaining Useful Life Prediction Based on Multi-Representation Domain Adaptation. *Mathematics* **2022**, *10*, 4647, doi:10.3390/math10244647.
32. Deng, F.; Bi, Y.; Liu, Y.; Yang, S. Deep-Learning-Based Remaining Useful Life Prediction Based on a Multi-Scale Dilated Convolution Network. *Mathematics* **2021**, *9*, 3035, doi:10.3390/math9233035.
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems; Curran Associates, Inc., 2017; Vol. 30.
34. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.; October 2 2020.
35. Mo, Y.; Wu, Q.; Li, X.; Huang, B. Remaining Useful Life Estimation via Transformer Encoder Enhanced by a Gated Convolutional Unit. *J. Intell. Manuf.* **2021**, *32*, 1997–2006, doi:10.1007/s10845-021-01750-x.
36. Ai, S.; Song, J.; Cai, G. Sequence-to-Sequence Remaining Useful Life Prediction of the Highly Maneuverable Unmanned Aerial Vehicle: A Multilevel Fusion Transformer Network Solution. *Mathematics* **2022**, *10*, 1733, doi:10.3390/math10101733.
37. Zhang, Z.; Song, W.; Li, Q. Dual-Aspect Self-Attention Based on Transformer for Remaining Useful Life Prediction. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11, doi:10.1109/TIM.2022.3160561.
38. Hu, Q.; Zhao, Y.; Ren, L. Novel Transformer-Based Fusion Models for Aero-Engine Remaining Useful Life Estimation. *IEEE Access* **2023**, *11*, 52668–52685, doi:10.1109/ACCESS.2023.3277730.
39. Li, X.; Li, J.; Zuo, L.; Zhu, L.; Shen, H.T. Domain Adaptive Remaining Useful Life Prediction With Transformer. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13, doi:10.1109/TIM.2022.3200667.
40. Ding, Y.; Jia, M. Convolutional Transformer: An Enhanced Attention Mechanism Architecture for Remaining Useful Life Estimation of Bearings. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10, doi:10.1109/TIM.2022.3181933.
41. Chadha, G.S.; Shah, S.R.B.; Schwung, A.; Ding, S.X. Shared Temporal Attention Transformer for Remaining Useful Lifetime Estimation. **2022**, *10*.
42. Zhang, Y.; Su, C.; Wu, J.; Liu, H.; Xie, M. Trend-Augmented and Temporal-Featured Transformer Network with Multi-Sensor Signals for Remaining Useful Life Prediction. *Reliab. Eng. Syst. Saf.* **2024**, *241*, 109662, doi:10.1016/j.ress.2023.109662.
43. Chen, D.; Hong, W.; Zhou, X. Transformer Network for Remaining Useful Life Prediction of Lithium-Ion Batteries. *IEEE Access* **2022**, *10*, 19621–19628, doi:10.1109/ACCESS.2022.3151975.
44. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.
45. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471, doi:10.1162/089976600300015015.

46. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the Proceedings of the 25th international conference on Machine learning; Association for Computing Machinery: New York, NY, USA, July 5 2008; pp. 1096–1103.
47. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management; October 2008; pp. 1–9.
48. Wu, Q.; Ding, K.; Huang, B. Approach for Fault Prognosis Using Recurrent Neural Network. *J. Intell. Manuf.* **2020**, *31*, 1621–1633, doi:10.1007/s10845-018-1428-5.
49. Sateesh Babu, G.; Zhao, P.; Li, X.-L. Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. In Proceedings of the Database Systems for Advanced Applications; Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H., Eds.; Springer International Publishing: Cham, 2016; pp. 214–228.
50. Wang, J.; Wen, G.; Yang, S.; Liu, Y. Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network. In Proceedings of the 2018 Prognostics and System Health Management Conference (PHM-Chongqing); October 2018; pp. 1037–1042.
51. Zhang, C.; Lim, P.; Qin, A.K.; Tan, K.C. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2306–2318, doi:10.1109/TNNLS.2016.2582798.
52. Kong, Z.; Cui, Y.; Xia, Z.; Lv, H. Convolution and Long Short-Term Memory Hybrid Deep Neural Networks for Remaining Useful Life Prognostics. *Appl. Sci.* **2019**, *9*, 4156, doi:10.3390/app9194156.
53. Mo, H.; Lucca, F.; Malacarne, J.; Iacca, G. Multi-Head CNN-LSTM with Prediction Error Analysis for Remaining Useful Life Prediction. In Proceedings of the 2020 27th Conference of Open Innovations Association (FRUCT); September 2020; pp. 164–171.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.