

Article

Data-Driven Deep Learning-Based Attention Mechanism for Remaining Useful Life Prediction: Case Study Application to Turbofan Engine Analysis

Amgad Muneer ^{1,2,*}, Shakirah Mohd Taib ^{1,2}, Sheraz Naseer ³, Rao Faizan Ali ³ and Izzatdin Abdul Aziz ^{1,2}

¹ Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32160, Malaysia; shakita@utp.edu.my (S.M.T); izzatdin@utp.edu.my (I.A.A.)

² Centre for Research in Data Science (CERDAS), Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia

³ Department of Computer Science, University of Management and Technology, Lahore 54728, Pakistan; sheraz.naseer@umt.edu.pk (S.N.); faizan.ali@umt.edu.pk (R.F.A.)

* Correspondence: amgad_20001929@utp.edu.my



Citation: Muneer, A.; Taib, S.M.; Naseer, S.; Ali, R.F.; Aziz, I.A. Data-Driven Deep Learning-Based Attention Mechanism for Remaining Useful Life Prediction: Case Study Application to Turbofan Engine Analysis. *Electronics* **2021**, *10*, 2453. <https://doi.org/10.3390/electronics10202453>

Academic Editors: Amir H. Gandomi, Fang Chen and Laith Abualigah

Received: 6 August 2021

Accepted: 30 September 2021

Published: 9 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Accurately predicting the remaining useful life (RUL) of the turbofan engine is of great significance for improving the reliability and safety of the engine system. Due to the high dimension and complex features of sensor data in RUL prediction, this paper proposes four data-driven prognostic models based on deep neural networks (DNNs) with an attention mechanism. To improve DNN feature extraction, data are prepared using a sliding time window technique. The raw data collected after normalizing is simply fed into the suggested network, requiring no prior knowledge of prognostics or signal processing and simplifying the proposed method's applicability. In order to verify the RUL prediction ability of the proposed DNN techniques, the C-MAPSS benchmark dataset of the turbofan engine system is validated. The experimental results showed that the developed long short-term memory (LSTM) model with attention mechanism achieved accurate RUL prediction in both scenarios with a high degree of robustness and generalization ability. Furthermore, the proposed model performance outperforms several state-of-the-art prognosis methods, where the LSTM-based model with attention mechanism achieved an RMSE of 12.87 and 11.23 for FD002 and FD003 subset of data, respectively.

Keywords: turbofan engine degradation; data-driven prognostic; deep neural network (DNN); prognostics and health management (PHM); remaining useful life (RUL); uncertainty

1. Introduction

Prognostics is a discipline of engineering that focuses on predicting a system's future state or behavior by using synthesis observations, calibrated mathematical models, and simulation [1]. It mainly refers to the research of predicting the precise period when a system will no longer be able to perform as intended. Prognostics is an attempt to estimate the remaining useful life (RUL) of a component in an engineering system. In many industries, rotating machinery is a critical component and is vulnerable to failure because of harsh working conditions and long operating hours [2]. Examples of the components of rotating machinery include gearboxes [3], motors, bearings [4], turbines [5], and engines [6]. To avoid critical damage and abrupt stopping of machine operation, rotating machinery failures should be detected as early as possible [7]. Failures can cause operational delays and enormous financial losses [8].

To avoid these failure scenarios, resource maintenance is frequently planned in advance [9]. However, in some industries, maintenance costs can account for up to 70% of total costs [10]. For this reason, maintenance cost reduction is viewed as a critical and significant advantage for manufacturers operating in a highly competitive manufacturing sector, such as power plant, aerospace, and oil and gas industries. Repairing an aircraft turbofan engine

following a breakdown can be more expensive than performing preventive maintenance in advance of a breakdown [11]. Therefore, industrial revolution has changed maintenance methods from preventive maintenance based on reliability assessment to condition-based maintenance (CBM) [12]. CBM strategy integrates real-time diagnosis of approaching failure with the prognosis of future equipment performance, making it easier to arrange essential repairs and maintenance in advance of a breakdown [11]. Numerous factors can complicate the maintenance operation of an aero engine system, including system configurations, maintenance resource costs, machine degradation profiles, maintenance schedules, and recent machine status [13].

Data-driven techniques, particularly those based on artificial intelligence (AI) such as deep learning (DL) [14–16], have gained increasing attraction in the manufacturing industry as the industrial Internet of Things (IoT) [17] and Big Data (BD) have grown in popularity. While considerable research has been conducted on the use of deep learning techniques relative to machine health monitoring, very few studies have focused on applying deep learning to the prediction of RUL with associated uncertainties [1,12,18]. Precise RUL prediction can considerably increase industrial components or systems' reliability and operational safety [19], prevent fatal failures, and lower maintenance costs [20]. Therefore, several attempts have been conducted in the literature to predict the RUL of a turbofan engine. However, these attempts still suffer from the disadvantages of high computational power [13], uncertainty prediction [21,22], and further architecture optimization is required [23] in order to provide high prediction accuracy because even a little uncertainty in prognostics prediction can result in huge losses [24]. Thus, there is a need for accurately predicting the RUL in practical aerospace applications [13] due to the presence of various uncertainties that affect prognostic calculations that, in turn, render turbofan engine RUL predictions uncertain [25]. Previous investigations [1,6,13,14,21,22,24] do not attempt to quantify the inherent uncertainty in their predictions. Hence, the research gaps identified by [25] still remain open in the context of RUL prediction in prognostics. While methods based on deep learning can achieve promising point prediction performance, they struggle to quantify the uncertainty associated with RUL prediction [24]. To fill these gaps, we have proposed a new data-driven approach that aims to accurately predict RUL and overcome the uncertainty inherent in DNNs predictions in the literature by incorporating the sliding time window technique for sample preparation and long short-term memory (LSTM) network with an attention mechanism to map the relationship between features and the RUL effectively. Then, we estimated and leveraged the uncertainties in DL models using bootstrap sampling.

Furthermore, it is challenging to estimate RUL for prognostics and health management (PHM) purposes [26–28]. Such RUL estimation is beneficial and one of the vital parts in PHM is that it can prevent unexpected failures of complex engineering systems [29] and help engineers to schedule maintenance, optimize operating efficiency, and avoid unplanned downtime [30]. Learning-based data-driven approaches have emerged as potential alternatives to model-based prognostic methods due to the complexity of the physics needed to properly simulate machine deterioration processes [31]. However, quantifying the remaining uncertainty for use in responding optimally to RUL predictions remains a research gap. Moreover, these data-driven models learn the degradation pattern entirely from previously stored historical data without incorporating physical models [29]. Therefore, this paper attempt to propose a new deep neural network (DNN) with an attention mechanism for machinery prognostics. The attention mechanism is an excellent method for enhancing the ability of models to learn new features. It may be thought of as a secondary screening of data information to emphasize the most critical pieces of information for analysis and precise prediction [32]. The primary contributions of this work are summarized as follows: (i) The suggested approach can quantify prognostic uncertainty by inferring the distribution across functions that map monitoring data to its associated RUL. (ii) In order to enhance the neural process's ability to predict RUL, an attention mechanism is integrated with one-layer LSTM to precisely capture critical

information included in the input signal and for which its performance in RUL prediction outperforms the traditional DL approaches in the literature. (iii) We have proposed a cost-effective approach to predict the remaining useful life of a turbofan engine where the parameters and computational cost of the training process are considerably decreased using dimensionality reduction processing. (iv) We have conducted a comparative analysis of four different deep neural network architectures in order to evaluate which technique relative to DNNs has excellent features extraction and generalization abilities. To verify the effectiveness of the proposed prognostic approach, the model was validated using Commercial Modular Aero Propulsion System Simulation datasets (C-MAPSS).

The rest of this paper is structured as follows. Section 2 highlights the background and the related works of the study. Section 3 outlines this work's research methodology, while Section 4 describes the experimental findings. Lastly, Section 5 concludes the paper and highlights future investigations.

2. Background and Related Work

Industrial internet advancements have enabled sensor data available from multiple machines across different domains and industries [17]. These sensor readings can determine the health of the equipment. As a result, there is a growing demand in the industry to perform maintenance on equipment based on their condition instead of the existing industry standard of time-based maintenance [33,34]. It has also been demonstrated that condition-based maintenance may save a lot of financial losses. As a result, developing these models can help achieve goals such as predicting the machine's RUL based on sensor data. RUL may be calculated by using historical trajectory data, which is useful for optimizing maintenance schedules in order to minimize engineering problems and reduce costs.

One of the data-driven methods used for RUL estimation is the convolutional neural network (CNN). CNN applications in RUL-related fields have also obtained massive attention in the literature [35]. Babu et al. [36] are the first to use the deep CNN approach for RUL prediction. The results showed that CNN outperformed multi-layer perceptron (MLP), support vector machine (SVM), and significance vector regression (SVR) models. The CNN approach proposed by [36] has been tested and evaluated on the C-MAPSS dataset. Wen et al. [37] have developed a new residual convolutional neural network (ResCNN). ResCNN employs the residual block, which uses shortcut connections to skip several blocks of convolutional layers and can aid in overcoming the vanishing gradient issue. Furthermore, ResCNN is improved by utilizing the k-fold ensemble technique. The proposed ensemble ResCNN is applied to NASA's C-MAPSS benchmark dataset. Another study proposed by [28] has presented a novel method for deep feature learning for RUL prediction using time-frequency representation (TFR) and multi-scale CNN networks (MS-CNN). TFR can effectively disclose the non-stationary nature of the bearing degradation signal.

Another practical approach is deploying a hybrid learning method. For instance, Li, Li, and He [38] have attempted to improve RUL estimations of the machines by suggesting a directed acyclic graph (DAG) network that combines CNN and LSTM to estimate the RUL. They reported that padding the signals in the same training batch will affect the prediction capacity of the integral approach when a single timestamp is utilized as an input. To address this limitation, the work in [38] generates a short-term sequence by sliding the time window (TW) with a single-phase duration. Furthermore, the piece-wise RUL technique is employed instead of the conventional linear function based on the degradation mechanism. Huang et al. [39] used the classic multi-layer perceptron (MLP) approach to predict the RUL of bearings in laboratory testing and found that the prediction results outperformed reliability-based alternatives. Muneer et al. [40] suggested an attention based DCNN model with time window approach to cope with the degradation and reliability of time-series forecasting problem. They performed a case study by estimating the RUL of turbofan engine on four subsets of C-MAPPS benchmark dataset. The main limitation of this work [40],

the uncertainty inherent in their DL model was not quantified. Khawaja, Vachtsevanos, and Wu [41] developed a confidence prediction NN method with a confidence distribution node to solve the downside that the confidence limits of RUL prediction cannot generally be explicitly acquired with NN techniques. Many researchers have also implemented fuzzy logic into MLP networks to collect more knowledge for PHM.

Malhi, Yan, and Gao [42] suggested applying competitive learning based on RNN methods to long-term prognostics of machine health status by using recurrent neural networks. The continuous wavelet transform is used to preprocess indicators of vibration from a rolling bearing with a fault, which is then used as model inputs. Yuan, Wu, and Lin [43] suggested an LSTM-based scheme for RUL prediction of aero engines in the case of highly complex operations, hybrid faults, and strong noises as an enhancement of the standard RNN. Zhao et al. [44] have used LSTM for tool wear health monitoring tasks. By integrating both the time domain and frequency domain functions, Ren et al. [45] proposed an optimized DL method for multi-bearing RUL collaborative estimation. The proposed method's feasibility and superiority were demonstrated by numerical tests on a real dataset. To fill the research gap, this research proposes a new DNN-based approach with an attention mechanism to predict the RUL of an aero engine system. Table 1 presents a summary of the state-of-art methods.

Table 1. Summary of the state-of-art methods findings.

Authors	Year	Technique Used	Benchmark Dataset	Results Achieved	Limitations/Gaps
Babu et al. [36]	2016	Deep CNN	C-MAPSS	RMSE for FD002 (30.29) RMSE for FD003 (19.81)	This study does not consider the uncertainty prediction associated with RUL.
Zhang et al. [21]	2016	Multi-objective deep belief networks ensemble	C-MAPSS	RMSE for FD002 (25.05) RMSE for FD003 (12.51)	This work does not consider the uncertainty prediction inherent in their DL model, which makes it impractical in practice.
Malhotra et al. [17]	2020	LSTM-ED	C-MAPSS	The authors have failed to report the RMSE value for each subset of data.	The model was not able to estimate the RUL and only able to capture the fault.
Li, Ding, and Sun, [22]	2018	DCNN	C-MAPSS	RMSE for FD002 (22.36) RMSE for FD003 (12.64)	This work does not consider the uncertainty prediction inherent in their DCNN model and further architecture optimization is still needed since the current training time is longer than most shallow networks in the literature.
Zheng et al. [15]	2017	Deep LSTM	C-MAPSS	RMSE for FD002 (24.49) RMSE for FD003 (16.18)	Uncertainty of LSTM model was not examined
Song et al. [46]	2019	Autoencoder-BLSTM	C-MAPSS	The model was only tested on a simpler subset of data called FD001 where the RMSE achieved is 13.63.	High computational load and uncertainty were not quantified.
Wen et al. [37]	2018	ResCNN	C-MAPSS	Authors have failed to report the RMSE value for each subset of data.	The limitations of the proposed method are that the imbalance of signal data is ignored and the tuning parameter process of the ensemble ResCNN is very time-consuming. Moreover, uncertainty was not predicted, which made this method impractical.
Li, Li, and He [38]	2019	CNN combined with LSTM	C-MAPSS	RMSE for FD002 (20.34) RMSE for FD003 (12.46)	The proposed DAG network suffers from a slow training time issue, which was reported to be (138.17 s), and the uncertainty was not quantified.
Muneer et al. [40]	2021	Attention-based DCNN	C-MAPSS	RMSE for FD002 (18.34) RMSE for FD003 (13.08)	The problem of RUL prediction with associated uncertainties of their DL model were not addressed.

Table 1. *Cont.*

Authors	Year	Technique Used	Benchmark Dataset	Results Achieved	Limitations/Gaps
Zhang et al. [47]	2019	CNN-XGB	C-MAPSS	RMSE for FD002 (19.61) RMSE for FD003 (13.01)	This method's main drawback is the computational speed, where the prediction time taken is around 621.7 s. It also does not consider the uncertainty prediction associated with RUL.
Ji et al. [48]	2020	PCA-BLSTM	C-MAPSS	The model was only tested on a simpler subset of data called FD003 where the RMSE achieved is 11.1.	Long training time where the researchers failed to indicate the time of training in seconds.
Wang et al. [49]	2021	MS-CNN	PRONOSTIA Bearing dataset	The MSE obtained with average test loss is (35.48) The R2 achieved with average test loss is (0.64)	This work does not consider the uncertainty prediction inherent in their DL model, which makes it impractical in practice
This study	2021	LSTM with Attention Mechanism	C-MAPSS	RMSE for FD002 (12.87) RMSE for FD003 (11.23)	Optimized network structure where the parameters and computational cost of the training process are considerably decreased using dimensionality reduction processing. The prediction time taken is around 117.3 s, and the uncertainty in DNNs predictions is examined.

3. Development of Data-Driven Model

The approach of this study provides a comparative analysis among four promising and well-known deep learning models. To achieve the optimum model for predicting RUL of different engine units, the proposed DNN-based models with attention mechanism were trained and evaluated, applying standard performance evaluators for prediction models. The first section focuses on the proposed deep candidate models, while the final two steps of the suggested methodology are discussed in subsequent sections.

3.1. Candidate Model Training and Optimization

This section focuses on describing the DNNs architecture and optimization utilized to develop RUL prediction candidate models. This study has employed commonly used neural network architectures such as “Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) with simple units, Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) units,” respectively. For DNN optimization, we applied the Randomized Hyperparameter search methodology employed in [50] to maximize the effectiveness of DNN candidate models. By performing a random search over a vast hyperparameter space, the randomized hyperparameter search provides improved hyperparameters for DNNs with a limited computing budget. The hyperparameters are randomly sampled, and models are created by using the parameters that are evaluated. The following subsections present a brief overview of each DNN architecture that is utilized to predict the RUL of the turbofan engine.

3.1.1. Recurrent Neural Networks

A shortcoming of traditional DNNs is that the weights learned by individual neurons preclude them from identifying exact representations for the corresponding features to RUL due to the complex structure of the system. An RNN circumvents the restriction via utilizing a repeating loop over timesteps to resolve the problem mentioned above. A sequence vector $\{x_1, \dots, x_n\}$ is manipulated by utilizing a recurrence of the form $r_t = f_\alpha(r_{t-1}, x_t)$, where f denotes the function of activation, α is a set of parameters applied at each time step t , and x_t is the input at timestep t .

For instance, three variations of recurrent neurons, a simple RNN unit, a gated recurring unit (GRU), and the LSTM unit, are used to develop the candidate RNN-based models for the proposed study. Each timestep is a simple recurrent neuron. The parameters governing the connections between the input and hidden layers and the horizontal connection between activations and the hidden layer to the output layer are shared. A basic recurrent neuron's forward pass can be expressed as follows:

$$a^t = g(W_a [a^{(t-1)}, X^t] + b_a) \quad (1)$$

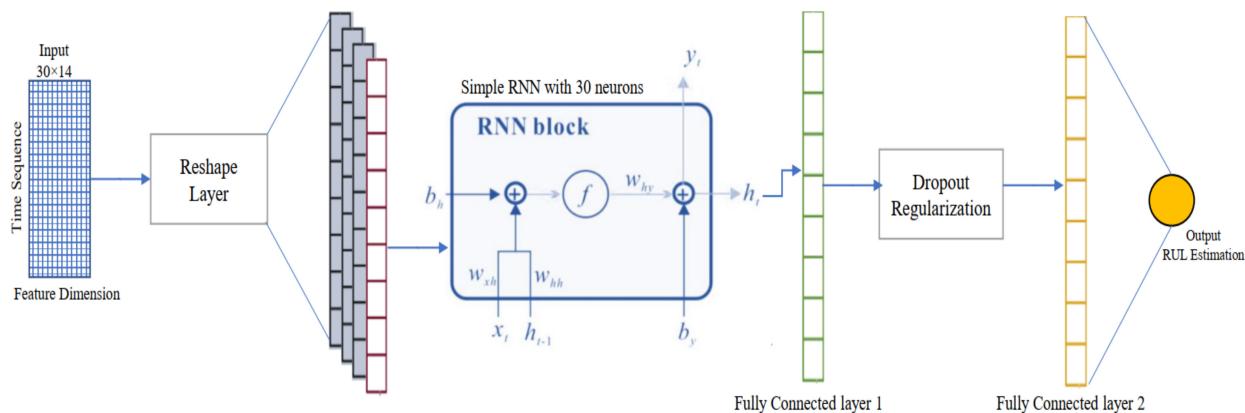
$$y^t = f(W_y a^t + b_y) \quad (2)$$

where g reflects an activation function when “ t ” represents the current timestep, the input of timestep is represented by X^t , b_a and b_y defines the bias, W_a and W_b are the respective weights, and timestep t of the activation output is denoted by a^t . If needed, this a^t activation could be employed to measure the y_t forecasts at time t .

Table 2 demonstrates the RNN method structural design with the simple RNN neurons. This model uses a reshape layer to convert sequence input to time-series input. The following layers of the DNN model interpreted these sequence vectors' geometric relationships to learn deep feature representations and evaluated them by the output layer in order to render predictions; a single linear unit is used. Figure 1 presents the proposed simple RNN-based model structure for RUL prediction.

Table 2. Simple RNN-based model architecture for RUL estimation of turbofan engine.

Layer Type	No. of Weights
Reshape layer (sequence to timeseries conversion)	No weights $35 \times 30 = 1050$
Simple RNN with 30 neurons	$(30 + 1) \times 20 = 620$
FC1 with 20 relu units	No weights
Dropout layer for regularization	$(20 + 1) \times 8 = 168$
FC2 with 8 relu units	$(8 + 1) \times 1 = 9$
Output Layer with One linear unit	

**Figure 1.** Proposed simple RNN-based model structure for RUL estimation.

Additionally, DNNs with simple RNN neurons indicate favorable outcomes in several applications. They are still prone to vanishing gradients problem [51] and show a limited capability to learn long-term dependencies of engine system degradation. The research community has provided a number of modified recurrent neuron architectures to overcome the simple RNN neurons drawback, including the Gated Recurrent Unit (GRU) technique proposed by [52] and the LSTM method presented by [53] in order to resolve the problem of gradients disappearing and to allow long-term dependences to be learned.

The authors in [54] presented GRU, which is capable of showing better performance for long-term relationship learning in time series data. The memory variable $H^t = a^t$ is used by the GRU unit at each stage t , which provides an updated list of all samples processed by the unit. Hence, the GRU unit considers overwriting the H^t at each timestep t , but the regulation of memory variable overwriting is implemented via the update gate Γ_u when the GRU unit superimposes the H^t value at each step “ t ” with the candidate value \bar{H}^t . GRU neuron functionality can be represented via the following series of equations:

$$\bar{H}^t = \tanh(W_c [\Gamma_r \times H^t, X^t] + b_c) \quad (3)$$

$$\Gamma_r = \sigma(W_r [H^{t-1}, X^t] + b_r) \quad (4)$$

$$\Gamma_u = \sigma(W_u [H^{t-1}, X^t] + b_u) \quad (5)$$

$$H^t = \Gamma_u * \bar{H}^t + (1 - \Gamma_u) * H^{t-1} \quad (6)$$

$$a^t = H^t \quad (7)$$

where W_r , W_c , and W_u represent the respective weights, and b_r , b_c , and b_u denote the subsequent bias terms for input X_t at timestep t . σ is the function of logistic regression, and the activation value at timestep t is represented by a^t . With the exception of the usage of GRU neurons, the implemented RNN model developed with GRU is similar

to that of simple RNNs. Table 3 presents the GRU-based RNN model architecture for RUL estimation.

Table 3. GRU-based RNN model architecture for RUL estimation in C-MAPSS benchmark dataset.

Layer Type	No. of Weights
Reshape layer (sequence to timeseries conversion)	No weights
GRU layer with 30 neurons	$78 \times 30 = 2340$
FC1 with 20 relu units	$(30 + 1) \times 20 = 620$
Dropout layer for regularization	No weights
FC2 with 8 relu units	$(20 + 1) \times 8 = 168$
Output Layer with One linear unit	$(8+1) \times 1 = 9$

As mentioned earlier, the authors in [53] have proposed the LSTM neuron with some improvements to the design of the Simple_RNN unit, which provides a more robust generalization of GRU. Prominent variations in LSTM and GRU cells are illustrated as follows:

1. No significance gate Γ_r is used in generic LSTM units for \bar{H}^t computation.
2. LSTM units utilize two distinct gates instead of an update gate Γ_u , namely output gate Γ_o and update gate Γ_u . The output gate tracks the content's visibility of the H^t memory cell to compute LSTM unit activation outputs for other hidden units in the network. To achieve H^t , the forget gate handles the extent of overwriting on H^{t-1} , for instance, how much memory cell information must be overlooked in order to function properly for memory cells.
3. LSTM is different from GRU architectures by the fact that the memory cell contents H^t may not be equivalent to the activation a^t at time t .

Moreover, the RNN approach-based LSTM is constructed with similar architecture as GRU and simple RNN models. The only difference is the presence of LSTM units in recurrent layers instead of GRU units. Both RNN networks, LSTM-based and GRU-based, are combined with an attention mechanism. Table 4 and Figure 2 show the model's architecture that used LSTM with an attention mechanism to build the RUL estimation model.

Table 4. RUL estimation using RNN-based LSTM neurons architecture.

Layer Type	No. of Weights
Reshape layer (sequence to timeseries conversion)	No weights
LSTM layer with 30 neurons	$144 \times 30 = 4320$
FC1 with 20 relu units	$(30 + 1) \times 20 = 620$
Dropout layer for regularization	No weights
FC2 with 8 relu units	$(20 + 1) \times 8 = 168$
Output Layer with linear unit	$(8+1) \times 1 = 9$

The integrated attention mechanism is utilized to maintain the intermediate output via the LSTM encoder. Thus, the suggested model is trained to extract usable representations from the input sequence and to correlate output and input sequences. As illustrated in Figure 2, the attention mechanism determines the score value for each variable using the LSTM hidden layers' intermediate variables. Each variable's weight value indicates its relative importance. Then, the context vector layer will combine the data and extract critical feature information from the input sequence, with the critical feature information offered greater weight. Moreover, by monitoring the mapping relation in multi-dimensional space, additional hidden layers provide a more precise distribution among network input and output, improving the model's capacity to deal with nonlinear complex features.

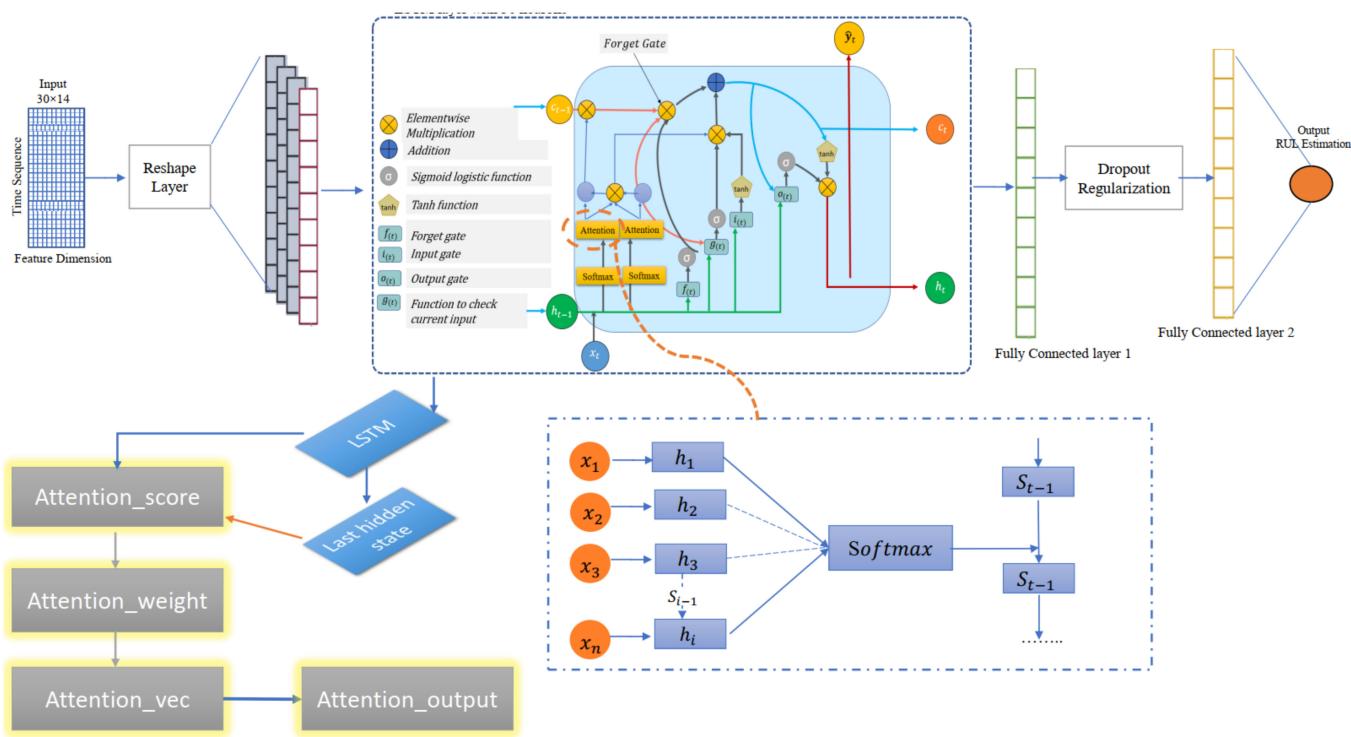


Figure 2. Proposed RNN-based LSTM with attention mechanism model structure for RUL estimation.

The listed formulas (8–12) illustrate the update rules for each parameter following the use of LSTM and the attention mechanism:

$$\begin{pmatrix} f_{(t)}^n \\ i_{(t)}^n \\ o_{(t)}^n \\ g_{(t)}^n \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} Z_{L+M}^n \begin{pmatrix} Ed_{t-1}^n \\ h_{t-1}^n \\ v_t \end{pmatrix} \quad (8)$$

$$v_t = \sum_i^N \beta_i^t x_i \quad (9)$$

$$s_i^t = W_s \tanh(W_h h_{(t-1)}^n + W_x x_i + b_s) \quad (10)$$

$$\beta_t^i = \frac{\exp(s_k^t)}{\sum_{k=1}^N \exp(s_k^t)} \quad (11)$$

$$\sum_{i=1}^N \beta_t^i = 1 \quad (12)$$

where Z_{L+M}^n indicates the LSTM model's n th layer parameters, and σ is the sigmoid function. Ed_{t-1}^n denotes an embedding matrix, and L denotes the LSTM network's dimension. v_t is the context vector, which is a representation of the input's relevant vector at time t . M is the dimension of v_t , and β_t^i denotes the attention weights assigned by the relevance score s_i^t . Once the relevant scores for the attributes $X = \{x_1, x_2, \dots, x_N\}$ are computed, the network is capable of obtaining the attention weights.

By amplifying the weights, the neural network may concentrate on learning the related data. In short, the weight matrices utilized in the suggested LSTM-based model at time point t are as follows:

$$w'_{t,ih} = w_{i,ih} \beta_t \quad (13)$$

$$w'_{i,oh} = w_{i,oh} \beta_t \quad (14)$$

$$w'_{i,f_h} = w_{i,f_h} \beta_i \quad (15)$$

where the following is the case.

$$\beta_i = \begin{bmatrix} 1 + \beta_{t,1} & 1 + \beta_{t,1} & \dots & 1 + \beta_{t',1} \\ 1 + \beta_{t,2} & 1 + \beta_{t,2} & \dots & 1 + \beta_{t',2} \\ \vdots & \vdots & & \vdots \\ 1 + \beta_{t,m} & 1 + \beta_{t',m} & \dots & 1 + \beta_{t',m} \end{bmatrix}_{m \times m}$$

Finally, the process of the LSTM with an attention mechanism is shown in Figure 2 (bottom).

3.1.2. Convolutional Neural Networks

CNNs are designed to handle learning problems involving high dimensional input data with complex spatial structures such as image classification [55], pattern recognition [56], amino acid sequence prediction [54,57], and time series failure signals. CNNs attempt to learn hierarchical filters that can transform large input data to accurate class labels using minimal trainable parameters. This is accomplished by enabling sparse interactions between input data and trainable parameters by using parameter sharing in order to learn equivariant representations (also called feature maps) of complex and spatially structured input information [58]. In a Deep CNN, units in the deeper layers may indirectly interact with a large portion of input due to the usage of pooling operations that replaces the output of Net at a certain location with a summary statistic and allows the network to learn complex features from this compressed representation [49]. The so-called “top” of the CNN is usually composed of a bunch of fully connected layers, including the output layer, which uses the complex features learned by previous layers to make predictions.

The CNN-based architecture of the RUL prediction model is presented in Table 5 with one reshape layer, two convolution layers with max pool blocks, flatten layer, fully connected layer, and an output layer of the sigmoid neuron. The CNN-based approach has been proposed.

Table 5. RUL prediction using CNN-based neurons architecture.

Layer Type	No. of Weights
Reshape layer	No weights
Conv-1D with 8 kernels of size 3	$((3 \times 2) + 1) \times 8 = 56$
Maxpool-1D	No weights
Regularization with 50% of probability	No weights
Conv-1D with 16 kernels of size 3	$((3 \times 8) + 1) \times 16 = 400$
Maxpool-1D	No weights
Flatten (to create array of scalars)	No weights
FC1 with 8 relu units	$(32 + 1) \times 8 = 264$
Output layer comprising of one sigmoid	$(8 + 1) \times 1 = 9$

The deep CNN learning ability is enhanced by utilizing multiple non-linear feature extractions. It can learn hierarchical representations from data on its own. As a result, the scale of the convolution kernel and the number of convolution layers significantly affect prediction performance. The proposed CNN architecture for RUL prediction in this research study is presented in Figure 3, and the attention mechanism employed is similar to the one shown in Figure 2 (bottom). Therefore, the input datum is two-dimensional (2D), where the feature number takes up one dimension, and the sensor’s time sequence is the other dimension.

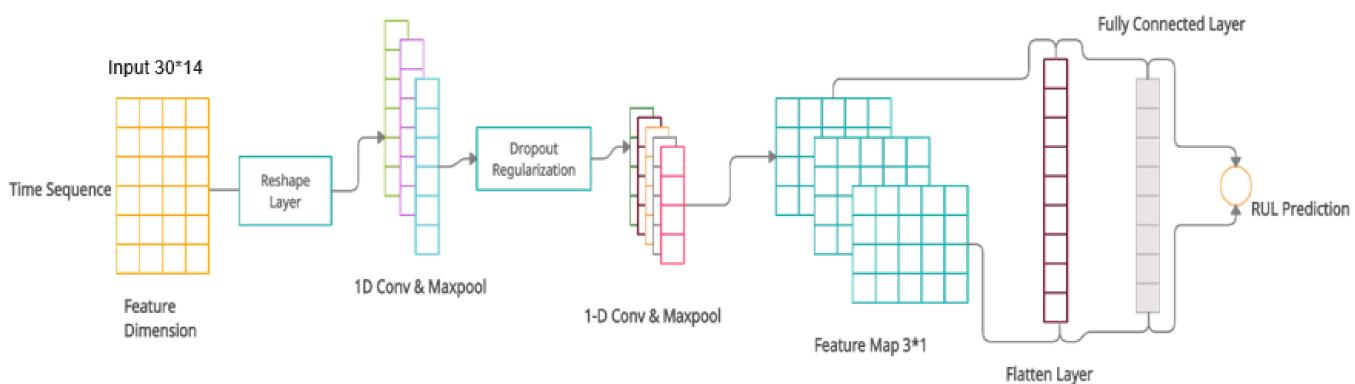


Figure 3. Proposed deep CNN structure for RUL prediction.

Following that, the reshape layer is employed for sequence to time-series conversion of the input data. The feature maps are then combined using a convolutional layer with one filter. After flattening, the feature will be connected with a fully connected layer. Moreover, the dropout method will be used to relieve overfitting. Additionally, RELU is the activation feature of each layer. In this study, Adam's optimization algorithms will serve as the optimizer. Due to the current state of the aero engine system datasets, we increased the penalty for lag prediction; the loss is expressed as given below:

$$\text{loss} = \frac{1}{N} \sum_{i=1}^N \omega(y_i - \hat{y}_i) \quad (16)$$

where y_i is the actual value, and \hat{y}_i is the predicted value. N is the validations' set number. When the actual value y_i is greater than the predicted value \hat{y}_i , the penalty coefficient $\omega = 1$, otherwise $\omega = 2$.

3.2. Prognostic Procedure

Figure 4 depicts the proposed prognostic experimental approach. To begin, the FD002 and FD003 subsets of data are preprocessed by selecting fourteen raw sensor measurements (discussed in Section 3.3.1) and normalize the accompanying data to fall within the range $[-1, 1]$. Next, the training and testing datasets are created, with each sample providing information about the time sequence within the N_{tw} length time frame. Hence, the normalized data that are prepared in 2D format have been directly fed into the DNNs models as input. There is no requirement for hand-crafted signal processing features such as skewness, kurtosis, and so on. As a result, the suggested method does not require prior knowledge of prognostics or signal processing. Following that, the suggested DNNs candidate models for RUL prediction are constructed, and its configuration is specified, including the number of hidden layers, the number and length of convolution filters, etc. Finally, the DNNs models receive normalized training data as input and output labelled RUL values for the training samples.

Additionally, back-propagation learning is employed in order to update the network's weights. For updates, the Adam optimization method is employed in conjunction with mini-batches. The samples are randomly separated into numerous mini-batches of 512 samples, each for each training epoch and loaded into the training DNN model. Following that, network information, for example, the weights in each layer, is optimized by using the mini-mean batch's loss function. It should be mentioned that batch size selection influences the performance of network training [59]. According to the experimental trials, the batch size of 512 samples was determined to be appropriate and is utilized in all of the case studies in this study. Additionally, a random search strategy is applied to provide optimal performance on a given subset of data (i.e., FD002) over a large hyperparameter space.

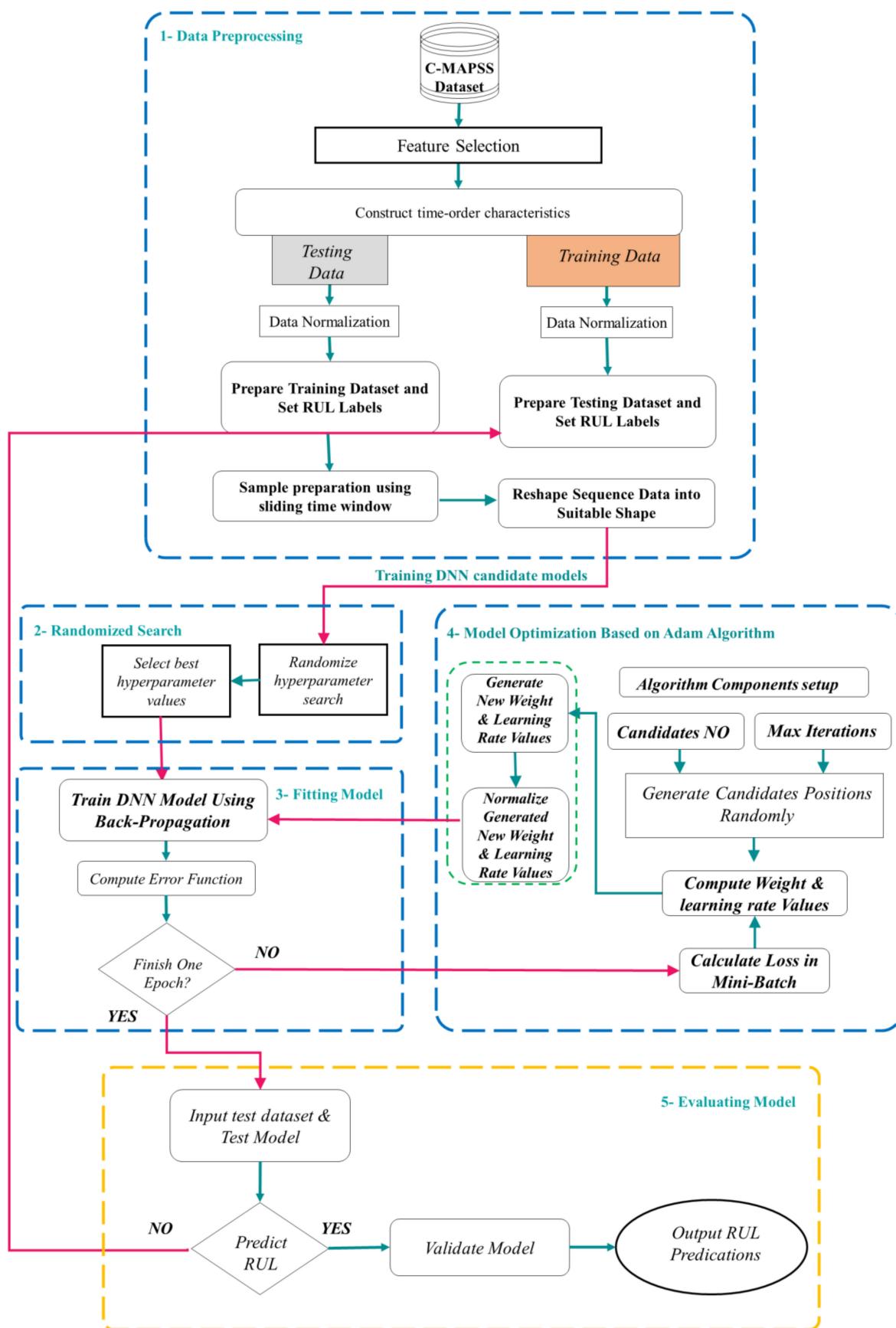


Figure 4. The prediction process of our proposed DNNs candidate models.

3.3. Data Pre-Processing and Normalization

3.3.1. Sensor Data Selection

The C-MAPSS dataset contains 21 sensors/features. Therefore, not all the sensors/features are highly correlated with RUL prediction. Hence, we have used gradient boost decision trees (GBDTs) for feature selection and dimensionality reduction in order to assess the association between sensor data and RUL values in order to identify significant features.

The GBDT method is a type of ensemble learning method that uses the decision tree as its primary estimator. It operates admirably when utilized for feature selection and can output the characteristics' relative value. The results are ranked in order of significance, and the top fourteen sensor measurements are chosen for further analysis. Hence, among 21 sensors only fourteen sensors/features are selected to be fit into the DNN candidate models.

3.3.2. Data Normalization

Several raw sensor data, operating parameters, and run to failure are granted in real-world applications. Sensor data must be standardized about each sensor before training and testing since the value scale of different sensors can differ. In this experiment, 21 sensors are employed, and the irregular or unchanged sensor data are eliminated. The min-max normalization technique is used to normalize each feature range to be [0, 1]. Additionally, for certain systems where the system health does not decay linearly from the operations start, we must use piece-wise functions to reduce the estimated RUL_{calc}^t goal. Different workloads, operational environments, and deterioration modes exist in certain applications, and we can incorporate this knowledge into the RUL estimation model if it is accessible.

For data normalization, sensor data can be normalized by letting μ_i denote the mean of the i -th sensor data from the engine, and σ_i denotes standard deviation. Z-score normalization is where x'_i is the normalized sensor output. The raw sensor data are scaled within the range of [0, 1] by using normalization of min-max as described in the following:

$$x'_i = \frac{x_i - \mu_i}{\sigma_i} \quad (17)$$

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i} \quad (18)$$

where $x^{i,j}$ is the i -th measuring point of the j -th sensor. $x^{i,j}_{norm}$ is the $x^{i,j}$ normalized result. x_{\max}^j and x_{\min}^j represent the minimum and maximum values of the j -th sensor.

3.4. Samples Preparation Using Sliding Time Window

The sliding window (SW) approach [60] is used to segment data, as illustrated in Figure 5. The entire lifespan of an engine is T , and the window and step sizes are specified individually as l and m . The i th sample has an input size of $l \times n$, where n is the selected sensor's total number and the operating condition (OC) data dimension, and the real RUL is $T_s - l - (i - 1) \times m$. For both training datasets FD0002 and FD003, the sliding window length is selected at 30 based on the experiments conducted with various window sizes (e.g., 5, 10, 15, 20, 25, 30, 35, etc.), and it was discovered that there was a considerable reduction in RUL estimation error when the sliding window size increased from 15 to 30. When the sliding window is greater than 30, no noticeable improvement in prognostic performance occurs. All the historical data in the SW are extracted at each time step to establish a high dimensional vector of length 14×30 as input data. Thus, based on feature selection and the dimensionality reduction method, 14 sensors measurements out of 21 sensors are selected-employed as the raw input features. These 14 sensors were also utilized by [23,61]. The dynamic characteristics of aero engine operating data under various operating conditions are significantly different, which results in different network

structures for extracting features. The structure of the proposed DNN-based models in this paper is designed to predict the RUL of aero engines under single and multiple operating conditions. Therefore, this paper utilizes the FD002 dataset obtained under six OCs and the FD003 dataset obtained under a single OC of the aero engines for experimental analysis.

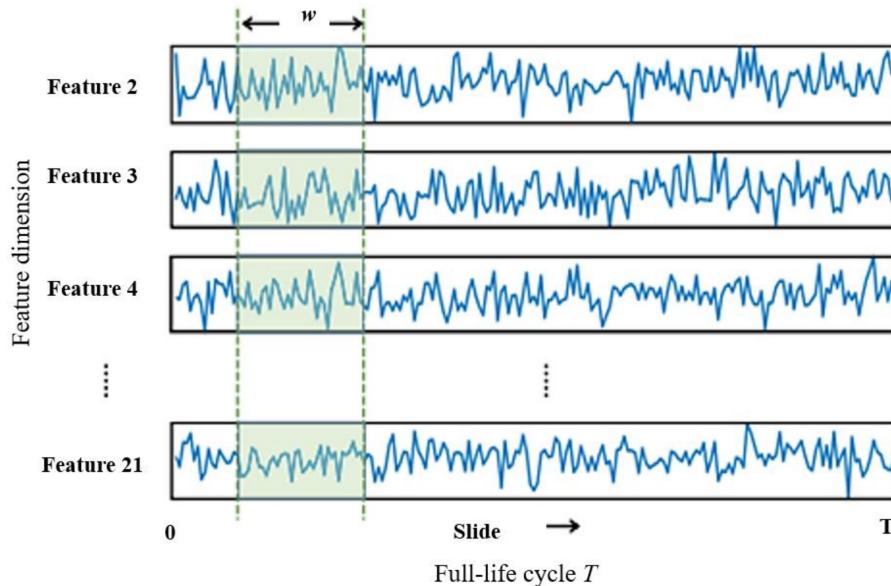


Figure 5. Schematic diagram of the sliding window approach.

Additionally, a sequence to target the problem can be expressed as RUL estimates. A time series of length T is $X = (x_t \mid t = 1, \dots, T)$ with $x_t \in R^{n \times m}$, where n is the sensors' number, and m the samples number per cycle. Hence, the goal is to estimate the corresponding output $y_T, y_T = f(x_t \mid t = 1, \dots, T)$, where x_t represents all samples at cycle t . When the sliding window technique is employed, the earlier formulation must be revised to $y_T = f(x_t^w \mid t = w, \dots, T)$, where w is the window size. Thus, the vector x_t^w contains all the samples in the TW, which is denoted as $x_t^w = (x_{t-w+1}, \dots, x_t)$. In our settings, the size of the sliding step is always set to one. The sequence length specifies how historical data are employed in any model, whereas the window size describes the complexity of dynamic features across time. Both parameters should be examined when optimizing the model because they have such a significant impact on its performance.

For the RUL target label, a piecewise linear function has been used [21], which is defined as the following:

$$Rul = \begin{cases} Rul, & \text{if } Rul \leq Rul_{\max} \\ Rul_{\max}, & \text{if } Rul > Rul_{\max} \end{cases} \quad (19)$$

where the following is the case.

Rul_{\max} is a preset value

Therefore, Rul_{\max} is set to 130 and 150 cycles, respectively, for FD002 and FD003 as in [21,38]. According to experimental analysis, m is 30, and l is 1. FD002 and FD003 have training samples of 53,759 and 24,720, respectively, and testing samples of 259 and 100, respectively, because only the most recent measurements of the test sets are used. The effectiveness of the piecewise linear function on this prediction problem has been confirmed in the literature [23,36,61], and the processed label values are smoothed.

4. Experimental Results and Discussion

This section summarizes the experiments' findings and discusses their significance. In the first section, Commercial Modular Aero Propulsion System Simulation datasets

(C-MAPSS) are introduced. In the second section, experimental results and performance analysis are introduced. The comparative analysis with literature is provided in the last section.

4.1. C-MAPSS Benchmark Dataset

This paper selects the C-MAPSS turbofan degradation dataset provided by NASA to verify the effectiveness of the proposed DNNs candidate models. The primary control system comprises three components: a fan controller, a regulator, and a limiter. The fan maintains normal flight conditions by directing air into the inner and outer culverts, as observed in Figure 6. The combustor is supplied with compressed high temperature, high-pressure gases via a low-pressure compressor, and a high-pressure compressor. Low-pressure turbines can be used to decelerate and pressurize air, hence, increasing the chemical energy conversion efficiency of aviation kerosene. High-pressure turbines generate mechanical energy by striking turbine blades with high temperatures and high-pressure gas.

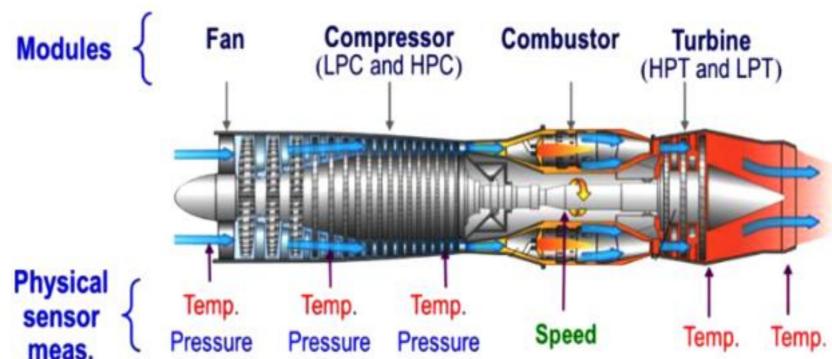


Figure 6. Illustration diagram of the turbofan engine [62].

The C-MAPSSs are extensively utilized in state-of-the-art prognostic studies, which contains four sub-datasets of the engine under different operating conditions and failure modes. Every subset of data contains a training set, testing set, and true RUL values and they consist of 21 sensors and 3 operation settings [39]. Every engine unit has varying degrees of wear. With time, the engine units begin to degrade until they reach system failure, which is described as an unhealthy time cycle. Therefore, the sensor records in the testing set are terminated before system fault. The dataset is collected in the form of a compressed text file. Each row represents a snapshot of the data collected during a single operation cycle, whereas each column represents a unique variable. The specific details of the dataset and the prognostic problem are provided in Table 6.

Table 6. Description of C-MAPSS benchmark dataset.

Dataset	C-MAPSS			
	FD001	FD002	FD003	FD004
Engine units for training	100	260	100	249
Engine units for testing	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

The experiment aims to predict the RUL of a single-engine unit randomly selected from the testing set. The second and third subsets of data, FD002 and FD003, are used for DNN model verifications in this research study. FD002 has 53,759 training samples and 259 test samples. FD003 has 24,720 training samples and 100 test samples. It also has the simplest operating conditions (OC) and the simplest fault type (FT), and FD002 is a more complicated subset and has six OCs and one FTs compared to the FD002 subset of data.

4.2. Experimental Results and Candidate Models Performance Analysis

This section aims to evaluate and validate our different DNN candidate models proposed for turbofan engine RUL estimation. For better prognostic characteristics analysis, the actual and predicted RULs of testing units are compared in Figure 7, where testing units are sorted in an ascending order by actual RUL. The predicted RUL values for the suggested LSTM-based attention mechanism architecture are comparable to the actual RUL in two random cases (engine 24 and engine 41; randomly selected), as shown in Figure 7. The second evaluation test is conducted for the GRU-based attention candidate model. The results presented in Figure 8 show that the model was able to predict the RUL for two random cases (engine 26 and engine 41). However, the RUL case of engine 41 was randomly predicted by using LSTM-based attention mechanism and GRU-based models. Hence, the LSTM-based attention mechanism model show better prediction of RUL engine 41, as shown in Figures 7, 8, 9 and 10b, respectively.

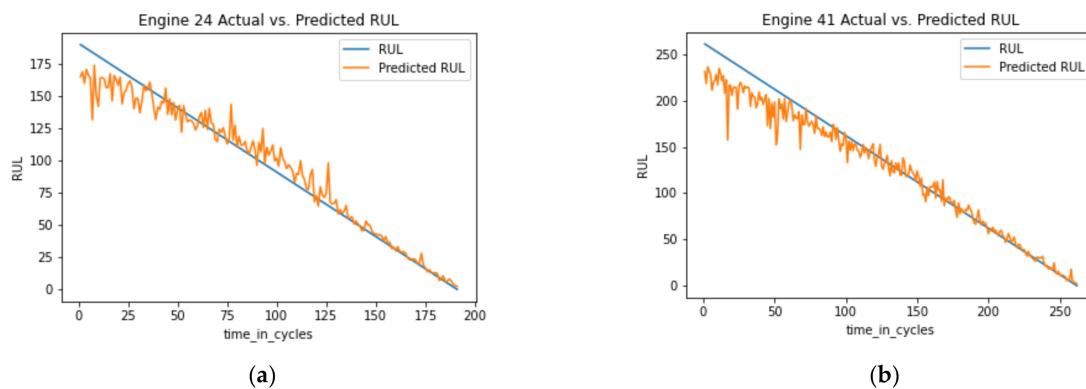


Figure 7. RUL prediction results of FD002 using the proposed LSTM with the attention mechanism model of the first verification test: (a) RUL prediction for the first randomly selected case (engine #24); (b) RUL prediction for the second randomly selected case (engine #41).

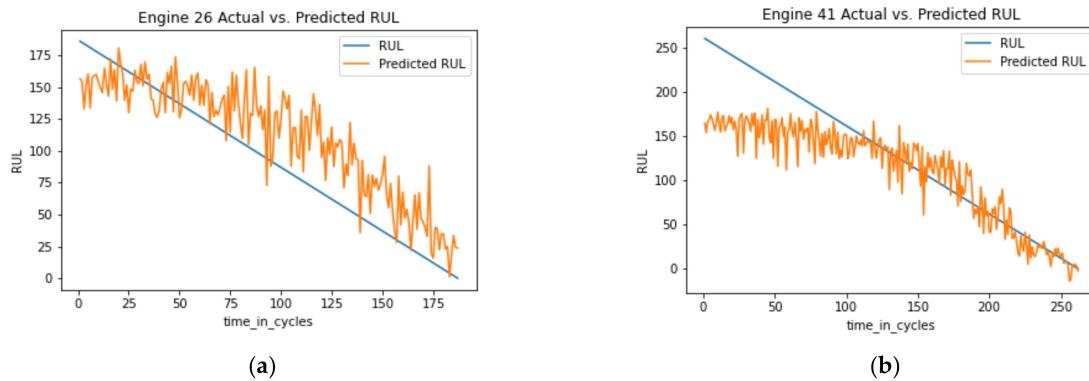


Figure 8. RUL prediction results of FD002 using the proposed GRU architecture of the second verification test: (a) RUL prediction for the first randomly selected case (engine #26); (b) RUL prediction for the second randomly selected case (engine #41).

The third evaluation test is conducted for S_RNN candidate model architecture where the RUL estimation in both cases was not accurate as of the other two RNN methods, LSTM and GRU. The S_RNN-based model performs the worse, as it can be observed in both cases (Figure 9). Lastly, the fourth evaluation test is performed for the proposed CNN candidate model, where the RUL prediction results were poor, as shown in Figure 10. The CNN model was not able to capture the degradation of the aero engine accurately. Thus, the LSTM-based model outperformed CNN, GRU, and S_RNN models in both verification cases in this study.

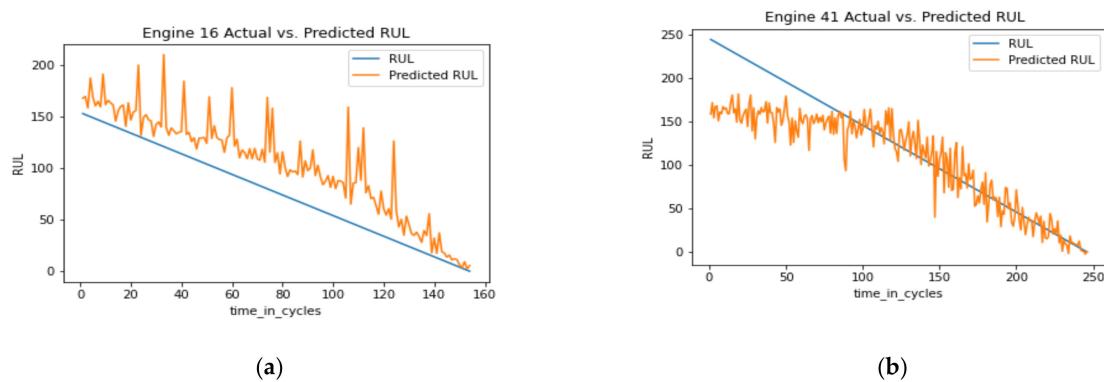


Figure 9. RUL prediction results of FD002 using the proposed S_RNN architecture of the third verification test: (a) RUL prediction for the first randomly selected case (engine #16); (b) RUL prediction for the second randomly selected case (engine #41).

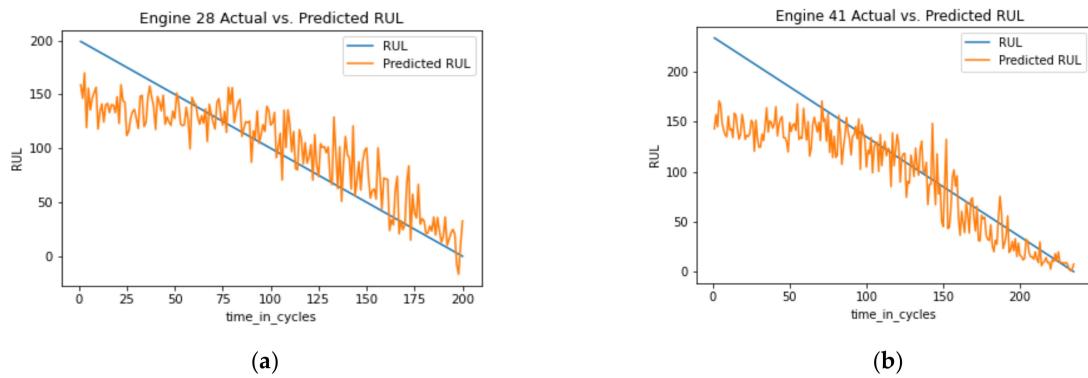


Figure 10. RUL prediction results of FD002 using the proposed CNN architecture of the fourth verification test: (a) RUL prediction for the first randomly selected case (engine #28); (b) RUL prediction for the second randomly selected case (engine #41).

For the evaluation of regression problems, residual analysis is another well-known evaluation metric that is used to assess the appropriateness of the model by defining residuals and examining residual plots. A residual e is defined as the difference between the observed value of a dependent variable “ y ” and its predicted value \bar{y} . For the i th datapoint, the residual e_i is given as follows.

$$e_i = \bar{y}_i - y_i \quad (20)$$

The residual analysis depicts the bias of the model towards underestimation or overestimation. A regression model is said to be overestimating if $e_i > 0$, which means the predicted values \bar{y}_i are persistently greater than the actual scores y_i . Contrariwise, a model is said to be underestimating if $e_i < 0$; i.e., the model is persistently predicting \bar{y}_i to be smaller than y_i . The residual plots provide insights into the model by illuminating the patterns of under/overestimation. A better regression model is expected to balance overestimation and underestimation, which means data distributions of residuals e_i are symmetric around the residual point zero with an approximately similar number of samples above and below zero. The residual analysis of models can be highlighted by using the scatter plot and distribution plot showing the distribution of e_i around zero. The more skewed the distribution of residuals is, the more biased the model is and vice versa. Figures 11 and 12 show the residual analysis of LSTM-based model for the test data of FD002 and FD003.

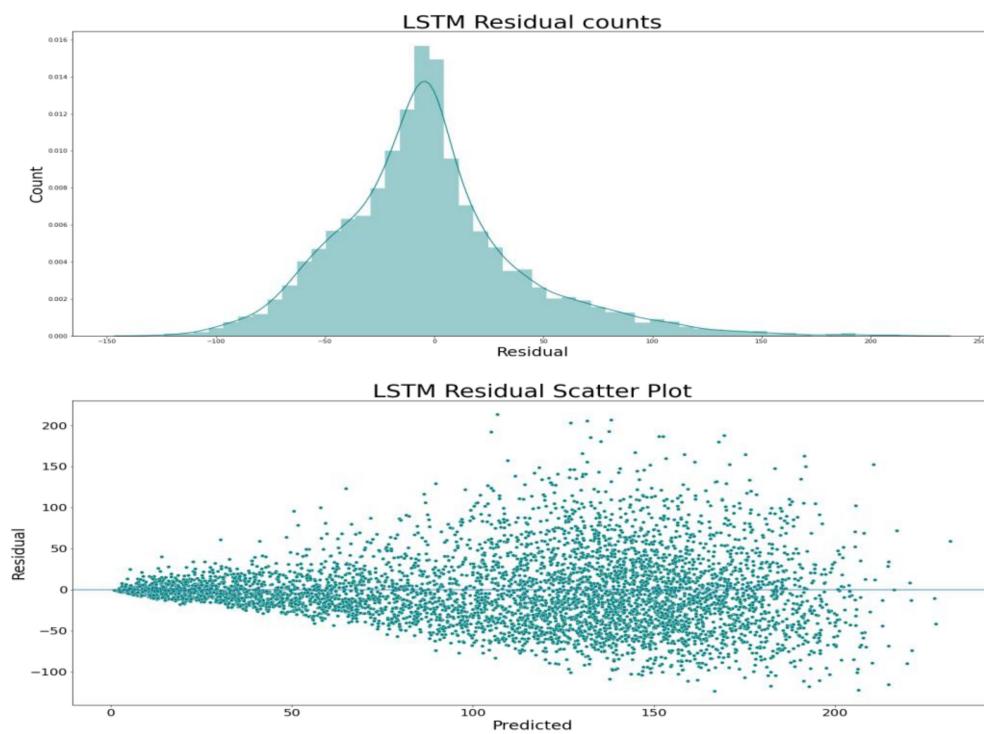


Figure 11. Residual analysis of LSTM with attention mechanism-based model on FD002 test dataset.

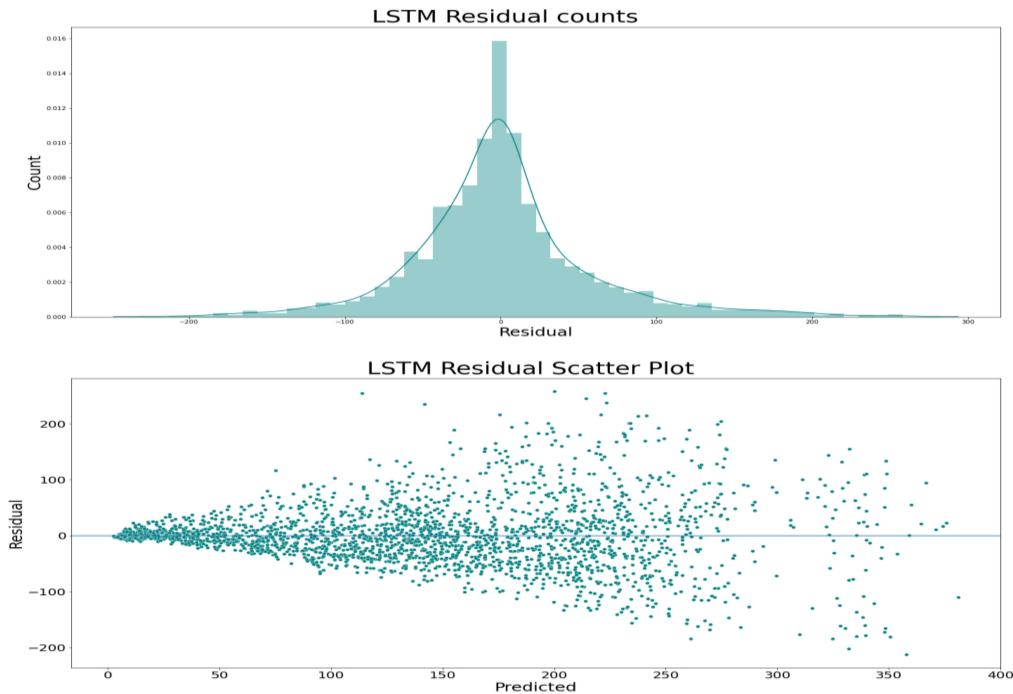


Figure 12. Residual analysis of LSTM with attention mechanism-based model on FD003 test dataset.

As observed in the LSTM residual counts portion of figures, the residual distribution of test data is almost symmetric around zero, which means the model is not biased towards over/underestimation. This fact is further corroborated from the residual scatter plot portion of Figures 11 and 12. The reader can observe that the population of residuals $e_i > 0$ is not drastically different from the population of $e_i < 0$, which demonstrates that the LSTM model is neither biased towards overestimation nor underestimation.

Additionally, another three metrics were utilized to evaluate the performance of the proposed prognostic approach. The three evaluation metrics are root mean square error (RMSE) [23,61], mean absolute error (MAE), and R-squared (R²). The motivation for choosing these three matrices is because they were widely utilized in the state of the art to evaluate various models' performance.

4.3. Model Uncertainty Quantification

In order to estimate the epistemic uncertainty of the proposed DNNs candidate models, model ensembling and bootstrap sampling [18,63] are utilized in our approach for capturing the uncertainty inherent in the RUL prediction. This requires creating a base training set from which all samples for training and validation are included as well as a different testing subset of data for estimating generalization performance following the bootstrap technique. Numerous subsets K of the complete training set are formed by sampling until a sampling budget is reached, with replacement. This budget is set to be equal to the sample's total number in the entire training set, as is customary. Following sampling, any remaining unused samples "indicated as out-of-bag samples" were utilized for validation. Next, K randomly initialized models and trained on the bootstrap sets. After training is complete, epistemic uncertainty is modeled by considering RUL predictions as random variables and estimating their projected value (averaging predictions over models' realizations), and variance can be described as follows:

$$E[\hat{y}] = \mu_{\hat{y}} = \sum_{k=1}^K M_k(x) \quad (21)$$

where $E[\hat{y}]$ is the predicted value for a given sample x , M_k denotes a single model realization, and K denotes the model realization's total number. Then, the variance is calculated as follows:

$$E[\hat{y}^2] - (E[\hat{y}])^2 = \sigma_{\hat{y}}^2 = \sum_{k=1}^K M_k(x)^2 - \left(\sum_{k=1}^K M_k(x) \right)^2 \quad (22)$$

where it has been utilized to calculate the standard deviation or spread around the mean for each data point in our predictions.

Uncertainty prediction for DNNs models is vital in both computational and real-world turbofan applications. Despite the enormous achievement of DNNs for RUL prediction, an inordinate amount of effort has been expended on improving point-estimate performance on widely employed benchmarks. While these standards aid in the advancement of DL, real-world safety-critical systems cannot rely solely on black-box prediction without insight into the inherent uncertainties in the models and data. Therefore, in this study, we attempted to predict the uncertainty in DNN models in order to make render them practical in practice because the presence of a few uncertainties in prognostics prediction can result in huge losses. Therefore, two cases based on our two models' best performance have been selected to predict the uncertainty associated with RUL. Figure 13 indicates that epistemic uncertainty is low across all time steps and roughly negligible near failure in both cases.

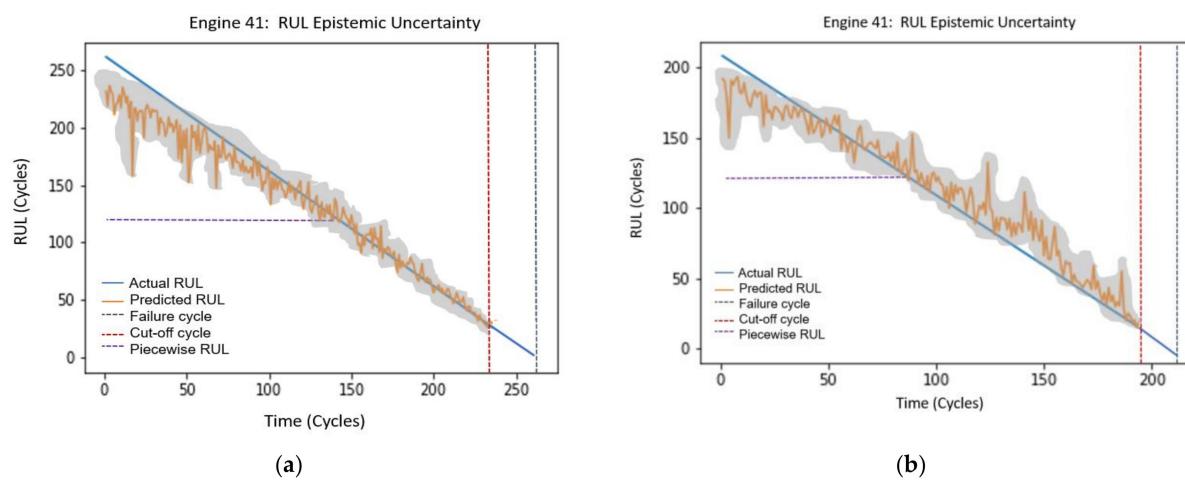


Figure 13. RUL prediction results of engine 41 with model uncertainty quantification: (a) uncertainty case prediction for attention-based LSTM model; (b) uncertainty case prediction for attention-based GRU model.

5. Comparison with Literature

This section aims to compare the proposed four DNN-based models with the state-of-the-art methods. In the literature, different DL methods have been used to predict RUL on the C-MAPSS benchmark dataset. Table 7 shows the comparison of the proposed deep candidate models with related literature contributions. The comparison is only shown for metrics available, but it shows the reader the promising results of the proposed DNNs-based predictor. The experimental results confirmed that the proposed DNNs-based predictors surpass the other previous methods for predicting RUL on independent benchmark testing of FD002 and FD003 subsets of data.

Table 7. Comparison of the proposed predictors with related literature contributions.

Prediction Model	FD002			FD003		
	MAE	RMSE	R2	MAE	RMSE	R2
Proposed Attention-based CNN Model	16.23	22.34	0.336	8.99	21.89	0.232
Proposed Attention-based SRNN Model	14.32	29.32	0.236	12.97	25.14	0.323
Proposed Attention-based GRU Model	14.75	18.38	0.162	13.32	19.28	0.423
Proposed Attention-based LSTM Model	1.42	12.87	0.236	5.47	11.23	0.168
MLP [21]	Not Reported	80.03	Not Reported	Not Reported	37.39	Not Reported
Deep LSTM [43]	-	24.49	-	-	16.18	-
RF [21]	-	20.23	-	-	22.34	-
DBN [21]	-	30.05	-	-	20.99	-
DW-RNN [22]	-	25.90	-	-	18.75	-
MTL-RNN [22]	-	25.78	-	-	17.98	-
DCNN [23]	-	22.36	-	-	12.64	-
DNN [23]	-	24.61	-	-	13.93	-
RNN [23]	-	24.03	-	-	13.36	-
LSTM [23]	-	24.42	-	-	13.54	-

Table 7. Cont.

Prediction Model	FD002			FD003		
	MAE	RMSE	R2	MAE	RMSE	R2
Attention-based Sequence to Sequence Model [64]	-	14.90	-	-	11.85	-
Attention-based DL model [65]	-	FD001 (14.53)	-	-	FD004 (27.08)	-
DARNN [66]; PRONOSTIA dataset	-	Bearing4 0.07 ± 0.002	-	-	Bearing5 0.07 ± 0.002	-

Table 7 shows that our proposed attention-based LSTM model has surpassed all the previous models in the literature. However, DCNN proposed by [23] is the only study that has obtained good results in the FD003 subset of data. Thus, our proposed attention-based LSTM predictor still achieved better results with an RMSE of 11.23. The proposed attention-based LSTM predictor has excellent capabilities of capturing long-term dependencies and extracts features from the time-frequency domain by incorporating the sliding TW technique and temporal information of signals. Based on the experimental findings, it has been observed that increasing the sliding TW results in improving RUL prediction accuracy. The proposed enhanced LSTM-based model predicts the RUL of aero engines with high accuracy without the requirement to comprehend engine construction or failure mechanism and without the need for professional knowledge and experience. The proposed model can be used as maintenance strategy for industrial equipment PHM with multivariate time series data obtained from various sensors.

6. Limitations and Future Research

Due to the nature of uncertainty inherent in the RUL prediction, our research enhanced the reliability of deterministic RUL prediction incorporating the uncertainty prediction based on the bootstrap sampling method. Therefore, further neural networks architecture and prediction interval optimization necessary to improve the prognostic performance. In future work, the proposed approach will be applied for the RUL prediction of aero engines with different operating conditions. When the operating conditions are more complex, the RUL prediction is more challenging, and this kind of problem deserves further study.

The model can be further enhanced in future work by increasing the number of convolutional nuclei and hidden neurons in the fully connected layer. Additionally, it is well-known that, in measurements, there are uncommon and inconsistent observations that outnumber the majority of the population of observations, referred to as anomalies. Since the raw vibration signals are used directly as input, the prognosis model requires a more complicated network structure in order to verify the correctness of the results, resulting in high calculation loads. Finally, a model combining stacked LSTM layers will be investigated to extract useful temporal features layer by layer and improve the model's robustness, while an attention mechanism will be used to efficiently address the problem of information loss throughout LSTM's long-distance signal transmission.

7. Conclusions

With the advancement of smart manufacturing, it is becoming increasingly critical to leverage huge historical data to predict the RUL of aero engine systems, identify potential problems early, and reduce the expense of manual inspection. This study proposed a data-driven approach for turbofan engine remaining useful life prediction using four different DNNs models with an attention mechanism. The sliding window technique is adopted to prepare data samples, and the testing engine units with shorter time cycles than N_{tw} are eliminated in the corresponding cases. We have evaluated the proposed DNN candidate models with two subsets of data, FD002, a simpler dataset, and FD003, a complex subset of data with six operating conditions. Therefore, among the four proposed models, the

attention-based LSTM model has outperformed the other three models due to its excellent capabilities to capture long-term dependencies and extract features from the time-frequency domain incorporating the sliding TW technique and temporal information of signals. The proposed approach is also compared with literature contributions and showed improved results in terms of better prediction quality and enhanced computational speed due to the dimensionality of reduction processing. The proposed attention mechanism-based LSTM model, presented in Figure 2, is an initial step towards improving the RUL prediction of the turbofan engine and paves the way for further research investigations aimed towards more robust and accurate DNN-based RUL prediction models because it can effectively map the relationship between features and the RUL.

Author Contributions: Conceptualization, A.M. and S.M.T.; methodology, A.M. and S.N.; software, A.M. and S.N.; validation, A.M., S.N. and S.M.T.; formal analysis, A.M., S.M.T. and R.F.A.; data curation, A.M. and S.N.; writing—original draft preparation, A.M.; writing—review and editing, S.M.T., I.A.A. and R.F.A.; visualization, A.M., R.F.A. and S.N.; supervision, S.M.T. and I.A.A.; project administration, A.M. and S.M.T.; funding acquisition, S.M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research/paper was fully supported by Universiti Teknologi PETRONAS, under the Yayasan Universiti Teknologi PETRONAS (YUTP) Fundamental Research Grant Scheme (YUTP-015LC0-123).

Acknowledgments: We are grateful to the editor and four anonymous reviewers for their valuable suggestions and comments, which significantly improved the quality of the manuscript.

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

References

1. Wang, Y.; Zhao, Y.; Addepalli, S. Remaining useful life prediction using deep learning approaches: A review. *Procedia Manuf.* **2020**, *49*, 81–88. [[CrossRef](#)]
2. Chen, X.; Wang, S.; Qiao, B.; Chen, Q. Basic research on machinery fault diagnostics: Past, present, and future trends. *Front. Mech. Eng.* **2018**, *13*, 264–291. [[CrossRef](#)]
3. Wei, Y.; Li, Y.; Xu, M.; Huang, W. A review of early fault diagnosis approaches and their applications in rotating machinery. *Entropy* **2019**, *21*, 409. [[CrossRef](#)] [[PubMed](#)]
4. Lu, B.L.; Liu, Z.H.; Wei, H.L.; Chen, L.; Zhang, H.; Li, X.H. A Deep Adversarial Learning Prognostics Model for Remaining Useful Life Prediction of Rolling Bearing. *IEEE Trans. Artif. Intell.* **2021**, *1*. [[CrossRef](#)]
5. Zeng, D.; Zhou, D.; Tan, C.; Jiang, B. Research on model-based fault diagnosis for a gas turbine based on transient performance. *Appl. Sci.* **2018**, *8*, 148. [[CrossRef](#)]
6. Chui, K.T.; Gupta, B.B.; Vasant, P. A Genetic Algorithm Optimized RNN-LSTM Model for Remaining Useful Life Prediction of Turbofan Engine. *Electronics* **2021**, *10*, 285. [[CrossRef](#)]
7. Souza, R.M.; Nascimento, E.G.; Miranda, U.A.; Silva, W.J.; Lepikson, H.A. Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Comput. Ind. Eng.* **2021**, *153*, 107060. [[CrossRef](#)]
8. Ahmed, U.; Ali, F.; Jennions, I. A review of aircraft auxiliary power unit faults, diagnostics and acoustic measurements. *Prog. Aerosp. Sci.* **2021**, *124*, 100721. [[CrossRef](#)]
9. de Jonge, B.; Teunter, R.; Tinga, T. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. *Reliab. Eng. Syst. Saf.* **2017**, *158*, 21–30. [[CrossRef](#)]
10. Seiti, H.; Tagipour, R.; Hafezalkotob, A.; Asgari, F. Maintenance strategy selection with risky evaluations using RAHP. *J. Multi-Criteria Decis. Anal.* **2017**, *24*, 257–274. [[CrossRef](#)]
11. Ozcan, S.; Simsir, F. A new model based on Artificial Bee Colony algorithm for preventive maintenance with replacement scheduling in continuous production lines. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1175–1186. [[CrossRef](#)]
12. Xie, Z.; Du, S.; Lv, J.; Deng, Y.; Jia, S. A Hybrid Prognostics Deep Learning Model for Remaining Useful Life Prediction. *Electronics* **2020**, *10*, 39. [[CrossRef](#)]
13. Zhao, Y.; Wang, Y. Remaining Useful Life Prediction for Multi-sensor Systems Using a Novel End-to-end Deep-Learning Method. *Measurement* **2021**, *182*, 109685. [[CrossRef](#)]
14. Hong, C.W.; Lee, C.; Lee, K.; Ko, M.S.; Kim, D.E.; Hur, K. Remaining Useful Life Prognosis for Turbofan Engine Using Explainable Deep Neural Networks with Dimensionality Reduction. *Sensors* **2020**, *20*, 6626. [[CrossRef](#)] [[PubMed](#)]
15. 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), Ottawa, Canada, 19–21 June 2017; pp. 88–95.

16. Ren, L.; Liu, Y.; Wang, X.; Lü, J.; Deen, M.J. Cloud-edge based lightweight temporal convolutional networks for remaining useful life prediction in iiot. *IEEE Internet Things J.* **2020**, *8*, 12578–12587. [[CrossRef](#)]
17. Malhotra, P.; Vishnu, T.V.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. Multisensor prognostics using an unsupervised health index based on LSTM encoder-decoder. *arXiv* **2020**, arXiv:1608.06154.
18. Cornelius, J.; Brockner, B.; Hong, S.H.; Wang, Y.; Pant, K.; Ball, J. Estimating and Leveraging Uncertainties in Deep Learning for Remaining Useful Life Prediction in Mechanical Systems. In Proceedings of the 2020 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit, MI, USA, 8–10 June 2020; pp. 1–8.
19. Ramadhan, M.S.; Hassan, K.A. Remaining useful life prediction using an integrated Laplacian-LSTM network on machinery components. *Appl. Soft Comput.* **2021**, *112*, 107817.
20. Ahn, G.; Yun, H.; Hur, S.; Lim, S. A Time-Series Data Generation Method to Predict Remaining Useful Life. *Processes* **2021**, *9*, 1115. [[CrossRef](#)]
21. Zhang, C.; Lim, P.; Qin, K.; Tan, K.C. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2306–2318. [[CrossRef](#)]
22. Aggarwal, K.; Atan, O.; Farahat, A.K.; Zhang, C.; Ristovski, K.; Gupta, C. Two Birds with One Network: Unifying Failure Event Prediction and Time-to-Failure Modeling. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 1308–1317.
23. 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. [[CrossRef](#)]
24. Gao, G.; Que, Z.; Xu, Z. Predicting Remaining Useful Life with Uncertainty Using Recurrent Neural Process. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020; pp. 291–296.
25. Sankararaman, S.; Goebel, K. Why is the Remaining Useful Life Prediction Uncertain? In Proceedings of the Annual Conference of the PHM Society, New Orleans, LA, USA, 14–17 October 2013; Volume 5, p. 1.
26. Chen, N.; Tsui, K.L. Condition monitoring and remaining useful life prediction using degradation signals: Revisited. *IIE Trans.* **2013**, *45*, 939–952. [[CrossRef](#)]
27. Zhai, Q.; Ye, Z.-S. RUL Prediction of Deteriorating Products Using an Adaptive Wiener Process Model. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2911–2921. [[CrossRef](#)]
28. Zhu, J.; Chen, N.; Peng, W. Estimation of Bearing Remaining Useful Life Based on Multiscale Convolutional Neural Network. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3208–3216. [[CrossRef](#)]
29. Rui, K.; Wenjun, G.; Chen, Y. Model-driven degradation modeling approaches: Investigation and review. *Chin. J. Aeronaut.* **2020**, *33*, 1137–1153.
30. Park, P.; Jung, M.; Di Marco, P. Remaining Useful Life Estimation of Bearings Using Data-Driven Ridge Regression. *Appl. Sci.* **2020**, *10*, 8977. [[CrossRef](#)]
31. Lei, Y. *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*; Butterworth-Heinemann: Oxford, UK, 2016.
32. Chen, Y.; Peng, G.; Zhu, Z.; Li, S. A novel deep learning method based on attention mechanism for bearing remaining useful life prediction. *Appl. Soft Comput.* **2019**, *86*, 105919. [[CrossRef](#)]
33. Ingemarsdotter, E.; Kambanou, M.L.; Jamsin, E.; Sakao, T.; Balkenende, R. Challenges and solutions in condition-based maintenance implementation-A multiple case study. *J. Clean. Prod.* **2021**, *296*, 126420. [[CrossRef](#)]
34. Kaparthi, S.; Bumblauskas, D. Designing predictive maintenance systems using decision tree-based machine learning techniques. *Int. J. Qual. Reliab. Manag.* **2020**, *37*, 659–686. [[CrossRef](#)]
35. Wen, L.; Dong, Y.; Gao, L. A new ensemble residual convolutional neural network for remaining useful life estimation. *Math. Biosci. Eng.* **2019**, *16*, 862–880. [[CrossRef](#)] [[PubMed](#)]
36. Babu, G.S.; Zhao, P.; Li, X.L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International Conference on Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2016; pp. 214–228.
37. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5990–5998. [[CrossRef](#)]
38. Li, J.; Li, X.; He, D. A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. *IEEE Access* **2019**, *7*, 75464–75475. [[CrossRef](#)]
39. Huang, R.; Xi, L.; Li, X.; Liu, C.R.; Qiu, H.; Lee, J. Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mech. Syst. Signal Process.* **2007**, *21*, 193–207. [[CrossRef](#)]
40. Muneer, A.; Taib, S.M.; Fati, S.M.; Alhussian, H. Deep-Learning Based Prognosis Approach for Remaining Useful Life Prediction of Turbofan Engine. *Symmetry* **2021**, *13*, 1861. [[CrossRef](#)]
41. Khawaja, T.; Vachtsevanos, G.; Wu, B. Reasoning about Uncertainty in Prognosis: A Confidence Prediction Neural Network Approach. In Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society, Redmond, WA, USA, 20–22 August 2020; pp. 7–12.
42. Malhi, A.; Yan, R.; Gao, R.X. Prognosis of Defect Propagation Based on Recurrent Neural Networks. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 703–711. [[CrossRef](#)]

43. 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), Beijing, China, 10–12 October 2016; pp. 135–140.
44. Zhao, R.; Wang, J.; Yan, R.; Mao, K. Machine Health Monitoring with LSTM Networks. In Proceedings of the 2016 10th International Conference on Sensing Technology (ICST), Nanjing, China, 11–13 November 2016; pp. 1–6.
45. Ren, L.; Sun, Y.; Wang, H.; Zhang, L. Prediction of bearing remaining useful life with deep convolution neural network. *IEEE Access* **2018**, *6*, 13041–13049. [[CrossRef](#)]
46. Song, Y.; Xia, T.; Zheng, Y.; Zhuo, P.; Pan, E. Remaining useful life prediction of turbofan engine based on Autoencoder-BLSTM. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 1611–1619.
47. Zhang, X.; Xiao, P.; Yang, Y.; Cheng, Y.; Chen, B.; Gao, D.; Liu, W.; Huang, Z. Remaining Useful Life Estimation Using CNN-XGB With Extended Time Window. *IEEE Access* **2019**, *7*, 154386–154397. [[CrossRef](#)]
48. Ji, S.; Han, X.; Hou, Y.; Song, Y.; Du, Q. Remaining Useful Life Prediction of Airplane Engine Based on PCA-BLSTM. *Sensors* **2020**, *20*, 4537. [[CrossRef](#)]
49. Wang, R.; Shi, R.; Hu, X.; Shen, C. Remaining Useful Life Prediction of Rolling Bearings Based on Multiscale Convolutional Neural Network with Integrated Dilated Convolution Blocks. *Shock Vib.* **2021**, *1*–11. [[CrossRef](#)]
50. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 305–345.
51. Alqushaibi, A.; Abdulkadir, S.J.; Rais, H.M.; Al-Tashi, Q.; Ragab, M.G.; Alhussian, H. Enhanced Weight-Optimized Recurrent Neural Networks Based on Sine Cosine Algorithm for Wave Height Prediction. *J. Mar. Sci. Eng.* **2021**, *9*, 524. [[CrossRef](#)]
52. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
53. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
54. Naseer, S.; Ali, R.F.; Muneer, A.; Fati, S.M. IAmideV-deep: Valine amidation site prediction in proteins using deep learning and pseudo amino acid compositions. *Symmetry* **2021**, *13*, 560. [[CrossRef](#)]
55. Muneer, A.; Fati, S.M. Efficient and Automated Herbs Classification Approach Based on Shape and Texture Features using Deep Learning. *IEEE Access* **2020**, *8*, 196747–196764. [[CrossRef](#)]
56. Durairajah, V.; Gobee, S.; Muneer, A. Automatic Vision Based Classification System Using DNN and SVM Classifiers. In Proceedings of the 2018 3rd International Conference on Control, Robotics and Cybernetics (CRC), Penang, Malaysia, 26–28 September 2018; pp. 6–14.
57. Naseer, S.; Ali, R.F.; Fati, S.M.; Muneer, A. iNitroY-Deep: Computational Identification of Nitrotyrosine Sites to Supplement Carcinogenesis Studies Using Deep Learning. *IEEE Access* **2021**, *9*, 73624–73640. [[CrossRef](#)]
58. Naseer, S.; Saleem, Y. Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks. *Trans. Internet Inf. Syst.* **2018**, *12*, 5159–5178.
59. Yang, B.; Liu, R.; Zio, E. Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9521–9530. [[CrossRef](#)]
60. Hou, G.; Xu, S.; Zhou, N.; Yang, L.; Fu, Q. Remaining useful life estimation using deep convolutional generative adversarial networks based on an autoencoder scheme. *Comput. Intell. Neurosci.* **2020**. [[CrossRef](#)] [[PubMed](#)]
61. Xiang, S.; Qin, Y.; Luo, J.; Pu, H.; Tang, B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107927. [[CrossRef](#)]
62. 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, Denver, CO, USA, 6–9 October 2008; pp. 1–9.
63. Osband, I.; Blundell, C.; Pritzel, A.; Van Roy, B. Deep exploration via bootstrapped DQN. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4026–4034.
64. Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C.K.; Yan, R.; Li, X. Attention Sequence to Sequence Model for Machine Remaining Useful Life Prediction. *arXiv* **2020**, arXiv:2007.09868.
65. Chen, Z.; Wu, M.; Zhao, R.; Guretno, F.; Yan, R.; Li, X. Machine Remaining Useful Life Prediction via an Attention-Based Deep Learning Approach. *IEEE Trans. Ind. Electron.* **2020**, *68*, 2521–2531. [[CrossRef](#)]
66. Zeng, F.; Li, Y.; Jiang, Y.; Song, G. A deep attention residual neural network-based remaining useful life prediction of machinery. *Measurement* **2021**, *181*, 109642. [[CrossRef](#)]