aerospace

MDPI

Article

A Comparative Study of Data-Driven Prognostic Approaches under Training Data Deficiency

Jinwoo Song [1], Seong Hee Cho [2], Seokgoo Kim [3], Jongwhoa Na [4] and Joo-Ho Choi [5,*]

[1] Department of Smart Air Mobility, Korea Aerospace University, Goyang 10540, Republic of Korea; jwsong@kau.kr
[2] Intelligent AI Department, Korea Shipbuilding & Offshore Engineering (KSOE), Seoul 03058, Republic of Korea; seonghee.cho@hd.com
[3] Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA; sgkim905@gmail.com
[4] School of Electronics and Information Engineering, Korea Aerospace University, Goyang 10540, Republic of Korea; jwna@kau.ac.kr
[5] School of Aerospace & Mechanical Engineering, Korea Aerospace University, Goyang 10540, Republic of Korea
* Correspondence: jhchoi@kau.ac.kr

Abstract: In industrial system health management, prognostics play a crucial role in ensuring safety and enhancing system availability. While the data-driven approach is the most common for this purpose, they often face challenges due to insufficient training data. This study delves into the prognostic capabilities of four methods under the conditions of limited training datasets. The methods evaluated include two neural network-based approaches, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks, and two similarity-based methods, Trajectory Similarity-Based Prediction (TSBP) and Data Augmentation Prognostics (DAPROG), with the last being a novel contribution from the authors. The performance of these algorithms is compared using the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) datasets, which are made by simulation of turbofan engine performance degradation. To simulate real-world scenarios of data deficiency, a small fraction of the training datasets from the original dataset is chosen at random for the training, and a comprehensive assessment is conducted for each method in terms of remaining useful life prediction. The results of our study indicate that, while the Convolutional Neural Network (CNN) model generally outperforms others in terms of overall accuracy, Data Augmentation Prognostics (DAPROG) shows comparable performance in the small training dataset, being particularly effective within the range of 10% to 30%. Data Augmentation Prognostics (DAPROG) also exhibits lower variance in its predictions, suggesting a more consistent performance. This is worth highlighting, given the typical challenges associated with artificial neural network methods, such as inherent randomness, non-intuitive decision-making processes, and the complexities involved in developing optimal models.

Keywords: prognostics and health management (PHM); remaining useful life (RUL); data deficiency; data augmentation prognostics (daprog); dynamic time warping (DTW)

check for updates

Citation: Song, J.; Cho, S.H.; Kim, S.; Na, J.; Choi, J.-H. A Comparative Study of Data-Driven Prognostic Approaches under Training Data Deficiency. *Aerospace* **2024**, *11*, 741. https://doi.org/10.3390/aerospace11090741

Academic Editor: Kan Xie

Received: 11 June 2024
Revised: 21 August 2024
Accepted: 3 September 2024
Published: 10 September 2024

Copyright: © 2024 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/).

1. Introduction

Predictive maintenance has become essential among modern industrial maintenance strategies, representing a paradigm shift from reactive to proactive maintenance. Essential part of the predictive maintenance is the prognostics that predicts the future health and remaining useful life (RUL) of the systems in concern. This proactive approach is key to maintaining uninterrupted operations across various industries, enhancing operational efficiency, and minimizing downtime.

Prognostics algorithms are typically categorized into two primary approaches, physics-based and data-driven, each offering distinct advantages and facing unique challenges. Physics-based methods estimate RUL by employing mathematical models that describe degradation processes or failure mechanisms. Examples include employing Paris' Law for predicting crack growth [1,2] and utilizing the governing equations of mechanical and electrical models for predicting motor degradation [3]. While these methods benefit from their reliance on established physical principles, they often encounter difficulties due to the complexity of developing accurate models, especially for systems with multiple components. This complexity can hinder the effectiveness of physics-based approaches in practical applications.

In contrast, the data-driven approaches rely on the acquired data to identify the hidden pattern of data. The deep learning models especially have been growing rapidly due to their applicability to general problems. These methods establish relationships between sensor readings and system health for RUL prediction. The mapping relationships between sensor signal and the system's health conditions could be established by the data-driven approaches for estimation of RUL. Deep learning models have already been applied in various ways for the prediction of RUL. A fundamental example is the multilayer perceptron (MLP). Gebraeel et al. [4] predicted the RUL of bearings by training MLP models with vibration signals obtained from accelerated life experiments. Similarly, Mahamad et al. [5] employed an MLP model using calculated features.

Particularly noteworthy in the domain of RUL estimation using deep learning models is the application of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models. Babu et al. [6] firstly predicted RUL using the CNN algorithm, while Li et al. [7] enhanced CNN performance by applying various configurations to the model. Zheng et al. [8] successfully implemented LSTM algorithms for RUL prediction. Those works have been instrumental, forming the basis of numerous subsequent studies. Jayasinghe et al. [9] proposed an architecture combining convolutional layers with LSTM layers for RUL estimation. Mo et al. [10] introduced a multi-head CNN-LSTM architecture, employing parallel CNN branches in series with LSTM. Hong et al. [11] utilized dimensionality reduction and Shapley additive explanation in a deep-stacked convolutional Bi-LSTM model, which reduces complexity and prevents overfitting while maintaining high accuracy. Zhan et al. [12] proposed an approach to improve uncertainty quantification in RUL prediction by integrating a Multi-Distribution Fusion structure with LSTM and CNN models, respectively. More advanced models, such as the Transformer model [13], have been explored for RUL prediction, continuing the trend of introducing innovative approaches in the field. Collectively, these models mark a substantial progression in RUL estimation, offering enhanced efficiency and improved capability for feature extraction.

Despite these advancements, data-driven methods still face numerous challenges and limitations, as reviewed by Ferreira et al. [14]. One of the most important challenges is the deficiency of Run-To-Failure (RTF) data, a concern that has become prevalent in the context of the automotive industry [14] and in broader applications of prognostics and health management [15]. Gathering sufficient RTF data presents significant challenges, either due to the rarity of faults in field operations where structures are often replaced before failure, or due to the high costs associated with obtaining such data in testbeds. Various strategies have been explored in the literature to overcome this limitation, including the use of censored data [16], conducting accelerated life experiments [17], and investigating self-supervised learning approaches [18].

To address this challenge, the authors have recently introduced Data Augmentation Prognostics (DAPROG), a novel approach that leverages Dynamic Time Warping (DTW)-based data mapping to generate virtual RTF data under different conditions for RUL prediction [19]. This method was applied to issues such as crack growth and battery degradation, utilizing RTF datasets from varied operating conditions. While DAPROG shares several similarities with the Trajectory Similarity-Based Prediction (TSBP) researched

by Wang et al. [20], it notably achieves significant performance improvements in data deficiency scenarios.

The focus of this paper is to evaluate the effectiveness of the DAPROG method in addressing the challenges posed by data scarcity and to compare its performance with the following distinct, well-established data-driven methods: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Trajectory Similarity-Based Prediction (TSBP). For this comparative analysis, we utilize the popular Commercial Modular Aero-Propulsion System Simulation (CMAPSS) dataset. The same approaches made by Li [7] and Zheng [8] are implemented for the CNN and LSTM, respectively, as close as possible, based on the provided information in the papers. For this experimental study, only a fraction of the available training data are employed, thereby creating a data-deficient environment. In doing so, it allows for an in-depth analysis and comparison of how each algorithm performs under such constrained conditions. This approach not only highlights the capabilities and limitations of each method in data-limited situations but also sheds light on the practical applicability of these algorithms in real-world scenarios where data insufficiency is a common challenge.

2. Four Approaches for the Prognostic Study

In this section, the four algorithms used to predict the RUL are briefly explained, which are CNN and LSTM, as the most fundamental artificial neural network (ANN) algorithms; TSBP, as another unique data-driven approach in the prognostics; and DAPROG, which was recently developed by the authors.

2.1. Convolutional Neural Network

Convolutional Neural Networks, primarily dominant in analyzing grid-like data structures such as images, also demonstrate considerable effectiveness in processing sequenced sensor signals for tasks like RUL prediction. This ability of CNN to capture and utilize spatial relationships and hierarchical feature representations makes them exceptionally well-suited for analyzing time-series data from sensors, where recognizing patterns over time is crucial. By leveraging their deep learning capabilities, CNN can autonomously identify subtle, predictive features in sensor data, thus providing accurate assessments of a system's health and predicting its RUL with high reliability.

Advanced variations, such as Multi-Scale Deep CNNs [21] and Multi-Scale Temporal Convolutional Networks [22], further enhance this capability. A Multi-Scale Deep CNN utilizes multiple convolutional layers at different scales, allowing the model to ensure a comprehensive feature extraction process. A Multi-Scale Temporal Convolutional Network adopts the temporal convolutional network framework, which is good at extracting time-sequence information. This model is specifically designed to integrate historical condition monitoring data, facilitating high-level representations that are critical for an accurate RUL estimation.

CNNs and their advanced forms are pivotal in developing robust predictive models in the field of predictive maintenance. CNNs are effectively integrated with various models, enhancing their capabilities and making them essential algorithms in the field of predictive analytics.

2.2. Long Short-Term Memory (LSTM) Network

LSTM networks, a specialized form of recurrent neural networks (RNNs), are designed to handle long-term dependencies in sequential data, making them highly effective for tasks such as time-series analysis and RUL prediction [8]. These networks utilize a series of gates—input, output, and forget—to selectively maintain or discard information, thereby enhancing their ability to make accurate predictions over extended periods. Expanding on the LSTM framework, Bidirectional LSTM (BiLSTM) networks process data from both past and future directions. This bidirectional approach enables the network to capture

contexts from both directions, enhancing the contextual understanding of the system's current state [23].

Another derivative of the traditional RNN is the Gated Recurrent Unit (GRU), which simplifies the LSTM with a simpler architecture that combines the input and forget gates into a single update gate. Despite their reduced complexity, GRUs retain the ability to effectively model long-term dependencies, providing a computationally efficient option for RUL prediction tasks. Bidirectional GRU (BiGRU) networks further enhance the capabilities of GRUs by processing data sequentially both forwards and backwards. This allows for a more comprehensive analysis of sequential data, improving the model's performance in accurately predicting the remaining useful life of complex systems [24].

While LSTM networks form the cornerstone of sequential data analysis, their extensions and variants like BiLSTM, GRU, and BiGRU may offer better solutions for various applications.

2.3. Trajectory Similarity-Based Prediction

Trajectory Similarity-Based Prediction (TSBP) is a non-parametric algorithm for RUL prediction that leverages the most similar degradation trajectories from existing RTF datasets [25,26]. TSBP consists of the following three key steps:

The first is the degradation trajectory abstraction. Here, a degradation trajectory for the health index (HI) is constructed for each unit in the RTF datasets. This is achieved by fitting the data to multivariate linear regression and exponential curves, with the resultant trajectory represented as a solid red curve in Figure 1a. The HI, indicative of the current degradation status from sensor measurements, is detailed in a later section.

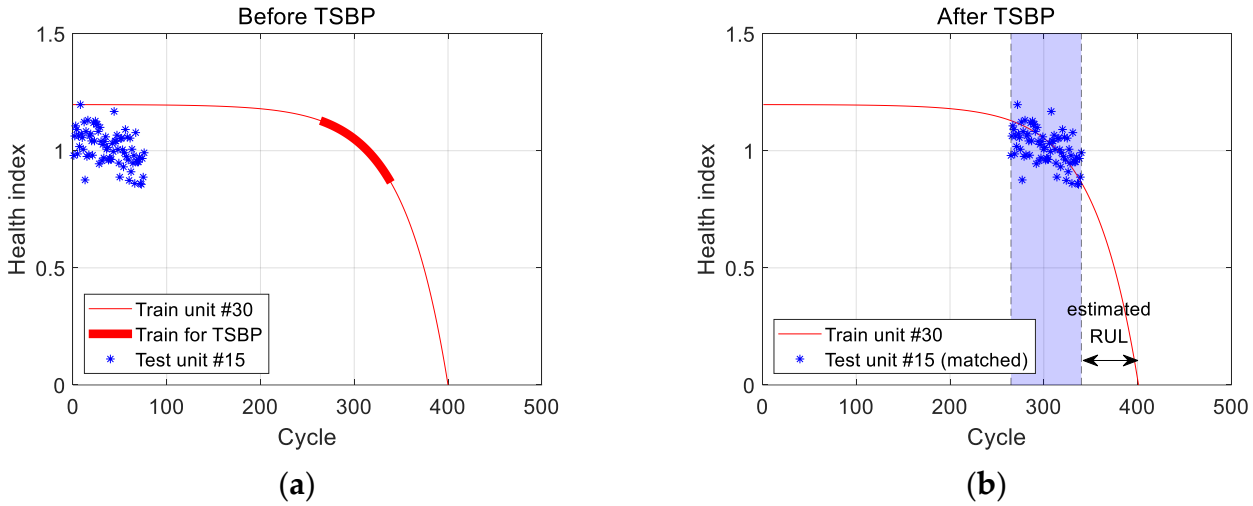


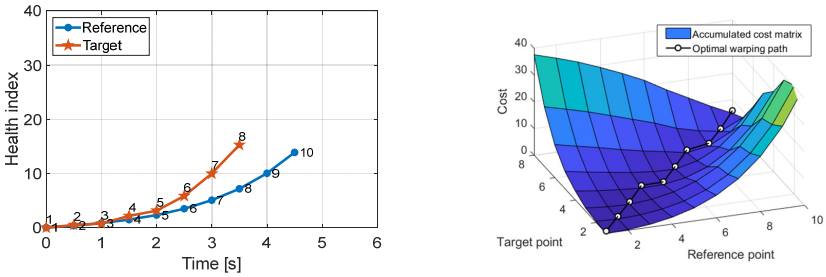
Figure 1. RUL estimation using TSBP method: (**a**) before, (**b**) after TSBP.

The second is the similarity evaluation. The current test data, shown as blue asterisks in Figure 1a, are evaluated for similarity against each training curve. This involves shifting the test data along the time axis to find the point of highest similarity to the curve, quantified by the sum of Euclidean distances normalized by the variance of the training curve. The location of maximum similarity is depicted as a thicker portion of the curve in Figure 1a. The RUL is then estimated based on the remaining duration of the curve from the last measurement of the test data, as shown in Figure 1b.

Finally, RUL estimates derived from each training curve are integrated into a single final RUL estimate. This involves applying outlier removal and appropriate weighting to the estimates.

2.4. Data Augmentation Prognostics

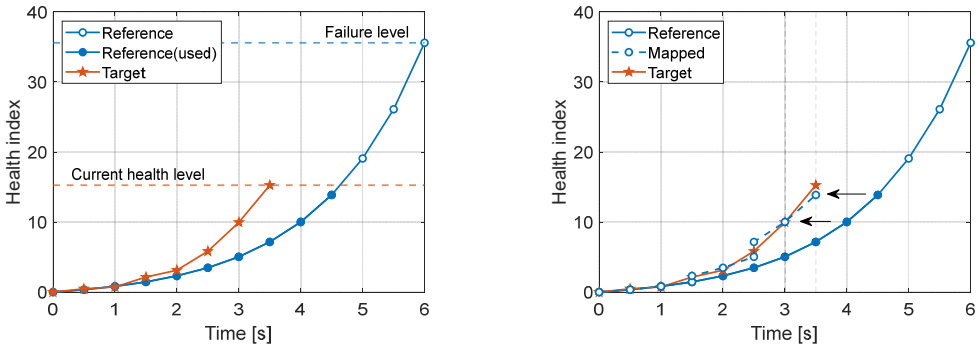
Developed by the authors, Data augmentation prognostics (DAPROG) is an RUL estimation algorithm that utilizes Dynamic Time Warping (DTW) to create virtual RTF data from existing RTF datasets [27]. DTW is an algorithm for aligning two time-series data in a way that minimizes the total cost defined by the DTW distance [26]. An example in Figure 2a shows two health index data series, with the blue circles and red asterisks representing the reference (training) and target (test) data with their sequence point number, respectively. Figure 2b displays the accumulated cost matrix, constructed by summing the distances for all possible paired paths between the two series. Each matrix entry at (i, j) represents the cost to align the i-th point of the target series with the j-th point of the reference series, plus the minimum cost from the following neighboring cells: (i – 1, j), (i, j – 1), and (i – 1, j – 1). The optimal warping path, which minimizes the total cost by accommodating the alignment of the sequences through stretching and compressing, is then determined to ensure the best alignment between the reference and target data.



(**a**) (**b**)

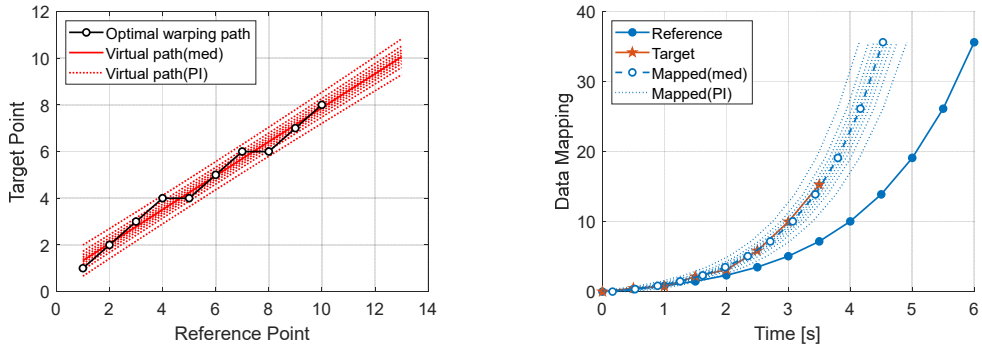
Figure 2. Illustration of DTW: (**a**) reference and target data, (**b**) optimal warping path by DTW.

DAPROG applies DTW to generate virtual RTF data using reference RTF data that have degraded to the failure level and target data which have not yet reached the failure level. For instance, in Figure 3, the reference extends to the failure level, representing the RTF data obtained under different conditions. DTW is applied between the target and the reference data up to the current health level (marked by a red dashed line in Figure 3a). The obtained optimal warping path (Figure 2b) is used to match the reference to the target, as demonstrated in Figure 3b. The linear regression of the warping path is then used to extrapolate into the future on the sequence domain. The scatter of the path informs the construction of predictive intervals (Pis) using the mean and variance of the regression parameters, with different significance values (α) used to generate various mapping paths, as illustrated in Figure 3c. These paths are then extrapolated to predict future outcomes, as shown by the multiple blue dotted curves in Figure 3d, which are used to predict the target RULs.



(**a**) (**b**)

Figure 3. *Cont.*



(**c**) (**d**)

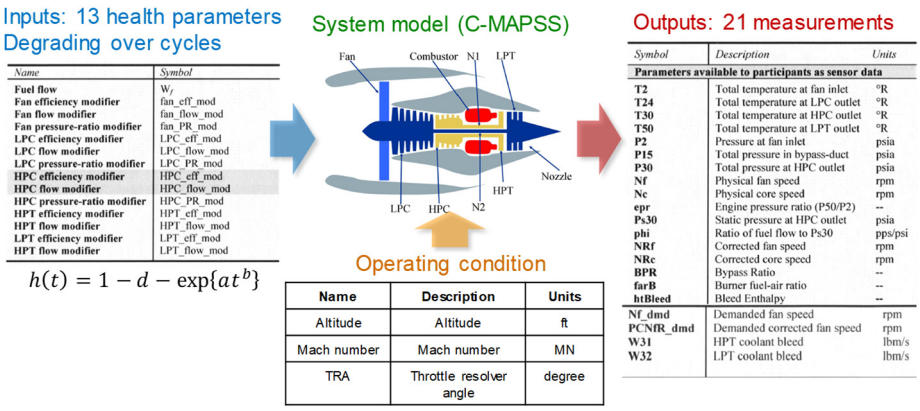
Figure 3. Illustration of DAPROG: (**a**) reference with RTF and target data, (**b**) data mapping result, (**c**) virtual warping path and extrapolation, and (**d**) virtual RTF data generation.

3. Experimental Study

In this section, a detailed analysis of the experimental conditions and results are provided. First, we will describe the datasets used, as well as the pre-processing steps and various configurations employed to construct each prediction model (CNN, LSTM, TSBP, DAPROG). Subsequently, we will delve into the methodologies used in the experiments and discuss the obtained results.

3.1. CMAPSS Dataset

The CMAPSS stands for the Commercial Modular Aero-Propulsion System Simulation, which is software that can generate virtual sensor measurements in the turbofan engine of the 90,000 lb thrust undergoing different operation conditions and health degradations [27]. It consists of the following five parts: the fan, low-pressure compressor (LPC), high-pressure compressor (HPC), high-pressure turbine (HPT), and low-pressure turbine (LPT), as shown in Figure 4. The operating conditions are given by three variables (altitude, Mach number, and throttle resolver angle). Thirteen health parameters are introduced to simulate the engine performance degradation, which deteriorates over cycles as given by $h(t)$, where t denotes the cycle. The outputs are the 21 sensor measurements, such as the temperature, pressure, and speed at the sections of the five parts [28]. Given a certain operating condition and degradation parameters, the sensor measurements are made by the CMAPSS once per cycle, starting from the normal state until failure.



Name	*Symbol*
Fuel flow	W_f
Fan efficiency modifier	fan_eff_mod
Fan flow modifier	fan_flow_mod
Fan pressure-ratio modifier	fan_PR_mod
LPC efficiency modifier	LPC_eff_mod
LPC flow modifier	LPC_flow_mod
LPC pressure-ratio modifier	LPC_PR_mod
HPC efficiency modifier	HPC_eff_mod
HPC flow modifier	HPC_flow_mod
HPC pressure-ratio modifier	HPC_PR_mod
HPT efficiency modifier	HPT_eff_mod
HPT flow modifier	HPT_flow_mod
LPT efficiency modifier	LPT_eff_mod
HPT flow modifier	LPT_flow_mod

Name	**Description**	**Units**
Altitude	Altitude	ft
Mach number	Mach number	MN
TRA	Throttle resolver angle	degree

Symbol	*Description*	*Units*
Parameters available to participants as sensor data		
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°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
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

Figure 4. Turbofan engine degradation simulation by CMAPSS [27].

While there are four datasets with different scenarios of health degradation and operating conditions, we concentrate on dataset FD001, as it embodies a single type of degradation—High-Pressure Compressor (HPC) failure—under a single operating condition. The reason to choose this is to only focus on the performance under the data deficiency

while excluding the other influences, such as the multiple failure modes or operation conditions. There are 100 training and test data, respectively, in dataset FD001. The training data are the RTF data of 21 sensor measurements that reached a pre-defined failure threshold, whereas the test data are those terminated at a certain cycle before failure. The test data are used to predict the RUL via the prognosis algorithm made by the training data, and the prediction accuracy is evaluated by comparing it with the actual RUL. Among the 100 training data, the plots of the sensor measurements are given in Figure 5 for engine #2 as an example where the end of life (EOL) is 287 cycles. Also, the EOLs of all engines in the training dataset are plotted in Figure 6, showing a large variance with each other ranging from a minimum of 128 and a maximum 362 cycles for engine 39 and 69, respectively.

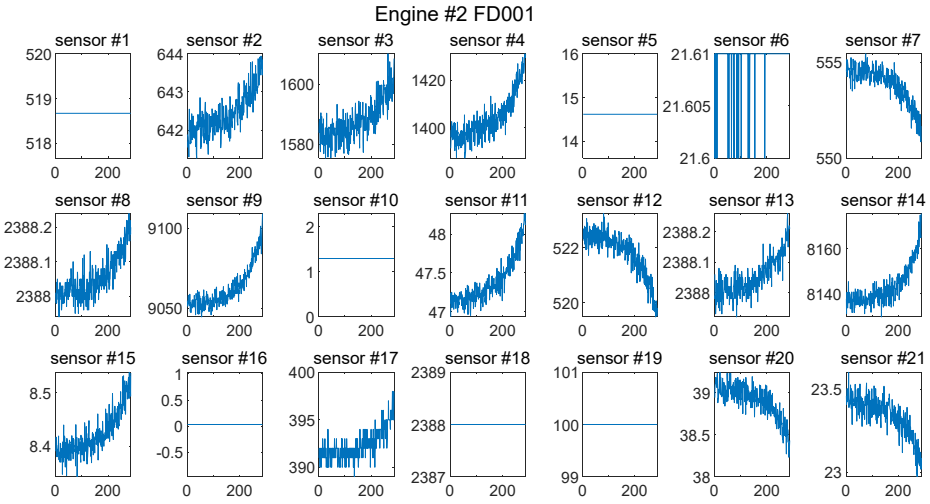


Figure 5. Sensor measurement of training engine #2 in dataset FD001.

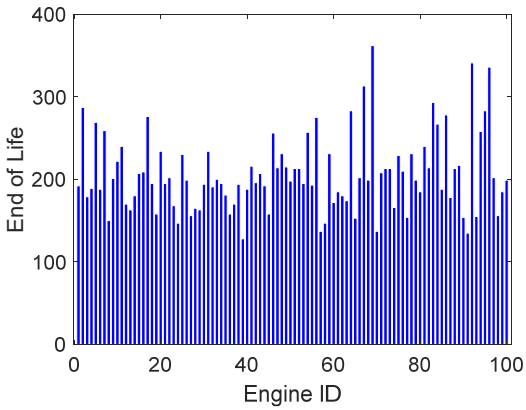


Figure 6. End of life of all training dataset FD001.

To enhance the training accuracy for all methods in RUL prediction, the number of input sensors is reduced by selecting only a few exhibiting higher Spearman rank correlation, which measures the relationship of the sensor data with respect to the cycle. Out of the original 21, only 7 sensors (indices 2, 3, 4, 7, 11, 12, 15) are utilized for further analysis.

3.2. Neural Network-Based Approaches: CNN, LSTM

To enhance the performance of both the CNN [7] and LSTM [8] models in RUL prediction, several forms of pre-processing are conducted in advance to the training procedure. The selected sensor values undergo normalization, which differs slightly between the two models. For the CNN model, the sensor data are normalized to fall within a range of -1 to 1. In contrast, the LSTM model utilizes a standard z-score normalization.

For the CNN model, a time window processing technique is incorporated, as implemented in Li's research [7]. This approach is particularly effective in multi-variate time series problems, like RUL estimation, where temporal sequence data often provide more insights than multi-variate data points sampled at a single time step. At each time step, sensor data within the time window are collected to form a high-dimensional feature vector, serving as the network's input. Consequently, for each time step in the CNN model, a normalized subset is prepared, consisting of a 7×30 matrix from the seven selected sensors and the window size of 30 time steps. This matrix represents data within a 30-time step window for a single engine unit in the training sub-dataset. On the other hand, the LSTM model utilizes the signals for the entire period from the seven sensors for training at once, providing a comprehensive dataset for more nuanced temporal analysis.

To more accurately estimate the RUL, a piece-wise linear function is employed, as defined in the studies [7,8] and illustrated in Figure 7. The RUL is assigned as constant at the early stage and then decreases linearly after a certain cycle. The reason is that the RUL accuracies at the early cycles are usually poor and unimportant. In this study, 130 cycles are taken, which is the round-off value of minimum EOL in the training set.

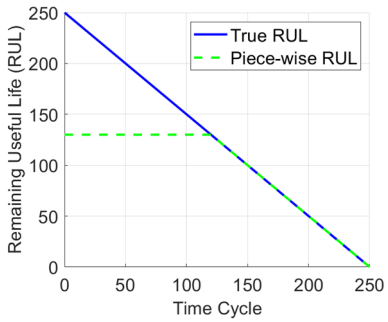


Figure 7. Piece-wise RUL (maximum RUL is 130 cycles).

3.2.1. RUL Prediction Using CNN

The CNN model architecture is constructed based on Li's work [7]. It comprises five CNN layers in the hidden layer section—four of these have a length of 10 and 10 channels, and one has a length of 3 with a single channel. Following the CNN layers are a flatten layer, a dropout layer with a dropout rate of 0.5, and a fully connected layer consisting of 100 nodes, as illustrated in Figure 8. All layers employ the tanh activation function.

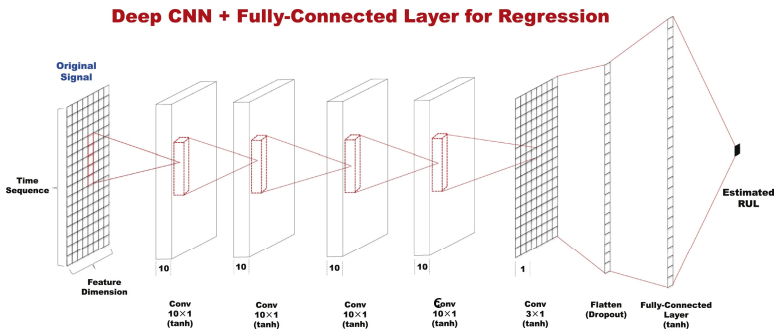


Figure 8. Implemented CNN architecture for RUL estimation [7].

For optimization, the Adam algorithm is utilized. The training process involves randomly dividing the samples into multiple mini batches, each containing 512 samples. The data split comprises 70% for training and 30% for validation. To enhance the efficiency of the training, an early stopping strategy is applied that halts the training process if there is no observable improvement in the performance of the validation dataset, ensuring that the model does not overfit to the training data while maximizing its predictive accuracy.

3.2.2. RUL Prediction Using LSTM Network

The LSTM network model in this study follows the design in Zheng's work [8]. This model's architecture includes two LSTM layers with 32 and 64 nodes, respectively, followed by two fully connected layers, each comprising eight nodes, as illustrated in Figure 9. Additionally, a dropout layer with a dropout rate of 0.5 is included in the hidden layers. The RMSprop optimizer is used for training with mini batches of 20 samples each. Similar to the CNN model, the data split for the LSTM model is 70% for training and 30% for validation, and an early stopping strategy is also employed during the training process.

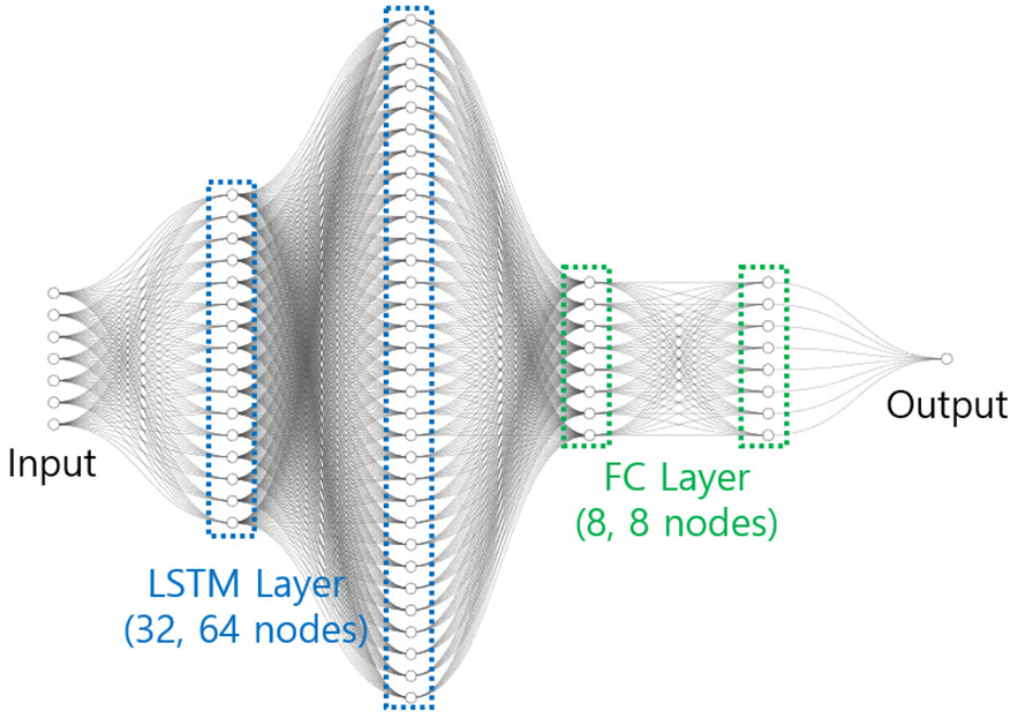


Figure 9. Implemented LSTM network architecture for RUL estimation.

3.3. Similarity-Based Approaches: TSBP, DAPROG

As noted earlier, among the four methods in this study, the CNN and LSTM employ the pre-processed sensor data and the RUL directly in their training, whereas the other two TSBP and DAPROG introduce the HI for the training process. To this end, the HI is defined as follows [25]:

$$HI = h_{HA}(\mathrm{x};\Theta) \text{ or } HI = \alpha + \beta \cdot \mathrm{x} + \varepsilon \quad (1)$$

where x are the selected sensor measurements and α, β are the coefficients to be determined via regression. The HI values are defined as 1 and 0 during the initial and near-failure periods defined properly. Applying the sensor data in these two periods and their corresponding HI values (1 or 0), one obtains the regression coefficients. Once obtained, the HI can be estimated for any sensor data x. The HI data are thus obtained but exhibit significant oscillations over the cycles. To rectify this, the data are fitted to the exponential function, which we call the degradation trajectory, as follows:

$$\hat{HI} = a \exp(bt + c) \quad (2)$$

where a, b and c are the coefficients and t is the cycle. To illustrate this, the HI data and their fitted trajectory are plotted for the four selected training units in Figure 10.

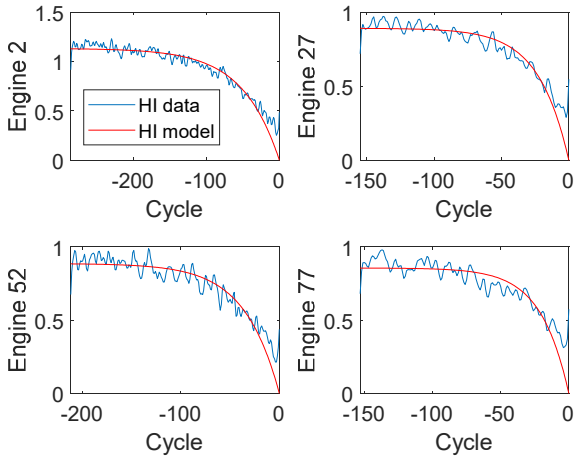


Figure 10. HI data and the degradation trajectory by exponential fitting for selected training units.

As a result, the degradation trajectories of all the 100 training units are plotted in Figure 11a. The HI ranges in the initial cycles are as wide as 0.7~1.3, which agrees with the concept that each engine starts with different degrees of initial wear and manufacturing variation [29]. For the test units, the HI data are obtained from the sensor data using Equation (1) until their current cycles as shown in Figure 11b. During HI construction, the same seven sensors are utilized as with the CNN and LSTM models. Note that trajectories like Figure 11a cannot be utilized as the test unit since they have not yet reached the failure level.

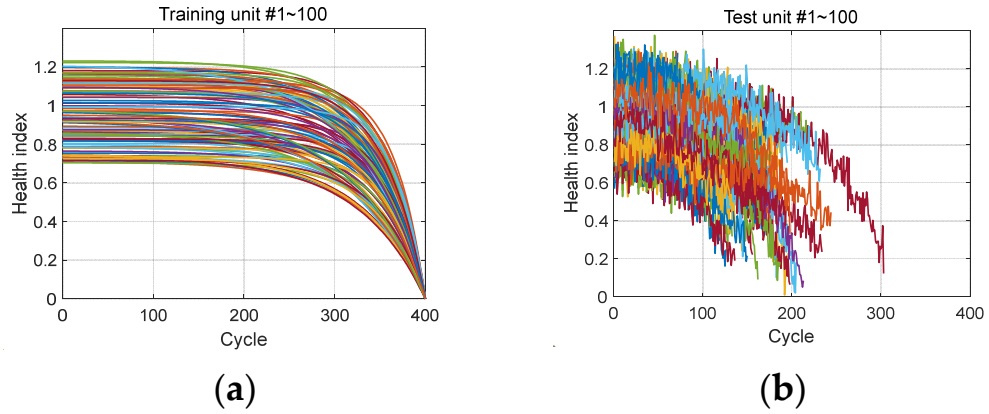


Figure 11. Degradation trajectories: (**a**) HI for the training units and (**b**) HI data for the test units.

3.3.1. RUL Prediction Using TSBP

In the TSBP, the RUL is predicted from the degradation trajectory of the training unit by moving the test data along the time axis and locating at the point which matches with the highest similarity. This is repeated for all the training units to obtain the same number of RULs, which is integrated into a single RUL based on the weighted sum method. However, during the TSBP execution, a problem may sometimes occur where it fails to find any trajectories that match the test data with a high similarity. This takes place frequently when the number of training data is too small, which gives rise to the missing trajectories that suit this end. An example is given in Figure 12 when Ratio $= 0.1$, in which the red curves and the blue asterisks represent the 10 training trajectories and the test data of unit #15, respectively. In the figure, the training curves and the test data barely or do not overlap when the test data moves along the time axis. In this case, a RUL with a large error is yielded.

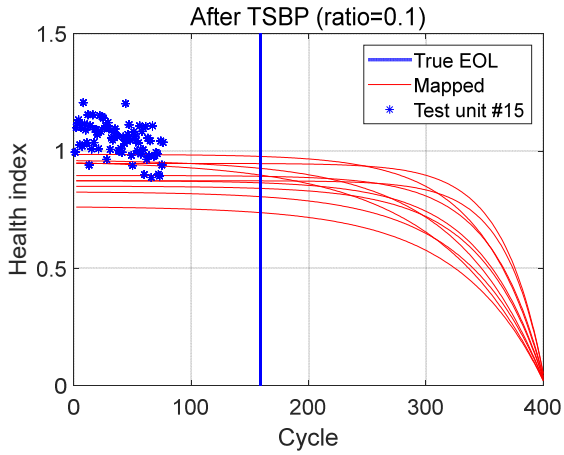


Figure 12. RUL estimation by TSBP when Ratio = 0.1.

3.3.2. RUL Prediction Using DAPROG

The DAPROG takes the same step as the TSBP, which is to uncover the portion of the trajectory curve by moving the test data along the time axis that matches with a high similarity. The DAPROG, however, enables the RUL estimation even when the failure is encountered to reveal the matching curve in the training set. An example is given in Figure 13 for illustration, in which Figure 13a is the ordinary case where we can discover the corresponding (thick) portion in the training (red solid) curve that matches with the test data (blue asterisk), which we call the DTW coverage. Then, the mapping by DTW applies only to this portion of the curve, from which the regression and virtual mapping paths are made, as shown in (c), and are extrapolated into future cycles. Next, the virtual paths are mapped to the test data, as shown in (e), from which the RULs are predicted, which are the cycles between the current cycle and the EOL of the virtual curves. Note that the EOLs predicted by virtual paths (x-intercepts of many red curves) are not too different from the true EOL. When the matching curve is not found as shown in (b), applying the DTW algorithm results in the trivial solution, with the DTW coverage involving the same cycles as the test unit (thick portion of the curve) and the warping path becoming simply the linear line, with the coefficient of determination being 1, as shown in (d). The resulting virtual RTF curve becomes (f) where the EOL simply becomes that of the training curve. This is a similar situation to the one addressed in the TSBP.

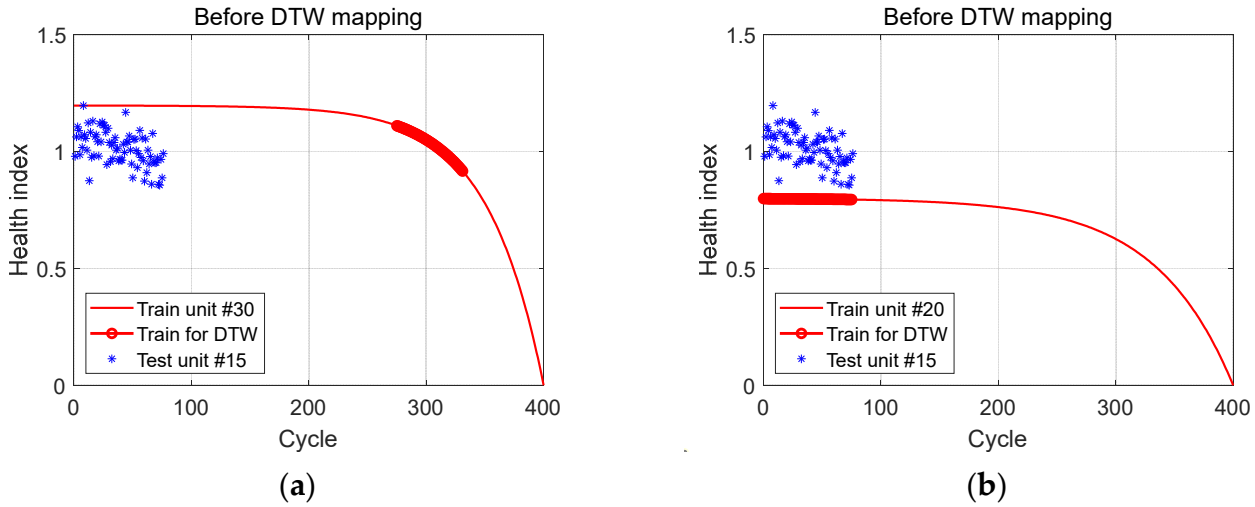


Figure 13. *Cont.*

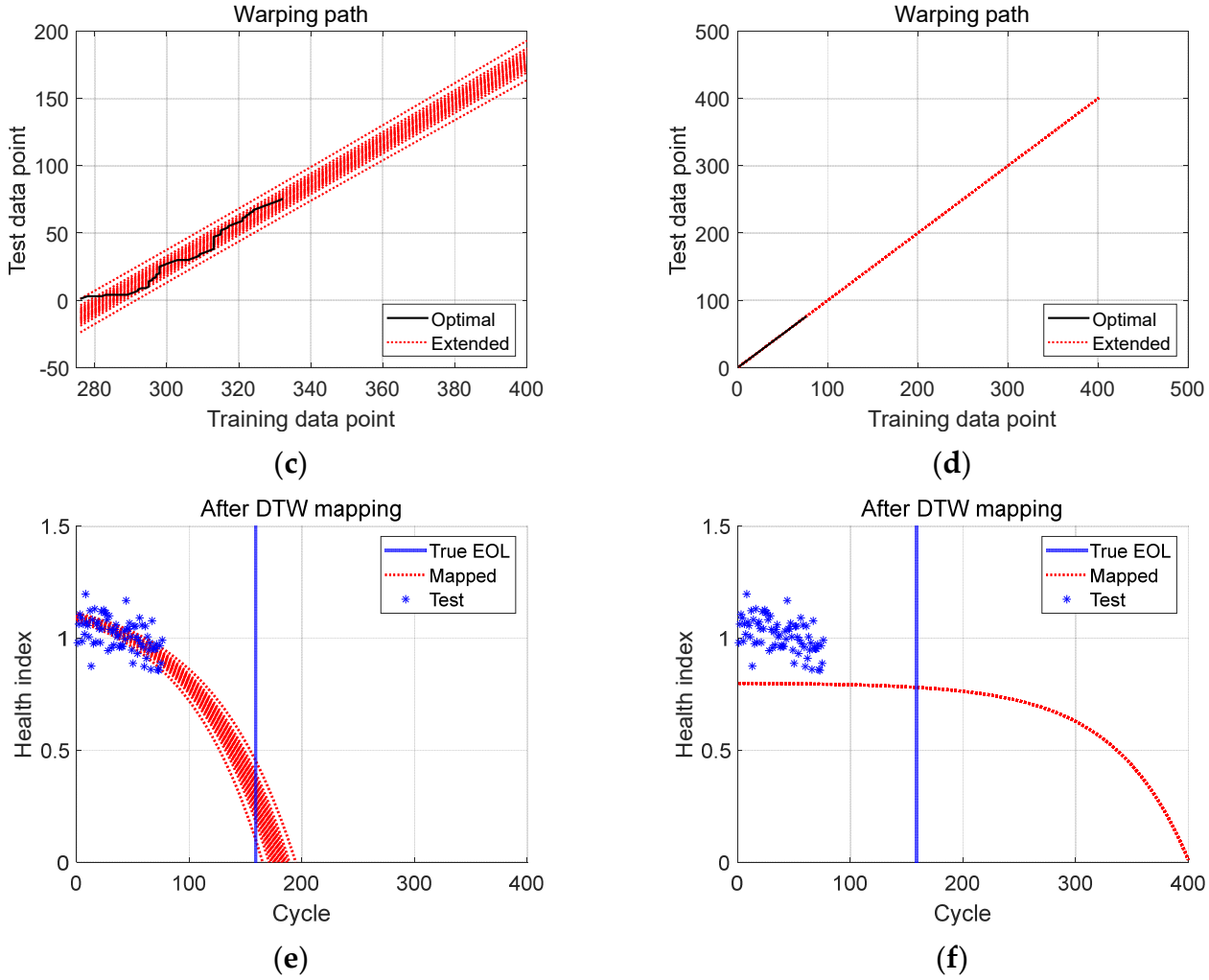


Figure 13. DAPROG results for: (**a**,**c**,**e**) Ordinary caseand (**b**,**d**,**f**) Non-matching case.

To solve this problem in the DAPROG, an additional procedure is carried out to uncover the DTW coverage; this procedure involves finding the matching portion in the training curve by not only moving the test data along the cycle (horizontally) but also along the HI (vertically). In fact, it is a 2D search for the location in the training curve. The principle is that the smaller the DTW distance and the higher the coefficient of determination, the greater the DTW mapping results we obtain. For this objective, optimization is performed to minimize the ratio of DTW distance and the coefficient of determination, as represented by Equation (3).

$$\begin{aligned} v^* &= \underset{v}{\operatorname{argmin}}(d_{DTW}/rsqr) \\ Y_{adj} &= Y - v^* \end{aligned} \tag{3}$$

Then, the location of the test data and the corresponding DTW coverage (thick portion of the curve) become those in Figure 14a, from which the warping path and virtual RTF curves are given by (b) and (c), respectively. In Figure 14c, the RULs are estimated as the distance between the EOL and the end cycle of the test data.

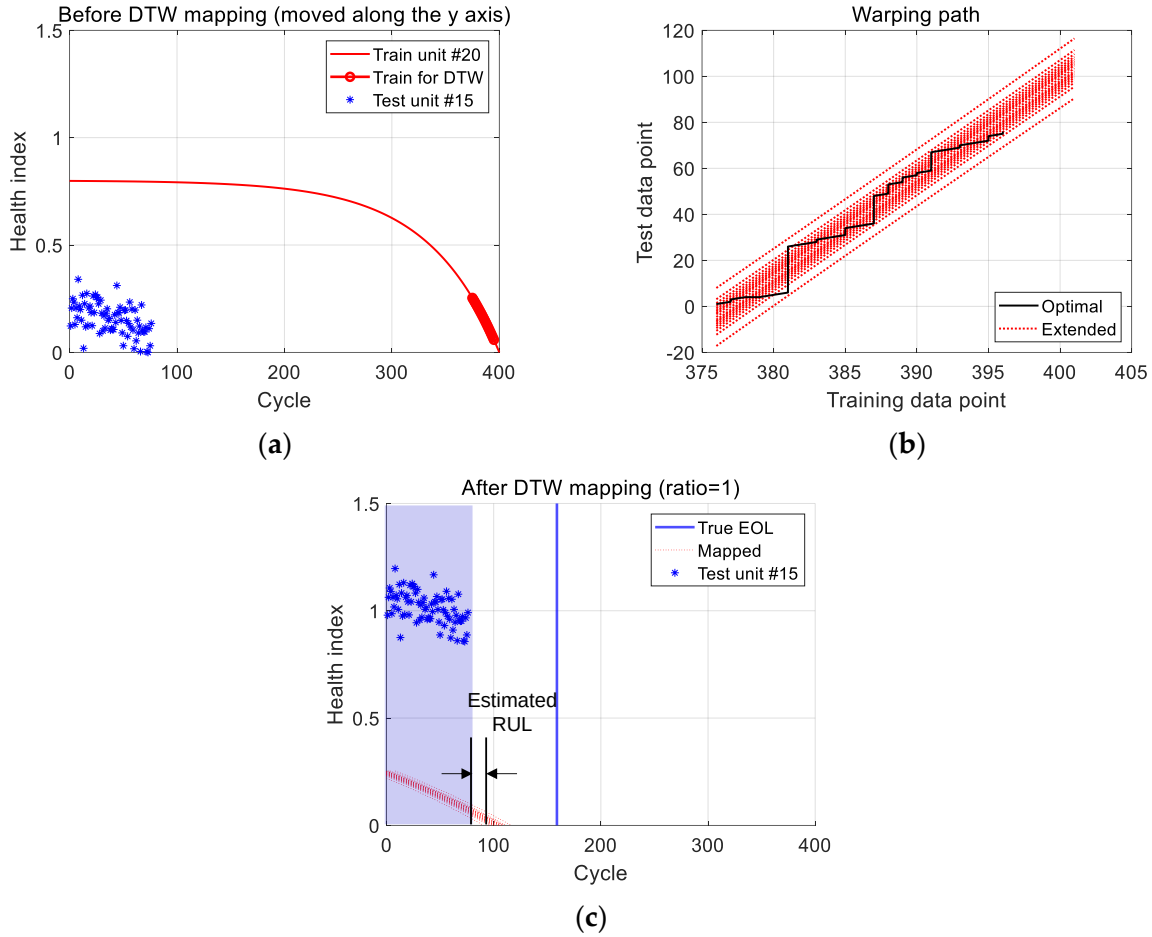


Figure 14. (**a**–**c**) DAPROG by 2D search to find out appropriate DTW coverage.

3.4. RUL Prediction under Data Deficiency

To investigate the impact of data deficiency, the size of the training dataset is intentionally reduced to a fraction of its original size (the number of training units), $N_{train} = 100$. This process is depicted in Figure 15, where the Ratio of the used training data is progressively increased from 0.1 to 0.9 in increments of 0.1. A smaller ratio signifies a more severe degree of data deficiency. Upon completion of training, these models are used to predict the RUL of test data, with accuracy evaluated against actual values. While the training dataset is reduced, the number of test units remains constant at the original number, $N_{test} = 100$. Given that the reduced training data are randomly selected, the prediction results may vary with each attempt. To account for this variability, the prediction process is repeated 300 times, allowing for the calculation of mean and standard deviation being used as the performance metrics. The process consists of the following steps:

1. Data Selection: A subset of the training data is randomly selected in the amount of $N_{train} \times$ Ratio.
2. Data Normalization and Sensor Selection: Sensor data are normalized, and sensors with high correlation to the cycles are selected to enhance training efficiency.
3. Model Training: The selected training data are used to train the model. For TSBP and DAPROG, this involves constructing the HI degradation trajectory for training units using specific equations and estimating the HI for test units. In ANN methods, model parameters such as weights and bias are determined using the training data, with sensor data as inputs and the RUL as outputs.
4. RUL Prediction: The RUL for each test unit is predicted using the trained model. TSBP obtains the RUL by applying similarity measures to the HI trajectory of training units, resulting in as many RUL estimates as there are training data. DAPROG generates

virtual training curves for each test unit using DTW, yielding multiple RUL estimates. In both methods, a weighted sum approach is used to aggregate these RUL estimates, assigning higher weights to more similar trajectories. For ANN models, the RUL is directly calculated by applying the sensor data of the test unit to the trained model.

5. Performance Evaluation: The RMSE are computed using the 100 predicted and actual RULs to assess method performance.

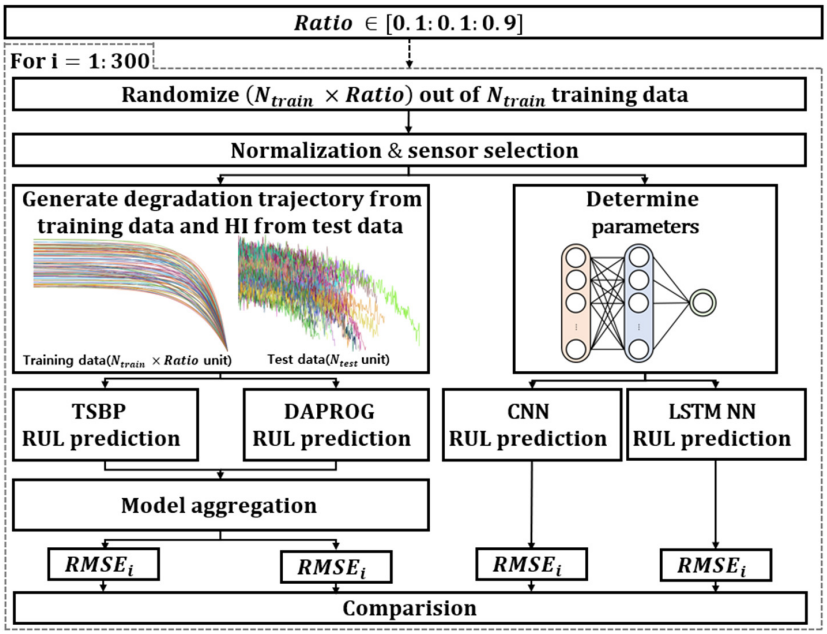


Figure 15. Flowchart of analysis for data deficiency.

Since this process is repeated 300 times for each ratio and method, a set of 300 RMSEs is obtained, representing the performance variance due to the random selection of training data.

3.5. Comparative Discussion: Overall Performance

As outlined earlier, our experiments, depicted in Figure 16, demonstrate the RMSE of RUL prediction across every scenario for each method. The box plots at each ratio illustrate the median, first, and third quartiles, with the whiskers indicating the maximum and minimum values, excluding outliers marked by red crosses. These results clearly show that an increase in the training data ratio leads to a decrease in RMSE, thereby improving the accuracy of RUL prediction for all four-methods, as anticipated.

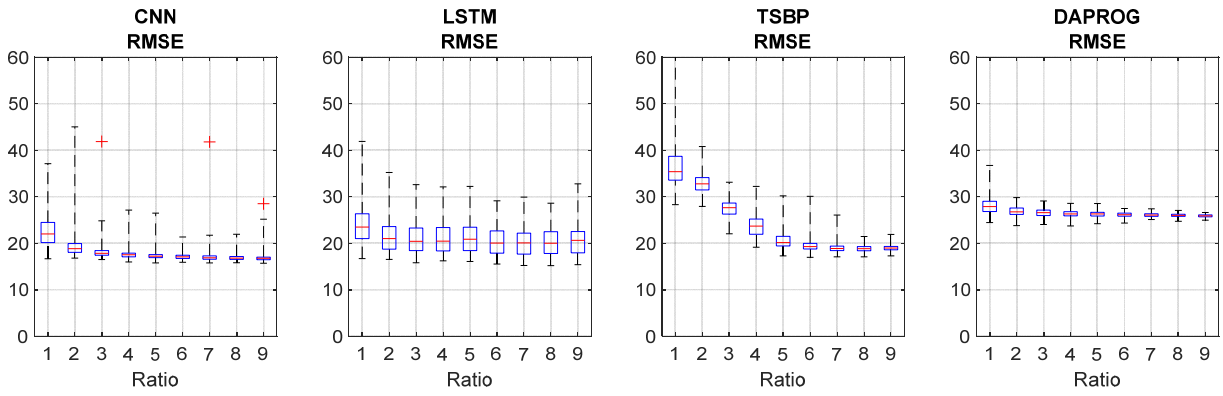


Figure 16. Performance comparisons of each method.

In view of the mean values of the RMSE result, the CNN and LSTM models appear to perform much better and are more stable than TSBP and DAPROG, with most values being below 20. The mean of TSBP is the worst in the small ratios but quickly decreases to less than 20 as the ratios increase, whereas the mean of DAPROG stays near 25. However,, the variance of the RMSE is much greater in the CNN and LSTM across the entire ratios, not only in the small ratios (10% to 30%) but even in the near original dataset. The reason may be due to the inherent randomness of the neural network in every training attempt. On the other hand, those of TSBP and DAPROG show a decrease as the ratio increases and seemingly converges to a single value.

Overall, it can be concluded that DAPROG is the most consistent method across various scenarios in view of the mean and variance. This consistency is further highlighted in Table 1, which numerically details the performance in data-limited scenarios (ratios of 10% to 30%). Bold letters indicate the best performance among four methods for each ratio. DAPROG maintains a smaller variance than other methods throughout all ratio ranges, indicating its robustness and lower susceptibility to fluctuations by the selected training data.

Table 1. Comparison of RUL estimation results for ratio 10~30%.

Ratio	Metric		CNN	LSTM	TSBP	DAPROG
10%	**RMSE**	**Mean**	**22.46**	24.10	36.38	27.96
		std	3.01	4.56	4.18	**1.67**
20%	**RMSE**	**Mean**	**19.28**	21.57	32.94	26.82
		std	2.37	3.46	2.19	**1.05**
30%	**RMSE**	**Mean**	**18.13**	21.03	27.46	26.51
		std	1.78	3.22	1.84	**0.84**

Figure 17a,b present the 300 RUL estimates for two engines, #2 and #10, which are chosen arbitrarily in the test dataset of 100 engines. The consistency of DAPROG is more evident in this figure, showcasing the smaller variance for all the ratios than the CNN and LSTM, as shown by the box plots. The green horizontal lines show the true RULs for engine #2 and #10, respectively.

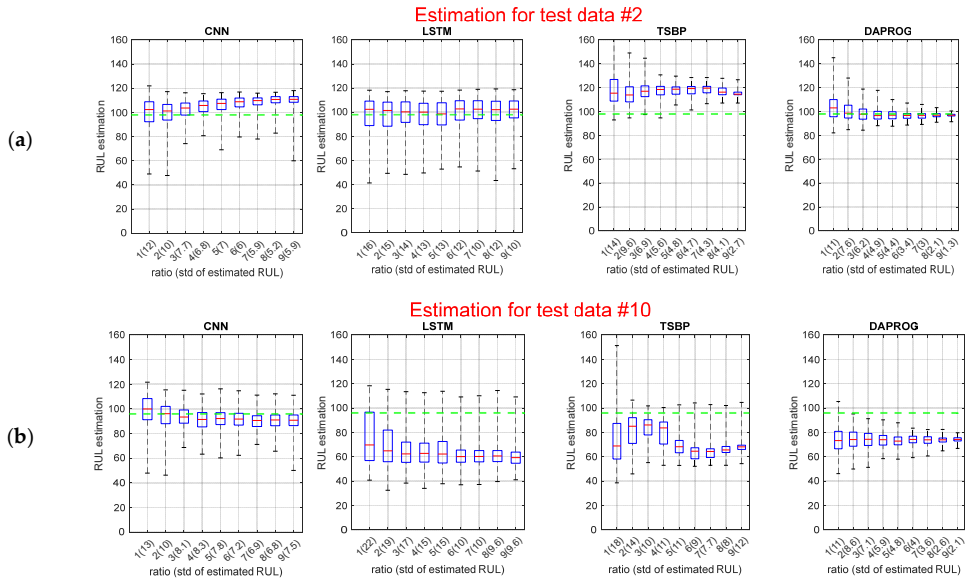


Figure 17. RUL estimation for the test data (**a**) #2 and (**b**) #10.

As previously mentioned, the RUL prediction for each test unit (or engine) is conducted across 2700 scenarios and consists of nine ratios multiplied by 300 repetitions. Therefore, we can obtain the 2700 RUL estimates, from which the variance can be calculated for each unit. This is used as a measure of how consistent the method is in RUL estimation. Since we have 100 test units, we obtain 100 variance values. Figure 18 presents a histogram of the variance for the 100 test units for the four methods. The numbers above each figure and the red vertical line indicate the average variance of the estimates. A lower average variance suggests a more consistent estimation. Notably, DAPROG achieved the lowest values, underscoring its superior consistency in RUL estimation.

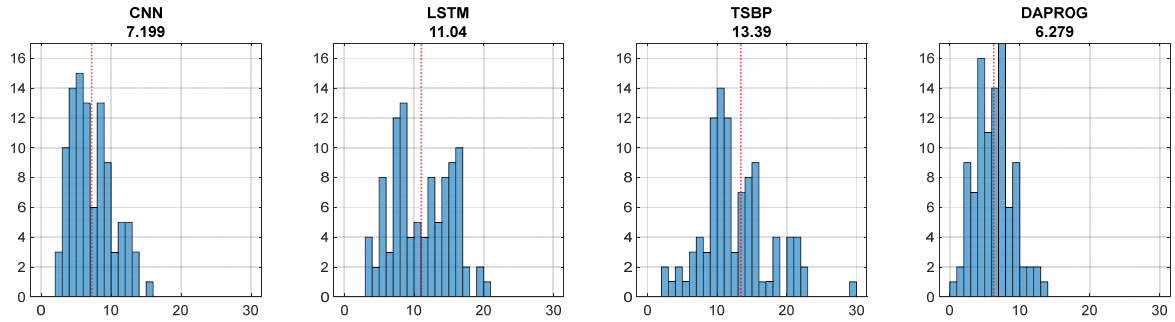


Figure 18. Variance of estimates for 100 test data.

When it comes to scenarios with limited data, methods like DAPROG, although slightly less accurate than ANN methods, demonstrate competitive potential. This is particularly noteworthy considering the challenges associated with the inherent randomness and the non-intuitive nature of ANN methods. Approaches like TSBP and DAPROG, with their more deterministic and interpretable frameworks, offer compelling alternatives in these contexts.

On the other hand, the critical drawback of the ANN method is its randomness, manifesting in aspects like weight initialization and the sequence of data presentation during the training process. This randomness can lead to variability in model performance, even under similar conditions. Moreover, the decision-making process within ANN methods, especially in deep learning models, is often unclear. The multiple layers of computations in these models obscure the transformation of input data into predictions, contributing to their 'black box' nature. This lack of transparency can be a significant hindrance in fields where understanding the rationale behind decisions is crucial. Another challenge in implementing ANN method is the complexity involved in designing their architecture. Selecting the appropriate architecture is not only crucial for optimal performance but also requires substantial time and expertise. There is no universal solution; the determination of the number of layers, types of layers, number of neurons, optimization methods, and mini-batch sizes demands considerable experimentation and domain-specific knowledge. While our experiments successfully implemented architectures based on Li and Zheng's models, it is worth noting that different architectures might not yield equally effective results.

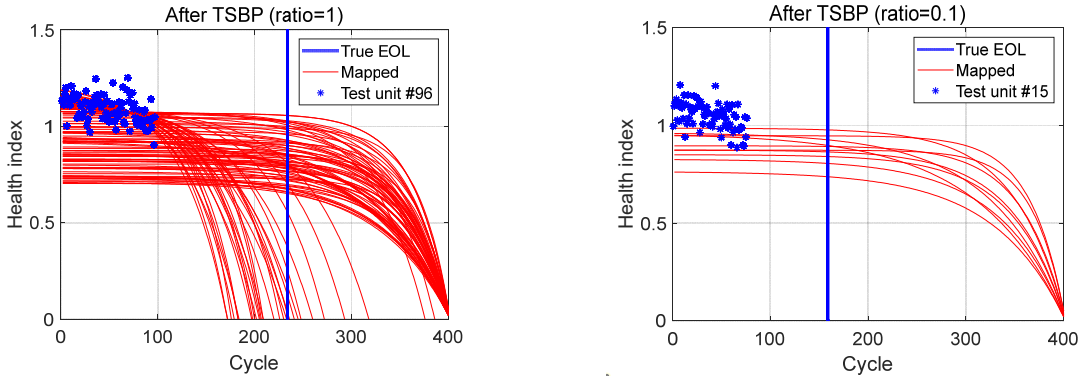
However, it is observed that, when the data ratio exceeds 0.4, DAPROG does not show significant performance improvements, in contrast to TSBP, which rapidly improves and eventually outperforms DAPROG. In the meantime, CNN and LSTM models consistently demonstrate strong and progressively improving performances. The reason for the differences between TSBP and DAPROG, and their implications in various scenarios, are explained in further detail in the following section.

3.6. Comparative Discussion: TSBP vs. DAPROG

Both TSBP and DAPROG methods share a foundational approach in RUL prediction, focusing on identifying similar trajectories within the training dataset. However, they diverge

significantly in their specific methodologies. While DAPROG creates multiple virtual Run-To-Failure (RTF) curves to predict various RULs, TSBP predicts a single RUL value.

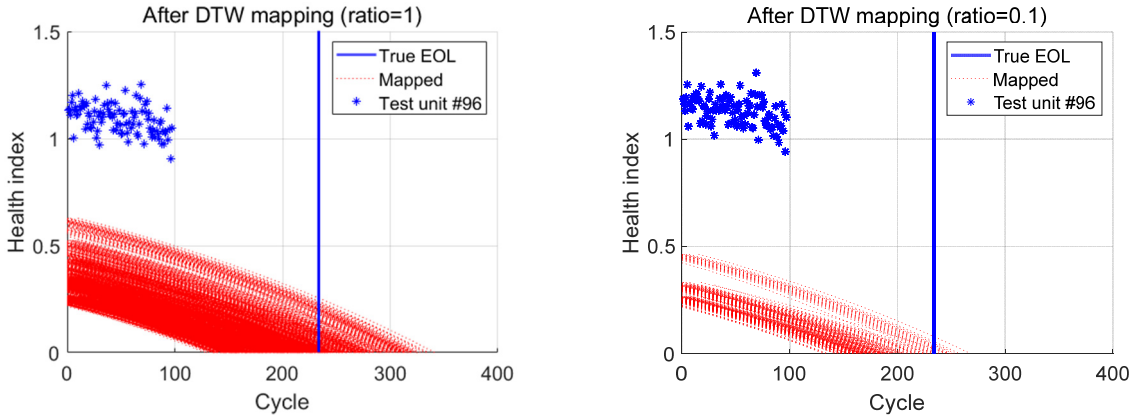
Figure 19a illustrates TSBP's prediction method using full training data (Ratio = 1). Here, test unit #96 (blue asterisks) is compared against similar segments of training data trajectories (red curves). Trajectories that fail to align with the test data, particularly those ending at 400 cycles, are excluded from the final RUL prediction. For instance, TSBP's RUL estimate of 120.7, closely approximating the actual value of 137, underscores its effectiveness with ample data. However, as seen in Figure 19b with a limited data ratio (0.1), the absence of matching trajectories in the training data significantly affects the accuracy of RUL predictions, as reflected by the high RMSE and variance for training ratios below 0.3 (Figure 16).



(**a**) (**b**)

Figure 19. RUL estimation of TSBP when (**a**) Ratio = 1 and (**b**) Ratio = 0.1.

In contrast, DAPROG demonstrates a consistent performance regardless of the training data ratio. Figure 20a,b showcase the virtual RTF curves for full (Ratio = 1) and reduced (Ratio = 0.1) data ratios. DAPROG's consistent RUL distributions, irrespective of data availability, are attributed to its methodology of vertically adjusting test data to enhance DTW (Dynamic Time Warping) coverage and extrapolating multiple virtual paths. This approach results in uniform RUL predictions with lower variance, which is evident in Figure 16.



(**a**) (**b**)

Figure 20. Comparison of RUL estimation based on DAPROG by (**a**) ratio = 1 and (**b**) ratio = 0.1.

DAPROG's performance may deteriorate with lower-quality test data, particularly if the data are too short or only reflect the early stages of degradation. This is linked to the differences in how training trajectories are matched to test data—TSBP using Euclidean

distance and DAPROG employing DTW distance. While Euclidean distance is more intuitive, relying on physical proximity for similarity, DTW can distort similarities if data series patterns vary drastically. In cases where test data are limited or early in the degradation phase, the resulting warped path might be too short for precise future extrapolation. This limitation, potentially affecting some of the 100 test datasets, can impact DAPROG's overall effectiveness.

When data deficiency is severe (low ratio), DAPROG's strength lies in its ability to conduct additional searches for matching trajectories and generate virtual RTF data. Conversely, with abundant data (high ratio), TSBP might outperform due to the increased probability of finding suitable training trajectories.

4. Conclusions

Estimating the Remaining Useful Life (RUL) of systems is an essential aspect of predictive maintenance that is crucial for enhancing both reliability and efficiency proactively. Because of the complexities of real-world application, data-driven approaches are widely adopted for RUL estimation. However, the performance of these methods heavily relies on the availability of substantial training data, which can be challenging to acquire due to the costs and complexities involved, especially in cases like obtaining Run-To-Failure (RTF) datasets for turbofan engines. Thus, addressing data scarcity is vital when applying prognostic algorithms for RUL prediction.

In our study, we introduced DAPROG, a novel approach that utilizes Dynamic Time Warping (DTW) to map data and generate virtual RTF datasets that is specifically aimed at mitigating issues related to data deficiency. We evaluated DAPROG's efficacy against other prominent algorithms in the field, including CNN, LSTM, and TSBP, using the CMAPSS dataset. This dataset includes multiple training and test subsets, allowing us to simulate data deficiency scenarios by selectively using portions of the training data for RUL prediction.

Our results show that, while the CNN model exhibits the best overall performance, DAPROG presents a competitive alternative, particularly in data-deficient conditions (with a data ratio of 10% to 30%). Notably, DAPROG displays less variance in its predictions, indicating a more consistent performance relative to the other methods. This consistency is especially significant considering the challenges of randomness and the non-intuitive nature inherent in artificial neural network (ANN) methods, as well as the complexities involved in designing optimal ANN architectures.

Nevertheless, DAPROG's performance improvement reaches a plateau in scenarios with abundant data where the ratio exceeds 30%, indicating a limitation of this approach. Future research could focus on developing strategies to assign differentiated weights to the multitude of generated RTF datasets, allocating lower weights to less credible augmented RTF data and higher weights to more credible ones, thereby potentially optimizing performance in every circumstance. Furthermore, integrating Dynamic Time Warping (DTW) with Artificial Neural Network (ANN) algorithms could significantly enhance predictive capabilities. Integrating DTW may pave the way for advancements in the field of Remaining Useful Life (RUL) prediction, potentially leading to more robust and accurate predictive models.

Author Contributions: Methodology, J.S., S.H.C. and S.K.; Formal analysis, J.S., S.H.C. and S.K.; Writing—original draft, J.S.; Writing—review & editing, J.N. and J.-H.C.; Supervision, J.N. and J.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF), grant funded by the Korea government (MSIT) (No. 2020R1A4A4079904).

Data Availability Statement: The C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is publicly available and can be accessed through NASA's Prognostics Center of Excellence Data Repository. The dataset includes simulated engine performance data under various

operational conditions and is widely used for research in predictive maintenance and prognostics. Detailed information, along with download links, can be found at the NASA repository.

Conflicts of Interest: Author Seong Hee Cho was employed by the company Korea Shipbuilding & Offshore Engineering (KSOE). The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhao, F.; Tian, Z.; Zeng, Y. Uncertainty Quantification in Gear Remaining Useful Life Prediction through an Integrated Prognostics Method. *IEEE Trans. Reliab.* **2013**, *62*, 146–159. [CrossRef]
2. An, D.; Kim, N.H.; Choi, J.H. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliab. Eng. Syst. Saf.* **2015**, *133*, 223–236. [CrossRef]
3. Lee, D.; Park, H.J.; Lee, D.; Lee, S.; Choi, J.H. A Novel Kalman Filter-Based Prognostics Framework for Performance Degradation of Quadcopter Motors. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3332389. [CrossRef]
4. Gebraeel, N.; Lawley, M.; Liu, R.; Parmeshwaran, V. Residual life predictions from vibration-based degradation signals: A neural network approach. *IEEE Trans. Ind. Electron.* **2004**, *51*, 694–700. [CrossRef]
5. Mahamad, A.K.; Saon, S.; Hiyama, T. Predicting remaining useful life of rotating machinery based artificial neural network. *Comput. Math. Appl.* **2010**, *60*, 1078–1087. [CrossRef]
6. Babu, G.S.; Zhao, P.; Li, X.-L. Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. In Proceedings of the 21st International Conference on Database Systems for Advanced Applications (DASFAA 2016), Dallas, TX, USA, 16–19 April 2016; Proceedings, Part I 21. Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
7. 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]
8. 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 2017), Piscataway, NJ, USA, 19–21 June 2017; pp. 88–95. [CrossRef]
9. Jayasinghe, L.; Samarasinghe, T.; Yuen, C.; Ni Low, J.C.; Ge, S.S.; Lanka, S. Temporal Convolutional Memory Networks for Remaining Useful Life Estimation of Industrial Machinery. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, Australia, 13–15 February 2019.
10. 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), Trento, Italy, 7–9 September 2020.
11. Hong, C.W.; Lee, C.; Lee, K.; Ko, M.-S.; Kim, D.E.; Hur, K.C.W. Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors* **2020**, *20*, 6626. [CrossRef] [PubMed]
12. Zhan, Y.; Kong, Z.; Wang, Z.; Jin, X.; Xu, Z. Remaining useful life prediction with uncertainty quantification based on multi-distribution fusion structure. *Reliab. Eng. Syst. Saf.* **2024**, *251*, 110383. [CrossRef]
13. Xiang, F.; Zhang, Y.; Zhang, S.; Wang, Z.; Qiu, L.; Choi, J.H. Bayesian gated-transformer model for risk-aware prediction of aero-engine remaining useful life. *Expert Syst. Appl.* **2024**, *238*, 121859. [CrossRef]
14. Theissler, A.; Pérez-Velázquez, J.; Kettelgerdes, M.; Elger, G. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 107864. [CrossRef]
15. Fink, O.; Wang, Q.; Svensén, M.; Dersin, P.; Lee, W.J.; Ducoffe, M. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng. Appl. Artif. Intell.* **2020**, *92*, 103678. [CrossRef]
16. Hu, C.; Youn, B.D.; Kim, T.; Wang, P. A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data. *Mech. Syst. Signal Process.* **2015**, *62*, 75–90. [CrossRef]
17. An, D.; Choi, J.-H.; Kim, N.H. Prediction of remaining useful life under different conditions using accelerated life testing data. *J. Mech. Sci. Technol.* **2018**, *32*, 2497–2507. [CrossRef]
18. Akrim, A.; Gogu, C.; Vingerhoeds, R.; Salaün, M. Self-Supervised Learning for data scarcity in a fatigue damage prognostic problem. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105837. [CrossRef]
19. Kim, S.; Kim, N.H.; Choi, J.H. Prediction of remaining useful life by data augmentation technique based on dynamic time warping. *Mech. Syst. Signal Process.* **2019**, *136*, 106486. [CrossRef]
20. Wang, T.; Lee, J. *Trajectory Similarity Based Prediction for Remaining Useful Life Estimation*; University of Cincinnati: Cincinnati, OH, USA, 2010.
21. Li, H.; Zhao, W.; Zhang, Y.; Zio, E. Remaining useful life prediction using multi-scale deep convolutional neural network. *Appl. Soft Comput.* **2020**, *89*, 106113. [CrossRef]
22. Deng, F.; Bi, Y.; Liu, Y.; Yang, S. Remaining Useful Life Prediction of Machinery: A New Multiscale Temporal Convolutional Network Framework. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3200093. [CrossRef]

23. 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 2018, Chongqing, China, 26–28 October 2018; pp. 1037–1042. [CrossRef]
24. She, D.; Jia, M. A BiGRU method for remaining useful life prediction of machinery. *Measurement* **2020**, *167*, 108277. [CrossRef]
25. Wang, T.; Yu, J.; Siegel, D.; Lee, J. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; pp. 1–6. [CrossRef]
26. Müller, M. Dynamic Time Warping. In *Information Retrieval for Music and Motion*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 69–84. [CrossRef]
27. Liu, Y.; Frederick, D.K.; Decastro, J.A.; Litt, J.S.; Chan, W.W. *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*; NASA Glenn Research Center: Cleveland, OH, USA, 2012; Volume 2012-217432, pp. 1–40.
28. Saxena, A.; Goebel, K. *Turbofan Engine Degradation Simulation Data Set*; NASA Ames Research Center: Mountain View, CA, USA, 2008.
29. Saxena, A.; Goebel, K. *PHM08 Challenge Data Set*; NASA Ames Research Center: Mountain View, CA, USA, 2008.

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.