aerospace

Article

Remaining Useful Life Prediction for Turbofan Engine Using SAE-TCN Model [†]

Xiaofeng Liu *, Liuqi Xiong, Yiming Zhang and Chenshuang Luo

School of Transportation Science and Engineering, Beihang University, Beijing 102206, China

* Correspondence: liuxf@buaa.edu.cn

[†] This paper is an extended version of our paper published in 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021.

Abstract: Turbofan engines are known as the heart of the aircraft. The turbofan's health state determines the aircraft's operational status. Therefore, the equipment monitoring and maintenance of the engine is an important part of ensuring the healthy and stable operation of the aircraft, and it is vital to monitor the remaining useful life (RUL) of the engine. The monitored data of turbofan engines have high dimensions and a long time span, which cause difficulties in predicting the remaining useful life of the engine. This paper proposes a residual life prediction model based on Autoencoder and a Temporal Convolutional Network (TCN). Among them, Autoencoder is used to reduce the dimension of the data and extract features from the engine monitoring data. The TCN network is trained on the obtained low-dimensional data to predict the remaining useful life. The model mentioned in this article is verified on the NASA public data set (C-MAPSS) and compared with common machine learning methods and other deep neural networks. The SAE-TCN model achieved better scores on the FD001 independent testing data set with an RMSE of 18.01 and a score of 161. The average relative error of the model relative to other common learning models is 0.9499 in RMSE and 0.2656 in Scoring Function. The experimental results show that the model proposed in this paper performs the best in the evaluation, and this conclusion has important implications for engine health.

Keywords: deep learning; turbofan engine; remaining useful life prediction; Temporal Convolutional Network; autoencoder; target generation

Citation: Liu, X.; Xiong, L.; Zhang, Y.; Luo, C. Remaining Useful Life Prediction for Turbofan Engine Using SAE-TCN Model. *Aerospace* **2023**, *10*, 715. https://doi.org/10.3390/aerospace10080715

Academic Editor: Sujin Bureerat

Received: 18 June 2023
Revised: 7 August 2023
Accepted: 9 August 2023
Published: 16 August 2023

1. Introduction

The turbofan engine is a critical system of an aeroplane. Its health determines whether the aircraft can run stably and reliably [1]. Determining the remaining useful life (RUL) of a turbofan engine is important for equipment monitoring and maintenance [2]. Traditional maintenance of the turbofan engine adopts simple passive and active strategies; that is, parts of the turbofan engine are repaired or replaced when the engine fails, or the parts are uniformly updated when the turbofan engine runs to the default time limit. Due to the complex structure of the turbofan engine, the operating state is easily affected by many environmental factors, and its service life range is relatively large. The fixed default time limit setting cannot balance the safety and economy of the maintenance strategy. The research on turbofan engine failure prediction and health management (Prognostics and Health Management, PHM) has been widely carried out in recent years. The remaining life indicates how long the turbofan engine can function before it fails. The remaining life prediction is an important link in the field of engine failure prediction and health management [3–8]. Accurate prediction on the remaining life of a turbofan engine can provide a reference for equipment overhaul and maintenance, thereby avoiding cost increases caused by excessive maintenance or potential safety hazards caused by neglected maintenance [9]. By accurately predicting the remaining service life and monitoring the health status of the aero engine in a timely manner, it can guide the engine management personnel to take reasonable maintenance measures to maintain the reliability of an aircraft while reducing the maintenance cost of the engine [10]. Therefore, establishing a suitable model to predict the remaining life of the engine in real time is a common goal of academia and industry, and it is of great significance to ensure the safety of aircraft flight.

RUL prediction methods in the PHM field are mainly divided into three categories: model-driven approaches, data-driven approaches, and combined approaches [11–13]. The model-driven method, also known as physical modeling, refers to a method of predicting the probability of equipment failure and the remaining service life by establishing an accurate physical model that can describe the operating conditions of the equipment system. The principle of establishing this physical model is to restore the natural state of equipment operation as much as possible. Therefore, once this physical model is established, a relatively accurate and precise prediction of the remaining life of the equipment and the occurrence of failures can be obtained. Model-driven forecasting methods are also widely used in research and practice. A model-based approach was applied to construct a real-time algorithm for predicting and detecting bearing and gear failures. Specifically, they combined sensor data with a constructed physical model to achieve the final prediction. In the model's construction, the authors used the Yu–Harris model to describe the state of bearings and gears before the crack occurred, and they used the crack growth model to describe the state of the equipment after the crack occurred. The results of the study found that using model-based estimation can make good predictions throughout the life of the equipment [14]. A new approach for equipment remaining life prediction based on a dynamical system approach to damage evolution is proposed. This approach does not rely on the knowledge of specific damage physics and is suitable for systems where damage evolves on a slower timescale than directly observable dynamics. The results of the study show that the model described in the article is very accurate in predicting the remaining life of spur gears [15]. The physical models are highly dependent on expert knowledge of turbofan engines to reveal the component degradation. Data-driven approaches construct predictive models provided data from multi-source sensors. Statistics, machine learning, and deep learning methods show their powerful abilities in this field. The combined approaches usually use data-driven methods to learn and tune the physical model parameters.

Accurately modeling a turbofan engine according to its physical law is challenging due to the complex engine structure. The data-driven method has been widely studied. The data-driven method is also an important method for remaining life prediction. This type of method does not require the establishment of a physical model that can accurately describe the dynamic change process of the component but, rather, is directly based on the operating parameters of the component obtained from the sensors. Historical data of the real operating state embody the information of the degradation process. Therefore, data-driven approaches usually require sufficient historical data to train the model. When the amount of data is large enough, the data-driven method can accurately identify the correlation between sensor parameters and equipment operating status from the data and make accurate predictions for the remaining life of the equipment. Initially, researchers used continuous-time hidden Markov models (HMM) [16] based on the time of normal operation, the time when the engine begins to degenerate, and the time of engine failure for the RUL prediction of turbofan engines. Recurrent Neural Network (RNN) models are also used to extract information from the time series [17,18]. RNN models transfer the state of the previous moment to the output of the current moment through its recursive structure. Some information will be forgotten in the long-term iterative process and will have weaker influence in the output; therefore, a Long Short-Term Memory (LSTM) model [19] is introduced. LSTM brings control gates to the RNN. These control gates cooperate to transmit effective long-term information. A straightforward and high-quality method for estimating remaining useful life based on random forest regression is proposed. Specifically, they first screened out the feature variables closely related to the remaining life of the engine from NASA's C-MAPSS data set through Lasso regression and further input these features into the random forest regression model to realize the prediction of the remaining service life of the engine [20]. However, these conventional methods cannot cope with the turbofan engine's degradation process, which is non-linear, time-variant, and high-dimensional. These methods are improved to enhance their prediction accuracy and robustness. To extract complex features, a 1D convolutional neural network with a complete convolutional layer and LSTM was proposed [21]. To alleviate the impact of different operating conditions, a CNN-XGB (extreme gradient boosting) method with an extended time window was proposed [22].

Ordinary convolution usually leads to the leakage of future information, and it may violate the truth that the future state of the engine is determined only by its past and present state. The complex dynamic process that comprises an engine state change means longer historical information is important for more accurate prediction. A Temporal Convolutional Network (TCN) is proposed in this paper to deal with these difficulties. In addition to those difficulties, the TCN can process an input sequence of any length to adapt to the dynamics and diversity of engine sensor data. Engine sensors' sampling characteristics, such as frequency and length, may vary with time and scenarios. Last but not the least, a TCN can parallelize calculations to improve the efficiency of model training and prediction. For engine condition monitoring, it is important to find out the remaining life accurately in a short time. Due to the high dimensionality of engine monitoring data, a stacked sparse Autoencoder (SAE) is used in this paper to accomplish data dimension reduction and improve the generalization ability of the model. Therefore, we propose an SAE-TCN turbofan engine remaining useful life prediction model in this paper. An SAE model is first trained to extract useful features from engine monitoring data. After that, a TCN model is trained to predict the RUL based on the reduced-dimensional data from the SAE model. The experimental results show that the SAE-TCN model can predict the RUL more accurately and efficiently by combining the advantages of Autoencoder and the TCN.

The following contents of this paper include three sections. Section 2 presents the structure of the SAE-TCN model. Section 3 includes the experiments and results. Conclusions can be found in the last section.

2. Feature Extraction Based on Stacked Autoencoder Neural Network

2.1. Sparse Autoencoder

2.1.1. Autoencoder

Autoencoder (AE) is a fully connected, multi-layer, unsupervised neural network [23]. Like a general neural network, AE also includes an input layer, three hidden layers, and an output layer. There are the same number of neurons in the input and output layer of AE so that the input data can be reconstructed. At the same time, the number of neurons in the middle hidden layer is arbitrary, which determines the final dimension of the data. Like multi-layer perceptrons, Autoencoder neural networks are trained by back propagation. The input layer and the first two hidden layers form an encoder, while the last two hidden layers and output layer form a decoder. After being well-trained, the encoder part functions as a feature extraction method. The original data can be better expressed as lower-dimensional feature vectors without losing key information.

The AE network can be trained by minimizing the difference between the output data and the input data. The objective function in the training process is measured by Equation (1).

$$J(\theta, \mathrm{x}) = \frac{1}{N}\sum_{\mathrm{n}=1}^{N}\frac{1}{2}\left\|\left(x'\right)^{n} - x^{n}\right\|^{2} \tag{1}$$

where x represents the original data input into the encoder network and z is the output. Then, z is input into the decoder network and $x^{'}$ is output. After training, z is the low-dimensional features the Autoencoder extracts out.

2.1.2. Sparse Autoencoder

In order to extract better feature vectors, it is necessary to add sparsity constraints to the loss function. The purpose is to constrain the way the encoder is reconstructed by adding constraints to the loss function, such as adding a regularization term to the loss function as a penalty term, which enables the autoencoder to learn the sparse features from the input data better [24,25]. This penalty term means a new constraint in the objective function. The sparsity constraint plays an important role in algorithm optimization, which limits most neurons to being suppressed. This method is usually adopted to prevent over-fitting.

A sigmoid function is used in the hidden layer as the activation function that outputs 1 when the neuron is active and 0 when it is inactive. KL divergence is introduced as a regular term, as shown in Equation (2).

$$KL(\rho\|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \tag{2}$$

where $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)} x^{(i)}\right]$, m is the number of training samples, $a_j^{(2)}$ is the j-th neuron in the hidden layer, and x^i is the i-th sample. The greater the difference between ρ and ρ^j, the greater the KL divergence.

Adding the KL divergence into the objective function as a penalty term, we obtain a new objective function, as shown in Equation (3):

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho\|\hat{\rho}_j) \tag{3}$$

where β is the weight of the sparsity penalty.

A stacked autoencoder is constructed by a multi-layer sparse autoencoder; the output of each autoencoder layer is used as the input of the next layer of the encoder, and the obtained features are used as the input of the neural network prediction model to complete the life prediction of the engine. Through the above methods, deep feature extraction is realized, and the extracted feature information is more representative.

2.1.3. Stacked Sparse Autoencoder

The stacked sparse autoencoder has more than one hidden layer in its encoder. It can obtain more representative eigenvalues using layer-by-layer feature extraction. In other words, the output of the previous layer of the encoder is the input of the next layer of the encoder. In this way, the autoencoder becomes deeper. The code of the autoencoder network, which is also the output of the encoder, is input into the Temporal Convolutional Network as the features. The structure is shown in Figure 1.

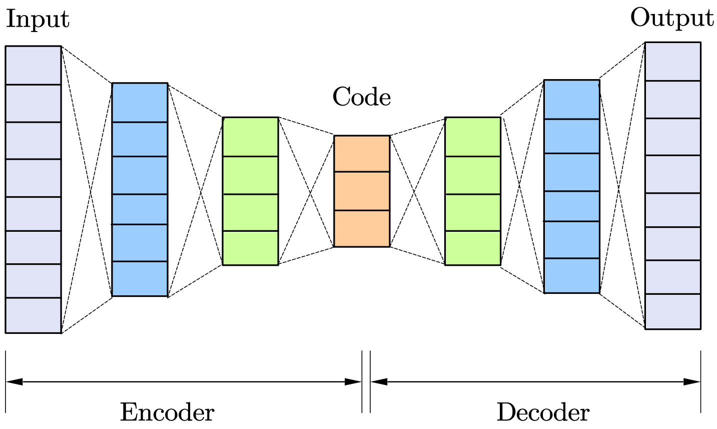

Figure 1. The structure of SAE model.

2.2. *Temporal Convolutional Network*

The Temporal Convolutional Network (TCN) is a variant of a Convolutional Neural Network, which can be used to process sequence modeling tasks [26–29]. Experiments on multiple data sets and tasks show that the TCN outperforms general recursive structures, such as RNN, GRU, LSTM, etc. [30–32]. Causal convolution is used in a TCN to extract and transmit temporal information. In addition, residual network and dilated convolution are combined to capture the long-term correlation between features.

2.2.1. Causal Convolutions

Causal convolution uses the information before a certain time to achieve convolution operation. It means the results are not interfered with by the future data, and the previous information is also completely saved.

Suppose that the input sequence is $s = (s_0, s_1, \cdots, s_t)$ and the output sequence is $y = (y_0, y_1, \cdots, y_T)$. The data dimensions of the input sequence and the output sequence are the same, and the output value depends on the sequence information before t. The input is processed before the convolution operation on the time series, and all inputs before the current moment are replaced with 0 to achieve equal-length causal convolution. The basic structure is shown in Figure 2.

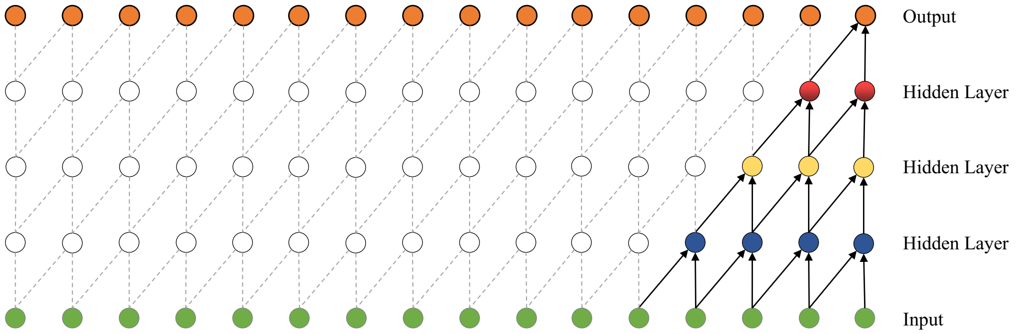

Figure 2. The basic structure of Temporal Convolutional Network (TCN).

2.2.2. Dilated Convolutions

To extract information from longer time-series information, dilated convolutions are used. The upper convolution window interval increases, so even if each hidden layer has the same size as the input sequence, the amount of calculation is greatly reduced. The hyperparameter of the Dilation Rate is added to the expansion convolution, that is, the number of intervals of the convolution kernel, so that each convolution output contains a wide range of information. Dilated Convolutions increase the dilated factor at the top level of convolutions to reduce the computational complexity, even if each hidden layer and the input sequence have the same dimension. Dilation is equivalent to introducing a fixed step between every two adjacent filter taps [28,33,34]. The size of the fixed step can be set as a hyperparameter, namely, the dilation factor. The bigger the dilation, the more information that can be obtained by each convolution output. An illustration of dilated convolutions is shown in Figure 3, where x_T is the input time series, $\hat{y}_T$ is the estimated value, d is the dilation factor in the dilated convolutions, and the size of convolution kernel is 3.

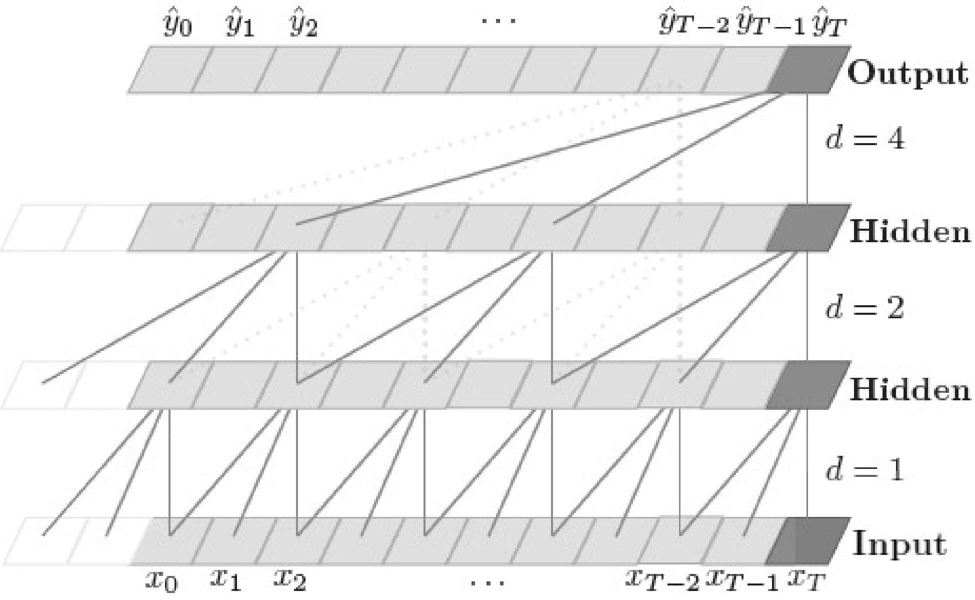
$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_{T-2}$ $\hat{y}_{T-1}$ $\hat{y}_T$
Output
$d = 4$
Hidden
$d = 2$
Hidden
$d = 1$
Input
x_0 x_1 x_2 ... x_{T-2} x_{T-1} x_T

Figure 3. TCN and Dilated Convolutions.

2.2.3. Residual Connections

The residual block is an important network structure in the TCN network. When the number of network layers increases, the phenomenon of gradient disappearance will appear. The residual link structure is summed by x and the $F(x)$ obtained after non-linear changes to form a residual link. After each convolution calculation, the parameter levels are normalized, and then the Rectified Linear Unit (Relu) activation function is used for non-linear calculations. The obtained result is then summed with the input to realize the residual link. Such a structure enables the meaning of the grid to be transmitted across layers, thereby solving the problem of gradient disappearance, and the shallow neural network is extended to a deep neural network.

As the network becomes deeper, gradient disappearance often occurs while training the model. Residual connections are proposed to solve this problem [35]. The residual connections are formed by summing the input features x and its non-linear transformation $F(x)$. After each convolution calculation, the parameter level is normalized, and then the Relu activation function is used for non-linear calculation. The obtained result is then summed with the input to realize the residual connections. The procedure is shown in Figure 4.

The Temporal Convolutional Network is a network structure based on the convolutional structure, which is specifically used to process sequence information. It can not only grasp the overall information of the sequence from local to global, but it can also use convolution instead of recursion, so it also has a great advantage in training speed. But at the same time, there is a shortcoming in which the Temporal Convolutional Network is sensitive to the data set.

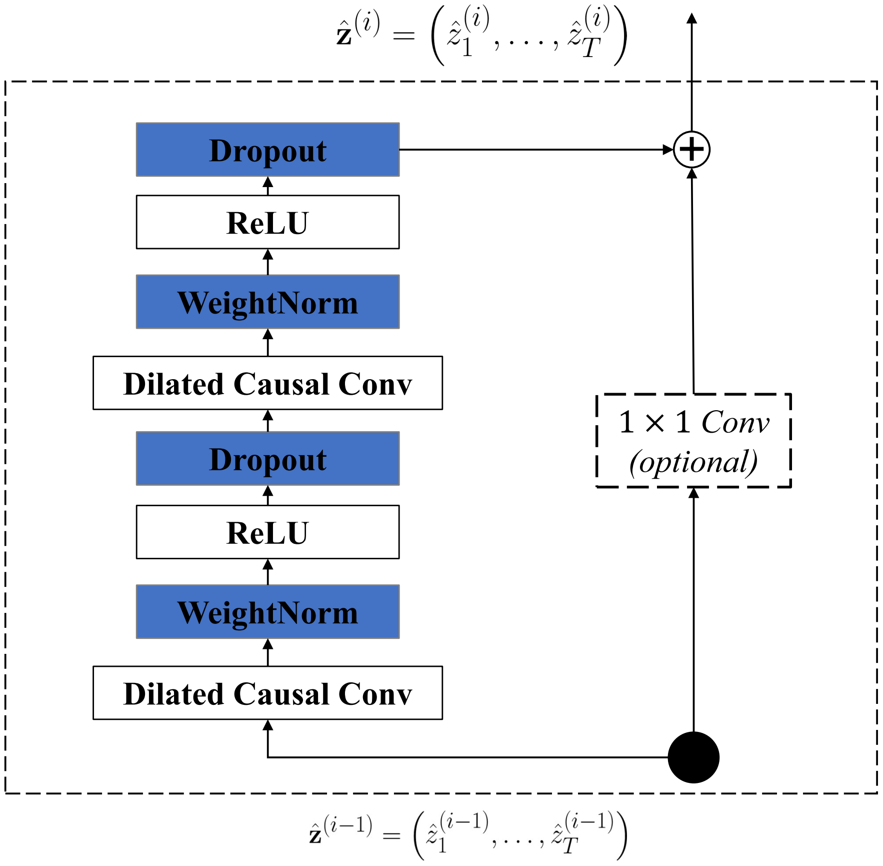
$\hat{\mathbf{z}}^{(i)} = \left(\hat{z}_1^{(i)}, \ldots, \hat{z}_T^{(i)}\right)$
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1 × 1 Conv (optional)
$\hat{\mathbf{z}}^{(i-1)} = \left(\hat{z}_1^{(i-1)}, \ldots, \hat{z}_T^{(i-1)}\right)$

Figure 4. TCN and Residual Connections.

2.3. SAE-TCN Prediction Model

The neural network prediction model used in this paper uses stacked Autoencoders (SAEs) to extract useful features and reduce the data dimensionality. When training the TCN, the network input is the output from the encoder. In this paper, a temporal neural network which consists of multiple temporal residual blocks is used as the processor for model prediction, and six residual blocks (TCN Block) and a fully connected layer are stacked to obtain the final life prediction value. The fully connected layer has 64 neurons, and it finally outputs one-dimensional data, which is the final prediction result of the remaining life. The minimum mean square error (MSE) is used as the loss function to measure the error between the real remaining life and the predicted value. Adaptive moment estimation (Adam) is adopted for model training and parameter optimization. The main structure is shown in Figure 5.

The modeling process based on a temporal convolutional neural network is as follows:

1. Obtaining the monitoring data set of each sensor parameter of the aero engine, performing data preprocessing and feature extraction, and using a series of preprocessing training sets in the time-series convolutional network model for iterative training.
2. Building the engine life prediction model: Preliminary construction of a temporal convolutional neural network model, given initial hyperparameters, including the dimension of the input matrix, the size of the convolution kernel (kernel size), the number of convolutional layers (number of filters), time steps (time steps), dropout rate, epoch, batch size, etc.
3. Engine remaining life prediction: According to the initially established TCN model, input the prepared data set for training and evaluate it on the test set. The minimum mean square error (MSE) is used as the loss function to measure the error between the real remaining life and the predicted value. Import the Adam module as an optimizer for model training and parameter optimization change.

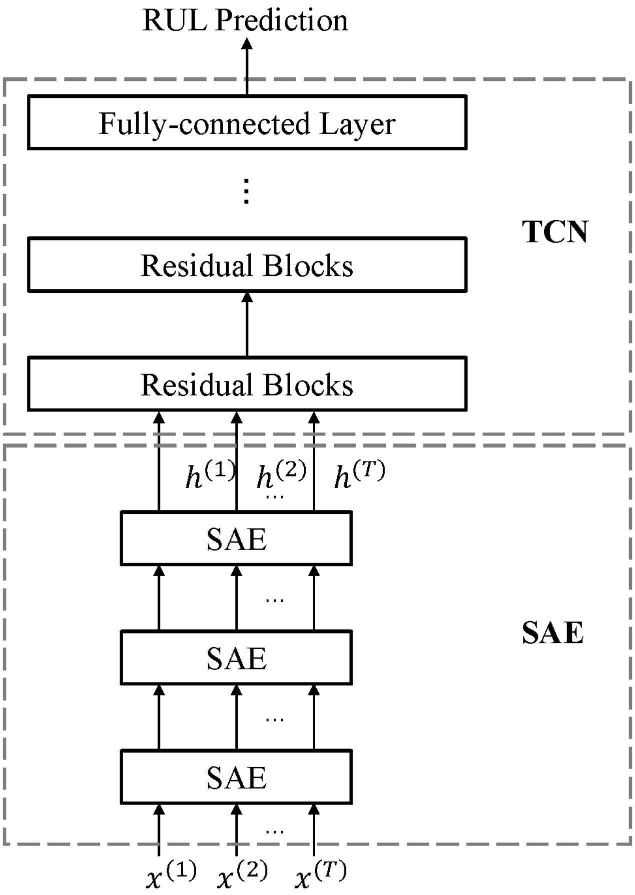

Figure 5. The structure of SAE-TCN model.

3. Experiments and Results

3.1. Benchmark Dataset

In this paper, an experiment is carried out on a widely used data set, which is generated using Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), which is a turbofan engine degradation simulation platform from NASA [36]. C-MAPSS simulates the degradation process of the turbofan engine under different operating settings and flight conditions, and its output is the engine state monitoring data of various sensor readings and operating condition indicators in the whole degradation process. Consistent with [37], all 21 sensor readings and three operating condition indicators included in the data set are used for RUL prediction in this paper. More detailed information of the selected indicators is shown in Table 1.

Table 1. Descriptions of selected parameters in C-MAPSS data set.

No.	Parameter Description	Units
1	Total temperature at fan inlet	R
2	Total temperature at LPC outlet	R
3	Total temperature at HPC outlet	R
4	Total temperature at LPT outlet	R
5	Pressure at fan inlet	psia
6	Total pressure in bypass inlet	psia
7	Total pressure at HPC outlet	psia
8	Physical fan speed	rpm
9	Physical core speed	rpm
10	Engine pressure ratio	-

Table 1. *Cont.*

No.	Parameter Description	Units
11	Static pressure at HPC outlet	psia
12	Ratio of fuel flow to static pressure at HPC outlet	pps/psiu
13	Ratio of static pressure	psia
14	Corrected fan speed	rpm
15	Corrected core speed	rpm
16	Bypass ratio	-
17	Burner fuel–air ratio	-
18	Demanded fan speed	rpm
19	Demanded corrected fan speed	rpm
20	HPT cooland bleed	lbm/s
21	LPT cooland bleed	lbm/s
22	Altitude	ft
23	Mach	-
24	Throttle resolver angle	deg

There are four sub-data sets in the C-MAPSS data set that record the run-to-failure time-series data of the turbofan engine under four different simulating settings. The first and third sub-data sets are under sea-level conditions, while the second and fourth sub-data sets have multiple operating conditions. This paper uses the first sub-data set, FD001, to train and test the model. FD001 contains a training set and a test set. They are mutually independent. The details of FD001 are shown in Table 2.

Table 2. Details of FD001 data set.

Name	Number of Engines in Training Set	Number of Engines in Test Set	Operating Conditions	Failure Modes	Training Sample	Test Sample	Sensors	Operating Parameters
FD001	100	100	Sea Level	HPC Degradation	20,630	13,095	21	3

The training set of FD001 records the condition monitoring information of 100 engines from normal operation to complete failure. Thus, the data of a total of 20,630 operating cycles is available. The task of the proposed SAE-TCN model is to predict the RUL of the engine before a complete failure. In addition to parameters shown in Table 1, we also include the operating cycle as one of the input features. Therefore, the inputs of the model are the 24 original features in Table 1 plus an operating cycle index. Before being put into the model, raw data must be normalized by Equation (4), so that the input data size is limited between [0, 1].

$$x_{i,j}^{\text{norm}} = \frac{x_{i,j} - x_j^{\text{min}}}{x_j^{\text{max}} - x_j^{\text{min}}}, \forall i, j \tag{4}$$

In the normal operation stage, the failure rate of the turbofan engine is constant and low. The degradation in this period can be ignored. As the number of cycles increases, the failure rate grows quickly until the end stage of the equipment. Therefore, data in the normal operation stage will not be used for model training. Similarly, the RUL tag is set to be a piecewise linear function, as shown in Figure 6. When remaining life exceeds or reaches a threshold value, the RUL is set to a constant. As the operating cycle increases and remaining life is less than the threshold value, the RUL monotonically decreases. The research shows that the mutation effect of the RUL in the training set is better when it is set at the 120th operation cycle, so the threshold value of the RUL is set to 120.

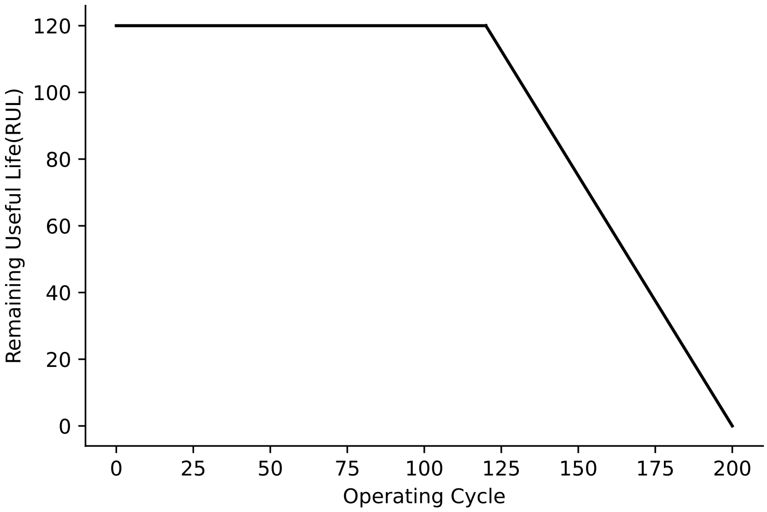

Figure 6. The remaining useful life setting.

3.2. Performance Measures

This paper uses the Root Mean Squared Error (RMSE) and Scoring Function to evaluate the accuracy of RUL prediction [13,37]. To assess the performance of the SAE-TCN model relative to each benchmark model, an average relative error indicator is used in this paper.

RMSE is widely used to measure how accurate a predictive model's result is. It is defined as Equation (5).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{5}$$

where m is the number of turbofan engines in the test set, y_i is the actual RUL of engine i, and $\hat{y}_i$ is the estimated RUL of engine i.

The Scoring Function is shown in Equation (6). Turbofan engines are complex equipment with extremely high safety requirements. If they are not sufficiently maintained, they will pose a great threat to the safety of an aircraft, which is not limited to the simple loss of thrust. It is usually not acceptable for aircraft to encounter hazardous damage in its core power system. Therefore, it is necessary to impose higher penalties on overestimation to ensure timely maintenance. The Scoring Function is designed to be an asymmetric function. The lower the score value is, the lower the possibility of overestimation. Figure 7 shows the score as a function of the error. This scoring strategy tends to suppress the overestimation of the RUL because the Scoring Function increases faster when the RUL is overestimated.

$$Score = \begin{cases} \sum_{i=1}^{m} \mathrm{e}^{-\left(\frac{\hat{y}_i - y_i}{13}\right)} - 1 & \hat{y}_i - y_i < 0 \\ \sum_{i=1}^{m} e^{\left(\frac{\hat{y}_i - y_i}{10}\right)} - 1 & \hat{y}_i - y_i \geq 0 \end{cases} \tag{6}$$

The average relative error is calculated by Equation (7).

$$average\ relative\ error = \frac{\Sigma_{m=1}^{K} error(SAE - TCN)/error(benchmark_m)}{K} \tag{7}$$

where K is the number of benchmark models. The function error$(\cdot)$ maps a model to its performance indicator. The RMSE and Scoring Function are selected to be the error$(\cdot)$ functions separately in this paper.

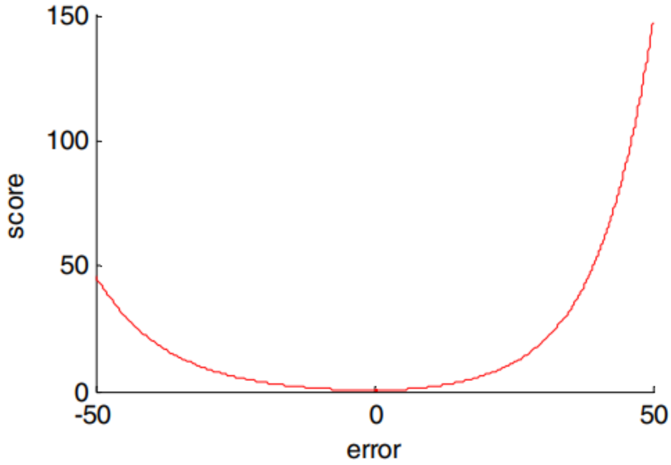

Figure 7. Score as a function of error.

3.3. Experimental Environment and Parameter Configuration

The training of the deep neural network is conducted using Keras 2.3.1, the Tensorflow 2.0 library and the Adam (Adaptive moment estimation) algorithm. The Adam algorithm is implemented by importing a relative module as an optimizer. It can adjust the speed of gradient descent by using momentum for automatic model optimization. Taking the high model complexity of the deep neural network into consideration, regularization is included to prevent over-fitting by convention. The details are as follows:

1. Dropout: Dropout means the model will be retrained under a new structure where some hidden-layer neurons are randomly deleted. For random descent algorithms, this allows different networks to be trained on each batch of data. Finally, these models are integrated to obtain average prediction results. This practice is intend to reduce the dependence between neurons and improve the generalization ability.
2. Early stopping: When a model has been trained for a long time, the performance of the model on the validation set might deteriorate [38]. This a common type of over-fitting, which can be avoid by early stopping. By early stopping, the training algorithm terminates when the performance of the model begins to decline. The balance between the training time and model generalization ability is expected to be achieved using this strategy.

The dimension number of the original data in the data set is compressed by the first layer of the encoder in the SAE from 24 to 20 and, finally, to 11. The 24-dimensional input of the encoder consists of 21 sensor readings and three operating condition indicators, which are described in Table 1. The 11-dimensional output of the encoder is then input into the TCN, and a prediction of the RUL is output.

The hyperparameters of the model greatly affect its performance. An appropriate hyperparameter setting is beneficial to improving the prediction performance of the model. The experiments show that there is almost no coupling between the main parameters involved in the TCN network structure, and the optimization results are shown in Figure 8. The acceptable hyperparameter settings of the model are shown in Table 3.

Table 3. The detailed parameters of SAE-TCN model.

Parameter	Value
Time Steps	10
Dropout Rate	0.4
Kernel Size	2
Number of Filters	64
Epochs	10
Batch Size	32

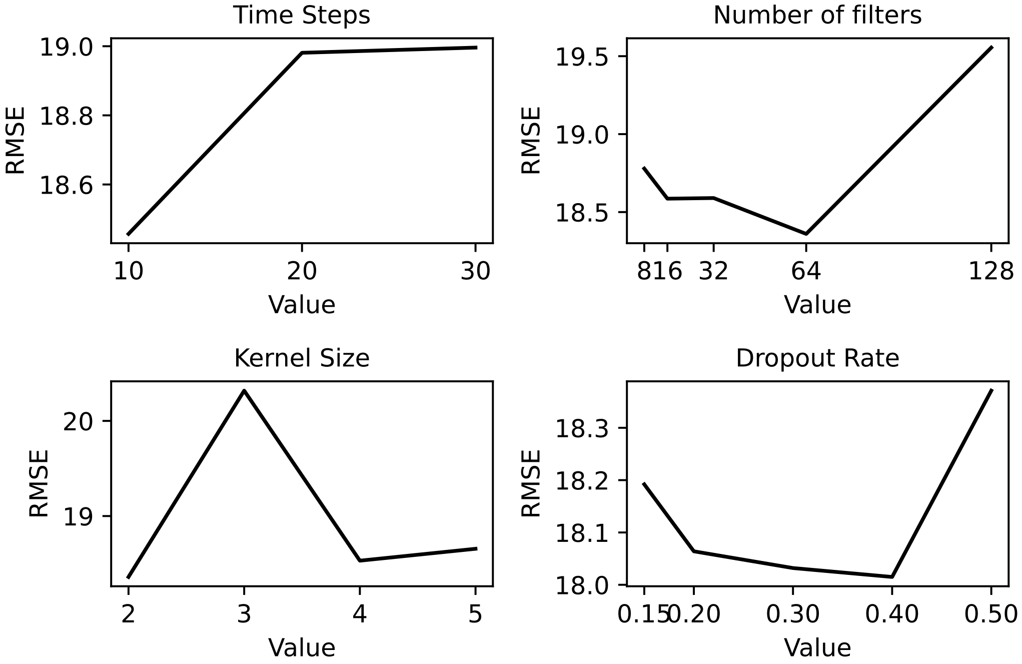

Figure 8. Model performance with different parameters.

3.4. Prediction Case

A complete performance evaluation is conducted in the next subsection. In this subsection, two engines are randomly taken out from the test set. The trained model is applied to these engines to show the details of the RUL prediction in the whole service history of an engine. The final results are displayed in Figures 9 and 10.

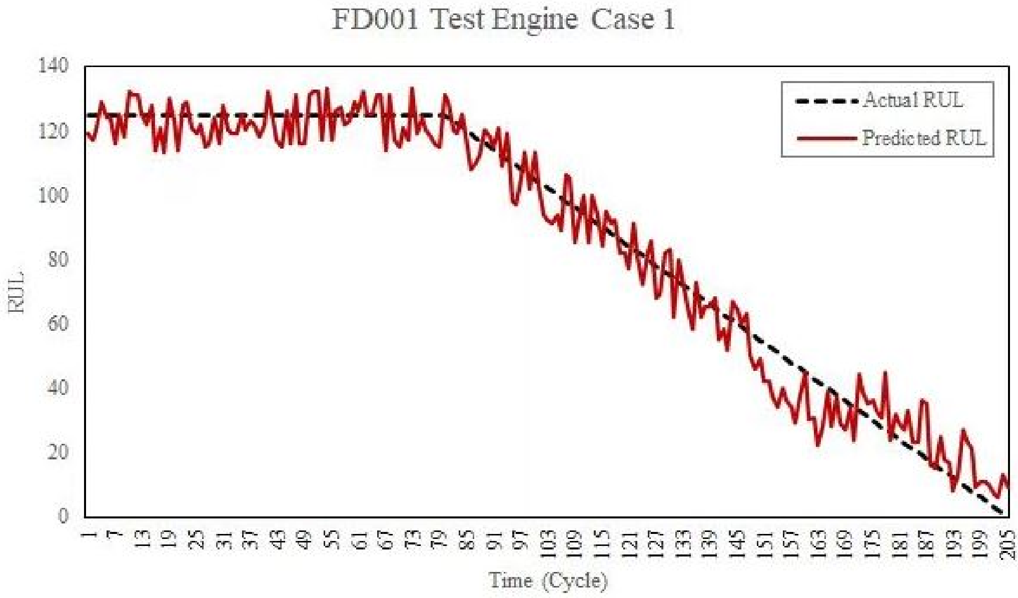

Figure 9. The predicted RUL of Engine 1.

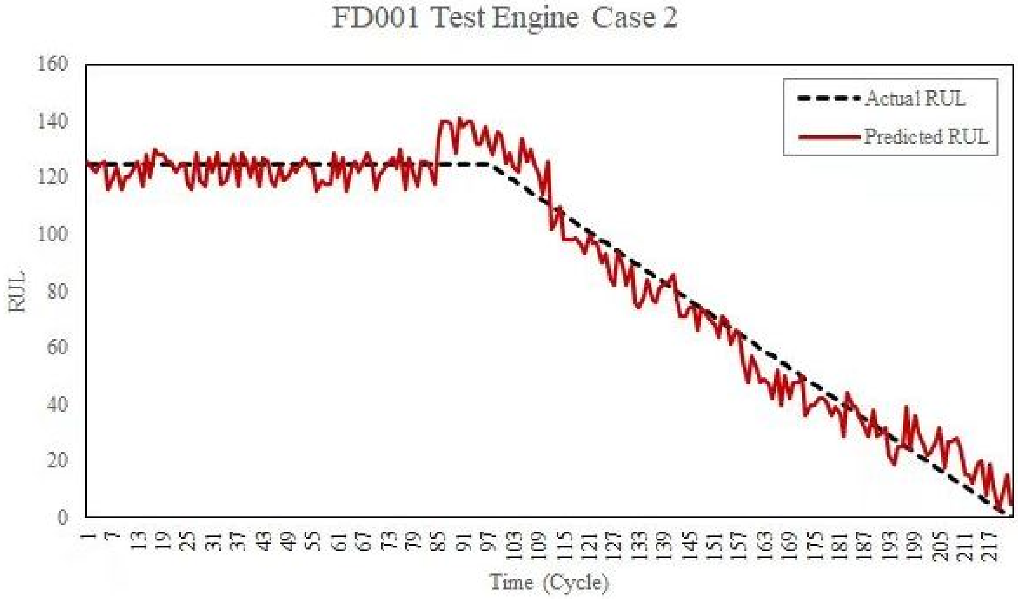

Figure 10. The predicted RUL of Engine 2.

3.5. Performance Comparison

Some machine learning models are included here to evaluate the performance of the SAE-TCN model by contrast. Standard TCN, Deep Long Short-Term Memory (DLSTM), Bi-directional Long Short-Term Memory (Bi-LSTM), Gate Recurrent Unit (GRU), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Support Vector Regression (SVR), and Multilayer Perceptron (MLP) models are trained as benchmark models. The DLSTM model is set to have 50 units and five layers. The CNN model is set to have 60 filters, with the kernel size being 3. And the activation function of the CNN model is the Relu function, with a pool size of 2. The SVR model is set to have the radial basis function (RBF) as the kernel function. The MLP model is set to have a network structure of (10, 5, 1). The performances of these different models are reported in Table 4. The average relative error of the DSAE-TCN relative to the other eight models is 0.9499 with respect to the RMSE. Using the Scoring Function as the error function, the average relative error is 0.2656.

Table 4. Performance for different models.

Model	RMSE	Score
DSAE-TCN	18.01	161
TCN	18.74	289
DLSTM	19.53	327
Bi-LSTM	19.94	435
GRU	20.60	885
RNN	22.59	780
CN4	22.93	1207
SVR	23.75	989
MLP	25.93	7890

From the experimental results, it can be seen that the SAE-TCN model performs better than the other standard models because it achieves the smallest RMSE and score. The SAE-TCN model has a smaller prediction error and shows a lower tendency to overestimate. Compared with the conventional methods, the model proposed in this paper further improves the prediction performance of the RUL. On the one hand, Autoencoder is effective at feature extraction. On the other hand, the TCN has an excellent applicability for long-time-span data processing. Therefore, the SAE-TCN model shows good prediction performance of a turbofan engine's remaining useful life, which might be helpful in turbofan engine health management.

The performance of the SAE-TCN model is compared with those of models in the public literature. The RMSE results are reported in Table 5. The average relative error of the RMSE is 0.9400. The average relative error of the score is 0.2979. Compared with some of the known works, the SAE-TCN model achieved better performance. Compared with the DCNN [39], the SAE-TCN shows a lower tendency to overestimate the RUL despite its larger RMSE in prediction.

Table 5. Performance for different models from public literature.

Model	RMSE	Score
DSAE-TCN	18.01	161
CNN [40]	31.2432	N/A
SVR [41]	20.96	1381
RVR [41]	23.80	1502
DLSTM [41]	18.33	655
DCNN [39]	11.81	223

4. Conclusions

In this paper, a remaining useful life prediction model, named SAE-TCN, based on Autoencoder and a TCN, is proposed to solve the problems of the high dimensions and long time span of monitoring data from a turbofan engine degradation process. Firstly, the model uses the Autoencoder model for feature extraction and data dimension reduction. Secondly, the model then uses the TCN to capture the relationship between the reduced features and the RUL. The experimental results in turbofan engine life prediction show that the proposed method performs better in RMSE and score than the conventional machine learning and deep learning models. Autoencoder brings in performance improvements to the standard TCN model via data dimension reduction. These conclusions help a lot in the health management of turbofan engines. It is worth noting that the method used in this paper is a data-driven method. How well a data-driven method performs usually depends on the quality of the data set. Also, the source of the data set limits the proposed method to engines of similar kinds. It remains to be studied whether the proposed method is suitable for other fault modes, especially those greatly different from engine gas path degradation.

Author Contributions: Conceptualization, X.L.; methodology, Y.Z. and C.L.; software, Y.Z.; validation, L.X.; writing—original draft preparation, Y.Z.; writing—review and editing, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation with grant number 68190923.

Data Availability Statement: C-MAPSS data set is currently unavailable for download now according to the information on NASA official website. It may be found from some individual researchers.

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

References

1. Richter, H. *Advanced Control of Turbofan Engines*; Springer: London, UK, 2012.
2. Salunkhe, T.; Jamadar, N.; Kivade, S. Prediction of Remaining Useful Life of mechanical components-a Review. *Int. J. Eng. Sci. Innov. Technol. (IJESIT)* **2014**, *3*, 125–135.
3. Wang, X.; Li, Y.; Xu, Y.; Liu, X.; Zheng, T.; Zheng, B. Remaining useful life prediction for aero-engines using a time-enhanced multi-head self-attention model. *Aerospace* **2023**, *10*, 80. [CrossRef]
4. Wang, H.; Li, D.; Li, D.; Liu, C.; Yang, X.; Zhu, G. Remaining Useful Life Prediction of Aircraft Turbofan Engine Based on Random Forest Feature Selection and Multi-Layer Perceptron. *Appl. Sci.* **2023**, *13*, 7186. [CrossRef]
5. Huang, Y.; Tao, J.; Sun, G.; Zhang, H.; Hu, Y. A prognostic and health management framework for aero-engines based on a dynamic probability model and LSTM network. *Aerospace* **2022**, *9*, 316. [CrossRef]
6. Chen, Z.; Cao, S.; Mao, Z. Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach. *Energies* **2017**, *11*, 28. [CrossRef]
7. Rohan, A. Deep Scattering Spectrum Germaneness for Fault Detection and Diagnosis for Component-Level Prognostics and Health Management (PHM). *Sensors* **2022**, *22*, 9064. [CrossRef]

8. 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]
9. Muneer, A.; Taib, S.; Naseer, S.; Ali, R.; Aziz, A. Data-Driven deep learning-based attention mechanism for remainging useful life prediction: Case study application to turbofan engine analysis. *Electronics* **2021**, *10*, 2453. [CrossRef]
10. Xie, Z.; Du, S.; Deng, Y.; Jia, S. A hybrid prognostics deep learning model for remaining useful life prediction. *Electronics* **2020**, *10*, 39. [CrossRef]
11. Kang, Z.; Catal, C.; Tekinerdogan, B. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. *Sensors* **2021**, *21*, 932. [CrossRef]
12. Zhao, C.; Huang, X.; Li, Y.; Yousaf Iqbal, M. A double-channel hybrid deep neural network based on CNN and BiLSTM for remaining useful life prediction. *Sensors* **2020**, *20*, 7109. [CrossRef]
13. Elsheikh, A.; Yacout, S.; Ouali, M.S. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* **2019**, *323*, 148–156. [CrossRef]
14. Orsagh, R.F.; Sheldon, J.; Klenke, C.J. *Prognostics/Diagnostics for Gas Turbine Engine Bearings*; American Society of Mechanical Engineers (ASME): New York, NY, USA, 2003; Volume 36843.
15. Chelidze, D.; Cusumano, J.P. A dynamical systems approach to failure prognosis. *J. Vib. Acoust.* **2004**, *126*, 2–8. [CrossRef]
16. Giantomassi, A.; Ferracuti, F.; Benini, A.; Ippoliti, G.; Longhi, S.; Petrucci, A. Hidden Markov model for health estimation and prognosis of turbofan engines. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC, USA, 28–31 August 2011; Volume 54808, pp. 681–689.
17. Wu, Q.; Ding, K.; Huang, B. Approach for fault prognosis using recurrent neural network. *J. Intell. Manuf.* **2018**, *31*, 1621–1633. [CrossRef]
18. Sikorska, J.; Hodkiewicz, M.; Ma, L. Prognostic modelling options for remaining useful life estimation by industry. *Mech. Syst. Signal Process.* **2011**, *25*, 1803–1836. [CrossRef]
19. Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; Liu, Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* **2018**, *275*, 167–179. [CrossRef]
20. Chen, X.; Jin, G.; Qiu, S.; Lu, M.; Yu, D. Direct remaining useful life estimation based on random forest regression. In Proceedings of the 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai), Shanghai, China, 16–18 October 2020; pp. 1–7.
21. Peng, C.; Chen, Y.; Chen, Q.; Tang, Z.; Li, L.; Gui, W. A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion. *Sensors* **2020**, *21*, 418. [CrossRef]
22. Zhang, X.; Xiao, P.; 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]
23. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; Technical Report; California University of San Diego, La Jolla Institute for Cognitive Science: La Jolla, CA, USA, 1985.
24. Ranzato, M.; Poultney, C.; Chopra, S.; LeCun, Y. Efficient learning of sparse representations with an energy-based model. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 1137.
25. Meng, L.; Ding, S.; Xue, Y. Research on denoising sparse autoencoder. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1719–1729. [CrossRef]
26. Lea, C.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks: A unified approach to action segmentation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–10 October 2016; pp. 47–54.
27. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
28. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
29. Wan, R.; Mei, S.; Wang, J.; Liu, M.; Yang, F. Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electronics* **2019**, *8*, 876. [CrossRef]
30. Chen, Y.; Kang, Y.; Chen, Y.; Wang, Z. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* **2020**, *399*, 491–501. [CrossRef]
31. Yan, J.; Mu, L.; Wang, L.; Ranjan, R.; Zomaya, A.Y. Temporal convolutional networks for the advance prediction of ENSO. *Sci. Rep.* **2020**, *10*, 8055. [CrossRef]
32. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
33. Oord, A.v.d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
34. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. 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.

37. 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), Dallas, TX, USA, 19–21 June 2017; pp. 88–95.
38. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
39. Muneer, A.; Taib, S.; Fati, S.; Alhussian, H. Deep-learning based prognosis approach for remaining useful life prediction of turbofan engine. *Symmetry* **2017**, *13*, 1861. [CrossRef]
40. Yuan, N.; Yang, H.; Fang, H. Aero-engine prognostic method based on convolutional neural network. *Comput. Meas. Control* **2019**, *27*, 74–78.
41. Sun, W.; Zhao, R.; Yan, R. Convolutional discriminative feature learning for induction motor fault diagnosis. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1350–1359. [CrossRef]

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.