Hindawi
International Journal of Aerospace Engineering
Volume 2024, Article ID 6465566, 14 pages
https://doi.org/10.1155/2024/6465566

*Research Article*

# Aeroengine Remaining Life Prediction Using Feature Selection and Improved SE Blocks

**Hairui Wang,[1] Shijie Xu,[1] Guifu Zhu,[2] and Ya Li[1]**

[1]*Kunming University of Science and Technology School of Information Engineering and Automation, Kunming 650000, China*
[2]*Information Technology Construction Management Center, Kunming University of Science and Technology, Kunming 650000, China*

Correspondence should be addressed to Ya Li; liya@kust.edu.cn

Received 18 November 2023; Revised 21 January 2024; Accepted 8 February 2024; Published 25 March 2024

Academic Editor: Antonio Concilio



Aeroengines use numerous sensors to detect equipment health and ensure proper operation. Currently, filtering useful sensor data and removing useless data is challenging in predicting the remaining useful life (RUL) of an aeroengine using deep learning. To reduce computational costs and improve prediction performance, we use random forest to evaluate the feature importance of sensor data. Based on the size of the feature corresponding to the Gini index, we select the appropriate sensor. This helps us to determine which sensor to use and ensures that the computational resources are not wasted on unnecessary sensors. Considering that the RUL of equipment changes in a progressively more complex manner as the equipment is used over time, we propose an improved squeeze and excitation block (SSE) and combine it with a convolutional neural network (CNN). By enhancing the feature selection ability of CNN through segmented squeeze and excitation block, the model can focus on important information within features to effectively improve prediction performance. We compared our experiments with other RUL experiments on the CMAPSS aeroengine dataset and then conducted ablation experiments to verify the critical role of the methods we used.

## 1. Introduction

Mechanical maintenance workers employ equipment fault diagnosis and health management techniques to guarantee the safety and dependability of mechanical industrial operations. They assess, analyze, and forecast the operational condition of mechanical equipment to ascertain its proper working or any abnormalities. In addition, they analyze the causes of equipment malfunctions and forecast the remaining lifespan of the equipment based on their expertise and the equipment's attributes. Research data reveals that more than 60% of aircraft failures are attributed to turbofan engine malfunctions, highlighting its critical importance as a component in any aircraft [1]. In order to mitigate the severe repercussions of abrupt malfunction, it is imperative to guarantee the smooth and consistent functioning of mechanical apparatus or components by precisely ascertaining their remaining useful life (RUL). Aviation engines produce a significant volume of complex data during their operation, characterized by its high dimensionality, nonlinearity, and temporal and spatial properties. Extracting meaningful features from the data and mapping them to RUL forecasts is a difficult task.

Several models have been created to forecast the remaining useful life (RUL) of turbofan engines. The models can be classified into three main categories: model-based, data-driven, and hybrid approaches [2]. The model-based approach entails developing a mathematical model that contains considerable a priori information of the equipment in question and using it to predict the remaining life of the equipment by stating its failure mechanism. However, this approach also imposes certain constraints on model predictions [3, 4]. Model-driven methods rely heavily on the accuracy of models and parameters, but as the complexity of prediction equipment increases, the complexity of model construction rises while the accuracy is difficult to guarantee. In the task of aeroengine residual life prediction, the adaptability and generalization ability of the model are also limited when faced with

multiple operating environments and different failure mechanisms. Data-driven methods relate to the use of machine learning and statistical techniques to anticipate the remaining life of mechanical equipment. These methods assess the operation information and characteristics of the equipment, discovering potential links with the RUL. Unlike traditional approaches, data-driven procedures do not require a huge amount of prior knowledge or understanding of the failure mechanism. This makes them easier to use, and they have become the mainstream of today's research [5–7]. Data-driven approaches require to verify the accuracy and completeness of the data and need to be trained and modelled with a huge amount of previous data from the device. If the amount of data is insufficient or unrepresentative, the prediction outputs of the model will likewise become less accurate. A hybrid technique is a combination of the first two methods to estimate the RUL of a device through a failure mechanism model and data-driven methods. Hybrid approaches not only include the selection of weights for the two methods but also increase the complexity of the overall prediction method as well as limit the interpretability of the prediction method, making the results harder to understand.

The aeroengine itself is characterized by accuracy and complexity, and when it functions, it generates multiple components and system status information and also causes mutual effects and linkages between various components. When a single component fails, it causes the failure of associated components, leading to the coupling of different failure modes. This makes the failure modes of the aeroengine numerous and complex, so it is difficult to develop a physical failure model based on the failure mechanism while facing the aeroengine based on the construction of a physical failure model. Therefore, a data-driven method is commonly chosen for the task of forecasting the remaining life of aeroengines.

The progress of artificial intelligence and sensor technologies has enabled us to use data-driven methodologies to train deep learning models. By mapping the job output end-to-end and automatically learning data representation and features, we may significantly reduce the complexity of the prediction task using a huge amount of collected data. Deep learning has attracted great attention from researchers due to its remarkable nonlinear fitting capabilities that can yield distributions closer to reality. Among the deep learning techniques, convolutional neural networks (CNN) have achieved great success in different computer vision domains such as image processing, automatic driving, and medical image analysis. In-depth research on network structure, optimization, regularization, normalization, and other key areas has substantially enhanced the performance of CNN, making it highly attractive in the industrial sphere. Li et al. introduced a deep convolutional network (DCNN), which learns to extract high-level abstract information from sensor input by stacking four convolutional layers to achieve RUL prediction [8]. Wang et al. proposed a combination of CNN and long short-term memory (LSTM) network, wherein CNN adaptively extracts deep features and inputs them into the LSTM network to construct trending quantitative health indicators [9]. Yang et al. introduced a dual CNN model architecture that predicts and distinguishes anomalies based on different degradation patterns, combining degradation models for RUL prediction [10]. Jin et al. introduced a CNN network based on positional coding to model sequential information, facilitating quicker training and acquisition of sequential information for reliable RUL prediction [11]. Lin et al. developed a signal selection approach that assesses the signal's ability to map RUL and picks more valuable signal data, which is fed into a fully convolutional network for RUL prediction [12].

Feature selection is crucial for remaining life prediction [13], and there have been several relevant works [14, 15]. Li et al. immediately input all original features into the model without considering feature relevance, which can increase model computation cost and reduce prediction accuracy owing to noisy data. Chen et al. created an LSTM network and employed self-attention methods to learn from the raw data, identifying representative sequential features and giving them more weight for RUL prediction [16]. Liu et al. presented a feature attention technique, establishing a feature attention weight matrix to analyze various feature inputs, achieving adaptive weighting to focus the model more on significant features [17]. However, they neglected that the remaining usage life of equipment frequently follows a linear diminishing trend, owning excellent fitting potential. Irregular feature changes might be disregarded; however, this data could include vital information for RUL prediction. To circumvent this, we applied random forests to evaluate multiple features [18, 19] and based on corresponding importance weight parameters to accurately retain the useful features for RUL prediction [20, 21].

Attention mechanisms have shown great success in the realms of image recognition and natural language processing. One strategy involves replicating the correlation of feature response space or channel relevance to enhance the representation capabilities of deep convolutional networks [22]. By introducing squeeze and excitation (SE) blocks into CNN networks, the SE block first runs the squeeze operation, which comprises global average pooling to obtain a scalar value for each channel. Subsequently, the excitation module turns these scalars into channel weights, enabling the network to learn to choose and emphasize the importance of channel information while suppressing less significant aspects. However, SE blocks do have certain limitations [23] since they apply the squeeze operation indiscriminately to global information, thereby hiding changes in particular data patterns. Therefore, we have explored integrating SE blocks into CNN networks for RUL prediction in turbofan engines. We have improved the SE blocks based on data fluctuation features and machine degradation patterns. We segment global information to keep more sequential data's upward and downward changes, and we refer to this as segmented squeeze and excitation blocks (SSE).

Aeroengines are equipped with sensors at multiple critical locations to collect degradation information, which results in a large amount of data showing degradation trends, but there is also a lot of redundancy and irrelevance in this data. Considering the failure coupling caused by the interconnections between aeroengine components, when a

single component starts to degrade, other related components will also start to degrade. Therefore, direct prediction of all the sensor characteristic data will not only increase the computational cost but also make the model overfit due to irrelevant and redundant data, and the accuracy of prediction results will be reduced. The main goal of our research is to select more comprehensive and representative sensor characteristic data from a large amount of sensor data, which contains more degradation trend information, so as to improve the accuracy of aeroengine residual life prediction. The paper leverages random forest to evaluate the relevance of sensor feature data. This is done to filter out the feature data that are relevant for forecasting RUL and remove the unnecessary information. By doing this, it minimizes the processing cost and enhances the prediction accuracy. Based on the RUL change condition, the device will initially run normally for some time; after a while, its performance will start to degrade. To increase the accuracy of RUL prediction, the SE in the field of image recognition has been improved. The global compression has been altered to segmented compression, which corresponds better with the actual RUL prediction condition. This new approach maintains more data change patterns and hence increases the prediction accuracy.

The main contributions of this paper are as follows:

(1) Random forest is used to analyze the feature importance of the sensor data of turbofan engines to filter the valuable information that can improve the predictive RUL prediction ability

(2) The SE attention mechanism is improved to retain more feature change trends for the characteristics of turbofan engine service life change trends

(3) Demonstrate the necessity and excellent RUL prediction performance of the feature selection of random forests and SSE combined CNN that we have used through comparison and ablation experiments

The rest of the paper is organized as follows: the adopted methodological framework is described in Section 2. A demonstration of our experimental setup is presented in Section 3. In Section 4, a series of comparison and ablation experiments are performed to demonstrate the excellent performance of our method. Finally, the paper is summarized and the directions for future research are outlined.

## 2. Methods

In our RUL prediction method, we utilize random forest to select features and combine CNN with the SSE. This section provides a detailed explanation of the RUL prediction task and the methodology employed. You can refer to the flowchart depicted in Figure 1 for an overview of the entire methodology.

*2.1. Multivariate RUL Projections.* The most crucial part of the experimental process is the multivariate RUL prediction task, which evaluates the RUL of the device in the future period using sensor data, operation records, temperature, vibration, and other information. The training set can be denoted as $\mathrm{TR} = \{X_i^{\mathrm{tr}}, Y_i^{\mathrm{tr}} | i \in [1, N^{\mathrm{tr}}]\}$, where $N^{\mathrm{tr}}$, $X_i^{\mathrm{tr}} \in \theta^{f*W}$, and $Y_i^{\mathrm{tr}}$ represent the total number of samples in the training dataset, the $i$-th input sample in the training set, and the $i$-th RUL label value in the corresponding training set, respectively. In $\theta^{F*W}$, $F$ represents the number of features in the input samples, $W$ represents the size of the time window, and the whole indicates that the $i$-th input sample contains all the data volume. The synoptic test set can be represented as $\mathrm{TE} = \{X_i^{\mathrm{te}}, Y_i^{\mathrm{te}} | i \in [1, N^{\mathrm{te}}]\}$. The prediction task of RUL is performed by constructing a remaining life prediction model $f(x)$, which is passed to the training set for earning, and finally allowing the prediction results $f(X_i^{\mathrm{te}})$ and $Y_i^{\mathrm{te}}$ to achieve a good fit.

*2.2. Random Forest.* To increase the accuracy of RUL prediction, we begin by applying random forests for data preprocessing. This helps us to find the most significant sensor features, which can further boost the RUL forecast accuracy. Random forests are particularly good for nonlinear classification issues because of the mixture of trees. The performance of RUL prediction can be further improved with increasing the number of forests [24]. The random forest algorithm is simple and has small computational cost, and through the random forest for importance analysis, it can be achieved at a reduced computational cost of feature screening, reduce the computing complexity of the future training phase, and increase the accuracy of the prediction. Through the important evaluation, it can be seen how much each feature contributes to each spanning tree, and through the summing and average, it can be seen that the relevance of each sensor data is compared, and the screening of feature data is accomplished.

Random forest is a sort of machine learning technique that uses several decision trees. Each tree is formed independently and learns and predicts from its own set of data, which provides the random forest some generalization power. When creating each tree, the bootstrap sampling method is utilized to randomly select samples from the original dataset with replacement. Every time a branch of the decision tree is constructed, it does not take into account all the features in the dataset. Instead, it randomly selects a subset of features to decrease the potential of model overfitting. Random forest employs the Gini index or out-of-bag data to evaluate the importance of attributes based on the makeup of the decision tree. We use the Gini index approach to evaluate feature relevance [19]. The flowchart of the feature importance calculation of the random forest algorithm is shown in Figure 2. The feature importance evaluation of random forest is to evaluate the feature corresponding to each node in the tree to see how much each feature contributes to the tree where it is located, and the importance evaluation of each feature is denoted by FIM. Assuming that there are a total of $J$ features, $I$ trees, and $C$ categories, the Gini index score $\mathrm{FIM}_j^{(\mathrm{Gini})}$ is performed on each feature $X_j$.

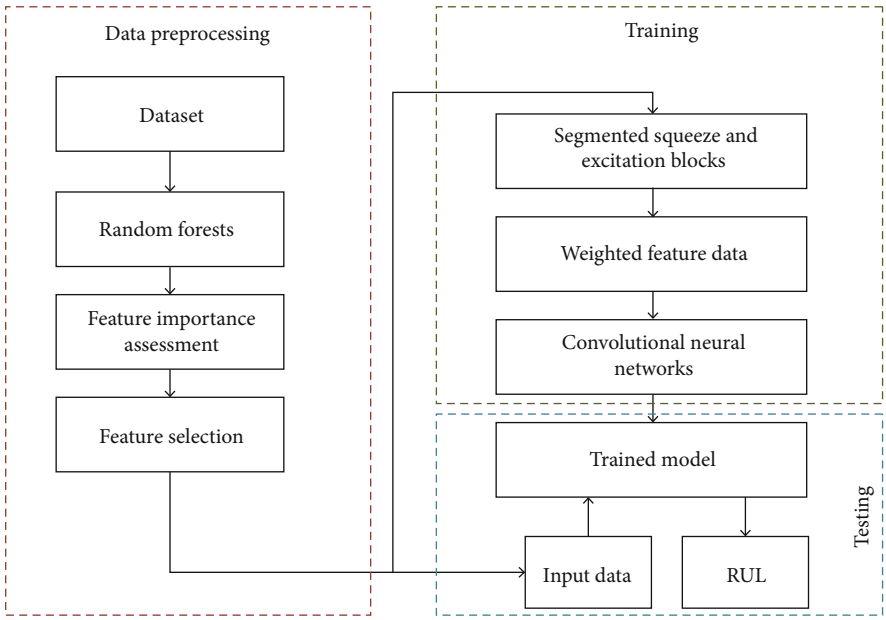


Figure 1: RUL prediction flowchart.

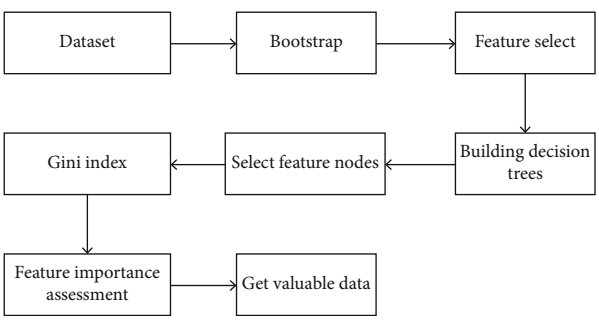


Figure 2: Flowchart of random forest feature importance calculation.

The Gini index of node $q$ in the $i$-th tree is given by

$$\mathrm{Gini}_q^{(i)} = \sum_{C=1}^{|C|} \sum_{C' \neq C} P_{qc}^{(i)} P_{qc'}^{(i)} = 1 - \sum_{C=1}^{|C|} \left(P_{qc}^{(i)}\right)^2. \quad (1)$$

Here, $P_{qc}$ represents the proportion of category $C$ in node $q$, where the number of categories $C$ is set as the number of original features. The degree of impurity of each sample in the node is calculated, after which the feature with the highest reduction in impurity is selected for the segmentation node. Finally the importance of features can be evaluated by calculating the number of times each feature has been selected in all decision trees and normalizing it to an importance score.

Based on the change in the Gini index before and after branching of node $q$, it can be concluded that the importance of feature on node q of the $i$-th tree is

$$\mathrm{FIM}_{jq}^{(\mathrm{Gini})(i)} = \mathrm{Gini}_q^{(i)} - \mathrm{Gini}_l^{(i)} - \mathrm{Gini}_r^{(i)}, \quad (2)$$

where $\mathrm{Gini}_l^{(i)}$ and $\mathrm{Gini}_r^{(i)}$ denote the Gini index of the two new nodes after the branching of node $q$.

If the set of feature occurrences in decision tree $i$ is $Q$, with a total of $I$ trees, then the importance in the forest is

$$\mathrm{FIM}_j^{(\mathrm{Gini})} = \sum_{i=1}^{I} \mathrm{FIM}_j^{(\mathrm{Gini})(i)}. \quad (3)$$

Finally, normalize the importance scores of all features.

$$\mathrm{FIM}_j^{(\mathrm{Gini})} = \frac{\mathrm{FIM}_j^{(\mathrm{Gini})}}{\sum_{j'=1}^{J} \mathrm{FIM}_{j'}^{(\mathrm{Gini})}}. \quad (4)$$

Observing the distribution of $\mathrm{FIM}_i$, higher numerical values indicates that the feature contains more valuable information and possesses a stronger mapping capability for RUL prediction. When FIM is 0, it signifies that the feature does not contribute to RUL prediction. These sensor data do not contain RUL label information when they are passed into the random forest for feature importance assessment, so the random forest cannot generate a mapping between the feature data and the corresponding RUL label values, which means that the feature data are not able to produce a practical connection with the RUL to obtain a higher weight advantage, and we can only rely on the feature data to train to achieve the global optimal point of the model and obtain the features that contain more information and value for the RUL prediction. It can only rely on the feature data to train the model to achieve the global optimal point and to obtain features that contain more information and are more valuable for RUL prediction.

*2.3. Convolutional Networks Based on Segmented Squeeze and Excitation Blocks.* The attention mechanism in deep learning is a strategy that mimics the human cognitive

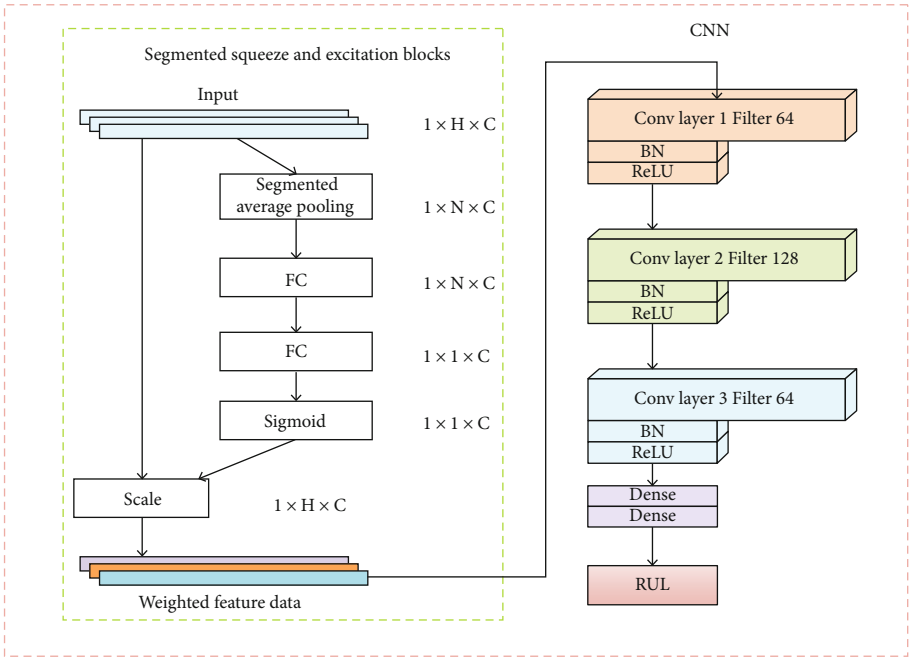


Figure 3: Structure of convolutional network based on segmented extrusion and excitation blocks.

function of vision, which allows neural networks to focus their attention on more useful information while processing incoming data. By incorporating the attention mechanism into the neural network, it enables the network model to automatically learn and focus on the key information of the input, thus delivering increased model performance and generalization. After analyzing the value of the features by random forest and filtering the features that contain more valuable information for RUL prediction, we opted to further upgrade the network model to extract more vital information from the incoming feature data. We discovered that SE in the image domain may reweight the feature data and deliver a good accuracy boost to the model via improved SE and combining it with CNN to accomplish RUL prediction for aeroengines. The specific network structure diagram is shown in Figure 3 in detail.

The SE consists of three primary procedures: first, it performs a global average pooling operation to compress and then independently extrude the whole feature data into a single value, with each value reflecting the relevance of the related feature to the total feature data. Then, the ongoing modification of the data in each extrusion channel is implemented through single or multiple fully connected layers, and the ReLU activation function is employed to capture and express the nonlinear data relationships. In this process, the network model learns the relevance of each channel. Finally, the feature representation is strengthened by repassing the learned channel feature weights into the CNN so that it can focus on more essential features in subsequent training.

However, we noticed that the global average pooling operation of SE works efficiently in the image domain, but when it comes to RUL prediction, this direct compression of the global information has major restrictions. The global pooling method can potentially conceal some data that indicates a local rising or downward trend, which could be essential for RUL prediction. The depreciation of industrial equipment often starts after a certain length of use [25]. Furthermore, the RUL of industrial equipment can be separated into two states: a smooth change state and a quick decline state. In RUL prediction, segmented linear deterioration is usually employed as the objective function to estimate this [26]. However, the straight use of SE for RUL prediction is impacted by the smooth state data and loses a substantial number of local data changes. This can substantially impair the accuracy of RUL predictions. To be able to make SE better for the RUL prediction task, we propose SSE, which not only fits the requirements of the RUL prediction task but also maintains more data variance ups and downs, thus giving a great improvement to the RUL prediction accuracy, and the segmented compression improvement of SE is shown in Figure 4. Inspired by the RUL segmentation deterioration of industrial equipment, we segment the entire data by setting a length value and dividing the data into several length segments to separate the two states of RUL change, smooth change, and rapid decrease, as much as feasible. Then, when the segmented data are evenly divided into two segments, the average pooling operation is done, to maintain more local data change trends to accomplish more accurate RUL prediction.

Transform the feature data $X$ into a tensor $U \in R^{1\times H\times C}$ as input to the SSE, and then convert tensor $U \in R^{1\times H\times C}$ into a tensor $F \in R^{1\times S\times C}$ through the squeezing operation, with the squeezing operation formula as follows:

$$F_{sq}(C, n) = \frac{1}{\mathrm{fl}(H/N)} \sum_{i=(n-1)*\mathrm{fl}(H/N)}^{n*\mathrm{fl}(H/N)} X(C, i), \tag{5}$$

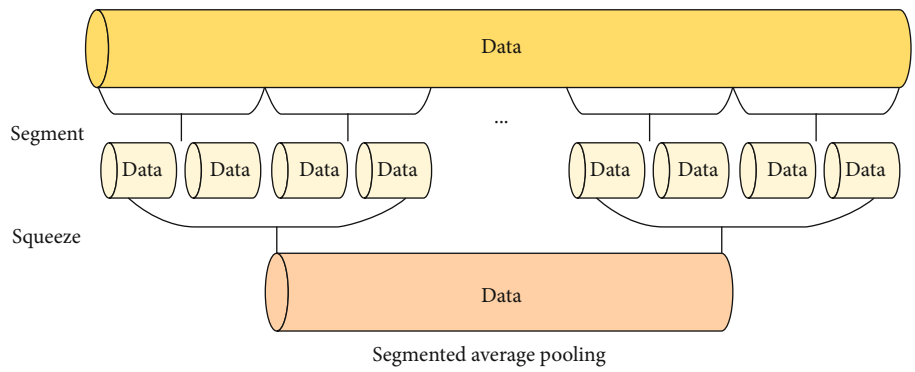


Figure 4: Improvements to SE—segmented average pooling.

where $C$ represents the number of features selected through the random forest, $\mathrm{fl}(x)$denotes the floor function, $H$ is the length of segmented feature data, $N$ is the length of segmented compressed data, $X(C, i)$ represents the $i$-th data of the corresponding feature channel, and $n$ represents the total number of data. Using the process of segmented compression helps to separate the two phases of RUL as much as possible. After that, average pooling is performed to retain more data variations and achieve better feature information characterization capability than SE. In this process, the value of $N$ is set to 2 based on the actual setting of RUL linear segmentation degradation. Meanwhile, the value of $H$ is set considering the average running period of each data subset and the complexity of the data. Specifically, the FD001 to FD003 datasets are set to 35, while the FD004 dataset is set to 25 considering that it is the most complex. The specific value setting and linear degradation segmentation point setting will be analyzed and demonstrated in Section 4 in combination with the SSE.

The squeeze operation formula is as follows:

$$F_{\mathrm{ex}}\left(F_{\mathrm{sq}}, W\right) = \sigma\left(\mathrm{g}\left(F_{\mathrm{sq}}, W\right)\right) = \sigma\left(\mathrm{g}\left(W_2 \delta\left(W_1 F_{\mathrm{sq}}\right)\right)\right), \quad (6)$$

where $\delta$ denotes the ReLU activation function, $\sigma$ signifies the sigmoid activation function, and $W_1$ and $W_2$ represents the weight matrices of two fully connected layers, respectively.

The output result of the SSE is as follows:

$$\tilde{X} = F_{\mathrm{scale}}(X, F_{\mathrm{ex}}). \quad (7)$$

The importance weights of all the features are obtained after the segmented squeeze and excitation operations. Then, the input feature data is multiplied with its importance weight by scale operation and reweighted to get new data with importance weight. Finally, the new data with importance weights are fed into the CNN to achieve RUL prediction.

The neurons of each layer in CNN are arranged in 3 dimensions, i.e., height, width, and depth. When dealing with one-dimensional data from engine sensors, the width should be set to 1. The CNN mainly consists of an input layer, a convolutional layer, a ReLU layer, a pooling layer, and a fully connected layer. Before the data is transferred to the CNN, it undergoes a pooling operation in the SSE, so we do not include a pooling layer in the CNN structure to avoid losing too much information and failing to capture the key features, which leads to the degradation of the model's performance. The CNN consists of three convolutional layers, and the formula of the convolutional layer is as follows:

$$\mathrm{Conv}_i = \mathrm{Re\ LU}\left(\mathrm{BN}\left(\mathrm{Fliter}_{c_i} \otimes \mathrm{Input} + b_i\right)\right), \quad (8)$$

where Input is the input of each layer and $b_i$ is the bias term for each layer. The first layer has 64 filter sizes, the second layer has 128 filter sizes, and the third layer has 64 filter sizes. After each layer, there is a normalization layer and a ReLU layer. Finally, as shown in Figure 3, the deep feature data extracted by the three convolutional layers is passed to the fully connected layer. The computation of the final RUL is achieved through global average pooling and two layers of fully connected layers.

## 3. Experimental Setup

In this section, we show the data preprocessing and experimental setup of the experiment.

*3.1. CMPASS Dataset.* In this section, we utilize the CMPASS dataset to further validate our approach [27]. The CMPASS dataset is acquired from the operation of turbofan engines, equipped with 21 sensors at various engine components and places. It comprises four subsets: FD001, FD002, FD003, and FD004. Each subgroup reflects diverse operational conditions and fault modes, with distinct initial states and degrees of wear. The training and test sets relate to the same turbofan engine, where the training data starts with normal operation and steadily records performance decrease until the occurrence of a problem. The test set includes sensor data only up to just before the engine fault. The specific details of the CMPASS dataset are presented in Table 1. Among the four subsets, FD002 and FD004 contain more constraints and training samples, and complex operating conditions and fault modes can challenge RUL prediction, leading to decreased prediction accuracy. Achieving good results in RUL prediction for both simple and complex datasets is a challenging task.

Table 1: CMPASS dataset.

| Data | CMPASS dataset | | | |
|---|---|---|---|---|
| | FD001 | FD002 | FD003 | FD004 |
| Training set | 100 | 260 | 100 | 249 |
| Average training set period | 206 | 206 | 247 | 245 |
| Test set | 100 | 259 | 100 | 248 |
| Test set average period | 130 | 130 | 165 | 166 |
| Operating conditions | 1 | 6 | 1 | 6 |
| Failure model number | 1 | 1 | 2 | 2 |
| Number of training samples | 17731 | 48819 | 21820 | 57522 |

*3.2. Data Processing.* First, after evaluating feature importance using random forest, we select sensor data with importance scores greater than 0.005. These sensor features, which provide more effective information for RUL prediction, are marked in red, while those that do not meet the threshold are marked in blue, as shown in Figure 5. Additionally, we apply normalization to each selected feature to expedite the model's search for the optimal solution.

When estimating the RUL of industrial equipment, the equipment will remain in a normal state for a period of time after it starts to run, and it is not acceptable to anticipate how long the equipment will survive when it is still in a healthy state. It is only reasonable to forecast the RUL when the performance of the equipment is degraded after a long period of use. Zheng et al. proposed to adopt a segmented linear degradation model in the RUL prediction task [28], in which the RUL value of the engine is set to a fixed value from the initial operation to the value that reaches the segmentation point after a period of time of operation and then begins to decline rapidly. Afterward, by observing the dataset, the average run length of all the engines was around 210 and the minimum run period was 127. Heimes first proposed to set the segmented degradation point between 120 and 130 [29], but this was the result of their observation and inference using a multilayer perceptron (MLP).

As we consider the tendency of segmental degradation in the RUL prediction process to improve the global compression of SE, we perform segmental compression on the engine sensor data as input and then divide each segment into two for squeezing and excitation, and at the same time, we adopt 115 as the segmentation point of linear segmental degradation in the experiments, which is shown in Figure 6, and at the same time, we set the time window size of all three data subsets from FD001 to FD003 in the experimental data to 35, which is the same as the time window size of the most complex data subset FD004 in Section 2. At the same time, to ensure the reduction of the mutual influence between segment compression and RUL segment degradation, the time window sizes of the three data subsets from FD001 to FD003 in the experimental data are all set to 35, and the time window of the most complex data subset FD004 is set to 25, which is consistent with the size of the pooling window of the pooling layer in the segment compression in Section 2. In this way, the improvement of SE is realized based on the actual degradation of RUL, and the mutual influence of different degradation states of RUL is avoided to the greatest extent, to realize the retention of more data change content and improve the model prediction performance. We realize the improvement of SE based on the distribution of RUL and actual data, combined with the feature importance assessment of random forest, and we will demonstrate its excellent performance in the following.

In the SSE, the hyperparameter "ratio" is set to 1, considering the differences between sequential data and image data to reduce the computational complexity and cost of the model [30, 31]. The batch size in the model is set to 1024, and the training epochs are set to 1000. Early stopping is configured with 40 patients during training and 20 during testing. The chosen optimizer is Adam. The learning rate is reduced by a factor of 0.5 when the loss stops improving, with the minimum learning rate set to 0.0001.

*3.3. Model Evaluation Indicator.* To evaluate the RUL prediction performance of the CNN model with feature importance selection and the SSE, we utilize two model evaluation metrics: root mean square error (RMSE) and the scoring function (S), where RMSE is a more common metric in prediction tasks. $y_i$ represents the true value and $f(x)$ represents the predicted value. The formula is shown in the following equation:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - f(x_i))^2}. \tag{9}$$

RMSE is generally used to quantify the divergence between expected values and actual values, and it is sensitive to outliers. When there is a considerable gap between anticipated values and actual values, RMSE grows appropriately.

The scoring function, on the other hand, is specifically designed for RUL prediction tasks and sets itself apart from other evaluation metrics by placing a stronger emphasis on predicting the timing of failure, particularly in the later stages of the task. In the realm of industrial equipment operation, it is crucial to minimize losses caused by equipment failures. When the predicted value exceeds the actual RUL

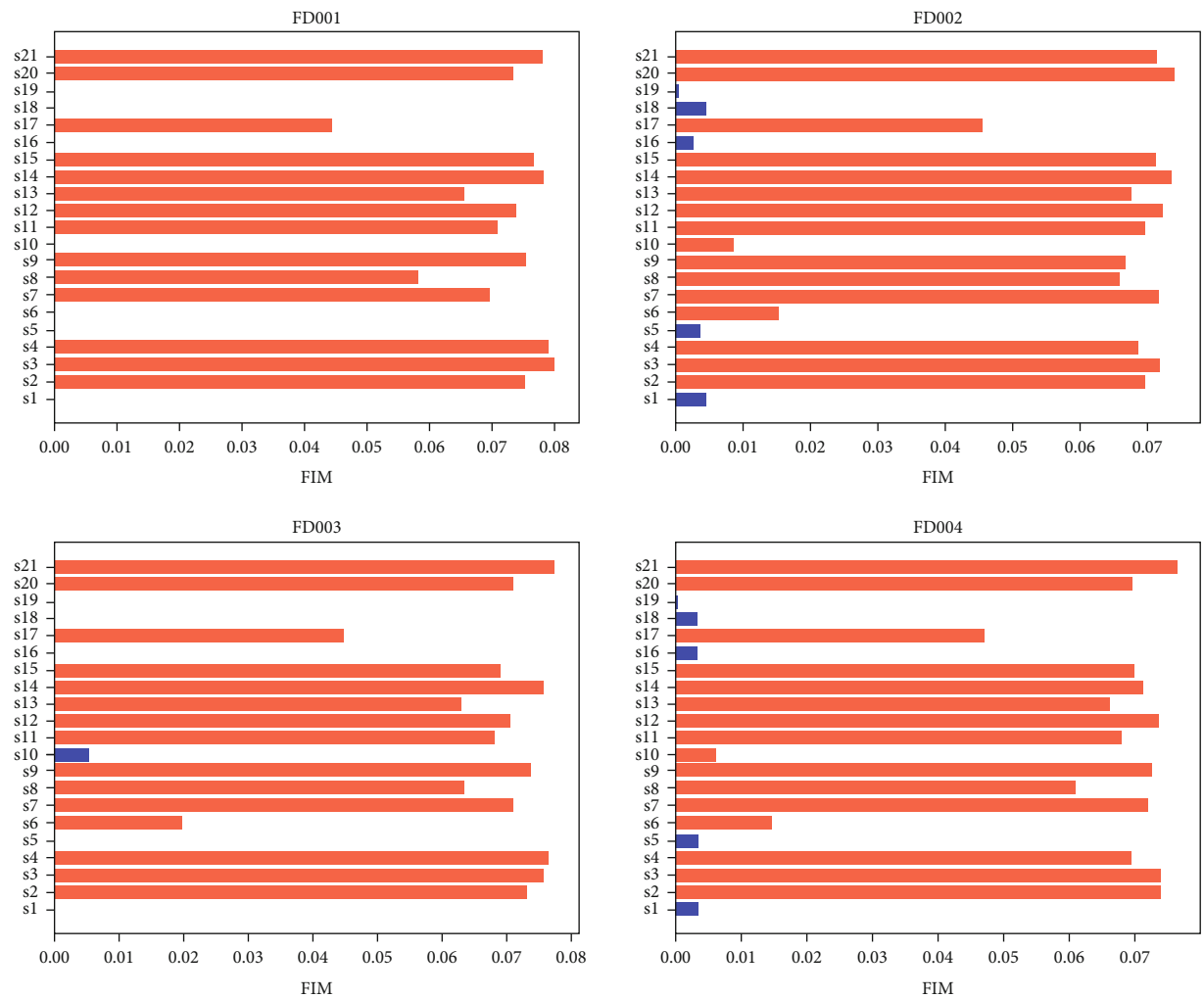


Figure 5: Random forest selection feature results.

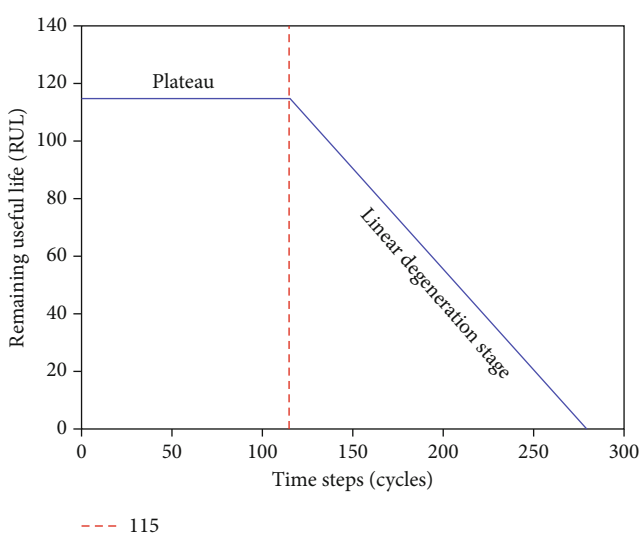


Figure 6: RUL segmented linear descent.

of the equipment, the score is penalized severely. $T_i$ represents the true value and $P_i$ represents the predicted value, as illustrated in the following formula:

$$S = \begin{cases} \sum_{i=1}^{N} \left( e^{(T_i - P_i)/13} - 1 \right), & T_i > P_i, \\ \sum_{i=1}^{N} \left( e^{(P_i - T_i)/10} - 1 \right), & T_i \leq P_i. \end{cases} \tag{10}$$

## 4. Comparative Analysis of Forecast Results

To demonstrate the importance and excellent performance of the random forest feature selection and improvement of SE that we use, we analyze the results of ablation experiments, comparative experiments, segmentation points, and different settings of time step corresponding to the RUL prediction results to demonstrate that our method can effectively improve the performance of the model for the RUL prediction of aeroengine.

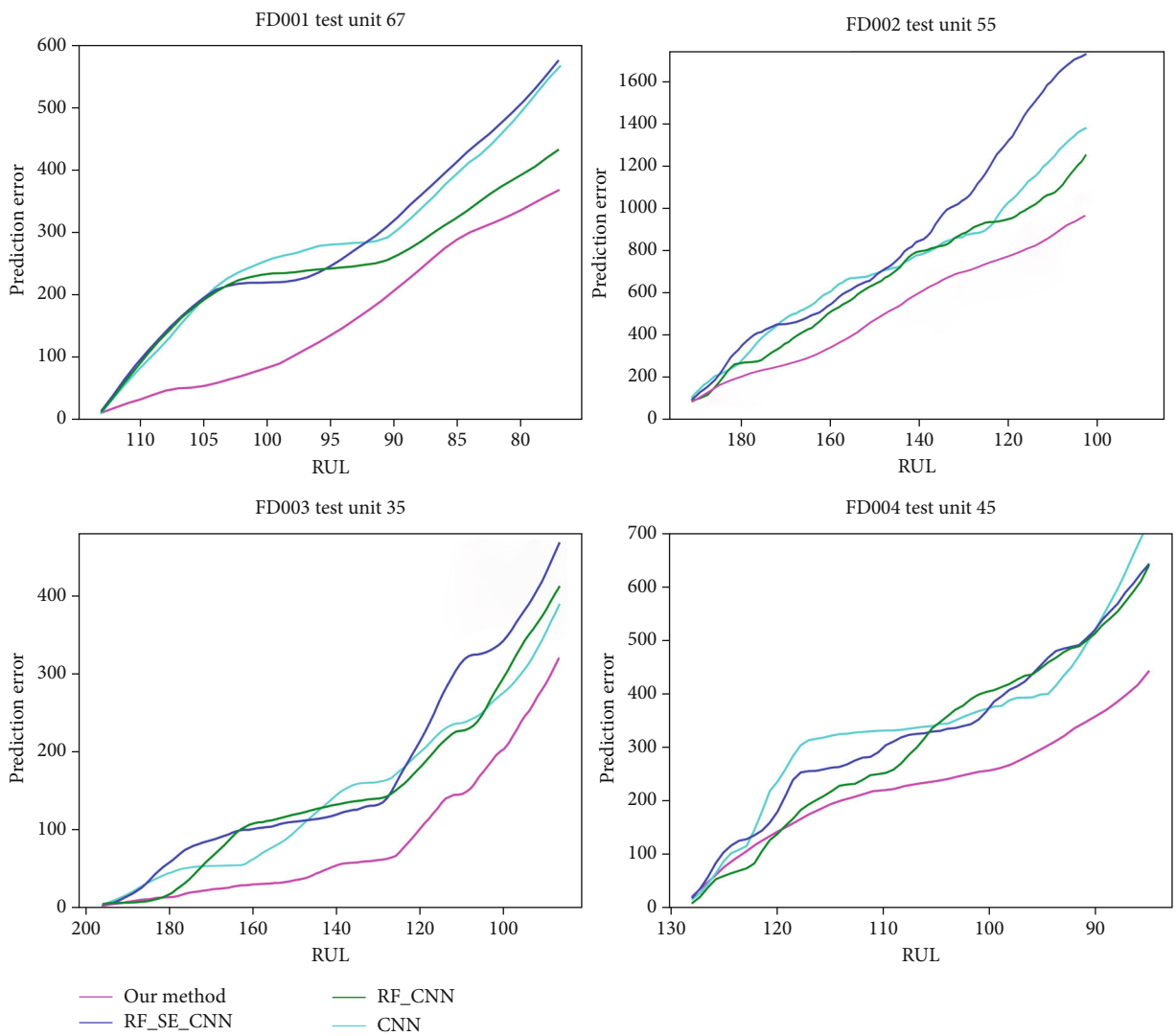


Figure 7: Comparison of methods for error accumulation across datasets.

*4.1. Ablation Experiment.* Firstly, the ablation experiments involve combinations of four different methods: RF+SSE +CNN (our method), RF+SE+CNN, RF+CNN, and CNN. RF symbolizes random forest for feature selection, while SSE represents the upgraded SE block, which features segmented squeezing and activation. To ensure the dependability of the experimental outcomes, each subset is explored with five times during training, and then, the averages are taken. These methods are used for RUL prediction, and the resulting errors are accumulated, as shown in Figure 7, we pick one engine from each of the four subdatasets for demonstration, and it can be seen from the figure that the error accumulation of our method for full life prediction on each test engine is smaller than the other methods, and as time goes by, the RUL prediction task is produced by our method on the smaller errors and more accurate predictions. The excellent performance of our proposed method also gets better and better with continuous prediction after our method passes the gentle decline phase in the RUL prediction task, i.e., when the decline segmentation point RUL is 115. For RUL prediction of aeroengines, the late prediction accuracy is very important. The complete ablation experiment results are shown in Table 2.

We accurately evaluate the engine RUL prediction by aggregating RMSE, score, and prediction loss on the four datasets, clearly and thoroughly exhibiting the engine RUL deterioration features to optimize and enhance the model to effectively improve the model performance.

*4.2. Comparison with Other Methods.* In this section, we will compare our method with other state-of-the-art approaches on the same dataset to demonstrate the excellence of our method and predictive performance. As shown in Table 3, included among these, the hybrid model which is based on a combination of CNN and LSTM exhibits better performance than deep LSTM. DCNN improves feature extraction by using a time window for data preparation. MODBNE is a multiobjective ensemble learning method that evolves multiple deep belief networks for RUL estimation models based on accuracy and diversity. Tafcn introduces a signal

Table 2: Comparison of ablation experiment results.

| Dataset | FD001 | | FD002 | | FD003 | | FD004 | |
|---|---|---|---|---|---|---|---|---|
| Evaluation | RMSE | Score | RMSE | Score | RMSE | Score | RMSE | Score |
| CNN | 14.75 | 433.3 | 17.95 | 3124.7 | 13.58 | 398.0 | 20.23 | 5803.9 |
| RF_CNN | 14.69 | 416.5 | 17.84 | 3045.2 | 13.50 | 344.3 | 20.19 | 3488.3 |
| RF_SE_CNN | 14.51 | 373.0 | 17.76 | 2380.7 | 13.55 | 351.8 | 19.94 | 4442.5 |
| Our method | **13.46** | **265.9** | **16.54** | **1465.1** | **11.79** | **222.0** | **19.39** | **2036.2** |

The bold contents represent the optimal experimental results. The ablation experiments show that our proposed method can effectively improve the prediction performance of the model.

Table 3: Comparison with other state-of-the-art methods on the CMPASS dataset.

| Dataset | FD001 | | FD002 | | FD003 | | FD004 | |
|---|---|---|---|---|---|---|---|---|
| Evaluation | RMSE | Score | RMSE | Score | RMSE | Score | RMSE | Score |
| Hybrid [16] | 14.53 | 322.4 | NA | NA | NA | NA | 27.08 | 5649.1 |
| DCNN [8] | **12.61** | 274 | 22.36 | 10412 | 12.64 | 284 | 23.31 | 12466 |
| MOBNE [32] | 15.04 | 334 | 25.05 | 5585 | 12.51 | 422 | 28.66 | 6558 |
| Deep LSTM [17] | 16.14 | 338.0 | 24.49 | 4450.0 | 16.18 | 852.0 | 28.17 | 5550.0 |
| BiLSTM [33] | 13.65 | 261 | 23.18 | 4130 | 12.74 | 317 | 24.86 | 5430 |
| SBRNN [34] | 13.58 | 228 | 19.59 | 2650 | 19.16 | 1727 | 22.15 | 2901 |
| BiGRU [35] | 12.65 | **213** | 18.9 | 2264 | 12.5 | 233 | 20.5 | 3610 |
| Tafcn [12] | 13.99 | 336 | 17.06 | 1946 | 12.01 | 251 | 19.79 | 3671 |
| Our method | 13.46 | 265.9 | **16.54** | **1465.1** | **11.79** | **222.0** | **19.39** | **2036.2** |

The bold contents represent the optimal experimental results. This table shows excellent performance by comparing the results of the multiple research methods and comparing our method with other state-of-the-art methods.

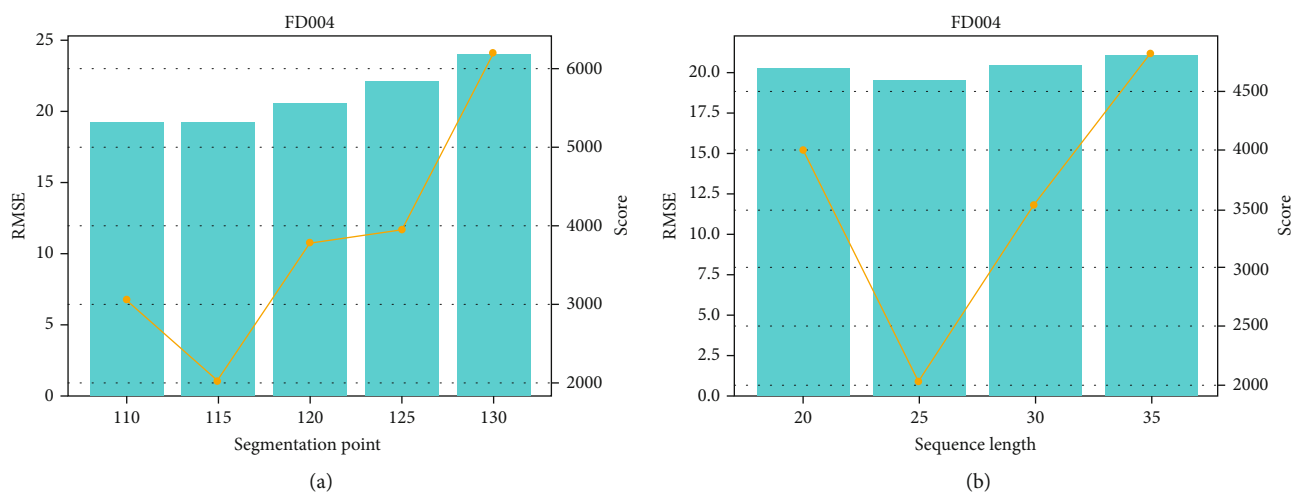


Figure 8: (a) Segmented point experiment results show. (b) Segmented point experiment results show.

selection method and an end-to-end RUL prediction model using a fully convolutional network, achieving highly accurate RUL predictions.

Our method obtains the best prediction performance compared to other methods, only slightly poorer than individual methods in the simpler FD001 dataset of the four subdatasets and practically concurrently in the other three datasets. A series of ablation tests and comparison experiments are sufficient to show that we realize the prediction of RUL through the feature selection of random forest and the improvement of SE, which is combined with CNN, and the experimental results confirm the feasibility of our whole method and demonstrate its excellent prediction performance.

*4.3. Analytical Experiment.* In this section, we analyze the effect on the method performance under different RUL degradation segmentation points and different time window parameters. Considering that FD004 is the most complex dataset among the four datasets, FD004 is chosen for experimental demonstration.

In the RUL prediction task, the segmentation degradation model is often taken as the objective function, and in

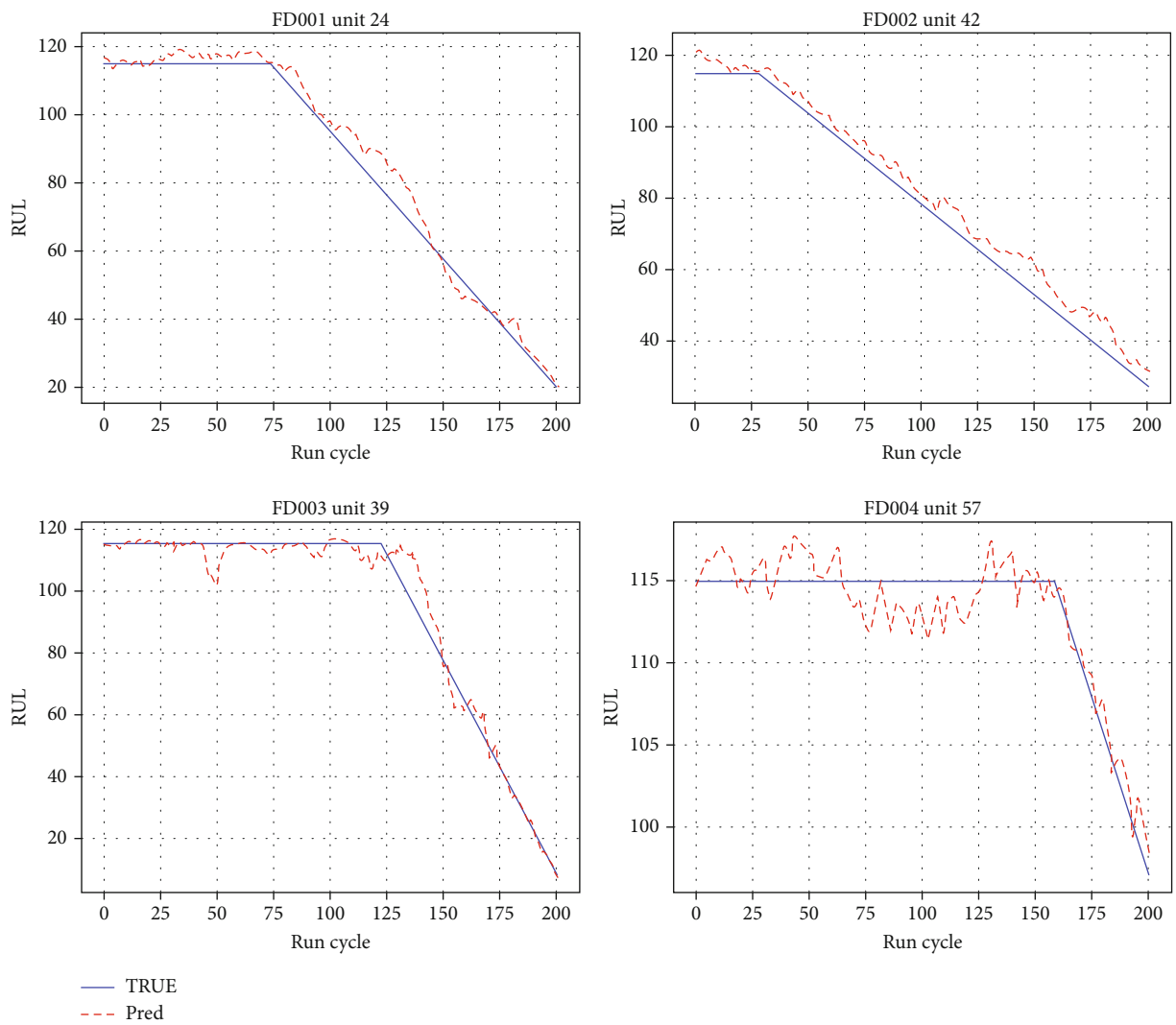


Figure 9: Test engine RUL prediction plots in four data subsets.

the CMPASS dataset, the average period of each engine run is about 210, and the authors who first proposed the segmentation point thought that it seems reasonable that the segmentation point should be set at 105, but the selection of the segmentation point between 120-130 was not systematically verified by experiments. We considered this issue and conducted experiments when making improvements to the SE. Considering that the setting of the segmentation point will set some values larger than the segmentation point as constants and that the segmentation point should not be too small, otherwise it will cover up some data changes, and we chose to set the segmentation point between 110 and 130 and demonstrated the experimental results, as shown in Figure 8(a). By observing the images, it can be clearly seen that the setting of the segmentation point has an impact on the experimental results. As the value of the segmentation point decreases, at the point of 115, both RMSE and score reach their lowest values. Subsequently, when the value of the breakpoints is further decreased, both RMSE and score values will rise, thus proving the rationality of the breakpoint setting and its ability to improve the performance of the model.

When improving the SE, to ensure as much as possible that the constant phase and the fast descending phase in the data are split, so that the two parts of the data do not affect each other as much as possible, combined with the fact that the average running period of all the engines is different, it is necessary to choose a suitable time window to accept the data and then compress the data after splitting it. Therefore, to satisfy most of the data segmentation cases as much as possible, we consider setting the time window value at 35. It should be noted that in the four data subsets, FD001 and FD002 and FD003 and FD004 have almost the same average engine running cycle, but the data volume and complexity of FD004 are much more than that of FD003. Therefore, we set the same time window, size 35, for FD001, FD002, and FD003. For FD004, set it to 25 individually, to better capture more useful change information from the complex data and reduce the impact on the experiment; the size of the window will also be affected. The setting of the time window for the

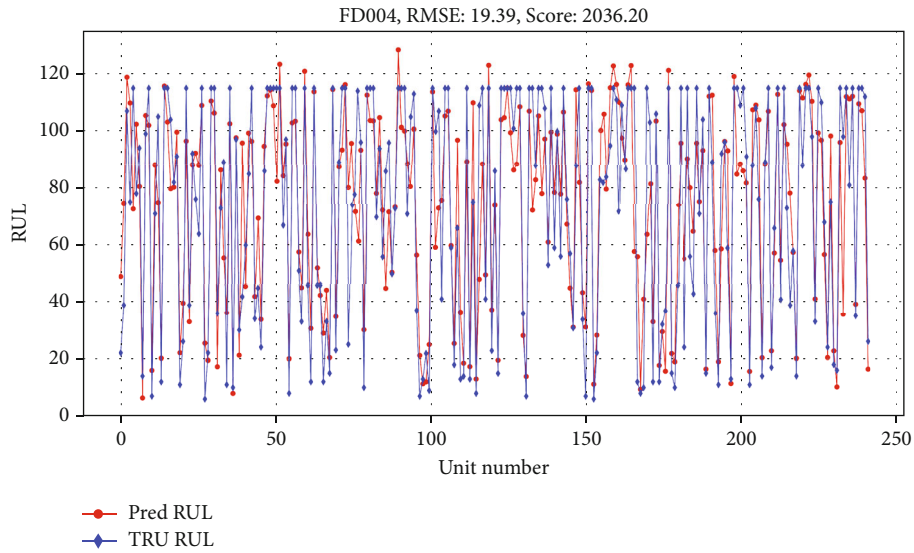


Figure 10: Predicted results for all test engines in FD004.

FD004 dataset and the impact it has on the experimental results is demonstrated as shown in Figure 8(b).

*4.4. Test Set Prediction Results.* To show the performance of the CNN model with feature selection and enhanced SE on the CMPASS dataset, we picked one test engine from each of the four datasets for RUL prediction. As can be seen in Figure 9, our predictions are very near to the genuine values and, in many cases, greater than the true values, and we have strong prediction performance on all datasets, which is highly essential in RUL prediction tasks. We also selected to provide all test engine prediction results for the more complex FD004 dataset, as shown in Figure 10. The results show that we are able to accurately achieve RUL prediction on both basic and sophisticated datasets.

## 5. Conclusions and Future Work

In this research, we propose to evaluate the importance of sensor features by random forest to filter more useful information for the prediction task. The SE is improved according to the actual changes in RUL, and segmented compression is realized to retain more data with upward and downward trends, which are then sent to the CNN for training to complete the prediction task. Experimental results reveal that our proposed approach may effectively improve the performance of CNN and provide high-accuracy RUL prediction results on the CMPASS dataset. To highlight the requirement of our strategy, we conduct a series of ablation and comparison experiments, which prove that the combination of random forest feature selection and improvement of SE may effectively enhance the performance of the model. In addition, we also compare our results with other state-of-the-art approaches, and our results indicate greater performance. In our study, we have realized the inspiration for the improvement of SE by combining the actual situation of RUL change; considering the different degrees of influence of feature dimension and window step, we will study the feature dimension, window step, and time series in the future and combine long short-term memory and SSE.

## Data Availability

Data used in this work have been properly cited within the article.

## Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Authors' Contributions

All the related authors contribute to the conceptualization, data curation, investigation, methodology, writing of the original draft, and writing, reviewing, and editing of the manuscript.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China 456 (61863016).

## References

[1] J. Wei, P. Bai, D. Qin, T. C. Lim, P. Yang, and H. Zhang, "Study on vibration characteristics of fan shaft of geared turbofan engine with sudden imbalance caused by blade off," *Journal of Vibration and Acoustics*, vol. 140, no. 4, 2018.

[2] P. Hong, H. U. Changhua, S. I. Xiaosheng, Z. Jianxun, P. Zhenan, and Z. Peng, "Review of machine learning based remaining useful life prediction methods for equipment," *Journal of Mechanical Engineering*, vol. 55, no. 8, pp. 1–13, 2019.

[3] M. Jouin, R. Gouriveau, D. Hissel, M. Péra, and N. Zerhouni, "Particle filter-based prognostics: review, discussion and

perspectives," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 2–31, 2016.

[4] D. Wang, F. Yang, K. L. Tsui, Q. Zhou, and S. J. Bae, "Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1282–1291, 2016.

[5] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 227–237, 2012.

[6] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2276–2285, 2017.

[7] A. Soualhi, H. Razik, G. Clerc, and D. Doan, "Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2864–2874, 2014.

[8] X. Li, Q. Ding, and J. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[9] Y. J. Wang, S. P. Li, S. Q. Kang, J. B. Xie, and V. I. Mikulovich, "A method for predicting the remaining service life of rolling bearings by combining CNN and LSTM," *Vibration Testing and Diagnostics*, vol. 3, no. 41, pp. 439–446, 2021.

[10] B. Yang, R. Liu, and E. Zio, "Remaining useful life prediction based on a double-convolutional neural network architecture," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9521–9530, 2019.

[11] R. Jin, M. Wu, K. Wu, K. Gao, Z. Chen, and X. Li, "Position encoding based convolutional neural networks for machine remaining useful life prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 8, pp. 1427–1439, 2022.

[12] L. Fan, Y. Chai, and X. Chen, "Trend attention fully convolutional network for remaining useful life estimation," *Reliability Engineering & System Safety*, vol. 225, article 108590, 2022.

[13] C. Peng, Y. Chen, Q. Chen, Z. Tang, L. Li, and W. Gui, "A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion," *Sensors*, vol. 21, no. 2, p. 418, 2021.

[14] Z. Shi and A. Chehade, "A dual-LSTM framework combining change point detection and remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 205, article 107257, 2021.

[15] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, and Y. Wang, "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network," *ISA Transactions*, vol. 97, pp. 241–250, 2020.

[16] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2021.

[17] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1197–1207, 2021.

[18] J. M. Corcoran, J. F. Knight, and A. L. Gallant, "Influence of multi-source and multi-temporal remotely sensed and ancillary data on the accuracy of random Forest classification of wetlands in Northern Minnesota," *Remote Sensing*, vol. 5, no. 7, pp. 3212–3238, 2013.

[19] M. Belgiu, I. Tomljenovic, T. J. Lampoltshammer, T. Blaschke, and B. Höfle, "Ontology-based classification of building types detected from airborne laser scanning data," *Remote Sensing*, vol. 6, no. 2, pp. 1347–1366, 2014.

[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018, Open Access.

[23] X. Jin, Y. Xie, X. Wei, B. Zhao, Z. Chen, and X. Tan, "Delving deep into spatial pooling for squeeze-and-excitation networks," *Pattern Recognition*, vol. 121, article 108159, 2022.

[24] A. Paul, A. Dey, D. P. Mukherjee, J. Sivaswamy, and V. Tourani, "Regenerative random forest with automatic feature selection to detect mitosis in histopathological breast cancer images," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., vol. 9350 of Lecture Notes in Computer Science(), Springer, Cham, 2015.

[25] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. N. Low, and S. S. Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, pp. 915–920, Melbourne, VIC, Australia, 2019.

[26] E. Ramasso and A. Saxena, "Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset," in *Annual Conference of the Prognostics and Health Management Society 2014*, Fort Worth, TX, USA, United States, 2014.

[27] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliability Engineering & System Safety*, vol. 183, pp. 240–251, 2019.

[28] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 88–95, Dallas, TX, USA, 2017.

[29] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 International Conference on Prognostics and Health Management*, pp. 1–6, Denver, CO, USA, 2008.

[30] G. Sateesh Babu, P. Zhao, and X. L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications. DASFAA 2016*, S. Navathe, W. Wu, S. Shekhar, X. Du, X. Wang, and H. Xiong, Eds., vol. 9642 of Lecture Notes in Computer Science(), Springer, Cham, 2016.

[31] C. Liu, L. Zhang, J. Niu, R. Yao, and C. Wu, "Intelligent prognostics of machining tools based on adaptive variational mode decomposition and deep learning method with attention mechanism," *Neurocomputing*, vol. 417, pp. 239–254, 2020.

[32] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017.

[33] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network," in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pp. 1037–1042, Chongqing, China, 2018.

[34] W. Yu, I. Y. Kim, and C. Mechefske, "An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme," *Reliability Engineering & System Safety*, vol. 199, article 106926, 2020.

[35] J. Zhang, Y. Jiang, S. Wu, X. Li, H. Luo, and S. Yin, "Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism," *Reliability Engineering & System Safety*, vol. 221, article 108297, 2022.