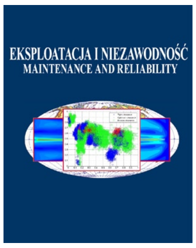


Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 23 (2021), Issue 4

journal homepage: http://www.ein.org.pl

Article citation info:
Lyu Y, Jiang Y, Zhang Q, Chen C. Remaining useful life prediction with insufficient degradation data based on deep learning approach. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2021; 23 (4): 745–756, http://doi.org/10.17531/ein.2021.4.17.

Remaining useful life prediction with insufficient degradation data based on deep learning approach

Indexed by: Web of Science Group

Yi Lyu[a,*], Yijie Jiang[b], Qichen Zhang[c], Ci Chen[b]

[a] University of Electronic Science and Technology of China Zhongshan Institute, School of Computer, Zhongshan, China, 528400
[b] Guangdong University of Technology, Guangzhou, School of Automation, China, 510006
[c] University of Electronic Science and Technology of China, School of Computer Science and Engineering, Chengdu 611731

Highlights

- Focus on the improvement of the RUL prediction effect in the case of insufficient degradation data.
- A data amplification network based on cycleGAN is designed to effectively increase the size of the degradation dataset.
- A RUL prediction framework is constructed with the sliding time window strategy and BiLSTM network.
- Experimental results show the RUL prediction performance has been significantly improved by the proposed data amplification approach.

Abstract

Remaining useful life (RUL) prediction plays a crucial role in decision-making in condition-based maintenance for preventing catastrophic field failure. For degradation-failed products, the data of performance deterioration process are the key for lifetime estimation. Deep learning has been proved to have excellent performance in RUL prediction given that the degradation data are sufficiently large. However, in some applications, the degradation data are insufficient, under which how to improve the prediction accuracy is yet a challenging problem. To tackle such a challenge, we propose a novel deep learning-based RUL prediction framework by amplifying the degradation dataset. Specifically, we leverage the cycle-consistent generative adversarial network to generate the synthetic data, based on which the original degradation dataset is amplified so that the data characteristics hidden in the sample space could be captured. Moreover, the sliding time window strategy and deep bidirectional long short-term memory network are employed to complete the RUL prediction framework. We show the effectiveness of the proposed method by running it on the turbine engine data set from the National Aeronautics and Space Administration. The comparative experiments show that our method outperforms a case without the use of the synthetically generated data.

Keywords

deep learning, remaining useful life, degradation data, data amplification, cycle-consistent generative adversarial network.

1. Introduction

Accurate prediction of the remaining useful life (RUL) is extremely valuable for decision-making in condition-based maintenance for preventing catastrophic field failure. For degradation-failed products, the data of performance deterioration process plays a major role in RUL estimating. The methods of RUL estimation can be divided into three categories: 1) method based on failure mechanism analysis [9, 22], 2) method based on data-driven approach, and 3) hybrid method that combines the first two. The key point of RUL prediction using the first method is to fully understand the degradation mechanism of the target equipment. Prior knowledge in the target field is indispensable when establishing a mathematical model of the degradation process. However, as the complexity of the equipment increases and automation advances, obtaining complete knowledge of the degradation mechanism becomes difficult [7, 14]. The aircraft turbine engine data set of the National Aeronautics and Space Administration (NASA) was built from more than ten sensors. These data should be analyzed together to reveal the health indicators of the turbine engine. Different from the method based on failure mechanism analysis, the data-driven approach does not require researchers to have a comprehensive understanding of the target equipment [12, 23]. After collecting sufficient degradation data from sensors, researchers could constructs a nonlinear mapping between degradation data and the real equipment health indicators, and meanwhile solves the dynamic dependency problems [8, 28]. This nonlinear mapping network can be used to predict the RUL of the equipment used on site.

Data-driven methods, especially the deep learning approach have developed substantially in recent years [3, 6, 16–18, 24]. Considering the problem of weak dependence of time-series information, Zhu [36] combined the information of the previous convolutional layer with the current layer and proposed a multiscale convolutional neural network (CNN) for RUL prediction. The long-range dependence problem exists in many studies on time-series data. Li [11] selected the long short-term memory network (LSTM) and CNN as the base model to build the RUL prediction model. LSTM can save past information for the current network parameter update and CNN has a

(*) Corresponding author.

E-mail addresses: Y. Lyu - lvyi913@zsc.edu.cn, Y. Jiang - yijiejiang@live.com, Q. Zhang - zhangqichen0708@163.com, C. Chen - gdutcc@gmail.com

strong ability in local feature extraction. The combination of the two improves the accuracy of the prediction network. Group method of data handling-type neural network (GMDH) can self-organize and generate the optimal network structure based on the training data [22]. Ge [4] generates three GMDH networks through different division of training data, and integrates the results of the three GMDH networks with a three-layer back propagation (BP) neural network to solve the disadvantage of local optimum of GMDH and improve the generalization ability. A.Ragab [19] developed a data-driven prognostic methodology using both the age and condition monitoring data as inputs, which can deal with any number of condition indicators. Under different test conditions, different workloads, environmental conditions and noise levels may lead to different distribution of training set and test set. To solve this problem, Wen [26] used domain-adversarial neural network(DANN) and proposed a data-driven framework with domain adaptability using a bidirectional gated recurrent unit (BGRU). This method can effectively reduce the impact on the performance of RUL prediction due to the different distribution of training data and testing data. Deep learning methods are adopted to address the RUL prediction issue of a specific field, such as bearings [20, 36], lithium-ion batteries [32, 34], lathe tool wear [37], and nuclear systems [38].

Nevertheless, the estimation effect of these mentioned methods is highly dependent on the capacity of the degradation data set. That means the scale of the dataset available in model training phase has a great influence on the RUL prediction accuracy [13]. Abdulraheem [1] explored the effect of the dataset size on prediction results under supervised learning techniques, their findings showed that the model with the largest dataset had the best prediction effect under three datasets listed as dataset size of 400, 800, and 1200. The larger the dataset is, the better is the model established. However, in many actual industrial production practices, obtaining a largescale dataset is not realistic due to the longer degradation time and high cost of collecting degradation data. The XJTUSY rolling bearings dataset mentioned by Wang [25] only collected the complete life cycle of 15 bearings (type LDK UER204), the entire life cycle is only 42h and 18min. Many restrictions on obtaining large-scale degradation data restrict the further development of deep learning data-driven methods in RUL prediction. Moreover, for those newly emerging equipment, there is also a lack of degradation data. Under these scenarios, the RUL prediction performance will be severely affected. Hence, how to improve the prediction accuracy with insufficient degradation data is yet a challenging task.

In the case of insufficient degradation data, the low accuracy of RUL prediction is mainly caused by the low sample diversity, which can be effectively improved by data augmentation [29]. Generative adversarial network (GAN) is a common data augmentation strategy, which can capture the characteristics hidden in the sample space and enrich the diversity of samples [30]. Yoon [31] applied the GAN to the task of generating medical data and produced a patient electronic health dataset containing discrete time series data. In the sequence data generation task, Li [10] utilized GAN to capture the temporal correlation of time series distributions, the generator and discriminator inside the GAN adopt the LSTM network as the basic network, which is friendly to time-series data. Subsequently, Xie [27] generated bearing datasets for various working conditions based on the cycle-consistent generative adversarial network (CycleGAN) framework and its GAN discriminator was trained for fault diagnosis.

Based on the above research, this study developed a complete framework to improve the RUL prediction performance when degradation data is insufficient. Four steps are involved in this framework. Firstly, constructing a data amplification model using the LSTM network which is also as the Generator inside the CycleGAN and mining the inherent distribution of existing degradation data samples of a machine. Second, a data preprocessing strategy is designed for time-series degradation data before they are sent to the augmentation network. Third, the obtained amplified data are preprocessed using sliding time window method and their labels for prediction model training are obtained. Finally, a data-driven method is built with amplified data for RUL prediction. The contributions of this study are summarized as follows:

- Proposed an amplification network for generating time series degradation data based on CycleGAN; this method uses a small amount of data to train CycleGAN and uses the designed generator based on the LSTM network for data amplification without excessive prior knowledge of the data.
- Designed a data preprocessing strategy to resize the time-series degradation data before they are sent to the designed amplification network.
- Constructed a data-driven RUL prediction model and integrated the above work into a complete set of RUL prediction methods, which is suitable for the degradation data of time-series.
- Compared the performance differences between RUL prediction models trained with amplified data obtained from various amounts of degradation data.

The rest of this paper is organized as follows. Theoretical foundation of the CycleGAN is introduced in section II. Proposed an amplification network based on LSTM and related theory of data preprocessing strategy and RUL prediction model constructed are introduced in section III. An experiment is introduced in section IV. The conclusions are summarized in section V.

2. Theoretical Foundation

CycleGAN is a type of unsupervised learning generative network that was designed to solve the problem of image-to-image translation in the field of vision and graphics by learning the mapping between a set of aligned image pairs from source domain to target domain. The key to achieve this function is an adversarial structure composed of two networks called generator and discriminator. The generator captures the distribution of the true image and constructs a fake one, and the discriminator estimates the probability that the image came from the true image rather than the generator. Ideally, the discriminator's recognition success rate should be approximately equal to 0.5, which means that the discriminator cannot distinguish whether the test image is real or generated, that is, the generator obtained the true mapping between image pairs. To ensure improved learning efficiency, we built a cycle-consistent structure from two directions. Two generators and two discriminators are used in each direction; one of the generators is used to transform the data from $domainA$ to $domainA$, and the other generator aims to reconstruct the generated data back to $domainA$. The structure is shown in Figure 1.

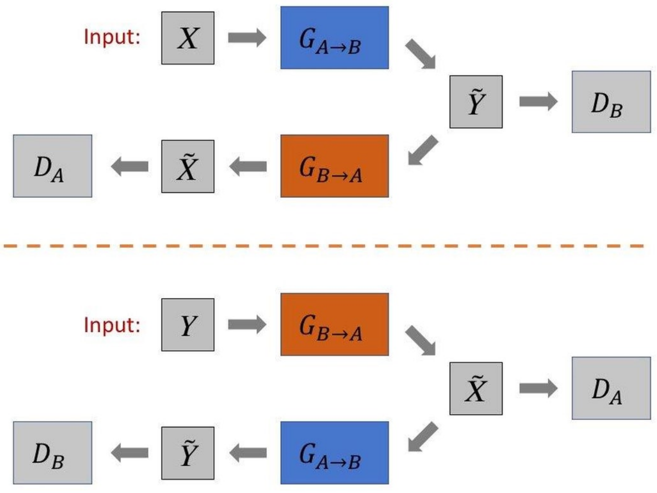


Fig. 1. Structure of CycleGAN

There two types of data X with $domainA$ and data Y with $domainB$. In the upper part, data $X = \left\{x_A^1, x_A^2, \ldots, x_A^m\right\}$ from $domainA$

are sent into the generator $G_{A\to B}$ randomly. The generated data $\tilde{Y}=\left\{\tilde{y}_B^1,\tilde{y}_B^2,\ldots,\tilde{y}_B^m\right\}$ obtained with probability distribution are similar to *domainB* , Discriminator D_B distinguishes the generated data $\tilde{Y}$ and from the real data $Y=\left\{y_B^1,y_B^2,\ldots,y_B^m\right\}$. The generated data $\widetilde{Y}$ obtained through $G_{A\to B}$ are sent to generator $G_{B\to A}$. The reconstructed data $\tilde{X}=\left\{\tilde{x}_A^1,\tilde{x}_A^2,\ldots,\tilde{x}_A^m\right\}$ obtained from the generator $G_{B\to A}$ are distinguished with the real data X of *domainA* via discriminator D_A.

Value function is shown in Formula 1. To simplify the function, we define $G_{A\to B}$ as G and $G_{B\to A}$ as F.

$$\begin{aligned}\min_G \max_{D_Y} \mathcal{L}_{GAN}(G,D_Y,X,Y) &= \mathbb{E}_{y\sim p_{data}(y)}\left[\log D_Y(y)\right] \\ &+ \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1-D_Y\left(G(x)\right)\right)\right]\end{aligned} \quad (1)$$

In the process of optimizing this value function, the distribution of the data generated by the generator G is updated close to *domainB* , and the discriminator D_Y distinguishes the generated data from the real data. The value function aims to minimize the generation error of G, and maximize the recognition success rate of D_Y . Similarly, we can obtain the value function of another generator F and discriminator D_X:

$$\begin{aligned}\min_F \max_{D_X} \mathcal{L}_{GAN}(F,D_X,Y,X) &= \mathbb{E}_{x\sim p_{data}(x)}\left[\log D_X(x)\right] \\ &+ \mathbb{E}_{y\sim p_{data}(y)}\left[\log\left(1-D_X\left(F(y)\right)\right)\right]\end{aligned} \quad (2)$$

Combining both two parts shown above can obtain a cycle-consistency loss:

$$\begin{aligned}\mathcal{L}_{cyc}(G,F) &= \mathbb{E}_{x\sim p_{data}(x)}\left[\| F\left(G(x)\right)-x \|_1\right] \\ &+ \mathbb{E}_{y\sim p_{data}(y)}\left[\| G\left(F(y)\right)-y \|_1\right]\end{aligned} \quad (3)$$

The value function is shown as Formula 4:

$$\begin{aligned}\operatorname*{argmin}_{G,F} \max_{D_Y,D_X} \mathcal{L}_{\text{GAN}}(G,F,D_X,D_Y) &= \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) \\ &+ \mathcal{L}_{\text{CAN}}(F,D_X,Y,X) \\ &+ \mathcal{L}_{\text{cyc}}(G,F)\end{aligned} \quad (4)$$

3. Methodology

In the task of RUL prediction with data-driven approach, the actual effect of the model is largely determined by the data size. Insufficient run-to-failure degradation data are the key to limit the reliability of the prediction model. This work focuses on how to mine potential data distribution information from limited samples and improve the effect of the RUL prediction model using deep learning technology.

In the model our primary hypothesis is that the time series degradation data used to construct RUL predictions are scarce. If the deep learning method is directly used to summarize the degradation features from the limited degradation data and perform RUL prediction, then the prediction effect will not be as good as expected. We proposed a method that consists of three parts. The first part is an amplification network designed by the LSTM network, which can mine the data distribution information from known samples to expanding sample size [15]. The second is a designed data preprocessing strategy. Owing to the time-dependent dynamic characteristics of the degradation data, the sliding time window strategy is used to fix the dynamic degradation information of the data and adjust the size of the degradation data before sending them to the amplification network to improve the network processing efficiency. The third is described as follows: using the amplified data obtained from the first part to construct a prediction network mainly based on bidirectional long short-term memory (BiLSTM); in the training process, the cyclic neural structure in BILSTM can effectively solve the problem of long-range dependence in time series, obtain the optimal parameters of the model through the backpropagation algorithm, and construct an RUL prediction network to predict the samples.

3.1. Data Amplification Network Based on CycleGAN

In CycleGAN, using the data of two different domains, the generator can make the mutual conversion of the data from the two domains through the adversarial with the discriminator. To obtain the information of the sparse degradation data in our hypothesis, we replaced the data of the two domains with the degradation data of a single domain. Unlike the previous CycleGAN in which the two generators learned the distribution information from one domain, the scheme we proposed aims to learn from each other with scarce degradation data, and the trained generator is used to complete the generation of degradation data.

The generator based on the LSTM was designed as the amplification network. LSTM is a type of recurrent neural network whose structure contains units with functions such as forgetting and remembering; this network is suitable for processing time series data [35]. In actual situations, the degradation data of the device is usually strongly correlated with time and can be used to solve long-range dependence problems [21].

To establish a connection in the calculation unit cycle at each moment, three gate structures in LSTM was designed, namely, forget gate layer, input gate layer and output gate layer. These gate structures control the information flow at different times, and store short-term time-step dependent information for network parameter update, which alleviate the problem of gradient disappearance or gradient explosion of the classic neural network structure during backpropagation. The LSTM cell structure at time t is shown in Figure 2. The input of the current moment consists of the data from current moment input and the data from previous output, the input of the next moment is composed of the data from the current moment output and the data from the next moment input. The related formula is shown as follows:

Forget gate layer:

$$f_t=\sigma\left(W_f\cdot\left[h_{t-1},x_t\right]+b_f\right) \quad (5)$$

Input gate layer:

$$i_t=\sigma\left(W_i\cdot\left[h_{t-1},x_t\right]+b_i\right) \quad (6)$$

$$\widetilde{C}_t=\tanh\left(W_C\cdot\left[h_{t-1},x_t\right]+b_C\right) \quad (7)$$

$$C_t=f_t^{*}C_{t-1}+i_t*\widetilde{C}_t \quad (8)$$

Output gate layer:

$$o_t=\sigma\left(W_o\left[h_{t-1},x_t\right]+b_o\right) \quad (9)$$

$$h_t=o_t{*}\tanh\left(C_t\right) \quad (10)$$

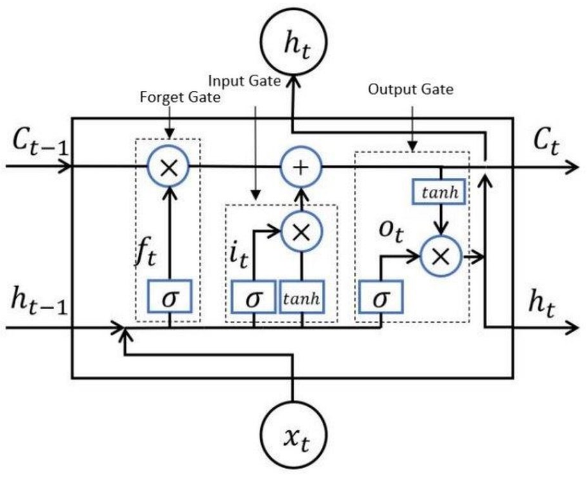


Fig. 2. Cell structure of LSTM at time t

where σ is the activation function, W_f, W_i, W_C, W_o are the weight matrices, x_t *is the input data at time* t, h_t is the output data at time t, C_t represents the information flow participates in parameters updated throughout the entire training process.

As the degradation data is basically a continuous time series, we improved the output form of the LSTM network and fixed the input and output sizes of the generator network to be consistent to improve the spatial structure of the sequence to reduce the loss of degraded information. Specifically, the dimensions of the input and output should be consistent. We saved the output obtained from each h_t of LSTM from timestep 1 to timestep m, which are used to form the final output from the network. The dimension of the output could be a series instead of a scale. The series can meet the requirements of the network for the input data with time dynamic characteristics. The schematic is shown in Figure 3. On the left is an input data with dimension $n{\times}s$, where n represents the length of input data and s represents the dimension of sensors in input data. In the center is the generator with timesteps equal to m. On the right is the first output data with dimension $m \times s$. The second output data are obtained with a dense operation at dimension $n \times s$.

The dimension of the input data $n \times s$ is given by the task, where n represents the length of input data and s represents the dimension of sensors. The timestep of LSTM is about to set a larger number than the length of the input data. In Figure 3, timestep ts is set to m, where $m > n$. In the training process, the first line of the input data $1 \times s$ is sent to the generator, the output of the generator with size of $1 \times timesteps$ consists of values obtained from each timestep After all the input data are sent into the network, all the outputs are combined into a matrix of dimension $n \times m$. Finally, a dense operation is performed to obtain an output data with size consistent the input data.

To ensure that the generated data are similar to the real data in distribution and avoid the difference of actual generated data that affects the characterization of degradation information, we add maximum mean difference (MMD) into the generator's loss function, which is shown as follows:

$$J(G) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}y_i - \tilde{y}_i^{\,2}\right) + MMD \tag{11}$$

where $J(G)$ is the loss function of the generator, n is the number of samples, y_i is the generated sample of i-th instance, and $\tilde{y}_i$ is the target sample of i-th instance.

MMD was designed to measure the difference in data distribution by comparing the statistical information of the two sets of data and was used as a training objective functions for generating networks. In practice, the inner product between the two samples is replaced with the kernel calculation, and the MMD formula is as follows:

$$\begin{aligned} \text{MMD} &= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}K\left(x_i, x_j\right) \\ &\quad - \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}K\left(x_i, y_j\right) \\ &\quad + \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}K\left(y_i, y_j\right) \end{aligned} \tag{12}$$

The inner products are replaced by Gaussian kernel between two samples, and the formula is as follows:

$$K(x, y) = \exp\left(-\|x-y\|^2 / \left(2\sigma^2\right)\right) \tag{13}$$

where σ is the bandwidth. We select a group of different σ, and the calculated MMD is averaged as the final value.

In the training process, we optimize the parameters of the generated model by gradient descent algorithm. The samples generated by the model further reduce the difference between the target samples and enable them to meet the task requirements.

3.2. Data Preprocessing Strategy for Amplification

The RUL of the degradation data for training under ideal conditions should be clear. However, even the same type of equipment has a various life cycle due to different qualities or operating environments. To accurately characterize the temporal dynamics of degradation data, we need a data preprocessing strategy before the degradation data with different life cycles is sent to the amplification network.

The strategy of processing data with inconsistent length of life span is as follows. We obtained the initial value of the rapid data degradation stage through statistical analysis. The initial value of the rapid degradation stage divides the degradation data into a normal stage and a rapid degradation stage. We retain the values of the rapid degradation stage. Then, the process of resizing the data occurs in the normal stage, because the value in the normal stage usually maintains a small range of changes and the significance of predicting the RUL in the normal stage is not as important in the rapid degradation stage.

Given time-series degradation data $X_{s,n}$ with size $s \times n$, as shown in Formula 14, we obtain the output $X_{s,n'}$ that meets the requirements with size $s \times n'$.

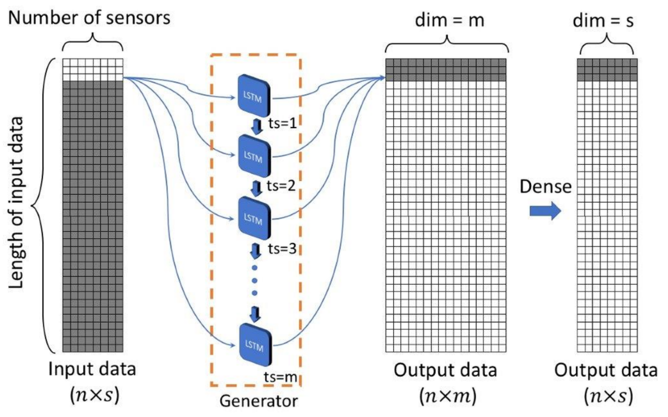


Fig. 3. Structure of the generator based on LSTM

$$\begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{s,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{s,2} \\ \vdots & \vdots & & \vdots \\ x_{1,m} & x_{2,m} & \cdots & x_{s,m} \\ \vdots & \vdots & & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{s,n} \end{bmatrix} \Rightarrow \{X\}_{s\times n} \quad (14)$$

where s represents the number of data features and n represents the life span of the degradation data. We assume that the initial value of the rapid degradation stage obtained by the statistical analysis is m.

In the rapid degradation stage, the value is directly retained without any processing. In the normal stage, two types of resize data strategies are proposed as follows:

1) If the current degradation data length is more than n', then we remove the excess part directly to obtain the data that meets the requirements as follows:

$$\begin{bmatrix} x_{1,l} & x_{2,l} & \cdots & x_{s,l} \\ x_{1,l+1} & x_{2,l+1} & \cdots & x_{s,l+1} \\ \vdots & \vdots & & \vdots \\ x_{1,m} & x_{2,m} & \cdots & x_{s,m} \\ \vdots & \vdots & & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{s,n} \end{bmatrix} \Rightarrow \{X'\}_{s\times n'} \quad (15)$$

The size of the processed data is $s \times n'$, where $l = n - n'$. The excess part is removed from the beginning.

2) If the length of the current degradation data is shorter than n', then we design a data padding strategy. We calculate the average value of the same sensor data in the first time window as the padding data. The substituted x' value for sensor s is expressed as Formula 16.

$$x_s' = \frac{\sum_{n=1}^{L_{tw}} \left(X_{s,n}\right)}{L_{tw}} + \frac{\gamma}{2} \quad (16)$$

where L_{tw} is the length of time window, and γ is a Gaussian noise in the range of $(x_{s,n}^{min} - x_{s,n}^{max})$, which are the maximum and minimum values of the data at sensor s in one time window. The processed data are shown as follows:

$$\begin{bmatrix} x'_{1,1} & x'_{2,1} & \cdots & x'_{s,1} \\ x'_{1,2} & x'_{2,2} & \cdots & x'_{s,2} \\ \vdots & \vdots & & \vdots \\ x'_{1,l} & x'_{2,l} & \cdots & x'_{s,l} \\ x_{1,1} & x_{2,1} & \cdots & x_{s,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{s,2} \\ \vdots & \vdots & & \vdots \\ x_{1,m} & x_{2,m} & \cdots & x_{s,m} \\ \vdots & \vdots & & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{s,n} \end{bmatrix} \Rightarrow \{X'\}_{s\times n'} \quad (17)$$

where $n^0 = l + n$.

3.3. Data Degradation Strategy

The degradation data of the generated network should be processed into the same dimensions as the data during training. The time-series degradation data can be expressed as follows:

$$\begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{s,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{s,2} \\ \vdots & \vdots & & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{s,n} \end{bmatrix} \Rightarrow \{X\}_{s\times n} \quad (18)$$

where s is the number of data features; for instance, bearings data may have features such as vibration, rotation speed, and temperature. n represents the length of the data on the time scale, which can reflect the working time or service life of the data; this value is directly related to RUL.

All the real degradation data are sent into the CycleGAN for training the generator. The first batch of degradation data are sent into the trained generator to obtain the first batch of amplified data. The degradation data of the next batch is obtained from the amplified data of the previous batch, and the amplification is stopped until a predetermined amount of amplified data is obtained. To ensure that the amplified data retains more original degradation information during the iterative process, the number of iterative amplifications should not be excessive.

3.4. RUL Prediction Model Construction

1) *Sliding Time Window Strategy:* For RUL prediction on time-series degradation data, the problem of label identification needs to be solved. One of the intuitive and efficient methods is the sliding time window method [11, 15, 33].

For example, given data sample $X = (x_1, x_2, \ldots, x_n), n = 1,2,3\ldots n$, where n is the length of the data sample on the time scale. we specify the sliding time window size l, then k time windows are obtained which $k = \frac{n}{l} + 1$. Each time window can be expressed as follows and the schematic is shown in Figure 4.

$$\begin{aligned} X^1 &= (x_1, x_2, \ldots, x_l) \\ X^2 &= (x_{l+1}, x_{l+2}, \ldots, x_{2l}) \\ &\ldots \\ X^{k-1} &= \left(x_{(k-1)l+1}, x_{(k-1)l+2}, \cdots, x_{(k-1)2l}\right) \\ X^k &= (x_{n-l}, \ldots, x_{n-1}, x_n) \end{aligned}$$

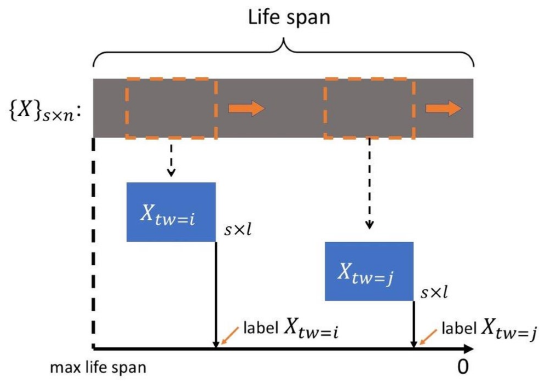


Fig. 4. Schematic of Sliding Time Window

where $X_{tw=i}$ is the ith window. The time window records a piece of information of the degradation data. For complete degradation data, we can obtain k pieces of degradation data and the RUL label of each segment in order.

2) *Prediction Model:* A non-linear mapping from data to labels is built by a data-driven method with sufficient labeled data. We use the deep BiLSTM network [5] to build a prediction model. The difference between LSTM and BiLSTM is that the latter increases the reverse transmission process of data information and contains more hidden layers. The structure of BiLSTM is shown in Figure 5.

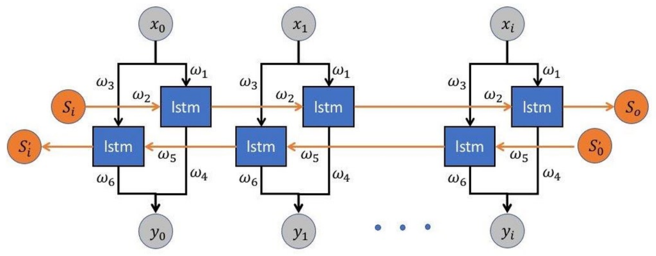


Fig. 5. Structure of BiLSTM

The final output y_t of the bidirectional LSTM consists of three parts: input of the model, input of the forward propagation process, and input of the reverse propagation process:

$$h_t = f\left(w_1 x_t + w_2 h_{t-1}\right) \tag{19}$$

$$h_t' = f\left(w_3 x_t + w_5 h_{t+1}'\right) \tag{20}$$

$$y_t = g\left(w_4 h_t + w_6 h_t'\right) \tag{21}$$

where w_{1-6} represents network parameters, x_t is the input in timestep t, h_t is the value from the forward propagation process, h_t' is the value from reverse propagation process, and g is the activation function.

Owing to the flexibility and versatility of the BiLSTM, a deep network with a stronger non-linear fitting ability was obtained, which is beneficial for RUL prediction by stacking the BiLSTM into three layers. Under this framework, the architecture of a mapping between time window and RUL tag is established, as presented in Figure 6.

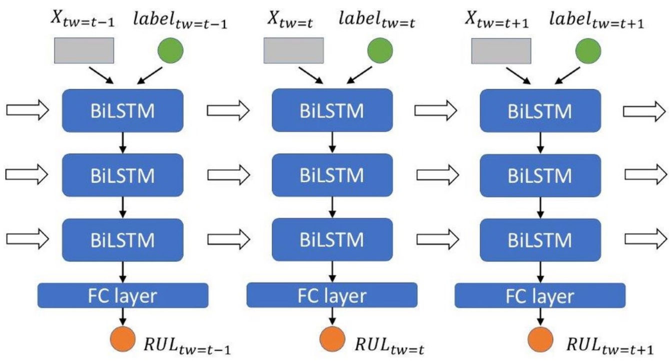


Fig. 6. Structure of RUL prediction model

The main components of the framework are composed of two parts. The first part is a deep learning network composed of stacked BiLSTM. The deep architecture has strong representation capabilities and can learn the time dynamic characteristics between time window degradation data. The other part is a fully connected neural network for regression tasks. Data from stacked BiLSTM which contains degradation information, are used to obtain the predicted RUL from the activation function with ReLU $\left(f(x) = max(0,x)\right)$.

3) *RUL prediction objective:* The parameters in the prediction network are obtained through the back propagation through time (BPTT) algorithm and the given value function is shown as Formula 22. It's defined as the error between the model output and the label:

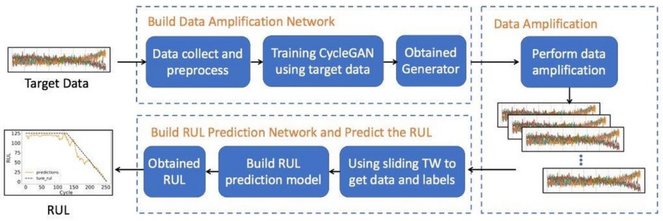
$$L_{rul}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \tilde{y}_i\right)^2 \tag{22}$$

where $\theta = [\mathbf{w_{1-6}}]$ is parameter set of the prediction model and n is the number of units in one batch, y_i and $\tilde{y}_i$ are the model output and label of i-th instance respectively.

3.5. Algorithm Summary

Algorithm of data amplification and RUL prediction is summarized in Algorithm 1. The entire flowchart of data amplification and RUL prediction is shown in Figure 7.

```
Algorithm 1 Algorithm of data amplification and RUL prediction
Input: Historical degradation data $X_{s,n}$
    Data Amplification :
1: Process the original training data by data preprocessing
   strategy to meet the requirements of the amplification network.
2: Train the CycleGAN network to obtain a generator based on designed LSTM.
3: Use the amplification data as the input to the generator and repeat this
   step until the specified amount of training data is obtained.
Output: Amplified Data
    RUL prediction :
4: Use the sliding time window method to obtain training data and corresponding labels.
5: Build a RUL prediction network using BiLSTM.
6: Predict the RUL of with the prediction network.
7: return RUL
```



Build Data Amplification Network
Data collect and preprocess
Training CycleGAN using target data
Obtained Generator
Data Amplification
Perform data amplification
Target Data
Build RUL Prediction Network and Predict the RUL
Obtained RUL
Build RUL prediction model
Using sliding TW to get data and labels
RUL

Fig. 7. Flowchart of data amplification and RUL prediction

4. Experiment

An experiment was conducted to validate that our proposed data amplification strategy can improve the prediction effect by data-driven methods when using insufficient training data. We selected the degradation data with the multi-sensor turbo aero engine dataset from NASA. This dataset contains the operational data of the complete life cycle of multiple turbo aero engines, and each engine contains multiple sensor data. The multi-sensor degradation data have higher requirements for the RUL prediction model and show the universality of our proposed methods.

4.1. Data Preprocessing and Analysis

The turbo-aero engine dataset is divided into four sub-datasets: FD001, FD002, FD003 and FD004. Differences only exist in operating conditions and failure modes, and no dependency exists among the sub-datasets. In this experiment, FD001 was selected as the experimental dataset. FD001 contains the complete degradation data of 100 turbine aero engines. the maximum life span is 362, which means that the entire working cycle of this turbine aero engine is 362. Details of the dataset are shown in Table I.

Table I. C-MAPSS dataset

items	FD001	FD002	FD003	FD004
Engines in dataset	100	260	100	248
Conditions	1	6	1	6
Fault Modes	1	1	2	2
Maximum life span(cycles)	362	378	525	543
Minimum life span(cycles)	128	128	145	128

The sensors are located in all important parts of the turbine aero engine and record the possible parameters related to corresponding degradation indicators. Data from more sensors are considered to provide comprehensive information on engine degradation. Details are shown in Table II.

Table II. C-MAPSS sensors dataset

Num	Symbol	Description	Units	trend
1	T2	Total temperature at fan inlet	°R	~
2	T24	Total temperature at LPC outlet	°R	↑
3	T30	Total temperature at HPC outlet	°R	↑
4	T50	Total temperature at LPT outlet	°R	↑
5	P2	Pressure at fan inlet	psia	~
6	P15	Total pressure in bypass-duct	psia	~
7	P30	Total pressure at HPC outlet	psia	↓
8	Nf	Physical fan speed	rpm	↑
9	Nc	Physical core speed	rpm	↑
10	epr	Engine pressure ratio (P50/P2)	–	~
11	Ps30	Static pressure at HPC outlet	psia	↑
12	phi	Ratio of fuel flow to Ps30	pps psi	↓
13	NRf	Corrected fan speed	rpm	↑
14	NRc	Corrected core speed	rpm	↓
15	BPR	Bypass Ratio	–	↑
16	farB	Burner fuel-air ratio	–	~
17	htBleed	Bleed Enthalpy	–	↑
18	Nf dmd	Demanded fan speed	rpm	~
19	PCNfR dmd	Demanded corrected fan speed	rpm	~
20	W31	HPT coolant bleed	lbm/s	↓
21	W32	LPT coolant bleed	lbm/s	↓

A total of 21 sensors were used. Among them, 14 were related to the potential degradation mechanism during the entire degradation process; these sensors are numbered 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21. In the stage of data preprocessing, to avoid any interference of useless information, we select the information of these 14 sensors as target data. Most of the equipment can be divided into normal stage and rapid degradation stage in its life cycle. For the purpose of RUL prediction, the prediction of RUL in the stage of rapid degradation is more important than the that under normal stage. According to the work of Babu [2], when 125 cycles remain, a clear degradation trend appears. The degradation failure threshold is set to 125 cycles, as shown in Figure 8.

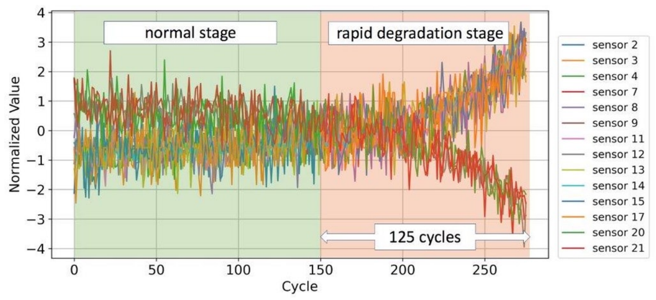


Fig. 8. Degradation failure threshold

Since the rapid degradation trend under normal stage is not obvious, the data intercepted before entering the rapid degradation stage is used as the training data, and the length of the data is set to 160 cycles.

To prevent the increase of network training difficulty caused by different sensor numerical scales, we need the z-score normalization for all the training data. The formula is shown as follows:

$$x_i' = \frac{x_i - u_i}{\sigma_i} \tag{23}$$

where u_i is the mean value and σ_i is the corresponding standard deviation.

4.2. Data Generated

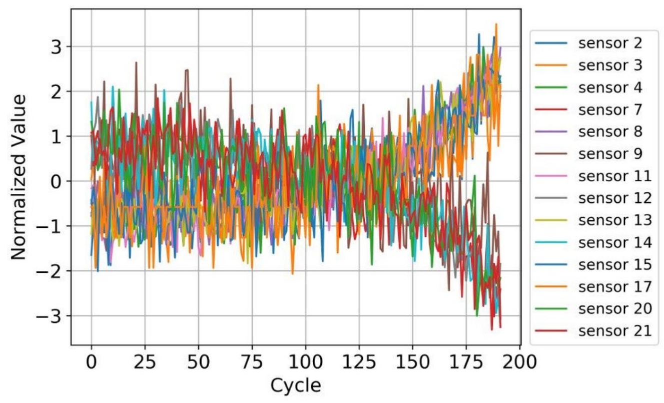
The preprocessed data are sent to the amplification network as input. Different from the regression task, the customized generator is a single-layer LSTM network that prevents the output of the network from becoming highly abstract and affecting the expression of the details of the original degradation data.

The number of parameters in LSTM has a positive correlation with the complexity of the model. In this experiment, the number of parameters in LSTM is set to 160. To keep the input and output dimensions of the network consistent, a dense operation is conducted at the output of the network. For the discriminator, features with degraded information need to be extracted extensively, so a stacked three-layer LSTM network is applied, and the number of parameters in LSTM is set to 100.

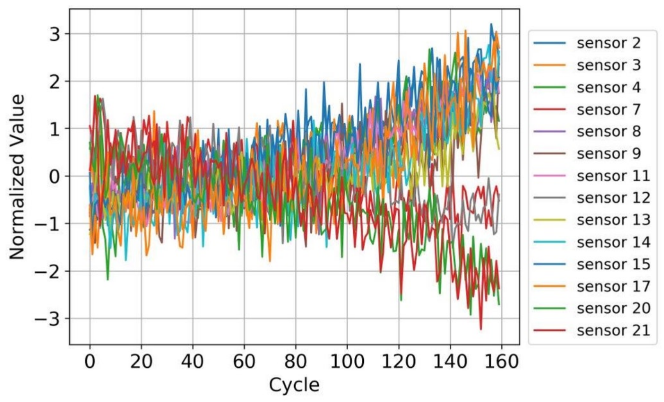
The trained generator from CycleGAN is used to amplify the training data. The data selected in this experiment are all from FD001. We divide 100 data into three groups of 7: 2: 1 as training set, validation set, and test set. In the training set, we use different numbers of data (10, 30, 50, and 70) to train the generator and explore the effect of various amounts of degradation data on the experimental results. The generators are constructed from different amounts of training data to generate FD001 Unit 1. The obtained data are shown in Figure 9. For further explanation, we number the data shown in Figure 9.

As shown in Figure 9, $1^{\#}$ indicates the original data, and $2^{\#}$, $3^{\#}$, $4^{\#}$, and $5^{\#}$ are the degradation data from the generator trained from the original data with different numbers of scales. In the case of the Generator built from less training data, such as Figure 9(b), the model can still learn the approximate distribution of samples. We compared the MMD differences between them, and the results are shown in Table III. Although these samples look similar from an intuitive point of view, they are not simply copied.

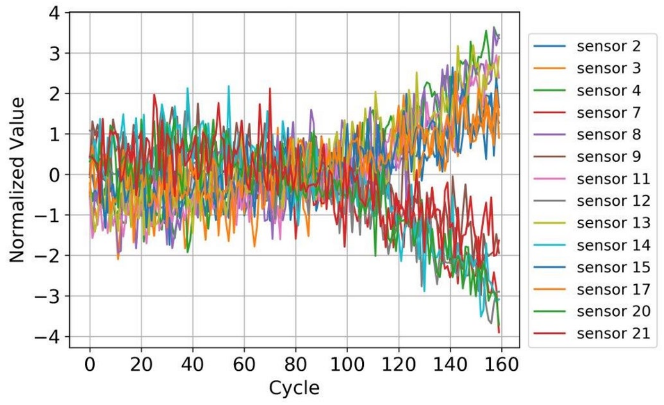
Furthermore, to find out the difference in the overall distribution of the generated degradation data, we compared the MMD between



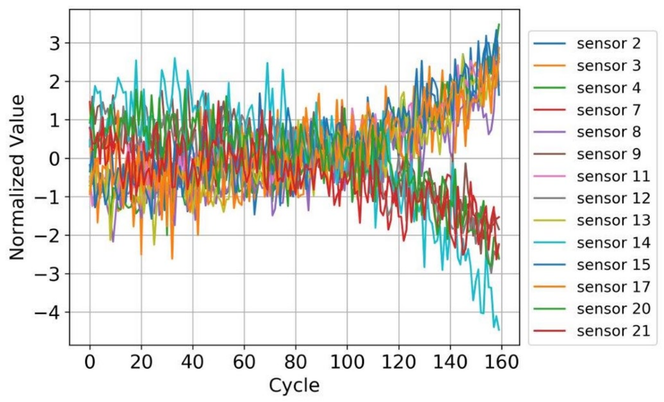
(a)1#: FD001 Unit 1 for real



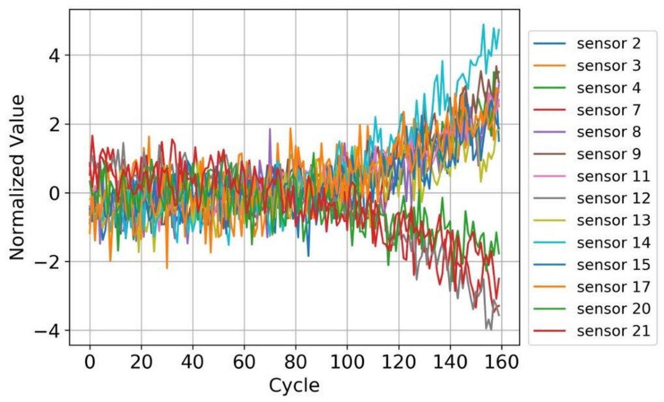
(b)2#: FD001 Unit 1 generated by Generator trained with 10 samples



(c) 3# : FD001 Unit 1 generated by Generator trained with 30 samples



(d) 4#: FD001 Unit 1 generated by Generator trained with 50 samples



(e) 5#: FD001 Unit 1 generated by Generator trained with 70 samples

Fig. 9. FD001 Unit 1 generated from generator trained with different numbers of samples

a single generated sample in Figure 9 and all original training data shown in Figure 10.

Table III. MMD between real FD001 Unit 1 and generated FD001 Unit 1

Data	1#	2#	3#	4#
MMD	0.194	0.148	0.143	0.113

As the amount of data participating in training increases, the MMD between the generated and target samples is gradually reduced, which means that the generated samples and overall real samples are getting closer in distribution. Simultaneously, the trend of data degradation generated by the generator is more obvious, because the network summarizes the distribution of overall training data and provides the most common distribution. The generated degradation data are close to the real data in distribution. As the amount of data increases, the differ-

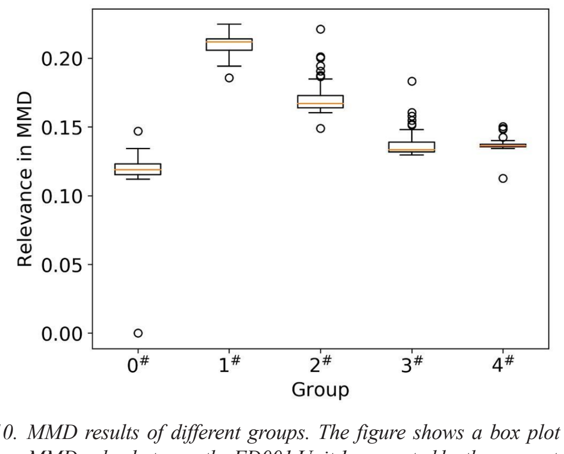


Fig. 10. MMD results of different groups. The figure shows a box plot of the MMD value between the FD001 Unit 1 generated by the generator constructed from different training samples and entire training samples

ence between the generated data and real data narrows, which is also in accordance with expectations.

C. RUL Prediction

To test whether the generated data can be used as training data to build a prediction network and explore the effect of our proposed model on training data of different sizes, we add the amplified data to the training data, and establish some prediction networks. To meet the requirements of controlled experiments, we built several sets of prediction models using real and generated data. The details are presented in the Table IV.

Table IV. Prediction network build with different data

Group	General Method	Proposed Method
A#	10 real samples	10 real + 10 generated samples
B#	30 real samples	30 real + 30 generated samples
C#	50 real samples	50 real + 50 generated samples
D#	70 real samples	70 real + 70 generated samples

To accurately measure the prediction effect of the model, we present the evaluation method of RUL for the multi-sensor turbo aero engine as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{24}$$

where $d_i = RUL_i^{'} - RUL_i$ indicates the prediction error of the *i*-th instance, $RUL_i^{'}$ and RUL_i respectively represent the predicted RUL from model and the actual RUL from dataset of the *i*-th instance. A score function is given as:

$$s = \sum_{i=1}^{N} s_i$$
$$s = \begin{cases} e^{-\frac{d_i}{13}} - 1, & \text{for } d_i < 0 \\ e^{\frac{d_i}{10}} - 1, & \text{for } d_i \geq 0 \end{cases} \tag{25}$$

The score function from the dataset provider has a more practical significance. The penalty with smaller prediction deviations is small, but that for a larger prediction deviation is larger. The difference is shown in Figure 11.

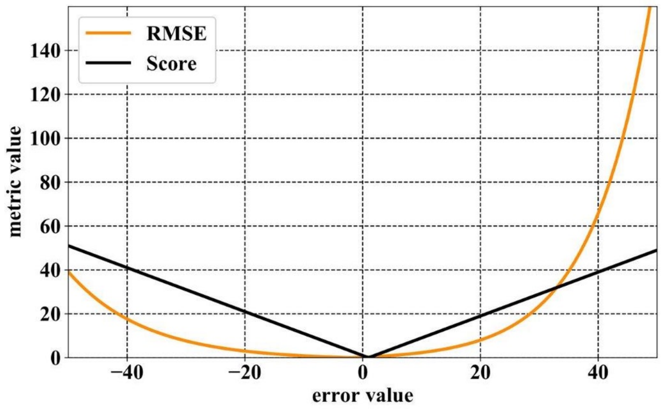


Fig. 11. Evaluation of RMSE and score function

The training data are grouped as $A^{\#}$, $B^{\#}$, $C^{\#}$, $D^{\#}$, and used to construct the RUL prediction model. To reduce the influence of the error on the experimental effect, each set of data builds a prediction model 10 times, and verifies the data on the test set. The average of the results is considered as the final results. The RMSE and scores are shown in Tables V and VI, and the results are plotted into the histogram in Figures 12 and 13.

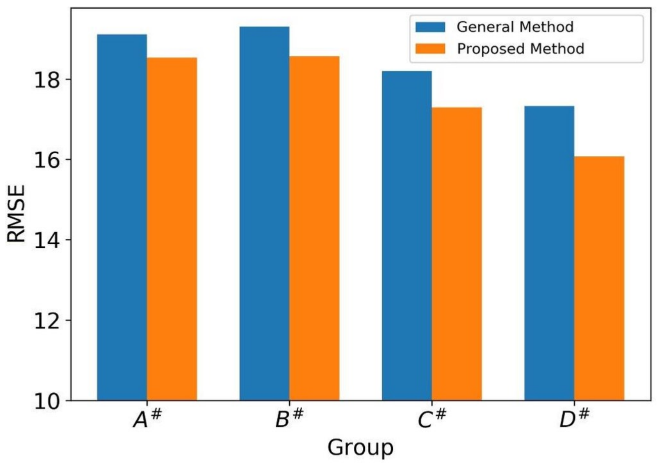


Fig. 12. RMSEs of the predicted RUL with the general method and proposed method

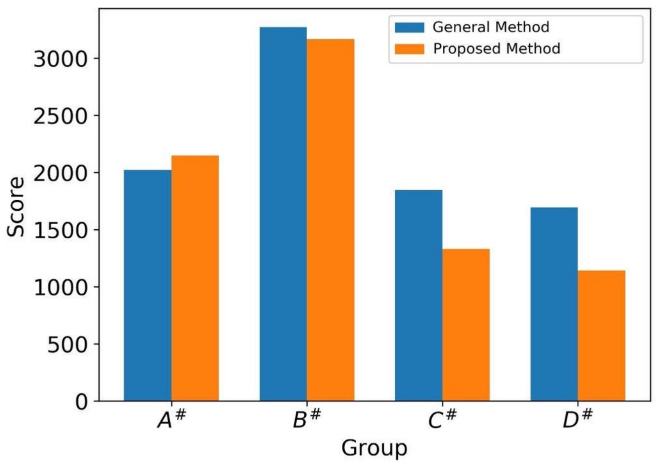


Fig. 13. Scores of the predicted RUL with the general method and proposed method

As shown in the Figure 12 and 13, the general method is reflecting on the blue histogram which is the result of a model built using the real data, and the proposed method is reflect on the orange histogram which is the result of a model built using not only the real data but also the generated data from the real data, what needs to be reminded is that both methods use the same predictive model, but the data used to build the model is different. When the MSE function is used to evaluate the test results, our proposed method achieves leading experimental results in all four groups of experiments. However, for the score function, the effect of groups $A^{\#}$ and $B^{\#}$ did not show obvious advantages, and the score of group $B^{\#}$ is higher than that of group $A^{\#}$. In our analysis, the difference between the individual and test samples in the middle part of group $B^{\#}$ is extremely large, and the score function is closer to the actual situation, resulting in the poor performance of the model built under the extremely small training data scale. When the real data increases, especially in the $C^{\#}$ and $D^{\#}$ groups, the proposed method performs better than the RUL prediction.

Intuitively, the RUL prediction in test set FD001 Unit 95 is shown in Figure 14. Sub-figures (a) to (d) indicate four results predicted by model constructed with real data. Sub-figures (e) to (h) indicate four

Table V. MSE of RUL results

Group	1	2	3	4	5	6	7	8	9	10	Average
General Method / A#	19.16	18.99	19.10	19.24	18.83	19.34	19.11	19.25	19.01	19.18	19.11
Proposed Method / A#	18.36	18.45	18.87	18.75	18.45	18.23	18.49	18.55	18.75	18.42	18.53
General Method / B#	19.92	19.64	19.03	19.87	18.88	19.08	19.20	18.95	19.21	19.22	19.30
Proposed Method / B#	18.67	19.20	18.54	17.05	18.74	19.03	18.26	19.09	19.08	18.06	18.57
General Method / C#	18.75	17.92	18.20	17.83	18.19	18.10	17.92	18.56	18.29	18.23	18.20
Proposed Method / C#	17.33	16.96	17.09	18.11	17.08	17.19	16.40	17.41	17.91	17.42	17.29
General Method / D#	16.77	17.34	18.47	17.10	17.32	18.02	17.09	16.26	18.09	16.86	17.33
Proposed Method / D#	17.05	16.33	15.66	15.11	16.44	15.58	16.24	16.07	15.57	16.60	16.07

Table VI. Score of RUL result

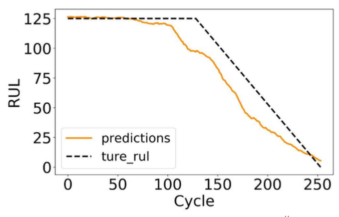
Group	1	2	3	4	5	6	7	8	9	10	Average
General Method / A#	2148	1839	2134	2040	2014	2178	2077	1919	1865	2012	2023
Proposed Method / A#	1996	2963	1999	2152	1829	1605	1851	2879	2251	1965	2149
General Method / B#	3616	3393	2906	3556	2832	3729	3105	3339	2821	3408	3270
Proposed Method / B#	2632	3669	2793	2835	2498	2475	2510	3417	5610	3243	3168
General Method / C#	2324	1573	1729	1523	1947	2077	1566	1812	2058	1835	1844
Proposed Method / C#	1246	1295	1149	1677	1034	1294	924	1509	1554	1605	1329
General Method / D#	961	1940	2143	1467	2328	2205	1034	1631	1711	1519	1694
Proposed Method / D#	1218	1164	1141	922	1215	1192	1098	1281	1053	1138	1142

results predicted by model constructed with mixed data. Each result is predicted by a model built with different amounts of training data. Various amounts of training data can also build prediction networks, but the prediction effect constructed by the mixed data composed of real data and generated data is better. By comparing the results of MSE, we find that the curve convergence is better than the model built from real data. On the other hand, the prediction model with more training data has better model prediction effect, especially at the end of the life cycle, where the accuracy of the prediction is improved.

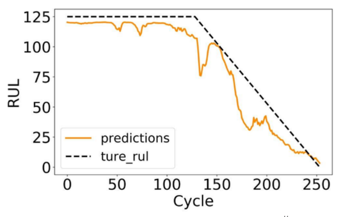
In the MSE evaluation, the proposed method can also improve the prediction accuracy. It is not obvious in the score evaluation of samples 10 and 30, but in samples 50 and 70, the proposed method has higher prediction scores.

4.3. Applicability analysis

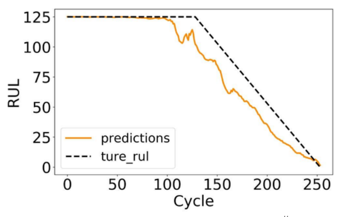
The method proposed in this study is suitable for devices with multiple sensors and degradation data presented in time series. On the premise of having a small number of run-to-failure degradation data, our proposed method shows good performance, when a small amount of data is obtained, the remaining useful life of the equipment can also be effectively predicted. In the case of having sufficient degradation data, the sample space of the degradation data is sufficiently complete, and the prediction model established on this basis already has good performance, our proposed method has limited improvement under such circumstances. In view of the fact that obtain large amount of degradation data in actual industrial production is still not ideal, our proposed method still has very important significance.



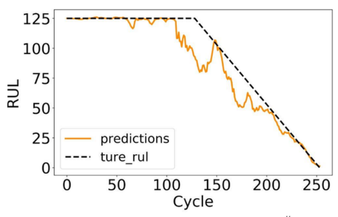
(a) General Method with $A^{\#}$



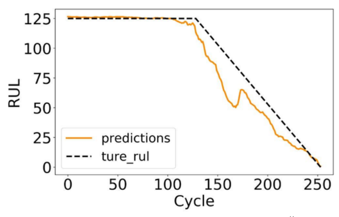
(b) General Method with $B^{\#}$



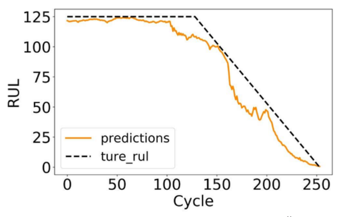
(c) General Method with $C^{\#}$



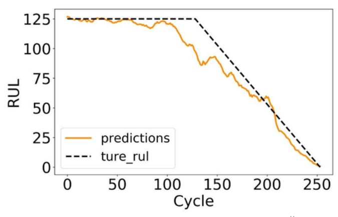
(d) General Method with $D^{\#}$



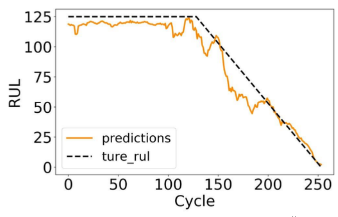
(e) Proposed Method with $A^{\#}$



(f) Proposed Method with $B^{\#}$



(g) Proposed Method with $C^{\#}$



(h) Proposed Method with $D^{\#}$

Fig. 14. RUL prediction results in the test set Unit 95

5. Conclusion

In this study, a framework for predicting the RUL with insufficient data was proposed, in which two main parts are involved. First, based on the characteristics of the sequence degradation data, an amplification network was designed using CycleGAN. Second, sliding time window strategy and deep BiLSTM network are jointly employed to construct the RUL prediction model based on the amplified degradation data. The following conclusions can be obtained: 1) Generating an adversarial network, as an unsupervised deep learning network, can indeed learn relevant information about data distribution. 2) The improved generated network based on LSTM can generate data with distribution similar to that of real data, and the RUL prediction network constructed using these amplified data has proved to be effective. 3) In the case where the RUL prediction accuracy is generally limited by the size of the training data, our proposed method provides a new reference for the development of RUL prediction.

Some possible topics for future research include the follows.

(1) In many applications, the test set and training set may come from different test conditions, under which the equipment workloads, environmental condition and noise levels may vary. That may lead to different distribution of training set and test set. It would be interesting to improve the domain adaptability of our RUL prediction framework.

(2) Due to the variability of raw materials quantity and manufacturing accuracy, it is common to see that the degradation characteristics of individuals may show unit-to-unit variability. How to improve prediction accuracy considering individual characteristics deserves further investigation.

Acknowledgements

This work was supported by the Major Special Projects of Zhongshan 200824103628344 and 2019A4018.

References

1. Abdulraheem A, Abdullah Arshah R, Qin H. Evaluating the Effect of Dataset Size on Predictive Model Using Supervised Learning Technique. International Journal of Software Engineering & Computer Sciences (IJSECS) 2015; 1: 75–84, https://doi.org/10.15282/ijsecs.1.2015.6.0006.
2. Babu G S, Zhao P, Li X-L. Deep convolutional neural network based regression approach for estimation of remaining useful life. International conference on database systems for advanced applications, Springer: 2016: 214–228, https://doi.org/10.1007/978-3-319-32025-0_14.
3. Deutsch J, He D. Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2017; 48(1): 11–20, https://doi.org/10.1109/TSMC.2017.2697842.
4. Graves A, Mohamed A, Hinton G. Speech Recognition with Deep Recurrent Neural Networks. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2013. doi:10.1109/ICASSP.2013.6638947, https://doi.org/10.1109/ICASSP.2013.6638947.
5. Guo L, Li N, Jia F et al. A recurrent neural network based health indicator for remaining useful life prediction of bearings. Neurocomputing 2017; 240: 98–109, https://doi.org/10.1016/j.neucom.2017.02.045.
6. KARABACAK E Yunus, GÜRSEL ÖZMEN N, GÜMÜŞEL L. Worm gear condition monitoring and fault detection from thermal images via deep learning method. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2020; 22(3): 544–556, http://dx.doi.org/10.17531/ein.2020.3.18.
7. Khan S, Yairi T. A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing 2018; 107: 241–265, https://doi.org/10.1016/j.ymssp.2017.11.024.
8. Le Son K, Fouladirad M, Barros A et al. Remaining useful life estimation based on stochastic deterioration models: A comparative study. Reliability Engineering & System Safety 2013; 112: 165–175, https://doi.org/10.1016/j.ress.2012.11.022.
9. Li D, Chen D, Jin B et al. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In Tetko IV, Kůrková V, Karpov P, Theis F (eds): Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series, Cham, Springer International Publishing: 2019: 703–716, https://doi.org/10.1007/978-3-030-30490-4_56.
10. Li J, Li X, He D. A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. IEEE Access 2019; 7: 75464–75475, https://doi.org/10.1109/ACCESS.2019.2919566.
11. Li X, Ding Q, Sun J. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety 2018; 172: 1–11, https://doi.org/10.1016/j.ress.2017.11.021.
12. Li Y, Wang K. Modified convolutional neural network with global average pooling for intelligent fault diagnosis of industrial gearbox. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2020; 22(1): 63–72, http://dx.doi.org/10.17531/ein.2020.1.8.
13. Lyu Y, Gao J, Chen C et al. Optimal Burn-in Strategy for High Reliable Products Using Convolutional Neural Network. IEEE Access 2019; 7: 178511–178521, https://doi.org/10.1109/ACCESS.2019.2958570.
14. Lyu Y, Gao J, Chen C et al. Joint model for residual life estimation based on Long-Short Term Memory network. Neurocomputing 2020; 410: 284–294, https://doi.org/10.1016/j.neucom.2020.06.052.
15. Mao W, He J, Zuo M J. Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. IEEE Transactions on Instrumentation and Measurement 2019. doi:10.1109/TIM.2019.2917735, https://doi.org/10.1109/TIM.2019.2917735.
16. Nieto P G, Garcia-Gonzalo E, Lasheras F S, de Cos Juez F J. Hybrid PSO–SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. Reliability Engineering & System Safety 2015; 138: 219–231, https://doi.org/10.1016/j.ress.2015.02.001.
17. Peng K, Jiao R, Dong J, Pi Y. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. Neurocomputing 2019; 361: 19–28, https://doi.org/10.1016/j.neucom.2019.07.075.
18. Ragab A, Ouali M-S, Yacout S, Osman H. Remaining useful life prediction using prognostic methodology based on logical analysis of data and Kaplan–Meier estimation. Journal of Intelligent Manufacturing 2016; 27(5): 943–958, https://doi.org/10.1007/s10845-014-0926-3.
19. Ren L, Sun Y, Wang H, Zhang L. Prediction of bearing remaining useful life with deep convolution neural network. IEEE Access 2018; 6: 13041–13049, https://doi.org/10.1109/ACCESS.2018.2804930.
20. Sagheer A, Kotb M. Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing 2019; 323: 203–213, https://doi.org/10.1016/j.neucom.2018.09.082.
21. Si X-S, Wang W, Hu C-H et al. A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. Mechanical Systems and Signal Processing 2013; 35(1–2): 219–237, https://doi.org/10.1016/j.ymssp.2012.08.016.

22. Sohani A, Sayyaadi H, Hoseinpoori S. Modeling and multi-objective optimization of an M-cycle cross-flow indirect evaporative cooler using the GMDH type neural network. International Journal of Refrigeration 2016; 69: 186–204, https://doi.org/10.1016/j.ijrefrig.2016.05.011.
23. Su C, Chen H, Wen Z. Prediction of remaining useful life for lithium-ion battery with multiple health indicators. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2021; 23(1): 176–183, http://dx.doi.org/10.17531/ein.2021.1.18.
24. Su C, Chen H, Wen Z. Prediction of remaining useful life for lithium-ion battery with multiple health indicators. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2021; 23(1): 176–183, http://dx.doi.org/10.17531/ein.2021.1.18.
25. Wang B, Lei Y, Li N, Li N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. IEEE Transactions on Reliability 2018. doi:10.1109/TR.2018.2882682, https://doi.org/10.1109/TR.2018.2882682.
26. Wen B, Xiao X, Wang X et al. Data-driven remaining useful life prediction based on domain adaptation. PeerJ Computer Science 7:e690 2021. doi:https://doi.org/10.7717/peerj-cs.690, https://doi.org/10.7717/peerj-cs.690.
27. Xie Y, Zhang T. A transfer learning strategy for rotation machinery fault diagnosis based on cycle-consistent generative adversarial networks. 2018 Chinese Automation Congress (CAC), IEEE: 2018: 1309–1313, https://doi.org/10.1109/CAC.2018.8623346.
28. Yin S, Ding S X, Xie X, Luo H. A review on basic data-driven approaches for industrial process monitoring. IEEE Transactions on Industrial Electronics 2014; 61(11): 6418–6428, https://doi.org/10.1109/TIE.2014.2301773.
29. Yinka-Banjo C, Ugot O-A. A review of generative adversarial networks and its application in cybersecurity. Artificial Intelligence Review 2020; 53(3): 1721–1736, https://doi.org/10.1007/s10462-019-09717-4.
30. Yoon J, Drumright L N, van der Schaar M. Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN). IEEE Journal of Biomedical and Health Informatics 2020; 24(8): 2378–2388, https://doi.org/10.1109/JBHI.2020.2980262.
31. Zhai Q, Ye Z-S. RUL prediction of deteriorating products using an adaptive Wiener process model. IEEE Transactions on Industrial Informatics 2017; 13(6): 2911–2921, https://doi.org/10.1109/TII.2017.2684821.
32. Zhang X, Xiao P, Yang Y et al. Remaining Useful Life Estimation Using CNN-XGB with Extended Time Window. IEEE Access 2019; PP: 1–1, https://doi.org/10.1109/ACCESS.2019.2942991.
33. Zhang Y, Xiong R, He H, Pecht M G. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Transactions on Vehicular Technology 2018; 67(7): 5695–5705, https://doi.org/10.1109/TVT.2018.2805189.
34. Zheng G, Sun W, Zhang H et al. Tool wear condition monitoring in milling process based on data fusion enhanced long short-term memory network under different cutting conditions. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2021; 23(4): 612–618, https://doi.org/10.17531/ein.2021.4.3.
35. Zhu J, Chen N, Peng W. Estimation of bearing remaining useful life based on multiscale convolutional neural network. IEEE Transactions on Industrial Electronics 2018; 66(4): 3208–3216, https://doi.org/10.1109/TIE.2018.2844856.
36. Zhu K, Liu T. Online tool wear monitoring via hidden semi-Markov model with dependent durations. IEEE Transactions on Industrial Informatics 2017; 14(1): 69–78, https://doi.org/10.1109/TII.2017.2723943.
37. Zio E, Di Maio F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. Reliability Engineering & System Safety 2010; 95(1): 49–57, https://doi.org/10.1016/j.ress.2009.08.001.