Research Article

Bearing Life Prediction Method Based on Parallel Multichannel Recurrent Convolutional Neural Network

Jianmin Zhou ^{1,2}, Sen Gao ^{1,2}, Jiahui Li,^{1,2} and Wenhao Xiong^{1,2}

¹Key Laboratory of Conveyance and Equipment of Ministry of Education, East China Jiaotong University, Nanchang 330013, China

²State Key Laboratory of Performance Monitoring and Protecting of Rail Transit Infrastructure, Nanchang 330013, China

Correspondence should be addressed to Jianmin Zhou; hotzjm@163.com

Received 20 July 2021; Accepted 24 September 2021; Published 13 October 2021

Academic Editor: Changqing Shen

Copyright © 2021 Jianmin Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To extract the time-series characteristics of the original bearing signals and predict the remaining useful life (RUL) more effectively, a parallel multichannel recurrent convolutional neural network (PMCRNN) is proposed for the prediction of RUL. Firstly, the time domain, frequency domain, and time-frequency domain features are extracted from the original signal. Then, the PMCRNN model is constructed. The front of the model is the parallel multichannel convolution unit to learn and integrate the global and local features from the time-series data. The back of the model is the recurrent convolution layer to model the temporal dependence relationship under different degradation features. Normalized life values are used as labels to train the prediction model. Finally, the RUL was predicted by the trained neural network. The proposed method is verified by full life tests of bearing. The comparison with the existing prognostics approaches of convolutional neural network (CNN) and the recurrent convolutional neural network (RCNN) models proves that the proposed method (PMCRNN) is effective and superior in improving the accuracy of RUL prediction.

1. Introduction

Prediction and health management is an effective method to improve the safety, integrity, and task success of the system under the actual operating conditions. Remaining life prediction is the most challenging technology in prediction and health management [1]. In modern production machinery, rolling bearings are the key parts to determine the health of machinery. Thus, it is significant to carry out real-time monitoring of the health status of rolling bearings in operation and prediction of Remaining Useful Life (RUL) [2, 3]. Among various machine learning algorithms, the deep learning network model can automatically learn multilayered features from the original data due to its strong scalability and representation learning ability. This machine learning algorithm is one of the hot spots and key points in the current research algorithms of equipment life prediction [2, 4, 5].

In recent years, many scholars have done a lot of research on the method of predicting the RUL of rolling bearings and

put forward a variety of network models. Correlative algorithms based on convolutional neural networks (CNNs) have better ability in dealing with time-series signals. CNN has been widely used in RUL prediction [6]. Mao et al. [7], first, used Hilbert–Huang to extract the edge spectrum and health status label of the original vibration signal of the bearing, using CNN to obtain the deep fault characteristics of bearings. These features were inputted to the long-short time memory (LSTM) network for training and obtained an effective prediction model. Cheng et al. [8] used the Hilbert–Huang to train bearings' raw vibration and employed CNN to identify the hidden pattern. Jiang et al. [9] used CNN to extract degradation features in parallel channels and used LSTM to be excellent for processing time-series features. Ren et al. [10] proposed a new feature extraction to obtain eigenvector, which was adapted for deep CNN. Similarly, Zhu et al. [11] used a new CNN method to predict the bearing RUL through time-series features and MSCNN. The method could keep global and local degradation features.

Although the above CNN model and the CNN-RNN integrated model have a good effect on RUL prediction of rotating machinery equipment, in the CNN model, there are two limitations in RUL prediction:

- (1) CNN mainly achieves feature extraction of adjacent areas through local connection and weight-sharing, by adjusting the size of the convolution kernels in CNN to extract the deep features in different time channels. Generally, global features require a large convolution kernel and local features require a small convolution kernel. How to take into account both global and local features and make full use of the system fault information contained in the monitoring data is the key to accurately predict the RUL.
- (2) CNN does not consider the temporal dependencies between different degradation states. During the period of running, the development of faults is an evolution process; also, the bearing will gradually degrade from normal operating states to complete failures over time [12]. Accordingly, the degradation states of rolling bearings corresponding to different time periods are related in the time channel. Therefore, to accurately predict the RUL, it is significant to establish correlation models for different degradation states on time-series sensor data. However, current CNN-based models ignore these dependencies during model simulation, which affects the veracity and limits their generalizations of the prediction model.

To overcome the above shortcomings and accurately predict the RUL of rolling bearings, this paper comes up with a new deep learning prediction framework. The framework is called parallel multichannel recurrent convolutional neural network (PMCRNN). In PMCRNN, a parallel multichannel convolution learning strategy is first proposed to automatically learn and integrate the characteristics of different time channels. Then, in different degenerated states, this paper constructs the recurrent convolutional neural network to model the temporal dependencies. Finally, the regression analysis on the learned high-level features is carried out and the RUL is predicted. By experimentally testing the data of rolling bearing life cycle under different situations, experimental results demonstrate the effectiveness in improving RUL prediction accuracy.

2. Architecture of the PMCRNN Model

The PMCRNN prediction model proposed in this paper is structured by parallel multichannel convolution building blocks, pooling layers, recurrent convolution layers (RCL), and a fully connected layer. In PMCRNN, parallel multichannel learning strategy is used to learn parallel multichannel features. In different degenerated states, this paper introduces the learned parallel multichannel features into RCL to model the temporal dependence. Finally, the pooling layer and the fully connected layer are used to perform the RUL estimation.

2.1. Parallel Multichannel Convolution Building Blocks. In convolutional networks, the selection of convolution kernel sizes is the key. In different time channels, different convolution kernel sizes are used to extract different information [13]. Specifically, global features require a big size convolution kernel, whereas local features require a small convolution kernel. For the rolling bearing monitoring data, as time goes by, monitoring data included more and more degradation features. This paper develops a parallel multichannel learning strategy to avoid such issues. Each multiscale learning strategy includes 3 different convolutional channels. Parallel arrangement of these convolutional channels and each channels utilize 3 different convolution kernel sizes, i.e., $F \times 1$, $2F \times 1$, and $4F \times 1$. During representation learning, 3 convolutional channels have no interaction and each channel independently extracts the degradation information from different time-series scales. Thus, the integrity of features is guaranteed.

The structure of the parallel multichannel convolution building blocks is shown in Figure 1. First, pass the original signal Z_1 at the time step 1 through convolution channel 1 and extract a period of time-series features F_1 , pass the original signal Z_2 at the time step 2 through convolution channel 2 and extract a period of time-series features F_2 , and so on, until the time-series features F_t at the time step m are extracted. Finally, the extracted time-series features $F_1 \sim F_t$ are spliced into a time-series feature with time dimension in a chronological order. This feature is used as the model input of the next stage.

2.2. Recurrent Convolution Layers. The core building block of CNN is the convolutional layer, which can independently extract degrade features from rolling bearing original data. Only the recurrent layer can feed back the output to the input. This means that features can only be transmitted within CNN. Accordingly, CNN only considers the current features at each degradation point but ignores the previous degradation features. Particularly, the existing rolling bearing RUL forecasting methods based on CNN cannot solve these issues. The prediction accuracy and generalization ability are reduced. Thus, a new recurrent convolutional layer is constructed in this paper. The recurrent layer output can be fed back to its input so that the information can be memorized over time, which improves the prediction performance of the network.

The output of the RCL depends on both the current input features and the all past information. This enables the RCL to model the temporal dependencies of different degradation states and to make good use of the degradation features which are extracted from rolling bearing data. In form, in i th RCL, the output x_t^i at the time series t can be expressed as

$$x_t^i = f(x_t^{i-1}, h_{t-1}^i), \quad (1)$$

where x_t^{i-1} is the input time-series features, $f(\bullet)$ is the nonlinear activation function, and h_{t-1}^i is the storage state of feedback of recurrent connection at the time step $t - 1$.

In actual application, the problems of disappearance or explosion gradients are often encountered in the training

FIGURE 1: Parallel multichannel convolution building blocks.

process of recurrent convolutional layers. In order to mitigate the effects of disappearance or explosion gradients and consider temporal dependencies, the recurrent convolutional layer introduces a gating mechanism [14].

As shown in Figure 2, the gated recurrent convolutional layer has two gates, i.e., update gate u_t^i and reset gate r_t^i , and they can be written as

$$\begin{aligned} r_t^i &= \sigma(K_r^i * x_t^{i-1} + W_r^i * h_{t-1}^i + b_r^i), \\ u_t^i &= \sigma(K_u^i * x_t^{i-1} + W_u^i * h_{t-1}^i + b_u^i), \end{aligned} \quad (2)$$

where $*$ denotes the convolution operator, $\sigma(\bullet)$ is the logistic sigmoid function, K_r^i , W_r^i , K_u^i , and W_u^i are the convolution kernels, and b_r^i and b_u^i are the bias terms. The state x_t^i of the gated RCL at each time step t can be written as

$$x_t^i = u_t^i \circ h_{t-1}^i + (1 - u_t^i) \circ \tilde{h}_t^i, \quad (3)$$

$$\tilde{h}_t^i = \tanh(K_h^i * x_t^{i-1} + W_h^i * (r_t^i \circ h_{t-1}^i) + b_h^i), \quad (4)$$

where $\circ$ denotes the Hadamard product, $\tanh(\bullet)$ is the tanh activation function, b_h^i is the bias term, $\tilde{h}_t^i$ is the newly generated state, and K_h^i and W_h^i are the convolution kernels. According to equations (3) and (4), the state x_t^i can be linearly combined of the current candidate state $\tilde{h}_t^i$ and the previous state h_{t-1}^i at the time step t and is controlled by the update and reset gates.

2.3. Pooling Layer and Fully Connected Layer. The proposed PMCRCNN also employs pooling layers and fully connected layers. Pooling layer (PL) is used to extract main degradation features and further reduce the features dimension. In PMCRCNN, each RCL is followed by a PL, and each of the input feature map is pooled independently. At the time step t , the state y_t^{ij} for the j th feature map in the i th pooling layer can be obtained by

$$y_t^{ij} = \text{pool}(x_t^{ij}, p, s), \quad (5)$$

where $\text{pool}(\bullet)$ is the downsampling function, p is the polling size, and s is the stride size.

The fully connected layers are used to regression analysis and form a fully connected network structure. The fully connected layer as the output layer is placed at the end of

PMCRCNN to estimate the RUL. At the time step t , the output of fully connected layer z_t^{i-1} at the i th can be written as

$$z_t^i = f(z_t^{i-1}) = f(W_f^i z_t^{i-1} + b_f^i), \quad (6)$$

where z_t^{i-1} is the fully connected layer input at the i th, i.e., b_f^i is the bias vector, and W_f^i is the weight matrix and the previous layer output at the $i-1$ th.

2.4. The Overall Layout of PMCRCNN. The proposed PMCRCNN model architecture is shown in Figure 3, which is structured by parallel multichannel convolution building blocks (denoted as Block), pooling layer (PL), recurrent convolution layer (RCL), and fully connected layer (FCL). In PMCRCNN, the degradation information from bearing vibration signals is integrated, using parallel multichannel time-series degradation features of size $H \times 1 \times C$ as the inputs to the predictive network, where C is the number of features and H is the length of each time-series sequence. Then, N pooling layers and N recurrent convolutional layers are used to automatically learn multiple features of representations from input time-series data. Then, the temporal dependencies of different degradation states are modeled. The number of kernels in the recurrent convolutional layer is set to be $2^{i-1}M$, where $i = 1, 2, \dots, N$ and the kernel size is $k \times 1$. For the first $N-1$ polling layers, a nonoverlapping window to perform pooling operation is used and the max pooling is used as the downsampling function. Then, a global max pooling layer is utilized to conduct the downsampling. From the N th recurrent convolutional layer, the max pooling layer correspondingly transforms the high-level representations into a vector of size $2^{i-1}M$. Finally, this vector is input into L FCLs to predict the bearing RUL. The proposed PMCRCNN, in this paper, have three fully connected layers. There are F neurons in each of the first two FCLs. To implement nonlinear activation, each of layers adopt rectified linear unit (ReLU). The last FCL is used as the output layer of PMCRCNN to predict bearings RUL, which has only one neuron.

2.5. Loss Function. For the issue of bearing RUL prediction, the predicted life value of the PMCRCNN will be higher or lower than the true. When the predicted value is less than the

FIGURE 2: Recurrent convolution layer gate control mechanism.

FIGURE 3: Architecture of parallel multichannel recurrent convolution network.

true, the relevant parts can be replaced or repaired in time. Compared with the delay prediction, predicting bearing degradations in advance can effectively reduce the loss and risks. The improved mean square error (MSE) proposed in this paper is the loss function:

$$\text{MSE}(y, y^*) = \frac{\sum_{i=1}^n \alpha (y_i - y_i^*)^2}{n}, \quad (7)$$

where y_i^* is the predicted life value of the i th data, y_i is the true life value of the i th data, and α is the correction factor. When $y_i - y_i^*$ is positive, α is 1. When $y_i - y_i^*$ is negative, α is 2.

3. Method and Step

This paper proposed a method named PMRCNN to predict the rolling bearing RUL. Figure 4 shows the flowchart of this method.

The specific steps are as follows:

Construct the degradation feature parameters' set: the features are extracted from the life-span vibration data of bearing, i.e., the time domain, frequency domain, and time-frequency domain feature. Time

domain and frequency domain features include root mean square, kurtosis, peak value, and skewness and are normalized. This paper selects the energy ratio of the frequency subband generated by the three-layer wavelet packet decomposition of the vibration signal, which is regarded as the time-frequency domain feature.

Define the RUL label: the start moment of degradation point to complete failure is taken as the rolling bearing RUL. Normalizing the RUL time label to (0, 1) and using the label to train PMRCNN network,

$$y = \frac{i - k}{n - k - 1}, \quad (8)$$

where i is the current time value, k is the start moment of degradation, and n is the bearing life value.

Train PMRCNN network: determine the network training parameters. The selected bearing degeneration features' set is taken as input, and the network using label is trained which is the RUL normalization time.

Predict the RUL: use the trained PMRCNN model to predict the test bearing RUL.

FIGURE 4: Flowchart of the proposed method.

4. Experimental Setup and Data Processing

4.1. Experimental Setup. To evaluate the performance of the PMCRNN-based method, the bearing life evaluation model is validated and analyzed with a rolling element bearing dataset [3]. The data were acquired from XJTU-ST bearing dataset [3]. According to the relevant description in [15–17], for the study of bearing degradation, the horizontal vibration signal provided more useful information than the vertical vibration signal. Therefore, this paper uses horizontal vibration signals for experiments. The experimental platform is shown in Figure 5.

The experiment started under a fixed speed condition. Accelerated degradation tests of rolling element bearings were carried out under different operating conditions. When maximum amplitude of the signals in horizontal or vertical exceeded $10 \times A_h$, the bearing is deemed to have failed and the relevant life test is terminated, where A_h is the maximum amplitude of vibration signals in one vertical under normal operating condition. During the experiment, tested bearings' failure of any kind may occur, i.e., outer-race fracture, outer-race faults, inner-race faults, and rolling parts faults. The type of tested rolling bearings is LDK UER204. The acceleration signal is collected in a continuous window. The sampling frequency is set to 25.6 kHz, recording each sampling in a total of 32768 data points (i.e., 1.28 s), and the period of the sampling is equal to 1 min. Data representing 3 different loads are considered (see Table 1).

The model in this paper uses the first set of data in condition 1 as the training set, i.e., the bearing 1_1 full life data; there are 123 sets of all samples. Bearings 1_2, 1_3, and 1_5 are test sets, respectively.

4.2. Degradation Feature Parameter Set Extraction. Extraction the frequency domain, time domain, and time-frequency domain feature parameters from the rolling bearing life-span tests. The features of frequency domain and time domain, respectively, include 11 classes, which include root mean square, kurtosis, peak value, and skewness. The bearing signal is decomposed by db5 wavelet packet in three layers, selecting the energy ratio of the frequency subband as the time-frequency domain feature. The extracted frequency domain, time domain, and time-frequency domain features are used as the degradation feature parameter set and training the PMCRNN model with set.

Take the rolling bearing 1_1 as an example. Figure 6 shows the full life vibration signal. At the 73th time point, compared with the normal standard amplitude, the rolling bearing vibration amplitude is significantly increased. This time point is the starting moment of bearing degradation and starts predicting the bearing RUL. According to its standards, the starting moments of four bearings are calculated and shown in Table 2. For each experimental sample (X_i, Y_i) , where X_i is the feature input, i.e., the vibration acceleration value collected for the time step i th, and Y_i is the label, i.e., the time ratio of the difference between the current time and the degradation time to the time interval between the start moment of degradation and the degradation time (normalized to (0, 1)).

5. RUL Prediction for Bearings

For traditional regression problems, in this paper, the number of nodes in the last output layer of the network model is 1, and its output value is the predicted value of the model. To verify the accuracy of the RUL prediction method,

FIGURE 5: Bearing test rig.

TABLE 1: Bearings' operation condition.

Cutting condition	Spindle speed (rpm)	Radial force (kN)
Condition 1	2100	12
Condition 2	2250	11
Condition 3	2400	10

FIGURE 6: The life-span vibration signal of bearing 1_1.

this paper uses two evaluation indicators as follows: (1) root mean square error (RMSE) and (2) mean absolute error (MAE). Their values can be computed by

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2}, \quad (9)$$

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |A_t - F_t|, \quad (10)$$

where A_t is the true value for the time step t and F_t is the predictive value.

In RUL prediction, PMCRNN hyperparameters are adjusted and selected by performing cross validation in the bearing original datasets, including the kernel size $1 \times k$, the number of RCLs, the number of convolution kernels M , the

number of neurons F , and the pooling size p . Simultaneously, to avoid overfitting, this model applies L2 and dropout regularization to each FCL and RCL. In addition, the PMCRNN model uses the RMSE as the evaluation loss function and Adam as the optimizer and optimizes network weights through iterative update. The data are arranged as depicted in Table 3.

The training set label value of the PMCRNN model is normalized to (0, 1), and the sigmoid function is used as the activation function of the final output layer. Then, bearing 1_1 is used as the training model for the training set. Bearings 1_2, 1_3, and 1_5 test and verify the effectiveness of the network. CNN and RNN are the chief component of PMCRNN. In order to verify effectiveness, the model is compared with the CNN model, the PMCCNN model, and the RCNN model. The test results of bearing 1_2 are shown in Figure 7.

TABLE 2: Bearing degradation points.

Number of bearing	The starting moment of degradation/1 min
Num.1_1 bearing	73
Num.1_2 bearing	37
Num.1_3 bearing	58
Num.1_4 bearing	119
Num.1_5 bearing	33

TABLE 3: Training parameters of the PMCRCNN model.

Hyperparameter	Value
Number of convolution, M	16
Number of kernels, N	4
Number of neurons, F	200
Kernel size, $1 \times k$	1×8
Pooling size, p	8
Weight decay coefficient, λ	10^{-5}
Dropout probability, π	0.15
Number of epochs	300

FIGURE 7: The prediction result of the proposed method.

In Figure 7, the x -label represents the bearing original data sets and the y -label represents the life percentage of the bearing tested at current moment. The black solid line is the actual life values and the four solid lines in different colors represent the predicted life values, respectively. Figure 8 shows that the RUL prediction method of the PMCRCNN model which is used to test the bearing 1_2, and the obtained RUL prediction value is closest to the real life value, the two curves are in good coincidence degree, and the prediction effect is better. As the bearing degradation gradually deepens, the fluctuation of the predicted values gradually decreases, and the coincidence degree between the label of the predicted value and the true value increases. However, when using CNN, PMCCNN, and RCNN, the coincidence degree between the predicted value and the true value are lower than the PMCRCNN model. The prediction value of CNN neural network fluctuates greatly and the prediction accuracy is low.

From equations (9) and (10), RMSE and MAE of four different prediction methods are calculated. As shown in Table 4, it can be seen that compared with other three models, PMCRCNN can provide more accurate RUL prediction results with stable prediction effect and close to the real RUL. The comparison results show that the PMCRCNN model can accurately predict the bearing RUL. Because the RCNN model has a single extraction feature in the time channels, the PMCCNN model does not consider the long-term dependencies of time-series features. In the PMCRCNN model proposed in this paper, the parallel multichannel convolution learning strategy has the ability to extract features on different time channels and the recurrent convolution has the ability to excavate continuous time-series features so that the extracted time-series features can effectively reduce the prediction error. Figure 8 shows that the PMCRCNN model effectively enhances the prediction performance of the convolutional network.

FIGURE 8: Performance estimate results of four different prognostics network. (a) RMSE values. (b) MAE values.

TABLE 4: Prediction error of different networks.

Test bearing	The evaluation index	CNN	PMCCNN	RCNN	PMCRCNN
Bearing 1_2	RMSE	0.098	0.082	0.078	0.052
	MAE	0.068	0.059	0.054	0.045
Bearing 1_3	RMSE	0.271	0.142	0.245	0.094
	MAE	0.191	0.173	0.187	0.106
Bearing 1_5	RMSE	0.105	0.099	0.112	0.077
	MAE	0.060	0.052	0.062	0.043

6. Conclusion

The accuracy of rolling bearing RUL prediction largely depends on the extraction of time-series features. The paper first extracts the feature parameters of the original signal of rolling bearing, i.e., frequency domain, time domain, and time-frequency domain. Then, the rolling bearing RUL is predicted by the PMCRCNN model, and the results demonstrated the effectiveness of this method. It has the following advantages.

PMCRCNN processes time-series features by stacking multiple convolutional blocks of different channels and obtaining a more complete time-series features by automatically learning and integrating the global and local features of bearing signals. Particularly, each fully connected layer and recurrent convolutional layer apply L2 and dropout, which effectively integrate parallel multichannel degradation features to prevent overfitting.

To model the temporal dependence relationship under different degradation features, the model constructs a recurrent convolutional layer. The max pooling layer is used to reduce features' dimension so that the extracted features are more compact and the high-level features of the original data can be obtained.

To better observe the track of rolling bearing RUL predicted value and improve the accuracy of predictions, the model determines the degenerate point. The training label is

normalized so that each bearing has the same variation interval and failure threshold. For prediction results, the influence of the uncertainty of failure threshold determination is avoided.

Data Availability

The data (XJTU-ST bearing data set) used to support the findings of this study have been deposited in the repository (A hybrid prognostics approach for estimating remaining useful life of rolling element bearings) (DOI: 10.1109/TR.2018.2882682).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (Grant no. 51865010). The authors also acknowledge Dr. Changqing Shen who made valuable suggestions to increase the technical quality of the paper.

References

- [1] D. El-Thalji and E. Jantunen, "A summary of fault modeling and predictive health monitoring of rolling element bearings," *Mechanical System PMC and Signal Processing*, vol. 1, pp. 60-61, 2015.
- [2] J. A. Carino, D. Zurita, M. Delgado, J. A. Redondo, and R. Romero-Troncoso, "Remaining useful life estimation of ball bearings by means of monotonic score calibration," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 1752-1758, Seville, Spain, March 2015.
- [3] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 99, pp. 1-69, 2020.

- [4] H. Dong, N. Hou, Z. Wang, and H. Liu, "Finite-horizon fault estimation under imperfect measurements and stochastic communication protocol: dealing with finite-time boundedness," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 1, pp. 117–134, 2019.
- [5] X. Li, H. Dong, Z. Wang, and F. Han, "Set-membership filtering for state-saturated systems with mixed time-delays under weighted try-once-discard protocol," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 2, pp. 312–316, 2019.
- [6] L. Guo, Y. Lei, N. Li, T. Yan, and N. Li, "Machinery health indicator construction based on convolutional neural networks considering trend burr," *Neurocomputing*, vol. 292, pp. 142–150, 2018.
- [7] W. Mao, J. He, J. Tang, and Y. Li, "Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network," *Advances in Mechanical Engineering*, vol. 10, no. 12, pp. 1–18, 2018.
- [8] C. Cheng, G. Ma, Y. Zhang et al., "A deep Learning-based remaining useful life prediction approach for bearings," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1243–1254, 2020.
- [9] J. Jiang, J. Lee, and Y. Zeng, "Time series multiple channel convolutional neural network with attention-based long short-term memory for prediction bearing remaining useful life," *Sensors*, vol. 20, no. 1, p. 166, 2020.
- [10] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of bearing remaining useful life with deep convolution neural network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018.
- [11] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 3208–3216, 2018.
- [12] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.
- [13] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June 2015.
- [14] K. Cho, B. V. Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, <https://arxiv.org/abs/1406.1078>.
- [15] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on Hilbert-huang transform, support vector machine, and regression," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 52–62, 2015.
- [16] R. K. Singleton, E. G. Strangas, and S. Aviyente, "Extended Kalman filtering for remaining-useful-life estimation of bearings," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1781–1790, 2014.
- [17] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, no. 3, pp. 98–109, 2017.