Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture

Boyuan Yang, Ruonan Liu, *Member, IEEE*, and Enrico Zio, *Senior Member, IEEE*

***Abstract*—Remaining useful life (RUL) prediction has been increasingly considered in many industrial fields for the reliability and safety of their systems. As a data analysis tool of deep learning, deep convolutional neural network (CNN) shows great potential for RUL prediction. This paper proposes an intelligent RUL prediction method based on a double-CNN model architecture. Given the powerful feature extraction capability of CNN, the proposed method is fed with original vibration signals with no need to resort to any feature extractor, which can also retain the useful information in maximum. The prediction includes two stages: first, incipient fault point is identified by the first CNN model and a proposed "3/5" principle; then, the second CNN model is constructed for RUL prediction. In practice, RULs of identical components are different from each other, which poses a major challenge in RUL prediction. To overcome this problem, an intermediate reliability variable is first calculated in this paper, instead of directly predicting the RUL value. Then, a mapping algorithm is proposed to map reliability to RUL. To demonstrate the effectiveness of the proposed method, data of four tests of bearing degradation are utilized for RUL prediction. Compared with state-of-the-art methods, the proposed method shows higher prediction accuracy and robustness. The prediction results and evaluation indexes demonstrated the effectiveness and superiority of the proposed method.**

***Index Terms*—Convolutional neural network (CNN), deep learning, prognostic and health management (PHM), remaining useful life (RUL) prediction.**

I. Introduction

As AN effective way to avoid unnecessary maintenance activities and improve safety reliability and availability, prognostics and health management (PHM) has gained a lot of attention in the last few decades, with remaining useful life (RUL) prediction constituting a challenging prognostic task [1]. A reliable and accurate estimation of RUL not only allows for effective predictive maintenance, thus protecting the system from faults and resource wasting, but also can avoid catastrophic failures and casualties [2], [3]. Therefore, it is important to develop RUL prediction methods [4].

According to the literature, there are different ways to categorize the RUL prediction methods. In general, they can be classified as follows: physics-based, data-driven, and hybrid [5]. Physics-based approaches, also known as model-based, take advantage of prior knowledge on the degradation and failure mechanisms to model the degradation and failure behavior of an equipment mathematically [6], e.g., Paris–Erdogan model [7] and exponential model [8] for crack propagation with particle filtering used as parameter estimator [9]. However, often in practice, it is difficult, even impossible, to obtain enough prior knowledge for physics-based methods. In this case, data-based approach is proposed as an alternative. It relies on historical run-to-failure data for estimating the RUL via different machine learning methods [10], [11]. Most of these methods consist of two stages: an offline learning with feature extractor and degradation state learner; an online stage for RUL estimation via the learned model. Support vector regression (SVR) [12] and artificial neural network (ANN) [13] are two widely used regression methods. The last method is hybrid, which combines the advantages of the previous two methods: physics knowledge is used to build the model, and the parameters are optimized by data-driven methods for accurate RUL estimation.

Although prognostics methods have been widely studied, they still suffer from some disadvantages. First, most methods are applied of feature level. A feature extractor can be beneficial, because it can transform the original signal into low-dimensional vectors for easier match or comparison [14]–[16]. However, on the one hand, it excludes a lot of information during the feature extraction process, while some information can be useful. On the other hand, it also makes the prognostic process time-consuming, because it is often designed entirely handcrafted and rather specific for every new task. Therefore, although data-driven methods reduce the dependence on prior knowledge, the results are still highly dependent on experience [17]. Second, most of the prognostic methods do not take into account the differences between degradation patterns, as well as the failure threshold (FT), which could influence the prediction

Manuscript received August 9, 2018; revised November 30, 2018, March 27, 2019, and May 20, 2019; accepted June 3, 2019. Date of publication July 1, 2019; date of current version July 31, 2019. *(Corresponding author: Ruonan Liu.)*

B. Yang is with the School of Electrical and Electronic Engineering, University of Manchester, Manchester M13 9PL, U.K. (e-mail: yangboyuanxjtu@163.com).

R. Liu is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: liuruonan04@163.com).

E. Zio is with the Centre for Research on Risk and Crisis, Mines ParisTech, PSL Research University, F-06904 Sophia Antipolis, France, and also with the Department of Energy, Politecnico di Milano, 20156 Milano, Italy (e-mail: enrico.zio@ecp.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2019.2924605

performance greatly and vary from many internal or external factors, such as the change of operation condition, mission profile, or processing technique. Finally, most methods in the literature only consider the current prediction but ignore the prediction of previous time, which can also be valuable. As a result, the theoretical lifetime of a mechanical equipment can be influenced greatly by accidental error and noise, and therefore be different from the actual RUL [18].

Deep learning is a burgeoning new area of artificial intelligence that can learn a hierarchy of features by building high-level features from low-level ones automatically and accurately [19]. As a representative deep learning model, convolutional neural network (CNN) has achieved tremendous success in image recognition and speech tasks, because of its superior feature extraction and object recognition performances, and is also attracting attention in the industrial field [20]. For instance, a CNN based on LeNet-5 is proposed for fault diagnosis, which converts signals into two-dimensional (2-D) images and then extracts features from these images to eliminate the effect of handcrafted features [21]. An NB-CNN method is proposed based on CNN and a Naive Bayes data fusion scheme to aggregate the information extracted from each video frame and enhance the overall performance of the system [22]. A deep normalized CNN is proposed in [23] for imbalanced fault classification of machinery. However, although the application of CNN has been developed well, there are few research works about the RUL prediction, which is always a difficult problem.

In this paper, a double-deep CNN framework-based intelligent RUL prediction method is proposed. The proposed method makes four contributions.

First, the proposed method is a signal-level method, which can deal with raw signals directly, instead of relying on the feature extractor. The reduction of the demands for prior knowledge, physical model, and human labor makes the proposed method more intelligent and adaptive.

Second, the proposed double-deep learning framework consists of two deep CNN architectures. Different degradation patterns are considered in the proposed framework. Then, the incipient failure (IF) point can be calculated in the first network. In this way, the RUL can be estimated in the second CNN layer through its corresponding degradation model from the IF point.

Third, due to changing operational conditions and strong background noise, the prediction results can be influenced by abnormal points. In the proposed method, predictions of earlier times are also taken into account by a weighted algorithm to mitigate the effects of abnormal points and ensure the stability of the algorithm performance.

Finally, the RUL of different components, even the same component of different batches, can be different from different manufacturing conditions. Therefore, in instead of directly calculating the RUL using deep learning method, the operational reliability is first computed and, then, mapped to the RUL.

The rest of this paper is organized as follows. In Section II, the proposed intelligent RUL prediction method based on deep multi-CNN is described in detail. Then, the model is evaluated by accelerated degradation tests of bearing in Section III. In Section IV, the results are discussed, as well as the comparison with state-of-the-art RUL prediction methods. Section V concludes this paper.

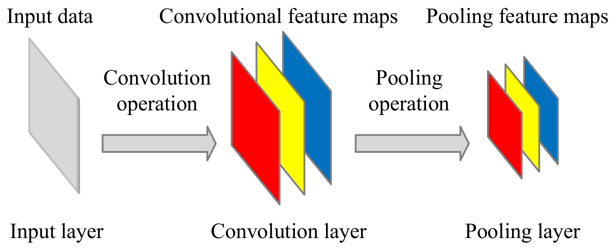


Fig. 1. An illustration of a CNN consisting of a pair of a convolutional layer and a pooling layer in succession.

II. Proposed Framework

Based on the traditional CNN, a deep double-CNN framework is proposed in this paper for RUL prediction. Two CNN architectures are integrated in the proposed framework: the first CNN for incipient failure threshold (IFT) identification, and the last one for RUL prediction.

A. Convolutional Neural Network

CNN is a special feed-forward neural network that can extract topological properties from the inputs [24]. Different from the traditional feed-forward ANN, it simulates three architectural properties of the visual cortex cell to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and subsampling [25]. As shown in Fig. 1, there are two basic modules in CNN, including convolution operation, which implements the first two properties of CNN, and pooling for subsampling [26].

Convolution is an operation on two functions. For example, suppose I is a 2-D image input, and K is a 2-D kernel. The convolution z of I and K can be calculated as

$$z[i,j] = \sum_m \sum_n I[i+m, i+n]K[m,n] = (I * K)[i,j]. \quad (1)$$

The convolution kernel, also called the weight, slips across the input image and performs the convolution operation with each corresponding local receptive field. In this way, the network greatly reduces the unnecessary parameters because all local parts can share the same weights and, therefore, avoids the overfitting problem.

Pooling is another important operation in CNN, through which the output of the net at a certain location is replaced with a summary statistic of the nearby outputs. It is performed mainly to reduce the calculation cost, characterize the translation invariance, cut down the input dimension, and control the overfitting risk. Invariance to local translation can be very useful because we care more about whether some feature is present than exactly where it is [27]: considering an industrial signal as an example, whether the mechanical component is faulty is much more important than finding out which part of the signal shows the failure information.

The most commonly used pooling operation is max-pooling, which outputs the maximum number within a rectangular neighborhood, called the pooling size. In this paper, the max-pooling operation is used in the deep CNN architecture.

Apart from local receptive fields and shared weights, dropout is also applied in the proposed method to reduce the risk of overfitting problem. Dropout refers to dropping out units (hidden and visible) in a neural network. That is, randomly let some neurons to be zero. The primary idea of dropout is to randomly drop components of neural network (outputs) from a layer of neural network. By dropping a unit out (that is, temporarily removing it from the network), along with all its incoming and outgoing connections, the corresponding unit is retained with a fixed probability independent of other units. The thinned outputs are then used as input to the next layer. The resulting neural network is used without dropout.

B. CNN Classification Model for IF Point Identification

The identification of the IF point is the first task of RUL prediction because signals offered before the IF point provide little information on the degradation process. Finding the IF point is difficult because the same machines in different conditions have a large variation in the signals [28]. If the IF point is determined earlier, the system is likely to give many false alarms; if it is too late, then some useful signals and information can be lost and make the prediction less accurate.

The IF point is often determined experimentally due to the difference between machines at failure [29], [30]. However, experimentation can be time- and energy-consuming. Therefore, the proposed framework addresses the problem by automatically extracting the inherent differences between normal and failure signals, thus reducing the need for prior knowledge, data, and experimentation.

This is done through the use of CNN, which is a type of deep learning model in which trainable filters and local neighborhood pooling operations are applied on raw inputs alternately, resulting in a hierarchy of increasing complex features [20].

Therefore, CNN is used to identify the IF point in this step. The degradation patterns are divided into two types: rapid degradation and slow degradation. Two CNN models are trained as follows:

1) The vibration signals of mechanical components life tests under these two patterns are collected and saved as the training data.
2) For each kind of vibration data, the signals in normal stage are labeled as 0; and those in faulty stage are labeled as 1.
3) These signals are used to train the IFT identification CNN models to distinguish the faulty and normal signals.

During the RUL prediction process of an unknown test, the vibration signals are imported into the CNN classification models successively. When the output of the CNN model becomes 1 from 0, IF point is detected. However, industrial systems are not ideal model, and they can be always influenced by strong noise or fickle operational conditions. Even when the system is normal, the output of the CNN classification model can be 1 due to the influence of accidental error. Therefore, inspired by [31], a "3/5" principle is applied in this step: if for three times the model output is one in five sequential times, then incipient failure is detected. In this way, the IF point detection can be more stationary. The flowchart of this step is shown in Fig. 2. The IF point here can be regarded as the alarm, which can remind the operator ahead of time and perform fail-safe reactions, such as shutting down the failed components and making repair/replacement.

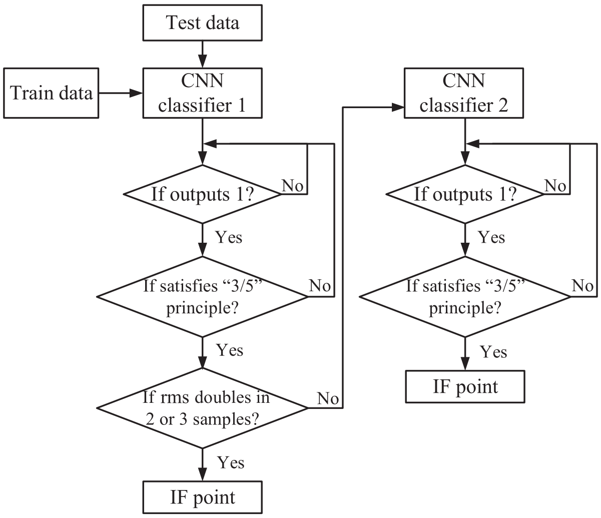


Fig. 2. Procedure for IF point identification. The label of normal sample is set to be 0; and the label of faulty sample is set to be 1.

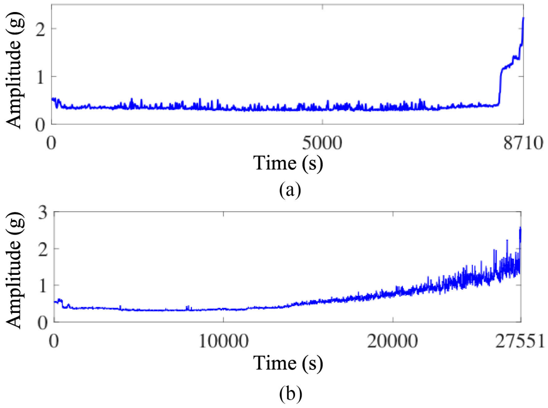


Fig. 3. RMS of two degradation patterns. (a) Sudden degradation. (b) Slow degradation.

C. Degradation Pattern Recognition

Degradation patterns on even identical complements vary due to different operational conditions, materials, manufacturing, or processing techniques [4]. Most of them follow two types of increasing trend of degradation patterns: for the first, in Fig. 3(a), the root-mean-square (rms) of vibration signal shows a sudden increasing trend; for the second, Fig. 3(b) shows a gradually increasing trend.

Because the first degradation pattern can lead to totally failure of industrial system in a short time, it should be considered first. Therefore, once the tested component begins to degrade, which can be determined by CNN classification model in the previous step, each degradation process is set to be the sudden degradation pattern at the beginning. Based on the physical manifestation of the sudden degradation pattern, if rms doubles after a few samples (usually two or three samples), this process is regarded as the first degradation pattern; otherwise, the alarm all-cleared, and this degradation process is regarded as the second pattern.

D. CNN Regression Model Between Original Signals and the Reliability Percentage

The RUL of some mechanical equipment can be quite different from each other. Therefore, instead of calculating the RUL directly, reliability is first calculated, and, then, the reliability is converted to the RUL. At the beginning, the equipment is normal and the output of the regression CNN (the second CNN) is one, and its reliability is taken equal to 100%. When the equipment is complete failure, its reliability is zero. Then, a mapping algorithm is proposed to map the reliability to the RUL. In this way, the problem of time variance can be solved.

The objective of this step is to fit the degradation process (usually the rms trend) with nonlinear functions. Exponential function is one of the most commonly used methods because typically the rms trend is a similar exponential form. However, exponential function cannot be fitted with all degradation process due to the variability of mechanical systems and operation conditions. Therefore, a CNN regression model is used to fit the nonlinear degradation process, which can calculate the reliability based on training data. This is because deep learning methods allow computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [26]. Therefore, CNN can fit more complex nonlinear functions than exponential functions theoretically. Two CNN models are trained in this step, one for regressing rapid degradation and the other one for slow degradation. Instead of using partial dataset to train the classification models in the second step, two regression CNN models are trained via all of the collected training data.

First, the collected vibration signals of the life test are imported into the CNN classification model as the training data to determine the IF point. Then, the labels corresponding to the training data are set to be linearly decreasing

$$y_2(i) = R(i) = \begin{cases} 1, & T_i \leq T_1 \\ 1 - \dfrac{T_i - T_1}{T_e - T_1}, & T_i > T_1 \end{cases} \tag{2}$$

where $y_2(i)$ is the label of the input at time t_i. $R(i)$ is the reliability percentage of the training data at time t_i. T_e is the total time of the test; T_1 is the start time of IF; and T_i is the run time for now. The graphical demonstration is shown in Fig. 4.

After importing the vibration signals, as well as their corresponding labels, the reliability prediction models under the two degradation patterns can be learned by training the CNN regression models.

E. Predicting the RUL of New Test Components

In this step, the trained CNN models are used to predict the RUL of a new test component. First, the vibration signals of the test component are collected. Then, based on the constructed framework, the collected signals are cut into a time series sample $X = x^{(1)}, x^{(2)}, ..., x^{(m)}$. At the beginning of the test, the degradation pattern is set to be the rapid degradation pattern, and the time series samples are imported into the CNN classification model of rapid degradation to identify the IF point. Based on the "3/5" principle, if three times the model outputs zero in five sequential times, then the experimental subject is judged as incipient failure. After a few samples (always two to five samples), if rms doubles, then the degradation pattern is confirmed to be the first one, and the reliability prediction can be continued. If rms does not double, this degradation is confirmed as the second pattern. Then, the historical time series samples are imported into the CNN classification model corresponding to the second degradation pattern to identify the IF point again. Just like the training data, the IF point of test data is also identified following Fig. 2. Once the IF point is confirmed, the collected time series samples are imported into the corresponding CNN regression model for reliability estimation.

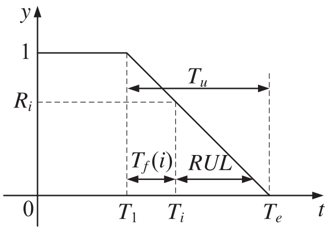


Fig. 4. Graphical illustration of reliability labels generation.

To solve the RUL time variability and obtain more accurate results, the labels of the CNN regression model is constructed as the reliability, and the results of the whole framework are the estimated reliability $R\hat{(}i)$. Then, the reliability is mapped to RUL by a mapping algorithm proposed in this paper.

As shown in Fig. 4, at moment t_i, let $T_f(i)$ be the time from IF point to T_i. So, $T_f(i)$ can be calculated as

$$T_f(i) = T_i - T_1. \tag{3}$$

Then, the estimated RUL from IF point to the final failure $T_u\hat{(}i)$ at t_i can be estimated by $T_f(i)$ and $R\hat{(}i)$ as in Fig. 4

$$T_u\hat{(}i) = \frac{T_f(i)}{1 - R(\hat{i})}. \tag{4}$$

To eliminate the influence of accidental errors, which is commonly seen in industrial systems, a weighted algorithm is proposed. That is, instead of using current signals to estimate RUL directly in traditional methods, the previous estimation results are also weighted into the proposed method, which can be mathematically expressed as

$$T_{wu}\hat{(}i) = \alpha_1 T_{wu}(\hat{i} - 1) + \alpha_2 T_u\hat{(}i) \tag{5}$$

where $T_{wu}\hat{(}i)$ is the weighted total RUL from IF point to the final failure at t_i; $T_{wu}(\hat{i} - 1)$ is the weighted RUL from IF point to the final failure at t_{i-1}. α_1 and α_2 are the weight factors of previous results and current results, and $\alpha_1 + \alpha_2 = 1$. If the system is easily to be influenced by noise or other accidental errors, the calculated results are more sensitive and more likely to occur

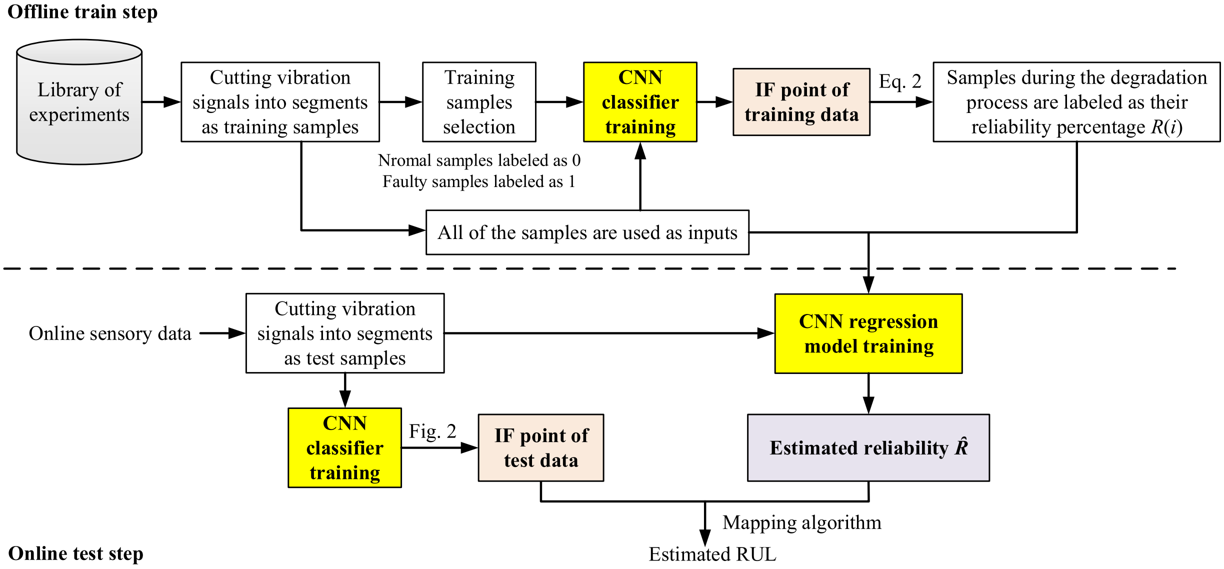

Offline train step
Library of experiments
Cutting vibration signals into segments as training samples
Training samples selection
CNN classifier training
IF point of training data
Eq. 2
Samples during the degradation process are labeled as their reliability percentage $R(i)$
Nromal samples labeled as 0
Faulty samples labeled as 1
All of the samples are used as inputs
Online sensory data
Cutting vibration signals into segments as test samples
CNN regression model training
CNN classifier training
Fig. 2
IF point of test data
Estimated reliability $\hat{R}$
Mapping algorithm
Estimated RUL
Online test step

Fig. 5. Proposed framework.

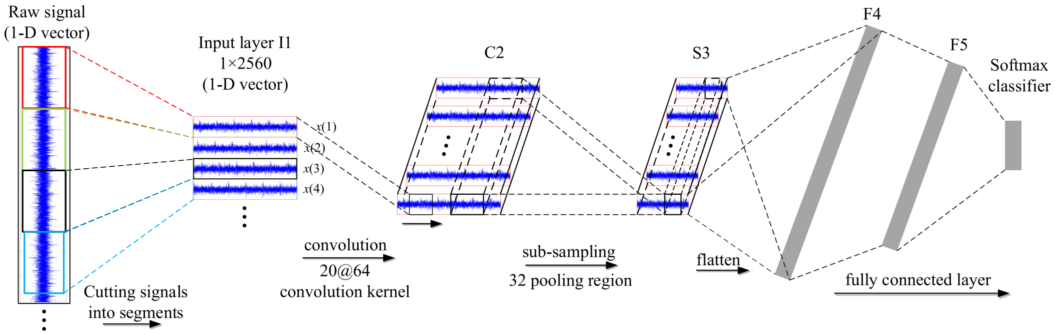


Fig. 6. The architecture of the signal-level CNN classifier, including an input layer I1, a convolutional layer C2 with 20×64 convolution kernel, a pooling layer F4 with 32 pooling size, a flatten layer F4, a fully connected layer F5, and a softmax classifier.

deviation, α_1 should set to be bigger. And if the test system is robust, the current result should take a larger proportion because it makes use of more timely information, and therefore α_2 should be bigger.

Based on $T_{w\hat{u}}(i)$ and $T_f(i)$, the estimated RUL is

$$\mathrm{RU\hat{L}}(i) = T_{w\hat{u}}(i) - T_f(i). \tag{6}$$

In this way, the proposed framework not only implements the mapping from reliability percentage to RUL, but also solves the RUL time variance problem, as well as the effect from background noise to the results.

The proposed framework for mechanical equipment RUL prediction is schematically represented in Fig. 5.

III. Case Study

A. Dataset Description

The experimental data come from the publicly available PRONOSTIA platform, which has been widely used to verify the effectiveness of RUL prediction methods [32]. During the experiments, a radial force of 4 kN is applied on the test bearings to conduct accelerated life tests. The rotating speed of bearings is set as 1800 r/min. Two vibration sensors with 25.6 kHz sampling frequency are mounted on the bearing to monitor the degradation process: one is set on the vertical axis, and the other one is on the horizontal axis. In this paper, the vertically collected signals are used for analysis. The length of every sample is 0.1 s with 2560 points, and the sampling is repeated every 10 s. All tests are stopped when the amplitude of the vibration signal exceeds 20 g. The experimental system, as well as the tested bearings before and after a test, can be found in [32].

B. Constructing CNN Classification Models for IF Point Identification

At the beginning of the analysis, a five-layer deep CNN model is first constructed as shown in Fig. 6, including an input layer (I1), a convolutional layer (C2), a subsampling layer (P3), a flatten layer (F4), a fully connected layer (F5), and a softmax classifier. In this experiment, the convolutional kernel in C2 layer is set to be 20 with 64 feature maps. Because the vibration signals are imported into the CNN model directly without any feature extraction, the convolutional kernel is a

TABLE I
PARAMETERS OF THE OPTIMIZATION ALGORITHM

Parameters	β_1	β_2	ϵ	learning rate	weight decay	epoch	batch size
Values	0.9	0.999	1×10^{-8}	0.001	0	50	32

one-dimensional vector instead of a 2-D matrix in traditional methods. Then, a pooling operation is applied in the following P3 subsampling layer. This leads to the same number of C2 feature maps with a smaller resolution. After flatting the outputs of P3 in F4 layer, the outputs are used as the input of F5 to dense the feature maps and reduce the dimension. Suppose the number of hidden units in F5 is a, then a $a \times 1$ vector is used as the input of softmax classifier and the final output is the number of categories. In this paper, there are 1024 hidden units in F5, so there are 1024 inputs imported into the softmax classifier. Dropout is added in S3 layer and F5 layer in Fig. 6. And dropout is set to be 0.25. The proposed method is implemented based on GPU: NVIDIA GeForce GTX 1060. For the first CNN model (classification CNN), the train time is 50.62 s, the test time is 0.32 s, and the GPU memory cost is 3416 Mb. For the second CNN model (CNN regression model), the train time is 57.96 s, the test time is 0.31 s, and the GPU memory cost is 3722 Mb. SVR is implemented based on CPU: Intel Core i7-6700 Processor, so there is no GPU memory cost. The train time is 3.69 s and the test time is 3.68 s. Python 3.5 platform is applied to implement the proposed method, and ReLU function is used as the activation function. Adam algorithm is used as the optimizer [33]. The parameters of the optimization algorithm are selected as in Table I. Two CNN classification models need to be trained to identify the IF point. In this experiment, the collected vibration signals are cut into segments with 2560 points (10 s), and each segment is regarded as a sample.

Two tests have been selected randomly for analysis. The vibration signals of the two tested bearings are shown in Fig. 7(a) and (b). It can be seen that the vibration amplitude of first bearing increases rapidly, while the second one increases slowly, which match with the two degradation patterns. Their vibration signals are cut into the normalized samples and used for learning CNN models in this paper.

For the first degradation pattern in Fig. 7(a), the rms of each sample is shown in Fig. 7(c). It can be seen that the vibration signal lasts 8710 s. And there are 871 samples, including about 828 normal samples, and about 43 faulty samples. The first 500 samples are selected as the normal training samples. And due to the rapid degradation trend, there are few faulty samples. For this test, all of the faulty samples labeled 1 are used for training. To control the input scale into a reasonable range, a "Min-Max scaling" normalization technique is applied, which can be mathematically represented as

$$x(i) = \frac{x_{\max} - x(i)}{x_{\max} - x_{\min}} \tag{7}$$

where $x(i)$ is the vibration amplitude at moment t_i. $x_{\max}$ and $x_{\min}$ are, respectively, the maximum and minimum values of all the samples.

In the proposed framework, the Rectified Linear Units (ReLU) function is used as the activation function for all CNNs, due to its fast convergence speed [34]

$$f(z) = \begin{cases} 0, & if \quad z < 0 \\ z, & \text{otherwise} \end{cases}. \tag{8}$$

The loss function is set to be the categorical cross-entropy.

For the second pattern, the waveform of its vibration signal and the rms are shown in Fig. 7(b) and (d). It can be seen that the vibration signal lasts 27 550 s, and there are 2755 signal segments. The bearing begins to degrade from 1415th sample and the degradation is slow. In this test, the first and the last 500 signal segments are used as normal and faulty samples to train the second CNN classification model, which is constructed the same as above.

After the CNN model is trained, all of the signal samples are imported into it to confirm the IF point. The classification results are shown in Fig. 8 in red lines. For comparison, another method is also used for IF point confirmation [35], whose results are marked with the green lines in Fig. 7(c) and (d). It can be seen that in Fig. 7(a), due to the rapid degradation, IF point can be easily confirmed at 828th sample (8280s) by both methods. However, the IF point can hardly be decided because the degradation process is slow and the indication of the bearing condition swings between normal and faulty in Fig. 7(b). This problem is solved by the proposed method because based on the "3/5" principle, the IF point is confirmed at 14 150 s. Compared with 17 330 s by the comparison method, this result is more reasonable.

C. Constructing CNN Regression Models

In this step, the CNN regression models are constructed for reliability estimation. For each test, the reliability can be calculated by (2) based on the IF point. Then, all of the samples after IF point are imported into the CNN regression model, as well as their corresponding reliability as their labels.

In this step, the CNN models are constructed almost the same as the CNN classification model. There are three differences between them.

1) Another fully connected network F6 with 32 hidden units is added between F5 and the last layer to avoid the dramatical shrink of the network and eliminate its influence on the analysis results.
2) Instead of using softmax classifier in the last layer, logistic regression is applied, and the outputs of logistic regression are used as the estimated reliability $\hat{R(i)}$.
3) Because the problem in this step is a regression problem, the mean squared error is used as the loss function in this model, instead of the categorical cross-entropy.

D. IF Point Identification of Test Data

After the CNN classification models and regression models have been trained, two random tests are selected to verify the effectiveness of the proposed method. The vibration signals of the two tests are shown in Fig. 9(a) and (b).

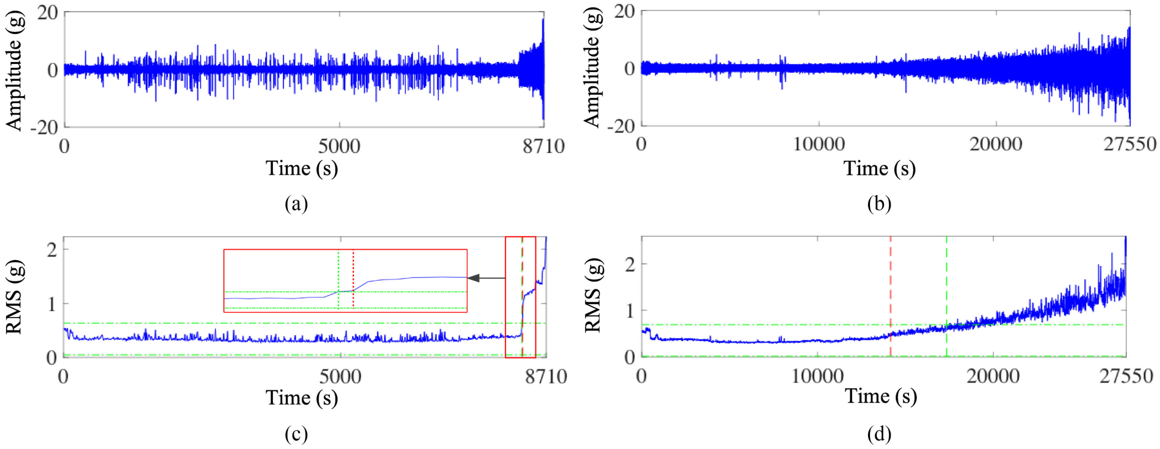


Fig. 7. Vibration signals of (a) the first train dataset and (b) the second train dataset. Waveforms of rms of (c) the first train dataset and (d) the second train dataset.

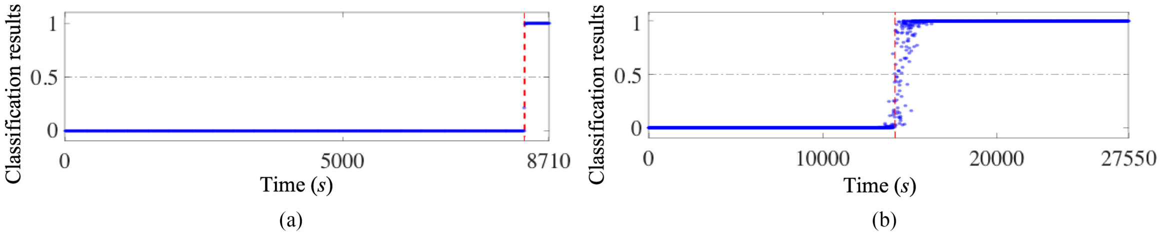


Fig. 8. IF point confirmation results of (a) the first test and (b) the second test.

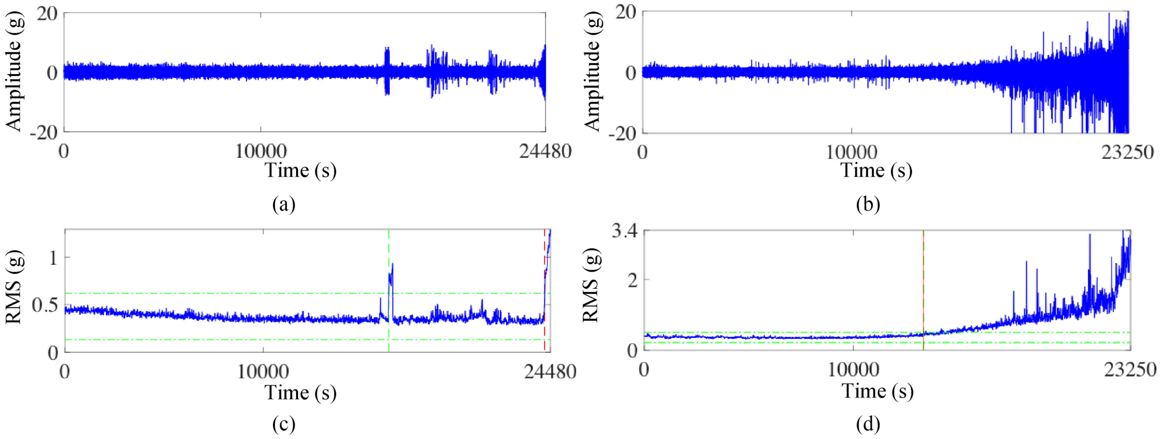
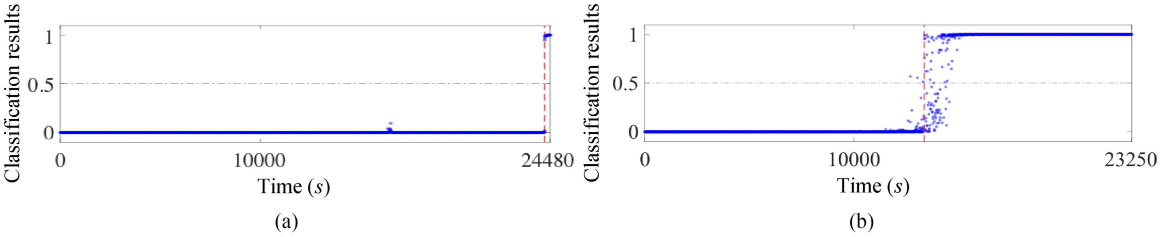


Fig. 9. Vibration signals of the two tests and their rms for verification of the proposed method: (a) vibration signal of the first test; (b) vibration signal of the second test; (c) rms of the first test; and (d) rms of the second test.

As described earlier in the proposed framework, the vibration signals are first cut into segments. Then, these segments are directly imported into the first CNN classification model for degradation pattern and IF point confirmation. The rms of the two tests are shown in Fig. 9(c) and (d). The IF points confirmation results of the first CNN model are shown in Fig. 10. It can be seen that the IF points of the two tests are determined as 24 190 and 13 350 s, respectively. The IF point of the second test is decided based on the “3/5” principle. As shown in Fig. 9(c), the rms of the first test doubles within two or three samples after its corresponding IF point, while the second test does not. Therefore, the first test is identified as a fast degradation pattern and the second test is identified as a slow degradation. Then, their signal segments are imported into the corresponding CNN classification models.

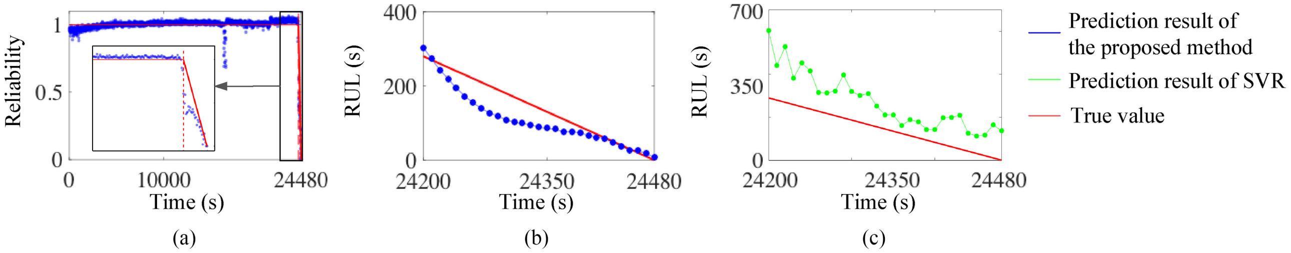
In addition, it can also be seen in Fig. 9(c) that there are some salient points during the experiment, which, however, do not mean that the bearing is faulty because the bearing still works for a long time after these points. If rms is considered as feature, these can hardly be distinguished from faulty samples by



(a)

(b)

Fig. 10. Classification results of (a) the first CNN model and (b) the second CNN model.

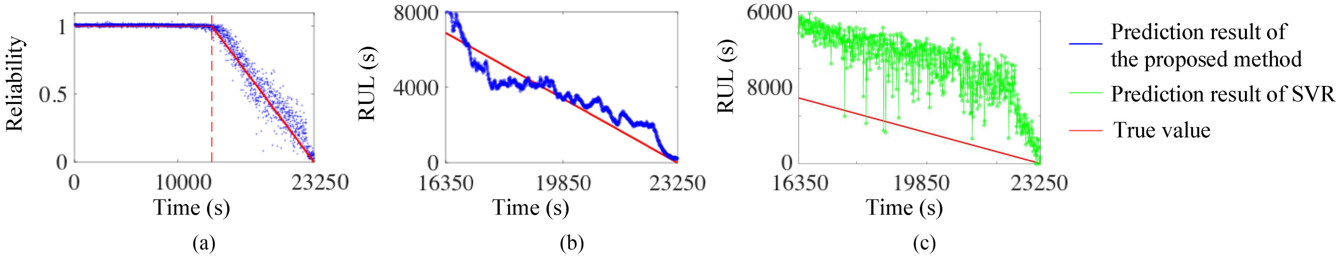


(a)

(b)

(c)

Fig. 11. RUL prediction results of the first verification test. (a) Estimated reliability percentage $\hat{R}$ by the proposed method. (b) RUL prediction results by the proposed method. (c) RUL prediction result by SVR.



(a)

(b)

(c)

Fig. 12. RUL prediction results of the second verification test. (a) Estimated reliability percentage $\hat{R}$ by the proposed method. (b) RUL prediction results by the proposed method. (c) RUL prediction result by SVR.

traditional methods. Based on the "3/5" principle, the proposed method solves the problem effectively.

E. RUL Prediction of Test Data

After the IF point of each test is confirmed, all the collected samples are imported into their corresponding CNN regression models. Based on the mapping algorithm proposed in this paper, the RUL can be estimated. To illustrate the superiority of the proposed method, SVR method is used to analyze the same signals for comparison. SVR is a regression method that can estimate a relationship between input and output random variables. It has been widely used in many research works for RUL prediction because it has shown to be very efficient in modeling nonlinear relationships between a target variable and a set of predictor variables [12], [36]. 11 time- and frequency-domain features, such as rms, mean, kurtosis of the samples after IF points of the training data, and the corresponding RUL are used as the training set of the SVR. The estimated reliability, RUL prediction results, and the comparison of the RUL prediction results of the two tests are shown in Figs. 11 and 12.

TABLE II
PREDICTION ERRORS OF THE PROPOSED METHOD AND SVR

		The proposed method (training)	The proposed method (testing)	SVR
First test	RMSE	2.13	4.30	10.28
	CRA	0.94	0.77	0.34
Second test	RMSE	37.74	72.03	727.55
	CRA	0.96	0.78	-1.51

The horizontal axis of Figs. 11(b) and (c) and 12(b) and (c) is the current time, and the vertical axis is the prediction of remaining time that can be operated, which is called the RUL. For a more intuitive comparison, root-mean-square error (RMSE) and cumulative relative accuracy (CRA) [37] are used as the prediction indexes, and shown in Table II. To illustrate the performance between raw data and the CNN model in training procedure, the RMSE and CRA of the two training datasets are also shown in Table II. It can be seen that the proposed method performs well both in training and testing procedures.

It can be seen from Figs. 11 and 12 that the prediction results of the proposed framework show high accuracy. Compared with more than three errors of SVR in the two tests, the errors of the proposed method have been decreased a lot. At the beginning of prediction process, which is the most difficult part, the RUL prediction results of the proposed method have little difference with the actual RUL, while the SVR results are nearly two times of the actual RUL. For the first test, the SVR result in Fig. 11(c) is always worse than the proposed method, but it gets better rapidly, because the first test is a rapid degradation pattern. However, for the fast degradation pattern, predicting the RUL early and accurately is important.

In the second test of slow degradation, it can be seen that the RUL prediction result of the proposed method still shows high accuracy and robustness for a long prediction time duration. However, the prediction result of SVR differs greatly from the actual RUL until the last 100 s, which is why in Table II the RMSE of SVR is nearly 10 times than the proposed method, and the CRA is also worse. This is because the proposed method is driven by original signals directly, and provides accurate IF points, which not only guarantee the operational information can be fully used, but also make sure the RUL regression CNN model is trained accurately. However, even though the IF points are calculated accurately, and the training data sets are guaranteed to be effective, there is still a large gap between the results of SVR and real RULs. This can be explained as follows. In the proposed method, a weighted algorithm is used to retain the information before the current time, which can avoid the influence of outliers, whereas SVR, as well as most RUL prediction methods, only uses the features of the current sample. And, once an outlier appears, the result can be influenced a lot, which can also be seen in Figs. 11(c) and 12(c) that the oscillating amplitudes of SVR results in waveforms that are much larger than those of the proposed method. In addition, different bearings show different degradation time. And training data sets are usually different from the test data sets, which can lead to an incorrect result if the prediction model stays the same with the training model if we estimate the RUL directly as traditional RUL prediction methods. However, in the proposed method, a relative reliability is first calculated, instead of mapping the absolute RUL value directly in traditional methods. By replacing the absolute value with a relative value, the time variance problem can be solved and RUL can be predicted more reasonably.

To illustrate the noise robustness of the proposed method, the raw signals are added with white Gaussian noise and colored noise. The white noise $n(t)$ is constructed with a mean value of 0, and variance of 1. The colored noise is constructed as follows:

$$n_c(t) = n(t) - 0.5n(t). \tag{9}$$

Then, the proposed method is used to analyze the two signals under two different noise conditions. The prediction errors of them are shown in Table III. It can be seen that the addition of different noise only has a minimal influence on the performance of the proposed method, which verified the noisy and disturbance rejection abilities of the proposed method.

TABLE III
PREDICTION ERRORS OF THE PROPOSED METHOD UNDER TWO DIFFERENT NOISE CONDITIONS

		White noise	Colored noise
First test	RMSE	6.22	5.35
	CRA	0.67	0.67
Second test	RMSE	78.83	55.44
	CRA	0.84	0.87

IV. Conclusion

A double-CNN framework was proposed in this paper for intelligent RUL prediction. There are four major contributions in this paper to improve the prediction accuracy. First, the proposed framework was driven by original signals directly, which can not only solve the problem of feature extraction, but also retain the operational information. Second, two most common degradation patterns (sudden increasing trend and slow increasing trend) were considered to build different IF distinguishing models and a RUL prediction model for each pattern to been trained. Third, a weighted algorithm, as well as a “3/5” principle was proposed to ensure the stability of the prediction and eliminate or reduce the influence of abnormal points. Finally, an intermediate variable was introduced in this paper to solve the RUL time and performance variance problem, instead of constructing a RUL regression model directly in traditional methods. To verify the proposed method, its performance is compared with SVR, which is widely used for RUL prediction. The results clearly demonstrate the effectiveness of the proposed method. When dealing with other datasets, the corresponding parameters should be adjusted to ensure the performance, such as kernel size and pooling region. Future work will focus on simplifying the classifier as three classifications, that is zero as normal; one as sudden degradation, and two as slow degradation, which will improve the proposed method to be more compact.

References

[1] O. Fink, E. Zio, and U. Weidmann, “A classification framework for predicting components’ remaining useful life based on discrete-event diagnostic data,” *IEEE Trans. Rel.*, vol. 64, no. 3, pp. 1049–1056, Sep. 2015.

[2] Z. Zhao, B. Liang, X. Wang, and W. Lu, “Remaining useful life prediction of aircraft engine based on degradation pattern learning,” *Rel. Eng. Syst. Safety*, vol. 164, pp. 74–83, 2017.

[3] S. Al-Dahidi, F. D. Maio, P. Baraldi, and E. Zio, “Remaining useful life estimation in heterogeneous fleets working under variable operating conditions,” *Rel. Eng. Syst. Safety*, vol. 156, pp. 109–124, 2016.

[4] N. Li, Y. Lei, J. Lin, and S. X. Ding, “An improved exponential model for predicting remaining useful life of rolling element bearings,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7762–7773, Dec. 2015.

[5] K. Javed, R. Gouriveau, and N. Zerhouni, “State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels,” *Mech. Syst. Signal Process.*, vol. 94, pp. 214–236, 2017.

[6] Y. Hu, P. Baraldi, F. D. Maio, and E. Zio, “A particle filtering and kernel smoothing-based approach for new design component prognostics,” *Rel. Eng. Syst. Safety*, vol. 134, pp. 19–31, 2015.

[7] L. Liao, “Discovering prognostic features using genetic programming in remaining useful life prediction,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2464–2472, May 2014.

[8] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, and J. Dybala, “A model-based method for remaining useful life prediction of machinery,” *IEEE Trans. Rel.*, vol. 65, no. 3, pp. 1314–1326, Sep. 2016.

[9] B. Saha and K. Goebel, “Model adaptation for prognostics in a particle filtering framework,” *Int. J. Prognostics Health Manage.*, vol. 2, pp. 61–70, 2011.

[10] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, “Health index-based prognostics for remaining useful life predictions in electrical machines,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2633–2644, Apr. 2016.

[11] E. Zio, “Prognostics and health management of industrial equipment,” Diagnostics and prognostics of engineering systems: methods and techniques. IGI Global, 2013, pp. 333–356.

[12] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, “Direct remaining useful life estimation based on support vector regression,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2276–2285, Mar. 2017.

[13] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, “Residual life predictions from vibration-based degradation signals: A neural network approach,” *IEEE Trans. Ind. Electron.*, vol. 51, no. 3, pp. 694–700, Jun. 2004.

[14] R. Liu, B. Yang, X. Zhang, S. Wang, and X. Chen, “Time-frequency atoms-driven support vector machine method for bearings incipient fault diagnosis,” *Mech. Syst. Signal Process.*, vol. 75, pp. 345–370, 2016.

[15] L. Liao, “Discovering prognostic features using genetic programming in remaining useful life prediction,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2464–2472, May 2014.

[16] B. Yang, R. Liu, and X. Chen, “Fault diagnosis for wind turbine generator bearing via sparse representation and shift-invariant K-SVD,” *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1321–1331, Jun. 2017.

[17] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, “Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine,” *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1310–1320, Jun. 2017.

[18] R. K. Singleton, E. G. Strangas, and S. Aviyente, “Extended Kalman filtering for remaining-useful-life estimation of bearings,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1781–1790, Mar. 2015.

[19] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, 2018.

[20] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[21] L. Wen, X. Li, L. Gao, and Y. Zhang, “A new convolutional neural network-based data-driven fault diagnosis method,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[22] F. Chen and M. R. Jahanshahi, “Nb-cnn: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[23] F. Jia, Y. Lei, N. Lu, and S. Xing, “Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization,” *Mech. Syst. Signal Process.*, vol. 110, pp. 349–367, 2018.

[24] F. Lauer, C. Y. Suen, and G. Bloch, “A trainable feature extractor for handwritten digit recognition,” *Pattern Recognit.*, vol. 40, no. 6, pp. 1816–1824, 2007.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[26] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016.

[28] 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, pp. 98–109, 2017.

[29] K. Javed, “A robust & reliable data-driven prognostics approach based on extreme learning machine and fuzzy clustering,” Ph.D. dissertation, Université de Franche-Comté, Besançon, France, 2014.

[30] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, “Health assessment and life prediction of cutting tools based on support vector regression,” *J. Intell. Manuf.*, vol. 26, no. 2, pp. 213–223, 2015.

[31] Z. Hameed, Y. Hong, Y. Cho, S. Ahn, and C. Song, “Condition monitoring and fault detection of wind turbines and related algorithms: A review,” *Renewable Sustain. Energy Rev.*, vol. 13, no. 1, pp. 1–39, 2009.

[32] N. Patrick *et al.*, “Pronostia: An experimental platform for bearings accelerated degradation tests,” in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, 2012, pp. 1–8.

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.

[35] A. Ginart, I. Barlas, J. Goldin, and J. L. Dorrity, “Automated feature selection for embeddable prognostic and health monitoring (PHM) architectures,” in *Proc. IEEE AutoTestCon*, 2006, pp. 195–201.

[36] T. H. Loutas, D. Roulias, and G. Georgoulas, “Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression,” *IEEE Trans. Rel.*, vol. 62, no. 4, pp. 821–832, Dec. 2013.

[37] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, “Metrics for offline evaluation of prognostic performance,” *Int. J. Prognostics Health Manage.*, vol. 1, no. 1, pp. 4–23, 2010.

Boyuan Yang received the B.A., M.A., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi’an Jiaotong University, Xi’an, China, in 2013, 2015, and 2019, respectively.

He currently is a Postdoctoral Researcher with the School of Electrical and Electronic Engineering, University of Manchester, Manchester, U.K. His research interests include intelligent manufacturing, machine learning, condition monitoring, and wind energy.

Ruonan Liu (M’19) received the B.A., M.A., and Ph.D. degrees in mechanical engineering from the School of Mechanical Engineering, Xi’an Jiaotong University, Xi’an, China, in 2013, 2015, and 2019, respectively.

She is currently a Postdoctoral Researcher with the Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Her research interests include intelligent manufacturing, natural language processing, computer vision, and machine learning.

Enrico Zio (SM’09) received the B.S. and Ph.D. degrees in nuclear engineering from the Politecnico di Milano, Milano, Italy, in 1991 and 1995, respectively, the M.Sc. degree in mechanical engineering from the University of California, Los Angeles, CA, USA, in 1995, and the Ph.D. degree in nuclear engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1998.

He is currently a Full Professor of Risk and Reliability with the Centre for Research on Risk and Crisis, Ecole de Mines, ParisTech, PSL Research University, Sophia Antipolis, France. He is also a Full Professor with the Politecnico di Milano. He has authored or coauthored seven books and more than 300 papers in international journals. His research interests include the modeling and analysis of complex systems, reliability, maintainability, prognostics, safety, vulnerability, resilience, and security characteristics.