International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Performance degradation assessment of rolling bearing based on convolutional neural network and deep long-short term memory network

Zheng Wang, Hongzhan Ma, Hansi Chen, Bo Yan & Xuening Chu

To cite this article: Zheng Wang, Hongzhan Ma, Hansi Chen, Bo Yan & Xuening Chu (2020) Performance degradation assessment of rolling bearing based on convolutional neural network and deep long-short term memory network, International Journal of Production Research, 58:13, 3931-3943, DOI: 10.1080/00207543.2019.1636325

To link to this article: https://doi.org/10.1080/00207543.2019.1636325

International Journal of Production Research, 2020
Vol. 58, No. 13, 3931–3943, https://doi.org/10.1080/00207543.2019.1636325

Performance degradation assessment of rolling bearing based on convolutional neural network and deep long-short term memory network

Zheng Wang, Hongzhan Ma, Hansi Chen, Bo Yan and Xuening Chu*

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, People's Republic of China

(*Received 22 October 2018; accepted 22 June 2019*)

Many traditional approaches for performance degradation assessment of rolling bearings, using sensor data, make assumptions about how they degrade or fault evolve. However, the sequential sensor data cannot be directly taken as input in the traditional models since the data always contain noise and change in length. To solve these problems, a convolutional neural network and deep long-short term memory (CNN-DLSTM) based architecture is proposed to obtain an unsupervised H-statistic for performance degradation assessment of rolling bearing using sensor time-series data. Firstly, a CNN is applied to extract local abstract features from raw sensor data. Secondly, a deep LSTM is explored to extract temporal features. CNN-DLSTM is trained to reconstruct the time-series sensor signal reflecting the health condition of rolling bearing. The D- and Q-statistic are used to compute H-statistic which is then used for performance degradation assessment. The proposed approach is evaluated on an experiment with rolling bearings and the results are presented on a public dataset of rolling bearing, verifying that the proposed approach outperforms several state-of-the-art methods.

Keywords: performance degradation assessment; feature extraction; convolutional neural network; long-short term memory network; deep learning

1. Introduction

In modern industrial production systems, different components work together to achieve a given goal (Negrichi, Mascolo, and Flaus 2017). As a key component in many production fields, the health condition of rolling bearing largely impacts the performance of entire mechanical systems. However, rolling bearings usually operate under tough and variable working environments, which makes them vulnerable to damage in engineering applications. Although rolling bearings serve with product components at the same working condition, their service lives could vary widely. Assessing the performance of rolling bearings can not only ensure the smooth and efficient operation of mechanical equipment, but also detect and eliminate unexpected failure events in service. Therefore, it is crucial to assess the performance degradation of rolling bearings based on analysing the real-time monitored sensor signals.

Assessing the performance degradation of rolling bearing is fundamentally classified as machine health monitoring system (MHMS) problem. The methods applied to MHMS generally include two types, i.e. physics-based models and data-driven models. Methods based on physics aims at studying the degradation mechanism of components and establishing a physical model of component performance degradation process, such as the Paris crack growth model (Li, Kurfess, and Liang 2000) and the Forman crack growth model (Kirubarajan 2002). One advantage of physics-based approaches is that they provide a physical understanding of the system. However, the disadvantage of these methods is that mathematical models may be unavailable or inaccurate due to the simplifications of many applications ignoring nonlinearities, dynamics or lack of information in most complex systems.

Signal feature extraction methods are the key in sensor signal data-driven assessment models, since the raw sensor data are always noisy. The researches on signal feature extraction mainly employ the data in time, frequency domain or both the two domains (Plaza and López 2018). A classic method of analysing time domain data is time direct analysis (TDA) since it is simple with low analytical-computational cost (Hessainia et al. 2013). Based on the classic TDA, more advanced methods were proposed, for instance, singular spectrum analysis (SSA) (Plaza and López 2017) and principal component analysis (PCA) (Shi and Gindy 2007), which improve the effectiveness in acquiring sensor data. In the frequency domain, most widely used methods include the fast Fourier transform (FFT) (Bhuiyan, Choudhury, and Dahari 2014) and power spectral density (PSD) methods (Wang, To, and Chan 2013). However, the aforementioned methods do not capacitate the

*Corresponding author. Email: xnchu@sjtu.edu.cn

temporary location of transient events, which largely limits the various process parameters monitoring. To address the issue, time–frequency analyses rise. The most popular approaches of this category are the short time Fourier transform (STFT) (Rehorn, Sejdić, and Jiang 2006) and wavelet transform (WT) (Plaza and López 2018). However, the two methods require manually extract features, which increases the negative effects of incorrect priori knowledge of human beings. Compared with the conventional methods listed above, the convolutional neural network (CNN) shows outstanding characteristics due to its special structure of the convolutional layer and pooling layer. The idea of shared weights and local sensing largely reduces the parameter numbers and makes CNN easy to use without tedious manual computation process of preplanned extracting feature and loss of feature information. CNNs are widely used in image recognition (Wang et al. 2018), human attribute recognition (Perlin and Lopes 2015) and natural language processing (Collobert and Weston 2008). However, this kind of method is rarely used in signal feature extraction field. In this method, firstly a CNN is used to extract local robust features.

The sensor signals not only contain the local information, but also the temporal information that need to be extracted. In recent years, machine learning techniques have progressed rapidly and they can uncover both the mapping relationship between feature expression and the output, and also the feature itself without any expert knowledge, which is called the representation learning. Through this method, the deeper and more robust information can be acquired. In addition, machine learning techniques can quickly adapt to new tasks with less manual intervening. As the most used technique of machine learning, neural networks have been widely explored owing to their superior function approximation ability and nonlinearity mapping ability. The outputs of hidden layers or output layers contain various internal information of the input data. Therefore, neural networks can also be considered as a kind of effective tool for feature extraction. These methods can generally acquire accurate results. And the parameters of model are adjusted dynamically in real time with machine health condition varies.

Traditional data-driven performance degradation methods include artificial back propagation neural network, radial basis function network, support vector machine (SVM), restricted Boltzmann machine (RBM) and auto-encoder (AE) (Sun et al. 2016; Lu et al. 2017). These methods can be regarded as the superficial feature extraction methods. With the popularity of deep learning methods since 2006, multiple hidden layers are constructed and massive data can be used for model training. Finally, more precise results have been achieved. Recently, data-driven MHMS receives widespread concern with the popularity and interconnection of low-cost sensors. At the same time, deep learning provides a golden opportunity for researchers to better process and analyse mass data. Deep learning methods applied into MHMS area mainly include deep auto-encoder network (DAEN) and its variants, deep Boltzmann machine (DBM), deep belief network (DBN), and recurrent neural network (RNN) including its variants. Zhu et al. (2016) firstly extracted frequency domain features of data samples using Fourier transform, then the feature vectors were fed into a sparse DAEN for hydraulic pump fault diagnosis. Tao et al. (2015) designed different structures of sparse DAEN with different sizes of hidden layer and tested the performances for fault identification task. DBM can be constructed by stacking multiple hidden layers of RBM, keeping the input and output layer unchanged (Rui et al. 2019). Li et al. (2016) applied DBM to extract advanced features of the sensor data after wavelet packet transform for gearbox fault diagnosis. Shao et al. (2016) presented DBN for fault diagnosis of induction motor using vibration signals as input. Ma et al. (2016) proposed an architecture based on DBN for assessing the degradation of bearings. In comparison to the traditional machine learning methods, deep learning-based MHMS is able to directly process raw data, which endows it with two advantages: extracting more informative features automatically and meeting end-to-end requirements in practical applications. However, these kinds of network are not convenient for time-series data processing.

RNN including long-short term memory (LSTM) and gated recurrent unit (GRU) models have become popular in processing sensor monitored data in time series without ancillary processing comparing with other deep learning architectures. RNN has a unique architecture where the outputs of neurons are applied recursively to their own inputs and each input is only connected to its own hidden neuron, which gives RNN the superior ability to encode information in time sequence. However, RNN has shortcomings in its restricted temporal features and poor capacity in back propagation (Yildirim 2018). In the gradient descent algorithm, the output of each step depends not only on the network of the current step, but also on the state of the network in the previous steps. As the iteration of the time step is performed, the back-propagated training method brings about the gradient vanishing problem. To address the issue, LSTM was proposed (Hochreiter and Schmidhuber 1997). In LSTM, memory cells (i.e. input gates, output gates and forget gates) replace each neuron in hidden layers of RNN. Specifically, input gates update the cell states; forget gates discard historical information and reset memory cells; output gates output cell states. The memory cells endow LSTM with the property of extracting temporal features by means of cell states. Yuan, Wu, and Lin (2016) conducted fault diagnosis and prognostics of aero engine using several types of RNN, including vanilla RNN and LSTM. The results show that the LSTM models outperformed vanilla RNN. Considering the successful applications of these deep architectures in representation learning (Bengio 2009), it is valuable to construct a deep LSTM (DLSTM) by stacking multiple LSTM layers for the purpose of improving the accuracy of results.

In this paper, CNN is combined with DLSTM to propose a health monitoring and assessment architecture for rolling bearing performance degradation, namely CNN-DLSTM. In the proposed method, CNN is used to extract local robust features. DLSTM is explored to extract the sequential information and learn feature representations. Finally, an H-statistic is computed to assess the health condition of rolling bearings. To verify the effectiveness and feasibility of the method, a case of performance degradation assessment of rolling bearing is performed. Results of several state-of-the-art models are also contrasted with that of the proposed method. It is worth mentioning that the work contributed by Rui et al. (2017) applied similar architecture for tool wear depth prediction. However, the main difference between their work and this paper lies in that the tool wear prediction task utilises the whole sequence of sensor data for both training and testing, while this paper uses only the early part of the data sequences under normal condition as training set and thus the model is applied to reconstruct the healthy condition of the rolling bearing. For measuring the degree of degradation, data samples at different time steps in the full-life cycle are fed into the model and the output data are compared to those using healthy data as input. Larger deviations indicate more serious degradation. Different training and operation modes challenge the effectiveness and generalisability of the architecture and also pose a novel way of solving the problem in this paper. In addition, a unidirectional LSTM module is applied in our architecture while Rui et al. (2017) used a bidirectional one.

The remainder is organised as follows: In Section 2, some preliminary studies are reviewed. In Section 3, the proposed CNN-DLSTM and the subsequent performance assessment module are illustrated. In Section 4, experimental results of performance degradation assessment of rolling bearings are presented and then methods comparison is conducted. In Section 5, conclusions are finally addressed.

2. Preliminaries

2.1. CNN

CNN was firstly proposed by LeCun et al. (1990) in image processing field and then made successes in many application fields with the help of two key properties: weight sharing and spatial pooling. CNN aims at learning abstract features by alternant convolution and pooling operations. Convolutional layers convolve several local filters with input, generating constant local features by using a fixed-length sliding window over the sequence (Rui et al. 2019). Pooling is also an important operation in CNN, which can reduce features while maintaining local invariance of features. Commonly used pooling operations are: spatial pyramid pooling, max pooling, mean pooling, stochastic pooling, etc.

To address sequential data, the convolutional layer in CNN is firstly applied over the raw input sequence. The feature maps of upper layers are convolved by convolutional kernels, and output feature maps are acquired after activation. Every output feature map combines values of multiple feature maps. Then, through pooling layers, the feature maps are compressed, which makes them smaller and simplifies the computation complexity of network. Thus, vital and informative local features are extracted. In 1D-CNN with a single convolutional layer and pooling layer, details are introduced as follows:

Convolution: The convolution operation is signified by multiplying filter vector $\boldsymbol{u} \in \mathbb{R}^{md}$ with concatenated vector $\boldsymbol{x}_{i:i+m-1}$, shown as follows:

$$\boldsymbol{x}_{i:i+m-1} = \boldsymbol{x}_i \oplus \boldsymbol{x}_{i+1} \oplus \cdots \oplus \boldsymbol{x}_{i+m-1} \tag{1}$$

where $\boldsymbol{x}_{i:i+m-1}$ denotes a duration of m continuous time steps beginning with the i-th time step. By adding the bias vector b in the convolution operation, the final equation can be denoted as:

$$c_i = \mathrm{g}(\boldsymbol{u}^T \boldsymbol{x}_{i:i+m-1} + b) \tag{2}$$

where Rectified Linear Units (ReLu) (Nair and Hinton 2010) is set to be the activation function g.

Each u represents a convolutional kernel, and c_i can be considered as the activated time duration.

Max-pooling: The pooling layer is constructed after convolutional layer, reducing the dimensionality of the upper convolution output by sliding the filtering window from start to end and taking a max of each part of the feature map to represent each corresponding area. A feature map can be given by a vector as follows:

$$\boldsymbol{c}_j = [c_1, c_2, \ldots, c_{l-m+1}] \tag{3}$$

where j denotes the j-th filter, corresponding to multiple windows $\{\boldsymbol{x}_{1:m}, \boldsymbol{x}_{2:m+1}, \ldots, \boldsymbol{x}_{l-m+1:l}\}$. Through pooling operation, the parameters of the model further decrease. The pooling length is given by s and the compressed feature vector is given as follows:

$$\boldsymbol{h} = [h_1, h_2, \ldots, h_{((l-m)/s)+1}] \tag{4}$$

where $h_j = \max(c_{(j-1)s}, c_{(j-1)s+1}, \ldots, c_{js-1})$. In general, different weights are initialised to generate various filters to obtain the output of the CNN.

2.2. Deep LSTM

The LSTM gating mechanism controls the selective passage of information, enabling the memory cells to preserve long-term dependencies while also preventing internal gradients from external disturbance in training phase. Each memory cell has a self-loop connected linear unit that allows the error to propagate at a constant value inside the memory unit, avoiding the problem of gradient vanishing and the long-term dependence of RNN.

Mathematically, the output of the memory unit $\boldsymbol{h}^t$ is computed iteratively by the following equations:

$$\begin{aligned}
\boldsymbol{i}^t &= \sigma(\boldsymbol{W}^i\boldsymbol{x}^t + \boldsymbol{V}^i\boldsymbol{h}^{t-1} + \boldsymbol{b}^i) \\
\boldsymbol{f}^t &= \sigma(\boldsymbol{W}^f\boldsymbol{x}^t + \boldsymbol{V}^f\boldsymbol{h}^{t-1} + \boldsymbol{b}^f) \\
\boldsymbol{o}^t &= \sigma(\boldsymbol{W}^o\boldsymbol{x}^t + \boldsymbol{V}^o\boldsymbol{h}^{t-1} + \boldsymbol{b}^o) \\
\boldsymbol{c}^t &= \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{i}^t \odot \tanh(\boldsymbol{W}^c\boldsymbol{x}^t + \boldsymbol{V}^c\boldsymbol{h}^{t-1} + \boldsymbol{b}^c) \\
\boldsymbol{h}^t &= \boldsymbol{o}^t \odot \tanh(\boldsymbol{c}^t)
\end{aligned} \tag{5}$$

where $\boldsymbol{i}^t$, $\boldsymbol{f}^t$, $\boldsymbol{c}^t$ and $\boldsymbol{o}^t$ are the output of the input gate, forget gate, memory cell and output gate, respectively; $\boldsymbol{W} \in \mathbb{R}^{d\times k}$, $\boldsymbol{V} \in \mathbb{R}^{d\times d}$ are the shared weight matrices and $\boldsymbol{b} \in \mathbb{R}^d$ is the shared bias vector, which are updated over and over again during training phase; k denotes a hyper-parameter indicating the dimension of hidden vectors; σ denotes the sigmoid activation function; $\odot$ represents the element-wise product.

The final output is obtained by taking the output at the terminal time step through a linear regression layer, shown as follows:

$$\bar{y}_i = \boldsymbol{W}^r\boldsymbol{h}_i^T \tag{6}$$

where $\boldsymbol{W}^r \in \mathbb{R}^{k\times z}$ and z represents the dimension of output. Mean squared error (MSE) is considered as the loss function for model training, where $\bar{y}$ denotes predicted value and y denotes the target value.

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n}(\bar{y}_i - y_i)^2 \tag{7}$$

where n represents the total number of training samples. Figure 1(a) illustrates the corresponding LSTM architecture.

As for 'deep learning', the 'deep' in it refers that the input data pass through multiple hidden layers while training or using the model. Specifically, in DLSTM, the input data are transformed into more advanced and abstract representations through each level of hidden layers. Deep learning obviates feature engineering by transforming data into advanced feature representations, and derives multilayered structures to remove redundancy in representation, finally outputs the data containing various internal information of the input data. As shown in Figure 1(b), the input to the DLSTM model are transformed into representations through both hidden layers at the same time step and sequential cells at the next time step. And the iterative formulas for the l-th layer are illustrated as follows:

$$\begin{aligned}
\boldsymbol{i}^t &= \sigma(\boldsymbol{W}_l^i\boldsymbol{h}_{l-1}^t + \boldsymbol{V}_l^i\boldsymbol{h}_l^{t-1} + \boldsymbol{b}_l^i) \\
\boldsymbol{f}_l^t &= \sigma(\boldsymbol{W}_l^f\boldsymbol{h}_{l-1}^t + \boldsymbol{V}_l^f\boldsymbol{h}_l^{t-1} + \boldsymbol{b}_l^f) \\
\boldsymbol{o}_l^t &= \sigma(\boldsymbol{W}_l^o\boldsymbol{h}_{l-1}^t + \boldsymbol{V}_l^o\boldsymbol{h}_l^{t-1} + \boldsymbol{b}_l^o) \\
\boldsymbol{c}_l^t &= \boldsymbol{f}_l^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{i}_l^t \odot \tanh(\boldsymbol{W}_l^c\boldsymbol{h}_{l-1}^t + \boldsymbol{V}_l^c\boldsymbol{h}_l^{t-1} + \boldsymbol{b}_l^c) \\
\boldsymbol{h}_l^t &= \boldsymbol{o}_l^t \odot \tanh(\boldsymbol{c}_l^t)
\end{aligned} \tag{8}$$

The input to the first layer of DLSTM is the output representations obtained from CNN, i.e. $\boldsymbol{h}_0^t = \boldsymbol{x}^t$. Finally, regression is applied using the output layer of DLSTM at the last time step as the input.

2.3. Performance degradation assessment using D- and Q-statistic

Quantifying moderately sensitive indices is one of the most important processes of performance degradation assessment. In this section, we mainly take the example by the reference (Yu 2012) and more details can be found in the paper. Based on the feature representation extracted from models, the idea of the D- and Q-statistic is applied to the continuous processes (Westerhuis, Gurden, and Smilde 2000) and employed as rolling bearing performance quantifying index for assessment.

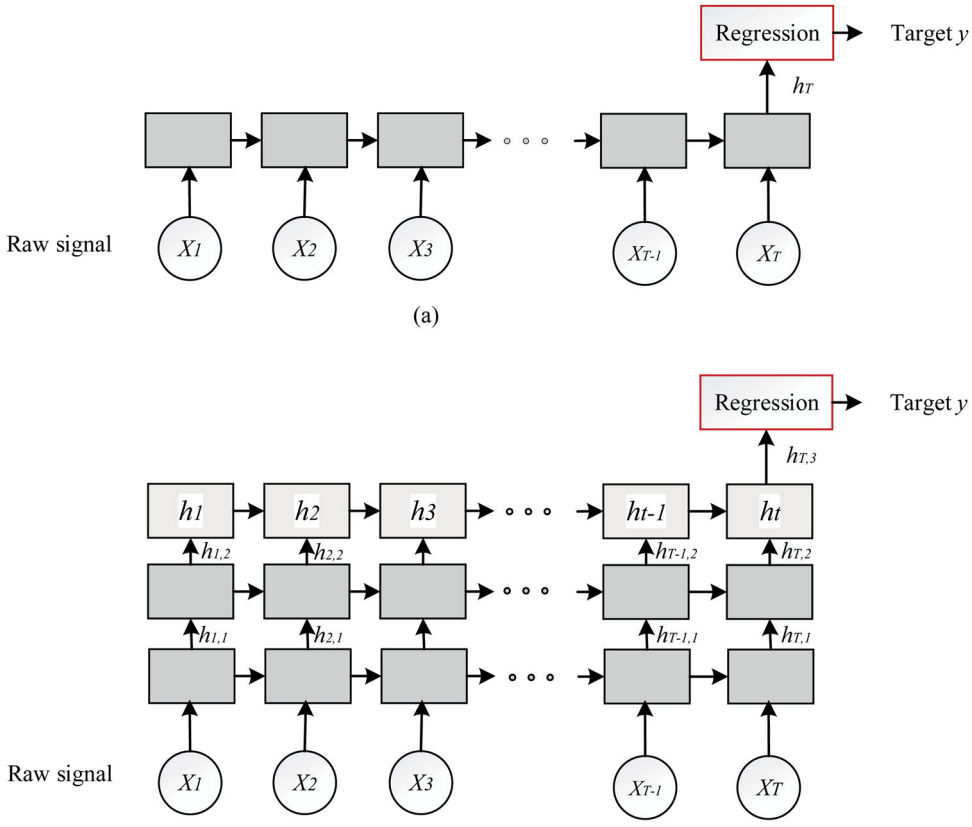

Figure 1. Illustrations for basic LSTM and deep LSTM model with multiple layers for sequential data regression problem. Grey block denotes a LSTM layer, and dark red block represents a linear regression layer.

Q-statistic is also called squared prediction error (SPE) and *D*-statistic is also called T^2 statistic. A detailed introduction and mathematical principles of *Q*-statistic and *D*-statistic can be found in Lee, Yoo, and Lee (2004) and Williams et al. (2006). An empirical latent variable model can be constructed to depict the data using feature representations derived from the CNN-DLSTM model in a group of consecutive time intervals under normal operating conditions.

$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^T$ in $\mathbb{R}^d$ denotes a normal feature representation matrix under normal operating condition, where N is the sample order number; d is the total number of features. $\mathbf{X}$ can be represented as a linear combination of m unknown latent variables and m is always much less than d. $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]^T$. The relationship between them is:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{Y}\boldsymbol{B}^{\mathrm{T}} + \boldsymbol{E} \\ \boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{B}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1} \\ \boldsymbol{E} &= \boldsymbol{X} - \boldsymbol{Y}\boldsymbol{B}^{\mathrm{T}} \end{aligned} \tag{9}$$

where $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_d] \in \mathbb{R}^{d \times m}$ denotes the actual model and represents the similarity of sequential time points. $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]^T$ in $\mathbb{R}^m$ is the latent variable matrix and represents the difference between sequential time points. $\boldsymbol{E} \in \mathbb{R}^{N \times d}$ denotes the residual matrix, containing the nonsystematic section not expressed by model. In many models, $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B}$.

As shown in Equation (9), a novel feature representation x_{new} is projected onto the model to identify whether the component or machine is in statistical control. Then two statistics with known distributions, namely *D*-statistic for the systematic part and the *Q*-statistic for the residual part of the process variation, are computed using the process model. The *D*-statistic describes the changes in latent variable space and the summation of normalised squared scores, given by:

$$D = \boldsymbol{y}_{\mathrm{new}}^T \boldsymbol{S}^{-1} \boldsymbol{y}_{\mathrm{new}} \tag{10}$$

where $\boldsymbol{S}^{-1} = ((\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})/(\boldsymbol{I}-1))^{-1}$. According to F-distribution, the $100(1-\alpha)\%$ control limit of D is calculated as follows (Westerhuis, Gurden, and Smilde 2000):

$$\eta_D = \frac{a(N-1)}{N-a}\mathrm{F}(a, N-1; \alpha) \tag{11}$$

where a and $N-1$ are the degrees of freedom of F-distribution and α is the level significance.

In addition, Q-statistic is able to capture the variation part not detected by the D-statistic. The Q-statistic indicates the degree of the conformation of the sample vectors to the model, measuring their projection on **E**. Q-statistic is defined as the summation of the square residual of e

$$Q = \sum_{i=1}^{d} Q_i = \sum_{i=1}^{d} {e_i}^2 = (x_i - \widehat{x_i})^2. \tag{12}$$

The predicted value $\widehat{x_i}$ is obtained from the (9) model. The nominal working range of Q-statistic can be defined using kernel density estimation (KDE) (Martin and Morris 1996), and the single-variable kernel estimator is given by

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{N} \mathrm{K}\left(\frac{x - x_i}{h}\right) \tag{13}$$

where K is a kernel function; x is the feature representation to be processed; x_i is the data value from the feature representation set under normal condition and h is the smoothing parameter. The kernel function follows the rule:

$$\int_{-\infty}^{\infty} \mathrm{K}(x)\mathrm{d}x = 1\mathrm{K}(x) > 0 \tag{14}$$

Multiple applicable kernel functions can be utilised for K. Practically, the most popular one is Gaussian kernel function. It is worth noting that estimating the confidence limit η_Q can be tough for data discretely distributed or clustered with minor changes while applying methods based on density estimation to define it (Chen, Kruger, and Leung 2004). Hence, the data distributions of the sensor data under normal condition are collected on the basis of the first two feature vectors. The result indicates the data do not follow the discrete distribution, thus the estimation is able to proceed. According to the confidence coefficient requested (99.9%), η_D and η_Q are calculated respectively to represent the confidence bounds of D- and Q-statistic by applying KDE method.

In modern industrial environment, using a single index for monitoring is important since it helps to lessen the work load and operating cost of manipulators. Taking the example by Yue and Qin (2001) who applied a combined index for malfunction detection, the combined H-statistic in this paper is used as follows (Yu 2012):

$$H = \frac{D}{\eta_D} + \frac{Q}{\eta_Q} \tag{15}$$

The combined index integrates the D- and Q-statistic into one and serves as a substitute index for bearing performance degradation assessment.

3. The proposed approach

The proposed approach is described according to its processes in this section. It mainly consists of four parts. They are one-layer CNN, 4-layer DLSTM (deep LSTM with four hidden layers), fully connected and linear regression layer, finally followed with a performance degradation assessment module. The main processes of the proposed approach are illustrated in Figure 2.

3.1. Local feature extraction based on CNN

The CNN used in this paper contains a convolutional layer and a pooling layer and takes the raw sensor data as input. Feature maps are firstly generated by sliding the convolutional kernels over the whole time-series input and can be considered as the convolutional activation of kernels over the sensor data. The pooling layer is then used to condense the features in feature maps and maintain the informative features. Through the convolutional layer and pooling layer, the local feature extraction from raw sensor data is finally attained. The architecture for the CNN with a convolutional layer and pooling layer is shown in Figure 3 to present a visual form.

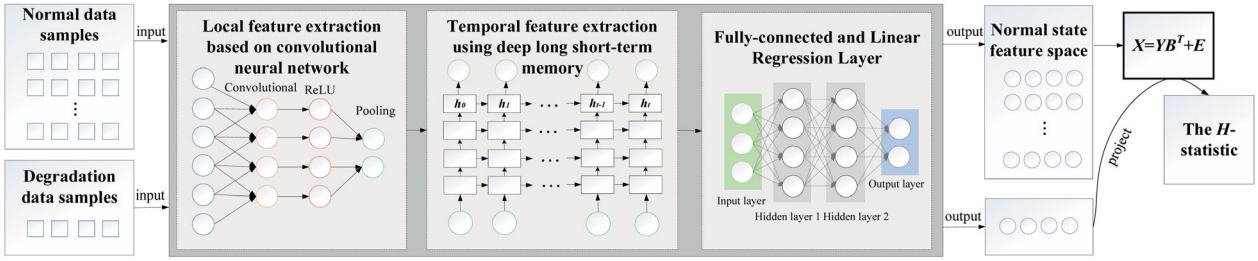

Figure 2. Mainly processes of the proposed approach.

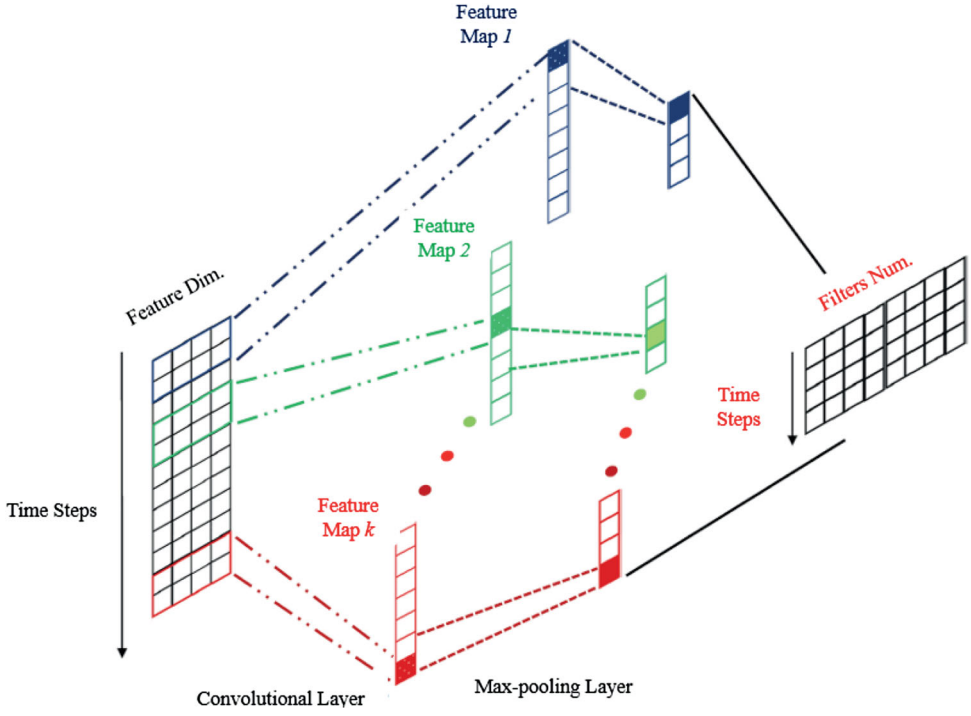

Figure 3. Illustrations for the local feature extraction based on CNN.

It can be calculated that the convolutional layer and pooling layer reduces the size of raw input from l to $((l-m)/s)+1$, where m is the filter size and s is the pooling length. The primary representation is the collected sensor data with the dimension of d which equals to the total sensor number at each time step. Robust and abstract feature representation is extracted through CNN and the dimension of the representation equals to k, the number of convolutional kernels. Hence, CNN extracts the local feature representations of the raw sensor data so as to output more informative feature vectors, which serve as the input to the latter DLSTM model.

3.2. *Temporal feature extraction using deep LSTM*

In this step, a 4-layer DLSTM is constructed after the CNN. The deep sequential model is used to extract the temporal feature from the output of CNN. The output of the DLSTM summarises the temporal information and is then fed into the next step, fully connected dense layers.

It is necessary to adopt regularisation for DLSTM for ensuring the robust performance of the deep learning model in machine monitoring problem. Dropout is introduced during model training for DLSTM (Srivastava et al. 2014) as this technique is practical in addressing gradient vanishing and overfitting problems (Zhu et al. 2016). In the training phase of neural network, the weights of certain neurons in hidden layers are randomly reset to zero. Since different nodes are affected by the reset in each iteration, the 'importance' of each node will be balanced. By dropout, all the neurons of the network will contribute and the condition that few high-weight neurons fully control the outputs can be avoided, thus largely reducing the structural risk of the network.

3.3. Fully connected and linear regression layers

Representations obtained from 4-layer LSTM contain both the local features and the temporal information of the sensor data. Then it is fed into the subsequent multilayers to obtain a more advanced representation. After the DLSTM, we stack two fully connected dense layers, where the former layer outputs the input of the latter. Mathematically, the relation can be denoted as follows:

$$\boldsymbol{o}^i = \mathrm{g}(\boldsymbol{W}_i \boldsymbol{h}^i + \boldsymbol{b}_i) \tag{16}$$

where $\boldsymbol{o}^i$, $\boldsymbol{h}^i$, $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ respectively represent the output, input, weight matrix and bias vector of the i-th fully connected layer. The activation function g() applies the ReLu function. The output of final fully connected layer is then taken as the input of the linear regression, which is used to reconstruct the sensor signal and restore the length and dimension of the raw input sensor signal.

3.4. Model training and performance degradation assessment

The whole model including CNN, DLSTM, fully connected and linear regression layers are trained using sensor signal data samples under normal operating condition as both model inputs and target outputs. Given the target outputs and actual outputs, the MSE of training samples is computed and the parameters of model can be updated through the back-propagated errors. Root Mean Square Propagation (RMSprop) is applied to optimise the parameters of the model, using the adjacent gradients to normalise the gradient. Then, the outputs of the neural networks are used to initialise the performance degradation assessment module. Then this entire model is used to test the performance degradation condition of bearings using new data samples.

The D- and Q-statistic are combined to generate a novel H-statistic used to provide a quantification index indicating the health condition of the rolling bearing for maintenance manipulators. In addition, the H-statistic at each time step constructs a monitor graph for monitoring the performance degradation process of the rolling bearing during the full-life cycle. The KDE mentioned in Section 2.3 is applied to set the threshold value for triggering the slight degradation alarm. In this paper, the vibration acceleration sensor data of rolling bearing 3 are used to model and generate H-statistic for assessing the performance degradation aggravation of the bearing.

4. Experiments

On the basis of a real-world dataset of rolling bearings, experiments are implemented to verify the effectiveness of the proposed method. Firstly, details about the experiment are given. Secondly, the proposed CNN-DLSTM method is applied to implement feature extraction for sensor data and the D- and Q-statistic approach is applied to generate the H-statistic for performance degradation assessment of the bearing. Finally, the reconstruction errors of several methods are compared to evaluate the performance of the CNN-DLSTM architecture for data samples reconstruction.

4.1. Experimental setup

To investigate the effectiveness of the proposed approach for degradation performance assessment of rolling bearing, a case is considered in this section. The data source for the method validation is the vibration acceleration signal collected in the life test of Rexnord ZA-2115 double row rolling bearings from Prognostics Center Excellence through the prognostic data repository contributed by Intelligent Maintenance System, University of Cincinnati (Lee et al. 2007). The structure of the test rig is shown in Figure 4. The number of bearing rolling elements is 16; the pitch circle diameter and the rolling element diameter are 2.815 and 0.331 in, respectively; the tapered contact angle is 15.17°; the motor rotation speed is 2000 r/min and the rotor is driven by the belt drive. The bearings supporting the rotor are forcibly lubricated, and each bearing housing is mounted with a PCB 353B33 High Sensitivity Quartz ICP Accelerometer in its horizontal x and vertical y directions. Bearing 2 and 3 are subjected to a radial load of 26 kN which is much greater than their rated dynamic load. The data sampling frequency remains at 20 kHz and the length of data is 20,480 time steps. Bearing 3 showed inner ring damage on the 35th day after the start of the test and the life test stopped, which is used to illustrate the performance of the proposed approach. The task is to work out the trend of H-statistic for the rolling bearing and analyse the degradation process. Python 3.6 and Macbook Pro with iOS operating system and 8GB of memory are used for model construction and data processing.

4.2. Performance degradation assessment results

Performance degradation assessment of rolling bearing aims to quantify the degree of degradation and to trail further deterioration of the bearing. The hyper-parameters in this research are chosen by manual selection in order to prevent the condition

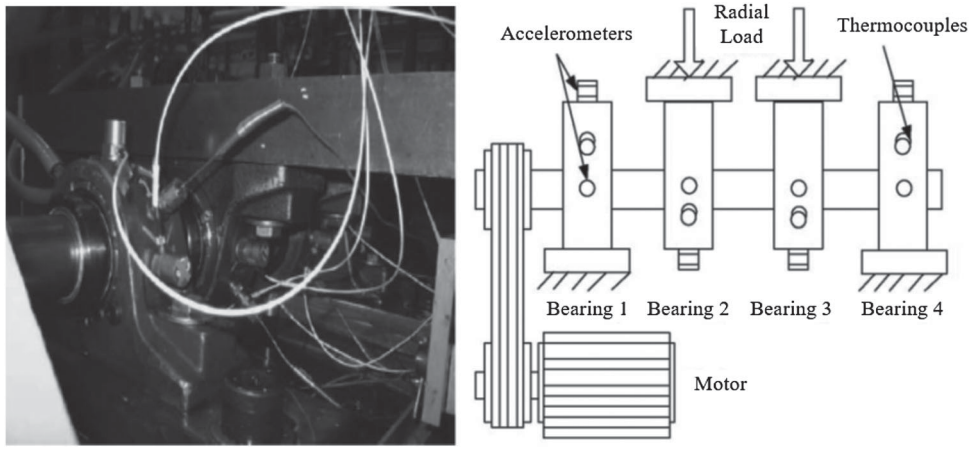

Figure 4. Test rig.

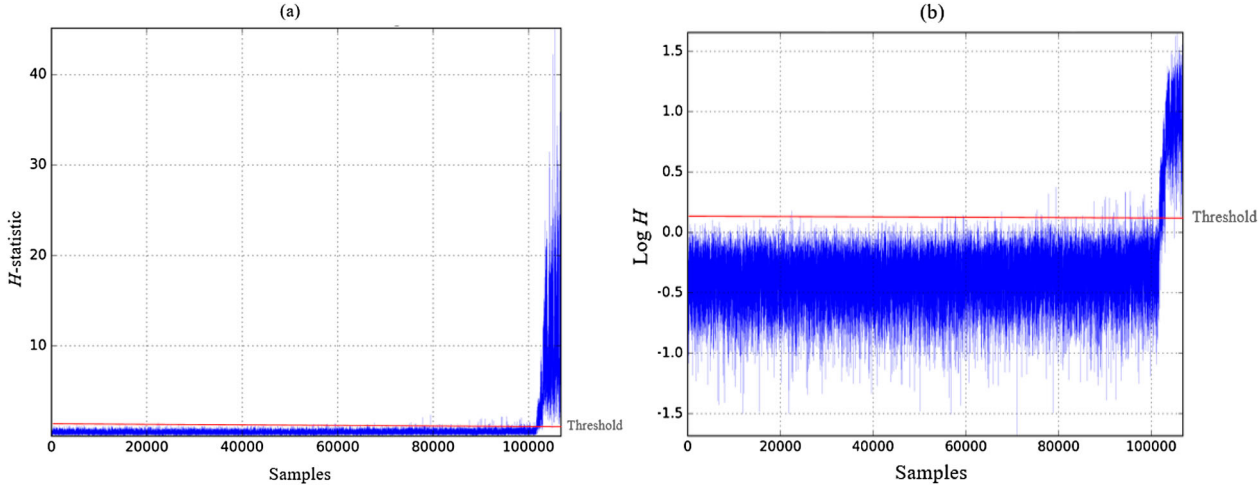

Figure 5. Performance degradation assessment of bearing 3 on its whole lifetime. (a) H-statistic of bearing 3. (b) Logarithmic H-statistic of bearing 3.

of overfitting and underfitting, thus the capacity of each model is adjusted to map the complexity of the task. More detailed instructions can be found in book 'Deep learning' chapter 11 written by Goodfellow, Bengio, and Courville (2016).

For the proposed CNN-DLSTM, firstly a 1D-CNN is applied, whose filter size, filter number and pooling size are 100, 10 and 50, respectively. Hence, the CNN transforms the shape of the raw sequential data from 106,800 $\times$ 1 into 2136 $\times$ 10. Subsequently, a 4-layer DLSTM with the layer sizes [178, 200, 400, 600] is built after the CNN and outputs the representations containing both local features and temporal features. Then, two fully connected layers with the size of [600, 890] are applied in advance of the linear regression layer. ReLu serves as the activation functions. The early quarter of data samples of bearing 3 on the x-axis are firstly used to train the neural network layers. And the outputs of neural networks are used to initialise the performance degradation assessment module. Data samples of bearing 3 on the y-axis are used as the testing set and to generate the H-statistic trend chart.

The result for the performance degradation assessment is presented with an H-statistic value that is shown from the start of the test until the end of the test, when the bearing condition and vibration are deemed too severe to continue or failure occurred. The D- and Q-statistic are applied to generate the H-statistic and KDE approach is used to set the threshold for triggering slight degradation alarm, shown in Figure 5(a). The logarithmic H-statistic and the threshold for bearing 3 are also shown in Figure 5(b) in view of the great discrepancy in scale between the H-statistic and the threshold in the final phase of the rolling bearing life.

H-statistic shows a stabilised and explicit change tendency for the degradation duration, greatly increasing the apprehensibility of the quantitative index in practice. It is shown that the degradation process of the bearing transits apparently from health, slight degradation, severe degradation and finally to failure. H-statistic shows a stable trend in the previous stage,

then increases with the continuous deterioration of the bearing, and increases even faster until it finally fails. In addition, the duration from the occurrence of severe degradation to failure is quite short, indicating that the maintenance actions based on performance monitoring should be taken to avoid the unexpected failure in case the slight degradation occurs. And a uniform threshold can be set up to trigger the alarm for the occurrence of the slight degradation (see Figure 5). It can be observed that several abnormal values occur and cause the false alarms, which cannot always be avoided because in that way the sensitivity of the monitoring system would decrease. These abnormal values possibly arise from certain harmful physical characteristics to the robustness of the D- and Q-statistic based approach, which requires further data processing by improving degradation detection sensitivity.

4.3. Experimental evaluation

Aiming to evaluate the proposed CNN-DLSTM model, the first 20,000 data samples of bearing 3 under normal operation condition on the x-axis are taken as the training set and those on the y-axis as the testing set. The following methods are applied for signal reconstruction and the reconstruction errors are finally compared.

(a) DAEN (Deep Auto-encoder network): An AE with 4 hidden layers using dropout on extracted features from the raw sensor data;
(b) DRNN (Deep RNN): An RNN with 4 hidden layers on raw sensor data;
(c) LSTM: An LSTM with one hidden layer using dropout on raw sensor data;
(d) DLSTM: An LSTM with 4 hidden layers using dropout on raw sensor data;

Three recurrent models, including DRNN, LSTM and DLSTM can be immediately fed into the sequential data, thus feature extraction is undesired. Overall 106,800 time steps with each a one-dimensional data vector are included in the time-series input. For DRNN and LSTM, four layers with sizes of [890, 200, 400, 600] and one-layer with a size of 890 are constructed; two fully connected layers with sizes of [600, 890] and linear regression are successively used subsequently. In DLSTM, the layers are replaced by the DRNN layers, keeping the rest parts unchanged with DRNN model. Since DAEN cannot process sequence directly, feature extraction is required and 13 measures shown in Table 1 are extracted from the data samples of bearing 3 in advance. Then, the signal can be transformed into a 13-dimensional vector, and is entered in the 4-layer DAEN with sizes of [890, 200, 400, 600], two fully connected layers and regression model successively with the settings kept unchanged.

To quantify the performances of the state-of-the-art models, two measures including MAE and Root Mean Squared Error (RMSE) are used. MAE is the average of the absolute values of the deviations of all sample values from their mean value. RMSE is the square root of the ratio of the square of the deviation between the predicted value with the true value and the total sample number. The equations are shown as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\overline{y_i} - y_i| \tag{17}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\overline{y_i} - y_i)^2} \tag{18}$$

where y_i and $\overline{y_i}$ are target sensor data and actual output of model. Part of the fitting curves and MAE of the proposed CNN-DLSTM method on the dataset reconstruction are shown in Figure 6 and the results of both MAE and RMSE of the methods are illustrated in Table 2.

From Table 2, observations and conclusions can be made that:

- Several recurrent models outperform DAEN to different degrees, owing to their superior ability in processing time-series data. Recurrent models show complicated dynamics due to the various feedforward and feedback connections. The connections endow these models with added merits over DAEN for dealing with sequential and dynamic tasks. A recurrent model with a small structure possibly balances with a complex category of feedforward structure.
- Among recurrent models, DLSTM containing multiple hidden recurrent layers outperforms its corresponding common LSTM containing one hidden layer. Deep models can extract more advanced and informative features owing to the multiple hidden layers as far as the experiment concerned.
- For different recurrent units, DLSTM performs better than DRNN. The reason may lie in the fact that the input gates, output gates and forget gates used in LSTM enable it to capture long-term dependency better than RNN.

Table 1. List of extracted features.

Domain	Features	Expression
Time	RMS	$z_{\text{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}$
	Variance	$z_{\text{var}} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2$
	Maximum	$z_{\max} = \max(z)$
	Minimum	$z_{\min} = \min(z)$
	Fourth central moment	$z_{m4} = E[(x - E(x))^3]$
	Peak-to-Peak	$z_{p-p} = \max(z) - \min(z)$
	Skewness	$z_{\text{skew}} = E\left[\left(\frac{z-\mu}{\sigma}\right)^3\right]$
	Kurtosis	$z_{\text{kurt}} = E\left[\left(\frac{z-\mu}{\sigma}\right)^4\right]$
Frequency	Mean frequency	$f_{\text{mnf}} = \frac{1}{n}\sum_{i=1}^{n} f_i$
	Spectral skewness	$f_{\text{skew}} = \sum_{i=1} k\left(\frac{f_i - \bar{f}}{\sigma}\right)^3 S(f_i)$
	Spectral kurtosis	$f_{\text{kurt}} = \sum_{i=1} k\left(\frac{f_i - \bar{f}}{\sigma}\right)^4 S(f_i)$
Time–frequency	Information entropy	$H_U = -\sum_{i=1}^{n} p_i \log p_i$
	Wavelet energy	$E_{WT} = \sum_{i=1}^{N} \omega t_{\emptyset}^2(i)/N$

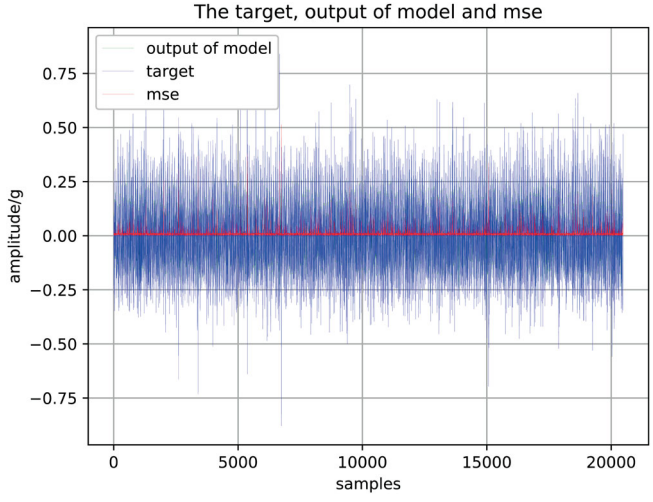

Figure 6. Part of the reconstruction fitting curves and the mean square error of the proposed CNN-DLSTM on the x-axis signal on bearing 3.

- The results show that the proposed CNN-DLSTM model outperforms the other state-of-the-art models. Compared to the most competitive model, DLSTM, CNN-DLSTM applies CNN to process the raw sensor data and constructs deep LSTM models after CNN. It has demonstrated that CNN plays the role of extracting features and filtering noises effectively to extract more advanced local features compared to the raw sequential data to facilitate the DLSTM models to better extract temporal features. Furthermore, CNN lowers the size of the input data, thus the

Table 2. Reconstruction error of each model on x-axis signal of bearing 3.

Methods	MAE (10^{-4})	RMSE (10^{-4})
DAEN	5.12	6.81
DRNN	2.48	3.31
LSTM	3.29	4.16
DLSTM	2.37	2.62
CNN-DLSTM	1.87	2.34

concise data can be more conveniently processed by subsequent layers. The results present the effectiveness and superior generalisation ability of the CNN-DLSTM model.

5. Conclusion

In this paper, a performance degradation assessment approach of rolling bearing is developed and investigated using a data-driven method. The proposed CNN-DLSTM and several competitive methods are applied to reach the target. In CNN-DLSTM, CNN is firstly used for local feature extraction over the collected time-series sensor data. A DLSTM model is then applied to extract the temporal features. More advanced and abstract feature representations are learned through each hidden layer of the DLSTM. Finally, the D- and Q-statistic are developed to generate the H-statistic for indicating the degree of the performance degradation of rolling bearing and KDE is proposed to set the threshold for warning line. Using the proposed method in this paper, traditional feature engineering and expert knowledge are not required. The conclusion is that the performance of CNN-DLSTM outperforms that of DAEN, DRNN and DLSTM as far as the rolling bearing vibration signal reconstruction task is concerned. Therefore, the proposed method can extract informative and robust features using sensor data for machine health monitoring.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was financially supported by the National Natural Science Foundation of China [grant numbers 51875345, 51475290].

References

Bengio, Y. 2009. "Learning Deep Architectures for AI." *Foundations & Trends® in Machine Learning* 2 (1): 1–127. doi:10.1561/2200000006.

Bhuiyan, M. S. H., I. A. Choudhury, and M. Dahari. 2014. "Monitoring the Tool Wear, Surface Roughness and Chip Formation Occurrences Using Multiple Sensors in Turning." *Journal of Manufacturing Systems* 33 (4): 476–487. doi:10.1016/j.jmsy.2014.04.005.

Chen, Q., U. Kruger, and A. T. Y. Leung. 2004. "Regularised Kernel Density Estimation for Clustered Process Data." *Control Engineering Practice* 12 (3): 267–274. doi:10.1016/S0967-0661(03)00083-2.

Collobert, R., and J. Weston. 2008. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning." In *Proceedings of the 25th International Conference on Machine Learning*, 160–167. Helsinki, Finland: ACM. doi:10.1145/1390156.1390177.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. Cambridge, MA: The MIT Press.

Hessainia, Z., A. Belbah, A. Yallese M, T. Mabrouki, and J.-F. Rigal. 2013. "On the Prediction of Surface Roughness in the Hard Turning Based on Cutting Parameters and Tool Vibrations." *Measurement* 46 (5): 1671–1681. doi:10.1016/j.measurement.2012.12.016.

Hochreiter, S., and J. Schmidhuber. 1997. *Long Short-Term Memory*. Berlin: Springer-Verlag.

Kirubarajan, T. 2002. "Physically Based Diagnosis and Prognosis of Cracked Rotor Shafts." *Proceedings of SPIE – The International Society for Optical Engineering*, 4733. doi:10.1117/12.475502.

LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. D. Jackel. 1990. "Handwritten Digit Recognition with a Backpropagation Neural Network." *Advances in Neural Information Processing Systems* 2 (2): 396–404.

Lee, J., H. Qiu, G. Yu, and J. Lin. 2007. "Data from: Bearing Data Set, IMS, Univ. Cincinnati" (dataset). NASA Ames Prognostics Data Repository. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

Lee, J. M., C. K. Yoo, and I. B. Lee. 2004. "Statistical Process Monitoring with Independent Component Analysis." *Journal of Process Control* 14 (5): 467–485. doi:10.1016/j.jprocont.2003.09.004.

Li, Y., T. R. Kurfess, and S. Y. Liang. 2000. "Stochastic Prognostics for Rolling Element Bearings." *Mechanical Systems & Signal Processing* 14 (5): 747–762. doi:10.1006/mssp.2000.1301.

Li, C., R. V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez. 2016. "Gearbox Fault Diagnosis Based on Deep Random Forest Fusion of Acoustic and Vibratory Signals." *Mechanical Systems & Signal Processing* 76-77: 283–293. doi:10.1016/j.ymssp.2016.02.007.

Lu, C., Y. Wang Z, L. Qin W, and J. Ma. 2017. "Fault Diagnosis of Rotary Machinery Components Using a Stacked Denoising Autoencoder-based Health State Identification." *Signal Processing* 130 (C): 377–388. doi:10.1016/j.sigpro.2016.07.028.

Ma, M., X. Chen, S. Wang, Y. Liu, and W. Li. 2016. "Bearing Degradation Assessment Based on Weibull Distribution and Deep Belief Network." In *2016 International Symposium on Flexible Automation*, 382–385. Cleveland, OH: IEEE. doi:10.1109/ISFA.2016.7790193.

Martin, E. B., and A. J. Morris. 1996. "Non-parametric Confidence Bounds for Process Performance Monitoring Charts." *Journal of Process Control* 6 (6): 349–358. doi:10.1016/0959-1524(96)00010-8.

Nair, V., and G. E. Hinton. 2010. "Rectified Linear Units Improve Restricted Boltzmann Machines." *Proceedings of the 27th International Conference on Machine Learning*, Haifa, 21 June 2010, 807–814.

Negrichi, K., M. D. Mascolo, and J. M. Flaus. 2017. "A Model Based Approach to Assess the Performance of Production Systems in Degraded Mode." *International Journal of Production Research* 55 (8): 2288–2303. doi:10.1080/00207543.2016.1237788.

Perlin, H. A., and H. S. Lopes. 2015. "Extracting Human Attributes Using a Convolutional Neural Network Approach." *Pattern Recognition Letter* 68 (P2): 250–259. doi:10.1016/j.patrec.2015.07.012.

Plaza, E. G., and P. J. N. López. 2017. "Surface Roughness Monitoring by Singular Spectrum Analysis of Vibration Signals." *Mechanical Systems and Signal Process* 84 (A): 516–530. doi:10.1016/j.ymssp.2016.06.039.

Plaza, E. G., and P. J. N. López. 2018. "Application of the Wavelet Packet Transform to Vibration Signals for Surface Roughness Monitoring in CNC Turning Operations." *Mechanical Systems & Signal Processing* 98: 902–919. doi:10.1016/j.ymssp.2017.05.028.

Rehorn, A. G., E. Sejdić, and J. Jiang. 2006. "Fault Diagnosis in Machine Tools Using Selective Regional Correlation." *Mechanical Systems and Signal Process* 20 (5): 1221–1238. doi:10.1016/j.ymssp.2005.01.010.

Rui, Z., Y. Ruqiang, W. Jinjiang, and M. Kezhi. 2017. "Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks." *Sensors* 17 (2): 273. doi:10.3390/s17020273.

Rui, Z., Y. Ruqiang, C. Zhenghua, M. Kezhi, W. Peng, and R. X. Gao. 2019. "Deep Learning and its Applications to Machine Health Monitoring." *Mechanical Systems and Signal Processing* 115: 213–237. doi:10.1016/j.ymssp.2018.05.050.

Shao, S., W. Sun, P. Wang, R. X. Gao, and R. Yan. 2016. "Learning Features from Vibration Signals for Induction Motor Fault Diagnosis." In *2016 International Symposium on Flexible Automation*, 71–76. Cleveland, OH: IEEE. doi:10.1109/ISFA.2016.7790138.

Shi, D., and N. N. Gindy. 2007. "Tool Wear Predictive Model Based on Least Squares Support Vector Machines." *Mechanical Systems and Signal Process* 21 (4): 1799–1814. doi:10.1016/j.ymssp.2006.07.016.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15 (1): 1929–1958. doi:10.1214/12-AOS1000.

Sun, W., S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen. 2016. "A Sparse Auto-Encoder-based Deep Neural Network Approach for Induction Motor Faults Classification." *Measurement* 89 (ISFA): 171–178. doi:10.1016/j.measurement.2016.04.007.

Tao, S., T. Zhang, J. Yang, X. Wang, and W. Lu. 2015. "Bearing Fault Diagnosis Method Based on Stacked Autoencoder and Softmax Regression." In *2015 34th Chinese Control Conference*, 6331–6335. Hangzhou, China: IEEE. doi:10.1109/ChiCC.2015.7260634.

Wang, H., S. Ding, D. S. Wu, Y. T. Zhang, and S. L. Yang. 2018. "Smart Connected Electronic Gastroscope System for Gastric Cancer Screening Using Multi-Column Convolutional Neural Networks." *International Journal of Production Research*, doi:10.1080/00207543.2018.1464232.

Wang, H., S. To, and C. Y. Chan. 2013. "Investigation on the Influence of Tool-tip Vibration on Surface Roughness and its Representative Measurement in Ultraprecision Diamond Turning." *International Journal of Machine Tools & Manufacturing* 69 (3): 20–29. doi:10.1016/j.ijmachtools.2013.02.006.

Westerhuis, J. A., S. P. Gurden, and A. K. Smilde. 2000. "Generalized Contribution Plots in Multivariate Statistical Process Monitoring." *Chemometrics & Intelligent Laboratory Systems* 51 (1): 95–114. doi:10.1016/S0169-7439(00)00062-9.

Williams, J. D., W. H. Woodall, J. B. Birch, and J. H. Sullivan. 2006. "Distribution of Hotelling's T^2 Statistic Based on the Successive Differences Estimator." *Journal of Quality Technology* 38 (3): 217–229. doi:10.1080/00224065.2006.11918611.

Yildirim, Özal. 2018. "A Novel Wavelet Sequences Based on Deep Bidirectional LSTM Network Model for ECG Signal Classification." *Computers in Biology & Medicine* 96: 189. doi:10.1016/j.compbiomed.2018.03.016.

Yu, J. 2012. "Local and Nonlocal Preserving Projection for Bearing Defect Classification and Performance Assessment." *IEEE Transactions on Industrial Electronics* 59 (5): 2363–2376. doi:10.1109/TIE.2011.2167893.

Yuan, M., Y. Wu, and L. Lin. 2016. "Fault Diagnosis and Remaining Useful Life Estimation of Aero Engine Using LSTM Neural Network." In *2016 IEEE/CSAA International Conference on Aircraft Utility Systems*, 135–140. Beijing, China: IEEE. doi:10.1109/AUS.2016.7748035.

Yue, H. H. and Qin, S. J. 2001. "Reconstruction-based Fault Identification Using a Combined Index." *Industrial and Engineering Chemistry Research* 40 (20): 4403–4414. doi:10.1021/ie000141 + .

Zhu, H., T. Rui, X. Wang, Y. Zhou, and H. Fang. 2016. "Fault Diagnosis of Hydraulic Pump Based on Stacked Autoencoders." In *2015 12th IEEE International Conference on Electronic Measurement and Instruments*, 58–62. Qingdao, China: IEEE. doi:10.1109/ICEMI.2015.7494195.