Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 127 (2018) 123–132

Procedia Computer Science

www.elsevier.com/locate/procedia

The First International Conference On Intelligent Computing in Data Sciences

Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network

Ahmed Zakariae Hinchi*, Mohamed Tkiouat

Laboratory for Applied Mathematics (LERMA),Mohammadia School of engineering, Mohammed V University, Rabat, Morocco

Abstract

The rolling element bearing is the leading cause of failures in rotating machinery; on that account, the accurate prediction of its remaining useful life (RUL) using sensor data is an important challenge to improve the reliability and decrease the maintenance costs. Classical data-driven approaches rely on manually extracted features from raw sensor data followed by an estimation of a health indicator, the degradation states and the prediction of RUL using a failure threshold.
Based on the recent success of deep neural networks in various artificial intelligence domains, we propose an end-to-end deep framework for RUL estimation based on convolutional and long-short-term memory (LSTM) recurrent units. First the neural network extracts the local features directly from sensor data using the convolutional layer, then an LSTM layer is introduced to capture the degradation process, finally the RUL is estimated using the LSTM outputs and the prediction time value. Experiments are conducted on the ball bearing data provided by FEMTO-ST Institute. The results demonstrate the efficiency of our approach.

Keywords: Rolling element bearing;Condition-based maintenance;Prognostics;Convolutional neural network;Long-short-term memory network

1. Introduction

In the increasingly complex and automated modern industry, the economic cost of a failure became increasingly high; faulty components can affect the reliability of other associated assets in the system; consequently, adopting effective maintenance strategies is crucial for the enhancement of the overall reliability and profitability.

Recently, the development of modern sensor technology and condition monitoring systems led to the adoption of condition-based maintenance (CBM) [1]. CBM is conducted by recommending maintenance actions in real time based

*Hinchi Ahmed Zakariae. Tel: 00212 682308539
E-mail address: ahm.zak.hin@gmail.com

10.1016/j.procs.2018.01.106

on the collected sensor data. The most challenging CBM activity is prognostics [2] defined as the prediction of future conditions and the residual life of the system.

There are mainly three paradigms for prognostics: model-based approaches [3], data-driven approaches [4] and hybrid approaches [5]. The application of general model-based or hybrid prognostic approaches relies on the understanding of the system's physics-of-failure which is difficult in a complex engineered systems. The data-driven approaches are mainly based on sensor data with less requirement on knowing the inherent system failure mechanisms. This led to a large increase in their popularity in recent years.

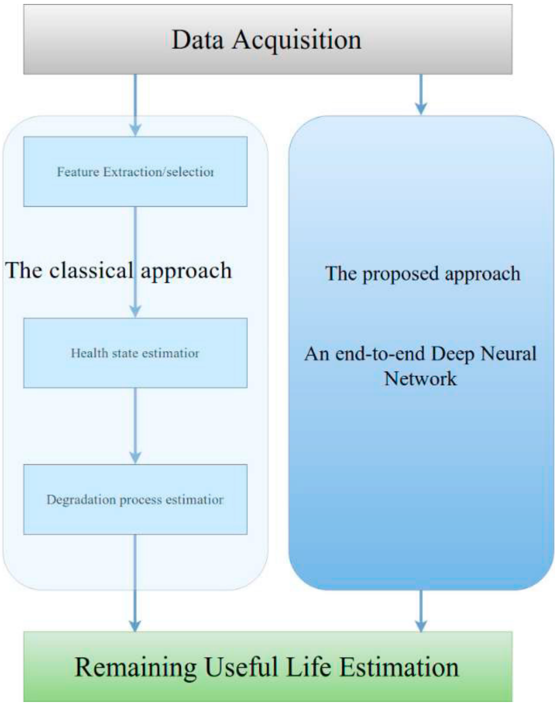

Fig. 1. The RUL estimation pipeline

The rolling element bearing is the most common component in rotating machinery; it accounts for 45–55% of these equipment failures [6], The corresponding vibration signals can be collected in real time at a relatively low cost. As a result, a wide range of studies on data-driven methods for rolling element bearing prognostics exists in the literature ([7], [8], [9], [10], [11], [12], [13], [14], [15]).

Most of these methods are composed of three steps (figure 1):

- A feature extraction and selection step from the raw vibration data;
- A health state estimation step typically done by combining the extracted features into a health indicator;
- A degradation process estimation step generally realized by extrapolating the health indicator until it reaches a failure threshold.

This classical pipeline present drawbacks which are required to be solved:

- The identification of the suitable features for prognostics requires considerable domain expertise and a great deal of human labor. These features can exhibit different degradation signatures at different stages of the degradation process. Some features don't change their values until the bearing is very close to the end of its useful life; others are not monotonic or very sensitive to the measurement noise;

- Most of the work in heath state estimation aims at constructing a one dimensional health indicator. This assumption of dimensionality fails at presenting the state of a bearing that can exhibit multiple failure modes [16];
- These methods typically require setting a failure threshold. A static threshold can be non-convenient, and an adaptive one is difficult to estimate.

In order to deal with the aforementioned shortcomings and motivated by the recent success of deep learning methods in speech recognition [17], image recognition [18] ,and diagnostics [19], the present work proposes a deep neural network for rolling elements bearing prognostics trained end-to-end (figure 2).To our knowledge, this is the first approach to predict the remaining useful life directly from high frequency sensor data. By combining a convolutional layer for feature extraction, with an LSTM for encoding the temporal representation, the neural network is automatically able to encode the degradation information and predict the residual life of the bearings.

The rest of the paper is organized as follows: the detailed construction procedure of the proposed model is presented in Section 2, then our approach is evaluated on run-to-failure bearing vibration data in section 3, finally concluding remarks and venues for future research are outlined in Section 4.

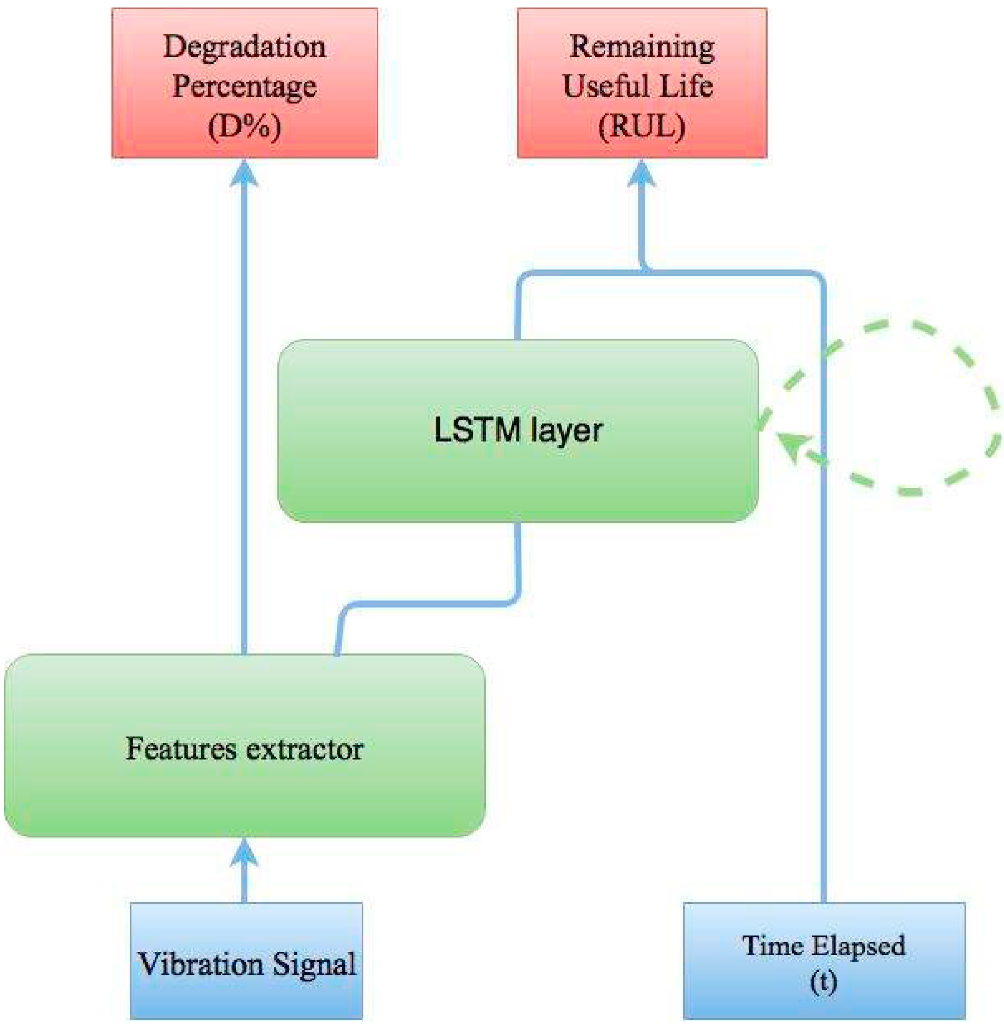

Fig. 2. The simplified architecture of our approach

2. Methodology

Let $rul_{t_i}^{(b)}$ the remaining useful life of the bearing (b) at prediction time t_i, and $EOL^{(b)}$ its time of end of life.

$$\mathrm{rul}_{t_i}^{(b)} = \mathrm{EOL}^{(b)} - t_i \,, t_i < \mathrm{EOL}^{(b)}$$

Our objective is to determine the function f that estimates the RUL of a bearing given the past in-process vibration data $x_{t_{1:i}}{}^{(b)}$ and the prediction time t_i:

$$\widehat{\mathrm{rul}}_{t_i}^{(b)} = f\left(x_{t_{1:i}}{}^{(b)}, t_i\right)$$

We propose a deep neural network to model the function f. The model predicts the RUL directly from the vibration data by stacking a convolutional layer, a global average pooling layer, and an LSTM layer. To regularize the model, we use batch normalization after each layer, and dropout for the input and the recurrent units of the LSTM layer; furthermore, we add an auxiliary loss function directly to the output of the global average pooling layer. The model architecture is shown in details in figure 3. The following sections will describe the details of each layer and the training process.

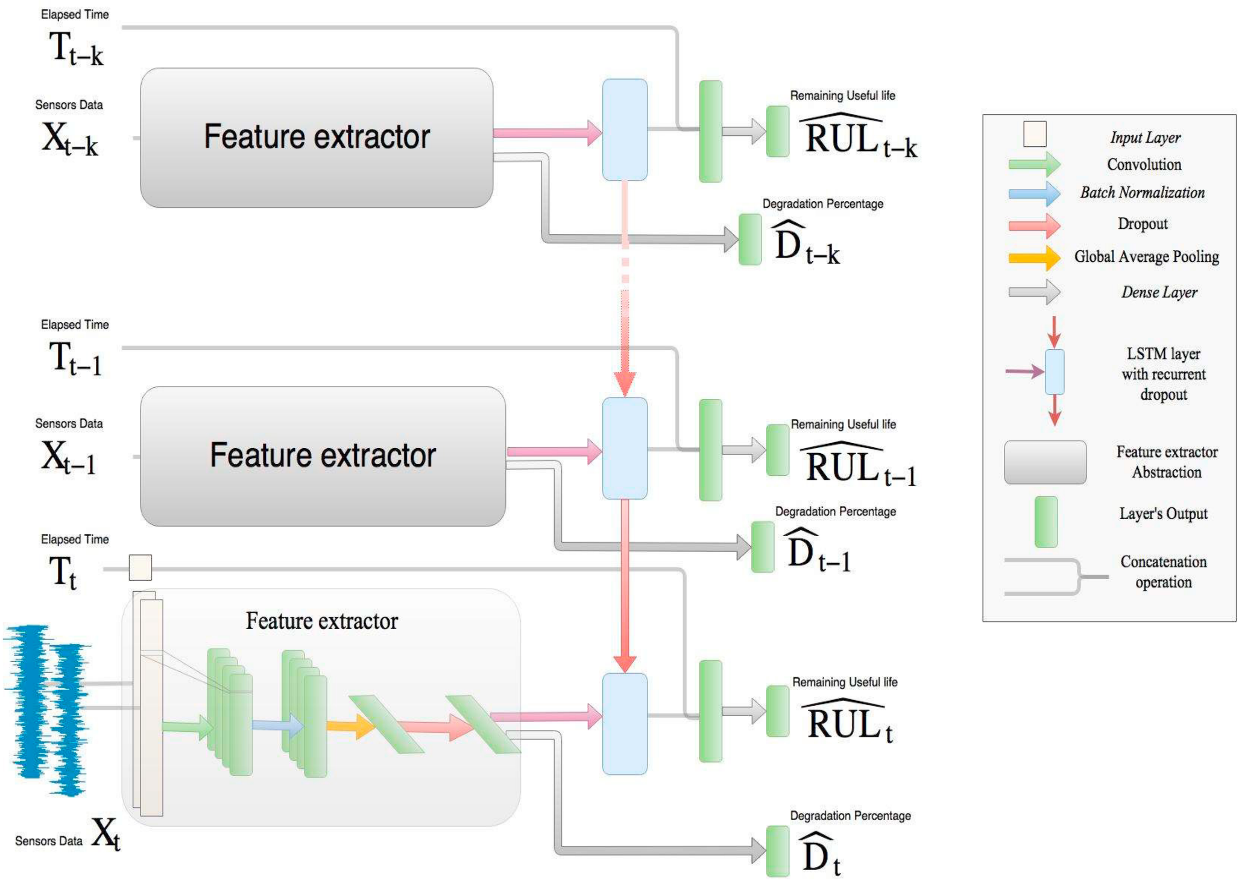

Fig. 3. The architecture of the deep neural network

2.1. The Convolutional layer

The convolutional layer [20] extracts a high level abstraction of the input data called a feature map from the input signal through a convolution operation of the signal with a filter. Each filter uses the same kernel to extract the local features of the input local region.

Formally, in a one dimensional signal with a single input channel, the convolutional layer calculates the results of a convolution operation with input X, a bank of filters K, and bias b followed by a nonlinear activation function σ:

$$(X*K)_i = \sigma\left(\sum_{m=0}^{d_k-1} K_m \cdot X_{i+m} + b\right)$$

Where d_k denotes the length of the kernel, and $\sigma = \max(0, x)$ a rectifier activation function as set in our model [21].

2.2. The global average pooling layer

Following the convolutional layer, the global average pooling layer aims at compressing the feature space [22].This layer calculates the average of each feature map, which improves the regularization and makes the network more robust to small shifts and distortions in the vibration signal.

2.3. The LSTM layer

The LSTM layer [23] is introduced to deal with the sequential nature of the data. Typically, a recurrent neural network RNN use of a recurrent connection where the activation of a neuron is fed back to itself; however, the inability of RNNs to learn long Term Dependency is explored in depth by [24].

As a solution to this problem, the LSTM layer offers a more rich internal state by using a cell state C_t in addition to the hidden state h_t and various gates: first, the forget gate helps the layer learn which pieces of the long-term memory to continue remembering and which to ignore using the new input:

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

Next, the input gate computes the information that can be learned from the input, and learn which parts of it are actually worth using and saving:

$$\xi_t = \mathrm{Tanh}\left(W_\xi\left[h_{t-1}, x_t\right] + b_\xi\right)$$

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

The next operation is updating the long-term memory presented by the cell state:

$$C_t = f_t C_{t-1} + i_t \xi_t$$

Finally the hidden state is updated from the input, the cell state, and the previous hidden state using the output gate:

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

$$h_t = o_t.\mathrm{Tanh}\left(C_t\right)$$

We note $(W_f, W_\xi, W_i, W_o, b_f, b_\xi, b_i, b_o)$ the weight matrices and biases corresponding to the various gates, and σ the sigmoid function.

In our model this layer is followed by a neuron that outputs the predicted RUL given the output of the LSTM hidden state and the prediction time t_i:

$$\widehat{RUL} = \sigma_s\left(W_r\left[h_t, t_i\right] + b_r\right)$$

Where $\sigma_s = ln(1 + e^x)$ the activation function, and (W_r, b_r) the weight matrix and the bias respectively.

2.4. Batch Normalization

As proposed in [25], Batch Normalization (BN) can help speed up the training and improve the performance of a deep neural network by controlling the input distribution across the layers. The transformation of the BN layer is described as follows:

$$\hat{x}^{(i)} = \frac{x^{(i)} - E\left[x^{(i)}\right]}{\sqrt{Var\left[x^{(i)}\right]}}$$

$$y^{(i)} = \gamma^{(i)}\hat{x}^{(i)} + \beta^{(i)}$$

Where $y^{(i)}$ is the output of one neuron response, $\gamma^{(i)}$ and $\beta^{(i)}$ are the scale and shift parameters to be learned respectively, and $x = (x(1), \ldots, x(p))$ the layer's input.

The first step is to standardize feature in each dimension independently, which helps to accelerate convergence. Then, $\gamma^{(i)}$ and $\beta^{(i)}$ are used to scale and shift each normalized feature to ensure the transformation inserted in the network can represent the identity transform.

2.5. Dropout

Dropout [26] is a recently introduced algorithm for training neural networks by randomly dropping units during training to prevent their co-adaptation. This helps the model to produce robust units that do not depend on the details of the activation of other individual units. In this work, as suggested by Gal in [27], we use the same dropout mask at each time step for both the inputs and the recurrent units of the LSTM.

2.6. The training of the model

We use the mean absolute percentage error MAPE to train our model:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{RUL_i - \widehat{RUL}_i}{RUL_i}\right|$$

Although the MAPE is biased towards lower values [28], this is beneficial in this application, since a low forecast of the RUL can mean an unnecessary maintenance operation, while a high forecast may cause a catastrophic failure.

The use of auxiliary loss functions has been investigated in [29] and [30]. In this work, we use an auxiliary loss function $\mathcal{L}_{aux}$ to constrain the low level layers output for feature extraction purposes; this improves the regularization of our network. $\mathcal{L}_{aux}$ minimizes the mean absolute error between the predicted and real degradation percentage:

$$\mathcal{L}_{\text{aux}} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \left| \text{D}_{\text{i}} - \widehat{\text{D}}_{\text{i}} \right|$$

Where $D_{t_i}^{(b)} = \frac{t_i}{EOL^{(b)}}$ is the degradation percentage at time t_i of bearing(b), and $\widehat{D_{t_i}^{(b)}} = \sigma_{si}(W_{si} gap + b_{si})$ is the predicted degradation percentage, $(gap, W_{si}, b_{si}, \sigma_{si})$ are respectively the outputs of the global average pooling layer, the corresponding weight matrix, the bias, and the sigmoid activation function.

During training, $\mathcal{L}_{aux}$ is added to the total loss of the network with a discount weight $\gamma < 1$. At inference time, The model is trained end-to-end using the specified loss functions, particularly, we use the backpropagation through time (BPTT) algorithm to compute the gradients, and the optimization is conducted with ADAM [31].

3. Experimental validation

In order to validate our approach, the experimental data-set from PRONOSTIA [32] was collected by conducting accelerated degradation tests of bearings. Two accelerometers were horizontally and vertically mounted on the bearing, the corresponding vibration values were sampled every 10 s, and the duration of the sampling lasted 0.1 s with a sampling frequency of 25.6 kHz. Seventeen different bearing were run to failure in three different operating conditions; the faults were not seeded therefore different failure mode can be present. Table 1 shows the different bearing and their experimental conditions.

Table 1: The experimental data set.

Dataset	Operating Conditions		
	1800 rpm,4000 N	1650 rpm,4200 N	1500 rpm,5000 N
Training Set	Bearing1_1 Bearing1_2	Bearing2_1 Bearing2_2	Bearing3_1 Bearing3_2
Testing set	Bearing1_3 Bearing1_4 Bearing1_5 Bearing1_6 Bearing1_7	Bearing2_3 Bearing2_4 Bearing2_5 Bearing2_6 Bearing2_7	Bearing3_3

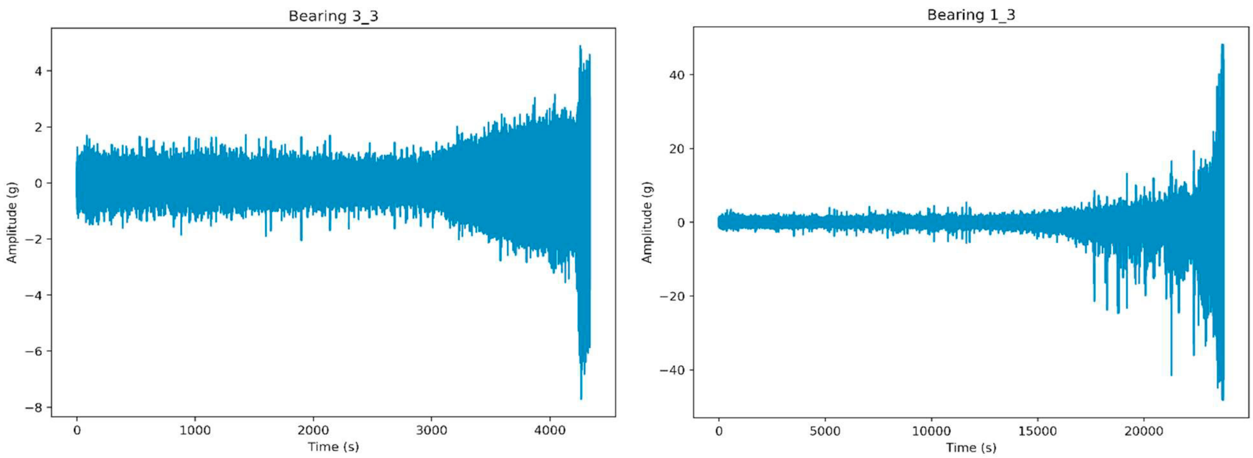

Fig. 4. (a) Vibration signal of Bearing 3_3; (b) Vibration signal of Bearing 1_3

Figure 4 shows the radial vibration signal of two tested bearing during their whole lifetime. It can be observed that the amplitudes of the vibration signals progress differently over time, hence the complexity of RUL estimation.

We construct a training set from the first six bearing, the training set contains the historical vibration data for every bearing until every prediction time t_i, the corresponding $\mathrm{rul}_{t_i}^{(b)}$ and the corresponding degradation percentage at each time step $\left[D_{t_k}^{(b)}, t_k \leq t_i\right]$ .Each time step contains 2560 axial vibration points and 2560 radial ones.

To train our model, at each training step, we feed the data related to a random prediction time t_i for each of the bearing in the training set; this serves to limit the bias towards bearings with a long lifetime. The axial and radial vibrations points are fed into a 2 channel convolutional layer, followed by a global average pooling layer, an LSTM layer and a prediction neuron. The network is trained end-to-end as shown in section 2.

We fix the parameters of the ADAM optimizer at their default values. Furthermore the receptive field of each neuron in the global average pooling layer is fixed to the value of one period of the input signal. We chose the rest of the hyper-parameters of the model by cross validation through the Sequential model-based hyper-parameter optimization algorithm using the HyperOpt library [33].

The model is implemented using the Tensorflow [34] library; the training and RUL prediction are run on an Ubuntu Linux machine with an Nvidia GTX 1070 GPU.

In the next step, we evaluated all the testing bearings. To benchmark our approach we contrast the obtained results with that of the Center for Advanced Life Cycle Engineering (CALCE), the winner of the 2012 PHM data challenge competition [7]. A higher score is achieved in [15], nevertheless our objective is to prove the viability of end-to-end prognostics.

First we calculate the percentage error of RUL prediction:

$$\mathrm{ER}_i = \frac{\mathrm{rul}_{t_i}^{(b)} - \widehat{\mathrm{rul}_{t_i}^{(b)}}}{\mathrm{rul}_{t_i}^{(b)}}$$

Then the corresponding score for each bearing:

$$\mathrm{AR}_i = \begin{cases} e^{-\ln(0.5)*(\frac{\mathrm{ER}_i}{5})}, & \mathrm{ER}_i < 0 \\ e^{\ln(0.5)*(\frac{\mathrm{ER}_i}{20})}, & \mathrm{ER}_i \geq 0 \end{cases}$$

The overall score is the mean of the scores calculated from the testing bearings.

$$\mathrm{score} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{AR}_i$$

The aforementioned score function does not considers underestimates (positive percentage error) and overestimates (negative percentage error) in the same manner, this is in line with the risk based scheduling decisions encountered in practice. Table 2 compares our methodology to that of [7] and the results shows the superiority of our approach.

Table 2 : The RUL predictions results

Bearing	Prediction Time(s)	The actual RUL(s)	ER_i of [7] (%)	ER_i of our approach (%)
Bearing1_3	18010	5730	37	54.73
Bearing1_4	11380	2900	80	38.69
Bearing1_5	23010	1610	9	-99.4

Bearing1_6	23010	1460	-5	-120.07
Bearing1_7	15010	7570	-2	70.65
Bearing2_3	12010	7530	64	75.53
Bearing2_4	6110	1390	10	19.81
Bearing2_5	20010	3090	-440	8.2
Bearing2_6	5710	1290	49	17.87
Bearing2_7	1710	580	-317	1.69
Bearing3_3	3510	820	90	2.93
Score			0.3066	**0.3828**

4. Conclusion

In this work we propose an end-to-end model for rolling element bearing prognostics. We showed that the model can automatically learn the degradation process from the historical vibration data, and predict the corresponding remaining useful life.

As prognostics is an inherently uncertain process [35], future work must identify the different uncertainties involved and include uncertainty estimation in RUL prediction. Furthermore, to test the generality of the framework, future experiments should test the architecture on the prognostics of different types of assets.

References

[1] Ellis, Byron A., and A. Byron. "Condition based maintenance." The Jethro Project 10 (2008): 1-5.

[2] Elattar, Hatem M., Hamdy K. Elminir, and A. M. Riad. "Prognostics: a literature review." Complex & Intelligent Systems 2.2 (2016): 125-154.

[3] Cubillo, Adrian, Suresh Perinpanayagam, and Manuel Esperon-Miguez. "A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery." Advances in Mechanical Engineering 8.8 (2016): 1687814016664660.

[4] Tsui, Kwok L., et al. "Prognostics and health management: A review on data driven approaches." Mathematical Problems in Engineering 2015 (2015).

[5] Yang, Wen-An, et al. "A hybrid prognostic approach for remaining useful life prediction of lithium-ion batteries." Shock and Vibration 2016 (2016).

[6] Rai, Akhand, and S. H. Upadhyay. "A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings." Tribology International 96 (2016): 289-306.

[7] Sutrisno, Edwin, et al. "Estimation of remaining useful life of ball bearings using data driven methodologies." Prognostics and Health Management (PHM), 2012 IEEE Conference on. IEEE, 2012.

[8] Ali, Jaouher Ben, et al. "Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network." Mechanical Systems and Signal Processing 56 (2015): 150-172.

[9] Wang, Xiao-lin, et al. "A SVR-Based Remaining Life Prediction for Rolling Element Bearings." Journal of Failure Analysis and Prevention 15.4 (2015): 548-554.

[10] Wu, Bo, Wei Li, and Ming-quan Qiu. "Remaining Useful Life Prediction of Bearing with Vibration Signals Based on a Novel Indicator." Shock and Vibration 2017 (2017).

[11] Wang, Yu, et al. "A two-stage data-driven-based prognostic approach for bearing degradation problem." IEEE Transactions on Industrial Informatics 12.3 (2016): 924-932.

[12] Wang, Biao, et al. "An improved fusion prognostics method for remaining useful life prediction of bearings." Prognostics and Health Management (ICPHM), 2017 IEEE International Conference on. IEEE, 2017: 18–24.

[13] Rai, Akhand, and Sanjay H. Upadhyay. "Intelligent bearing performance degradation assessment and remaining useful life prediction based on self-organising map and support vector regression." Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science (2017): 0954406217700180.

[14] Guo, Liang, et al. "A recurrent neural network based health indicator for remaining useful life prediction of bearings." Neurocomputing 240 (2017): 98-109.

[15] Ren, Likun, Weimin Lv, and Shiwei Jiang. "Machine prognostics based on sparse representation model." Journal of Intelligent Manufacturing (2015): 1-9.

[16] Wang, Dong, and Kwok-Leung Tsui. "Theoretical investigation of the upper and lower bounds of a generalized dimensionless bearing health indicator." Mechanical Systems and Signal Processing 98 (2018): 890-901.

[17] Amodei, Dario, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." International Conference on Machine Learning. 2016: 173–182.

[18] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770–778.
[19] Zhao, Rui, et al. "Deep Learning and Its Applications to Machine Health Monitoring: A Survey." arXiv preprint arXiv:1612.07640 (2016).
[20] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361.10 (1995): 1995.
[21] Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton. "Improving deep neural networks for LVCSR using rectified linear units and dropout." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.
[22] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).
[23] Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems (2017).
[24] Hochreiter, Sepp, and Jürgen Schmidhuber. "LSTM can solve hard long time lag problems." Advances in neural information processing systems. 1997
[25] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International Conference on Machine Learning. 2015.
[26] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of machine learning research 15.1 (2014): 1929-1958.
[27] Gal, Yarin, and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks." Advances in neural information processing systems. 2016.
[28] Tofallis, Chris. "A better measure of relative prediction accuracy for model selection and model estimation." Journal of the Operational Research Society 66.8 (2015): 1352-1362
[29] Guo, Yong, et al. "The Shallow End: Empowering Shallower Deep-Convolutional Networks through Auxiliary Outputs." arXiv preprint arXiv:1611.01773 (2016).
[30] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[31] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[32] Nectoux, Patrick, et al. "PRONOSTIA: An experimental platform for bearings accelerated degradation tests." IEEE International Conference on Prognostics and Health Management, PHM'12.. IEEE Catalog Number: CPF12PHM-CDR, 2012.
[33] Bergstra, James, et al. "Hyperopt: a python library for model selection and hyperparameter optimization." Computational Science & Discovery 8.1 (2015): 014008.
[34] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).
[35] Sankararaman, Shankar. "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction." Mechanical Systems and Signal Processing 52 (2015): 228-247.