# A Novel Bearing Health Prognostic Method Based on Time-frequency Analysis and LSTM

Jian Duan
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
duanjian121@outlook.com

Tielin Shi
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
tlshi@hust.edu.cn

Hongdi Zhou
School of Mechanical Engineering
Hubei University of Technology
Wuhan, China
zh_hongdi@163.com

Jie Duan
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
jduan@hust.edu.cn

Jianping Xuan
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
jpxuan@hust.edu.cn

Jun Zhang
Foxconn Industrial Internet Co., Ltd
Shenzhen, China
461919043@qq.com

***Abstract*—Bearing plays an enormous role in modern factories. It is of great value to monitor the bearing health conditions. With the rapid development of data acquisition and transmission technology, it has been available to collect, transmit and store enormous operating data of the bearings, which will reflect the health conditions thoroughly. Deep learning is a promising method to completely extract hidden information from the big data and Recurrent Neural Network (RNN) is designed to process time relations of the input signals. However, there are still some shortcomings on the vanilla RNN, and Long Short-Term Memory (LSTM) neural network is proposed to optimize the model. In this paper, a new method based on time-frequency analysis and LSTM is proposed for bearing prognosis. Firstly, vibration raw signals are collected. Then, a time-frequency analysis will be applied on these raw signals. LSTM neural network is established and the time-frequency information is fed into the target network to train the target one or diagnose the bearing health state. Selection of some vital parameters in our model is discussed in detail, and then the proposed method is compared with other methods, such as vanilla RNN, MLP and SVM. Result shows that the proposed method achieves the highest accuracy rate over other methods.**

***Keywords-Bearing prognostic; Time-frequency Analysis; Deep learning; LSTM***

## I. INTRODUCTION

Rotary machinery, as one of the most important components in our modern manufacturing system, is properly worked based on different bearings [1]. According to surveys on rotary machinery, up to 45%-55% of all failures are caused by bearings [2]. Faulty bearings have been one of the tremendous danger to the rotary machinery [3]. It is of great importance to prognosis bearing faults at the early stage even the signals will be interfered by plenty of environmental noises [4-7].

Faulty Diagnosis of bearing has been widely studied by signal processing methods [1, 6, 7] or machine learning approaches [5, 8]. In order to extract and select apparent features or indexes, these methods require much prior domain knowledge, together with plenty of expert experience. With the rapid development on signal sensing, signal transferring and data storage, it has become more and more easy, cheap and reliable to acquire much more signals on the machine among different conditions than before. How to learn meaningful information from the machinery big data is a new challenge [9]. Deep learning methods are able to provide a series of powerful solutions on features extraction and selection with less prior domain knowledge and expert experience [9-11].

Recurrent Neural Network (RNN) [12], which is well-known as one of the most important kind of deep learning model, , is designed for time series analysis by introducing a special network structure, namely hidden layer. As a typical

This paper is supported by the National Natural Science Foundation of China (Grant No. 51575202，51675204) and the National Science and Technology Major Project of China (Grant No. 2018ZX04035002-002).
978-1-7281-0860-5/19/$31.00 ©2019 IEEE

kind of the time series, vibration signals should have better performance using RNN models.

However, vanilla RNN is still facing some challenges [12-14]. One is the vanishing/exploding gradient problem. They are also the common problems in other deep learning models. It can be relieved by adopting proper model structure, applying learning rate update policies and suitable optimizing functions et al. Another is the lack of the ability to handle long-term dependencies, which means this model are not that suitable to process common long time series.

Recently, many variances of RNN try to relief the vanish/explode gradient problems and improve the ability to analysis long-term dependencies, such as Long Short-Term Memory(LSTM) neural network [14], Gated Recurrent Unit (GRU) neural network [15], Clockwork RNN neural network [16] and Skip RNN neural network [17] etc. And these models have been quite successful in a lot of scenarios, such as Natural Language Processing (NLP)[18], handwriting recognition [19], driving [20] and so on. Among these models, Greff [21] found that LSTM neural network has better performance in many situations.

In recent years, RNN and its variances, especially LSTM, have also come to be applied in other domains, such as tool wear prediction during milling process [22], remaining useful life (RUL) estimation of aero-engine [23], railway track circuit fault diagnosis [24], gearbox fault diagnosis on wind turbine [25], induction motors fault diagnosis [26] and so on. But there are not that much publications about fault diagnosis on bearings [10]. Liang [27] proposed a health indicator (HI) for bearing RUL prediction with LSTM neural network. Liu [28] proposed a Kernel PCA (KPCA) based RUL prediction of rolling bearings and RNN-based auto-encoders. However, these models do not utilize the time-frequency relations behind the signals. In this paper, LSTM is introduced for bearing fault diagnosis by processing time-frequency relations.

This paper is organized as follows. In Section 2, the basic structure and the model topology of vanilla RNN and LSTM are reviewed in depth. In Section 3, massive experiments are conducted in sequence in order to validate the outstanding performance of the proposed method comparing to other prognostic methods. Finally, some conclusions are provided in Section 4.

## II. METHOLOGY

### A. Data Preprocess

Fast Fourier Transfer (FFT) is a popular tool to obtain the frequency spectrum maps of stable signals, which is the bridge between fault frequencies and the fault types. However, FFT is not that suitable for those non-stationary signals, which is more common in reality. Instead, time-frequency analysis are designed for these non-stationary ones. Among these time-frequency representation and analysis methods, Short-Time Fourier Transfer (STFT) is a simple but reliable and powerful one.

As for STFT, the time series at enough short intervals of the unstable signals is able to be regarded as the stable one, which will be analyzed by FFT. Then next time series at the same short intervals will be analyzed sequentially until the original unstable signal is totally analyzed.

By STFT processing, a time-frequency map will be easily obtained. From this map, the frequency results of the signal sequence with the same length at each time steps can be identified, which is the representation of the changing procedure of the unstable origin signals. LSTM is able to efficiently and effectively extract features of the time series and select the best combinations while the data is “flowing”. Hence, in this paper, taken these frequency at different time steps into account with LSTM, the information or features behind will be fully utilized.

### B. The basic Theory of vanilla RNN

The basic structure and the folded topology of vanilla RNN are depicted in Fig. 1. Vanilla RNN shares the hidden layer *H*, which is used for storing history information. Compared with other kind of deep neural networks, the output layer *Y* is not only determined by the input layer *X* but also the previous results of hidden layer. Therefore, vanilla RNN is able to process the time series by introducing these time notion into the model.

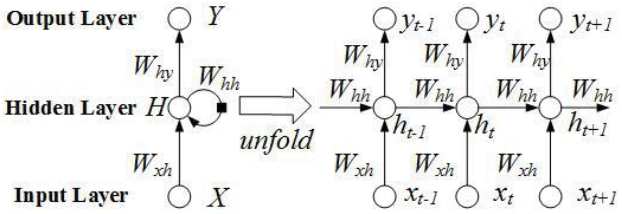


Figure 1. A typical vanilla RNN structure and the unfold topology

At the time step *t*, the input signal $x_t$ is fed into the vanilla RNN model. Mathematically, the hidden layer results $h_t$ and the output layer results $y_t$ at time step *t* can be calculated as following equations:

$$h_t = f(W_{xh}x_{xh} + W_{hh}h_{t-1} + b_{ht}) \quad (1)$$

$$y_t = g(W_{hy}h_{ht} + b_{ht}) \quad (2)$$

Where $W_{xh}$ is the weight between the input layer and the hidden layer, $W_{hh}$ is the weight between the hidden layers, and $W_{hy}$ is the weight between the hidden layer and the output layer. $b_{ht}$ is the bias value of the hidden layer at time *t*, and $b_{yt}$ is the bias value of the output layer at time *t*. *f(.)* is the activation function of the hidden layer. *g(.)* is the activation function of the output layer. In general, activation function is applied to decrease the linearity of the results, and the bias is

adapted for learning the offset. In order to train RNN, the back-propagation through time (BPTT) method is adapted.

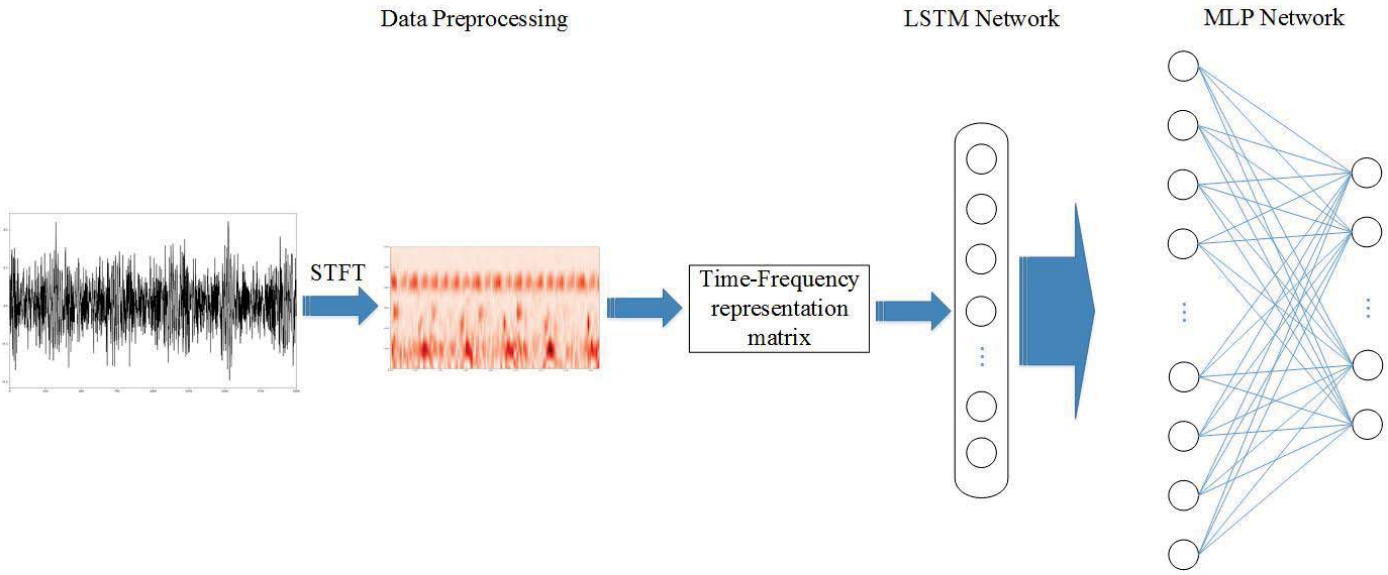


Figure 3. The structure of the proposed methods

### C. *LSTM -- One of the optimized RNN structures*

Vanilla RNN structure lacks of the ability to deal with the gradient vanishing/exploding problem and memorize too long term memory. As one of the most popular and successful variances of the RNN, LSTM is capable of dealing with the gradient vanishing/exploding problem, and analysis long term signals with the help of the memory cell in the hidden layer, which is designed to control if this cell sate will be utilized or not.

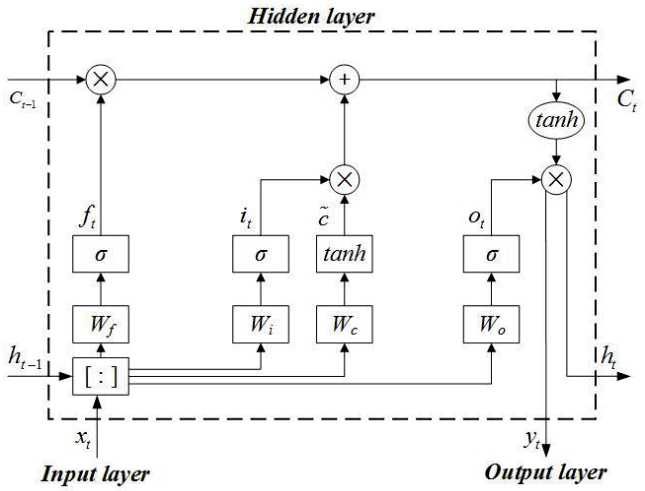


Figure 2. A typical LSTM structure

A typical structure of LSTM is depicted in Fig. 2. In general, there are the memory cell, hidden state and three different gates in the hidden layer of LSTM, namely input gate, forget gate and output gate. At the time step *t*, the input signal $x_t$ is fed into the LSTM model, outputs of these cells and gates will be updated as follows:

input gate:

$$i_t = \sigma[W_i(x_t + h_{t-1}) + b_i] \tag{3}$$

output gate:

$$f_t = \sigma[W_f(x_t + h_{t-1}) + b_f] \tag{4}$$

memory cell state:

$$\tilde{c}_t = c_{t-1} \odot f_t + i_t \odot \tanh[W_c(x_t + h_{t-1}) + b_f] \tag{5}$$

hidden state:

$$h_t = \tanh(\tilde{c}_t) \odot o_t \tag{6}$$

Where activation function $\sigma$ stands for the *Sigmoid* function. $\odot$ is the Hadamard product, i.e. element-wise product. [ : ] denote the concatenation of the input arrays. $W_i$, $W_f$, $W_o$, $W_c$ are the weights in the input gate, the forget gate, the output gate and the memory cell, respectively. And $b_i$, $b_f$, $b_o$, $b_c$ are the bias in the input gate, the forget gate, the output gate and the memory cell, respectively.

In general, the final outputs of LSTM cell are output gate and memory cell state. In this paper, the raw signals will be processed by Short-Time Fourier Transformer (STFT), then the time-frequency maps are sent to a LSTM layers with some time steps. The output array of the LSTM model at the final time steps is directly sent to a two layer multiple perception (MLP) layers. These models will fully gather the

necessary time relations information or features without any interfacing with sophisticated experts.

### D. Proposed method

The brief structure of the proposed methods is shown as Fig. 3. In general, here are 3 parts in the methods. Firstly, The 1D raw signals is processed by STFT to obtain time-frequency maps and 2D time-frequency matrices. Then, 2D these time-frequency matrices will be sent to a LSTM layer according to the time sequence relation. Finally, the results will be sent to 2 two-layer MLP networks to classify the bearing states.

The activation of LSTM and the first MLP layer is *Leaky RELU*, a popular variances of *RELU*. *Leaky RELU* is depicted as followed mathematically:

$$Leaky\ RELU = \begin{cases} x, x > 0 \\ \alpha x, x \le 0 \end{cases} \tag{7}$$

*RELU* forces the results of the neurons to be 0 if they are below 0, and these neurons will be "died" subsequently. As for *Leaky RELU* function, results of the neurons below 0 will still keep little minus value by introducing the parameter α, and these neurons will still keep "alive". The value of α should be small, and in this paper, α is set as 0.1 based on experience. As for the last MLP layer, the activation is *softmax* function. The final goal of the model is minimize the cross-entropy cost function $L$ between the real data $y$ and the predict results $\hat{y}$, which is depicted as follows:

$$L = -\sum y \log \hat{y} \tag{8}$$

To avoid the LSTM model from over-fitting, an early-stop mechanism [29] is introduced. If the validation accuracy results is no more increasing after some batches, the model will stop training and save the current parameters value as the final choice.

The gradient of the model may be bigger at first, but will be lower in the end. Faster convergence rate will reduce the time to be close to the best optimal results. Lower convergence rate is beneficial to find the exact best optimal results. According to the parameters weight $w$ and bias $b$ update formulas by the traditional mini-batch Gradient Descent:

$$w = w - \lambda \frac{\Delta_L}{\Delta_w} \tag{9}$$

$$b = b - \lambda \frac{\Delta_L}{\Delta_b} \tag{10}$$

where $\lambda$ means the learning rate, $L$ denotes the loss function. In order to adjust the gradient, learning rate update policy should be adjust at first, or choose another optimizer function. In the proposed method, learning rate $\lambda$ will be descended by exponential decay calculated by:

$$\lambda = \lambda_{start} \cdot \lambda_{decay}^{\frac{global\_steps}{decay\_steps}} \tag{11}$$

where $\lambda_{start}$ denotes the learning rate at the beginning, $\lambda_{decay}$ denotes the decay rate per epoch, operation $\cdot$ denotes the scalar multiplication. And *global_steps* means the current epoch number, *decay_steps* denotes the epoch numbers per epoch.

In addition, a novel optimizer function named Nesterov Accelerated Gradient (NAG) [30] is adopted as the model optimization method. Compared with the mini-batch Gradient Descent, NAG is able to prevent the gradients from decreasing at too fast or too slow speed by considering both the direction of the previous and the current accumulated gradients.

All these programs in this paper are running on the nvidia GeForce GTX TITAN xp graphic card with 12G graphic memory and 3840 CUDA cores, and two Intel Xeon E5-2620v4. TensorFlow 1.5 provided by Google is utilized as the model backend except SVM. And the SVM is built with Scikit-learn 0.20.1.

## III. Experiment and Results

### A. Data Discription

In this section, all the models will be trained and tested under the Case Western Reserve University motor bearing dataset [4, 5].

Signals will be collected on the driving end at the sampling frequency of 12 kHz. There are three different fault types, namely inner race fault (IR), outer race fault (OR) and ball fault (BF), and the normal condition. Each type of fault contains three different diameters, i.e. 0.007 inch, 0.014 inch and 0.021 inch. All the 10 experiment conditions are conducted under different loads varying among 1 *hp*, 2 *hp* and 3 *hp*.

Deep learning is related to big data. An enlarged method named average enhancement is applied on the raw data. At the beginning, under the real industrial condition, signals will be interfered by some hidden noise. In this paper, 3dB Gaussian white noise are added in the raw signals. Then, the raw long signals will be divided to training part and the testing part. And the length of the testing part is set to 50000. Finally, signal segments with 2048 points will be selected uniformly from training part and the testing part separately to make up the training and the testing dataset. Totally, there are 12000 samples in the training dataset and 1500 samples in the training dataset.

### B. Parameters Selection in the Method

The window function and the length of the STFT matters which determines the time and frequency resolution. The windows function is Hann window for the better balance

of the width of main lobe and attenuation of the side lobes in frequency domain. Time and frequency resolution determined by windows length are impossible to reach the best at the same time. Unfortunately, better time resolution is sure to get worse frequency resolution and better frequency resolution is sure to get worse time resolution. In this paper, the windows length is set to 64 empirically. For better reproducibility, all the experiments will be conducted for 3 times, and the average value will be treated as the final model prediction results.

The time steps and the neural number of LSTM cell network is determined by the time and frequency matrices, where the time steps is determined by the total times interval number, and the neural number in LSTM cell is set to the number of the features number in frequency axis. The output array of the LSTM model at the final time steps is directly sent to a two layer MLP layers. As for the two-layer MLP network, the middle neurons number should be determined by experiments. In this paper, the middle s number is set to range in the list [8, 16, 32, 64, 128]. The last neurons number is set to be same with the output category number 10. Assuming the learning rate is 0.05, the results is shown is Fig 4. Obviously, middle neurons number in the proposed method is better to be 32 while the accuracy rate is up to 0.9527.

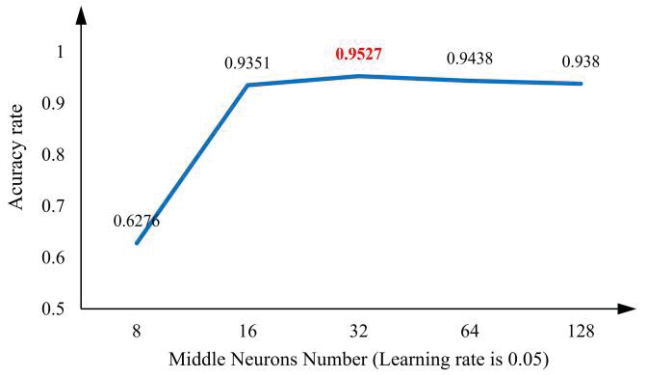


Figure 4. Accuracy rate over different middle neurons number

TABLE I. ACCURARY RATE OVER DIFFERENT LEARNING RATE IN THE PROPOSED METHOD

| **Learning rate** | 0.005 | 0.05 | 0.5 |
|---|---|---|---|
| **Accuracy rate** | 0.8911 | 0.9527 | 0.1 |

The batch size will affect the usage of graphical memory and the training speed. In this paper, batch size is set to 16. Learning rate influences the convergence speed, and will be descended by exponential decay. Results will be worse if the learning rate is too big or too small. In this paper, learning rate will be compared in [0.005, 0.05, 0.5], and result is shown in Table I. In this experiment, the optimized learning rate is determined as 0.05.

### *C. Performance Comparison*

In order to validate the promising performance of the proposed method, these methods will be compared:

- Vanilla RNN with the same method parameters;
- 4 layer MLP with flatten array input;
- SVM with flatten array input.

The learning rate of Vanilla RNN, MLP and the proposed method is set to 0.05 to keep the same. Since MLP and SVM is unable to process the 2D array directly, these 2D data will be flatten to 1D array firstly. The neuron number in the layers of MLP is [2193, 362, 64, 10]. As for SVM, the linear kernel is selected. And the penalty parameter C of the error term in SVM is set to range in [0.1, 1, 10, 100].

The results of other models and the proposed model are shown in Fig. 5. Compared with other models, the accuracy of the proposed one achieves the highest rate. And MLP performs the worst, even worse than SVM, which means that the shadow learning method might also perform better than some deep learning ones. Contrasted to Vanilla RNN, the proposed method performs better for the ability to better relief the gradient vanishing/exploding problem and capture the longer term memory.

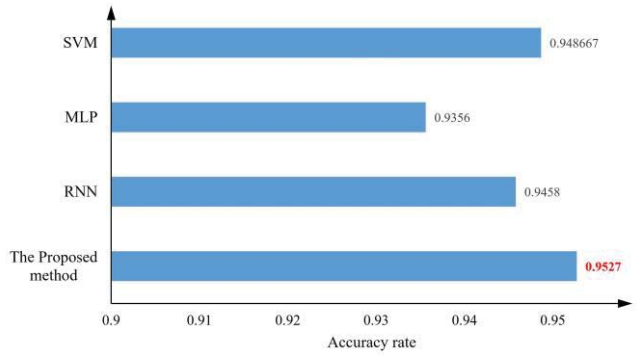


Figure 5. The accuracy rate over different models and the proposed method

## IV. CONCLUSIONS

In this paper, a novel bearing diagnosis method based on time-frequency analysis and LSTM is present. As a unstable signal, vibration signals is better to be analyzed by time-frequency methods. As a popular and efficient variances of RNN, LSTM is utilized to process the time-frequency data processed by STFT. By better capturing the time-frequency relation, LSTM is able to achieve higher accuracy rate in bearing state classification. Further experiments compared with Vanilla RNN, MLP and SVM have proven the better performance of the proposed method.

In the further research, our efforts will be mainly focused on these four aspects. First, optimize the proposed method furtherly. Second, try to figure out the influence of the real noise on the proposed method. Third, try to combine the deep learning models with deep learning approach. Last but not the least, try to apply this method in the practical conditions.

## ACKNOWLEDGMENT

The authors would like to thank Su Jiang, Ruizhen Jing and Zisheng Wang for their kind help in model training and validation.

## REFERENCES

[1]. J. Duan, T. Shi, J. Duan, J. Xuan, and Y. Zhang, "A narrowband envelope spectra fusion method for fault diagnosis of rolling element bearings," Measurement Science and Technology, vol. 29, p. 125106, 2018.

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

[3]. D. Abboud, M. Elbadaoui, W. A. Smith, and R. B. Randall, "Advanced bearing diagnostics: A comparative study of two powerful approaches," Mechanical Systems and Signal Processing, vol. 114, pp. 604--627, 2019.

[4]. M. Cerrada, R. E. V. S A Nchez, C. Li, F. Pacheco, D. Cabrera, J. E. V. de Oliveira, and R. E. V A Squez, "A review on data-driven fault severity assessment in rolling bearings," Mechanical Systems and Signal Processing, vol. 99, pp. 169--196, 2018.

[5]. C. Wang, M. Gan and C. A. Zhu, "Fault feature extraction of rolling element bearings based on wavelet packet transform and sparse representation theory," Journal of Intelligent Manufacturing, vol. 29, pp. 937--951, 2018.

[6]. Y. Li, X. Wang, S. Si, and S. Huang, "Entropy based fault classification using the Case Western Reserve University data: A benchmark study," IEEE Transactions on Reliability, 2019.

[7]. Z. Zhao, S. Wu, B. Qiao, S. Wang, and X. Chen, "Enhanced sparse period-group lasso for bearing fault diagnosis," IEEE Transactions on Industrial Electronics, vol. 66, pp. 2143--2153, 2019.

[8]. H. Zhou, T. Shi, G. Liao, J. Xuan, J. Duan, L. Su, Z. He, and W. Lai, "Weighted kernel entropy component analysis for fault diagnosis of rolling bearings," Sensors, vol. 17, p. 625, 2017.

[9]. Q. T. Tran, S. D. Nguyen and T. Seo, "Algorithm for Estimating Online Bearing Fault Upon the Ability to Extract Meaningful Information From Big Data of Intelligent Structures," IEEE Transactions on Industrial Electronics, vol. 66, pp. 3804--3813, 2019.

[10]. D. Hoang and H. Kang, "A survey on Deep Learning based bearing fault diagnosis," Neurocomputing, vol. 335, pp. 327--335, 2019.

[11]. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," Mechanical Systems and Signal Processing, vol. 115, pp. 213-237, 2019.

[12]. Z. C. Lipton, J. Berkowitz and C. Elkan, "A critical review of recurrent neural networks for sequence learning," arXiv preprint arXiv:1506.00019, 2015.

[13]. Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," nature, vol. 521, p. 436, 2015.

[14]. S. Hochreiter and J. U. R. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, pp. 1735--1780, 1997.

[15]. K. Cho, B. Van Merri E Nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[16]. J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn," arXiv preprint arXiv:1402.3511, 2014.

[17]. V. I. C. Campos, B. Jou, X. Gir O I-Nieto, J. Torres, and S. Chang, "Skip rnn: Learning to skip state updates in recurrent neural networks," arXiv preprint arXiv:1708.06834, 2017.

[18]. A. Nugaliyadde, K. W. Wong, F. Sohel, and H. Xie, "Language Modeling through Long Term Memory Network," arXiv preprint arXiv:1904.08936, 2019.

[19]. R. Ghosh, C. Vamshi and P. Kumar, "RNN Based Online Handwritten Word Recognition in Devanagari and Bengali Scripts using Horizontal Zoning," Pattern Recognition, 2019.

[20]. Y. Zheng, I. H. Izzat and J. H. Hansen, "Exploring OpenStreetMap Availability for Driving Environment Understanding," arXiv preprint arXiv:1903.04084, 2019.

[21]. K. Greff, R. K. Srivastava, J. Koutn I K, B. R. Steunebrink, and J. U. R. Schmidhuber, "LSTM: A search space odyssey," IEEE transactions on neural networks and learning systems, vol. 28, pp. 2222--2232, 2017.

[22]. R. Zhao, J. Wang, R. Yan, and K. Mao, "Machine health monitoring with LSTM networks,": IEEE, 2016, pp. 1--6.

[23]. M. Yuan, Y. Wu and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network,": IEEE, 2016, pp. 135--140.

[24]. T. de Bruin, K. Verbert and R. Babu V S Ka, "Railway track circuit fault diagnosis using recurrent neural networks," IEEE transactions on neural networks and learning systems, vol. 28, pp. 523--533, 2017.

[25]. R. Yang, M. Huang, Q. Lu, and M. Zhong, "Rotating Machinery Fault Diagnosis Using Long-short-term Memory Recurrent Neural Network," IFAC-PapersOnLine, vol. 51, pp. 228--232, 2018.

[26]. D. Xiao, Y. Huang, C. Qin, H. Shi, and Y. Li, "Fault Diagnosis of Induction Motors Using Recurrence Quantification Analysis and LSTM with Weighted BN," Shock and Vibration, vol. 2019, 2019.

[27]. 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.

[28]. H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," ISA transactions, vol. 77, pp. 167--178, 2018.

[29]. L. Prechelt, "Early stopping-but when?" in Neural Networks: Tricks of the trade: Springer, 1998, pp. 55--69.

[30]. Y. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/k2),".