Machine Health Monitoring with LSTM Networks

Rui Zhao*, Jinjiang Wang†, Ruqiang Yan‡, Kezhi Mao*
*School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
Email: {rzhao001, ekzmao}@ntu.edu.sg
†School of Mechanical and Transportation Engineering
China University of Petroleum, Beijing, China
Email: jwang@cup.edu.cn
‡School of Instrument Science and Engineering
Southeast University, Nanjing, China
Email: ruqiang@seu.edu.cn

***Abstract*—Effective machine health monitoring systems are critical to modern manufacturing systems and industries. Among various machine health monitoring approaches, data-driven methods are gaining in popularity due to the development of advanced sensing and data analytic techniques. However, sensory data that is a kind of sequential data can not serve as direct meaningful representations for machine conditions due to its noise, varying length and irregular sampling. A majority of previous models focus on feature extraction/fusion methods that involve expensive human labor and high quality expert knowledge. With the development of deep learning methods in the last few years, representation learning from raw data has been redefined. Among deep learning models, Long Short-Term Memory networks (LSTMs) are able to capture long-term dependencies and model sequential data. Therefore, LSTMs is able to work on the sensory data of machine condition. Here, the first study about a empirical evaluation of LSTMs-based machine health monitoring systems is presented. A real life tool wear test is introduced. Basic and deep LSTMs are designed to predict the actual tool wear based on raw sensory data. The experimental results have shown that our models, especially deep LSTMs, are able to outperform several state-of-arts baseline methods.**

***Index Terms*—Machine Health Monitoring, Tool Wear Prediction, RNN, LSTMs**

I. Introduction

With the advent of modern manufacturing systems, the practical systems are becoming more and more complicated which require them to meet the demand of better quality, higher reliability and increased availability [1], [2], [3]. As a result, machine monitoring systems including diagnosis and prognosis approaches have received considerable and extensive attention in recent years. Diagnosis systems focus on identification and detection of faults after the occurrence of certain faults, while prognosis systems aim at estimating the future working conditions and predicting the remaining useful life (RUL) [4], [5], [6], [7], [8]. These existing machine monitoring systems can be divided into two major categories: physics-based models and data-driven ones [9], [10]. In physics-based models, domain knowledge of physical models and laws with measure data are incorporated into model constructing via mathematical equations. Some effective models including Paris crack growth model [11], Forman crack growth model [12] and so on have proven to be successful in industry. However, the physics-based models have certain disadvantages. Firstly, the performance of physics-based models are heavily dependent on the quality and accuracy of domain knowledge about the practical mechanical systems. In real life, due to complexity and noisy working condition, such kind of high-quality domain knowledge are often unavailable, which hinders the robustness of these physics-based models. Secondly, most of physics-based models are unable to be updated with on-line measured data, which limits the effectiveness and flexibility of applications of physics-based models. However, data-driven models try to derive models based on historical measured data and make decision from the online data collected from sensors on working machines. And these model parameters can be updated in real time when working statue of machines changes. In addition, the development of advanced sensors and computing systems make the research topic: data-driven machine monitoring systems more and more attractive. And in this paper, our work also focuses on this data-driven framework.

As shown in Figure 1, data-driven models take various sensor data as inputs and perform feature selections and extractions to derive representation of machine conditions. Then, representations are fed into various algorithms, which normally consist of two parts: one is model training based on historical data and the other is model prediction based on current sampled data. The core step in data-driven models is representation learning of these sensory data. Sensory data are in nature time series data, which are sampled by sensors and expressed in a sequential form. Some previous works focus on multi-domain feature extractions including statistical (variance, skewness, Kurtosis), frequency (spectral skewness) and time-frequency (wavelet coefficients) features. However, these methods do not belong to sequence models, which can not model the intrinsic sequential characteristic behind sensory data. And how to select these features is another big challenge for these methods. These models require intensive expert knowledge or feature engineering. Feature engineering is defined as the usage of domain knowledge to design features that will be feed into machine learning algorithms. As the keystone

978-1-5090-0795-0/15/$31.00 ©2016

to the application of machine learning, feature engineering is both difficult and expensive. Except from these feature extraction-based methods, some sequence models including Markov models, Kalman filters and conditional random fields are powerful for addressing sequential data, which only access to raw time series. However, they have been criticized for its inability to capture long-range dependencies. In sensory data for machine monitoring, two informative and discriminative signal may be separated by many indiscriminative or even noisy signal occupying a long time period. Therefore, the long delays that separating some important features in time-scale may lead to failures of these above sequences models. During recent years, recurrent neural networks (RNN), especially Long Short-Term Memory (LSTM) that were proposed to relief the problem of gradient exploding or vanishing in RNN, has emerged as one popular architectures to handle sequential data with various applications including image captioning, speech recognition, genomic analysis and natural language processing [13], [14], [15], [16], [17]. LSTMs are able to address sequences of varying-length data and capture long-term dependencies. As one kind of neural networks, LSTMs incorporate representation learning and model training together which require no additional domain knowledge. And this architecture can enable us to discover some unseen and hidden structure to improve the generality of model.

In this paper, we present the first empirical study about machine health monitoring systems based on LSTMs. Here, we adopt one open source dataset: dynamometer, accelerometer, and acoustic emission data sampled from a high-speed CNC milling machine cutters [1]. The corresponding task is defined as the estimation of tool wear conditions based on sensory signals. In our setting, this problem has been transformed into a regression problem with sequential data, in which each sequential data, i.e., sensory data, represents one certain tool wear condition that is quantified by the actual tool wear width. We explore the direct application of LSTMs on raw time-series data to predict the tool wear condition. We will investigate the performances of single-layer LSTMs and deep LSTMs. For comparison, a set of baseline methods including feature extraction based methods and original RNN on raw signal are introduced. Through this work, we are trying to shed some lights on the possibility of the application of LSTMs models on machine health monitoring.

This paper is organized as follows. In Section II, some related work including LSTMs and its various applications are reviewed. Then, our proposed LSTM-based machine health monitoring system are presented in Section III. Then in the following Section IV, experimental results of prediction of tool wear condition are illustrated. Finally, concluding remarks are provided in Section V.

[1]The dataset has been kindly provided at https://www.phmsociety.org/competition/phm/10

II. Related Work

A. From RNN to LSTM

Recurrent Neural Networks (RNNs) were proposed for sequence learning [18]. RNNs build connections between units from a directed cycle. Different from basic neural network multilayer perceptron that can only map from input data to target vectors, RNNs are able to map from the entire history of previous inputs to target vectors in principal. RNNs allow a memory of previous inputs to be kept in the network's internal state. RNNs can be trained via backpropagation through time for supervised tasks with sequential input data and target outputs. However, the vanishing gradient problem during backpropagation of model training hinders the performance of RNNs. It means that traditional RNN may not capture long-term dependencies. Therefore, LSTMs were firstly presented to prevent backpropagated errors from vanishing or exploding [19]. Forget gates were introduced in LSTMs to avoid the long-term dependency problem. These adopted forget gates are able to control the utilization of information in the cell states. To capture nonlinear dynamics in time-series sensory data and learn effective representations of machine conditions, LSTMs should be superior compared to traditional RNNs due to its capability to capture long-term dependencies. Considering LSTMs are able to capture long-range dependencies and nonlinear dynamics in time-series data, LSTMs have been successfully applied in various applications, including speech recognition, image captioning, handwriting recognition, genomic analysis and natural language processing [13], [14], [15], [16], [17]. Here, we are going to presented the first study of LSTMs on machine health monitoring.

B. Neural Network for Machine Health Monitoring

Due to the strong representation capability of multi-layer neural networks, neural networks have been widely applied to machine health monitoring problems [20], [21], [22], [23], [24], [25]. Most of previous works do not consider the sequential nature behind sensory data. Before feeding raw data into neural network, feature extraction and selection are performed firstly [20], [21], [22], [23]. These works do not consider the order of signal and require some feature engineering. In addition, some papers have applied RNNs to machine health monitoring problems [24], [25]. In this paper, we further proposed LSTMs to address machine health monitoring problems. Raw sensory data are directly fed into LSTMs without any manual labor.

III. LSTM Models for Machine Health Monitoring

Here, we cast machine health monitoring problems into a specific one: tool wear prediction problem. The in-process sensory data as time-series observations are used as input data. Our task is defined as designing a model to infer the tool wear conditions from these in-process multi-sensory signals. Some adopted notations in this paper are introduced firstly. Let a series of observations $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(T)}]$ denotes the acquired data for the i-th machine condition sample. And

978-1-5090-0795-0/15/$31.00 ©2016

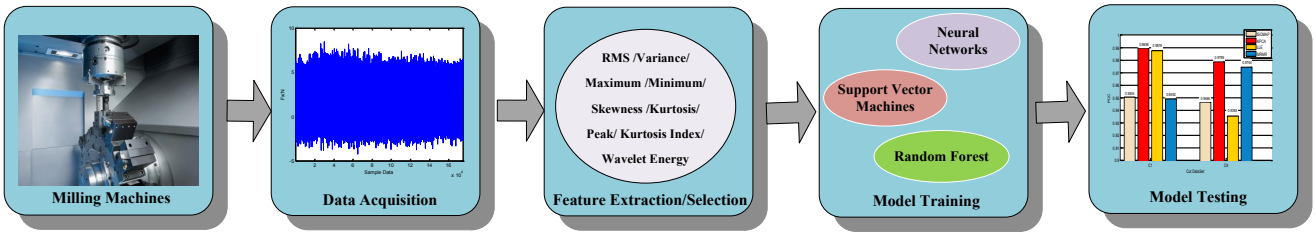

Fig. 1. Framework of data-driven machine monitoring systems.

$\mathbf{x}_i^{(t)} \in \mathbb{R}^d$ represents the multi-sensory data sampled at time step t, which is a vector and d is the dimensionality of sensory data. T is the length of sensory signal. For each sequential data $\mathbf{x}_i$, the corresponding actual tool wear condition (flank wear width) is measured and recorded as y_i. LSTMs are used to predict $\bar{y}_i$ based on sequential sensory data $\mathbf{x}_i$.

A. Basic LSTMs

The core idea behind LSTMs lies that at each time step, a few gates are used to control the passing of information along the sequences that can capture long-range dependencies more accurately. In our paper, we adopt one popular LSTM framework proposed in [26]. At each time step t, hidden state $\mathbf{h}^t$ is updated by current data at the same time step $\mathbf{x}^t$, hidden state at previous time step $\mathbf{h}^{t-1}$, input gate $\mathbf{i}^t$, forget gate $\mathbf{f}^t$, output gate $\mathbf{o}^t$ and a memory cell $\mathbf{c}^t$. The following updating equations are given as follows:

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

where model parameters including all $\mathbf{W} \in \mathbb{R}^{d\times k}$, $\mathbf{V} \in \mathbb{R}^{d\times d}$ and $\mathbf{b} \in \mathbb{R}^d$ are shared by all time steps and learned during model training, σ is the *sigmoid* activation function, $\odot$ denotes the element-wise product, k is a hyper-parameter that representing the dimensionality of hidden vectors.

Firstly, the basic LSTMs is constructed to process the sequential data in time order. And the output at the terminal time step is used to predict the output by a linear regression layer, as shown in the following equation.

$$\bar{y}_i = \mathbf{W}^r\mathbf{h}_i^T \tag{2}$$

where $\mathbf{W}^r \in \mathbb{R}^{k\times z}$ and z is the dimensionality of output. In our tasks, the output is the flank wear width so that $z = 1$. For model training, the predicted tool wear value $\bar{y}$ is compared with the true tool wear value y to derive mean squared error (MSE) as model loss.

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n}(\bar{y}_i - y_i)^2 \tag{3}$$

where n is the training sample size. The corresponding LSTMs architecture is shown in Figure 2. (a).

B. Deep LSTMs

During recent years, deep architectures have shown to be successful in representation learning [27], [28]. Therefore, it is meaningful to stack multiple LSTM layers to form a deep LSTM neural network. The core idea behind deep neural network is that inputs to model should go through multiple non-linear layers. When it comes to deep LSTMs, the input to the model can be passed through multiple LSTM layers and time series. As shown in the Figure 2. (b), the hidden output of one LSTM layer is not only propagated through time, but also used as the input data to the next LSTM layer. Therefore, the updating equations for the l-th layer can given as follows:

$$\begin{aligned}
\mathbf{i}_l^t &= \sigma(\mathbf{W}_l^i\mathbf{h}_{l-1}^t + \mathbf{V}_l^i\mathbf{h}_l^{t-1} + \mathbf{b}_l^i),\\
\mathbf{f}_l^t &= \sigma(\mathbf{W}_l^f\mathbf{h}_{l-1}^t + \mathbf{V}_l^f\mathbf{h}_l^{t-1} + \mathbf{b}_l^f),\\
\mathbf{o}_l^t &= \sigma(\mathbf{W}_l^o\mathbf{h}_{l-1}^t + \mathbf{V}_l^o\mathbf{h}_l^{t-1} + \mathbf{b}_l^o),\\
\mathbf{c}_l^t &= \mathbf{f}_l^t \odot \mathbf{c}_l^{t-1} + \mathbf{i}_l^t \odot \tanh(\mathbf{W}_l^c\mathbf{h}_{l-1}^t + \mathbf{V}_l^c\mathbf{h}_l^{t-1} + \mathbf{b}_l^c),\\
\mathbf{h}_l^t &= \mathbf{o}_l^t \odot \tanh(\mathbf{c}_l^t).
\end{aligned} \tag{4}$$

For layer one, the input data is the raw signal, i.e., $\mathbf{h}_0^t = \mathbf{x}^t$. And the outputs of the last LSTM layer at the terminal time step are adopted as the representation of the input signal, which is used for regression. The advantages of deep LSTMs are two-folders. One is that stacking layers enable the model to learn the characteristic of raw signal at different time scales. The other is that parameters can be distributed over the space, i.e., layers, instead of increasing memory size, which can contribute to more effective non-linear operations of the input raw signal.

C. Regularizations for LSTMs

Due to the model complexity of deep learning models, the large scale of training data is vital for model's robust performance. In machine monitoring problems, it is hard to obtain a large scale of training data. Therefore, it is necessary to adopt regularization for deep LSTMs. Dropout was introduced during model training for deep LSTMs [29]. Via dropout, we randomly mask parts of hidden outputs so that these neurons will not influence the forward propagation during training procedures. When it comes to testing phases, the dropout

978-1-5090-0795-0/15/$31.00 ©2016

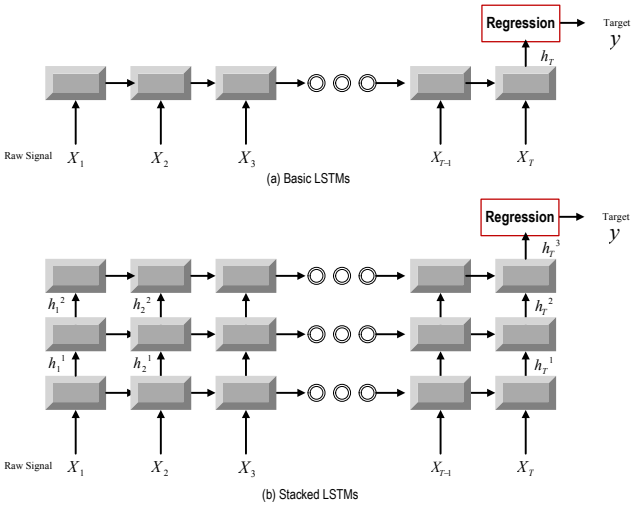

Fig. 2. Illustrations for basic LSTMs and three-layers deep LSTMs model for sequential data regression problem. Grey block denote a LSTMs layer, while dark red block represents a linear regression layer.

will be turned off and the outputs of all hidden neurons will make effects on model testing. In other view, dropout can be regarded as an approach to enlarge the training data size. During each training epoch, the application of random masking noise creates novel variants of data samples. In our models, we adopted dropout before the regression layer, which means that the output of the last hidden layer at the terminal time step will be masked partially in an random way during training phase.

IV. Experiments

In this section, we empirically evaluated the performances of LSTMs. The tool wear monitoring task is conducted. Firstly, the dataset descriptions are given. Then, details about the experimental setup are provided. Finally, the comparison results are shown and discussed.

A. Descriptions of Datasets

To experimentally verify the performance of LSTMs, a high speed CNC machine was run under dry milling operations [30]. The schematic diagram of experimental platform has been shown in Figure 3. The operation parameters are as follows: the running speed of the spindle was 10,400 rpm; the feed rate in x direction was 1,555 mm/min; The depth of cut (radial) in y direction was 0.125 mm; the depth of cut (axial) in z direction was 0.2 mm. To acquire data related to this CNC machine's operation condition, a Kistler quartz 3-component platform dynamometer was mounted between the workpiece and the machining table to measure cutting forces, while three Kistler Piezo accelerometers were mounted on the workpiece to measure the machine tool vibration in x, y, z directions, respectively [30]. DAQ NI PCI1200 was adopted to perform in-process measurements including force and vibration in three directions (x,y,z) with a continuous sampling frequency of 50

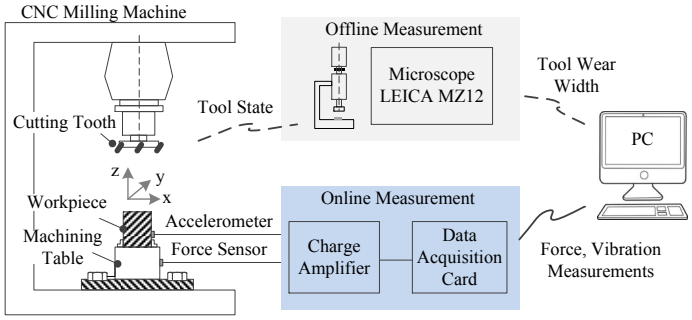

Fig. 3. Schematic diagram of experimental setup.

KHz during the tool wear test. Therefore, the dimensionality of raw signal at each time step is 7. The corresponding flank wear of each individual flute was measured offline using a LEICA MZ12 microscope after finishing each surface which is considered to be one cut number in the following data analysis and the target value. Finally, three tool life tests named C1, C4 and C6 were selected as our dataset. Each test contains 315 data samples, while each data sample has a corresponding flank wear. For training/testing splitting, a three-folder setting is adopted that two tests are used as training domain and the rest one is used as testing domain.

B. Experimental Setup

The following methods will be compared:

* LR: Linear Regression on extracted features of raw signal;
* SVR: Support Vector Regression on extracted features of raw signal;
* MLP: Multi-layer Neural Network on extracted features of raw signal;
* RNN: Orginal RNN on raw signal;
* Basic LSTMs: A single-layer LSTMs with dropout on raw signal;
* Deep LSTMs: A three-layers LSTMs with dropout on raw signal.

Since LR, SVR and MLP can not address sequential data, feature extraction is conducted firstly. The same setting in [31] was adopted here and 54 features sets (e.g. RMS, variance, and wavelet energy, etc.) were obtained. Then, each machine condition can be represented by a 54-dimensional vector, which is fed into the subsequent regression models including LR, SVR and MLP. LR has no hyperparameter. In SVR, we search the best regularization parameter C from $\{0.001, 0.01, 0.1, 1, 10\}$. For MLP, three fully-connected layers with layers' sizes of $[108, 108, 54]$ and $[162, 162, 108]$ are designed. And the activation function of each hidden layer is set to *tanh*.

For RNN, Basic LSTMs and Deep LSTMs, the input data can be time-series data so that feature extraction is not required here. Considering the sampling frequency is quite high that each data sample has over 100 thousands time steps, the whole sequence are divided into 100 sections and the max value of each section is kept to form a new time step. Via doing this,

978-1-5090-0795-0/15/$31.00 ©2016

each data sample is converted into a 100 time steps sequential data with a dimensionality of 7. For RNN, two configurations are investigated, in which hidden layer sizes are set to 14 and 21, respectively. For basic LSTMs, one layer LSTMs is considered and two different layer sizes including 14 and 21 are used. For deep LSTMs, three layer LSTMs is considered and two configurations including $[14, 21, 14]$ and $[21, 28, 21]$ are considered. And for these three models, a dense layer with a size of 56 and an activation function of *tanh* is added before the final regression layer. And the final regression layer is added into these models with a dropout operation that the masking ratio is set to 0.7.

Here, two measures to evaluate regression loss are adopt including Mean absolute error (MAE) and Root mean squared error (RMSE). MAE is the average value of the absolute values of the errors. RMSE is the square root of the average of the square of all of the errors. The corresponding equations for the calculations of these two measures are given as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\bar{y_i} - y_i| \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\bar{y_i} - y_i)^2} \tag{6}$$

C. Experimental Results on Tool Wear Prediction

In this section, we show a comparison of LSTMs with several benchmark methods. And MAE and RMSE of all methods on three different datasets are shown in Tables I and II, respectively. Datasets C1, C4 and C6 denote three different training/testing splitting scenarios that C1, C4 and C6 are used as testing datasets, respectively.

We firstly observed that among regression models including LR, SVR and MLP based on expert features, LR performs worst. It can be explained by the limitation of linear models. Consider SVR with *rbf* kernel and MLP are both nonlinear models that can capture the nonlinear relationships between the expert features and the tool wear. However, these models based on expert features all underperform compared to LSTMs models. LSTMs models all work on raw signals instead of expert features. Especially, deep LSTMs achieve a significant performance gain compared to these models. It has shown that LSTMs are able to learn meaningful representations from raw signal without any feature engineering.

RNN, basic LSTMs and deep LSTMs all belong to recurrent neural networks model. For a fair comparison, RNN share the same hidden sizes with basic LSTMs. It is obvious that basic LSTMs perform slightly better than RNN. The reasons may be the fact that gate functions employed in LSTMs can enable it to capture long-term dependency better than RNN. Among all the models, deep LSTMs achieve the best and robust performance. Compared to basic LSTMs, deep LSTMs stacked three LSTM layers and is more capable to learn robust and abstract representations from raw data. And the applied dropout layer can relieve the possible overfitting problem due to the increased model complexity of deep LSTMs.

TABLE I
MAE FOR COMPARED METHODS ON THESE THREE SDATASETS. BOLD FACE INDICATES LOWEST ERRORS

Category	Methods/Layer Sizes	Datasets		
		C1	C4	C6
Regression Models	LR	24.4	16.3	24.4
	SVR	15.6	17.0	24.9
MLP	108-108-54	25.3	17.1	23.6
	162-162-108	24.5	18.0	24.8
RNN	14	23.6	16.2	29.5
	21	13.1	16.7	25.5
Basic LSTMs	14	19.8	15.8	19.5
	21	19.6	15.6	25.3
Deep LSTMs	14-21-14	12.7	9.7	16.2
	21-28-21	**8.3**	**8.7**	**15.2**

TABLE II
RMSE FOR COMPARED METHODS ON THESE THREE SDATASETS. BOLD FACE INDICATES LOWEST ERRORS

Category	Methods/Layer Sizes	Datasets		
		C1	C4	C6
Regression Models	LR	31.1	19.3	30.9
	SVR	18.5	19.6	31.5
MLP	108-108-54	31.4	18.2	29.6
	162-162-108	31.2	20.0	31.4
RNN	14	28.7	23.1	38.7
	21	15.6	19.7	32.9
Basic LSTMs	14	24.3	19.5	28.9
	21	23.9	20.8	32.4
Deep LSTMs	14-14-21	15.6	12.8	19.8
	21-21-28	**12.1**	**10.2**	**18.9**

At last, to qualitatively demonstrate the effectiveness of LSTM models, the predicted tool wears under different datasets are illustrated in Fig 4, using deep LSTM model. The actual tool wear conditions measured offline by a microscope are also displayed, respectively. It is found that the predicted tool wear overall are able to follow the trend of the truth data well.

V. CONCLUSION

In this work, we have investigated the performances of LSTMs-based machine health monitoring systems. LSTMs do not require any expert knowledge and feature engineering, which may not be accessible in practice. And the introduction of forget gates can make LSTMs to capture long-term dependencies. Therefore, LSTMs is able to capture and discover meaningful features under sensory signal for machine health monitoring. And the experimental results have verified the superior performance of LSTMs for machine health monitoring.

While our adopted LSTMs are able to achieve satisfying and promising results, this only serves as a first trial in the research of LSTM-based machine health monitoring systems. In future work, we plan to propose a task-specific LSTMs model. For example, it is meaningful to introduce wavelet transformation, which is an effective tool to analyze machine sensory signal, into the LSTMs models.

ACKNOWLEDGMENT

This work was partially funded by National Science foundation of China (No. 51575102 and 51504274), Science

978-1-5090-0795-0/15/$31.00 ©2016

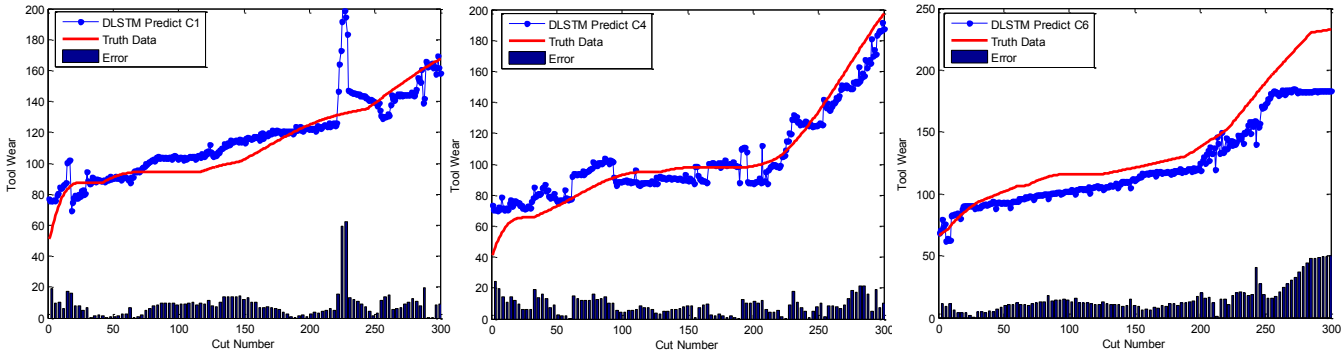

Fig. 4. Performance evaluation of Deep LSTMS under three datasets: C1, C4 and C6, respectively.

Foundation of China University of Petroleum, Beijing (No. 2462014YJRC039 and 2462015YQ0403).

REFERENCES

[1] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, Jan 2015.

[2] R. Kothamasu, S. H. Huang, and W. H. VerDuin, "System health monitoring and prognostics–a review of current paradigms and practices," in *Handbook of Maintenance Management and Engineering*. Springer, 2009, pp. 337–362.

[3] X. Chen, R. Yan, and Y. Liu, "Wind turbine condition monitoring and fault diagnosis in china," *IEEE Instrumentation Measurement Magazine*, vol. 19, no. 2, pp. 22–28, April 2016.

[4] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: a review with applications," *Signal Processing*, vol. 96, pp. 1–15, 2014.

[5] Y. Qian and R. Yan, "Remaining useful life prediction of rolling bearings using an enhanced particle filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 10, pp. 2696–2707, Oct 2015.

[6] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2633–2644, April 2016.

[7] J. Wang, R. X. Gao, and R. Yan, "Integration of eemd and ica for wind turbine gearbox diagnosis," *Wind Energy*, vol. 17, no. 5, pp. 757–773, 2014.

[8] R. Zhao, R. Yan, and R. X. Gao, "Dual-scale cascaded adaptive stochastic resonance for rotary machine health monitoring," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 529–535, 2013.

[9] M. Yu, D. Wang, and M. Luo, "Model-based prognosis for hybrid systems with mode-dependent degradation behaviors," *Industrial Electronics, IEEE Transactions on*, vol. 61, no. 1, pp. 546–554, 2014.

[10] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[11] Y. Li, T. Kurfess, and S. Liang, "Stochastic prognostics for rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 747–762, 2000.

[12] C. H. Oppenheimer and K. A. Loparo, "Physically based diagnosis and prognosis of cracked rotor shafts," in *AeroSense 2002*. International Society for Optics and Photonics, 2002, pp. 122–132.

[13] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks." in *EMNLP*, vol. 3, no. 8, 2013, p. 0.

[14] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[15] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[16] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins: Structure, Function, and Bioinformatics*, vol. 47, no. 2, pp. 228–235, 2002.

[17] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[18] J. Schmidhuber, "A local learning algorithm for dynamic feedforward and recurrent networks," *Connection Science*, vol. 1, no. 4, pp. 403–412, 1989.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] H. Su and K. T. Chong, "Induction machine condition monitoring using neural network modeling," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 241–249, Feb 2007.

[21] H. Yoon, C.-S. Park, J. S. Kim, and J.-G. Baek, "Algorithm learning based neural network integrating feature selection and classification," *Expert Systems with Applications*, vol. 40, no. 1, pp. 231–241, 2013.

[22] J. Rafiee, F. Arvani, A. Harifi, and M. Sadeghi, "Intelligent condition monitoring of a gearbox using artificial neural network," *Mechanical systems and signal processing*, vol. 21, no. 4, pp. 1746–1754, 2007.

[23] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, 2016.

[24] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, March 2011.

[25] P. Tse and D. Atherton, "Prediction of machine deterioration using vibration based fault trends and recurrent neural networks," *Journal of vibration and acoustics*, vol. 121, no. 3, pp. 355–362, 1999.

[26] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[27] G. E. Hinton, "Learning multiple layers of representation," *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[28] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] X. Li, B. Lim, J. Zhou, S. Huang, S. Phua, K. Shaw, and M. Er, "Fuzzy neural network modelling for tool wear estimation in dry milling operation," in *Annual conference of the prognostics and health management society*, 2009, pp. 1–11.

[31] J. Wang, J. Xie, R. Zhao, L. Zhang, and L. Duan, "Multisensory fusion based virtual tool wear sensing for ubiquitous manufacturing," *Robotics and Computer-Integrated Manufacturing*, 2016.

978-1-5090-0795-0/15/$31.00 ©2016