Neurocomputing 487 (2022) 22–33

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A hierarchical scheme for remaining useful life prediction with long short-term memory networks

Tao Song [a], Chao Liu [a,b,*], Rui Wu [a], Yunfeng Jin [a], Dongxiang Jiang [a,c]

[a] Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China
[b] Key laboratory for Thermal Science and Power Engineering of Ministry of Education, Tsinghua University, Beijing 100084, China
[c] State Key Laboratory of Control and Simulation of Power System and Generation Equipment, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 23 January 2021
Revised 29 January 2022
Accepted 11 February 2022
Available online 16 February 2022

Keywords:
Remaining Useful Life (RUL) prediction
Long-short Term Memory (LSTM)
Hierarchical optimization

ABSTRACT

Remaining useful life (RUL) prediction is essential in prognostics and health management (PHM) applications, where data-driven approaches employ the tendency of the degradation process using operating data of complex systems, and have attracted more and more attention. With the idea that forecasting the time period before the equipment reaches the critical degradation stage (e.g., failure, fault, etc.), RUL prediction is usually formed as an optimization problem (in particular, a regression problem between the inputs–real-time measurements and the outputs–the RUL predictions). This work formulates the RUL prediction as a bi-level optimization problem, (i) the lower level is intended to forecast the time-series in the near future, and (ii) the upper level is to predict the RULs by integrating the available measurements up-to-date and the predicted ones by the lower-level prediction. To tackle the hierarchical optimization problem, a bi-level deep learning scheme is proposed for the machine RUL prediction, where long short-term memory (LSTM) networks are applied as of the unique characteristics in processing time-series and extracting recursive and non-recursive features among them. Case studies using PHM08 data challenge data set, 4 data sets in C-MAPSS package and 1 data set in the new CMAPSS dataset are implemented, to validate the proposed framework. The results show that the presented method outperforms the state-of-the-art approaches.

1. Introduction

Prognostics and health management (PHM), focusing on predicting the failure or fault in advance and managing the complex system more efficiently and reliably, attracts more and more attention in diverse applications [1–4]. Taking the civil aviation industry as an example, the maintenance cost accounts for 10%-20% in the total cost [5], where the cost of aero-engine maintenance takes up about 30% [6]. For reducing the cost and improving the reliability, manufacturers are building PHM systems that can detect the faults in the early-stage, predict the remaining useful life (RUL) and optimize the maintenance strategy, including maximizing the service life and reducing the downtime [7]. The concept of remaining useful life (RUL) prediction is the study of predicting the length of time before components (of a service system) hit a threshold (e.g. its failure), given their history and current state [8,9]. Among the PHM topics, RUL prediction is future-oriented and essential in terms of estimating the available service time before failure, preparing the replacement in a more efficient way, etc. It is critical for prognostics and condition-based maintenance for various industrial scenarios [10,11].

As of the challenges of model-based approaches for nonlinear systems, data-driven methods (especially the deep learning methods) [1,12–15], gain more and more attention in RUL prediction. This is also inspired by the rapid development of artificial intelligence theory and methods. Typically, the parameters and working conditions that can characterize operating states of the components (e.g. the temperature, pressure, load, etc. in the thermal system) are sampled in the time domain and stored as time series data. As time-varying characteristics are essential, RNN and its improved models like long-short term memory (LSTM), Bidirectional LSTM (Bi-LSTM) [16] and deep LSTM framework [17] are widely used, where the historical states of the equipment are stored for current health state evaluation. Other networks like convolutional neural networks (CNN) [18] are also used for RUL prediction to achieve more accurate results. Except for the neural networks, research also focuses on the statistical probability meth-

* Corresponding author at: Department of Energy and Power Engineering, Tsinghua University, Beijing 100084, China.
E-mail address: cliu5@tsinghua.edu.cn (C. Liu).

https://doi.org/10.1016/j.neucom.2022.02.032
0925-2312/

ods, including particle filtering (PF), extended Kalman filtering (EKF), etc [19].

Among the data-driven algorithms, recurrent neural network (RNN) and its variant, LSTM, attract more and more attention, as of the unique characteristics in processing time-series and extracting recursive and non-recursive features among them [20,21,17]. LSTM and bi-directional LSTM models can better track the system degradation and consequently [22,23]. And more complex parallel LSTM models are also introduced using attention mechanism and degradation stage assessment, to obtain better performance [24,25].

They are applied in various fields in the industrial system, such as the Internet of Things (IOT) [26], body pose prediction [27], etc., and have made good progress.

On the other hand, time-series is widely used for the input of the RUL prediction problem, and at least in two aspects, (i) time-series (sequence) prediction that intends to forecast the trends of the system in several time steps ahead [28], and (ii) RUL prediction that tries to estimate the time left before failure or maintenance [29]. The above two aspects are considered separately and widely applied in diverse scenarios, and an integrated prediction framework for RUL is necessary in terms of improving the RUL prediction performance by utilizing the forecasting time-series and properly handling the interaction between the time-series forecasting (sequence prediction) and RUL prediction.

Considering the large number of equipment degradation process in the PHM problem, this presents a novel framework to drive the neural network model to capture the degradation trend explicitly from the signal as a hierarchical optimization problem, to better predict the equipment RUL.

Specifically, a bi-level prediction scheme with several optimization parameters is defined, integrating sequence forecast and RUL prediction. The corresponding neural network framework with clear decoupled bi-level structure is designed.

The contributions of this work include:

1. A hierarchical optimization problem is formulated for the RUL prediction and a bi-level prediction scheme is defined including the lower-level optimization problem of forecasting the time-series based on the measurements and the upper-level optimization problem of prediction the RUL based on the available time-series and the predicted ones,
2. The algorithm for the bi-level prediction framework is formed using LSTM networks, where the loss functions for the lower-level optimization and the upper-level optimization are defined as well as the joint loss function for the framework,
3. Cases studies are carried out for validating the presented framework using the datasets including dataset of Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) by NASA [30], the challenge data set by PHM society [31] and the new CMAPSS dataset (N-CMAPSS) [32]. The results show that the presented method outperforms the state-of-the-art approaches.

The remaining sections are outlined as follows. Section 2 provides the preliminaries of LSTM and applications using LSTM for sequence prediction and regression problems. The bi-level prediction framework for RUL prediction is illustrated in Section 3, including the problem formulation of the hierarchical optimization, the bi-level prediction scheme, and the flowchart of the framework with LSTM networks. Experimental results are shown in Section 4. Concluding remarks are given in Section 5 as well as future work.

2. Preliminaries

2.1. RNN and long short-term memory

Recurrent neural network (RNN) is designed to handle time-series where the recurrent features are essential for the time-series data, and RNN has been widely used in time-series prediction problems (e.g., natural language processing, NLP). To overcome the long-time sequence processing issues, Long Short-term Memory (LSTM) is proposed upon RNN, with the idea of learning to forget. LSTM network has been widely applied in recent years for diverse scenarios [33].

Recurrent neural network (RNN), was first proposed in [34], with the intention to obtain temporal dynamic behavior. By associating the connections between output and previous input, recurrent hidden states provide an efficient tool for processing the time-series data, and the applications validate their representing capabilities.

Formally, $x_t \in \mathbb{R}^k$ is the time step t in given sequence $X = [x_1, x_2, \ldots, x_n]$. At this time step, The hidden state $h_t \in \mathbb{R}^d$ is updated by

$$h_t = f(h_{t-1}, x_t) = f(Uh_{t-1} + Wx_t + b) \tag{1}$$

where parameters $U \in \mathbb{R}^{d\times d}, W \in \mathbb{R}^{d\times k}, b \in \mathbb{R}^d$ can be learned throughout training, d is the hidden layer size and f is a non-linear function.

To overcome the issues of gradient exploding and vanishing in RNNs, Long Short-Term Memory (LSTM), is proposed by Hochreiter and Schmidhuber [35], by designing a memory cell in each LSTM unit. A memory cell c_t at the time step t by encoding memory of observed input information, is formulated by three gates: input gate i_t, output gate o_t and forget gate f_t. The behavior of the memory can be maintained by [35]:

$$\begin{aligned}
i_t &= \text{sigmoid}(U_i h_{t-1} + W_i x_t + b_i)\\
f_t &= \text{sigmoid}(U_f h_{t-1} + W_f x_t + b_f)\\
o_t &= \text{sigmoid}(U_o h_{t-1} + W_o x_t + b_o)\\
\tilde{c}_t &= \tanh(U_c h_{t-1} + W_c x_t + b_c)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{2}$$

where parameters $U \in \mathbb{R}^{d\times d}, W \in \mathbb{R}^{d\times k}, b \in \mathbb{R}^d$ are the parameters learned during training, d is the size of the hidden layer, and the operator $\odot$ denotes the element-wise multiplication. In the LSTM unit, structures called *gates* (i_t, f_t, o_t) is designed using Sigmoid function to help the network to update or forget the data. Sigmoid function is one of the non-linear activation functions, which maintains the output values between 0 and 1. The outputs from Sigmoid function act on the data stream to control the gates. By training and optimizing the parameters in the network, LSTM can learn to capture important temporal information in time-series inputs. Due to the memory storage in the LSTM node, it is well able to capture timing dependent information from the input data. Therefore, LSTM is widely used in the processing of time-series.

2.2. LSTM for sequence and RUL prediction

As mentioned in the Introduction section, two scenarios are discussed using LSTM networks, (i) time-series (sequence) prediction, and (ii) RUL prediction (regression).

Time-series (sequence) prediction. Sequence prediction applies the LSTM model to predict sequences from existing sequence data. The main difference is that the goal of regression is multi-value, instead of single-value. For a given time series data of length $J: \tilde{X} = (\mathscr{x}_{t-J+1}, \mathscr{x}_{t-J+2}, \ldots, \mathscr{x}_t)$, a sequence extension prediction model can as a function $\Phi: \tilde{X} \to \hat{X}$,where $\hat{X} = (\hat{\mathscr{x}}_{t+1}, \hat{\mathscr{x}}_{t+2}, \ldots, \hat{\mathscr{x}}_{t+K})$:

$$\begin{aligned} \hat{X} &= \Phi\left(\tilde{X}\right) \\ &= \Phi(\mathscr{x}_{t-J+1}, \mathscr{x}_{t-J+2}, \ldots, \mathscr{x}_t) \\ &= \underset{\mathscr{x}_{t+1},\ldots,\mathscr{x}_{t+K}}{\arg\max} p(\mathscr{x}_{t+1}, \ldots, \mathscr{x}_{t+K} | \mathscr{x}_{t-J+1}, \ldots, \mathscr{x}_t) \end{aligned} \tag{3}$$

In practice, such models are usually implemented with a LSTM encoder-decoder structure, which is widely used in machine translation [36], weather forecasting [37], equipment health index characterization [28], etc.

RUL prediction (regression). For a given time series data of length $L: \overline{X} = (\mathscr{x}_{t-L+1}, \mathscr{x}_{t-L+2}, \ldots, \mathscr{x}_t)$, a single-valued (RUL) regression LSTM model can be defined as a function $\Psi: \overline{X} \to \widetilde{T}_{\text{RUL}}$:

$$\begin{aligned} \widetilde{T}_{\text{RUL}} &= \Psi(\overline{X}) \\ &= \Psi(\mathscr{x}_{t-L+1}, \mathscr{x}_{t-L+2}, \ldots, \mathscr{x}_t) \\ &= \underset{T_{\text{RUL}}}{\arg\max}\ p(T_{\text{RUL}} | \mathscr{x}_{t-L+1}, \mathscr{x}_{t-L+2}, \ldots, \mathscr{x}_t) \end{aligned} \tag{4}$$

The regression problem formulation has been successfully applied in RUL predictions based on the available time-series data [29,33,38]. Most of the existing LSTM RUL prediction methods applies a similar single-valued regression structure.

3. Hierarchical optimization scheme for RUL prediction using LSTM networks

3.1. Problem formulation

As discussed in the previous sections, the RUL prediction is usually treated as a regression problem between the measured parameters (time-series) and the RULs. And intensive research has been conducted on optimizing the regression problem and finding out a better strategy to minimize the prediction errors, in terms of extracting more effective parameters for inputs, adding necessary measurements for better representing the status of the equipment, building more accurate regression model for approximating the regression problem, etc. In this work, we formulate the RUL prediction problem to gain more in the time scale, by predicting the time-series (parameters) to get a sense of the measurements of the machine in one or several time stamps ahead. The reason is that the sequence prediction has achieved great success in short-time prediction, and the predicted time-series can be leveraged for us to understand the trend of the machine degradation better.

In this context, the RUL estimation in this work is defined as a bi-level prediction problem, where the lower-level prediction $\mathscr{g}$ is to forecast the time-series of parameters or features that used for estimating the healthy state of the machine, and the upper-level prediction $\mathscr{F}$ is to estimate the RUL of the machine upon the forecasting state of the machine. With this setup, the bi-level optimization problem for machine RUL prediction is defined as:

Definition: Given a lower objection function $\mathscr{g}$ and an upper objection function $\mathscr{F}$, the bi-level optimization problem is given by

$$\begin{aligned} &\min_{J\in N}\left\{\mathscr{F}\left(\widehat{T}_{RUL}, T_{RUL}\right) : |\Psi\left(\overline{X} \cup \widehat{X}\right) - T_{RUL}|_q\right\} \\ &s.t. \quad L = 1, 2, \ldots, M \\ &\widehat{X} \in \arg\min_{K\in N}\left\{\mathscr{g}\left(\widehat{X}, \widehat{X}^r\right) : |\Phi\left(\widetilde{X}\right), \widehat{X}^r|_p\right\} \end{aligned} \tag{5}$$

where $J \in N$ is the depth of time-series used for sequence prediction, K is the depth of the predicted sequences, L is the depth of measured time-series used for RUL prediction. M is the constraint for the upper-level optimization, which is to be optimized by both the lower and upper objective functions.

In this definition, the time-series forecasting is considered as a lower level optimization problem:

$$\min \mathscr{g}\left(\widehat{X}, \widehat{X}^r\right) = \min |\widehat{X} - \widehat{X}^r|_p = \min |\Phi\left(\widetilde{X}\right) - \widehat{X}^r|_p \tag{6}$$

where $\widehat{X} = (\hat{\mathscr{x}}_{t+1}, \ldots, \hat{\mathscr{x}}_{t+K})$ is the time-series predicted by $\widetilde{X} = (\mathscr{x}_{t-J+1}, \mathscr{x}_{t-J+2}, \ldots, \mathscr{x}_t)$, $\widehat{X}^r = (\mathscr{x}_{t+1}, \ldots, \mathscr{x}_{t+K})$ are the measured time-series used for the ground truth of $\widehat{X}$, and Φ is the prediction function for mapping the input time-series to the output time-series discussed in Eq. (3).

The RUL prediction is defined as an upper-level optimization problem:

$$\min \mathscr{F}\left(\widehat{T}_{RUL}, T_{RUL}\right) = \min |\Psi\left(\overline{X} \cup \widehat{X}\right) - T_{RUL}|_q \tag{7}$$

where, $\widehat{T}_{RUL}$ is the predicted RUL, T_{RUL} is the ground truth, $\overline{X} = (\mathscr{x}_{t-L+1}, \mathscr{x}_{t-L+2}, \ldots, \mathscr{x}_t)$, and Ψ is the regression function for estimating the RUL based on the inputs $\overline{X} \cup \widehat{X}$ discussed in Eq. (4).

The defined bi-level optimization scheme predicts the RUL based on the measured time-series of the mechanical system until the current timestamp and the predicted time-series by the lower-level time-series prediction algorithm. The scheme is shown in Fig. 1. Therefore, it can be treated as a joint prediction of the time-series and the RUL, where the two optimization targets are dependent. It is usually intractable for this kind of hierarchical optimization problem, especially for the case of pessimistic position. In this work, a machine-learning framework using LSTM networks is presented to solve the bi-level optimization problem.

It should be noted that the input time-series $X = [x_1, x_2, \ldots, x_n]$ can be the raw measurements from the equipment (e.g., pressure, temperature, etc.) For the sake of representing capacity, data preprocessing and feature extraction methods are usually applied, such as denoising, moving average, distance measure, Principal Component Analysis (PCA), etc. In the data sets used in this work, PCA is applied to find out the components that can better represent the degradation process. However, the data preprocessing and feature extraction vary in cases, and the presented scheme is from the feature extraction methods.

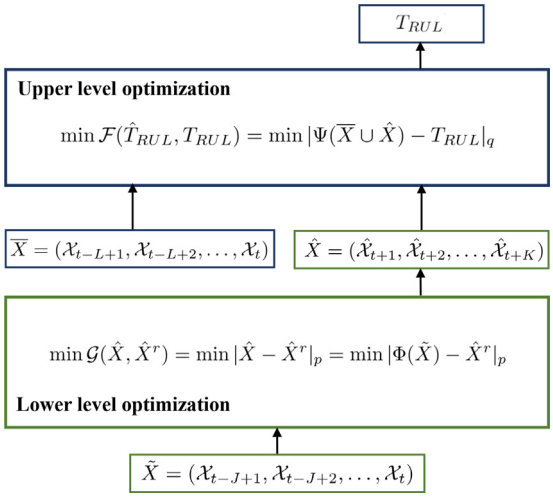

Fig. 1. Bi-level optimization scheme for RUL prediction.

3.2. Bi-level optimization scheme for RUL prediction using LSTM networks

3.2.1. Procedure

This work adopts LSTM for the prediction approach for the sequence prediction and RUL estimation because of its advantages in processing the time-series data, including two steps:

1. **LSTM model for time-series forecasting**. A deep structure with LSTM networks is defined for multi-variate time-series forecasting, where the inputs are the time-series data till the current time, and the outputs are the time-series in the next several steps.
2. **LSTM model for RUL estimation**. An LSTM regression structure is formed for RUL predictions, where the inputs are the predicted time-series data, and the output is the predicted RUL. The LSTM structure consists of a few LSTM layers to capture the recurrent features along with the time-series data and several fully connected layers to approximate the relationship between the learned recurrent features and the target RUL values.

3.2.2. The bi-level LSTM framework

The bi-level framework implemented with LSTM structures for RUL prediction is demonstrated in Fig. 2.

For the lower-level LSTM networks, $\widetilde{X} = (\mathscr{X}_{t-J+1}, \mathscr{X}_{t-J+2}, \ldots, \mathscr{X}_t)$ is used as the inputs of the LSTM model, and $\widehat{X} = (\hat{\mathscr{X}}_{t+1}, \ldots, \hat{\mathscr{X}}_{t+K})$ is the predicted sequence. The loss function for the lower-level LSTM networks is:

$$\mathscr{L}_{lower} = |\left(\widehat{X}\right) - \widehat{X}^r|_p \tag{8}$$

For the upper-level LSTM networks, the inputs consist of two components, $\overline{X}$ and $\widehat{X}$, where $\overline{X}$ is the time-series used for RUL prediction

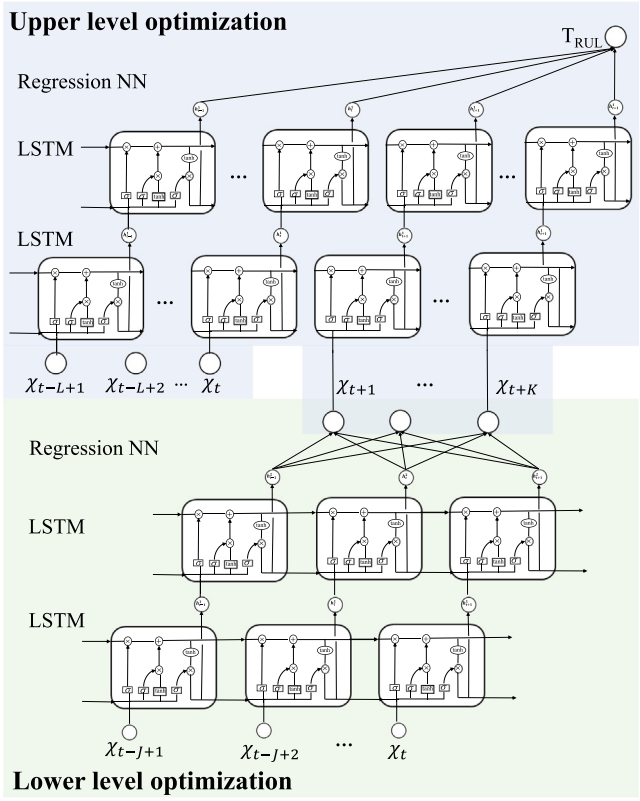

Fig. 2. Bi-level LSTM model structure for RUL prediction.

that is real measurements from the mechanical system, $\widehat{X}$ is the sequence that is predicted from the lower-level LSTM networks. The output of the upper-level LSTM networks is the predicted RUL $\widehat{T}_{RUL}$. In this setup, the loss function for the upper-level LSTM networks is:

$$\mathscr{L}_{upper} = |\widehat{T}_{RUL} - T_{RUL}|_q \tag{9}$$

Then the bi-level LSTM framework for RUL prediction is formed, and the overall loss function of the framework is defined as the weighted sum of the loss functions of the two levels,

$$\begin{aligned} \mathscr{L} &= (1-\lambda) \times \mathscr{L}_{lower} + \lambda \times \mathscr{L}_{upper} \\ &= (1-\lambda) \times |\left(\widehat{X}\right) - \widehat{X}^r|_p + \lambda \times |\left(\widehat{T}_{RUL}\right) - T_{RUL}|_q \end{aligned} \tag{10}$$

where λ is a hyper-parameter. With such a coupling form, the parameter gradients descent during training will simultaneously optimize the error of sequence prediction and RUL estimation.

3.2.3. Algorithm

To implement the bi-level LSTM framework, the pseudocode is listed in Algorithm 1.

Algorithm 1: Bi-level LSTM framework for RUL prediction

Input: Inputs for lower-level prediction
$\widetilde{X} = (\mathscr{X}_{t-J+1}, \mathscr{X}_{t-J+2}, \ldots, \mathscr{X}_t)$ with number of time steps J, inputs for upper-level prediction
$\overline{X} = (\mathscr{X}_{t-L+1}, \mathscr{X}_{t-L+2}, \ldots, \mathscr{X}_t)$ with number of time steps $\max(L,J)+K$, learning rate η, number of epochs E, total number of batches B
Output: the models' parameters x, the RUL predictions T
Initialize the model parameters x
Do choose the length J of inputs for the upper-level prediction
Do choose the length L of inputs for the upper-level prediction
Do determine the length K of outputs for the lower-level prediction
while *epoch* $k < E$ **do**
  for each batch $e < B$ **do**
    do compute the loss for the lower-level
    $\mathscr{L}^e_{lower,k} = |\left(\widehat{X}^e_k\right) - \widehat{X}^{r,e}|_p$
    do compute the loss for the upper-level
    $\mathscr{L}^e_{upper,k} = |\widehat{T}^e_{RUL,k} - T^e_{RUL}|_q$
  end for
  do compute loss $\mathscr{L}_k = \frac{1}{B}\sum_{e=1}^{B}\left((1-\lambda)\mathscr{L}^e_{lower,k} + \lambda\mathscr{L}^e_{upper,k}\right)$
  do gradient descent, $g_k = -\nabla\mathscr{L}_k(w_k)$
  do update parameters, $w_{k+1} = w_k + \eta g_k w_k$
end while

Remark 1. The bi-level optimization problem is defined in this section for the RUL prediction in mechanical systems, where the lower-level optimization is intended to forecast next few steps of the measurements or parameters, and the upper-level optimization is to prediction the RUL based on the known (measurements) parameters and the forecasting parameters. For the defined bi-level optimization problem, a bi-level LSTM framework is presented above for the purpose of getting an approximate solution of the optimization problem. And other types of algorithms (e.g., linear regression, support vector regression, RNN, GRU) can also be adopted in the scheme of the bi-level optimization problem, and it will be discussed in future work.

To validate the proposed framework, case studies on a turbofan engine degradation data set are carried out in the following section.

4. Results and Discussions

4.1. Data sets

Four data sets are applied in this section for validating the efficacy of the proposed bi-level LSTM framework for RUL prediction, (i) GT dataset, (ii) C-MAPSS dataset, (iii) PHM08 dataset, and (iv) N-CMAPSS dataset. Most of them are widely used by the community, especially (ii) and (iii). Intensive researches have been conducted using the two data sets [28,39,40].

GT dataset. A thermal system simulation model of single shaft gas turbine (GT) is simulated in this work. In order to avoid the influence of non-degradation factors on the performance of gas turbine, the rotation speed (n), fuel flow rate (G_f) and ambient temperature (T_0) are controlled as constants. Three parameters are selected to measure the degradation of the gas turbine, including the output power (N_e), the compressor outlet pressure (P_2) and the turbine outlet temperature (T_4). Based on the health state, additional degradation parameters are added to simulate the degradation under different life cycles. Two fault modes are considered in the GT dataset: turbine flow degradation (GT01) and combustor efficiency degradation (GT02). The variations of measured parameters under the two fault modes are shown in Fig. 3.

Based on the health state, additional degradation parameters are added to simulate the degradation under different life cycles. Two fault modes are considered in the GT dataset: turbine flow degradation (GT01) and combustor efficiency degradation (GT02). The variations of measured parameters under the two fault modes are shown in Fig. 3.

By embedding the degradation curve into the parameters generated by the simulation model, the degradation data in the full life cycle can be obtained and used as the training set. The test set can be generated in the same way, and part of the data is intercepted to verify the method established in this work. There are 400 samples of data contained in the GT dataset, including 100 samples of training data and 100 samples of test data for each fault mode.

C-MAPSS dataset. The turbofan engine degradation dataset [30], is simulated by NASA using a model-based simulation program, i.e., Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). C-MAPSS dataset contains four subsets, which differ in the amount of data, the operating conditions, and the number of fault types. Each subset consists of a training set and a test set. The training set has temporal data on the full life cycle of hundreds of engines, while the test set has the same amount of incomplete data. Each engine temporal data contains three operational setting parameters and 21 sensor measurement data. The description for each subset of data is listed in Table 1.

Table 1
Information for C-MAPSS dataset and PHM08 Challenge Dataset.

Data set	C-MAPSS				PHM08
	FD001	FD002	FD003	FD004	
Units for training	100	260	100	249	218
Units for testing	100	259	100	248	218
Operate conditions	1	6	1	6	N/A
Fault modes	1	1	2	2	N/A

PHM08 dataset. PHM08 Prognostics Data Challenge Dataset is a dataset used for the prognostics challenge competition at the International Conference on Prognostics and Health Management (PHM08). Its data composition is the same as the C-MAPSS dataset, and its information is also shown in Table 1. However, the real RUL results of the test set are not visible, but the predicted result can only be evaluated by uploading to NASA Data Repository website [31].

N-CMAPSS dataset. The new CMAPSS dataset [32], is a further developed dataset from the original CMAPSS dataset with completely simulated flights record of a commercial jet and increased fidelity of degradation modelling by relating the onset of the degradation process. It contains 10 data subsets with different engine units, flight distance and failure modes. The first subset DS01 with 10 engine units, nearly a hundred flights for each unit, is selected as the experimental data set. Both scenario descriptors w and the measurements x_s are given as the input to fit the RUL Y_{dev} in cycles.

4.2. Data preparation

The time-series from the aforementioned data sets are preprocessed before feeding to the proposed bi-levelLSTM networks, including data normalization, feature selection with principal components analysis (PCA), and sliding-window selection, as shown in Fig. 6.

4.2.1. Normalization

Since the raw data from the datasets is collected by different sensors, it has widely various scales. It is necessary to normalize the data for each component from the raw data. Min–max normal-

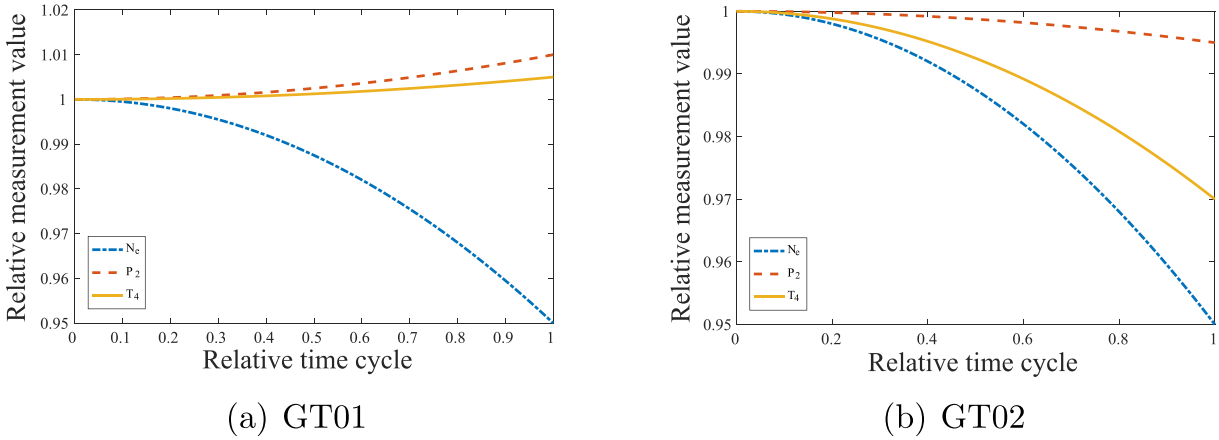

Fig. 3. The degradation curves of gas turbine under different fault modes.

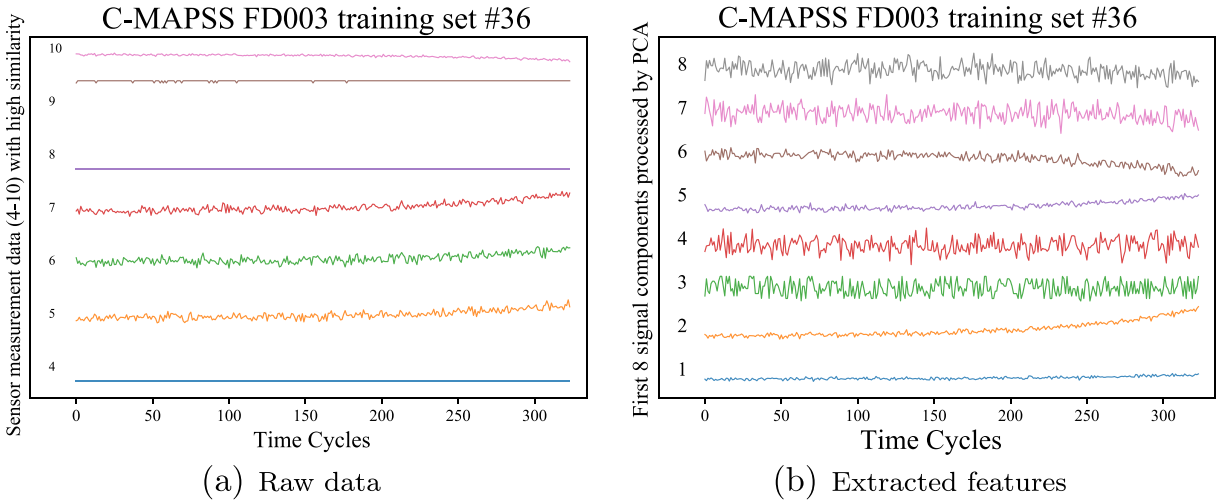

Fig. 4. Raw data (#4-#10) and extracted features (with PCA, $n = 8$) of #36 in C-MAPSS FD003.

ization is adopted to the data to scale it within the range of $[0, 1]$, which is demonstrated as follows:

$$\mathscr{X}_i' = \frac{\mathscr{X}_i - \min \mathscr{X}_i}{\max \mathscr{X}_i - \min \mathscr{X}_i} \tag{11}$$

where $\mathscr{X}_i$ is the raw data from the ith component.

4.2.2. Feature selection

As shown in Fig. 4a, there are similarity and redundancy between the different sequences which were provided by the sensors measuring relevant physical quantities. PCA is applied in the raw data for selecting the useful features for the LSTM inputs.

Through PCA, the number of features used for LSTM inputs is noted as $\mathbb{M}$, which is reduced from the original 24 features. Specifically, The process of PCA transforms the multivariate data from $s \in \mathbb{R}^{\mathbb{L}\times 24}$ to $s \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}}$, where $\mathbb{L}$ is the raw data length and $\mathbb{M} < 24$ is the number of new dimensions for the data (PCA features). The PCA features $\mathbb{M}$ is set to the hyperparameter of the model for parameter optimization adjustment and differs for different datasets. Fig. 4b shows an example of the feature selection with PCA. It shows that the selected features present more obvious degradation trends than the original 24 features (shown in Fig. 4a).

4.2.3. Sliding-window selection and RUL target generation

For the presented bi-level LSTM framework, there are two parameters for the selection of the time-series, i.e., J and L, where J is the number of time steps for the lower-level LSTM networks, and L is the number of the time steps used for upper-lower LSTM networks. Note that the total length of the time-series is $\max(L, J) + K$ for the upper-level LSTM networks, where K is the number of time steps that are predicted by the lower-level LSTM networks (as shown in Fig. 2). J and L are overlapped in the time-series as both are the time-series end at $\mathscr{X}_t$. However, J does not have to be equal to L, where both of them are the hyperparameters for the bi-level LSTM framework.

Formally, a time window of length $\mathbb{T} = \max(L, J) + K$ is applied to the data, which transforms it to $s \in \mathbb{R}^{\mathbb{T}\times\mathbb{M}}$, where $\mathbb{M}$ is the number of selected PCA features. In this work, the window sizes are J and L for the lower-level and upper-level LSTM networks. The time window length $\mathbb{T}$ is optimized by J, L and K.

For training, target RUL for RUL estimate and sequence predict target is generated at the same time as the time window processing, as displayed on the left side of Fig. 6. For RUL target generation, a piecewise linear degradation RUL label proposed in [38] is adopted in this work, as demonstrated in Fig. 5.

4.3. Structure of the bi-level LSTM networks

The bi-level LSTM framework consists of lower-level LSTM networks and upper-level LSTM networks, where each level of the LSTM networks contains multiple layers of LSTM units and fully connected layers, as shown on the right side of the Fig. 6.

Specifically, the model has two outputs, and the model can be split into two levels according to the output layer. The first level can be considered as a sequence predict neural network, whose output is fitted to a subsequent sequence of input by training. The second level can be regarded as an RUL estimate neural network whose output is fitted to the RUL from the first level's output together with the input sequence. The two levels of the model have similar structures and are connected in series, which allows the parameter gradients to propagate throughout the bi-level model during training. This means that the two levels of the model are highly coupled, which is different from the two-step prediction. The detailed information of the data processing in each epoch during training the framework is shown in Fig. 6.

Hyperparameters. For different data sets, the hyperparameters in the model will be searched and optimized during the training process to obtain better results. In addition to optimizing the number of hidden units in each layer of the model, learning rate and

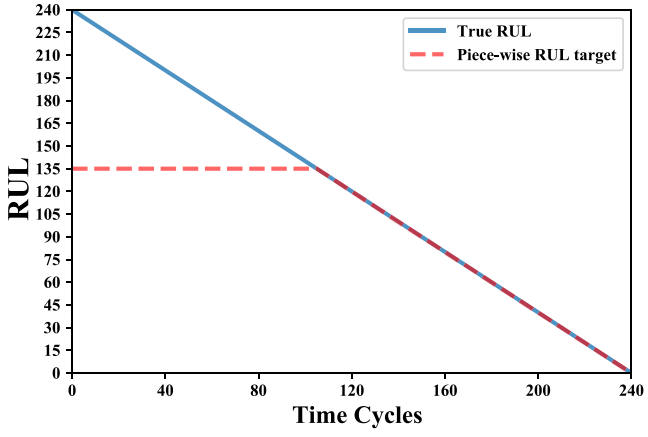

Fig. 5. Piecewise linear degradation RUL label.

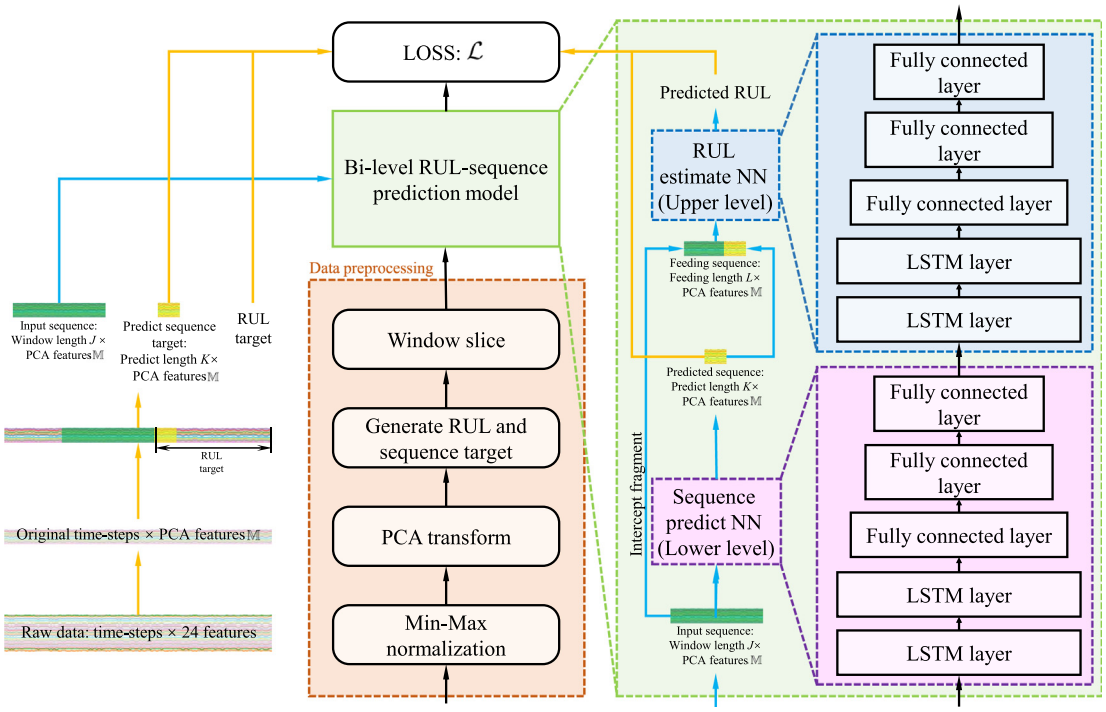

Fig. 6. Data processing in each epoch during training process of the Bi-level deep LSTM networks.

Table 2
Hyperparameter optimization on different data sets.

Hyperparameters	C-MAPSS				PHM08	N-CMAPSS
	FD001	FD002	FD003	FD004		
Features selected by PCA $\mathbb{M}$	4	6	4	8	8	/
Steps for lower level J	68	76	60	76	80	64
Steps for upper level L	36	32	36	32	40	61
Predictions by lower level K	5	8	6	8	5	3
Hidden units, LSTM-1&2	28	32	36	32	40	24
LSTM Dropout, LSTM-1&2	0.12	0.21	0.28	0.18	0.20	0
Lower Hidden units, FC-1	100	144	96	144	100	48
Hidden units, FC-2	50	72	48	72	50	36
Hidden units, LSTM-1&2	36	36	32	36	40	24
LSTM Dropout, LSTM-1&2	0.036	0.086	0.077	0.072	0.087	0
Upper Hidden units, FC-1	140	224	120	96	100	24
Hidden units, FC-2	70	112	60	48	50	12
Batch size	256	128	128	448	320	256
Learning rate for Adam	8.1E-5	1.5E-4	4.3E-4	9.3E-5	4.6E-5	1E-3
λ	0.59	0.98	0.87	0.56	0.76	0.75

batch size, we also set the PCA components $\mathbb{M}$, the depth of time-series used for sequence prediction J, the depth of the predicted sequences K, the depth of measured time-series used for RUL prediction L and loss function weighting factor λ as hyperparameters for searching. The list of hyperparameters and optimal values searched in our experiments are shown in Table '2 below.

Training.

The whole training process of the Bi-level model is illustrated in Algorithm 1. The full life cycle data is pre-processed by the methods described in 4.2 and demonstrated in Fig. 6 to generate the input sequence, predict sequence target, and RUL target. The input sequence is fed to the bi-level RUL-sequence prediction model to obtain the predicted sequence and RUL. MSE (mean square error) is adopted as the metric for evaluating the loss term $\mathscr{L}$ described in Equ.10

$$\mathrm{MSE}\left(\hat{Y}, Y\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2 \tag{12}$$

where $\hat{Y}$ are the predictions, and Y is the ground truth.

The model is trained with dropouts and Adam optimizer to minimize the loss function.

Testing. In the testing process, the regularization scalers and model fitted by training data are used for the RUL prediction. As there is no ground truth for the outputs of the lower-level model (the predicted sequence $(\mathscr{X}_{t+1}, \ldots, \mathscr{X}_{t+K})$) for the testing data, which is different from the training process. The predicted sequence is directly applied for the upper-level model for the RUL prediction in the upper-level model and the performance can be testing by estimating the error between the predicted RUL and the true RUL.

4.4. Results on two case studies

Performance and comparisons. The prediction performance of the model is characterized by the error between predicted RUL and real RUL. For comparing the performance of the proposed bi-levelprediction model with state-of-the-art approaches, root-mean-square error (RMSE), mean squared error (MSE), mean absolute error (MAE) and the error score defined by NASA are applied to the testing result, which are defined by:

$$\mathrm{RMSE}\left(\hat{T}_{RUL}, T_{RUL}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{T}_{RUL} - T_{RUL}\right)^2} \tag{13}$$

$$\mathrm{MSE}\left(\hat{T}_{RUL}, T_{RUL}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{T}_{RUL} - T_{RUL}\right)^2 \tag{14}$$

$$\mathrm{MAE}\left(\hat{T}_{RUL}, T_{RUL}\right) = \frac{1}{n}\sum_{i=1}^{n}||\hat{T}_{RUL} - T_{RUL}|| \tag{15}$$

$$\mathrm{Score}\left(\hat{T}_{RUL}, T_{RUL}\right) = \begin{cases} \sum_{i=1}^{n}\left(e^{\frac{T_{RUL}-\hat{T}_{RUL}}{13}} - 1\right) & T_{RUL} > \hat{T}_{RUL} \\ \sum_{i=1}^{n}\left(e^{\frac{\hat{T}_{RUL}-T_{RUL}}{10}} - 1\right) & \hat{T}_{RUL} \geqslant T_{RUL} \end{cases} \tag{16}$$

Through model optimization, the best performance of the bi-level model on the GT dataset, the C-MAPSS dataset and the N-CMAPSS dataset is listed in Table 3, 4, 7, while the performance comparisons with state-of-the-art approaches from the literature are listed in Table 3, 5, 7.

In addition, we replaced the LSTM structure in the traditional model (the upper level model) with its new variant, gated recurrent units (GRUs) [41], and trained it under the same conditions. The testing results are also listed in the tables below.

Table 3
Performance on GT dataset (RMSE metric).

Method	GT01	GT02
LSTM Network(Upper level only)	0.85	0.88
GRU Network(Upper level only)	0.79	0.92
Proposed Bi-level LSTM scheme	**0.42**	**0.58**

Table 4
Performance on the C-MAPSS dataset.

CMPASS Sub-dataset	FD001	FD002	FD003	FD004
MAE	9.53	16.28	9.58	16.95
MSE	3.44	4.81	3.52	4.83
RMSE	11.80	23.14	12.37	23.38
Score	194	3771	224	3492

Table 5
Performance comparisons of different methods on the C-MAPSS dataset characterized by RMSE.

CMPASS Sub-dataset	FD001	FD002	FD003	FD004
NN [42]	14.80	25.64	15.22	25.80
DNN [39,42]	13.56	24.61	13.93	24.31
RNN [42]	13.44	24.03	13.36	24.02
CNN [43]	18.45	30.29	19.82	29.16
Deep LSTM [40]	16.14	24.49	16.18	28.17
LSTM [42]	13.52	24.42	13.54	24.21
BiLSTM [44]	13.65	23.18	13.74	24.86
GRU	14.86	24.94	14.90	25.95
BiGRU	14.92	24.30	13.55	25.63
Similarity-based [45]	16.43	23.36	17.43	23.36
Dual-task LSTM [46]	12.29	**17.87**	14.34	**21.81**
LSTM with attention [25]	14.53	-	-	27.08
Proposed Bi-level LSTM Scheme	**11.80**	**23.14**	**12.37**	**23.38**

For PHM08 dataset, since the true RULs are not revealed, only the error score is applied by NASA website. The performance of our presented bi-level LSTM framework on PHM08 dataset is listed in Table 6 as well as other methods.

As listed in Table 3 and 5, with the bi-level LSTM framework, the model achieves the smallest error on data with obvious degradation trends(GT01, GT02 and C-MPASS FD001, FD003), comparing with the state-of-the-art methods. Also, it also reduces errors on complex datasets(C-MPASS FD001, FD003 and PHM08), which is very close to the best results in the literature, as listed in Table 5–7, indicating the excellent performance of our presented method.

Performance of time-series prediction. As the bi-level LSTM framework predicts the time-series (in the lower-level LSTM networks) and the RULs (in the upper-level LSTM networks) together, which solve the optimization problem in a joint manner. The performance of the time-series prediction will influence the RUL predictions. Here, the time-series prediction results are shown in Fig. 7, where four steps are predicted based on the available time-series data.

The results show that the degradation trends are well captured by the lower-level LSTM networks (the bottom two lines in the left panel of Fig. 7), which should be considered as indicators of deterioration. And the predictions follow the characteristics of the ground truth (the right panel in Fig. 7). This verifies that the time-series prediction is optimized through the bi-level LSTM framework. Although the sequence predictions are not the final output and the objective, the bi-level LSTM framework succeeds in the optimization problem by adding the loss function $\mathscr{L}_{lower}$ to the optimization term.

Performance of RUL prediction upon time-series prediction. The RUL predictions based on the available time-series and the predicted sequences are shown in Fig. 8, where 6 steps are predicted ahead of the current timestamp for FD003 and 8 step for FD004. The results show that the RUL predictions are in good accordance with the ground truth. In the meantime, it is found that the predicted error increases with the RUL, indicating that the RUL is difficult to be predicted at the early stage of the degradation. In this context, the bi-level prediction framework makes sense as it predicts the time-series first and intends to foresee a few steps ahead to find more obvious degradation characteristics and then predict the RUL upon the forecasting time-series. This can also be explained by the error analysis in the next section.

It should be noted that it may be difficult to compare the RUL predictions by the bi-level LSTM framework to the ones without

Table 6
Performance comparisons of different methods on the PHM08 dataset characterized by Score.

Method	Score
MLP [43]	3212
SVR [43]	15886
RVR [43]	8242
CNN [43]	2056
LSTM [40]	1862
LSTM with attention [25]	**1584**
Proposed Bi-level LSTM scheme	**1608**

Table 7
Performance comparisons of different methods on the N-CMPASS DS01 dataset.

Method	MAE	MSE	RMSE	Score
NN	6.392	78.929	8.713	3.093e+46
DNN	5.422	61.098	7.816	3.491e+8
LSTM	5.029	49.683	7.049	2.208e+6
Proposed Bi-level LSTM Scheme	**4.929**	**47.152**	**6.867**	**2.188e+6**

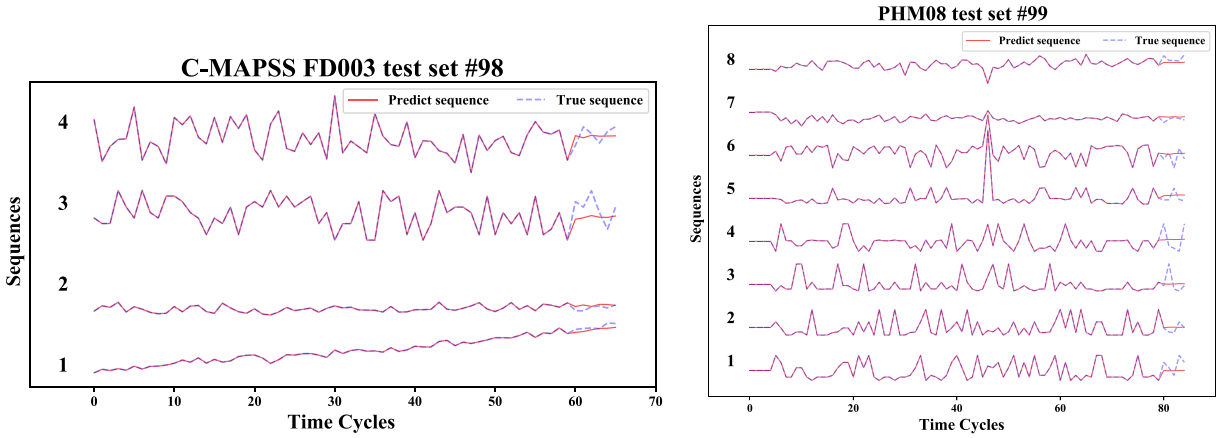

Fig. 7. Comparison between 4-step series predict sequence and original sequence from #35 and #90 in FD001 test set.

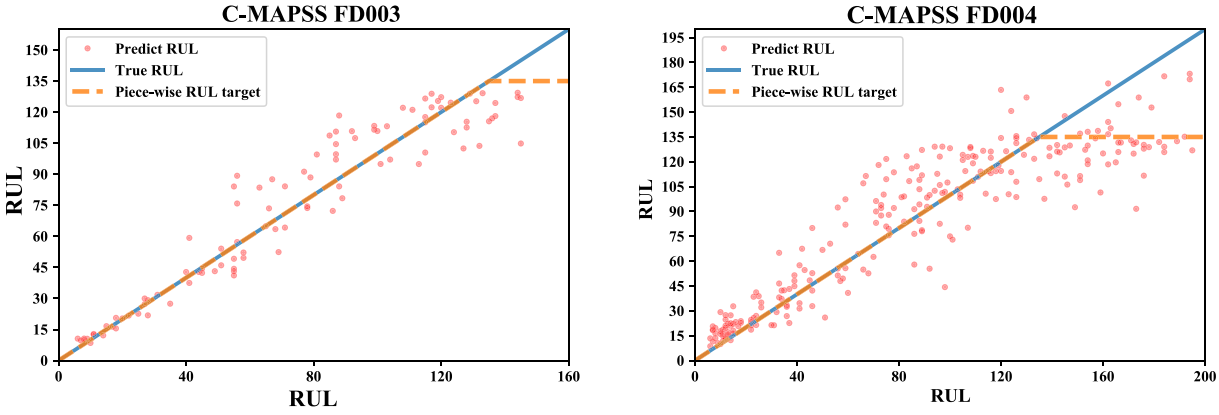

Fig. 8. RUL predictions for testing set.

using the bi-level scheme (e.g. CNN or LSTM only [43,40]). The reason is that the hyperparameters in different models vary a lot and the predictions are not comparable. However, the RMSEs for the data set can be compared and listed in Table 5.

4.5. Error analysis of the bi-level prediction scheme

Based on the predictions shown in Figs. 7 and 8, the error analysis is carried out in this section, to interpret the efficacy of the proposed bi-level prediction framework. [47] discusses the error accumulation for the RUL prediction and time-series predictions, where the two separate LSTM networks are formed for the purpose of reducing the error of RUL prediction by forecasting the time-series several steps ahead (as shown in Fig. 1 in [47]). This is considered to be a strong confirmation of the difference between short-term sequence forecasting and long-term RUL prediction.

The uncertainty of RUL prediction is more significant when there is not obvious degradation tendency in the measured parameters, other than at the time when it is close to the failure point [22], which can be inferred from Fig. 8. In this context, RUL prediction is more difficult for long-term prediction. And if we can get a sense of the measurements in the future, it is probably beneficial for reducing RUL prediction uncertainty. Therefore, a two-step prediction framework can be formed by forecasting the sequences (time series) and predicting RUL based on the forecasting sequences, where the errors from the sequence prediction and RUL estimation are expressed as:

$$\Delta E = \Delta E_{ser-pre} + \Delta E_{RUL-est} \quad \begin{cases} \Delta E_{seq-pre} & \geqslant 0 \\ \Delta E_{RUL-est} & \leqslant 0 \end{cases} \tag{17}$$

where $\Delta E_{seq-pre}$ is the additional error caused by sequence prediction, and $E_{RUL-est}$ is the additional RUL estimation error ($\Delta E_{RUL-est} \leqslant 0$) after sequence prediction.

In this work, the proposed hierarchical optimization and the implemented bi-level LSTM framework guarantees that the error is above error term is less than 0, i.e., the prediction error is reduced.

$$\Delta E \leqslant 0 \rightarrow |\Delta E_{ser-pre}| \leqslant |\Delta E_{RUL-est}| \tag{18}$$

Therefore, the performance for the proposed framework is better than the RUL estimation only. Also, through optimizing the length of sequence forecasting and the data used for RUL prediction, the formed hierarchical framework is better than the prediction with two individual steps, which are verified by comparing the results of this work and [47].

4.6. Discussions

To improve the performance of RUL prediction, this work formulates a hierarchical optimization scheme and presents a bi-level LSTM framework for both sequence prediction and RUL estimation. Inspired by the error accumulation characteristics for the time-series prediction and RUL prediction, the hierarchical optimization scheme intends to minimize the error of the RUL

prediction by forecasting a few steps ahead and getting a sense of the deterioration trends for the time-series data in an affordable cost (in terms of the additional error term). The presented bi-level framework follows the idea of the hierarchical optimization problem formulation and provides a flexible solution for tackling the optimization problem. The algorithm is applied for the RUL prediction, with the bi-level LSTM framework. Through the hierarchical design, the new loss function $\mathscr{L}$ is introduced, and the optimization problem is solved with the backpropagation of the gradient through the framework, with the purpose of minimizing the losses of the entire framework during training. Thus, the integrated workflow short-term sequence forecasting and long-term RUL prediction is achieved. Moreover, the lower-level short-term sequence forecasting is not only trained for better the output sequence, but also with the intention to find out a superior output for the RUL prediction in the upper-level model.

Using the C-MAPSS data set and PHM08 data set, the case studies are carried out on the presented bi-level LSTM framework for RUL prediction. Through the comparisons with the state-of-the-art approaches, the proposed bi-levelLSTM networks outperform the aforementioned approaches and validate the efficacy. Also, the time-series predictions and the RUL predictions are compared with the ground truth (in C-MAPSS data set), the results show that the LSTM framework can also capture the degradation trends in the time-series, besides predicting the RULs well. And this validates the adaptability of the formed hierarchical optimization scheme for the RUL prediction applications.

Compared to the single LSTM model for RUL prediction, the additional computational cost is needed for the bi-level framework to learn a lower-level sequence prediction model with LSTM. As of increased hyperparameters, the bi-level framework consumes more time for the training process. Once the model is pre-trained, the computational cost of the inference process is not significantly increased. For a trained LSTM model, the computational complexity is $\mathscr{O}(4\mathscr{I}\mathscr{H} + 4\mathscr{H}^2 + 3\mathscr{H} + \mathscr{H}\mathscr{P})$, where $\mathscr{I}$ is the number of inputs for LSTM, $\mathscr{H}$ is the number of hidden units, and $\mathscr{P}$ is the number of outputs. Therefore, the complexity for the bi-level framework is $\mathscr{O}(4J\mathscr{H}_l + 4\mathscr{H}_l^2 + 3\mathscr{H}_l + \mathscr{H}_l\mathscr{K} + 4(L+J)\mathscr{H}_h + 4\mathscr{H}_h^2 + 3\mathscr{H}_h + \mathscr{H}_h)$, where $\mathscr{H}_l$ is the number of hidden units for the lower level and $\mathscr{H}_h$ is the number of hidden unit for the upper level. For the LSTM networks, $J, K, L \ll \mathscr{H}_l, \mathscr{H}_h$. Thus, the complexity is close to $\mathscr{O}(4\mathscr{H}_l^2 + 4\mathscr{H}_h^2)$, where the additional complexity is $\mathscr{O}(4\mathscr{H}_l^2)$ comparing to the single LSTM model for RUL prediction. For the C-MAPSS dataset, the average time consumption (wall time) with the presented framework is 14.5 ms/prediction, while the upper-level is about 7.3 ms (measured on an NVIDIA GeForce RTX 2080 Ti GPU). Both of them are fast enough for a real-time implementation. To speed up computation, it is possible to install dedicated hardware (e.g. field programmable gated arrays, or FPGAs) for parallel processing in actual implementation. It should be noted that the hierarchical framework presented in this work doesn't have to be LSTM based, other types of structures (e.g., CNNs, RNNs, GRU, SVR, RF, etc.) can also be applied. For the computational complexity when different structures applied, the additional computational cost is also the part to learn a lower-level sequence prediction model. Considering the lower-level model is fairly close the upper-level model in terms of the number of parameters to be learned and the computational complexity, the additional computational cost is similar to the one tested with LSTM in this work.

In this work, we used PCA to preprocess the data, which removed redundant information from the raw signal and exposed the degradation trend explicitly. It does cause a certain performance loss in terms of the computational complexity.

However, the complexity of PCA is low, especially when it is compared to LSTM neural networks. Note that PCA preprocessing is not mandatory and other parameter selection methods can be used, or the raw measurements can also be applied. Furthermore, additional optimization parameters are introduced to define the optimization problem, which are defined as hyperparameters for the bi-level optimization problem. Hyperparameter searching strategies can be applied to find out best-fit sets of hyperparameters for different scenarios, which can make the model adapt to different types of data and improve the generalization ability of the framework. For example, since C-MAPSS FD004 has a higher complexity than the FD001, a larger $\mathbb{M}$ and K are chosen for better fitting. On the other hand, the optimized parameters give us a glimpse of understanding neural network structure and provide guidance for model training. For example, the steps for lower level J is always bigger than steps for upper level L, revealing that the information required for sequence forecasting is more than RUL prediction. It also illustrates that the benefit of integrating short-term sequence forecasting and long-term RUL prediction.

It should be noted that the presented hierarchical optimization scheme is based on the optimization problem formulation in a general way, which means the optimizer does not have to be RUL estimation or LSTM networks.

We can apply the scheme in sequence prediction problems by modifying λ, and with different neural networks, such as CNNs, RNNs, Markov models, etc.

In future work, more structures will be analyzed. Also, the RUL prediction problems widely exist in machine prognostics and health monitoring community, and the presented bi-level prediction framework will also be applied in more scenarios in future work.

5. Conclusions

Inspired by the error accumulation characteristics for the time-series prediction and RUL prediction, this work formulates a hierarchical optimization scheme for the machine RUL predictions, intending to minimize the error of the RUL prediction by forecasting a few steps ahead and getting a sense of the deterioration trends of the machine in an affordable cost (in terms of the additional error term). With the setup of bi-level optimization problem formulation, the bi-level deep learning scheme is presented where the lower level is to forecast the time-series in the near future, and the upper level is to predict the RULs combining the available time-series and the predicted ones by the lower-level prediction. Due to the unique characteristics in processing time-series and extracting recursive and non-recursive features among them, LSTM networks are applied and the algorithm is proposed for the bi-level RUL prediction. Case studies using PHM08 data challenge data set and four data sets in C-MAPSS package show that the presented method outperforms the state-of-the-art approaches. Also, the proposed bi-level LSTM framework can capture the degradation trends in the time-series, besides predicting the RULs well, which validates the adaptability of the formed hierarchical optimization scheme for the RUL prediction applications.

Future work will pursue (i) transferring the proposed framework from laboratory to practical cases [10], and (ii) more case studies in applying the proposed framework in diverse scenarios.

CRediT authorship contribution statement

Tao Song: Methodology, Validation, Investigation, Writing - original draft, Writing - review & editing. **Chao Liu:** Methodology, Validation, Investigation, Conceptualization, Formal analysis,

Writing - original draft, Writing - review & editing. **Rui Wu:** Investigation, Writing - review & editing. **Yunfeng Jin:** Investigation, Writing - review & editing. **Dongxiang Jiang:** Supervision, Investigation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by National Key R&D Program of China (Grant No. 2019YFF0216104, 2019YFF0216103) and National Natural Science Foundation of China (Grant No. 11802152).

References

[1] X. Li, W. Zhang, H. Ma, Z. Luo, X. Li, Data alignments in machinery remaining useful life prediction using deep adversarial neural networks, Knowledge-Based Systems 105843 (2020).
[2] J. Ma, S. Xu, P. Shang, W. Qin, Y. Cheng, C. Lu, Y. Su, J. Chong, H. Jin, Y. Lin, et al., Cycle life test optimization for different li-ion power battery formulations using a hybrid remaining-useful-life prediction method, Applied Energy 262 (2020) 114490.
[3] S. Haidong, C. Junsheng, J. Hongkai, Y. Yu, W. Zhantao, Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing, Knowledge-Based Systems 188 (2020) 105022.
[4] D. Jiang, C. Liu, Machine condition classification using deterioration feature extraction and anomaly determination, IEEE Transactions on Reliability 60 (2011) 41–48.
[5] U.D. Kumar, J. Knezevic, J. Crocker, Maintenance free operating period–an alternative measure to mtbf and failure rate for specifying reliability?, Reliability Engineering & System Safety 64 (1999) 127–131.
[6] Dixon, Matthew, Maintenance costs of aging aircraft, Costs of Aging Aircraft Insights from Commercial Aviation (2006).
[7] Y. Jin, C. Liu, X. Tian, H. Huang, G. Deng, Y. Guan, D. Jiang, A hybrid model of lstm neural networks with thermodynamic model for condition-based maintenance of compressor fouling, Measurement Science and Technology (2021).
[8] C. Okoh, R. Roy, J. Mehnen, L. Redding, Overview of remaining useful life prediction techniques in through-life engineering services, Procedia Cirp 16 (2014) 158–163.
[9] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to rul prediction, Mechanical systems and signal processing 104 (2018) 799–834.
[10] Y. Jiang, S. Yin, J. Dong, O. Kaynak, A review on soft sensors for monitoring, control and optimization of industrial processes, IEEE Sensors Journal (2020).
[11] Y. Jiang, S. Yin, O. Kaynak, Performance supervised plant-wide process monitoring in industry 4.0: A roadmap, IEEE Open Journal of the Industrial Electronics Society (2020b).
[12] H. Wang, C. Liu, D. Jiang, Z. Jiang, Collaborative deep learning framework for fault diagnosis in distributed complex systems, Mechanical Systems and Signal Processing 156 (2021) 107650.
[13] T. Han, C. Liu, W. Yang, D. Jiang, A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults, Knowledge-Based Systems 165 (2019) 474–487.
[14] J. Guo, C. Liu, J. Cao, D. Jiang, Damage identification of wind turbine blades with deep convolutional neural networks, Renewable Energy 174 (2021) 122–133.
[15] C. Liu, K.G. Lore, Z. Jiang, S. Sarkar, Root-cause analysis for time-series anomalies via spatiotemporal graphical modeling in distributed complex systems, Knowledge-Based Systems 211 (2021) 106527.
[16] C.-G. Huang, H.-Z. Huang, Y.-F. Li, A Bidirectional LSTM Prognostics Method Under Multiple Operational Conditions, IEEE Transactions on Industrial Electronics 66 (2019) 8792–8802.
[17] Z. Jiang, C. Liu, B. Ganapathysubramanian, D.J. Hayes, S. Sarkar, Predicting county-scale maize yields with publicly available data, Scientific Reports 10 (2020) 1–12.
[18] B. Yang, R. Liu, E. Zio, Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture, IEEE Transactions on Industrial Electronics 66 (2019) 9521–9530.
[19] Y. Jin, C. Liu, X. Tian, H. Huang, G. Deng, Y. Guan, D. Jiang, A novel integrated modeling approach for filter diagnosis in gas turbine air intake system, Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy (2021), 09576509211044392.
[20] Y. Jin, C. Qin, Y. Huang, W. Zhao, C. Liu, Multi-domain modeling of atrial fibrillation detection with twin attentional convolutional long short-term memory neural networks, Knowledge-Based Systems 105460 (2020).
[21] Y. Liu, C. Yang, K. Huang, W. Gui, Non-ferrous metals price forecasting based on variational mode decomposition and lstm network, Knowledge-Based Systems 188 (2020) 105006.
[22] J. Zhang, P. Wang, R. Yan, R.X. Gao, Long short-term memory for machine remaining life prediction, Journal of manufacturing systems 48 (2018) 78–86.
[23] J. Lei, C. Liu, D. Jiang, Fault diagnosis of wind turbine based on long short-term memory networks, Renewable Energy 133 (2019) 422–432.
[24] H. Miao, B. Li, C. Sun, J. Liu, Joint Learning of Degradation Assessment and RUL Prediction for Aeroengines via Dual-Task Deep LSTM Networks, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS 15 (2019) 5023–5032.
[25] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, X. Li, Machine remaining useful life prediction via an attention based deep learning approach, IEEE Transactions on Industrial Electronics (2020).
[26] M. Wozniak, J. Silka, M. Wieczorek, M. Alrashoud, Recurrent neural network model for iot and networking malware threat detection, IEEE Transactions on Industrial Informatics 17 (2021) 5583–5594.
[27] M. Woźniak, M. Wieczorek, J. Siłka, D. Połap, Body pose prediction based on motion sensor data and recurrent neural network, IEEE Transactions on Industrial Informatics 17 (2020) 2101–2111.
[28] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder, arXiv preprint arXiv:1608.06154 (2016).
[29] M. Yuan, Y. Wu, L. Lin, Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network, in: 2016 IEEE International Conference on Aircraft Utility Systems (AUS), 2016, pp. 135–140. 10.1109/AUS.2016.7748035.
[30] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 international conference on prognostics and health management, IEEE, 2008, pp. 1–9.
[31] A. Saxena, K. Goebel, Nasa prognostics respository, https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/, 2008.
[32] M.A. Chao, C. Kulkarni, K. Goebel, O. Fink, Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics, Data 6 (2021) 5.
[33] Y. Wu, M. Yuan, S. Dong, L. Lin, Y. Liu, Remaining useful life estimation of engineered systems using vanilla lstm neural networks, Neurocomputing 275 (2018) 167–179.
[34] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[36] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, CoRR abs/1406.1078 (2014).
[37] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, 2015, pp. 802–810.
[38] F.O. Heimes, Recurrent neural networks for remaining useful life estimation, in: 2008 international conference on prognostics and health management, IEEE, 2008, pp. 1–6.
[39] P. Lim, C.K. Goh, K.C. Tan, A time window neural network based framework for remaining useful life estimation, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 1746–1753.
[40] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, in: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2017, pp. 88–95.
[41] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. arXiv:1406.1078.
[42] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliability Engineering & System Safety 172 (2018) 1–11.
[43] G.S. Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: International conference on database systems for advanced applications, Springer, 2016, pp. 214–228.
[44] Z.-Q. Wang, C.-H. Hu, X.-S. Si, E. Zio, Remaining useful life prediction of degrading systems subjected to imperfect maintenance: Application to draught fans, Mechanical Systems and Signal Processing 100 (2018) 802–813.
[45] X. Jia, H. Cai, Y. Hsu, W. Li, J. Feng, J. Lee, A novel similarity-based method for remaining useful life prediction using kernel two sample test, in: Proceedings of the Annual Conference of the PHM Society, volume 11, 2019.
[46] H. Miao, B. Li, C. Sun, J. Liu, Joint learning of degradation assessment and rul prediction for aeroengines via dual-task deep lstm networks, IEEE Transactions on Industrial Informatics 15 (2019) 5023–5032.
[47] T. Song, C. Liu, D. Jiang, A novel framework for machine remaining useful life prediction based on time series analysis, in: 2019 Prognostics and System Health Management Conference (PHM-Qingdao), IEEE, 2019, pp. 1–6.

Tao Song is currently working toward the B.S. degree in energy and power engineering at Tsinghua University, Beijing, P.R. China. His research interests include machinery condition monitoring, intelligent fault diagnosis and prognostics.

Chao Liu received the B.Sc. degree from Huazhong University of Science and Technology, Wuhan, China, in 2008, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2013. He was a postdoctoral researcher with Tsinghua University, Beijing, China, from 2013 to 2015. Currently, he is a research assistant professor with Department of Energy and Power Engineering, Tsinghua University. His research interests are in dynamics, machine learning and health monitoring in machinery and cyber-physical systems.

Rui Wu is currently working toward the Ph.D. degree in energy and power engineering at Tsinghua University, Beijing, P.R. China. He received the B.S. degree in energy and power engineering from Tsinghua University in 2018. His research interests include machinery condition monitoring, signal processing, and remaining useful life prediction of rotating machinery.

Yunfeng Jin received the B.S. degree in energy and power engineering from Tsinghua University, Beijing, China, in 2017. He is currently working toward the master degree in power engineering and engineering thermophysics at Tsinghua University, Beijing, China. His research interests include prognostics and diagnostics of gas turbine performance.

Dongxiang Jiang is a Professor at the Department of Energy and Power Engineering, Tsinghua University, Beijing, China. He received the B. Sc. degree in Electronic Engineering from the Shenyang University of Technology in 1983, the M. Sc. degree in Electrical Engineering from Harbin Institute of Technology in 1989, and the Ph. D. degree in Astronautics and Mechanics from Harbin Institute of Technology in 1994. He was a postdoctoral researcher at the Department of Thermal Engineering, Tsinghua University, from 1994 to 1996. His research interests include condition monitoring, and diagnostics for rotating machinery, and wind energy.