Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4670982, 9 pages
https://doi.org/10.1155/2019/4670982

Research Article

Prediction Model Design for Vibration Severity of Rotating Machine Based on Sequence-to-Sequence Neural Network

Zhiqiang Wang, Hong Qian, Dongliang Zhang, and Yingchen Wei

Shanghai University of Electric Power, Shanghai 200090, China

Correspondence should be addressed to Hong Qian; qianhong.sh@163.com

Received 15 May 2019; Accepted 31 July 2019; Published 22 September 2019

Academic Editor: Sabri Arik

Steam turbine rotor system is a main part of the power production process. Accurate prediction of the turbine rotor operation state leads to timely detection of the hidden danger and accordingly ensures the efficient power production. The vibration severity reflects the vibration intensity and the working condition as well. Since the accuracy of the normal prediction method is not enough, a new model is proposed in this paper that combines the sequence prediction model with the gated recurrent unit (GRU). According to the obtained results, the accuracy is improved through the proposed model. To verify the effectiveness of the model, simulations are performed on the steam turbine rotor unbalance fault data. The experimental results demonstrate that the proposed approach could be utilized for vibration severity prediction as well as state warning of the steam turbine.

1. Introduction

Working condition prediction provides various benefits for rotating machine maintenance. It is known as the main way for early indication of the hidden danger and providing the overhaul reference. The accurate prediction improves the safety level of the rotating machine [1–3].

At present, the statistical prediction methods and artificial intelligence-based methods like artificial neural networks have been commonly employed for the trend analysis. In a spectrum analysis-based approach, a gray prediction model has been employed to forecast the vibration severity with a small amount of the steam turbine vibration data in a data reorganization form [4]. The ensemble empirical mode decomposition (EEMD) method has been adopted for the vibration data analysis and obtaining the intrinsic mode function (IMF). Although the prediction model has been employed to predict the IMF value for obtaining the dynamic vibration data [5], this model is not suitable for timing prediction. In another work, the information fusion has been combined with the back-propagation neural network (BP neural network) to construct a prediction model. In the mentioned neural network, the data measured by the sensor were considered as its input and its output was applied to the decision fusion model to obtain the final prediction results [6]. Although the BP neural network is always utilized to build a predictive model, it could not work with the time-series data. The support vector machine (SVM) method has been employed to construct this prediction model for every component processed by the EMD. Moreover, different corn functions with different components have been adopted to acquire the final prediction results [7]. In [8], a bidirectional LSTM (Bi-directional Long-Short Term Memory, BI-LSTM) unit has been employed by a deep learning method for data prediction. However, due to its complex structure with too many parameters, its convergence speed is slow.

In this paper, a gated recurrent unit- (GRU-) based sequence prediction model is proposed for prediction and analysis of the vibration severity. Although accurate classification could be obtained through traditional deep learning methods like BP and SVM, their data prediction capability for time series is not acceptable. Since the sequence prediction model with GRU has the memory cell, it can solve the gradient disappearance problem. The obtained results reflect the development trend of the vibration severity and meet certain precision requirements.

2. Materials and Methods

The prediction system structure is shown in Figure 1. The details of the prediction model and vibration severity are illustrated in the following.

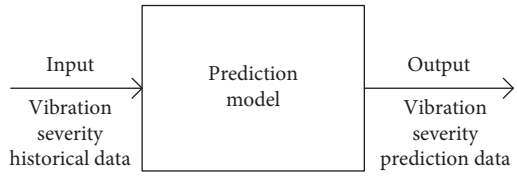

Figure 1: The prediction system structure.

2.1. The GRU-Seq2Seq Model Structure. The GRU-Sequence-to-Sequence (GRU-Seq2Seq) model is a variation of the recurrent neural networks (RNN) [9]. The main difference between this model and the RNN is that input and output sequences in the RNN have similar lengths while the Seq2Seq model allows different lengths for these sequences. In this paper, the base frequency amplitude for the past month is adopted by the model to predict the next week data. This model is called the encoder-decoder model, and its structure is shown in Figure 2.

In Figure 2, $x(1) \sim x(t)$ and $y(1) \sim y(n)$ are the input and output sequences, respectively. ⟨EOS⟩ is a signal that is employed to start the decoder part.

This model contains the following four different layers:

(1) The input layer: The historical data $X \in [x(k), x(k+1), \ldots, x(t)]$, $x(k) \in R^{1\times 1}$ denote the model input where $x(k)$ is the input signal at time k.

(2) The encoder layer: The dynamic equations of the encoder layer are given as

$$h = \begin{bmatrix} h_t^{(1)} \\ h_t^{(2)} \\ \vdots \\ h_t^{(u)} \end{bmatrix} = \left(1 - \sigma\left(\begin{bmatrix} W_{z(1)} \\ W_{z(2)} \\ \vdots \\ W_{z(u+1)} \end{bmatrix} \bullet \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \\ x(k) \end{bmatrix} + \begin{bmatrix} b_z^{(1)} \\ b_z^{(2)} \\ \vdots \\ b_z^{(u)} \end{bmatrix}\right)\right) \circ \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \end{bmatrix} + \sigma\left(\begin{bmatrix} W_{z(1)} \\ W_{z(2)} \\ \vdots \\ W_{z(u+1)} \end{bmatrix} \bullet \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \\ x(k) \end{bmatrix} + \begin{bmatrix} b_z^{(1)} \\ b_z^{(2)} \\ \vdots \\ b_z^{(u)} \end{bmatrix}\right) \circ \begin{bmatrix} \widetilde{h}_{t(1)} \\ \widetilde{h}_{t(2)} \\ \vdots \\ \widetilde{h}_{t(u)} \end{bmatrix}. \tag{1}$$

The GRU of this layer encodes the input $X \in R^{1\times t}$ to the hidden state $h \in R^{u\times 1}$:

$$\widetilde{h} = \begin{bmatrix} \widetilde{h}_{t(1)} \\ \widetilde{h}_{t(2)} \\ \vdots \\ \widetilde{h}_{t(u)} \end{bmatrix} = \tanh\left(\begin{bmatrix} W_{\widetilde{h}(1)} \\ W_{\widetilde{h}(2)} \\ \vdots \\ W_{\widetilde{h}(u+1)} \end{bmatrix} \bullet \left(\left(\sigma\left(\begin{bmatrix} W_{r(1)} \\ W_{r(2)} \\ \vdots \\ W_{r(u+1)} \end{bmatrix} \bullet \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \\ x(k) \end{bmatrix} + \begin{bmatrix} b_r^{(1)} \\ b_r^{(2)} \\ \vdots \\ b_r^{(u)} \end{bmatrix}\right)\right)\right) + \begin{bmatrix} b_{\widetilde{h}}^{(1)} \\ b_{\widetilde{h}}^{(2)} \\ \vdots \\ b_{\widetilde{h}}^{(u)} \end{bmatrix}\right), \tag{2}$$

where $W \in R^{1\times u}$, $W = (W_{z(i)}, W_{r(i)}, W_{\widetilde{h}(i)})$ and $b \in R^{1\times 1}$, $b = (b_z^{(i)}, b_r^{(i)}, b_{\widetilde{h}}^{(i)})$ are the weights and the biases of the upgrade, reset, and memory cell gate, respectively; $h_{t-1}^{(i)} \in R^{1\times 1}$ is the hidden layer output at the last time; $x(t) \in R^{1\times 1}$ is the input at time t; $\widetilde{h}_t^{(i)} \in R^{u\times 1}$ is the memory cell at time t; $i \in [1, u]$; σ is the sigmoid activation function; and u is the number of hidden units.

h_{end} is the hidden state of the encoder layer at time t_{end} and is considered as the decoder layer input.

(3) The decoder layer:

$$\widehat{h}(j) = \begin{bmatrix} \widehat{h}_t^{(1)}(j) \\ \widehat{h}_t^{(2)}(j) \\ \vdots \\ \widehat{h}_t^{(u)}(j) \end{bmatrix} = \left(1 - \sigma\left(\begin{bmatrix} W_{z(1)} \\ W_{z(2)} \\ \vdots \\ W_{z(u+1)} \end{bmatrix} \bullet \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} b_z^{(1)} \\ b_z^{(2)} \\ \vdots \\ b_z^{(u)} \end{bmatrix}\right)\right) \circ \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \end{bmatrix} + \sigma\left(\begin{bmatrix} W_{z(1)} \\ W_{z(2)} \\ \vdots \\ W_{z(u+1)} \end{bmatrix} \bullet \begin{bmatrix} h_{t-1}^{(1)} \\ h_{t-1}^{(2)} \\ \vdots \\ h_{t-1}^{(u)} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} b_z^{(1)} \\ b_z^{(2)} \\ \vdots \\ b_z^{(u)} \end{bmatrix}\right) \circ \begin{bmatrix} \widetilde{h}_{t(1)} \\ \widetilde{h}_{t(2)} \\ \vdots \\ \widetilde{h}_{t(u)} \end{bmatrix}, \tag{3}$$

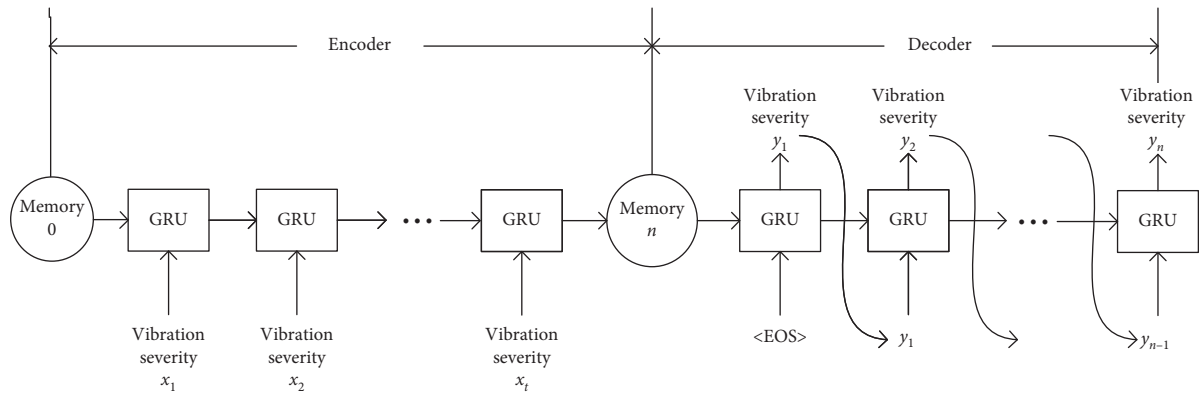

Figure 2: The GRU-Seq2Seq model structure.

where $y_{t-1} \in R^{1\times 1}$ is the encoder layer output at the last time; $\widehat{h}(j) \in R^{u\times 1}$ is the hidden state of the decoder layer at time j. To realize the recurrent forecasting, the decoder layer input is considered as the decoder layer output at the last time, or $h_{t-1} = h_{\text{end}}$, $y_{t-1} = \langle \text{EOS} \rangle$ at the time $j = 1$. The decoder layer starts to decode when it receives the signal $\langle \text{EOS} \rangle$.

(4) The output layer:

$$y_t = \tanh\left(W_{y(t)} \bullet \widehat{h}(t) + b_y^{(i)}\right), \tag{4}$$

where $\widehat{h}(t) \in R^{u\times m}$ is the hidden state of the decoder layer at time t; y_t is the output at time t; $W_{y(t)} \in R^{1\times u}$ and $b_y^{(i)} \in R^{1\times 1}$ are the weight and bias of the output layer, respectively. The final output sequence of the model is denoted by $Y = [y_1, y_2, \ldots, y_m] \in R^{1\times m}$.

Operator definition: $\bullet$ denotes the direct multiplication of the matrix; $\circ$ represents the multiplication of the corresponding elements.

2.2. The Basic Unit of GRU-Seq2Seq. The GRU as a variation of the RNN unit is the basic unit employed in this paper. The gradient disappearance happens when the normal RNN unit processes the time-series data [10] while the GRU avoids this problem by adding the gate. It catches the relationship between different time-series data [11].

The GRU structure is shown in Figure 3.

This structure contains three different gates including the upgrade gate, the reset gate, and the memory gate. The upgrade gate output denoted by z_t determines how much information could be transmitted from the last state h_{t-1} to the next state h_t. The reset gate output denoted by r_t determines the importance of the last state h_{t-1} to the next state $\widetilde{h}_t$. In other words, if this value is equal to zero, the information will not be conveyed to the next memory cell $\widetilde{h}_t$. The memory gate output denoted by $\widetilde{h}_t$ is a combination of the input x_t at the current time and the state h_{t-1} at the last time. The new vector $\widetilde{h}_t$ includes the information of the last sequence and the input at this time.

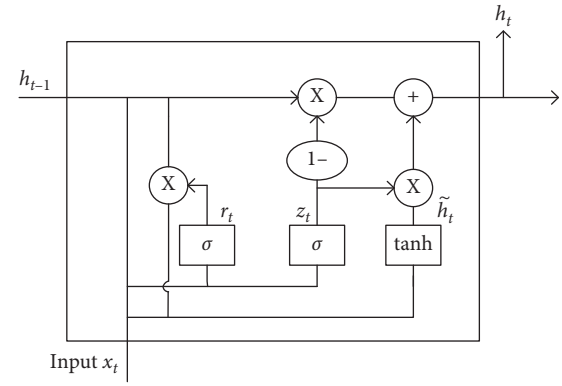

Figure 3: GRU structure diagram.

The operation of the mentioned three gates is illustrated as follows:

(1) The upgrade gate function is described as

$$z_t = \sigma\left(W_z \bullet \left[h_{t-1}, x_t\right] + b_z\right). \tag{5}$$

(2) The reset gate function is described as

$$r_t = \sigma\left(W_r \bullet \left[h_{t-1}, x_t\right] + b_r\right). \tag{6}$$

(3) The memory gate function could be described as

$$\widetilde{h}_t = \tanh\left(W_{\widetilde{h}} \bullet \left[r_t \circ h_{t-1}, x_t\right] + b_c\right). \tag{7}$$

The input at time t could be converted to the hidden state h_t where h_t is the input at the next time:

$$h_t = \left(1 - z_t\right) \circ h_{t-1} + z_t \circ \widetilde{h}_t. \tag{8}$$

In equations (5)–(8), z_t, r_t, $\widetilde{h}_t$, $h_t \in R^{u\times 1}$ are the corresponding outputs of these three gates; W_z, W_r, $W_{\widetilde{h}} \in R^{u\times u+1}$; b_z, b_r, $b_{\widetilde{h}} \in R^{u\times 1}$ are the weights and biases of these three gates, respectively.

2.3. Model Training. The loss function could be defined as

$$J(W,b) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widetilde{y}_i\right)^2 + \frac{\lambda}{2n}\|W\|_2^2, \tag{9}$$

where y_i and $\widetilde{y}_i$ are the predicted and real outputs, respectively; W and b are the weight and the bias of the output, respectively; λ is the regularization parameter; α is the learning rate; and n is the length of the output vector.

The details of the gradient descent method as the training algorithm are given as follows:

(1) Calculation of the weight gradient:

The weight gradient of the upgrade gate is obtained as

$$dW_z = \left[dh_{\text{next}} \circ h_{\text{prev}} \circ (-1) + dh_{\text{next}} \circ \widetilde{h}_t\right] \circ z_t \circ (1 - z_t) \circ \left[h_{prev}, x_t\right]. \tag{10}$$

The weight gradient of the memory gate could be calculated as

$$dW_{\widetilde{h}} = dh_{\text{next}} \bullet z_t \circ \left(1 - \widetilde{h}_t^2\right) \circ \left[h_{\text{prev}}, x_t\right]. \tag{11}$$

The weight gradient of the reset gate is computed as

$$dW_r = d\widetilde{h}_t \circ h_{\text{prev}} \circ r_t \circ (1 - r_t) \circ \left[h_{\text{prev}}, x_t\right]. \tag{12}$$

In relations (10)–(12), $dh_{\text{next}} \in R^{u\times 1}$ is the state gradient at the last time; $h_{\text{prev}} \in R^{u\times 1}$ is the input state at the last time; dz_t, dr_t, $d\widetilde{h}_t \in R^{u\times 1}$ are the corresponding gradients of these three gates; $\alpha \in R^{1\times 1}$ is the learning rate; and $\lambda \in R^{1\times 1}$ is the regularization parameter.

(2) Updating the parameters:

The upgrade gate update is described as

$$W_z = W_z\left(1 - \frac{\lambda\alpha}{n}\right) - \alpha\, dW_z. \tag{13}$$

The memory gate could be updated as

$$W_{\widetilde{h}} = W_{\widetilde{h}}\left(1 - \frac{\lambda\alpha}{n}\right) - \alpha\, dW_{\widetilde{h}}. \tag{14}$$

The reset gate is updated as

$$W_r = W_r\left(1 - \frac{\lambda\alpha}{n}\right) - \alpha\, dW_r. \tag{15}$$

(3) The training process is shown in Figure 4. The training loop repeats until the parameters converge to their corresponding expected values.

2.4. The Rolling Prediction. In actual operating conditions, the prediction results are affected by different factors. Thus, the prediction data should be updated to ensure the accuracy of the results. The rolling prediction method is illustrated in Figure 5.

In Figure 5, the prediction model input at this time is denoted by $X \in [x(k), x(k+1), \ldots, x(k+t)]$, $x(k) \in R^{1\times 1}$ where $x(k)$ is the input at time k; the prediction output is denoted by $Y = [y(k), y(k+1), \ldots, y(m)]$ where $y(m)$ is the output at time m. After the model acquires the real output $x(k+t+1)$, the input and output of the model at this time are described by $\widetilde{X} = [x(k+1), x(k+2), \ldots, x(t+1)]$ and $\widetilde{Y} = [y(k+1), y(k+2), \ldots, y(m+1)]$, respectively.

2.5. The Vibration Severity Analysis. The vibration severity is extracted from the unbalance fault model. Then, the prediction accuracy of the model is evaluated. The GRU-Seq2Seq model flowchart employed in the steam turbine vibration severity analysis is shown in Figure 6.

2.6. Extracting Data from the Mathematical Model. Since the steam turbine rotor system operates in the high-speed working state for a long time, the incidence of the unbalance fault reaches 70%. The mathematical model [12] of this fault is described as

$$\begin{cases} m\ddot{x} + c_1\dot{x} + k(x - x_b) + k_s\left((x - x_b)^2 + (y - y_b)^2\right)(x - x_b) = me\omega^2\cos(wt), \\ m\ddot{y} + c_1\dot{y} + k(y - y_b) + k_s\left((x - x_b)^2 + (y - y_b)^2\right)(y - y_b) = me\omega^2\sin(wt) - mg, \\ m_b\ddot{x}_b + c_b\dot{x}_b + k(x_b - x) + k_s\left((x_b - x)^2 + (y_b - y)^2\right)(x_b - x) = F_x, \\ m_b\ddot{y}_b + c_b\dot{y}_b + k(y_b - y) + k_s\left((x_b - x)^2 + (y_b - y)^2\right)(y_b - y) = F_y - m_bg, \end{cases} \tag{16}$$

where e is the mass eccentricity; m is the equivalent mass of the rotor and disk; k, k_s are the linear and nonlinear stiffness coefficients of the rotating shaft, respectively; c_b, c_1 denote the damping coefficients of the rotor at bearing and disk, respectively; x_b and y_b are the horizontal and vertical displacements, respectively.

The mass eccentricity variation can simulate the unbalance fault level. The relationship between the mass eccentricity and time is described as e_1:

$$e_1 = 1.8 \times 10^{-4} + 1.15 \times 10^{-19}t^2. \tag{17}$$

The mass eccentricity variation is related to the actual working condition. To make the prediction result closer to

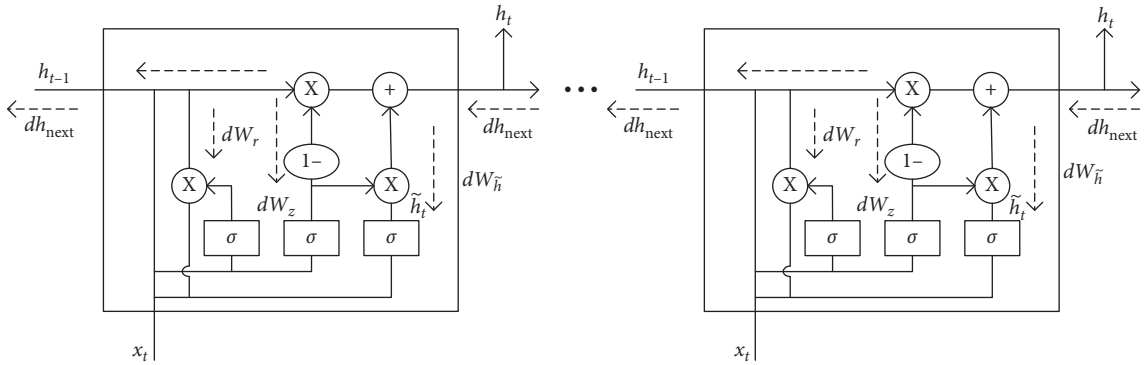

Figure 4: Model training process diagram.

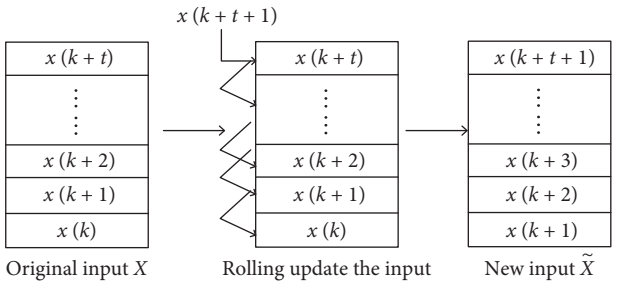

Figure 5: Scroll prediction diagram.

the actual effect, the following time-varying mass eccentricity (denoted by e_2) is proposed:

$$e_2 = 1.8 \times 10^{-4} + 2.9614 \times 10^{-15} t + 1.4 \times 10^{-19} t^2 + 1.15 \times 10^{-40} t^3. \quad (18)$$

The corresponding units of the unbalance fault and eccentricity are meters and seconds, respectively. The formulas (17) and (18) are brought into the fault model (16). Moreover, the Runge–Kutta method as a conventional method for solving differential equations is employed to obtain the horizontal displacement vibration data of the turbine rotor. By considering the sampling time as 1 sec and the sampling interval as 6 hours, the continuous sampling for one year is performed to obtain a matrix containing 1440 sets of the vibration displacement data and storage.

2.7. The Displacement Vibration Data Processing. According to the previous section, the displacement vibration data could be obtained from the mathematical model of the unbalance fault given in (16). To extract the useful data from the total data and employ it as the input of the GRU-Seq2Seq prediction model, more processing is required.

The spectrum plot is extracted from the time domain displacement vibration data by using the fast Fourier transformation. The vibration data and the spectrum plot are shown in Figures 7 and 8, respectively.

Vibration severity contains all the frequency information and reflects the vibration intensity. Since the vibration

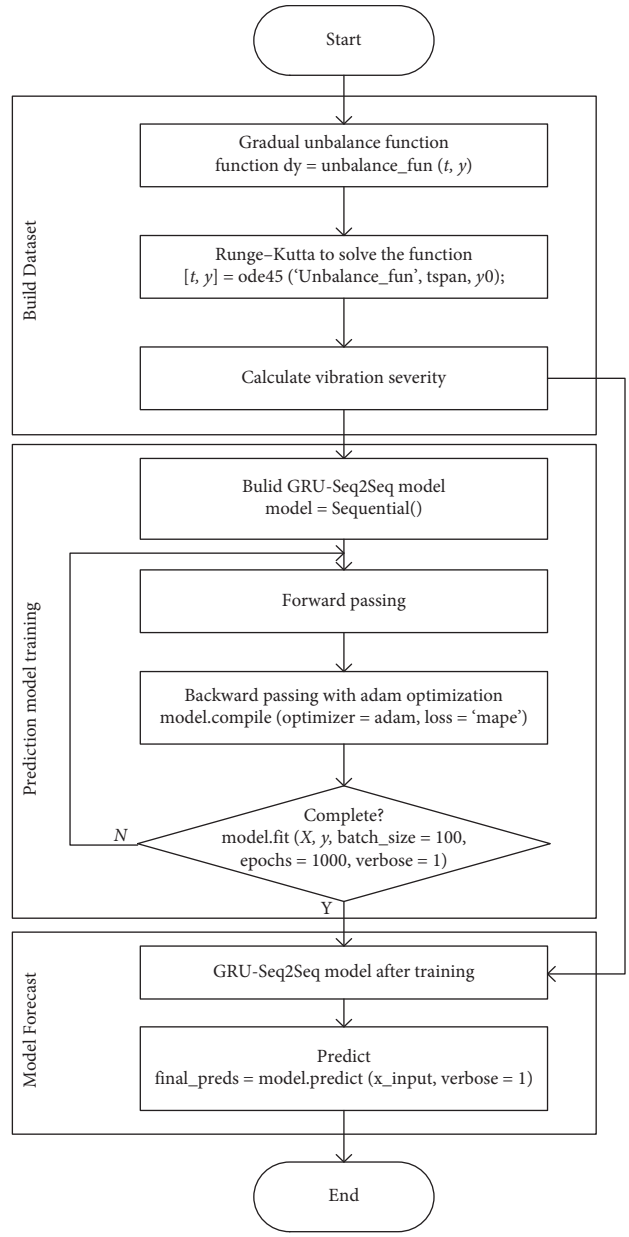

Figure 6: The vibration severity prediction flowchart.

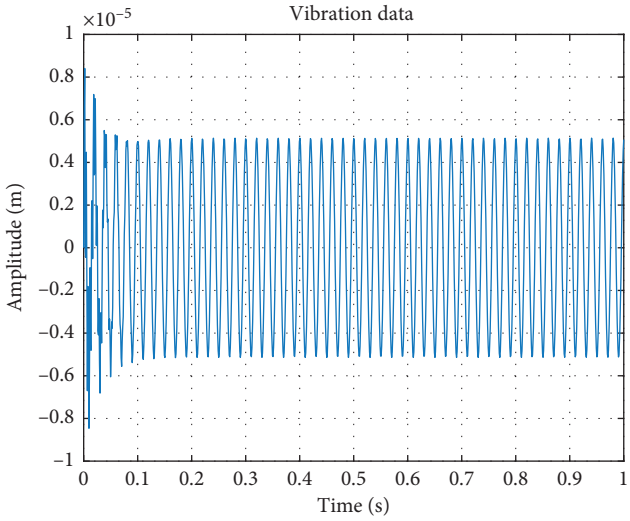

Figure 7: The vibration data.

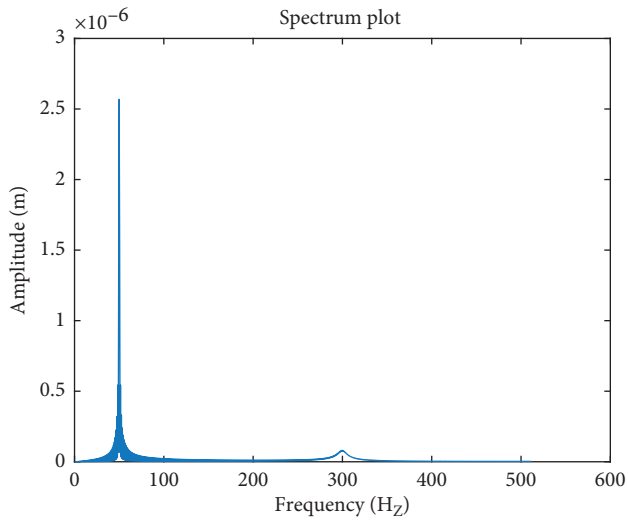

Figure 8: The spectrum plot.

severity is a representation of the energy, it is known as an important parameter for estimating the working condition of the turbine rotor system. By predicting it, the engineer could detect the rotor system faults and prevent the hidden danger.

For each signal $x(t)$, its power is defined as

$$P = \frac{1}{T}\int_0^T x(t)^2 dt. \tag{19}$$

The vibration severity function could be written as

$$V_{\text{ims}} = \sqrt{\frac{1}{T}\int_0^T x(t)^2 dt}. \tag{20}$$

Comparing relations (19) and (20), gives

$$P = V_{\text{ims}}^2. \tag{21}$$

Consider that $x(t)$ is a periodic signal satisfying the Dirichlet conditions. Now, it could be described with the following Fourier series expansion:

$$x(t) = a_0 + \sum_{n=1}^{\infty} A_n \sin\left(n\omega_0 t + \phi\right). \tag{22}$$

According to relations (19) and (22), we have

$$P = \frac{1}{T_0}\int_0^{T_0} \left[a_0 + A_n \sin\left(n\omega_0 t + \phi\right)\right]^2 dt = a_0 + \frac{1}{2}\sum_{n=1}^{\infty} A_n^2. \tag{23}$$

Thus, the vibration function could be written as

$$V_{\text{ims}} = \sqrt{\frac{1}{2}\sum_{ka}^{kb} \left(2\pi f_k A_k\right)^2} = \sqrt{\frac{1}{2}\sum_{ka}^{kb} \left(2\pi \frac{k f_s}{N}\frac{2}{N}|X(k)|\right)^2}$$
$$= \frac{2\sqrt{2}\pi f_s}{N^2}\sqrt{\sum_{ka}^{kb} \left(k|X(k)|\right)^2}, \tag{24}$$

where $x(n)$ is the vibration displacement signal.

$$V_{\text{ims}} = \sqrt{\frac{1}{2}\sum_{ka}^{kb} A_k^2} = \sqrt{\frac{1}{2}\sum_{ka}^{kb} \left(\frac{2}{N}|X(k)|\right)^2} = \frac{1}{N}\sqrt{\sum_{ka}^{kb} |X(k)|^2}, \tag{25}$$

where $x(n)$ is the vibration speed signal. ka, kb are the frequency boundaries; A_k is the amplitude of the frequency k; and N is the number of the sample points.

Equation (16) could be employed for finding the vibration severity. The obtained vibration severities for e_1 and e_2 are shown in Figures 9 and 10, respectively.

The vibration severity is considered as the prediction model input and is denoted by matrix $\tilde{X} \in R^{1\times 1440}$. In this paper, the last month (120 data) data are employed to predict the next week data (28 data). To validate the accuracy of the results, 80% and 20% of the data are considered as the training and test sets, respectively.

The vibration severity for e_1 and e_2 are shown in Tables 1 and 2. More details about data of e_1 and e_2 could be found in the Data Availability section.

3. Results and Discussion

The root mean square error (RMSE) and mean absolute percentage error (MAPE) are chosen as the criteria to evaluate the accuracy of the prediction model.

By constructing the GRU-Seq2Seq rolling prediction model with two steps, the results shown in Figures (11)–(14) are obtained. First and second rolling prediction results for e_1 are shown in Figures 11 and 12, respectively.

The obtained criteria for the first step prediction are given as RMSE = 0.01210 and MAPE = 0.2570%. Moreover, the obtained maximum deviation is 0.01983.

The calculated criteria for the second step prediction are given as RMSE = 0.01258 and MAPE = 0.2631%. The calculated value for the maximum deviation is 0.02102.

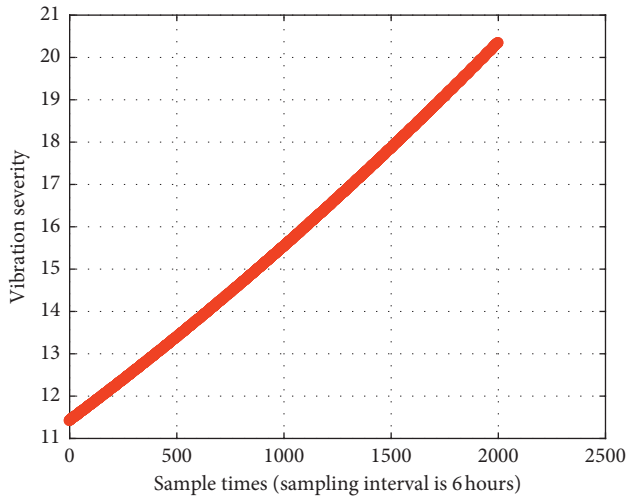

Figure 9: The vibration severity for e_1.

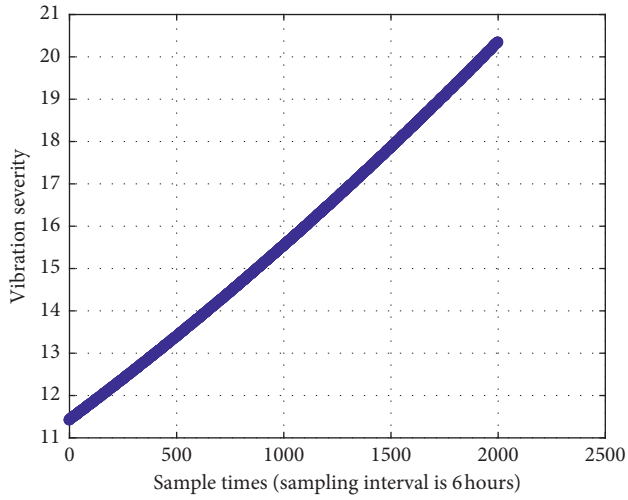

Figure 10: The vibration severity for e_2.

Table 1: The vibration severity for e_1.

Vibration severity (mm/s)	11.4110639 ...	20.3362240 ...	25.8401974

Table 2: The vibration severity for e_2.

Vibration severity (mm/s)	11.4149997 ...	21.6064546 ...	25.8455654

First and second rolling prediction results for e_2 are presented in Figures 13 and 14, respectively.

In this case, the following criteria are obtained for the first step prediction: RMSE = 0.00617 and MAPE = 0.1204%. In addition, the maximum derivation is obtained as 0.01453.

The corresponding criteria for the second step prediction are RMSE = 0.00676 and MAPE = 0.1324%. Moreover, the maximum derivation in this case is 0.01483.

According to these results, the output accuracy is good enough to satisfy the standard of the vibration severity monitoring. To avoid overfitting, EarlyStopping

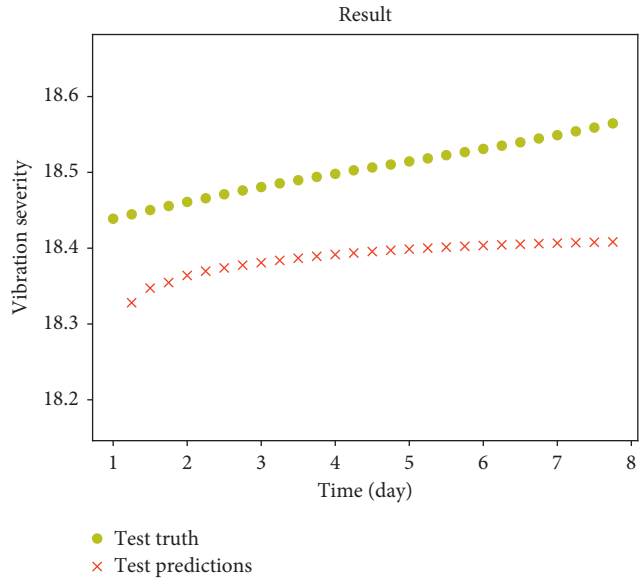

Figure 11: First rolling prediction results for e_1.

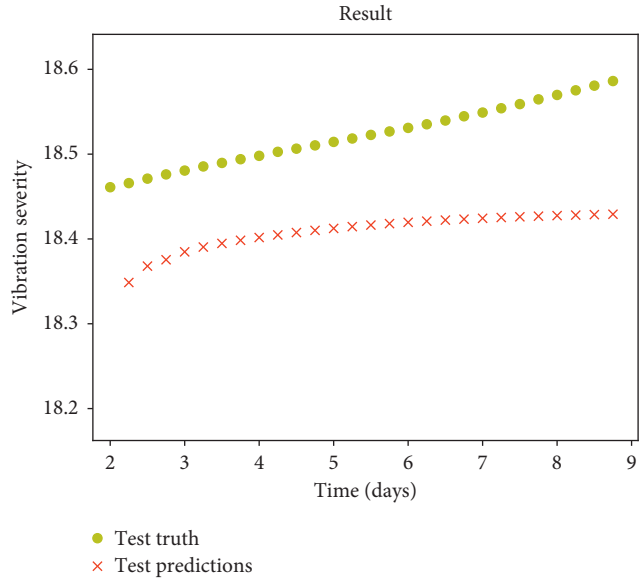

Figure 12: Second rolling prediction results for e_1.

is utilized. This means that the performance indicators are continuously monitored in each iteration to stop training if the training accuracy is satisfactory. The mean square error (MSE) is employed as the performance indicator in the EarlyStopping procedure. By choosing e_1, the convergence time and the obtained results are compared with BP and LSTM-Seq2Seq models (as shown in Table 3).

It is obvious that the rolling prediction results for these three models are stable while higher accuracy could be obtained with the GRU-Seq2seq model. Moreover, this model gives a considerable improvement in the prediction accuracy compared with the BP and LSTM-Seq2Seq models. It is obvious that the GRU-Seq2seq model provides higher prediction accuracy and is more suitable for

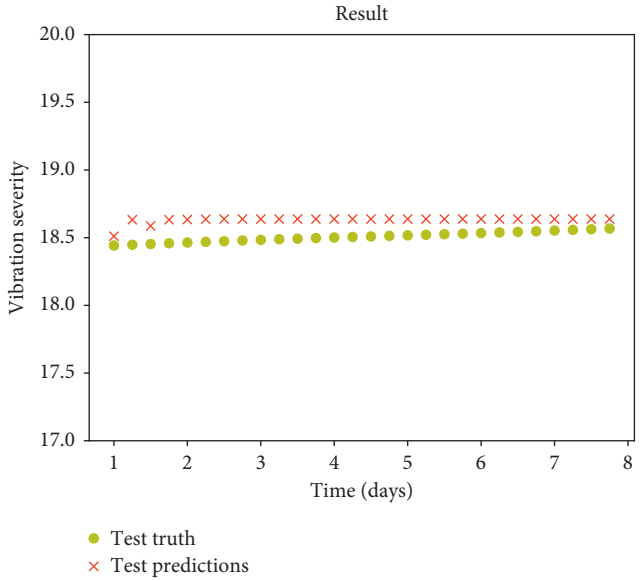

Figure 13: First rolling prediction results for e_2.

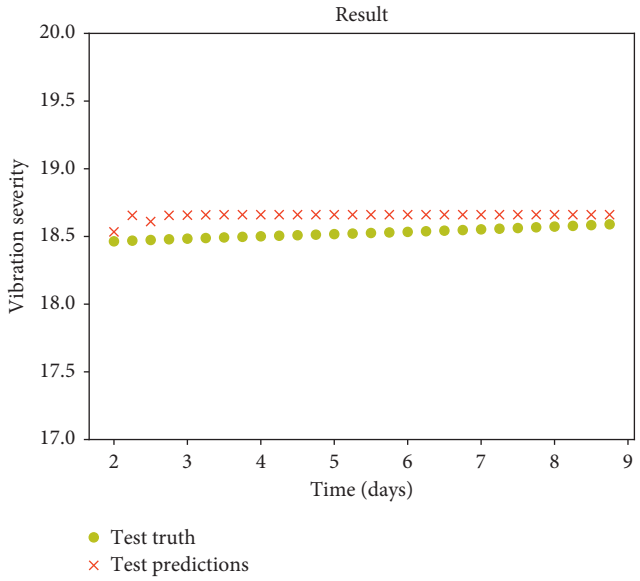

Figure 14: Second rolling prediction results for e_2.

Table 3: Comparison between the BP, LSTM-Seq2Seq, and the GRU-Seq2seq models.

Model	RMSE (μm)		MAPE (%)	MSE (%)	Convergence time (s)
BP	First	2.94314	32.441	30.0617	2
	Second	2.96978	32.426		
LSTM-Seq2seq	First	0.11228	1.1799	2.7626	268
	Second	0.11544	1.2147		
GRU-Seq2seq	First	0.00617	0.1204	0.4948	70
	Second	0.00676	0.1324		

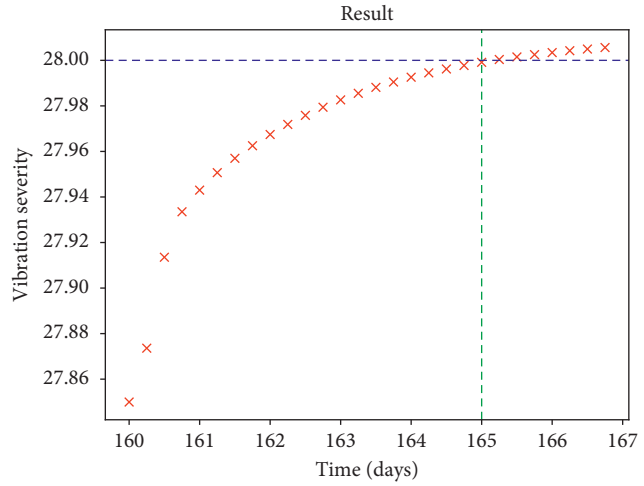

Figure 15: Prediction running until exceeding the ISO standard.

data prediction of the precision machinery such as steam turbines.

According to the ISO 2372 standard, the vibration severity values that are higher than 28.0(mm/s) are not acceptable for steam turbine rotating machines. The rolling prediction results for e_1 show that the time point could be acquired when the vibration severity exceeds the ISO standard (Figure 15).

As could be seen from Figure 15, the rolling prediction task stops as the vibration severity exceeds the ISO standard (28.0(mm/s)). This has occurred after 165 days. By using these prediction results, the working condition of the rotating machine could be estimated and its maintenance could be performed in time.

4. Conclusion

The results prove that the GRU-Seq2Seq prediction model is suitable for data prediction of the vibration severity. Considering the defects of the existing trend analysis algorithms for long-term time-series prediction, a sequence prediction model combined with the GRU is proposed in this paper. This model provides the vibration severity characteristics of the rotating turbine machine. In this paper, the sliding time window is employed to continue the rolling prediction procedure. Accordingly, the possible risks for the rotating machine could be indicated. This facilitates the equipment maintenance. As a future work, the prediction results could be combined with the maintenance to ensure the safe operation of the rotating machines.

Data Availability

The readers can calculate the vibration severities for different values of e. To attain this goal, the required code to calculate the vibration severity from the mathematical model and the GRU-Seq2Seq neural network is available through the following link and password: https://pan.baidu.com/s/1ZltiZdezzj85cuaQHJepQ; password: y0uK. The readers can

access the data after registration. The registration link is https://passport.baidu.com/v2/?reg&tt=1555071708648&overseas=1&gid=B4F5FD0-6269-450C-B929-CD00BF68770D&tpl=netdisk&u=https://pan.baidu.com/disk/home.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors acknowledge the financial support from the Shanghai Science and Technology Commission Local Capacity Building Project.

References

[1] W. C. Laws and A. Muszynska, "Periodic and continuous vibration monitoring for preventive/predictive maintenance of rotating machinery," *Journal of Engineering for Gas Turbines and Power*, vol. 109, no. 2, pp. 159–167, 1987.

[2] K. Madani, "A survey of artificial networks base fault detection and fault diagnosis techniques," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 3442–3446, Washington, DC, USA, July 1999.

[3] R. Isermann, "Process fault detection based on modelling and estimation methods: a survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[4] Y. Feng, *Research on Diagnosis and Early Warning System of Important Rotating Equipment Status and Performance in Nuclear Power Plant*, Shanghai University of Electric Power, Shanghai, China, 2018.

[5] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden Day, San Francisco, CA, USA, 1976.

[6] J. van Amerongen, H. R. van Nauta Lemke, and J. C. T. van der Veen, "Autopilo Tforships designed with Fuzzy sets," in *Proceedings of the 5th IFAC/IFI PConference on Digital Computer Applications to Process Control*, pp. 479–487, The Hague, Netherlands, June 1977.

[7] W. T. James, E. Patrick, M. Sharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and times series model," *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 775–782, 2009.

[8] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *International Journal of Electrical Power & Energy Systems*, vol. 109, pp. 470–479, 2019.

[9] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, http://arxiv.org/abs/1406.1078.

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

[11] J. Chung, C. Gulcehre, K. H. Cho et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, http://arxiv.org/abs/1412.3555.

[12] L. Zhao, *Investigation on Failure Mechanism and Shift Orbits Recognition Methods for Rotating Machine*, Dalian University of Technology, Dalian, China, 2010.