Investigation of Time-series Prediction for Turbine Machinery Condition Monitoring

Chang-Jun Liu¹, Hai-Tao Liu¹, Chao Bian¹, Xu-Dong Chen¹, Shu-Hua Yang¹,², *Xiao-Fang Wang*¹

¹ School of Energy and Power Engineering, Dalian University of Technology, Dalian 116024, China
² Shenyang Blower Works Group Corporation, 110869 Shenyang, China

*Corresponding Author, Email: dlwxf@dlut.edu.cn

Abstract. A running compressor generates a huge amount of data every day. Some of the data contains useful information that allows the user to determine in advance whether the machine will fail or not. Forecasting in advance cannot only avoid the huge loss caused by unnecessary downtime but also ensure the safety of staff. The main tasks of compressor condition monitoring include data acquisition, data preprocessing, data analysis, alarm module, response strategy analysis, etc. Prediction is one of the tasks of condition monitoring. This paper mainly studies the performance of different time prediction models on the running data produced by a centrifugal compressor. In the data preprocessing part, this paper uses Bayesian wavelet denoising and probabilistic principal component analysis methods proposed by the research group to remove the useless parts in the data. In the part of time series model, the prediction performance of LSTM, GRU and ARIMA on the same dataset is compared. The comparative results indicate that since the dataset is non-stationary, the LSTM and GRU outperform the statistical ARIMA model.

1. Introduction

The diagnosis of rotating machinery includes the diagnosis of equipment performance and fault diagnosis of equipment [1]. Complex rotating machinery, such as fans, compressors, steam turbines and gas turbines, is a key production tool in important modern production departments such as petroleum, chemical industry, metallurgy, aerospace and electric power, etc. It is of great significance to carry out performance monitoring and fault diagnosis for the equipment.

With the increasing complexity and reliability requirements of rotating machinery, the traditional operation and maintenance mode has changed from traditional post maintenance and regular maintenance to more efficient condition-based maintenance (CBM) and time-based maintenance (TBM). Rosmani Ahmad et al. summarized the development of CBM [2].

One task of CBM is condition-based monitoring. According to the monitoring principle, it can be divided into non-performance based CBM, which contains oil sample, thermography, strain analysis, vibration, metal temperature, load analysis, acoustics; and performance-based CBM which usually refers to gas analysis. In addition, the monitoring method can be divided into online monitoring and offline monitoring [3].

An important purpose of condition monitoring is to reduce the cost of equipment operation [4]. And the loss caused by the unexpected shutdown is very heavy for an enterprise relying on centrifugal compressor for production. It will not only cause losses, but also cause casualties [5].
If the trend of turbine performance can be predicted in advance, the corresponding strategy can be formulated in advance to avoid unexpected shutdown through the analysis results. Therefore, researchers need a system that can predict the performance of centrifugal compressor units in advance.

With the advent of the era of Engineering 4.0, the storage technology of data is more and more advanced. These massive data bring unprecedented opportunities to the technology based on digital drive. The problem of using deep learning and other methods to predict data has been widely studied. The establishment of deep network is helpful for mining high-order and abstract information in data. When the effective feature representation of data is relatively complete, it can get better results whether it is used for damage classification or prediction regression [6].

Before developing a time series prediction model, we need to pre-process the data due to the possible noise. Si et al. [7] used wavelet packet technology for noise elimination, and took out the characteristic values of signal faults as the sample data for network training and testing. Xu and Liu et al. [8] used Hilbert-Huang transformation to obtain the instantaneous frequency of the decomposed signal, and calculated the statistical feature set of the basic mode component and the instantaneous frequency. Xu and Jiang et al. [9] proposed a probabilistic signal processing method consisting of discrete wavelet packet transform, Bayesian hypothesis test and probabilistic principal component analysis.

After data pre-processing, we turn to build up the time series prediction model. In recent years, Bottieau et al. [10] used LSTM to estimate the impact of prediction accuracy on the ability of wind turbines, Yin et al. [11] proposed a fault diagnosis method for wind turbine gearboxes based on optimized LSTM neural networks with cosine loss. The above researches mainly focus on the analysis of vibration signals of wind turbines by using neural network models, while the data generated by compressors and other forms of data other than vibration are rarely studied. Zhao et al. [12] used the LSTM neural network to predict the dynamic response of the rotating machinery under external excitation, and obtained the health probability of the machinery by using residual data between sensors and model prediction through Bayesian method. However, this study did not compare the performance of LSTM model with other models. In view of the lack of horizontal comparison between models and the lack of experiments on compressor data, this paper makes a comparative analysis of three different models on the same compressor data set.

In this paper, the structure is as follows: section 1 briefly introduces the basic methods including LSTM (long-short term memory), GRU (gated recurrent unit) and ARIMA (autoregressive integrated moving average model). Thereafter, sections 2 and 3 conduct comparative study of the three time series prediction models on a real centrifugal compressor dataset to comprehensively investigate their characteristics and performance. Finally, section 4 gives the conclusion.

2. Basic methods

2.1. Time-series prediction models

2.1.1. LSTM neural network prediction model. Before we introduce the model, let's determine the form of the input data. There are many input forms of LSTM neural network. Here we introduce the single variable time series data. Let's assume that there is a time axis with n time points, which are evenly distributed on the time axis in a discrete form, \( X_t \ (t = 1, 2, 3 \ldots n) \) is a multi-dimensional time variable at each time point. These random variables constitute the input of LSTM neural network [13][13].
The LSTM neural network contains a lot of parameters [14]. Specifically, $C_t$ is called cell state, it contains part of the information summarized from the historical data of neural network, and runs through the whole unit. $h_t$ is the output of each cell. $X_t$ is the input for each cell. There are four gate-like structures in LSTM. Their function is to filter through the information [15].

The first is the forgetting gate formulated as:

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

where $\sigma(\cdot)$ is the sigmoid activation function used for the nonlinear transformation of input, $W_f$ is the weight of the neural network, and $b_f$ is the bias.

Secondly, the update gate, also known as the input gate, is used to update the state of the cell. The specific formula is as follows:

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

$$ \tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), $$

$$ C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t, $$

where $C_t$ represents the state of the unit, which is used to store the historical information, and is the degree of updating information.

Finally, the output gate responsible for the output results and is formulated as follows:

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

$$ h_t = o_t \ast \tanh(C_t) $$

2.1.2. GRU neural network prediction model. GRU is a special form of LSTM. It also has a gate-like structure in LSTM. However, the difference is that the GRU neural network simplifies four gates into two gates, namely reset gating and update gating.

The first is to reset the gating $r_t$. Its main function is similar to the hidden state in LSTM. It integrates the input information into $h_{t-1}$, and controls how much information is written to the current candidate set $h_{t-1}$ at the previous moment. The formula is as follows:

$$ r_t = \sigma(W_r[h_{t-1}, x_t]), $$

$$ \tilde{h}_t = \tanh(W_r[r_t \ast h_{t-1}, x_t]) $$

The second is update gate $z_t$. The larger the update gate is, the more state information of the previous moment is brought in. The second is the update gate $z_t$, which is expressed as:

$$ z_t = \sigma(W_z[h_{t-1}, x_t]), $$

$$ z_t = \sigma(W_z[h_{t-1}, x_t]). $$


\[ h_t = (1-z_t) h_{t-1} + z_t \tilde{h}_t \] (11)

The larger the update gate is, the more state information of the previous moment is brought in.

2.1.3. **ARIMA time series prediction model.** The ARIMA model is one of the commonly used statistical models for time prediction. It is based on the integration of autoregressive model (AR) and moving average model (MA). Before introducing the model, we need to explain the conditions they used.

2.1.3.1. **Stability requirement.** Stationarity means that the curve fitted based on time series samples has the inertia which continues according to the existing form. It can be divided into strict stationarity and broad stationarity. Strict stationarity means that no matter how much window size, as long as the same number of consecutive data, they constitute the same joint distribution. But usually, the strict stability condition is difficult to satisfy. Hence, the strict stability condition is weakened to the wide stability condition. At this time, only the second moment is required to exist at any time. The mathematical expectation of random variables does not change with time. The autocorrelation coefficient of random variables at two different time points is only related to the time difference and does not change with time.

2.1.3.2. **Basic theory of ARIMA.** Autoregressive model is a time series model based on its own data to predict the future. It needs to meet the stationary condition, and the data must have correlation. If the autocorrelation coefficient is less than 0.5, it should not be used. The p-order autoregressive process formula is as follows:

\[ y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \epsilon_t \] (12)

where \( \mu \) is a constant term, \( p \) is the order, and \( \gamma_i \) is the autocorrelation coefficient, \( \epsilon_t \) is the error.

The moving average model focuses on the accumulation of errors in the autoregressive model. The formula is as follows:

\[ y_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \] (13)

The moving average method can effectively eliminate the random fluctuation in the prediction. The ARMA method is a combination of autoregressive model and moving average model as:

\[ y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \] (14)

The ARIMA model is based on the ARMA model by adding difference steps to ensure that the data meets the stability conditions. The ARIMA model contains three parameters \( p \), \( q \), and \( d \). The parameter \( p \) is the order of autoregression, which is also the lag number of time series data itself, called the AR term. The parameter \( q \) is the lag number (lags) of the prediction model using random error, which is called the MA term. The parameter \( d \) means that the data has experienced several orders of difference before obtaining wide stationarity, also known as the integrated term.

3. **Comparative study**

3.1. **Dataset**

The data comes from a group of centrifugal compressors with the shaft speed of 5556 RPM. The data contains 11 variables, and the specific variables are shown in the figure 2. One of the variables (inlet air temperature) selected from the raw data without any processing is shown in figure 3. From March 1, 2013 to October 16, 2013, the data was obtained hourly, including a total of 3520 data points. Note that the data points during the maintenance outage are excluded in this study. There is one outage
happened because of the failure of multiple impellers. The run data is recorded manually, so the data inevitably contains outliers, human error, outliers, missing values, and noise.

3.2. Data pre-processing
In this section, the data used in the data analysis model is processed by the Bayesian wavelet denoising and the probabilistic principal component analysis (PPCA), see Appendix. That is, we first use the Bayesian wavelet to clean the data because there are a large number of noises which contains useless information. By this way we can get the denoised data, see figure 3. Thereafter, we use PPCA to reduce the dimension of the data to make the prediction easier. Consequently, the resulting data contains six principal components that contribute the most to the original information. In the experiments below, we take the first principle component (PPCA1) for analysis.

The purpose of this example is to compare the performance of the aforementioned three time series models on the real data. We run the experiments on a personal laptop with intel 2.40 GHz CPU and 16 G RAM.

![Figure 2. Monitor variables of the centrifugal compressor [9].](image)

![Figure 3. (a) The raw data and (b) the denoised data regarding the inlet air temperature.](image)

4. Results and discussions

4.1. The predictions of LSTM
The process predicted by the LSTM model is shown in figure 4.
First of all, we read the data file and change the form of the data. The random variables corresponding to each time point ($t_1,t_2,t_3,...,t_{3520}$) are defined as $x_1,x_2,...,x_{3520}$. A window of size 2 is specified. After traversing the entire data set by this window, the format of the data set is changed to $(x_1,x_2),(x_2,x_3)\ldots(x_{3519},x_{3520})$, which ensures that the input meets the input form required by the LSTM neural network.

The parameters of the model should be set before training. In machine learning, these parameters are called hyperparameters. We set the training epoch (training rounds) as 50, the input size (input dimension) as 2, the number of LSTM hidden layer as 1, the number of hidden units (hidden layer neurons) as 50, and the learning rate as 0.01.

After setting the parameters, the dataset should be divided into the training set and the test set. We choose 70% of the data as the training set for model training and the remaining 30% as the test set to test the model performance.

After dividing dataset, in order to optimize the prediction ability of the model, the Z-score criterion is used to normalize the data. This method changes the overall distribution of data into normal distribution along each dimension, and can eliminate the gradient dispersion or gradient disappearance caused by the outliers in the error back propagation.

Next step is the final step before training model, we should make the data normalization. Because when we calculate the minimum value of the loss function, we used the method based on gradient descent. And the data normalization can make the gradient descent distance shorter, so the time of calculating will be cut.

Then we start to train the model. The prediction experiment is designed to predict the next time step with double time steps. Considering the low complexity of the task, the single-layer one-way LSTM neural network was used for prediction.

The training process is as follows the data from the LSTM network output is fed into a fully connected layer, and the output of the full connection layer should input into the MSE loss function together with the truth value. Then, the Adam [16] optimization method based on gradient descent mentioned above is used to optimize the function value, and the model training is completed after 50 rounds of repetition.

After training, we begin to test model on the test set. There are 1056 data points in the test set. The test results are shown in the figure 5.
The root-mean-square error of the model on the training set is 0.00857. In order to exclude the possibility of overfitting, the performance of the model on the test set should be evaluated. In figure 4, the green curve is the true value on the test set, and the red curve is the predicted value on the test set after inversed normalization. Additionally, figure 6 shows the residual curve between the predicted value and the true value. It can be observed that the maximum absolute residual error is 0.687. It can be found that, although the fluctuation range of the true data is large, the trained LSTM network well captures the trend of data fluctuation on the train set, and generalizes well on the data never seen by the model, which may be related to the ability of the LSTM model itself to remember the features of historical data. However, based on experience, the model cannot reach such a degree of accuracy in practical engineering applications. After analysis, the most likely reason is that the task of single-step prediction is too simple for the model. The LSTM model can easily predict the trend of the next time step by summarizing the data characteristics of the first two time steps by the ability to remember
historical data characteristics. It can be seen from the figure 7 that the predicted value lags behind the real value to a certain extent. The hysteresis of the predicted value is also related to the characteristics of the model itself. The reason for "hysteresis" is that LSTM pays too much attention to the data at the last \( t-1 \) moment in the time step, resulting in the prediction result being the repetition of the data at the \( t-1 \) moment. However, from the overall prediction results, the LSTM model performs very well in this group of data.

4.2. The predictions of GRU
Since the GRU is a variant of the LSTM, it is the same as the previous section in data input. It can be seen from the figure 8 and figure 9 that the prediction of GRU is slightly better than that of LSTM, and the maximum error on the test set is 0.39. Similarly, after 50 rounds of data training, the GRU converges slightly faster than the LSTM, which can be explained by the structure of the model. The GRU has only two doors (update and reset gates) and the LSTM has three doors (forget, input, and output gates). The GRU directly passes hidden state to the next unit, while the LSTM uses the memory cell to wrap the hidden State. Hence, the GRU is easier to converge because of fewer parameters. However, in the case of large data sets, the LSTM representation performance is better.

![Figure 8. Performance of GRU on test set.](image)

![Figure 9. Residual error between the predicted value and the true value.](image)

4.3. The predictions of ARIMA
According to the requirements, the Autocorrelation coefficient (ACF) stability test was first conducted on the original data.

From the ACF stability test, we can find that \( P > 0.05 \), and the \( t \) statistic is outside the 5% confidence interval. Hence, it can be determined that the time series is not a stationary time series, and further difference processing is needed for the data.

After the first-order difference, the data meets the demand of the Dickey fuller detection, but the mean and standard deviation still fluctuate.
We thereafter use the Bayesian information criterion to find $p$ and $q$ for the ARIMA model. This method is based on the principle of making the model as simple as possible to select parameters. The smaller the BIC value is, the simpler the model is. As shown in figure 10, we arrive at $p = 20$, $q = 4$.

![Bayesian Information Criterion diagram](image)

**Figure 10.** Bayesian Information Criterion diagram.

However, the ARIMA (20,1,4) model does not perform well in the prediction set as shown in figure 11, since it only catching part of the trend of the data. It is completely distorted after 200 steps of prediction. The reason for the poor predictive performance of ARIMA is that the ARIMA model is limited. The reason why it requires the data to be stationary is that the model relies on the continuation of historical data when making predictions. If the existing underlying system is too complex and the data lacks "inertia", it is impossible to find an ARIMA model with the right parameters to predict. Nonstationary sequences such as random walk sequences are not suitable for prediction with this model. However, the ARIMA also has its advantages in data prediction. If the basic model is simple enough, it is much more efficient than other deep learning methods and saves a lot of computing costs [17].

![Performance of ARIMA on test set](image)

**Figure 11.** Performance of ARIMA on test set.

5. Conclusion

This paper compares the performance of three time-series prediction models. In this group of compressor operation data, two time series models based on recurrent neural network (LSTM and GRU) perform very well by accurately predicting the trend of the data. Considering that the GRU is not only slightly better than LSTM in terms of test results, but also simpler in structure than LSTM, its capability in time series prediction is worth further exploration.

The ARIMA model based on statistical method does not perform well in this dataset. But the problem can be attributed to the following reasons:
• The data itself is non-stationary, the autocorrelation of data is not strong, and the difference of data cannot meet the requirements of the model.
• Unlike the SARIMA model [19], the ARIMA model does not consider some factors, for example, the seasonality, which could further improve the prediction.

Appendix A

Bayesian wavelet denoising
Discrete wavelet packet decomposition, also known as the optimal sub-band tree structure, is the optimization of wavelet transform [19].

In wavelet transform, if the scale parameter is $2^i$, that is, the scale is discretized by power series. Then, this kind of wavelet transform is called dyadic wavelet transform.

In dyadic wavelet transform, the scale function and wavelet function of adjacent series also have a recursive relationship when the wavelet packet is decomposed at all levels.

Note that the father wavelet $\varphi(t)$ in the wavelet packet transform is $\mu_0^0(t)$, the mother wavelet $\psi(t)$ is $\mu_1^0(t)$, where the superscript represents the decomposition order of the wavelet packet, and the subscript represents the position of the wavelet packet in its level. The recurrence relation is expressed as follows:

$$
\begin{align*}
\mu_{2n}(t) &= \sum_k s_k \mu_n(2t-k) \\
\mu_{2n+1}(t) &= \sum_k w_k \mu_n(2t-k)
\end{align*}
$$

(A.1)

where $s_k$ represents the scale coefficient and $w_k$ represents the wavelet coefficient.

Because the noise should account for a small part of the whole data, we carry out Bayesian hypothesis test on each group of coefficients on the decomposition time series and set the coefficients outside the confidence interval to zero. Then we reorganize the denoised parts to obtain the following formula:

$$
\phi(t_i) = \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \left[ S_j(k) \varphi_{j,k}(t_i) \right] + w_j(k) \omega_{j,k}(t_i),
$$

(A.2)

where $f(t_i)$ is the time series after a series of decomposition and reorganization. The double summation symbol decomposes the scale and wavelet subspace to the second level subspace at the same time, so as to decompose the signal in time domain and frequency domain.

Probabilistic principal components analysis
Probabilistic PCA model [20] is a special form of linear factor model. The data generation process of the linear factor model is as follows:

$$
h \sim p(h),
$$

(A.3)

where $p(h)$ is the distribution of $h$ satisfying $p(h) = \prod p(h_i)$, which is easy to take sample from the distribution. Next, under the given $h$ distribution, the observable variables of true value are sampled as:

$$
x = Wh + b + \text{noise}
$$

(A.4)

In factor analysis, the priori distribution of $h$ is a Gaussian distribution.

$$
h \sim N(h;0,I)
$$

(A.5)

It is assumed that the noise is extracted from the Gaussian distribution of the diagonal covariance matrix. The covariance matrix is $\psi = \text{diag}(\sigma^2)$, where $\sigma^2 = [\sigma_1^2, \sigma_2^2, \sigma_3^2, \ldots, \sigma_n^2]^T$ denotes a vector, and each variable represents the variance of a variable.

Consequently, $x$ obeys multidimensional normal distribution and satisfies the following conditions:

$$
x \sim N(x; b, WH^T + \psi)
$$

(A.6)
For the PPCA method, we need to modify the factor analysis model to make the conditional variance \( \sigma_i^2 \) being equal to the same value. Then the conditional distribution of \( x \) becomes:

\[
x \sim N(x; b, WW^T + \sigma^2 I)
\]

Or, it is equivalent to:

\[
x = Wh + b + az,
\]

where \( z \sim N(z; 0, I) \) is Gaussian noise. The probabilistic principal component analysis model takes advantage of the observation that most of the changes in the data can be described by the latent variable \( h \), except for some small residual reconstruction errors. When \( \sigma \to 0 \), the probability PCA degenerates to PCA.

References

[1] He Y, An Y and W H 2007 Chinese Journal of Turbomachinery 6 68-72
[2] Ahmad R and Kamaruddin S 2012 Comput. Ind. Eng. 63 135-49
[3] Tahan M, Tsoutsanis E and Muhammad M 2017 Appl. Energy 198 122-44
[4] Bernalte Sanchez P J and Garcia Marquez F P 2020 Fourteenth Int. Conf. on Management Science and Engineering Management (Chisinau) vol 2 ed Xu J et al (Cham: Springer International Publishing) pp 791-800
[5] Yang S, Jiang X, Xu S and Wang X 2019 IEEE Trans. Ind. Electron. 67 10887-96
[6] Wen C, Lv F, Bao Z and Liu M 2020 J. Electron. Inf. Technol. 42 234-48
[7] Si J, Ma J, Niu J and Wang E 2017 J. Vib. Shock 36 164-71
[8] Xu Z, Liu K, Zhang H, Wang D, Zhang M and Wu X 2014 J. Vib. Shock 23 133-9
[9] Xu S, Jiang X, Huang J, Yang S, Wang X 2016 Mech. Syst. Signal Process. 80 1-18
[10] Bottieau J, Vallee F, De Greve Z and Toubeau J F 2018 IEEE Int. Energy Conf. (Limassol) (Piscataway: IEEE) p 6
[11] Yin A, Yan Y, Zhang Z, Li C and Sanchez R 2020 Sensors 20 1-13
[12] Zhao H, Jiang X, Xu S, Lin L and Gong Y 2020 Chinese Journal of Turbomachinery 62 69-76
[13] Zhang C 2020 Chinese Journal of Turbomachinery 62 77-82
[14] Hochreiter S and Schmidhuber J 1997 Neural Comput. 9 1735-80
[15] Liu X, Zhou Q, Zhao J, Shen H and Xiong X 2019 Sensors 19 1-20
[16] Li C, Wu X, Zhu P, Song L, Li J and Feng Z 2018 Chinese Journal of Turbomachinery 60 1-11
[17] Contreras J, Espinola R, Nogales F J and Conejo A J 2003 IEEE Trans. Power Syst. 18 1014-20
[18] Dimri T, Ahmad S and Sharif M 2020 J. Earth Syst. Ence 129 1-16
[19] Dong L, Wei G, Hui H, Li H and Wang J 2011 Chinese Journal of Turbomachinery 5 37-41
[20] Zhou Q, Liu C, Zhao J, Shen H, Xiong X 2018 J. Vib. Shock 37 31-7

Acknowledgments

This work was supported by the National Key Research and Development Project (2020YFA0714403), the MIIT Marine Welfare Project (Z135060009002), the research and Innovation in Science and Technology Major Project of Liaoning Province (2019JH1-10100024), the National Natural Science Foundation of China (52005074), and the Fundamental Research Funds for the Central Universities (DUT19RC(3)070) at Dalian University of Technology.