# A dilated convolution network-based LSTM model for multi-step prediction of chaotic time series

# A dilated convolution network-based LSTM model for multi-step prediction of chaotic time series

# A Dilated Convolution Network Based LSTM Model for Multi-step Prediction of Chaotic Time-series

**Rongxi Wang [1,*], Caiyuan Peng [1], Jianmin Gao [1], Zhiyong Gao [1] and Hongquan Jiang [1, 2]**

[1] State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China; rongxiwang@163.com (R.W.); pcyggg@stu.xjtu.edu.cn (C.P.); gjm@xjtu.edu.cn (J.G.); zhygao@xjtu.edu.cn (Z.G.);

[2] Laboratory for Manufacturing and Productivity, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; xjtujhq@mail.xjtu.edu.cn

* Correspondence: rongxiwang@163.com; http://orcid.org/0000-0001-5958-0642



**Abstract:** Aiming to solve the problems of low accuracy of multi-step prediction and difficult determining of the maximum number of prediction steps of chaotic time-series, a multi-step time-series prediction model based on the dilated convolution network and Long Short-Term Memory (LSTM), named the Dilated Convolution- Long Short-Term Memory (DC-LSTM), is proposed. The dilated convolution operation is used to extract the correlation between the predicted variable and correlational variables. The features extracted by dilated convolution operation and historical data of predicted variable are input into LSTM to obtain desired multi-step prediction result. Furthermore, cross-correlation analysis (CCA) are applied to calculate the reasonable maximum prediction steps of chaotic time series. Actual applications of multi-step prediction were studied to demonstrate the effectiveness of the proposed model which has superiorities in RMSE, MAE and prediction accuracy because of the extracting of correlation between the predicted variable and correlational variables. Moreover, the proposed DC-LSTM model provides a new method for prediction of chaotic time series and lays a foundation for scientific data analysis of chaotic time series monitoring systems.

**Keywords:** Chaotic time-series; Multi-step prediction; Dilated convolution network; Long Short-Term Memory

## 1. Introduction

The monitoring time series in the complex electromechanical systems are typical chaotic time series. As a typical complex electromechanical systems, process industry production system is composed of many subsystems, which are coupled by pressure, temperature, flow, vibration, rotation speed and other media networks (Wang et al. 2018).The distributed control system (DCS) database contains a large amount of monitoring data which includes abundant system status information. Scientific guidance for fault tracing and maintaining of process industry production system is important to ensure its operational safety and reliability. To guide the fault tracing and maintaining of process industry production system scientifically, it is necessary to recognize the changing trend of system performance in real time. Therefore, an accurate multi-step time series prediction research is of great significance.

As an important research area in the data analysis field, time series prediction plays an important role in process industry (Wang et al. 2017b), financial analysis (Cao et al. 2019), clinical medicine (Alexander et al. 2018), social phenomena analysis (Wang et al. 2017a) and other fields (Ai et al. 2019), because of analyzing the historical data of a dynamic system and predicting its future operation pattern. The research of time series forecasting methods has begun with the regression analysis in mathematical statistics methods, such as polynomial interpolation method, least square method, and others. With the introduction of the Auto-Regressive Integrated Moving Average (ARIMA) model, the application of regression analysis method in time series forecasting has become mature; however, regression analysis method can be applied only to linear time series (Pannakkong et al. 2018). Due to the non-linear characteristics of time series in practice, many scholars have carried out the research on non-linear time series prediction using the Back Propagation (BP) algorithm (Pal and Kar 2017), Support Vector Machine (SVM) (Xiao et al. 2019), Radial Basis Function (RBF) (Awad and Qasrawi 2018), Echo State Networks (ESN) (Lopez et al. 2018; Liang et al. 2018), and other models (Yeh et al. 2019). The traditional artificial neural networks cannot capture a long-term dependence of time series data. Because the current output in Recurrent Neural Networks (RNN) depends on the previous computations, RNN are regarded as recurrent and frequently used in time series forecasting tasks (Chen et al. 2018). However, there is a problem of gradient disappearance in the RNN training process. To avoid this problem, an improved RNN model, the Long Short Term Memory Network (LSTM), has been proposed (Hochreiter and Schmidhuber 1997). The LSTM model addresses the problems of gradient disappearance, gradient explosion, and insufficient long-term memory ability of RNNs, and can effectively use the time series information of long distance to predict time series. The traditional time series prediction models are single variable forecasting models. The Convolutional Recurrent Neural Network (CRNN) (Cirstea et al. 2018) takes the results of convolution and pooling operations in each dimension of multi-dimensional data as an input of an RNN to realize multi-variable forecasting, but does not consider the correlation between variables. At present, the multi-step high-accuracy time series prediction methods such as the fusion convolutional long short-term memory network (FCL-Net) model (Ke et al. 2017), multi-output SVM model (Zhou et al. 2019) and a new hybrid vector error correction and nonlinear autoregressive neural network (VEC-NAR) model (Cheng et al. 2019), are all iterative prediction methods. With the increase in the number of prediction steps, the prediction error of the iterative prediction method increases rapidly, and the multi-step prediction accuracy decreases continuously.

Convolution neural network was applied to document recognition when it was proposed firstly (Lecun et al. 1998). The deep learning method, gated CNN was applied to multi-step day-ahead time series prediction respectively, and the results were better than those of ARIMAX, a traditional time series prediction method (Cai et al. 2019). A mathematical theory of convolutional neural networks for feature extraction was proposed and proved (Wiatowski and Bölcskei 2017). As a variant of convolution neural network, dilated convolution network can implement the effect of convolution and pooling operations in convolution neural network by introducing a new parameter called the dilated rate into convolution operation (Li et al. 2018). Dilated convolution has been widely applied to the image segmentation tasks (Wang et al. 2019) and context features extraction (Sun et al. 2019). Many researchers (Strubell et al. 2017) (Chang et al. 2018) have proved that dilated convolutions can play a good role in extracting the sequence dependency, thus, dilated convolutions is applied to extract the features of multi-step time series prediction and capture the time dependence between variables in multivariate chaotic time series. It has been proved that dilated convolutions can be applied successfully to forecast financial time series of limited length (Borovykh et al. 2019). At the same time, as a typical RNN, LSTM which is often used in time series prediction can effectively utilize the long-distance temporal information of single variable (Liu et al. 2018). In addition, an improved LSTM which is called MSM-LSTM has shown good performance on the forecasting of typical chaotic time series obtained from the Lorenz system and the Kuramoto-Sivashinsky equation (Vlachas et al. 2018). In conclusion, the proposed DC-LSTM model, which is an improved LSTM based on a dilated convolution network, extracts the correlation between variables in multivariate chaotic time series using dilated convolution networks and making full use of the useful information of historical data. At the same time, it captures the long-distance temporal information of chaotic time series data using the LSTM model, so as to effectively utilize a long-distance time series information for chaotic time series prediction.

The rest of the paper is organized as follows. In Section 2, the dilated convolution network, LSTM and multi-step prediction of chaotic time series are introduced. An improved LSTM model based on dilated convolution network is presented in Section3. The detailed process of the proposed framework for multi-step prediction is described in Section 4. In Section 5, the experimental setup includes the benchmark, dataset and performance metrics are introduced. The prediction results and some extend discussions are analyzed in Section 6. The conclusions are drawn in Section 7.

## 2. Preliminaries

In order to extract temporal correlation and realize feature extraction between variables in multi-dimensional chaotic time series, a dilated convolution network is introduced. Each convolution output of the dilated convolution network contains large range input information, and the dilated convolution network realizes the convolution and pooling operations of an ordinary convolution network without losing the input information. At the same time, to extract a long-term dependence between variables in multi-dimensional chaotic time series and effectively use the long-distance historical data for the multi-step chaotic time series prediction, an LSTM network model is used.

### *2.1 Dilated Convolution Network*

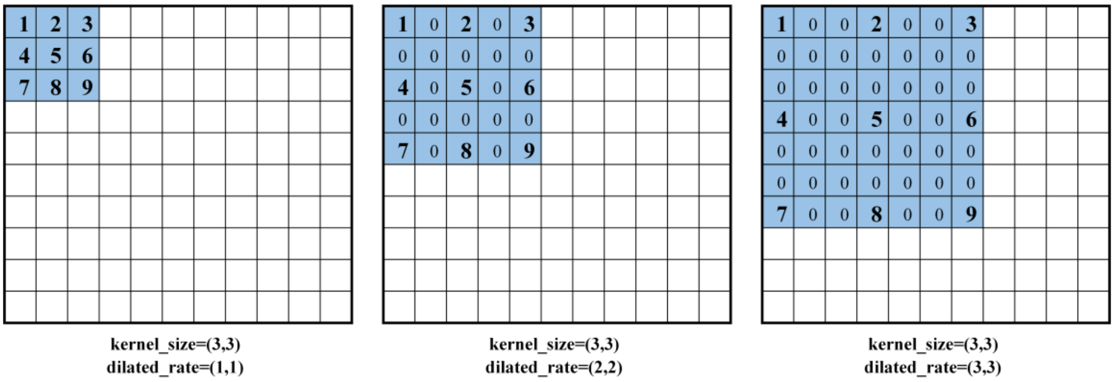
Convolution networks denote a neural network type that use convolution operation (Lecun et al. 1998). In convolution networks, the convolution kernel determines the weight matrix between the input and output, and the receptive fields determines the corresponding relationship between the output and input. The larger the receptive field is, the larger the amount of information obtained from the input is. The convolution kernel moves through the input matrix according to the certain rules and convolutes the input matrix in the receptive field to get the output matrix of the convolution network.

The dilated convolution is also known as an atrous convolution, and it introduces a new parameter called the dilated rate to the ordinary convolution. This parameter defines the stride of dilated convolution when convolution kernels process the data (Li et al. 2018). A 2-D dilated convolution can be defined by(Li et al. 2018):

$$y(m,n)=\sum_{i=1}^{M}\sum_{j=1}^{N}x(m+r\times i,n+r\times j)w(i,j), \tag{1}$$

where $x(m,n)$ denotes the dilated convolution input, $w(i,j)$ denotes the convolution kernel of dilated convolution, $(r,r)$ denotes the dilated rate of dilated convolution, and $y(m,n)$ denotes the output of dilated convolution. When the size of convolution kernel is $k\times k$, the size of the receptive field of 2-D ordinary convolution is also $k\times k$, and a receptive field of 2-D dilated convolution having the dilated rate of $(r,r)$ is expressed as $(k+(k-1)(r-1))\times(k+(k-1)(r-1))$.

As shown in Figure 1(a), when the size of the convolution kernel is $3\times 3$, and the dilated rate is $(1,1)$, and the receptive fields of dilated convolution have the size of $3\times 3$, which is the same as ordinary convolution. However, as shown in Figures 1(b) and 1(c), when the size of convolution kernel is $3\times3$, and the dilated rate is $(2,2)$ and $(3,3)$, respectively, the receptive fields of dilated convolution will be $5\times5$ and $7\times7$, and the receptive fields of 2-D ordinary convolution will have the size of $3\times3$.




(a) (b) (c)

**Figure 1.** The receptive fields of dilated convolution

Dilated convolution has been widely applied to the problems where an image needs the global information or voice text has a long sequence information dependence, such as in the image segmentation tasks (Wang et al. 2019), speech synthesis model (Tan et al. 2019), and context features extraction (Sun et al. 2019).

*2.2 LSTM*

Recently, due to the continuous development of deep learning, some deep learning models have been gradually applied to the research of time series data. In many deep learning models, RNNs introduce the concept of time sequence into the network structure design, which makes it more adaptable to the time series data analysis. In some RNNs, the LSTM model solves the problems of gradient disappearance, gradient explosion, and insufficient long-term memory ability, so that LSTM can effectively utilize the long-distance temporal information (Liu et al. 2018).

The hidden layer of the original RNN has only one state $h$, and it is very sensitive to the short-term input. In the LSTM, the state $c$ is added to save the long-term state of a sequence, thus realizing the long-term memory of the sequence. The LSTM model has three inputs: $x_t$ which is the input value of the current LSTM network, $h_{t-1}$ which is the output value of the previous LSTM network, and $c_{t-1}$ which is the unit state of the previous LSTM network. The outputs of the LSTM model are $h_t$ and $c_t$ which denote the output value and the unit state of the current LSTM network, respectively. Gates, such as forget gates, input gates, and output gates, are used to control the model state in LSTM (Hochreiter and Schmidhuber 1997). Forget gates aim to control the amount of information transmitted from the previous unit state $c_{t-1}$ to the current unit state $c_t$, input gates aim to control the amount of information transmitted from the current network input $x_t$ to the current unit state $c_t$, and the output gates aim to control the amount of information transmitted from the previous unit state $c_{t-1}$ to the current network output $h_t$.

The LSTM model has been successfully used in time series data research in different fields, including text sequence-related language modeling (Sundermeyer et al. 2015), speech recognition (Zhao et al. 2019), machine translation (Baniata et al. 2018), media sequence-related audio and video data analysis (Liu et al. 2019), picture title modeling (Hua et al. 2019), transportation related traffic flow prediction (Tian et al. 2018), aircraft fault prediction (Zhang et al. 2018), and so on.

### *2.3 Multi-step Prediction of Chaotic Time Series*

Many natural phenomena and industrial systems, such as meteorological prediction, financial analysis and industrial field, exhibit chaotic behaviors which is non-periodic and not completely randomized. For the aim to predict the future behavior and mode of chaotic systems, the data in chaotic systems was generally saved into time series data, and the research on chaotic time series forecasting is essential. Unfortunately, chaotic time series prediction is a challenge and difficult work, because the data do not have a similar pattern. Phase space reconstruction method using known chaotic time series data was proposed by Takens (1981). Chaotic time series display some stochastic behavior in time domain, at the same time, the determined behavior of chaotic time series was shown in phase space structure. Therefore, chaotic time series in phase space can be analyzed and predicted (Sivakumar 2002; Dhanya and Kumar 2010; Zhou et al. 2016). With the development of Artificial Intelligence, some machine learning models, such as ANN (Karunasinghe and Liong 2006), SVM (Pano-Azucena et al. 2018) and ESN (Liang et al. 2018), have showed good prediction performance in chaotic time series forecasting.

Time-series prediction refers to the problem of predicting the future of sequential data based on some finite history. In its simplest form, the problem is restricted to predicting a single time-step into the future. However, for real time series, such as financial data and industrial monitoring data, only accurate multi-step prediction can meet the practical application needs. According to the data requirement of realizing multi-step prediction, multi-step prediction can be divided into univariate prediction which refers to using only the historical data of predicted variable to achieve multi-step prediction and multivariable prediction which refers to using the historical data of predicted variable and correlational variables to achieve multi-step prediction of predict variable. In addition, according to the algorithm of multi-step time series prediction, multi-step prediction can be divided into iterative prediction and direct prediction. Iterative prediction refers to recursively apply a single-step prediction model by feeding its output as input to the next time-step. Direct prediction refers to directly predict the data after multi-step by using historical data to train a multi-step time series prediction model.

What we study in this paper is multi-step chaotic time series prediction, in which the multi-step prediction method is multivariable direct prediction in order to make full use of multivariable data and avoid error accumulation in iterative prediction.

## 3. Improved LSTM Model Based on Dilated Convolution Network

The data processed in this paper represent multivariate time series data which can be expressed as $X=(X^1,X^2,...,X^m)$, where $m$ denotes the dimension of time series data. The data for each dimension is expressed as $X^i=(x_1{}^i,x_2{}^i,...,x_n{}^i)$, where $i=1,2,...,m$ and $n$ is the number of sampling points of time series data. The predicted sequence is expressed as $\hat{Y}=(\hat{y}_1,\hat{y}_2,...,\hat{y}_p)$, where $p$ denotes the continuous prediction length of time series. At the same time, the target observations $Y$ of predicted sequence $\hat{Y}$ in this model is equal to one-dimension data of multivariate time series $X^j$, where $1 \le j \le m$. Assuming that the current time is denotes as $t$, the input data of this model is historical data and it can be expressed as $X = \left(X_{(t-l+1)}, X_{(t-l+2)}, \dots, X_{(t)}\right)$ with the time length of $l$, and the output data of this model is predicted data which can be expressed as $\hat{Y} = (\hat{y}_{(t+1)}, \hat{y}_{(t+2)}, \dots, \hat{y}_{(t+p)})$ with the time length of $p$; namely, the p-step continuous prediction of time series data is realized. In this model, the number of data samples of training data is $N_X = 0.9 \times (n - 2l)$, and of testing data is $N_C = 0.1 \times (n - 2l)$. Variables including the predicted variable and the correlational variable are represented by V in this paper.

The improved LSTM based on a dilated convolution network, i.e., the DC-LSTM, is a UY-Y method (Zhang et al. 2017), which uses the correlational variable $U$ and the predicted variable $V$ as an input data to predict the value of $V$ after $p$ steps, which can be expressed as follows:

$$v_{t+p} = F_{UV \bullet V}\left(v_{(t)}, v_{(t-1)}, \dots, v_{(t-l+1)}, u^1{}_{(t)}, u^1{}_{(t-1)}, \dots, u^1{}_{(t-l+1)}, \dots, u^k{}_{(t)}, u^k{}_{(t-1)}, \dots, u^k{}_{(t-l+1)}\right). \quad (2)$$

Equation (2) expresses the value of the predicted variable $V$ after $p$ steps ($v_{t+p}$) by using $l$ historical data of the predicted variable $V$ and $l$ historical data of the k-dimensional correlational variable that is given by $U^1, U^2, \dots, U^k$. Because the historical information of the correlational variable $U$ and the predicted variable $V$ is used, the predicted results can be obtained better.

In this work, to understand the model easily, the correlational variable is $V^i, i = 2, 3, \dots, m$, and the predicted variable is $V^1$.

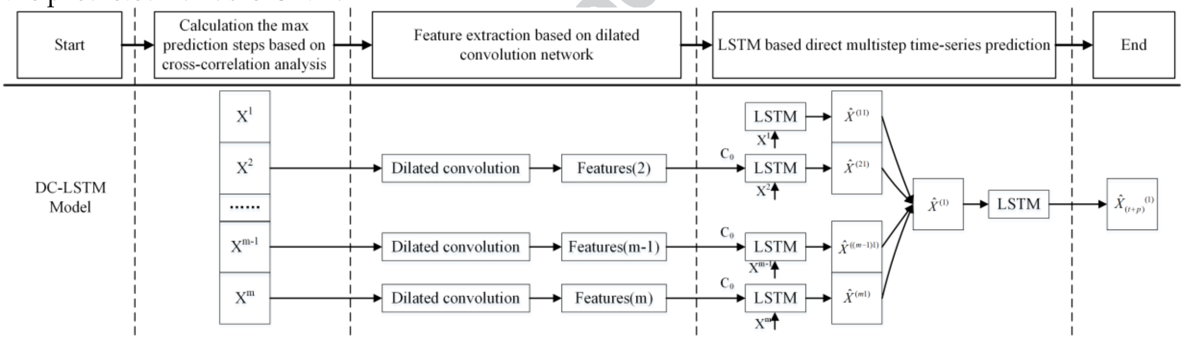



**Figure 2.** The improved LSTM model based on dilated convolution network

The DC-LSTM model is shown in Figure 2, where it can be seen that the DC-LSTM input model is $X = (X^1, X^2, \dots, X^m)$. First, the dilated convolution of multivariate historical data $X = (X^1, X^2, \dots, X^m)$ is performed at *kernel_size*=$(2,2)$ and *dilated_rate*=$(i,1), i = 1, 2, \dots, l$, so that the lead-lag matrix *features*(2) of the correlational variable $V^2$ and predicted variable $V^1$ is extracted. Then, *features*(2) is used as the initial unit state $c_0$ of the LSTM network, and the historical data $X^2$ of correlational variable $V^2$ is used as the input of the LSTM network. The LSTM network outputs a simple prediction value $\hat{X}^{(21)}$ of the predicted variable $V^1$ obtained by using the correlational variable $V^2$. Similarly, $\hat{X}^{(31)}, \dots, \hat{X}^{(m1)}$ can be obtained by using the correlational variables $V^3, \dots, V^m$. At the same time, the historical data $X^1$ of the predicted variable $V^1$ denotes the direct input into the LSTM network which is used to predict $X^{(11)}$. Finally, $\hat{X}^{(11)}, \hat{X}^{(21)}, \dots, \hat{X}^{(m1)}$ are merged into a matrix $\hat{X}^{(1)}$, which denotes the input into the LSTM network which is used to get the value of the predicted variable $V^1$ after $p$ steps ($\hat{y}^1_{(t+p)}$).

## 4. General Framework of the Multi-step Prediction of Chaotic Time-series

The general framework of the multi-step prediction of chaotic time series is shown in Figure 3, wherein it can be seen that it includes three parts: data preprocessing, DC-LSTM prediction, and multi-step continuous prediction. In addition, the pseudocode of this framework can be seen in Figure 4.

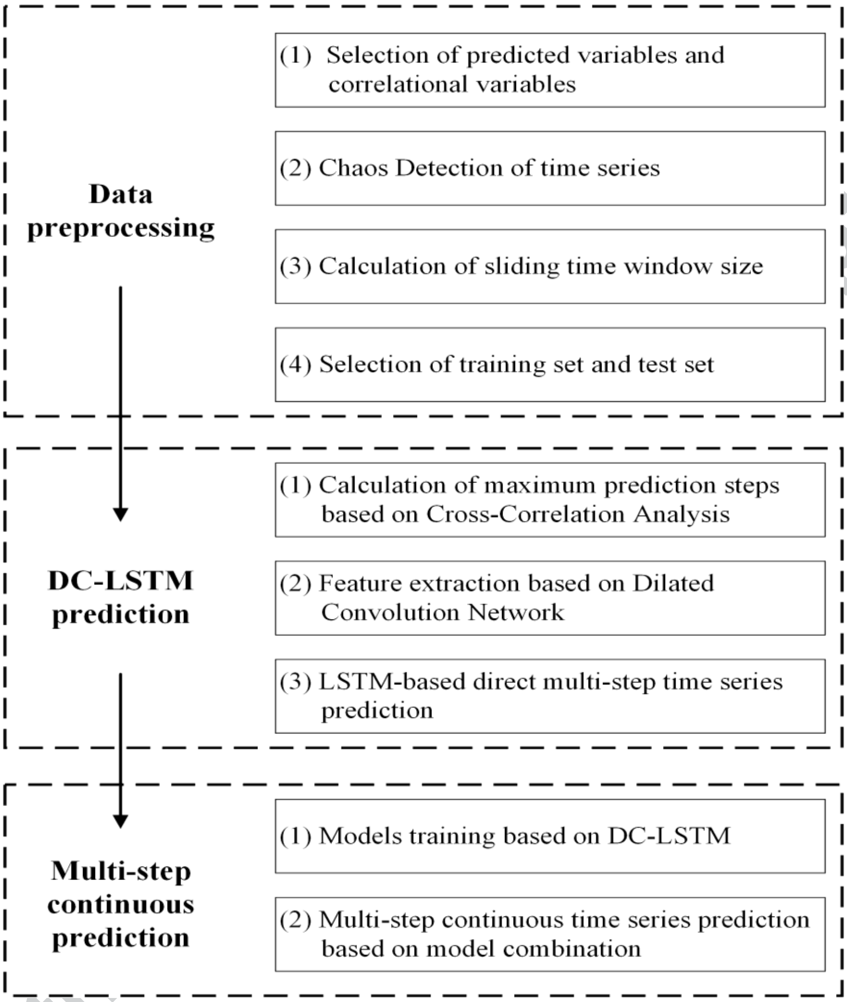



**Figure 3.** The general framework of the multi-step prediction of chaotic time series

### *4.1 Data Preprocessing*

(1) Selection of Predicted Variable and Correlational Variables

The DC-LSTM model introduced in this paper is suitable for the multi-step prediction of chaotic time series. The variables that need to be predicted by using the multi-step time series are the predicted variable $V$, and the other monitoring variables are the correlational variable $U$. Obviously, the multi-step prediction of each dimension monitoring variable can be realized by changing the predicted variable, so as to realize the state prediction of the monitoring system.

(2) Chaos Detection of Time Series

The basic characteristic of chaotic motion is that it is very sensitive to the initial conditions. Namely, the trajectories generated by two very close initial values become exponentially separated over time. This phenomenon can be quantitatively described by the Lyapunov exponent. The Lyapunov exponent $\lambda$ refers to the eigenvalue obtained by averaging the whole phase space trajectory after phase space reconstruction (Chlouverakis and Adams 2003). Starting from the sensitivity of the initial value of a chaotic system, it quantitatively describes the speed of exponential separation of phase trajectories generated by two similar initial values over time. When $\lambda > 0$, the reconstructed phase space trajectories are separated rapidly, and the system is highly sensitive to the initial value and tends to be chaotic, which indicates that the information sequence has chaotic characteristics. On the other hand, when $\lambda < 0$, the system is insensitive to the initial value. Lastly, $\lambda = 0$ represents the boundary state and needs to be studied further. Among a series of $\lambda$ obtained from the phase space trajectory, the maximum value $\lambda_{\max}$ represents the divergence speed of the phase space trajectory, and the minimum value $\lambda_{\min}$ represents the convergence speed of the phase space trajectory. Therefore, in this study, $\lambda_{\max} > 0$ is taken as the criterion of whether the information sequence has chaotic characteristics (Chlouverakis and Adams 2003).

**input**: $X = (X^1, X^2, ..., X^m)$

**output**: $(\hat{Y}_{(1)}, \hat{Y}_{(2)}, ..., \hat{Y}_{(N_C - l)})$

calculate $l$, $p_{\max}$, $N_X$, $N_C$

**for** $i \leftarrow 1$ **to** $p_{\max}$ **do**

    **for** $t \leftarrow l$ **to** $N_X - p_{\max}$ **do**

        *input* $\leftarrow \left(X_{(t-l+1)}, X_{(t-l+2)}, ..., X_{(t)}\right)$

        *target observation* $\leftarrow y_{(t+i)}$

    **end**

    train DC-LSTM $Model_i$

**end**

**for** $t \leftarrow l$ **to** $N_C$ **do**

    **for** $i \leftarrow 1$ **to** $p_{\max}$ **do**

        *input* $\leftarrow \left(X_{(t-l+1)}, X_{(t-l+2)}, ..., X_{(t)}\right)$

        input $input$ into $Model_i$ to get $\hat{y}_{(t+i)}$

    **end**

    output $\hat{Y}_{(t-l+1)} = (\hat{y}_{(t+1)}, \hat{y}_{(t+2)}, ..., \hat{y}_{(t+p_{\max})})$

**end**

output $(\hat{Y}_{(1)}, \hat{Y}_{(2)}, ..., \hat{Y}_{(N_C - l)})$

**Figure 4. The pseudocode of the multi-step prediction of chaotic time series**

To obtain $\lambda_{\max}$, the mutual information method proposed by Fraser (Fraser and Swinney 1986) is used to determine the delay time $\tau$. The basic idea of delay time selection based on the mutual information method is to calculate the mutual information function $I(\tau)$ at different values of time delay $\tau$. The value of $\tau$ denotes the delay time that corresponds to the first local minimum of the mutual information function $I(\tau)$, and a suitable phase space delay coefficient $\tau$ can be determined. Here, the Cao method (Cao 1997) which denotes an improved pseudo-nearest neighbor method(Rhodes and Morari 1997) was used to determine the embedding dimension. The Cao method can effectively distinguish the random signal and the deterministic signal. This method requires only the value of time delay to calculate the embedding dimension, and can calculate the embedding dimension $d$ with a small amount of data. Next, the Wolf algorithm (Wolf et al. 1985) is used to calculate the maximum Lyapunov exponent $\lambda_{\max}$. Under the Euclidean distance, using the initial point $x_1$ as the base point, we can find a phase point $x_{n_1}$ which is the closest point to $x_1$ as an endpoint in the phase space, and record the distance between these two points as $L_1 = \left\| x_1 - x_{n_1} \right\|$. For each point $X_i$ in the phase space, the Euclidean distance $L'_i(j) = X_{j+i} - X_{j+i}, i = 1, 2, ..., \min(m-j, m-j)$ of the corresponding nearest neighbor pair is calculated after $i$ -steps. Finally, $y(i) = \frac{1}{n\Delta t}\sum_{j=1}^{n} lnL'_i(j)$ is calculated, where $n$ refers to the number different from zero in $L'_i(j)$. The slope of the fitting line which is obtained by the least square method represents the maximum Lyapunov exponent $\lambda_{\max}$.

(3) Calculation of Sliding Time Window Size

The time lead-lag $d$ between the predicted variable $V$ and the correlational variable $U$ is an important parameter for calculating the sliding time window size of time series (Zhang et al. 2017). In theory, as long as the correlational variable $U$ is ahead of the predicted variable $V$, the predicted variable $V$ can be predicted accurately. The cross-correlation analysis(CCA) (Perkel et al. 1967) is an effective method for correlation analysis between variables. At the same time, the CCA can extract the temporal lead-lag relationship between variables, including the significant lead and significant lag. Therefore, we calculate the lead-lag relationship between the predicted variable $V$ and the correlational variable $U$ by using the CCA. When the correlation variable $X^k, 2 \le k \le m$ which is ahead of the predicted variable $V^1$ is found, the leading order $d_k, 2 \le k \le m$ is calculated, and $l = \max(|d_k|)$ is taken as the sliding time window size. Then the time series are slipped sequentially to obtain the input and output of the DC-LSTM model.

(4) Selection of training set and test set

When $l$ m-dimension historical data samples $\hat{X} = (X_k, X_{k+1}, ..., X_{k+l-1})$ are used to predict the value of the predicted variable after $p$ steps, $x^1_{k+l-1+p}$, first, the multidimensional time series data $X = (X^1, X^2, ..., X^m)$ are slipped along the time sequence. Then, the first $l$ historical data samples $X = (X_k, X_{k+1}, ..., X_{k+l-1})$ are selected as the input data of the model, while $\hat{y}^1_{k+l-1+p}$ denotes the output data of the model, so as the input dataset and output dataset of the model are obtained. At the same time, the first 90% data of the input database and output database are used as model training data, while the last 10% data of input database and output database are used as model test data. In other words, the number of samples of training data is $N_X = 0.9 \times (N - l)$ and the test data is $N_C = 0.1 \times (N - l)$.

*4.2 DC-LSTM prediction*

(1) Calculation of maximum number of prediction steps using cross-correlation analysis

As already mentioned, the time lead-lag $d$ between the predicted variable $V$ and the correlational variable $U$ is an important parameter for determination of the maximum number of prediction steps of chaotic time series. In theory, as long as at least one selected correlational variable $U_k$ is ahead of the predicted variable $V$, the maximum leading order of multiple correlation variable $U_k$ is the maximum prediction step. In this paper, we calculate the lead-lag relationship between the predicted variable $V^1$ and the correlational variable $V^i, i = 2,3,\ldots,m$ by using the CCA. When the correlation variable $V^k, 2 \le k \le m$ which is ahead of the predicted variable $V^1$ is found, the leading order $d_k, 2 \le k \le m$ is calculated, and $p_{\max} = \max\left(\left|d_k\right|\right)$ is taken as the maximum number of prediction steps.

(2) Feature Extraction Based on Dilated Convolution Network

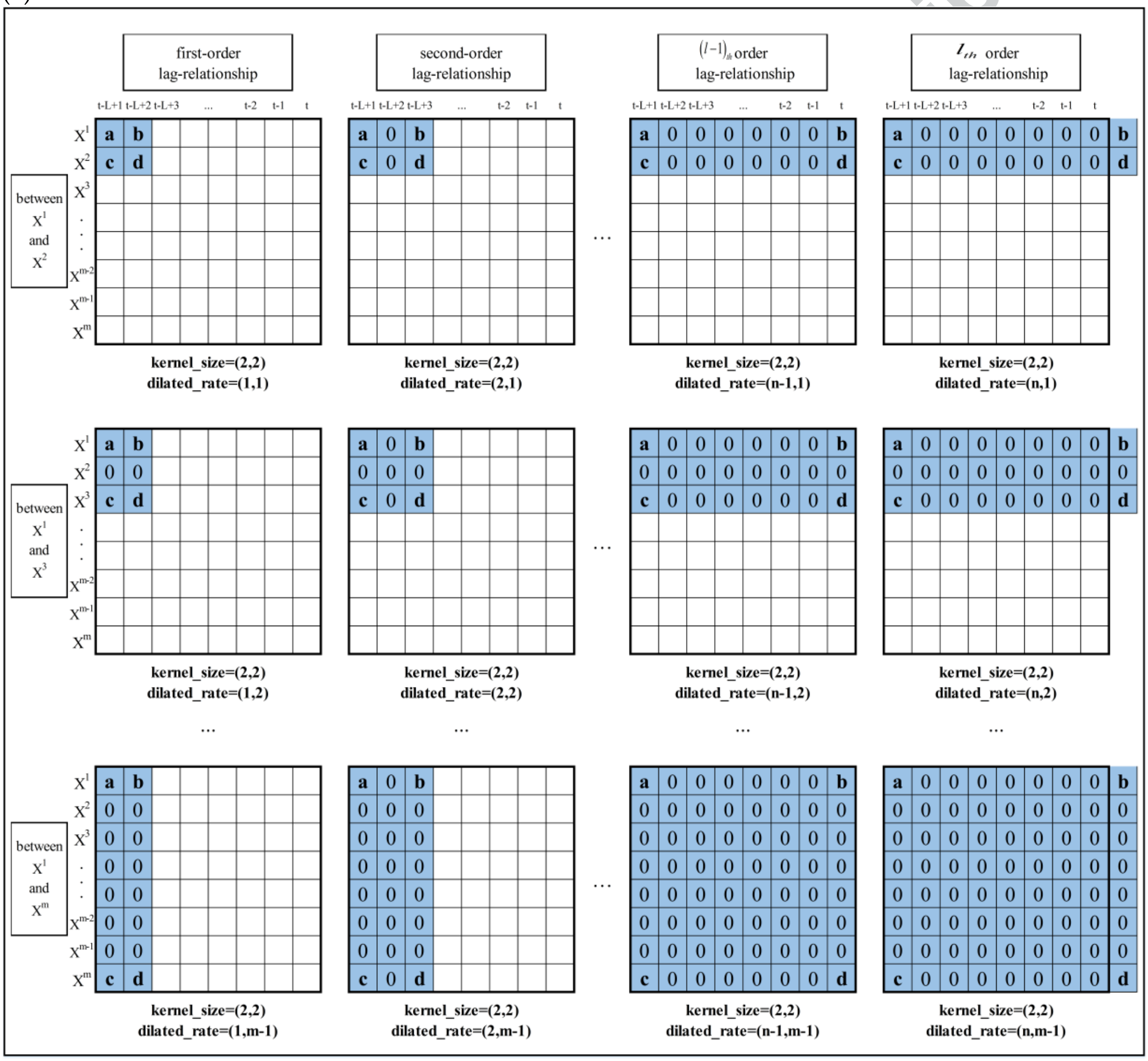



**Figure 5.** Feature extraction based on dilated convolution network

As shown in Figure 5, the convolution matrix with the size of $m \times l$ is obtained by the dilated convolution operation using the convolution kernel with the size of $2 \times 2$ and the dilated rate of $(1,1)$, along with inputting the set $X = (X^1, X^2, \ldots, X^m)$ with the size of $m \times l$ into the model after MinMaxScaler. The first row of the convolution matrix denotes the first-order lag-relationship vector $r_1^{(12)}$ between the predicted variable $V^1$ and correlational variable $V^2$. Following the same principle, the convolution matrix with the size of $m\times l$ is obtained by the dilated convolution operation using the convolution kernel size of $2\times 2$ and the dilated rate of $(l,1)$, along with inputting the set $X=(X^1,X^2,...,X^m)$ with the size of $m\times l$ into the model. The first row of the convolution matrix denotes the $l^{th}$-order lag-relationship vector $r_l^{(12)}$ of the predicted variable $V^1$ and the correlational variable $V^2$. All the vectors, from the first-order lag-relationship vector $r_1^{(12)}$ to the $l^{th}$-order lag-relationship vector $r_l^{(12)}$ of the predicted variable $V^1$ and the correlational variable $V^2$ are merged into a matrix with size of $l\times l$. The feature relationship matrix $feature(2)$ of the predicted variable $V^1$ and the correlational variable $V^2$ is obtained by the pooling operation.

Similarly, the feature relationship matrix $feature(3)$ of the predicted variable $V^1$ and the correlational variable $V^3$ is obtained, and the same operation is repeated until the feature relationship matrix $feature(m)$ of the predicted variable $V^1$ and the correlational variable $V^m$ is obtained.

(3) LSTM-Based Direct Multi-step Time Series Prediction

Firstly, the preliminary prediction value $\hat{X}^{(i1)}, i=2,3,...,m$ is obtained by the LSTM prediction model using the feature relationship matrix $feature(i), i=2,3,...,m$ as the initial unit state value $c_0$ of the LSTM, and using the correlational variable $X^i, i=2,3,...,m$ as the LSTM input. Simultaneously, a simple prediction value $\hat{X}^{(11)}$ is obtained by the LSTM prediction model using the correlational variable $V^1$ as the LSTM input. Then, all the preliminary predictive values $\hat{X}^{(i1)}, i=1,2,...,m$ are merged into a matrix which denotes the LSTM, thus, the $k^{th}$ step predictive value $\hat{X}^1_{(t+k)}$ of the predicted variable $V^1$ is obtained by the LSTM model, where *k=1,2,…,p*. After the inverse transform of MinMaxScaler, the original scale of the predictive value $\hat{X}^1_{(t+k)}$ is obtained.

*4.3 Multi-step Continuous Prediction*

(1) DC-LSTM model training

In the case of optimal parameters, p different DC-LSTM models $Model_1, Model_2, ..., Model_p$ are trained, and then, $\hat{Y}^1_{(t+1)}, \hat{Y}^1_{(t+2)}, ..., \hat{Y}^1_{(t+p)}$ are predicted by the corresponding DC-LSTM models.

(2) Multi-step continuous time series prediction based on model combination

Previously predicted $\hat{Y}^1_{(t+1)}, \hat{Y}^1_{(t+2)}, ..., \hat{Y}^1_{(t+p)}$ by different DC-LSTM models are merged into a continuous vector $\hat{Y}^1(p)=(\hat{Y}^1_{(t+1)}, \hat{Y}^1_{(t+2)}, ..., \hat{Y}^1_{(t+p)})$, which denotes the *p*-step continuous prediction result of chaotic time series $X^1$.

**5. Experimental Setup**

*5.1 Time series data*

We conduct our experiments over two different datasets, the conditional monitoring dataset obtained from the UCI data set and the compressor group dataset obtained from the practical compressor group of an energy and chemical enterprise. Details of the two datasets are described below.

(1) Conditional Monitoring dataset

Conditional monitoring dataset represents an open time series data set obtained from the UCI dataset(https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems)(Helwig et al. 2015). It is a multi-sensor based monitoring data-set for the hydraulic test-bed. The sampling frequency of different sensors are 1 Hz, 10 Hz or 100 Hz for 2205 minutes. In this work, the variables are unified to 0.1 Hz by downsampling to unify the time correspondence of different sensors. After unification, there are 13,230 data samples, including 17-dimensional indicators such as pressure, flow, and temperature.

(2) Compressor group dataset

Compressor group dataset is obtained from the practical compressor group of plant (https://pan.baidu.com/s/1xdiyrMzId3ULA1lB33cu3g (fetch code: c7cq)). The time series of the compressor group monitoring included the multi-dimensional monitoring data such as pressure, temperature, flow rate, rotating speed, vibration, and so on (Wang et al. 2017c). The 16-dimensional monitoring time series of the compressor group dataset are sampled for 120 hours, so a total of 7200 data samples are obtained at the sampling frequency of 1/60Hz.

(3) Parameter Calculations

To verify the proposed model, variable $V_1$ is chosen as a predicted variable, while the variables $V_i, i = 2,3,\ldots,m$ is chosen as the correlational variables.

First, the time delay $\tau$ of phase space reconstruction is calculated by using the mutual information method (Fraser and Swinney 1986), and the embedding dimension $d$ of the phase space reconstruction is calculated by using the Cao method (Cao 1997); then, the maximum Lyapunov exponent $\lambda_{\max}$ is calculated by using the Wolf algorithm (Wolf et al. 1985). The calculated $\tau$, $d$ and $\lambda_{\max}$ of the predicted variable $V_1$ are as follows:

**Table 1** The calculation results of chaotic detection parameters

| Dataset | Variable name | $\tau$ | $d$ | $\lambda_{\max}$ |
|---|---|---|---|---|
| Conditional monitoring dataset | $V_1$ | 2 | 10 | 0.015697334 |
| Compressor group dataset | $V_1$ | 10 | 10 | 0.001008544 |

As can be seen from Table 1, $\lambda_{\max} > 0$, the variable $V_1$ of the conditional monitoring dataset and the variable $V_1$ of the compressor group dataset are both chaotic time series.

Then, through the cross-correlation analysis, the time lag relationships between each correlated variable $V_i, i = 2,3,\ldots,m$ and predicted variable $V_1$ are obtained and showed in Table 2 and Table 3. Notably, the negative lag relationship indicates that the correlated variable $V_i, i = 2,3,\ldots,m$ is ahead of the predicted variable $V_1$, and the insignificant indicates that the lag relationship between the correlated variables $V_i, i = 2,3,\ldots,m$ and predicted variable $V_1$ is not significant, while the positive lag relationship indicates that the predicted variable $V_1$ is ahead of the correlated variable $V_i, i = 2,3,\ldots,m$.

**Table 2** The time lag relationship between $V_i$, i=2,…,17 and $V_1$ in the conditional monitoring dataset

| Correlated Variables | Time Lag | Correlated Variables | Time Lag |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Variable 2 | 0 | Variable 10 | -4 |
| Variable 3 | -2 | Variable 11 | -4 |
| Variable 4 | -2 | Variable 12 | 0 |
| Variable 5 | Insignificant | Variable 13 | -5 |
| Variable 6 | Insignificant | Variable 14 | 0 |
| Variable 7 | 0 | Variable 15 | -1 |
| Variable 8 | -2 | Variable 16 | -1 |
| Variable 9 | Insignificant | Variable 17 | -2 |

**Table 3** The time lag relationship between $V_i$, i=2,…,16 and $V_1$ in the compressor group dataset

| Correlated Variables | Time Lag | Correlated Variables | Time Lag |
|---|---|---|---|
| Variable 2 | -4 | Variable 10 | 36 |
| Variable 3 | 13 | Variable 11 | -11 |
| Variable 4 | -5 | Variable 12 | -5 |
| Variable 5 | -12 | Variable 13 | -18 |
| Variable 6 | Insignificant | Variable 14 | -1 |
| Variable 7 | -13 | Variable 15 | Insignificant |
| Variable 8 | -10 | Variable 16 | Insignificant |
| Variable 9 | 94 | | |



Last, as shown in Tables 2 and 3, there are three kinds of time lag relationships, positive, negative and insignificant. Since only negative time lag relationships play a theoretical role in our multi-step prediction, we take the value with the largest absolute value in negative as the input length of the model to maximize the utilization of useful information in historical data for multi-step prediction. So that the sliding time window size of the conditional monitoring dataset is calculated to be $l = \max\left(\left|d_k\right|\right) = \max(|-2|,|-2|,|-2|,|-4|,|-4|,|-4|,|-5|,|-1|,|-1|,|-2|) = 5$. By calculation, the input length $l = 5$ and the maximal predicted length $p_{\max} = 5$ of the conditional monitoring dataset. The number of samples in training data is $N_X = 0.9 \times \left(N - 2l\right) = 0.9 \times (13230 - 2 \times 5) = 11898$, and the test data is $N_C = 0.1 \times \left(N - 2l\right) = 0.1 \times (13230 - 2 \times 5) = 1322$ of the conditional monitoring dataset. At the same time, as shown in Table 3, about the compressor group time series, the size of the sliding time window $l = \max\left(\left|d_k\right|\right) = \max(|-4|,|-5|,|-12|,|-13|,|-10|,|-11|,|-5|,|-18|,|-1|) = 18$. The input length $l = 18$, and the maximal predicted length $p_{\max} = 18$. The number of data samples of training data is $N_X = 0.9 \times \left(N - l\right) = 0.9 \times (7200 - 2 \times 18) = 6447$, and the test data is $N_C = 0.1 \times \left(N - l\right) = 0.1 \times (7200 - 2 \times 18) = 717$.

*5.2 Benchmark models*

To verify the validity of our proposed DC-LSTM model, several well-known and effective chaotic time series forecasting models such as the LSTM time series forecasting model, the Multi-task convolutional neural network (MTCNN) model, the encoder-decoder model and the convolutional recurrent Neural Network (CRNN) model are selected for comparison.

As an improved RNN, the LSTM model (Hochreiter and Schmidhuber 1997) solves the problems of gradient disappearance, gradient explosion and insufficient long-term memory of a typical RNN, an it has been often used in time series prediction. As a baseline, input shape is set to be $(l,1)$, and the units of LSTM layer is set to be 32.

The encoder-decoder model (Wang and Zhang 2018) uses one LSTM model to encode the time series in the encoder part, and then uses another LSTM model to decode the coding vector in the decoder part. The encoder-decoder model has made great progress in the machine translation field, but it can also be used in time series prediction. As a baseline, input shape is set to $(l,1)$, and the units of LSTM layer in encoder and decoder are both set to 32.

The MTCNN model (Pang et al. 2017) uses the Convolutional Neural Networks (CNNs) to extract features from the multi-variable time series data, and then uses the feature data as an input of the Artificial Neural Network (ANN) for non-linear processing, so as to realize the prediction of time series data. The MTCNN model utilizes the feature extraction ability of CNN to improve the prediction accuracy of time series data. As a baseline, input shape is set to $(\mathrm{m},l,1)$. The filters is set to 8 and kernel_size is set to $(1,2)$ of Conv2D layer, next come a MaxPooling2D layer whose pool_size is $(1,2)$, then two Dense layers and one Dropout layer are behind the merge of $\mathrm{m}-1$ MaxPooling2D layers' output.

The CRNN model (Cirstea et al. 2018) uses CNN to extract the correlation features of time series data and uses an RNN to capture the time dependence of time series data. First, CNN is applied to extract the features from the multi-variable time series data, and then the RNN is applied to the results of feature extraction to realize the time series prediction. As a baseline, input shape is set to $(\mathrm{m},l,1)$. The filters is set to 8 and kernel_size is set to $(1,2)$ of Conv2D layer, next come a MaxPooling2D layer whose pool_size is $(1,2)$, then LSTM layer whose units is set to 32 is behind the merge of $\mathrm{m}-1$ MaxPooling2D layers' output.

As a control group, the Yesterday model use the current data as the $p$ steps prediction data to verify whether the prediction results are affected only by the current time data, where $p \in [1, p_{\max}]$.

About DC-LSTM model, input shape is set to $(\mathrm{m},l,1)$, $l-1$ AtrousConv2D layers whose *kernel_size*=$(2,2)$ and *dilated_rate*=$(i,1), i=1,2,\ldots,l$ and $l-1$ LSTM layers whose units is set to $l$ are set, then LSTM layer whose units is set to 32 is behind the merge of $\mathrm{m}$ LSTM layers's output.

All experiments are carried out in the Python compiling environment using an Intel Core i7-7770 CPU machine. Implementation of all benchmark and DC-LSTM models rely on the Python package: pandas, numpy. All Deep-Learning based models are developed using the Keras (https://github.com/keras-team/keras).

*5.3 Performance metrics*

We use multiple criteria to evaluate our model, including the rooted mean squared error (RMSE) and the mean absolute error (MAE), both of which are widely used in regression tasks. The RMSE and MAE are respectively defined by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i^1 - \hat{x}_i^1\right)^2}\,, \tag{3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|x_i^1 - \hat{x}_i^1\right|, \tag{4}$$

where $N$ denoted the number of test data samples, $x_i^1$ denoted the true value of the $i^{th}$ test data sample, and $\hat{x}_i^1$ denoted the predicted value of the $i^{th}$ test data sample.

## 6. Results and discussion

### *6.1 Model comparison*

Through the training, the parameters of the LSTM, encoder-decoder, CRNN, MTCNN, and DC-LSTM are optimized, for example, the 'epochs' of DC-LSTM model is 10 and the 'batch_size' is $0.1N_X$. The error comparisons of the conditional monitoring dataset and the compressor group dataset are shown in Table 4 and Table 5. Notably, the RMSE and MAE values in Tables 4 and 5 denote the average values of 20 successive model predictions under the same conditions to avoid the randomness of the results. The column of 'Improved %' indicates the % improvement from the best result among the benchmark models to DC-LSTM model. The raw of 'Average Improved %' means the average % improvement of all $p, p = 1, 2, ..., p_{\max}$ steps.

**Table 4** The average error comparison for the conditional monitoring dataset of 20 runs

| Steps | Error | LSTM | Encoder -decoder | MTCNN | CRNN | Yesterday | DC -LSTM | Improved % |
|---|---|---|---|---|---|---|---|---|
| 1 step | RMSE | 1.156 | 1.144 | **1.039** | 1.235 | 6.701 | **0.819** | **21.174%** |
| | MAE | 0.738 | 0.741 | **0.615** | 0.804 | 5.418 | **0.315** | **48.780%** |
| 2 steps | RMSE | 1.419 | 2.026 | **1.079** | 1.084 | 8.488 | **0.834** | **22.706%** |
| | MAE | 1.218 | 1.497 | 0.669 | **0.667** | 6.573 | **0.350** | **47.526%** |
| 3 steps | RMSE | 2.642 | 2.611 | **1.070** | 1.374 | 9.319 | **0.795** | **25.701%** |
| | MAE | 1.880 | 1.732 | **0.684** | 0.957 | 7.756 | **0.270** | **60.526%** |
| 4 steps | RMSE | 2.676 | 1.879 | **1.032** | 1.230 | 8.485 | **0.808** | **21.705%** |
| | MAE | 1.937 | 1.373 | **0.642** | 0.847 | 6.573 | **0.309** | **51.869%** |
| 5 steps | RMSE | **0.836** | 0.898 | 1.112 | 1.078 | 6.702 | **0.832** | **0.478%** |
| | MAE | **0.380** | 0.472 | 0.687 | 0.692 | 5.419 | **0.344** | **9.474%** |
| Average Improved % | RMSE | | | | **18.353%** | | | |
| | MAE | | | | **43.635%** | | | |

**Table 5** The average error comparison for the compressor group dataset of 20 runs

| Steps | Error | LSTM | Encoder -decoder | MTCNN | CRNN | Yesterday | DC -LSTM | Improved % |
|---|---|---|---|---|---|---|---|---|
| 1 step | RMSE | 0.063 | 0.108 | 0.409 | 0.226 | **0.024** | **0.022** | **6.925%** |
| | MAE | 0.066 | 0.100 | 0.402 | 0.215 | **0.015** | **0.020** | **-35.630%** |
| 2 steps | RMSE | 0.047 | 0.096 | 0.474 | 0.241 | **0.043** | **0.013** | **69.653%** |
| | MAE | 0.054 | 0.083 | 0.468 | 0.227 | **0.027** | **0.010** | **62.711%** |
| 3 steps | RMSE | 0.071 | 0.157 | 0.317 | 0.190 | **0.059** | **0.023** | **61.315%** |
| | MAE | 0.077 | 0.151 | 0.311 | 0.179 | **0.038** | **0.021** | **44.088%** |
| 4 steps | RMSE | 0.132 | 0.196 | 0.522 | 0.205 | **0.074** | **0.033** | **55.650%** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | MAE | 0.136 | 0.190 | 0.517 | 0.190 | **0.047** | **0.029** | **38.517%** |
| 5 steps | RMSE | **0.083** | 0.146 | 0.345 | 0.263 | 0.088 | **0.033** | **60.357%** |
| | MAE | 0.088 | 0.138 | 0.337 | 0.250 | **0.056** | **0.027** | **51.964%** |
| 6 steps | RMSE | **0.078** | 0.114 | 0.485 | 0.180 | 0.101 | **0.045** | **42.608%** |
| | MAE | 0.083 | 0.106 | 0.478 | 0.165 | **0.064** | **0.035** | **45.519%** |
| 7 steps | RMSE | **0.079** | 0.100 | 0.420 | 0.273 | 0.113 | **0.053** | **33.088%** |
| | MAE | 0.084 | 0.084 | 0.412 | 0.261 | **0.072** | **0.044** | **38.537%** |
| 8 steps | RMSE | **0.081** | 0.276 | 0.345 | 0.374 | 0.125 | **0.048** | **40.424%** |
| | MAE | 0.090 | 0.269 | 0.336 | 0.363 | **0.078** | **0.030** | **61.740%** |
| 9 steps | RMSE | **0.083** | 0.366 | 0.363 | 0.184 | 0.135 | **0.067** | **19.270%** |
| | MAE | 0.090 | 0.362 | 0.355 | 0.174 | **0.085** | **0.048** | **43.306%** |
| 10 steps | RMSE | **0.069** | 0.444 | 0.404 | 0.241 | 0.145 | **0.067** | **3.115%** |
| | MAE | **0.080** | 0.440 | 0.395 | 0.227 | 0.091 | **0.043** | **46.515%** |
| 11 steps | RMSE | **0.090** | 0.161 | 0.628 | 0.220 | 0.154 | **0.060** | **33.006%** |
| | MAE | 0.101 | 0.149 | 0.623 | 0.206 | **0.097** | **0.037** | **61.895%** |
| 12 steps | RMSE | 0.087 | **0.075** | 0.480 | 0.236 | 0.163 | **0.072** | **3.383%** |
| | MAE | 0.100 | **0.053** | 0.474 | 0.223 | 0.103 | **0.043** | **19.219%** |
| 13 steps | RMSE | 0.124 | **0.079** | 0.189 | 0.202 | 0.171 | **0.087** | **-9.960%** |
| | MAE | 0.133 | **0.059** | 0.176 | 0.190 | 0.109 | **0.054** | **8.448%** |
| 14 steps | RMSE | **0.070** | 0.324 | 0.349 | 0.214 | 0.179 | **0.070** | **-0.488%** |
| | MAE | **0.087** | 0.317 | 0.339 | 0.203 | 0.115 | **0.050** | **42.634%** |
| 15 steps | RMSE | **0.080** | 0.268 | 0.382 | 0.223 | 0.186 | **0.076** | **5.105%** |
| | MAE | **0.094** | 0.259 | 0.374 | 0.211 | 0.120 | **0.049** | **47.974%** |
| 16 steps | RMSE | **0.073** | 0.152 | 0.482 | 0.205 | 0.192 | **0.053** | **26.916%** |
| | MAE | **0.087** | 0.141 | 0.474 | 0.194 | 0.125 | **0.030** | **65.449%** |
| 17 steps | RMSE | **0.083** | 0.309 | 0.421 | 0.319 | 0.197 | **0.070** | **15.156%** |
| | MAE | **0.091** | 0.301 | 0.413 | 0.308 | 0.130 | **0.044** | **51.662%** |
| 18 steps | RMSE | **0.104** | 0.123 | 0.299 | 0.166 | 0.203 | **0.070** | **33.003%** |
| | MAE | **0.121** | 0.112 | 0.287 | 0.151 | 0.135 | **0.042** | **62.561%** |
| Average Improved % | RMSE | | | | **27.696%** | | | |
| | MAE | | | | **42.062%** | | | |

About the conditional monitoring dataset, as can be seen in Table 3, the RMSE and MAE values of DC-LSTM model are lower than the minimum values of the other five models in all 1-, 2-, 3-, 4- and 5- steps prediction. And the RMSE and MAE of the DC-LSTM method were respectively reduced by 18.353% and 43.635% on average compared with the minimum values of the other five methods. In addition, as shown in Table 4, except the RMSE in 13- , 14- steps and the MAE in 1- step, other steps' RMSE and MAE of DC-LSTM model are lower than the minimum values of the other five models, so that 27.696% of RMSE and 42.062% of MAE decreased on average. The above results verify the effectiveness and superiority of the DC-LSTM method in the multi-step time series prediction of the conditional monitoring dataset.

*6.2 Results of multi-step continuous prediction based on DC-LSTM model*

As shown in Figure 6, after training the DC-LSTM model with the conditional monitoring dataset, the data of 12000$^{th}$, 12200$^{th}$, 12400$^{th}$, 12600$^{th}$, 12800$^{th}$ and 13000$^{th}$ were predicted successively in five steps. In Figure 6, it can be seen that the multi-step continuous prediction results of the DC-LSTM model were good, and the dynamic characteristics of time series data could be captured. Meanwhile, the direct prediction results of 18 steps for the testing data are shown in Figure 7. Using the combination of the DC-LSTM models, the compressor data of 6600$^{th}$, 6700$^{th}$, 6800$^{th}$, 6900$^{th}$, 7000$^{th}$ and 7100$^{th}$ were predicted in 18 steps. As shown in Figure 7, the predicted results could capture the development trend of real data.

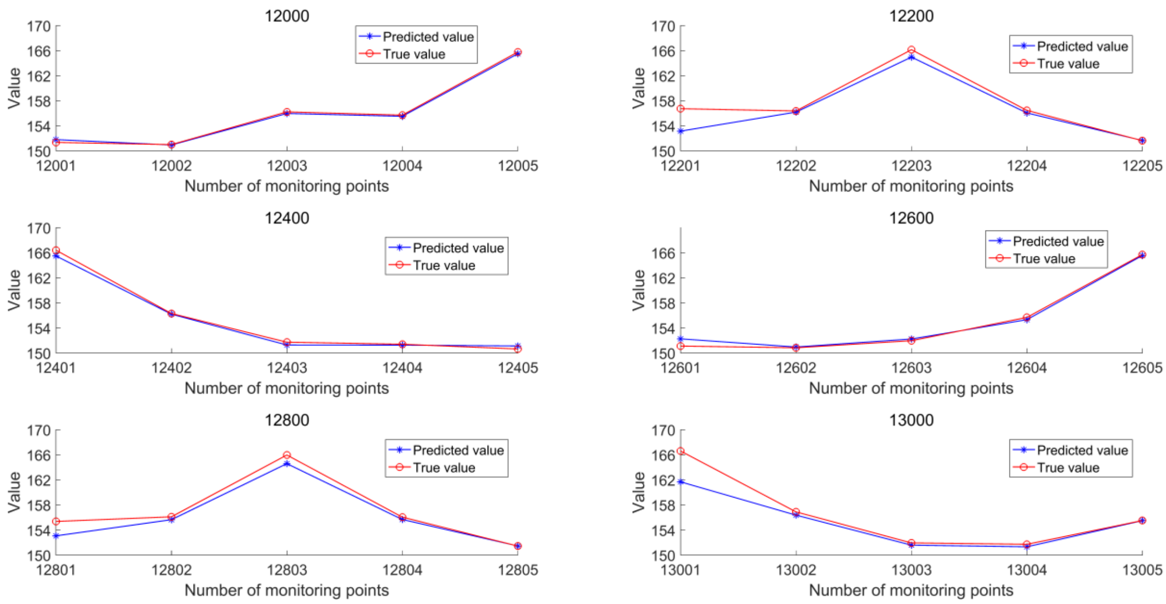



**Figure 6.** The 5-step continuous prediction of the conditional monitoring dataset by DC-LSTM

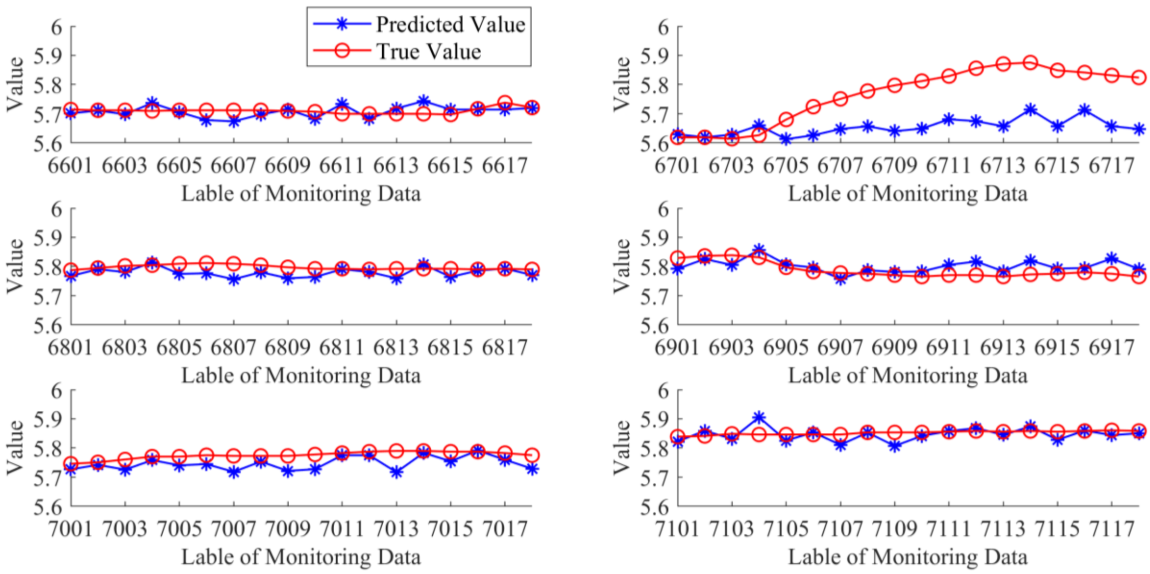



**Figure 7.** The 18-step continuous prediction result of the compressor group dataset by DC-LSTM

*6.3 Discussion*

(1) Complexity analysis

As described in the former sections, the LSTM model mainly consists of a LSTM layer, the Encoder-decoder model mainly consists of two LSTM layers, the MTCNN model mainly consists of $m-1$ Conv2D layers and two Dense layers, the CRNN model mainly consists of $m-1$ Conv2D layers and a LSTM layer, and the proposed DC-LSTM model consists of $l-1$ AtrousConv2D layers and $l+m-1$ LSTM layers. The number of training parameters about all benchmark models and the proposed DC-LSTM model are illustrated in Table 6. It can be seen that there is no order of magnitude difference in training parameters between the DC-LSTM model and other models. In other words, the training complexity of the DC-LSTM model approximately equals to that of other models. The reason for this is that the training parameters of a AtrousConv2D layer are far less than a Conv2D layer because of many parameters in a AtrousConv2D layer are preset to 0 and don't need to be trained.

**Table 6** The training parameters of all benchmark models and DC-LSTM model

| Model | LSTM | Encoder -decoder | MTCNN | CRNN | DC -LSTM |
|---|---|---|---|---|---|
| Training parameters | 4385 | 8737 | 6360 | 5922 | **7146** |

(2) The advantage of the proposed DC-LSTM

In the prediction of chaotic time series, the calculation of a reasonable maximum number of prediction steps is always challenging. Namely, if the maximum number of prediction steps is set to be too small, the prediction requirements cannot be met; on the other hand, if the maximum number of prediction steps is set to be too large, the prediction accuracy is very low. The DC-LSTM model calculates the time-delay relationship between the predicted variable and correlational variables using the CCA, so as to calculate the maximum number of prediction steps. The verification which is conducted using the conditional monitoring dataset and compressor group data showed that the maximum number of prediction steps calculated by the DC-LSTM model could satisfy the prediction requirements on the premise of guaranteeing the prediction step size.

The multistep prediction of chaotic time series can be divided into the iterative prediction and direct prediction. Due to the error accumulation of the iterative multi-step prediction, direct prediction was used to realize the multi-step prediction in DC-LSTM model. The continuous multi-step prediction of chaotic time series is realized using the combination of developed DC-LSTM models. In the verification with the conditional monitoring dataset, the RMSE and MAE values of the DC-LSTM multi-step prediction are respectively 18.353% and 43.635% lower than the minimum values of the other methods, and they are 27.696% and 42.062% about the compressor group dataset. Consequently, the proposed DC-LSTM multi-step prediction method can reduce the multi-step prediction error of chaotic time series.

(3) The future challenge

In the previous hypothesis and argument, one of the most important advantages of the DC-LSTM is the ability to take advantage of the correlation between the predicted variable and correlational variables by use dilated convolution in the DC-LSTM model. After validation of the conditional monitoring dataset and the compressor group dataset, we can conclude that extracting the correlation between the predicted variable and correlational variables is useful for multi-step prediction of chaotic time series, but it cannot quantify whether the correlation extracted by DC-LSTM model is maximized at present. So that the next step is to study how to maximize the useful information contained in multi-dimensional time series to serve the prediction of multi-step time series. Moreover, the current DC-LSTM model can get good accuracy in $[1, p_{\max}]$ steps prediction, but the accuracy cannot be guaranteed if step number is bigger than $p_{\max}$. Therefore, the next research plan is to optimize the DC-LSTM model so that it can be used in long-period chaotic time series prediction.

## 7. Conclusions

Prediction of multi-step chaotic time series is always a difficult problem. In order to determine the correlation between the predicted variable and correlational variables, the dilated convolution operation was introduced in DC-LSTM model. At the same time, the features extracted by dilated convolution operation and historical data of predicted variable are input into LSTM to achieve good multi-step prediction result. Additionally, CCA was used to calculate the reasonable maximum prediction steps of chaotic time series.

In this study, a new general framework for the multi-step prediction of chaotic time series was proposed, and data preprocessing, DC-LSTM prediction, and multi-step continuous prediction included. The RMSE and MAE of the conditional monitoring dataset and the compressor group dataset were obtained based on the trained DC-LSTM model. Comparing with the existing models, such as the LSTM model, the MTCNN model, the Encoder-decoder model and CRNN model, the DC-LSTM model proposed in this paper has superiorities in RMSE, MAE and prediction accuracy because of the extracting of correlation between the predicted variable and correlational variables. Thus, the proposed DC-LSTM model provides a new method for prediction of chaotic time series and lays a foundation for scientific data analysis of chaotic time series monitoring systems.

**Author Contributions:** Formal analysis, Rongxi Wang and Hongquan Jiang; Investigation, Rongxi Wang, Caiyuan Peng, Jianmin Gao and Zhiyong Gao; Methodology, Rongxi Wang; Project administration, Rongxi Wang and Jianmin Gao; Writing – original draft, Rongxi Wang; Writing – review & editing, Rongxi Wang..

**Funding:** This research was funded in part by the National Nature Science Foundation of China, grant number 51905409, in part by the National Key R&D program of China, grant number 2017YFF0210500, and in part by the China Postdoctoral Science Foundation project, grant number 2017M620446.

**Acknowledgments:** All authors, especially the corresponding author Rongxi Wang, would like to thank the anonymous reviewers for their time and invaluable comments and suggestions on this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

References


[1] Ai S, Chakravorty A, Rong C (2019) Household power demand prediction using evolutionary ensemble neural network pool with multiple network structures. Sensors-Basel 19 (3). doi:10.3390/s19030721

[2] Alexander J, Jr., Edwards RA, Brodsky M, Manca L, Grugni R, Savoldelli A, Bonfanti G, Emir B, Whalen E, Watt S, Parsons B (2018) Using time series analysis approaches for improved prediction of pain outcomes in subgroups of patients with painful diabetic peripheral neuropathy. Plos One 13 (12). doi:10.1371/journal.pone.0207120

[3] Awad M, Qasrawi I (2018) Enhanced RBF neural network model for time series prediction of solar cells panel depending on climate conditions (temperature and irradiance). Neural Comput Appl 30 (6):1757-1768. doi:10.1007/s00521-016-2779-5

[4] Baniata LH, Park S, Park S-B (2018) A multitask-based neural machine translation model with part-of-speech tags integration for arabic dialects. Appl Sci-Basel 8 (12). doi:10.3390/app8122502

[5] Borovykh A, Bohte S, Oosterlee CW (2019) Dilated convolutional neural networks for time series forecasting. J Comput Financ 22 (4):73-101. doi:10.21314/jcf.2018.358

[6] Cai M, Pipattanasomporn M, Rahman S (2019) Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Applied Energy 236:1078-1088. doi:10.1016/j.apenergy.2018.12.042

[7] Cao J, Li Z, Li J (2019) Financial time series forecasting model based on CEEMDAN and LSTM. Physica A-Statistical Mechanics and Its Applications 519:127-139. doi:10.1016/j.physa.2018.11.061

[8] Cao LY (1997) Practical method for determining the minimum embedding dimension of a scalar time series. Physica D 110 (1-2):43-50. doi:10.1016/s0167-2789(97)00118-8

[9] Chang S-Y, Li B, Simko G, Sainath TN, Tripathi A, van den Oord A, Vinyals O, Ieee (2018) Temporal modeling using dilated convolution and gating for voice-activity-detection. 2018 Ieee International Conference on Acoustics, Speech and Signal Processing. doi:10.1109/ICASSP.2018.8461921

[10] Chen WL, Yeo CK, Lau CT, Lee BS (2018) Leveraging social media news to predict stock index movement using RNN-boost. Data & Knowledge Engineering 118:14-24. doi:10.1016/j.datak.2018.08.003

[11] Cheng F, Li T, Wei Y-m, Fan T (2019) The VEC-NAR model for short-term forecasting of oil prices. Energy Economics 78:656-667. doi:10.1016/j.eneco.2017.12.035

[12] Chlouverakis KE, Adams MJ (2003) Stability maps of injection-locked laser diodes using the largest Lyapunov exponent. Opt Commun 216 (4-6):405-412. doi:10.1016/s0030-4018(02)02357-x

[13] Cirstea R-G, Micu D-V, Muresan G-M, Guo C, Yang B (2018) Correlated time series forecasting using multi-task deep neural networks. Cikm'18: Proceedings of the 27th Acm International Conference on Information and Knowledge Management. doi:10.1145/3269206.3269310

[14] Dhanya CT, Kumar DN (2010) Nonlinear ensemble prediction of chaotic daily rainfall. Adv Water Resour 33 (3):327-347. doi:10.1016/j.advwatres.2010.01.001

[15] Fraser AM, Swinney HL (1986) Independent coordinates for strange attractors from mutual information. Physical Review A 33 (2):1134-1140. doi:10.1103/PhysRevA.33.1134

[16] Helwig N, Pignanelli E, Schutze A, Ieee (2015) Condition monitoring of a complex hydraulic system using multivariate statistics. 2015 Ieee International Instrumentation and Measurement Technology Conference. Ieee, New York. doi:10.1109/I2MTC.2015.7151267

[17] Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9 (8):1735-1780. doi:10.1162/neco.1997.9.8.1735

[18] Hua Y, Mou L, Zhu XX (2019) Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. Isprs J Photogramm 149:188-199. doi:10.1016/j.isprsjprs.2019.01.015

[19] Karunasinghe DSK, Liong SY (2006) Chaotic time series prediction with a global model: Artificial neural network. J Hydrol 323 (1-4):92-105. doi:10.1016/j.jhydrol.2005.07.048

[20] Ke J, Zheng H, Yang H, Chen X (2017) Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transport Res C-Emer 85:591-608. doi:10.1016/j.trc.2017.10.016

[21] Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the Ieee 86 (11):2278-2324. doi:10.1109/5.726791

[22] Li Y, Zhang X, Chen D, Ieee (2018) CSRNet: Dilated Convolutional Neural Networks for understanding the highly congested scenes. In: 2018 IEEE/Cvf Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition. pp 1091-1100. doi:10.1109/cvpr.2018.00120

[23] Liang YJ, Gao ZY, Gao JM, Wang RX, Zhao H (2018) Data fusion combined with echo state network for multivariate time series prediction in complex electromechanical system. Comput Appl Math 37 (5):5920-5934. doi:10.1007/s40314-018-0669-4

[24] Liu A-A, Shao Z, Wong Y, Li J, Su Y-T, Kankanhalli M (2019) LSTM-based multi-label video event detection. Multimed Tools Appl 78 (1):677-695. doi:10.1007/s11042-017-5532-x

[25] Liu J, Zhang T, Han G, Gou Y (2018) TD-LSTM: Temporal Dependence-based LSTM networks for marine temperature prediction. Sensors-Basel 18 (11). doi:10.3390/s18113797

[26] Lopez E, Valle C, Allende H, Gil E, Madsen H (2018) Wind power forecasting based on echo state networks and long short-term memory. Energies 11 (3). doi:10.3390/en11030526

[27] Pal SS, Kar S (2017) Time series forecasting using fuzzy transformation and neural network with back propagation learning. J Intell Fuzzy Syst 33 (1):467-477. doi:10.3233/jifs-161767

[28] Pang N, Yin F, Zhang X, Zhao X (2017) A robust approach for multivariate time series forecasting. Proceedings of the Eighth International Symposium on Information and Communication Technology. ACM. doi:10.1145/3155133.3155172

[29] Pannakkong W, Sriboonchitta S, Huynh V-N (2018) An ensemble model of arima and ann with restricted boltzmann machine based on decomposition of discrete wavelet transform for time series forecasting. J Syst Sci Syst Eng 27 (5):690-708. doi:10.1007/s11518-018-5390-8

[30] Pano-Azucena AD, Tlelo-Cuautle E, Tan SXD, Ieee (2018) Prediction of chaotic time series by using ANNs, ANFIS and SVMs. 2018 7th International Conference on Modern Circuits and Systems Technologies. Ieee, New York. doi:10.1109/MOCAST.2018.8376560

[31] Perkel DH, Gerstein GL, Moore GP (1967) Neuronal spike trains and stochastic point processes .2. Simultaneous spike trains. Biophysical Journal 7 (4):419-+. doi:10.1016/s0006-3495(67)86597-4

[32] Rhodes C, Morari M (1997) The false nearest neighbors algorithm: an overview. Computers & Chemical Engineering 21:S1149-1154. doi:10.1016/s0098-1354(97)00204-4

[33] Sivakumar B (2002) A phase-space reconstruction approach to prediction of suspended sediment concentration in rivers. J Hydrol 258 (1-4):149-162. doi:10.1016/s0022-1694(01)00573-x

[34] Strubell E, Verga P, Belanger D, McCallum A (2017) Fast and accurate entity recognition with iterated dilated convolutions. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing:2670–2680. doi:10.18653/v1/D17-1283

[35] Sun X, Dong K, Ma L, Sutcliffe R, He F, Chen S, Feng J (2019) Drug-drug interaction extraction via recurrent hybrid convolutional neural networks with an improved focal loss. Entropy-Switz 21 (1). doi:10.3390/e21010037

[36] Sundermeyer M, Ney H, Schlueter R (2015) From feedforward to recurrent LSTM neural networks for language modeling. Ieee-Acm T Audio Spe 23 (3):517-529. doi:10.1109/taslp.2015.2400218

[37] Takens F Detecting strange attractors in turbulence. In, 1981. Dynamical systems and turbulence, Warwick 1980. Springer Berlin Heidelberg, pp 366-381. doi:10.1007/BFb0091924

[38] Tan K, Chen J, Wang D (2019) Gated residual networks with dilated convolutions for monaural speech enhancement. Ieee-Acm T Audio Spe 27 (1):189-198. doi:10.1109/taslp.2018.2876171

[39] Tian Y, Zhang K, Li J, Lin X, Yang B (2018) LSTM-based traffic flow prediction with missing data. Neurocomputing 318:297-305. doi:10.1016/j.neucom.2018.08.067

[40] Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P (2018) Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proceedings of the Royal Society a-Mathematical Physical and Engineering Sciences 474 (2213). doi:10.1098/rspa.2017.0844

[41] Wang B, Lei Y, Tian S, Wang T, Liu Y, Patel P, Jani AB, Mao H, Curran WJ, Liu T, Yang X (2019) Deeply supervised 3D fully convolutional networks with group dilated convolution for automatic MRI prostate segmentation. Med Phys 46 (4):1707-1718. doi:10.1002/mp.13416

[42] Wang H, Yang W, Yuan C, Ling H, Hu W (2017a) Human activity prediction using temporally-weighted generalized time warping. Neurocomputing 225:139-147. doi:10.1016/j.neucom.2016.11.004

[43] Wang J, Zhang C (2018) Software reliability prediction using a deep learning model based on the RNN encoder-decoder. Reliab Eng Syst Safe 170:73-82. doi:10.1016/j.ress.2017.10.019

[44] Wang R, Gao J, Gao Z, Gao X, Jiang H (2017b) Analysis of multifractality of multivariable coupling relationship of complex electromechanical system in process industry. Proceedings of the Institution of Mechanical Engineers Part E-Journal of Process Mechanical Engineering 231 (6):1087-1100. doi:10.1177/0954408916653149

[45] Wang R, Gao J, Gao Z, Gao X, Jiang H, Liang Z (2017c) Interaction analysis-based information modeling of complex electromechanical systems in the processing industry. Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering 231 (8):638-651. doi:10.1177/0959651817718454

[46] Wang R, Gao X, Gao J, Gao Z, Kang J (2018) An information transfer based novel framework for fault root cause tracing of complex electromechanical systems in the processing industry. Mech Syst Signal Pr 101:121-139. doi:10.1016/j.ymssp.2017.08.030

[47] Wiatowski T, Bölcskei H (2017) A mathematical theory of deep convolutional neural networks for feature extraction. IEEE Transactions on Information Theory 64 (3):1845-1866. doi:10.1109/TIT.2017.2776228

[48] Wolf A, Swift JB, Swinney HL, Vastano JA (1985) Determining Lyapunov exponents from a time series. Physica D Nonlinear Phenomena 16 (3):285-317. doi:10.1016/0167-2789(85)90011-9

[49] Xiao J, Zhu X, Huang C, Yang X, Wen F, Zhong M (2019) A New Approach for Stock Price Analysis and Prediction Based on SSA and SVM. Int J Inf Tech Decis 18 (1):287-310. doi:10.1142/s021962201841002x

[50] Yeh C-H, Lin M-H, Lin C-H, Yu C-E, Chen M-J (2019) Machine learning for long cycle maintenance prediction of wind turbine. Sensors (Basel, Switzerland) 19 (7). doi:10.3390/s19071671

[51] Zhang H, Li H, Yang B, Lu J (2017) Process time series prediction based on application of correlated process variables to CNN time delayed analyses. CIESC Journal 68 (9):3501-3510. doi:10.11949/j.issn.0438-1157.20170197

[52] Zhang J, Wang P, Yan R, Gao RX (2018) Long short-term memory for machine remaining life prediction. J Manuf Syst 48:78-86. doi:10.1016/j.jmsy.2018.05.011

[53] Zhao J, Mao X, Chen L (2019) Speech emotion recognition using deep 1D & 2D CNN LSTM networks. Biomed Signal Proces 47:312-323. doi:10.1016/j.bspc.2018.08.035

[54] Zhou TL, Gao SC, Wang JH, Chu CY, Todo Y, Tang Z (2016) Financial time series prediction using a dendritic neuron model. Knowledge-Based Syst 105:214-224. doi:10.1016/j.knosys.2016.05.031

[55] Zhou Y, Chang F-J, Chang L-C, Kao IF, Wang Y-S, Kang C-C (2019) Multi-output support vector machine for regional multi-step-ahead PM2.5 forecasting. Sci Total Environ 651:230-240. doi:10.1016/j.scitotenv.2018.09.111