



# A novel fault diagnosis method based on CNN and LSTM and its application in fault diagnosis for complex systems

Ting Huang<sup>1,2,3</sup> · Qiang Zhang<sup>1,2,3</sup>  · Xiaoan Tang<sup>1,2,3</sup> · Shuangyao Zhao<sup>1,2,3</sup> · Xiaonong Lu<sup>1,2,3</sup>

Published online: 2 April 2021

© The Author(s), under exclusive licence to Springer Nature B.V. 2021

## Abstract

Fault diagnosis plays an important role in actual production activities. As large amounts of data can be collected efficiently and economically, data-driven methods based on deep learning have achieved remarkable results of fault diagnosis of complex systems due to their superiority in feature extraction. However, existing techniques rarely consider time delay of occurrence of faults, which affects the performance of fault diagnosis. In this paper, by synthetically considering feature extraction and time delay of occurrence of faults, we propose a novel fault diagnosis method that consists of two parts, namely, sliding window processing and CNN-LSTM model based on a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM). Firstly, samples obtained from multivariate time series by the sliding window processing integrates feature information and time delay information. Then, the obtained samples are fed into the proposed CNN-LSTM model including CNN layers and LSTM layers. The CNN layers perform feature learning without relying on prior knowledge. Time delay information is captured with the use of the LSTM layers. The fault diagnosis of the Tennessee Eastman chemical process is addressed, and it is verified that the predictive accuracy and noise sensitivity of fault diagnosis can be greatly improved when the proposed method is applied. Comparisons with five existing fault diagnosis methods show the superiority of the proposed method.

**Keywords** Fault diagnosis · Convolutional neural network · Long short-term memory network · Data-driven · Deep learning · Tennessee eastman chemical process

---

✉ Qiang Zhang  
qiang\_zhang@hfut.edu.cn

✉ Xiaoan Tang  
sichuanshengxiaoan@163.com

Extended author information available on the last page of the article

## 1 Introduction

In actual production activities, the occurrence of system faults is inevitable due to internal factors (e.g., the wear of parts) or environmental factors (e.g., drastic changes in temperature). Furthermore, these are more common in complex systems that are composed of many units with diverse relations. Once a fault happens to a complex system, it usually affects the normal operation of the system and may further lead to an immeasurable loss. Therefore, it is important to develop a fault diagnosis method for complex systems.

In recent years, as large amounts of data can be collected efficiently and economically, data-driven methods based on deep learning (DL) have achieved remarkable results of fault diagnosis of complex systems. However, existing techniques rarely consider the time delay of the occurrence of faults, which affects the performance of fault diagnosis. In this paper, we synthetically consider two aspects of fault diagnosis of complex systems, namely, feature extraction and time delay of the occurrence of faults.

- (1) *Feature extraction* Raw data collected from production environments is generally high-dimensional, and the attributes of the data are usually highly correlated. These kinds of characteristics may seriously affect the performance of subsequent learning algorithms, which further influences the results of fault diagnosis. When handling fault diagnosis, one first needs to identify those features that significantly affect the occurrence of faults.
- (2) *Time delay of the occurrence of faults* The occurrence of faults is generally a cumulative process (e.g., the wear of parts), thus there may be some time delay. That is to say, the occurrence of faults at the current moment may depend on the change of the system state at the previous moments. Capturing the time delay information contributes greatly to the performance improvement of fault diagnosis.

Over the past several decades, many feature extraction methods for fault diagnosis have been reported in the literature. These methods were generally proposed based on the fault diagnosis of certain particular systems. In other words, the existing feature extraction methods for fault diagnosis are system-dependent and unavailable to other systems. Developing a feature extraction method that can be suitable for the fault diagnosis of different systems is an urgent problem to be solved. In recent years, the feature learning methods based on DL has been used for fault diagnosis. The feature learning methods can automatically learn useful and predictive implicit features hidden in massive data, which overcomes the shortcoming of feature extraction methods that rely excessively on domain knowledge.

Currently, delay fault diagnosis is mainly concentrated in the field of circuits. However, the delay fault diagnosis methods in the field of circuits cannot be applied to other fields due to their specificity. In other fields, such as fault diagnosis of chemical process, rotating machines, and so on, although various fault diagnosis methods have been developed in the past decades, they rarely consider time delay of the occurrence of faults. However, it is obvious that time delay of the occurrence of faults exists widely in these systems. Therefore, there is an urgent need to develop a method to deal with time delay of the occurrence of faults.

DL has made breakthroughs in the fields of image recognition, speech recognition, machine translation, and so on. Convolutional Neural Network (CNN) is an important technology of DL that was first used in the field of image recognition (Krizhevsky et al. 2017). Through a series of convolution operations, the CNN automatically extracts features

layer by layer. Recurrent Neural Network (RNN) is also an important DL technique. However, traditional RNN cannot handle long-term time correlations because of the problem of recursion, weighted exponential explosion, or disappearance (Kolen and Kremer 2009). Long Short-Term Memory Network (LSTM) is a special RNN that contains LSTM blocks that are smart units that can remember the value of the uncertain length of time. This property of the LSTM ensures that it can capture the time delay information. The use of DL to solve key problems in other fields has also been well studied. Considering the superiority of DL, it is interesting to study its application to fault diagnosis.

Based on the above considerations, to manage the two aspects related to fault diagnosis of complex systems, this paper develops a DL-based fault diagnosis method by combining CNN and LSTM. In the proposed method, feature information and time delay information are first comprehensively integrated into one type of 2D image-like data obtained from a multivariate time series (MTS) by sliding window processing that is then regarded as the input of the CNN. Using CNN layers that can perform automatic feature learning, the feature maps as the input of LSTM layers are identified. The time delay information that is hidden in the 2D image-like data can be captured with the use of the LSTM layers. Following that, the learned feature information and the captured time delay information are integrated via fully connected layers as the basis for fault diagnosis. The proposed method ultimately outputs the occurrence probability of each type of pre-defined faults. Results show that the proposed method can greatly improve the performance of fault diagnosis of complex systems. Comparisons with several existing fault diagnosis methods, such as the CNN, LSTM, Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) methods, demonstrate the superiority of the proposed method.

In summary, the main contributions of this study are shown as follows:

- (1) The feature information and time delay information of MTS are integrated into one type of 2D image-like data by the proposed sliding window processing, which provides sufficient fault diagnosis information for DL model.
- (2) Based on the 2D image-like data obtained by sliding window processing, a novel DL model combining CNN and LSTM for fault diagnosis of complex systems is proposed, which achieves the goal of feature learning and capturing the time delay of the occurrence of faults.
- (3) The fault diagnosis of the Tennessee Eastman (TE) chemical process is addressed based on the proposed method, and it is verified that the predictive accuracy and noise sensitivity of fault diagnosis can be greatly improved.

This paper is organized as follows: Some existing fault diagnosis methods are reviewed in Sect. 2. Section 3 presents the proposed CNN-LSTM fault diagnosis method. The implementation of the proposed method to deal with the fault diagnosis of the TE chemical process is illustrated in Sect. 4. Conclusions and future work are provided in Sect. 5.

## 2 Literature review

Traditional fault diagnosis methods, such as physics of failure (Li et al. 2018a, b; Yang et al. 2013; Zhu et al. 2016) and fault tree analysis (Kabir 2017), generally focus on the operating mechanism or theoretical analysis of systems. They have been widely used in such fault diagnosis fields with high reliability requirements as aerospace and electronic.

However, when faced with such complex systems as chemical process, they are usually not feasible because it is difficult to mathematically model or analyze for these complex systems.

With the development of sensor technology, lots of data related to system operation can be easily collected and acquired. These heterogeneous and multi-source data may involve rich information about faults. As such, in recent years, with the use of these collected data, many data-driven methods (Cai et al. 2016; Cai et al. 2017b; El-Koujok et al. 2014; Li 2018; Serdio et al. 2015; Zhang et al. 2020) have been developed for fault diagnosis and they have been widely applied in a variety industry sectors. In particular, the data-driven methods based on DL (Lei et al. 2016; Li 2018; Li et al. 2019a; Rodríguez Ramos et al. 2019; Zhang et al. 2017; Wang et al. 2020) have achieved remarkable results of fault diagnosis of complex systems due to their superiority in feature extraction.

In general, data-driven fault diagnosis can be treated as a classification task (Aydin et al. 2012). The use of the data-driven methods for fault diagnosis usually involves the following four steps: data preprocessing, feature extraction, classifier building, and fault diagnosis.

## 2.1 Data preprocessing

The raw data generally cannot be directly used as the input to the data-driven fault diagnosis methods, which implies that there is the need for data conversion to meet the input requirements of these methods. In recent years, many data conversion methods have been developed to meet the input requirements of the fault diagnosis method. For example, Liu et al. (2019) proposed an input tensor transformation scheme to transform MTS into appropriate tensor representation so that the model based on CNN can handle these data. A Signal-to-Image conversion method was proposed by Wen et al. (2018) for the fault diagnosis based on CNN. The data conversion methods proposed in these studies are differences due to their different model input requirements. Therefore, data conversion methods need to be designed according to model input requirements.

## 2.2 Feature extraction

Feature extraction is a vital step to extract the features that can accurately and completely cover the information of the original data by reducing the dimension of the data and the correlation between the attributes. Over the past several decades, many feature extraction methods for fault diagnosis have been reported in the literature (Gao and Hou, 2016; Hong and Dhupia, 2014; Jing and Hou 2015; Rai and Mohanty, 2007; Yan et al. 2014). These methods were generally proposed based on the fault diagnosis of certain particular systems. For example, wavelets have been commonly used for fault diagnosis of rotating machines (Yan et al. 2014) while Principal Component Analysis (PCA) has been used in chemical systems (Jing and Hou 2015). Feature learning is a useful way to replace feature extraction. The feature learning method can automatically learn the features suitable for the problems at hand so that it overcomes the shortcoming that the feature extraction method depends heavily on specific problem and domain knowledge. In recent years, with the development of DL, DL-based feature learning methods for fault diagnosis have been extensively studied. For example, Janssens et al. (2016) proposed a DL model for condition monitoring by using CNN and proved that the feature learning method was significantly better than the feature extraction method in the fault diagnosis of rotating machines. A novel hierarchical

learning rate adaptive deep CNN model was proposed by Guo et al. (2016) to solve the problem of extracting features automatically without significantly increasing the demand for machine expertise and the problem of maximizing accuracy without overcomplicating machine structure. Furthermore, Razavian et al. (2014) demonstrated that the generic features learned from CNN are very powerful.

## 2.3 Classifier building

The aim of this step is to build a classifier that can be used for fault diagnosis based on the extracted features. Many classification algorithms can be used to conduct the construction of the classifier, such as KNN (Casimir et al. 2006; Lei and Zuo 2009), SVM (Goyal et al. 2020), ANN (Seera et al. 2016), and so on. In recent years, with the development of DL, the multilayer neural network-based classifiers for fault diagnosis have emerged. In the construction of those classifiers, such as RNN-based fault diagnosis classifiers (de Bruin et al. 2017; Liu et al. 2018) and CNN-based fault diagnosis classifiers CNN (Feng et al. 2021; Wen et al. 2018; Wu and Zhao 2018), the FC layers are generally used to map the feature vector to the fault probability space and perform fault classification.

## 2.4 Fault diagnosis

In this step, the fault state of the system can be predicted based on the output of the constructed classifier.

On the other hand, for various systems, such as electric, pneumatic, hydraulic networks, chemical processes, long transmission lines, robotics, and so on, there may be some time delay of the occurrence of faults. In the fault diagnosis of these systems, the existence of time delay of the occurrence of faults must be considered; otherwise, the performance of fault diagnosis will drop significantly. Delay fault diagnosis has been extensively studied in circuit systems (Sivaraman and Strojwas 2001; Wang et al. 2005). More recently, a fault prediction method is proposed based on FFT, PCA, and CNN for delay circuit systems in the study (Khalil et al. 2020). In the fault diagnosis of complex electronic systems, Cai et al. (2017a) used Dynamic Bayesian network to model the dynamic degradation process of electronic products considering that the performance of electronic products degrades over time. In the fault diagnosis of rolling bearings, RNN has been used to handle the time dependence of signals (Liu et al. 2018). More generally, Gers et al. (2002) proved that LSTM can solve the problem of time delay.

From the above analysis, we can see that DL-based fault diagnosis methods are more suitable for complex systems. In recent years, although some DL-based fault diagnosis methods have been proposed, they rarely consider the time delay of the occurrence of faults. Therefore, it is interesting to propose a DL-based fault diagnosis method for complex systems that considers both feature extraction and time delay of the occurrence of faults.

## 3 Proposed DL-based fault diagnosis method for complex systems

As shown in Fig. 1, the framework of the proposed DL-based fault diagnosis method for complex systems consists of two part, namely, sliding window processing and CNN-LSTM model. In what follows, the two parts will be introduced in detail.

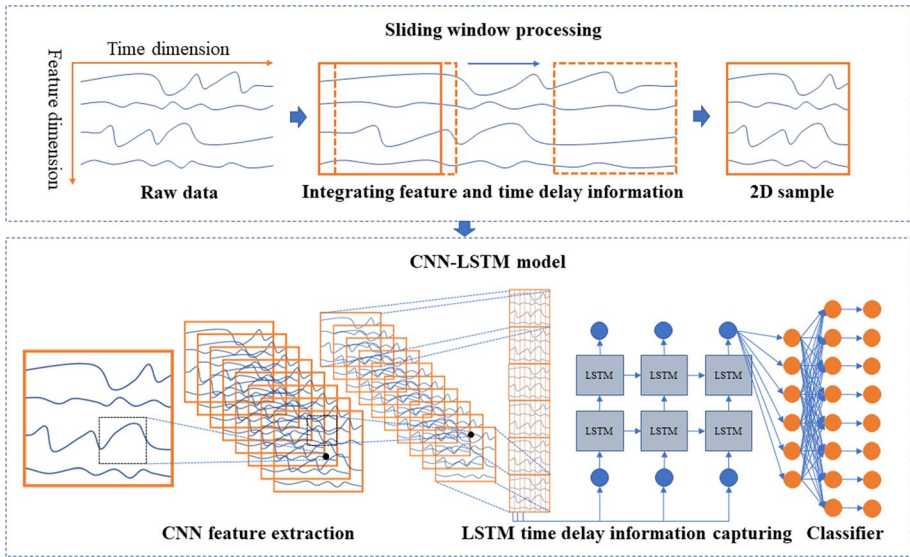


Fig. 1 The framework of the proposed DL-based fault diagnosis method for complex systems

### 3.1 Sliding window processing

In the sliding window processing, 2D samples can be obtained based on the raw data  $X$  (Some data preprocessing has been done, such as normalization) by simultaneously considering its feature information and time delay information. A schematic diagram of sliding window processing is shown in Fig. 2.

Let  $X = \{x_t | t = 1, 2, 3 \dots n\}$  ( $x_t \in R^m$  and  $X \in R^{n \times m}$ ) be the raw data that is an MTS, where  $x_t$  is the observation vector of the system at the time  $t$ ,  $m$  represents the

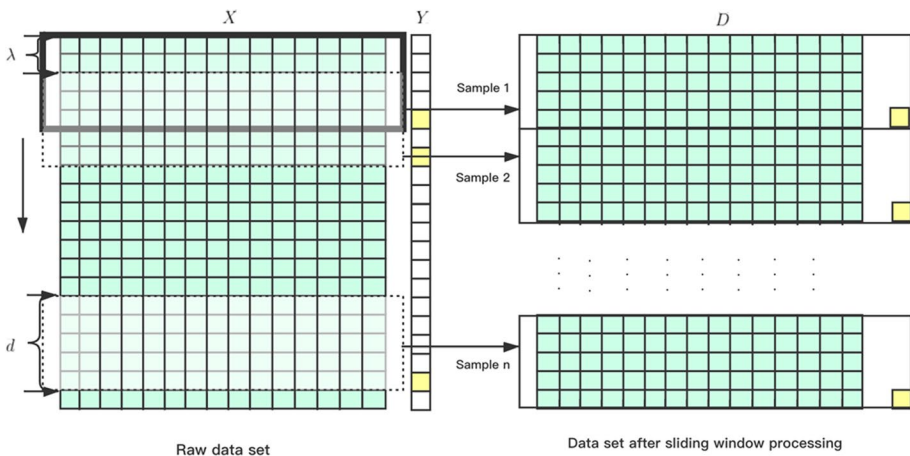


Fig. 2 Schematic diagram of sliding window processing

number of observation attributes, and  $n$  denotes the length of the observation time. Let  $Y = \{y_t | t = 1, 2, 3 \dots n\}$  ( $y_t \in R$  and  $Y \in R^n$ ) be the label of  $X$ . Then, samples of the raw data can be expressed as  $D = \{(X, Y)\}$ . As shown in Fig. 2, a sliding window is represented by a rectangular frame whose length is the number of attributes of  $X$  and width is  $d$  ( $0 < d \leq n$ ). The samples used to train the model are continuously obtained from  $X$  by moving the sliding window. The step size of the movement of the sliding window is an integer  $\lambda$  ( $0 < \lambda \leq n - d$ ). At the  $t$  th movement of the sliding window, assuming that the framed sub-series of  $X$  is denoted as  $X[t : t + d]$ , then the label of  $X[t : t + d]$  is  $Y[t + d]$ , where  $[i:j]$  denotes the operation that extracts elements between  $i$  and  $j$  from an ordered set. In this way, the sample  $(X[t : t + d], Y[t + d])$  will be used as the input feature maps of the proposed model.

The original 1D samples (data of a certain moment of MTS) only contain feature information. After sliding window processing, the 1D samples are converted into 2D samples, which integrates feature information and time delay information. Practical meaning of these 2D samples can be interpreted as that the system state at the current time may be related to the system states of previous  $d$  moments. In this way, using these 2D samples to train the fault diagnosis classification model can make it learn feature information and time delay information simultaneously, which can greatly improve the performance of fault diagnosis. Moreover, the size of feature information and time delay information contained in these 2D samples are adjustable. Specifically, the size of the feature information can be adjusted by adjusting the size of  $m$ . The size of the time delay information can be adjusted by adjusting the size of  $d$ . By integrating more attributes (greater  $m$ ) and longer time delay (greater  $d$ ) in sliding window processing, the 2D samples more suitable for fault diagnosis of complex systems can be obtained.

The obtained 2D samples are used as the learning material of CNN-LSTM model to identify the fault category. In what follows, we will present CNN-LSTM in detail.

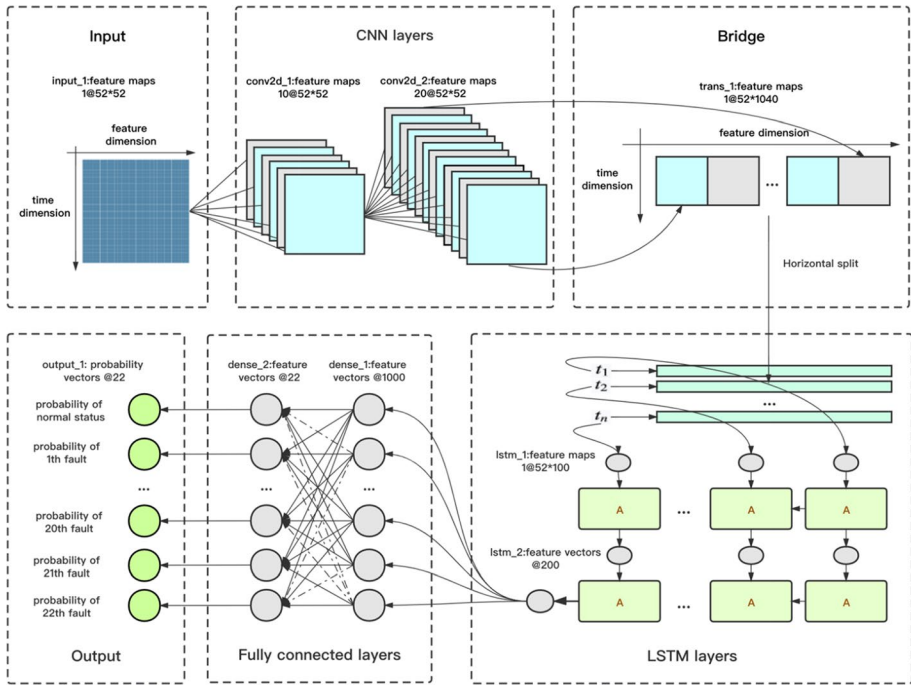
### 3.2 CNN-LSTM model

This subsection investigates the proposed CNN-LSTM model, which consists of CNN layers, LSTM layers, and fully connected layers as shown in Fig. 3. In what follows, the proposed model will be described in detail from these layers.

#### 3.2.1 CNN layers

The CNN layers consist of two operation, namely, convolution operation and activation operation. The input of the convolution operation is a 3D data  $X \in R^{w \times h \times c}$ , where  $w$ ,  $h$ , and  $c$ , respectively represent its width, its height, and the number of channels. In particular, when  $c = 1$ , it is reduced to 2D data, which is used as the input of the proposed model. Specifically, the horizontal axis of the 2D data represents the feature dimension and the vertical axis depicts the time dimension. The axis scales are respectively characterized by the numbers of attributes and the width of the sliding window.

To illustrate the convolution operation clearly, we first define four super-parameters, namely, the number of convolution kernels  $K$ , the size of each convolution kernel  $F$ , the step size of the convolution  $S$ , and the number of zeros  $P$ . Then, for the convolution operation of each convolution kernel, zeros are filled into the data  $X \in R^{w \times h \times c}$  by which the transformed data  $X^* \in R^{(w+P) \times (h+P) \times c}$  is obtained. After that, the slicing operation is:



**Fig. 3** Architecture diagram of CNN-LSTM model for fault diagnosis of TE chemical process

$$X_{t_w, t_h} = X^* [t_w : t_w + F, t_h : t_h + F, :], \tag{1}$$

where  $t_w : t_w + F$  indicates the operation of picking up the subset positioned between Rows  $t_w$  and  $t_w + F$ ,  $t_h : t_h + F$  indicates the operation of picking up the subset positioned between Columns  $t_h$  and  $t_h + F$ , and  $:$  indicates cutting all data in the channel direction.  $t_w$  and  $t_h$  take values from 1 and change by step  $S$ . As per the  $(t_w, t_h)$ , data  $X_{t_w, t_h}$  can be extracted from  $X^*$ . Assume the convolution kernel is denoted as  $Kernel \in R^{F \times F \times C}$ , then  $T_{t_w, t_h} \in R^{F \times F \times C}$  is defined as element-by-element multiplication of  $Kernel$  and  $X_{t_w, t_h}$  as follows:

$$T_{t_w, t_h} = Kernel \times X_{t_w, t_h}, \tag{2}$$

Subsequently,  $y_{t_w, t_h}$  can be obtained in the following manner:

$$y_{t_w, t_h} = \sum_{i=0}^F \sum_{j=0}^F \sum_{k=0}^C T_{t_w, t_h} [i, j, k], \tag{3}$$

where  $T_{t_w, t_h} [i, j, k]$  denotes the element of  $T_{t_w, t_h}$ , and  $Y \in R^{w^* \times h^*}$  can be obtained by  $y_{t_w, t_h}$  arranged in the order of the size of  $t_w, t_h$ . It can be noted that

$$w^* = \frac{w-F+2*P}{S} + 1, h^* = \frac{h-F+2*P}{S} + 1. \tag{4}$$

For each convolution kernel, one can obtain a channel such as  $Y$ , and then by repeating the above operations on the  $K$  convolution kernels, we can obtain a new 3D data



$\hat{X} \in R^{w^* \times h^* \times K}$ . In general, a specific combination of  $(F, P, S)$  is set to make  $w^* = w$  and  $K > c$ .

After the convolution operation, the activation operation is essential. It enables the network to acquire a nonlinear expression of the input to enhance the representation ability and make the learned features more dividable. With the use of rectified linear unit, which is widely a used one, one can activate each element of  $\hat{X}$  in the following manner:

$$A[i, j, k] = f(\hat{X}[i, j, k]) = \max \{0, \hat{X}[i, j, k]\}, \tag{5}$$

where  $\hat{X}[i, j, k]$  denotes the element of  $\hat{X}$ , and  $A[i, j, k]$  denotes the element of  $A \in R^{w^* \times h^* \times K}$  that is the data obtained after activation operation.

### 3.2.2 Connection between CNN and LSTM layers

The output of CNN layers is 3D data in which the extracted 2D feature maps are stacked together. The required input of LSTM layers is 2D data, where one dimension is denoted as time and the other is features. In order to connect the CNN layers and the LSTM layers, it is needed to convert the 3D data output from CNN layers into 2D data input to LSTM layers.

In this study, a bridge is developed to connect the CNN layers and the LSTM layers. The operation of the bridge is shown in Fig. 4. In the bridge, the output of the CNN layers is rearranged as the input of the LSTM layers. The bridge first arranges the channels of the 3D data output from CNN layers so that all the extracted features are arranged together while keeping the time dimension unchanged. After this arrangement, the 3D data becomes 2D data. Then, the bridge cuts this 2D data horizontally to obtain samples for each time step of the LSTM layers.

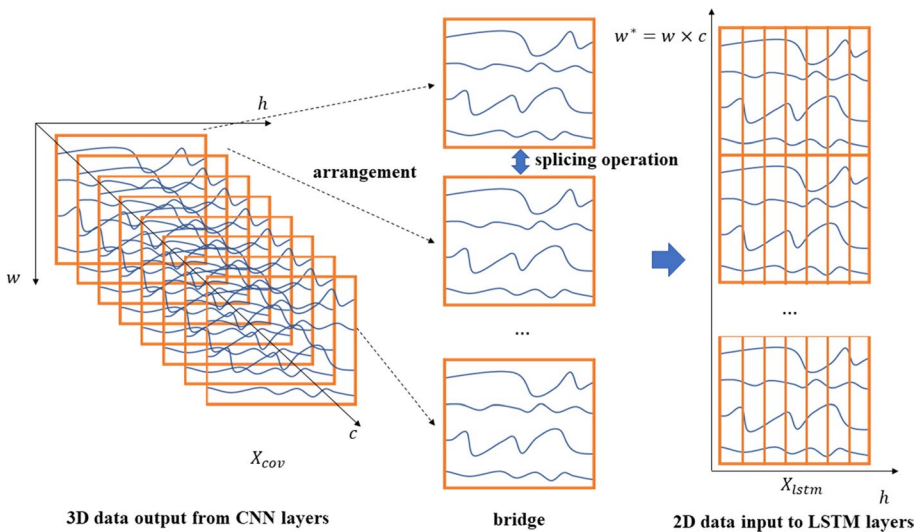


Fig. 4 The bridge used to connect the CNN layers and the LSTM layers

Specifically, Suppose the output of the CNN layers is  $X_{cov} \in R^{w \times h \times c}$ , then the input of the LSTM layers obtained by the bridge can be denoted as  $X_{lstm} \in R^{w^* \times h}$ , where  $w^* = w \times c$ .

$$X_{lstm} = X_{cov}[:, :, 0] \leftrightarrow X_{cov}[:, :, 1] \dots X_{cov}[:, :, c - 2] \leftrightarrow X_{cov}[:, :, c - 1], \tag{6}$$

where  $\leftrightarrow$  denotes the splicing operation of 2D matrixes in the row direction.

### 3.2.3 LSTM layers

An LSTM layer generally involves several smart units, each of which contains three gates, namely, forget gate, input gate, and output gate. The forget gate tells which information should be forgotten, the input gate determines which inputs need to be remembered, and the output gate decides which information needs to be output. The specific calculation process of these three gates is shown in Fig. 5.

At the forget gate, the information  $f_t$  that needs to be discarded can be recognized in the following way:

$$f_t = \sigma(W_f \cdot x_t + R_f \cdot h_{t-1} + b_f) = \frac{1}{1 + e^{-(W_f \cdot x_t + R_f \cdot h_{t-1} + b_f)}}, \tag{7}$$

where  $x_t$  denotes the data at time  $t$ ,  $h_{t-1}$  denotes the output at time  $t - 1$ ,  $W_f$  and  $R_f$  are the weight matrices associated with  $f_t$ ,  $b_f$  is the corresponding bias vector, and  $\cdot$  indicates the dot product operation.

At the input gate, the information  $i_t$  that needs to be inputted and the candidate  $\tilde{C}_t$  for the state value of the unit are identified based on the following operation:

$$i_t = \sigma(W_i \cdot x_t + R_i \cdot h_{t-1} + b_i) = \frac{1}{1 + e^{-(W_i \cdot x_t + R_i \cdot h_{t-1} + b_i)}} \tag{8}$$

and

$$\tilde{C}_t = \varphi(W_c \cdot x_t + R_c \cdot h_{t-1} + b_c) = \frac{e^{(W_c \cdot x_t + R_c \cdot h_{t-1} + b_c)} - e^{-(W_c \cdot x_t + R_c \cdot h_{t-1} + b_c)}}{e^{(W_c \cdot x_t + R_c \cdot h_{t-1} + b_c)} + e^{-(W_c \cdot x_t + R_c \cdot h_{t-1} + b_c)}}, \tag{9}$$

where  $W_i$  and  $R_i$  are the weight matrices associated with  $i_t$ ,  $b_i$  is the corresponding bias vector,  $W_c$  and  $R_c$  are the weight matrices associated with  $C_t$ , and  $b_c$  is the corresponding bias vector.

At the output gate, by integrating  $f_t$ ,  $i_t$ ,  $\tilde{C}_t$  and the state value  $C_{t-1}$  of the unit at time  $t - 1$ , the unit's state value  $C_t$  at time  $t$  can be obtained as follows:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \tag{10}$$

where  $\times$  indicates the element-by-element multiplication of vectors.

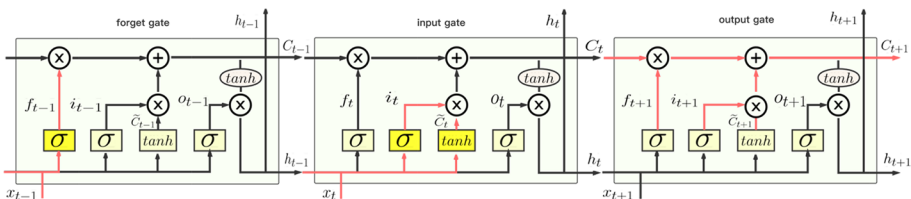


Fig. 5 One LSTM layer

Finally, with the aid of the latest state value  $C_t$ , the input  $x_t$  of the unit, and the output  $h_{t-1}$  of the previous unit, one can get the output  $h_t$  of the unit as follows:

$$O_t = \sigma(W_o \cdot x_t + R_o \cdot h_{t-1} + b_o) = \frac{1}{1 + e^{-(W_o \cdot x_t + R_o \cdot h_{t-1} + b_o)}} \quad (11)$$

and

$$h_t = O_t * \frac{e^{C_t} - e^{-C_t}}{e^{C_t} + e^{-C_t}}. \quad (12)$$

## 4 Experiment and results

### 4.1 Introduction of the data set

The TE chemical process is a simulation process based on the actual process flow of a chemical company. The details of the TE chemical process can be seen in prior work (Downs and Vogel 1993). Based on the simulation process, 41 measured variables denoted by  $XMEAS$  and 11 manipulated variables denoted by  $XMV$  are associated with the TE chemical process data set. Accordingly, by observing the TE chemical process, an observation vector that can reflect the production at the time  $t$  of the TE chemical process can be obtained as follows:

$$x_t = [XMEAS(1), XMEAS(2), \dots, XMEAS(41), XMV(1), \dots, XMV(11)]^T.$$

Table 7 of the Appendix shows the details of these variables. Among these variables, the sampling frequency was 20 times an hour for  $XMV(1)$ - $XMV(11)$  and  $XMEAS(1)$ - $XMEAS(22)$ , 10 times an hour for  $XMEAS(23)$ - $XMEAS(36)$ , and 4 times an hour for  $XMEAS(37)$ - $XMEAS(41)$ .

As shown in Table 8 of the Appendix, the TE chemical process included 21 pre-set faults and 1 normal state. For each fault as well as the normal state, the status of the TE chemical process was reflected in a training data set and a test data set. Each training data set contained 480 samples, and each test data set contained 960 samples. With a total of 22 system statuses, there were  $22 \times 480 = 10560$  original training samples and  $22 \times 960 = 21120$  original test samples.

### 4.2 Comparative experiments

The proposed CNN-LSTM fault diagnosis model belongs to the category of feature learning-based methods. To demonstrate the superiority of the proposed model, we conducted two sets of comparative experiments by considering the performance of the corresponding models in three aspects: *predictive accuracy (PA)*, *noise sensitivity (NS)*, and *predictive real-time (PR)*.

- (1) The first set of comparative experiments was a comparison between the proposed model with three feature extraction-based fault diagnosis models (i.e., KNN, SVM, and ANN);
- (2) and the second set was a comparison with two classical DL models, such as CNN and LSTM.

### 4.2.1 Sliding window processing for the data set

The sliding window processing ( $d = 52, \lambda = 1$ ) denoted as  $W(52, 1)$  is used to obtain the 2D samples for CNN-LSTM and CNN models from the TE process data sets.  $W(52, 1)$  is performed on each original training set and test set that can be viewed as MTS. The length of these MTS is 480 and 960 for each original training set and test set, respectively. One can easily calculate that the number of the obtained 2D samples by  $W(52, 1)$  is 428 for original training set and it is 908 for original test set. There is a total of 22 training sets and 22 test sets, thus the total number of samples for training and test are  $22 \times 428 = 9416$  and  $22 \times 908 = 19976$ . The result of  $W(52, 1)$  for each original training set and test set is shown in Table 1.

### 4.2.2 Experiment setup

Similar to our proposed model, the classical CNN model also belongs to the type of feature learning-based fault diagnosis methods. The ANN, KNN and SVM models belong to the type of feature extraction-based fault diagnosis methods. The LSTM model has no feature extraction process, and thus it does not belong to the above two types of methods, but it performs well in capturing the time delay of the occurrence of faults. Below are some settings for these models.

- The PCA method, which is a commonly used method at the feature extraction phase of fault diagnosis of the TE chemical process, was employed when using the feature extraction-based methods (i.e., KNN, SVM, and ANN) to conduct fault diagnosis of the TE chemical process.
- The number of layers of all the hierarchical models (i.e., CNN-LSTM, CNN, LSTM, and ANN) was supposed to be 6 and the parameters of the models were roughly the same to make their complexity roughly equal. The detailed settings of each layer of the four models are shown in Table 2.
- We added L2 regularization on the LSTM layers and used the dropout method in the fully connected layers to prevent over-fitting.

### 4.2.3 Evaluation index

To comprehensively evaluate the performance of each model, three indexes, namely, PA, NS, and PR are introduced in this subsection.

**Table 1** The result of sliding window processing for each training set and test set

Categories	Data set	Length of time series	The number of the obtained 2D samples
Normal or Faults 1–21	Training set	480	428
	Test set	960	908

**Definition 1** In the test phase of the model, assume that the number of samples in the test set is  $n$ , and the number of samples accurately predicted by the model is  $m$ . Then, the predictive accuracy of a model is

$$PA = \frac{m}{n} \times 100\%. \tag{13}$$

The index of PA reflects the predictive accuracy degree of the model. The larger the PA is, the higher the predictive accuracy degree is.

**Definition 2** Given an Additive white Gaussian noise (AWGN)  $N(0, \sigma^2)$  (Hughes, 1991) and is added to the test set, the index of  $PA_{noise}$  is obtained based on the test set with noise, and the index of  $PA_{none}$  is obtained based on the test set without noise. Then, the noise sensitivity of a model is.

$$NS = \frac{PA_{none} - PA_{noise}}{PA_{none}} \times 100\%. \tag{14}$$

NS is an index that reflects the adaptability of the model used in the actual production environment. The smaller the NS is, the stronger the noise immunity of the model is.

*Note:* We comply with the concept of AWGN defined in Hughes (1991). According to the definition of AWGN, the mean is usually 0 and the variance reflects the size of noise. Below we introduce how to determine the variance of noise.

First of all, we recall the notion of signal to noise ratio (SNR) as shown below,

$$SNR = 10 \log \left( \frac{P_{signal}}{P_{noise}} \right) \tag{15}$$

where  $P_{signal}$  denotes power of signal and  $P_{noise}$  denotes power of noise.

Then, we assume that SNR is 30 dB, which is a general value for sensor (Murata et al. 2020). Moreover, the average power of the normalized signal  $P_{signal}$  can be calculated by its amplitude and it is equal to 1. In this way, according to the following equation, we have  $P_{noise} \approx 0.001$ .

$$30 = 10 \log \left( \frac{1}{P_{noise}} \right) \tag{16}$$

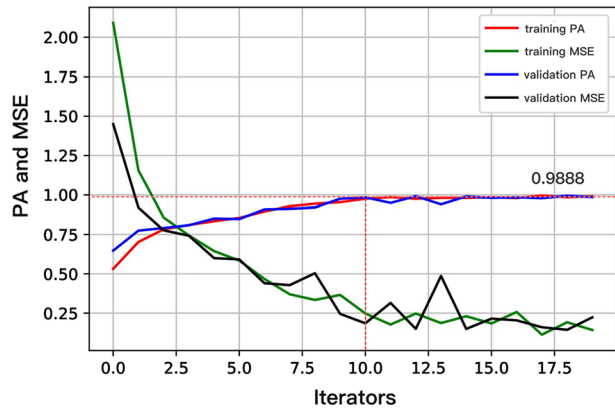
For AWGN, its variance  $\sigma^2$  is defined as  $P_{noise}$ , so we have  $\sigma^2 = P_{noise} \approx 0.001$ .

**Definition 3** Assume that there are three test sets, namely, a test set with small sample size, one with a middle sample size, and one with large sample size, and the numbers of the samples in the three sets are respectively defined as  $n_s$ ,  $n_m$ , and  $n_l$ . Moreover,  $t_s$ ,  $t_m$  and  $t_l$  respectively denote the time taken for the prediction when the three test sets are predicted by a model. Then, the predictive real-time of the model is.

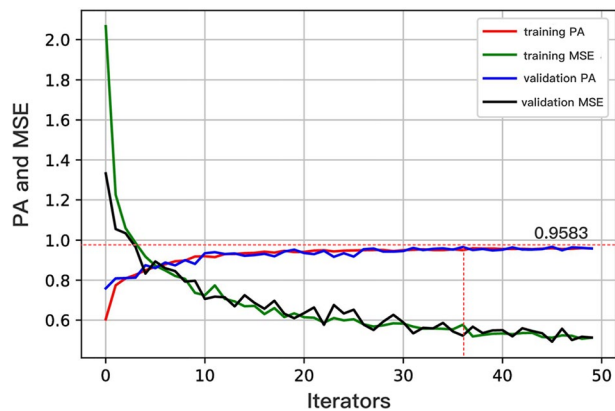
$$PR = \frac{t_s + t_m + t_l}{n_s + n_m + n_l}. \tag{17}$$

PR is an index that reflects the ability of the model to meet the timeliness of a particular scenario. The smaller the PR is, the better the prediction real-time performance of the model is.

**Fig. 6** Variation trends of *PA* and *MSE* obtained by training and validating the CNN-LSTM model



**Fig. 7** Variation trends of *PA* and *MSE* obtained by training and validating the CNN model

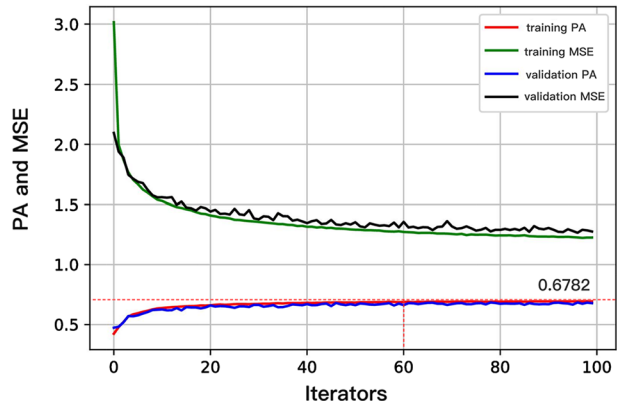


#### 4.2.4 Results and analysis

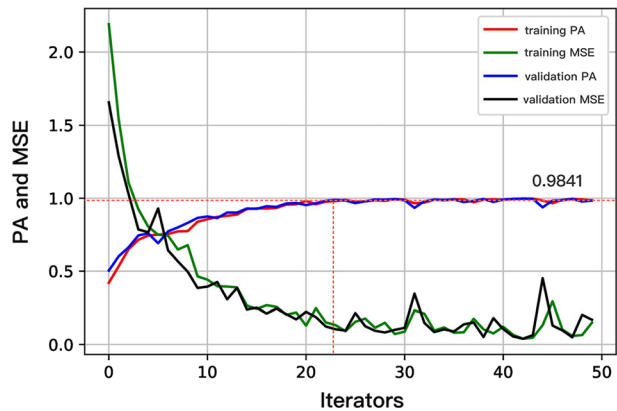
The changes of *PA* and mean square error (*MSE*) in the training and validation processes of the CNN-LSTM model, CNN model, LSTM model, and ANN model are depicted in Figs. 6, 7, 8, 9. The training *PA* and the validation *PA* increased while the training *MSE* and the validation *MSE* decreased along with the increase of the iterations, and they tended to be stable when the iterations reached or exceeded certain numbers. Specifically, the training *PA* and the validation *PA* obtained by using the CNN-LSTM model remained at 0.9888 (see Fig. 6). In the case of the CNN model, the ANN model, and the LSTM model, the values were 0.9583, 0.6783, and 0.9841, respectively (see Figs. 7, 8, 9). The results shown in Figs. 6, 8, 9 indicate that the corresponding models did not result in severe over-fitting. Besides, in terms of convergence speed, the CNN-LSTM model outperformed the other three models. The CNN-LSTM model converged after 10 iterations (see Fig. 6), while the CNN, ANN, and LSTM models converged after 36, 60, and 22 iterations, respectively (see Figs. 7, 8, 9).

Since the super-parameters of the KNN model and the SVM model are few, a grid search algorithm that considers all the possible values of super-parameters was used to select the optimal super-parameters of the KNN model and the SVM model. At each

**Fig. 8** Variation trends of PA and MSE obtained by training and validating the ANN model



**Fig. 9** Variation trends of PA and MSE obtained by training and validating the LSTM model



super-parameter selection process, the grid search algorithm uses a combination of super-parameters to evaluate the model’s performance. The optimal super-parameters of the KNN model obtained by the grid search algorithm are given below.

- The weight function used in prediction was selected as “distance”. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- The number of neighbors was selected as 1.
- The power parameter for the Minkowski metric (it only works when the weight function is set to “distance”) used to calculate distance was selected as 1.

The optimal super-parameters of the SVM model obtained by the grid search algorithm are given below.

- The kernel function used in the SVM model (it must be one of “linear”, “polynomial”, “radial basis function”, and “sigmoid”.) was selected as “radial basis function”. The details of these kernel functions is provided in Table 3.
- The penalty parameter of the error term was selected as 20.

**Table 2** Detailed settings of the CNN-LSTM model, CNN model, LSTM model, and ANN model, where bold indicates the operations defined in Sect 3.2.2 that realizes the connection between CNN and LSTM layers

Model	Name of layer	Output shape	Kernel size/Kernel number/ Stride/ Zero-padding	Number of parameters	
CNN	Layer 1	conv2d_1(Conv2D)	(None,10,52,52)	(3,3)/10/1/1	100
	Layer 2	conv2d_2(Conv2D)	(None,20,52,52)	(3,3)/20/1/1	1820
	Layer 3	conv2d_3(Conv2D)	(None,35,52,52)	(3,3)/35/1/1	6335
	Layer 4	max_ pooling2d_1(MaxPooling2)	(None,35,26,26)	–	0
		conv2d_4(Conv2D)	(None,55,26,26)	(3,3)/55/1/1	17,380
	Layer 5	max_ pooling2d_1(MaxPooling2)	(None,55,13,13)	–	0
		flatten_1(Flatten)	(None,9295)	–	0
		dense_1(Dense)	(None,100)	–	929,600
	Layer 6	dropout_1(Dropout)	(None,100)	–	0
		dense_2(Dense)	(None,22)	–	2222
LSTM	Layer 1	lstm_1(LSTM)	(None,52,100)	–	61,200
	Layer 2	lstm_2(LSTM)	(None,52,250)	–	351,000
	Layer 3	lstm_3(LSTM)	(None,52,200)	–	360,800
	Layer 4	lstm_4(LSTM)	(None,100)	–	120,400
	Layer 5	dense_1(Dense)	(None,1000)	–	101,000
	Layer 6	dropout_1(Dropout)	(None,1000)	–	0
	dense_2(Dense)	(None,22)	–	22,022	
ANN	Layer 1	dense_1(Dense)	(None,60)	–	3180
	Layer 2	dense_2(Dense)	(None,400)	–	24,400
	Layer 3	dense_3(Dense)	(None,1000)	–	401,000
	Layer 4	dense_4(Dense)	(None,400)	–	400,400
	Layer 5	dense_5(Dense)	(None,300)	–	120,300
	Layer 6	dense_6(Dense)	(None,22)	–	6622
CNN-LSTM	Layer 1	conv2d_1(Conv2D)	(None,10,52,52)	(3,3)/10/1/1	100
	Layer 2	conv2d_2(Conv2D)	(None,20,52,52)	(3,3)/20/1/1	1820
	Layer 3	<b>trans_1(Bridge)</b>	<b>(None,52,1040)</b>	–	<b>0</b>
		lstm_1(LSTM)	(None,52,100)	–	456,400
	Layer 4	lstm_2(LSTM)	(None,200)	–	240,800
	Layer 5	dense_1(Dense)	(None,1000)	–	201,000
	Layer 6	dropout_1(Dropout)	(None,1000)	–	0
	dense_2(Dense)	(None,22)	–	22,022	

**Table 3** Details of the kernel functions

Name of kernel function	Formula
Linear	$k(x, x_i) = x \cdot x_i$
Polynomial	$k(x, x_i) = (x \cdot x_i)^d$
Radial basis function	$k(x, x_i) = \exp(-  x - x_i  ^2)$
Sigmoid	$k(x, x_i) = \frac{\exp(x-x_i) - \exp(-(x-x_i))}{\exp(x-x_i) + \exp(-(x-x_i))}$



**Table 4** The *PA* of using a variety of models for noisy and noise-free data sets,  $\mu = 0, \sigma^2 = 0.001$

Index	Feature learning-based models		LSTM	Feature extraction-based models		
	CNN-LSTM	CNN		ANN	KNN	SVM
$PA_{none}$	0.9906	0.9583	0.9841	0.6782	0.7467	0.7662
$PA_{noise}$	0.9851	0.9085	0.9516	0.5376	0.6541	0.7107
$\Delta_{acc}$	0.0055	0.0498	0.0325	0.1406	0.0926	0.0555
<i>NS</i>	0.6	5.2	3.3	20.7	12.4	7.2

**Table 5** Cumulative time-consumption of prediction using different models ( $n_s = 5, n_m = 10, n_l = 100$ )

	Feature learning-based models		LSTM	Feature extraction-based models		
	CNN-LSTM	CNN		ANN	KNN	SVM
$t_s(ms)$	209	39.60	18.10	9.06	9.58	7.92
$t_m(ms)$	336	60.20	23.40	12.20	16.10	14.20
$t_l(ms)$	2670	457	143	80	162	128
<i>PR(ms)</i>	27.96	4.84	1.60	0.88	1.63	1.31

The results of *PA* are shown in Table 4. The values of  $PA_{none}$  and  $PA_{noise}$  obtained by using the feature extraction-based models were below 80%, whereas the  $PA_{none}$  and  $PA_{noise}$  obtained by using the feature learning-based models were above 95%. This indicates that the performance of the feature learning-based models was significantly better than the feature extraction-based models in terms of predictive accuracy degree.

On the other hand, compared to the values of  $PA_{none}$  and  $PA_{noise}$  obtained by using the CNN model and the LSTM model, the values obtained by using the CNN-LSTM model were larger, which indicates that the CNN-LSTM model outperforms the CNN model and the LSTM model in terms of *PA*. Theoretically, the CNN model has strong feature learning capabilities, and the LSTM model can capture time delay information. The improvement of the CNN-LSTM model may be attributed to the integration of CNN and LSTM. Furthermore, the value of  $PA_{none}$  ( $PA_{noise}$ ) dropped by 3.23% (7.66%) when only CNN was applied, and it dropped by 0.65% (3.35%) under the situation where only LSTM was used. We can see that the decrease of  $PA_{none}$  ( $PA_{noise}$ ) when only LSTM is used is smaller than the one when CNN is applied, which may result from that time delay information extracted by LSTM contributes more to fault diagnosis than feature information extracted by CNN does.

Regarding the measurement of *NS*, the feature learning-based models generally outperformed the feature extraction-based models. This shows that the former kind of models is superior to the latter one in terms of adaptability. The value of *NS* associated with the CNN-LSTM model was lower than the ones associated with the CNN and LSTM models, which indicates that the integration of feature information and time delay information is beneficial to the model against noise.

Tables 5 and 6 present the computing results of *PR* under the situations where different models were applied for fault diagnosis of the TE chemical process. One can see from Table 5 that the feature learning-based models showed worse performance than the feature extraction-based models in terms of *PR*. This is mainly due to the relatively high

**Table 6** Average time-consumption of prediction using different models ( $n_s = 5, n_m = 10, n_l = 100$ )

	Feature learning-based models		LSTM	Feature extraction-based models		
	CNN-LSTM	CNN		ANN	KNN	SVM
$t_s/n_s(ms)$	41.80	7.92	3.62	1.81	1.92	1.58
$t_m/n_m(ms)$	33.6	6.02	2.34	1.22	1.61	1.42
$t_s/n_s(ms)$	26.7	4.75	1.43	0.8	1.62	1.28

computational cost of CNN for feature learning. One can also find that the CNN-LSTM model performed worse in terms of *PR* compared with the CNN and LSTM models. This is mainly because the CNN-LSTM model has a bridge that is used to combine the CNN layer and the LSTM layer and the bridge greatly increases the computational complexity of the CNN-LSTM model.

The number in Table 6 indicates how long it took for a model to predict a single sample. Interestingly, the average sample prediction time of the DL-based models (i.e., CNN-LSTM model, CNN model, and LSTM model) decreased along with the increasing of the sample size. When the sample size reaches a relatively large scale, whether the performance of *PR* of the DL-based models will be superior to the feature extraction-based models is a question that remains to be further studied.

The experimental results show that the proposed CNN-LSTM model had a good performance in terms of *PA* and *NS*. However, there is a trade-off relationship between *PA*, *NS*, and *PR*. A better combination of *PA* and *NS* is usually accompanied by a worse *PR*, which is results from the high complexity of a model. Therefore, the bad performance of the CNN-LSTM model in terms of *PR* is inevitable compared to the other five models. The proposed CNN-LSTM model is an effective tool for solving the fault diagnosis problems where the requirements for *PA* and *NS* are high, but the requirements for *PR* performance are very low.

### 4.3 Fault modes analysis

We visualize the output of each layer (Conv2d\_1, Conv2d\_2, Lstm\_2, Dense\_1 and Dense\_2) of the proposed CNN-LSTM model to observe the fault modes learned by the model. Firstly, we select two different samples with the same fault from the test set. Then, we use the trained CNN-LSTM model to predict the two samples. Finally, the outputs of each layer of the model are visualized as shown in Fig. 9.

One can summarize the following findings from Fig. 10:

- (1) The outputs of Conv2d\_1 layer and Conv2d\_2 layer indicate that some attributes have been extracted as shown in the red box. Moreover, the extracted attributes are the same

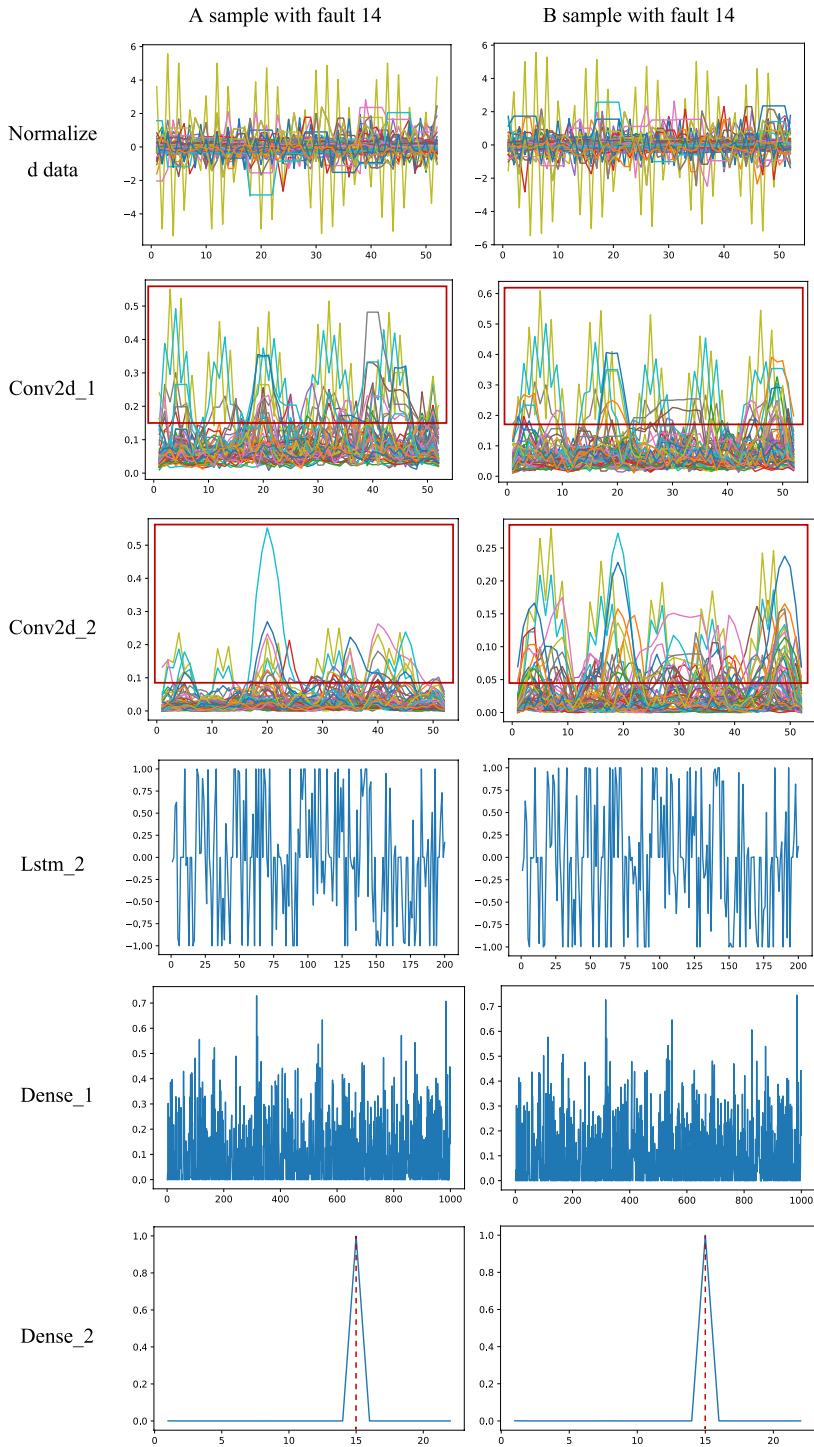


Fig. 10 The visualizations of fault diagnosis for two samples with the same fault

for these two different samples, which may indicate that those extracted attributes highly correlated with fault mode.

- (2) The outputs of Lstm\_2 layer model the hidden state of the last LSTM unit that can be regarded as the extracted time feature. We can find that the extracted time features of the two different samples are almost identical, which may indicate that the extracted time feature highly correlated with fault model.
- (3) The outputs of Dense\_1 integrate the extracted attributes information and the extracted time feature, which can be regarded as an abstract fault mode identified by the CNN-LSTM model.
- (4) The abscissa corresponding to the peak of the visualization curve of outputs of Dense\_2 is regarded as the result of fault diagnosis. We can see that the fault diagnosis results of the two different samples are consistent with their labels, which verifies the validity of the identified fault mode.

Figure 11 shows the visualizations of fault diagnosis for two samples with different faults. One can find that the extracted attributes information, time feature and fault modes are different for the two samples with different faults, and the fault diagnosis results of the two different samples are consistent with their labels, which further verify the validity of the identified fault modes.

## 5 Conclusions and future works

In this paper, we propose a novel DL-based method for the fault diagnosis of complex systems by synthetically considering feature extraction and time delay of the occurrence of faults simultaneously. The proposed fault diagnosis method consists of two parts, namely, sliding window processing and CNN-LSTM model. Samples for the development of CNN-LSTM model can be obtained by sliding window processing integrating the feature information and time delay information of MTS. The developed CNN-LSTM model is a combination of CNN layers and LSTM layers. Automatic feature learning can be performed through the CNN layers. Time delay information can be captured through the LSTM layers. Through addressing the fault diagnosis of the TE chemical process, it is verified that the predictive accuracy and noise sensitivity of fault diagnosis can be greatly improved when the proposed method is applied. Comparisons with several existing fault diagnosis methods show the superiority of the proposed method.

The following aspects are worthy of future research:

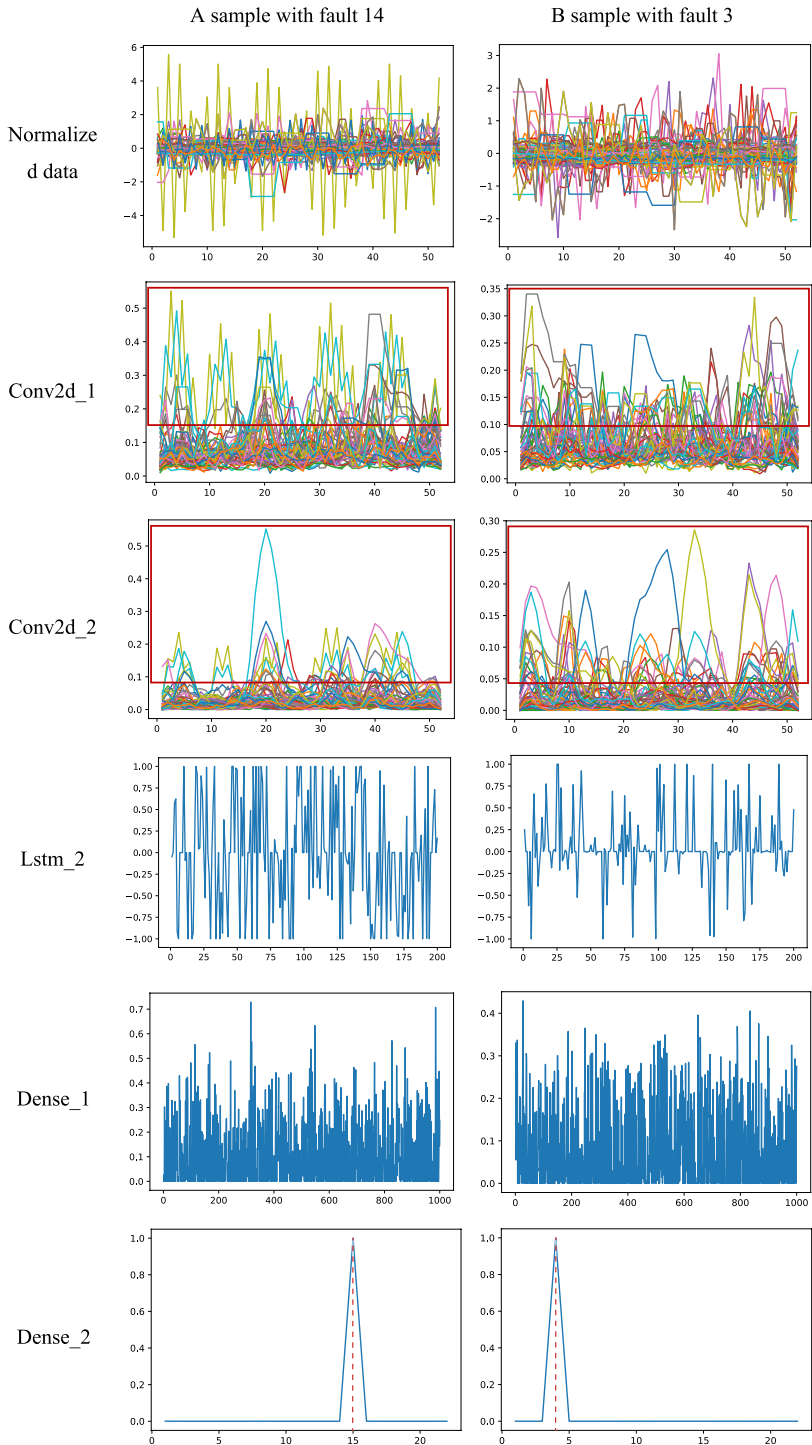


Fig. 11 The visualizations of fault diagnosis for two samples with different faults

- (1) The fault diagnosis of TE chemical process is considered as an application of the proposed method. To further verify the generalization of the proposed method, using it to solve fault diagnosis of such complex systems as wind turbine blades (Du et al. 2020), rotor (Nath et al. 2020), gearbox (Wu et al. 2019), subsea pipelines (Cai et al. 2020) and so on, deserves further study.
- (2) DL is a kind of “black box” approach with little interpretability, which weakens the reliability of fault diagnosis that is a critical goal for fault diagnosis. On the other hand, such traditional fault diagnosis methods as physics of failure and fault tree analysis are reliable. How to integrate these reliable methods into DL to improve the interpretability of DL-based fault diagnosis methods is worthy of further research.
- (3) The fault data collected from real industrial scene is rare owing to that the system is not allowed to run for a long time under fault conditions. It is a big challenge for DL when the collected data is insufficient. It may be a good attempt to integrate knowledge of complex systems into DL models for solving this problem (Feng et al. 2021; Li et al. 2019b; Yu and Liu 2020). Therefore, how to integrate system knowledge into the DL-based fault diagnosis model is worthy of in-depth study.

## Appendix

See Tables 7 and 8.

**Table 7** Variables involved in the TE chemical process and their description

Variable	Description	Unit
<i>XMV</i> (1)	D Feed Flow (stream 2) (Corrected Order)	kg/hr
<i>XMV</i> (2)	E Feed Flow (stream 3) (Corrected Order)	kg/hr
<i>XMV</i> (3)	A Feed Flow (stream 1) (Corrected Order)	kscmh
<i>XMV</i> (4)	A and C Feed Flow (stream 4)	kscmh
<i>XMV</i> (5)	Compressor Recycle Valve	%
<i>XMV</i> (6)	Purge Valve (stream 9)	%
<i>XMV</i> (7)	Separator Pot Liquid Flow (stream 10)	m <sup>3</sup> /hr
<i>XMV</i> (8)	Stripper Liquid Product Flow (stream 11)	m <sup>3</sup> /hr
<i>XMV</i> (9)	Stripper Steam Valve	%
<i>XMV</i> (10)	Reactor Cooling Water Flow	m <sup>3</sup> /hr
<i>XMV</i> (11)	Condenser Cooling Water Flow	m <sup>3</sup> /hr
<i>XMEAS</i> (1)	A Feed (stream 1)	kscmh
<i>XMEAS</i> (2)	D Feed (stream 2)	kg/hr
<i>XMEAS</i> (3)	E Feed (stream 3)	kg/hr
<i>XMEAS</i> (4)	A and C Feed (stream 4)	kscmh
<i>XMEAS</i> (5)	Recycle Flow (stream 8)	kscmh
<i>XMEAS</i> (6)	Reactor Feed Rate (stream 6)	kscmh
<i>XMEAS</i> (7)	Reactor Pressure	kPa gauge
<i>XMEAS</i> (8)	Reactor Level	%
<i>XMEAS</i> (9)	Reactor Temperature	Deg C
<i>XMEAS</i> (10)	Purge Rate (stream 9)	kscmh
<i>XMEAS</i> (11)	Product Sep Temp	Deg C
<i>XMEAS</i> (12)	Product Sep Level	%
<i>XMEAS</i> (13)	Prod Sep Pressure	kPa gauge
<i>XMEAS</i> (14)	Prod Sep Underflow (stream 10)	m <sup>3</sup> /hr
<i>XMEAS</i> (15)	Stripper Level	%
<i>XMEAS</i> (16)	Stripper Pressure	kPa gauge
<i>XMEAS</i> (17)	Stripper Underflow (stream 11)	m <sup>3</sup> /hr
<i>XMEAS</i> (18)	Stripper Temperature	Deg C
<i>XMEAS</i> (19)	Stripper Steam Flow	kg/hr
<i>XMEAS</i> (20)	Compressor Work	kW
<i>XMEAS</i> (21)	Reactor Cooling Water Outlet Temp	Deg C
<i>XMEAS</i> (22)	Separator Cooling Water Outlet Temp	Deg C
<i>XMEAS</i> (23)	Component A in stream 6	mole %
<i>XMEAS</i> (24)	Component B in stream 6	mole %
<i>XMEAS</i> (25)	Component C in stream 6	mole %
<i>XMEAS</i> (26)	Component D in stream 6	mole %
<i>XMEAS</i> (27)	Component E in stream 6	mole %
<i>XMEAS</i> (28)	Component F in stream 6	mole %
<i>XMEAS</i> (29)	Component A in stream 9	mole %
<i>XMEAS</i> (30)	Component B in stream 9	mole %
<i>XMEAS</i> (31)	Component C in stream 9	mole %
<i>XMEAS</i> (32)	Component D in stream 9	mole %
<i>XMEAS</i> (33)	Component E in stream 9	mole %
<i>XMEAS</i> (34)	Component F in stream 9	mole %
<i>XMEAS</i> (35)	Component G in stream 9	mole %

**Table 7** (continued)

Variable	Description	Unit
<i>XMEAS</i> (36)	Component H in stream 9	mole %
<i>XMEAS</i> (37)	Component D in stream 11	mole %
<i>XMEAS</i> (38)	Component E in stream 11	mole %
<i>XMEAS</i> (39)	Component F in stream 11	mole %
<i>XMEAS</i> (40)	Component G in stream 11	mole %
<i>XMEAS</i> (41)	Component H in stream 11	mole %

**Table 8** Details of faults of the TE chemical process

No	Description	Fault type
1	A/C feed flow ratio changes, component B content remains the same (flow 4)	Step
2	The content of component B changes, and the A/C feed flow ratio does not change (flow 4)	Step
3	The temperature of material D changes (flow 2)	Step
4	Reactor cooling water inlet temperature changes	Step
5	Condenser cooling water inlet temperature changes	Step
6	Material A loss (flow 1)	Step
7	Material C pressure loss (flow 4)	Step
8	The composition of materials A, B, and C changes (flow 4)	Random variables
9	The temperature of material D changes (flow 2)	Random variables
10	The temperature of material C changes (flow 2)	Random variables
11	Reactor cooling water inlet temperature changes	Random variables
12	Condenser cooling water inlet temperature changes	Random variables
13	Random Variables	Slow drift
14	Reactor cooling water valve	stick to
15	Condenser cooling water valve	stick to
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown
21	Flow 4 valve is fixed in steady state position	Constant position

**Acknowledgements** This research was supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 71521001), and the National Natural Science Foundation of China (Nos. 71690230, 71690235, 71501056, 71601066, 71901086, 71501055, 71571060, 71501054 and 71571166).



## References

- Aydin I, Karakose M, Akin E (2012) An adaptive artificial immune system for fault classification. *J Intell Manuf* 23(5):1489–1499
- Cai B, Liu H, Xie M (2016) A real-time fault diagnosis methodology of complex systems using object-oriented Bayesian networks. *Mech Syst Signal Process* 80:31–44
- Cai B, Liu Y, Xie M (2017a) A dynamic-bayesian-network-based fault diagnosis methodology considering transient and intermittent faults. *IEEE Trans Autom Sci Eng* 14(1):276–285
- Cai B, Zhao Y, Liu H, Xie M (2017b) A data-driven fault diagnosis methodology in three-phase inverters for pmsm drive systems. *IEEE Trans Power Electron* 32(7):5590–5600
- Cai B, Shao X, Liu Y, Kong X, Wang H, Xu H, Ge W (2020) Remaining useful life estimation of structure systems under the influence of multiple causes: subsea pipelines as a case study. *IEEE Trans Industr Electron* 67(7):5737–5747
- Casimir R, Boutleux E, Clerc G, Yahoui A (2006) The use of features selection and nearest neighbors rule for faults diagnostic in induction motors. *Eng Appl Artif Intell* 19(2):169–177
- de Bruin, T., Verbert, K., & Babuška, R (2017) Railway track circuit fault diagnosis using recurrent neural networks. *IEEE Trans Neural Netw Learn Syst* 28(3), 523–533
- Downs JJ, Vogel EF (1993) A plant-wide industrial process control problem. *Comput Chem Eng* 17(3):245–255
- Du Y, Zhou S, Jing X, Peng Y, Wu H, Kwok N (2020) Damage detection techniques for wind turbine blades: a review. *Mech Syst Signal Process* 141:106445
- El-Koujok M, Benammar M, Meskin N, Al-Naemi M, Langari R (2014) Multiple sensor fault diagnosis by evolving data-driven approach. *Inf Sci* 259:346–358
- Feng J, Yao Y, Lu S, Liu Y (2021) Domain knowledge-based deep-broad learning framework for fault diagnosis. *IEEE Trans Industr Electron* 68(4):3454–3464
- Gao X, Hou J (2016) An improved SVM integrated GS-PCA fault diagnosis approach of tennessee eastman process. *Neurocomputing* 174:906–911
- Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with LSTM recurrent networks. *J Mach Learn Res* 3:115–143
- Goyal D, Choudhary A, Pabla BS, Dhama SS (2020) Support vector machines based non-contact fault diagnosis system for bearings. *J Intell Manuf* 31:1275–1289
- Guo X, Chen L, Shen C (2016) Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* 93:490–502
- Hong L, Dhupia JS (2014) A time domain approach to diagnose gearbox fault based on measured vibration signals. *J Sound Vib* 333(7):2164–2180
- Hughes B (1991) On the error probability of signals in additive white Gaussian noise. *IEEE Trans Inf Theory* 37(1):151–155
- Janssens O, Slavkovikj V, Vervisch B, Stockman K, Loccufer M, Verstockt S, Van de Walle R, Van Hoescke S (2016) Convolutional neural network based fault detection for rotating machinery. *J Sound Vib* 377:331–345
- Jing C, Hou J (2015) SVM and PCA based fault classification approaches for complicated industrial process. *Neurocomputing* 167:636–642
- Kabir S (2017) An overview of fault tree analysis and its application in model based dependability analysis. *Expert Syst Appl* 77:114–135
- Khalil K, Eldash O, Kumar A, Bayoumi M (2020) Machine Learning-Based Approach for Hardware Faults Prediction. *Regular Papers, IEEE Transactions on Circuits and Systems I*, pp 1–13
- Kolen JF, Kremer SC (2009) Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. *Wiley-IEEE Press, In A Field Guide to Dynamical Recurrent Networks*
- Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90
- Lei Y, Zuo MJ (2009) Gear crack level identification based on weighted K nearest neighbor classification algorithm. *Mech Syst Signal Process* 23(5):1535–1547
- Lei Y, Jia F, Lin J, Xing S, Ding S (2016) An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Trans Industr Electron* 63(5):3137–3147
- Li C (2018) Improving forecasting accuracy of daily enterprise electricity consumption using a random forest based on ensemble empirical mode decomposition. *Energy* 165:1220–1227
- Li C, Cerrada M, Cabrera D, Sanchez RV, Pacheco F, Ulutagay G, Valente de Oliveira J (2018a) A comparison of fuzzy clustering algorithms for bearing fault diagnosis. *Journal of Intelligent and Fuzzy Systems* 34(6):3565–3580

- Li H, Huang H, Li Y, Zhou J, Mi J (2018b) Physics of failure-based reliability prediction of turbine blades using multi-source information fusion. *Appl Soft Comput* 72:624–635
- Li C, de Oliveira JV, Cerrada M, Cabrera D, Sanchez RV, Zurita G (2019a) A systematic review of fuzzy formalisms for bearing fault diagnosis. *IEEE Trans Fuzzy Syst* 27(7):1362–1382
- Li X, Zhang W, Ding Q (2019b) Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Process* 161:136–154
- Liu H, Zhou J, Zheng Y, Jiang W, Zhang Y (2018) Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Trans* 77:167–178
- Liu C, Hsiao WH, Tu Y (2019) Time series classification with multivariate convolutional neural network. *IEEE Trans Industr Electron* 66(6):4788–4797
- Murata, M., Kuroda, R., Fujihara, Y., Otsuka, Y., Shibata, H., Shibaguchi, T., et al. (2020) A high near-infrared sensitivity over 70-dB SNR CMOS image sensor with lateral overflow integration trench capacitor. *IEEE Trans Electron Devices* 67(4), 1653–1659
- Nath AG, Udmale SS, Singh SK (2020) Role of artificial intelligence in rotor fault diagnosis: a comprehensive review. *Artif Intell Rev*. <https://doi.org/10.1007/s10462-020-09910-w>
- Rai VK, Mohanty AR (2007) Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert-Huang transform. *Mech Syst Signal Process* 21(6):2607–2615
- Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 512–519
- Rodríguez Ramos A, Domínguez Acosta C, Rivera Torres PJ, Serrano Mercado EI, Beauchamp Baez G, Rifón LA, Llanes-Santiago O (2019) An approach to multiple fault diagnosis using fuzzy logic. *J Intell Manuf* 30(1):429–439
- Seera M, Lim CP, Loo CK (2016) Motor fault detection and diagnosis using a hybrid FMM-CART model with online learning. *J Intell Manuf* 27(6):1273–1285
- Serdio F, Lughofer E, Pichler K, Pichler M, Buchegger T, Efendic H (2015) Fuzzy fault isolation using gradient information and quality criteria from system identification models. *Inf Sci* 316:18–39
- Sivaraman M, Strojwas AJ (2001) Path delay fault diagnosis and coverage-a metric and an estimation technique. *IEEE Trans Comput Aided Des Integr Circuits Syst* 20(3):440–457
- Wang, Y., Pan, Z., Yuan, X., Yang, C., & Gui, W (2020) A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Transactions* 96, 457–467
- Wang Z, Marek-Sadowska MM, Tsai KH, Rajski J (2005) Delay-fault diagnosis using timing information. *IEEE Trans Comput Aided Des Integr Circuits Syst* 24(9):1315–1325
- Wen L, Li X, Gao L, Zhang Y (2018) A New Convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans Industr Electron* 65(7):5990–5998
- Wu H, Zhao J (2018) Deep convolutional neural network model based chemical process fault diagnosis. *Comput Chem Eng* 115:185–197
- Wu Q, Guo Y, Chen H, Qiang X, Wang W (2019) Establishment of a deep learning network based on feature extraction and its application in gearbox fault diagnosis. *Artif Intell Rev* 52(1):125–149
- Yan R, Gao R, Chen X (2014) Wavelets for fault diagnosis of rotary machines: a review with applications. *Signal Process* 96:1–15
- Yang L, Agyakwa PA, Johnson CM (2013) Physics-of-failure lifetime prediction models for wire bond interconnects in power electronic modules. *IEEE Trans Device Mater Reliab* 13(1):9–17
- Yu J, Liu G (2020) Knowledge extraction and insertion to deep belief network for gearbox fault diagnosis. *Knowl-Based Syst* 197:105883
- Zhang W, Peng G, Li C, Chen Y, Zhang Z (2017) A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 17(2):425
- Zhang C, Guo Q, Li Y (2020) Fault detection in the tennessee eastman benchmark process using principal component difference based on k-nearest neighbors. *IEEE Access* 8:49999–50009
- Zhu S, Huang H, Peng W, Wang H, Mahadevan S (2016) Probabilistic physics of failure-based framework for fatigue life prediction of aircraft gas turbine discs under uncertainty. *Reliab Eng Syst Saf* 146:1–12

## Authors and Affiliations

Ting Huang<sup>1,2,3</sup> · Qiang Zhang<sup>1,2,3</sup>  · Xiaonan Tang<sup>1,2,3</sup> · Shuangyao Zhao<sup>1,2,3</sup> · Xiaonong Lu<sup>1,2,3</sup>

Ting Huang  
huangting@mail.hfut.edu.cn

Shuangyao Zhao  
zsyjiu91@163.com

Xiaonong Lu  
xnlu@hfut.edu.cn

<sup>1</sup> School of Management, Hefei University of Technology, Box 270, Hefei 230009, Anhui, People's Republic of China

<sup>2</sup> Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Box 270, Hefei 230009, Anhui, People's Republic of China

<sup>3</sup> Ministry of Education Engineering Research Center for Intelligent Decision-Making and Information System Technologies, Hefei 230009, Anhui, People's Republic of China