Research Article

A Multichannel LSTM-CNN Method for Fault Diagnosis of Chemical Process

Bilin Shao,¹ Xiaoli Hu ,¹ Genqing Bian ,² and Yu Zhao¹

¹School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China

²School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China

Correspondence should be addressed to Xiaoli Hu; 1305238001@qq.com

Received 19 July 2019; Revised 19 September 2019; Accepted 9 October 2019; Published 5 December 2019

Academic Editor: Juan P. Amezcua-Sanchez

Copyright © 2019 Bilin Shao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The identification and classification of faults in chemical processes can provide decision basis for equipment maintenance personnel to ensure the safe operation of the production process. In this paper, we combine long short-term memory neural network (LSTM) with convolutional neural network (CNN) and propose a new fault diagnosis method based on multichannel LSTM-CNN (MCLSTM-CNN). The primary methodology here includes three aspects. In the initial state, the fault data are input into the LSTM to obtain the output of the hidden layer, which stores the relevant temporal and spatial domain information. Due to the diversity of data features, convolutional kernels with different sizes are utilized to form multiple channels to extract the output characteristics of the hidden layer simultaneously. Finally, the fault data are classified by fully connected layers. The Tennessee Eastman (TE) chemical process is used for experimental analysis, and the MCLSTM-CNN model is compared with the LSTM-CNN, LSTM, CNN, RF and KPCA + SVM models. The experimental results show that the MCLSTM-CNN model has higher diagnostic accuracy, and the fault classification results are superior to other models.

1. Introduction

With the continuous improvement of science and technology, great changes have taken place in chemical processes. Many industries, such as chemical, metallurgy, and electric power, are becoming more intelligent and automated. As a consequence, chemical processes become more and more large-scale and complicated. The fault of equipment or module may affect the safety and reliability of the entire production process. However, if the fault cannot be detected in time, it may lead to chemical accidents, resulting in severe property loss, casualties, environmental pollution, and other hazards. In this case, the fault diagnosis of chemical processes is essential, for which timely and accurate fault identification and effective measures can ensure operation safety during the production process.

At present, scholars have conducted a series of research studies on fault diagnosis of chemical process, including knowledge-based, model-based, and data-driven methods.

Among them, knowledge-based and model-based methods are limited in application scope due to the difficulty in modeling and strong dependence on expert experience. Therefore, the data-driven method has become a hot research topic. This method mainly uses statistical analysis, machine learning, and deep learning to build a fault diagnosis model. Common statistical methods include principal component analysis (PCA) [1, 2], partial least squares (PLS) [3, 4], independent component analysis (ICA) [5, 6], and linear discriminant analysis (LDA) [7, 8]. However, they have great limitations in dealing with nonlinear dynamic and non-Gaussian process. Deng introduced the kernel function into PCA and proposed the fault diagnosis method of KPCA. It can transform variables from nonlinear input space to linear feature space and solve the nonlinear problem of chemical process data [9]. To solve the dynamic problem of non-Gaussian process, Stefatos proposed a dynamic ICA method [10], connecting the current data matrix with the previous data matrix to form the enhanced data matrix. DPCA is a method similar to the dynamic ICA method [11].

Although it can reduce the dimensionality of high-dimensional data and perform well in fault detection, the separation effect on faults is poor. In order to effectively separate faults, machine learning methods such as the support vector machine (SVM) [12, 13], K-nearest neighbor (KNN) [14, 15], Bayes [16, 17], and decision tree (DT) [18, 19] are used for fault diagnosis. Zhang used KPCA to reduce the data dimension and then classified fault data by using the SVM [20]. Random forest (RF) is composed of multiple decision trees. Zhang et al. used the RF for fault diagnosis of rolling bearings and compared it with the diagnosis results of traditional classifiers (SVM, KNN, ANN, and DT) and found that the accuracy of the RF was the highest [21]. Besides, there are various combinatorial methods such as ICA-PCA [22], KICA-SVM [23], and PCA-ANN [24]. Although these methods have their advantages, they have limited expression in some complex tasks and are prone to dimensional disasters.

In the past few years, deep learning [25, 26] has developed rapidly and has performed well in many fields. Deep learning has a deep network structure, which can transform raw data into more abstract and higher-level data expression through the combination of a series of nonlinear functions, so it can effectively process complex process data. Deep learning has made some research achievements in the field of fault diagnosis. Sun et al. extracted the data features using a sparse self-encoder and then trained the network to complete the fault classification [27]. In view of the traditional fault diagnosis, methods are mostly studied from the spatial domain, and the temporal domain features have been hardly studied; Zhang and Zhao proposed a fault diagnosis model of an extensible deep belief network (DBN). DBN subnet extracts spatial and temporal domain features of data and then classifies faults. The average diagnostic accuracy of TE is 82.1% [28]. Zhang et al. utilized a variational auto-encoder (VAE) to learn the higher-level abstract features of the original data and then input them into DBN to diagnose the fault [29]. Considering that the fault data are time series, Zhao et al. presented a long short-term memory neural network (LSTM) method which can adaptively learn the dynamic information of the original data, and the results show that the model has good fault diagnosis performance [30]. Wu and Zhao applied convolutional neural network to fault diagnosis of chemical processes, which can effectively extract fault features and classify them [31].

There are abundant research achievements on fault diagnosis of chemical process. However, the accuracy of fault diagnosis is still not high enough, and there is a certain gap from the practical application, so further research is needed. This paper proposes a fault diagnosis method for MCLSTM-CNN. By constructing a multichannel deep network model, we can effectively extract features of data in the temporal and spatial domains, thus achieving classification diagnosis of fault data. This model was applied to the TE dataset, and the classification effects of LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM models were compared. The results reveal that this model has higher diagnostic accuracy and good reliability.

2. Relevant Theory

2.1. LSTM. The recurrent neural network (RNN) is a network structure used to process sequence data. Its characteristic is that the current output depends not only on the current input information but also on the previous output information. Although this network has certain advantages, when the information that the RNN needs to learn is far away from the current predicted value, the gradient disappearance or gradient explosion phenomenon is prone to occur. LSTM is a variant of the RNN. It can realize the self-circulating weight change through a structure called gate unit, which can effectively alleviate the problem of gradient disappearance and gradient explosion. Therefore, it is more suitable for processing time series data. The basic structure of LSTM is shown in Figure 1.

Figure 1 presents the LSTM architecture, which contains four neural network layers and interacts in a special structure. It selects information mainly through three gate structures: input gate, forget gate, and output gate. Suppose the shape of the input data is $\{X_1, X_2, X_3, \dots, X_m\}$. Taking the t^{th} sample as an example, X_t is the input data at the current moment. To select valid data information, we need to discard the data of the cell state selectively. The forget gate controls the current retain of the amount of previous cell state. The formula is expressed as follows:

$$f_t = \sigma(W_f h_{t-1} + W_f X_t + b_f), \quad (1)$$

where f_t is the forget value, h_{t-1} is the data output at the previous moment, W and b are the corresponding weight coefficient matrix and the offset term, and σ is the sigmoid activation function.

The implementation process of the input gate mainly includes three parts. Initially, the sigmoid function is used to determine the data to be updated; then, the tanh function is used to generate candidate value vectors; eventually, the cell state updates. The calculation is expressed in the following equations:

$$\begin{aligned} i_t &= \sigma(W_i h_{t-1} + W_i X_t + b_i), \\ C'_t &= \tanh(W_c h_{t-1} + W_c X_t + b_c), \\ C_t &= f_t * C_{t-1} + i_t * C'_t, \end{aligned} \quad (2)$$

where i_t is the information to be updated, C'_t is the alternative content to be updated, $\tanh$ is the hyperbolic tangent activation function, C_{t-1} is the old cell state, and C_t is the updated cell state.

The output gate determines the output value. The sigmoid function is used to determine the output portion of the cell state, and then the cell state is processed by the tanh activation function. Finally, the output value is obtained by multiplication. The formulas are expressed by

$$\begin{aligned} O_t &= \sigma(W_o h_{t-1} + W_o X_t + b_o), \\ h_t &= O_t * \tanh(C_t), \end{aligned} \quad (3)$$

FIGURE 1: Structure of LSTM.

where O_t is the information to be output and h_t represents the output value at time t .

2.2. CNN. Convolutional neural network (CNN) is a deep neural network, which can effectively extract data features and achieve great success in image recognition and detection. It mainly consists of a convolutional layer, pooling layer, and fully connected layer.

2.2.1. Convolutional Layer. The convolutional layer, which is the core of the convolutional neural network, mainly convolves the input features through a set of weighted convolutional kernels. Each convolutional kernel can only extract one feature, and the information obtained is limited. Therefore, it is necessary to use multiple convolutional kernels for calculation, and the weights and parameters between the input features and each convolutional kernel are shared. Convolutional layers have two main benefits: firstly, for some input features, users need not perceive all of their information, and only need to extract some key features, which makes it more efficient to deal with problems; secondly, the convolutional operation can greatly reduce the number of parameters, thereby reducing the amount of calculation and improving the calculation efficiency. Figure 2 shows a typical convolutional operation.

The process of convolutional operation includes the following steps:

Step 1: the data are processed into a two-dimensional matrix. The specific operation is to reorganize the data into a two-dimensional matrix by calling the reshape function to perform dimensional transformation.

Step 2: the convolutional kernel and input data are used to perform the operation of matrix inner product, and then a bias is added to calculate an eigenvalue. The aforementioned calculation steps are repeated, moving the convolutional kernel from left to right and from top to bottom on the input data to calculate the next value. The distance to slide once is called step size. Taking Figure 2 as an example, the input feature size is 5×5 while the convolutional kernel is 3×3 with the step size

of 1. The first element of the output feature can be calculated by $1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 0 + 1 \times 1 + 0 = 4$. The calculation of the second output element after sliding step size 1 is similar to the first element. Therefore, we can use equation (4) to calculate the output features of the l^{th} layer:

$$x_j^l = f \left(\sum_{i=1, \dots, M} x_i^{l-1} * k_{ij}^l + b_j^l \right), \quad j = 1, \dots, N, \quad (4)$$

where x_i^{l-1} represents the i^{th} input feature, k_{ij}^l are the corresponding weights of i^{th} input feature and convolutional kernel j , the symbol $*$ represents the convolutional operation, b_j^l is the bias of j^{th} convolutional kernel, x_j^l is the j^{th} output feature, and f is the activation function.

Step 3: the number of convolutional kernels are set to obtain multiple features.

2.2.2. Pooling Layer. The pooling layer, also known as the sampling layer, is usually connected to the convolutional layer to pool the output of the convolutional layer. There are two types of pooling operations: max pooling and average pooling. The max pooling operation is used to calculate the maximum value of each pooling area as an output, and the average pooling operation is used to calculate the average value of each pooling area as an output. After the convolutional operation, the dimension of the data will increase. Meanwhile, the pooling operation can reduce the feature dimension, simplify the calculation, and prevent overfitting. In addition, the pooling layer can integrate similar chemical features, making the test results more reliable. Figure 3 shows the calculation process of the pooling layer, where the input feature has a size of 4×4 , a kernel size of 2×2 , and a step size of 2.

2.2.3. Dropout Layer. The dropout layer is a mechanism used in deep learning to prevent overfitting. During the training process, the dropout layer makes the neuron stop working according to a certain probability, namely, setting its output value to 0. This allows neurons to work with other randomly

FIGURE 2: Convolutional operation.

FIGURE 3: Pooling operation.

selected neurons, thereby reducing the complex coadaptation between neurons. Also, the dropout layer avoids the situation that some features only work when influenced by other specific features. It forces the network to learn more robust features and improve the generalization ability of the model.

2.2.4. Fully Connected Layer. The fully connected (FC) layer can map the feature representations learned by the convolutional layer and pooling layer to the sample mark space so as to realize data classification. The basic structure is shown in Figure 4. Each neuron in the FC layer is connected with all neurons in the previous layer to integrate the extracted features. Finally, the activation function is used to nonlinearly transform the input so that the network has a nonlinear learning ability. Assuming that the number of input neurons is P and the number of output neurons is Q , the output value can be calculated by the following equation:

$$x_j^l = f\left(\sum_{i=1, \dots, P} x_i^{l-1} \times w_{ij}^l + b_j^l\right), \quad j = 1, \dots, Q, \quad (5)$$

where x_j^l is the output value of the j^{th} neuron, x_i^{l-1} represents the i^{th} input neuron, and w_{ij}^l are the corresponding weights of i^{th} input neuron and j^{th} output neuron, b_j^l represents the bias of the j^{th} output value, and f represents the activation function.

It can be seen from Figure 4 that the input of the FC layer must be a one-dimensional array. However, the data

processed by the convolutional layer are a multidimensional array, so they cannot be directly input into the FC layer for classification. The flatten layer can flatten the data, that is, processing the multidimensional input into one dimension to meet the input requirements of the FC layer.

3. Fault Diagnosis Method Based on MCLSTM-CNN

Feature extraction has a great influence on the final classification results of faults. How to extract more effective features becomes an important factor in improving the accuracy of diagnosis. LSTM can effectively deal with the long-term dependence of time series, and CNN has a strong ability to extract features from multidimensional data. In view of the time-varying and high-dimensional characteristics of chemical process data, this paper uses LSTM to process the fault data, and the hidden layer output obtained contains the spatial and temporal information of the original data. However, chemical process data have abundant spatial features, and it is difficult to express the spatial features accurately by LSTM alone. To this end, multiple parallel convolutional layers are used to extract the output features of the hidden layer simultaneously. Generally, the convolutional kernel size of each convolutional layer is the same, so the generated features are relatively single. However, the convolutional kernels of parallel convolutional layers have different sizes, and the extracted feature types are different, thus increasing the diversity of output features of the hidden layer. Figure 5 shows the method framework, which mainly includes two stages: offline modeling and online monitoring.

3.1. Offline Modeling Stage

Step 1: obtain fault data of chemical process and divide the training set and testing set

Step 2: preprocess data including standardizing data, converting data into a two-dimensional matrix, and encoding labels

FIGURE 4: Fully connected layer.

FIGURE 5: Fault diagnosis framework based on MCLSTM-CNN.

Step 3: define the network structure, including the number of nodes and layers of the network layer, the size of the convolutional kernel, the dimension of the input data, and the activation function

Step 4: set training parameters, such as epoch, batch_size, and optimizer

Step 5: input the training set into the MCLSTM-CNN model for training and adjust the network weight and bias continually

Step 6: determine whether the loss value is less than the threshold; if it is less than the threshold, the training is completed; if it is greater than the threshold, the training is continued

3.2. Online Monitoring Stage

Step 1: perform the same data preprocessing process for the test data

Step 2: input the test data into the trained MCLSTM-CNN model

Step 3: complete the classification of fault data and analyze the results

Figure 6 shows the specific architecture of the MCLSTM-CNN network. Firstly, the preprocessed fault data are input into LSTM to obtain the hidden layer outputs and reshaped into a two-dimensional matrix to serve as the input of the convolutional layer. We choose different sizes of convolutional kernels $\text{Conv}_1, \text{Conv}_2, \dots, \text{Conv}_i$ (i is the number of parallel convolutional layers) to convolve the output data of LSTM simultaneously. The output of the aforementioned convolutional layers is then merged using the concatenated function. This is done, on the one hand, to connect to the next neural network layer, and on the other hand, to increase the diversity of features by merging them because the features extracted from the convolutional kernel of different sizes are different. Next, the pooling layer is introduced to compress the features so that the main features of the network layer are preserved while reducing the number of dimensions and parameters. The features are further extracted by using the Conv layer. The input data of the FC layer are required to be one-dimensional array, so the multidimensional output data of the Conv layer need to be flattened into one-dimensional data by the flatten layer. The dropout layer can randomly select neurons according to a certain probability to make them stop working during training, which can reduce the dependence between neurons and increase the generalization ability of the model. The FC layer is equivalent to the classifier in the model. The FC layer carries out a series of nonlinear operations on the feature values learned by the aforementioned network layer. The output results are the corresponding probability values of different classes, among which the largest probability value is the classification result.

4. Simulation Experiment and Analysis

4.1. TE Chemical Process. The TE (Tennessee Eastman) chemical process was first proposed at the American Chemical Society, which is a simulation of an actual chemical process. The TE chemical process has been widely utilized in research fields such as process control, optimization, monitoring, and fault diagnosis. The flow diagram is shown in Figure 7. In addition to normal data, the TE dataset also includes 21 preset different types of faults, including 16 known faults and 5 unknown faults 16–20. Each fault has 52 process variables, including 22 continuous process measurements, 19 compositions, and 11 manipulated variables. The data chosen for the experiment were all from public TE datasets.

4.2. Diagnostic Model Based on MCLSTM-CNN

4.2.1. Data Preprocessing. The experiment randomly selected 16,800 samples from the TE dataset, and each fault includes 800 samples, choosing 70% of the data for training and the remaining 30% for testing. Firstly, the data need to

FIGURE 6: The MCLSTM-CNN architecture.

FIGURE 7: Flow diagram of TE process.

be deformed according to sample size, timestep, and feature number to meet the input requirements of LSTM. Here, five time series are taken as a group to predict the output value at the next moment, so one input data is 5×52 , where 5 represents the time length of the sample and 52 represents the process variable. Because chemical process data have various dimensions, if the difference is too large, the

parameters in the model cannot learn other characteristics normally, which will have a great impact on the model results. Therefore, it is necessary to standardize the data according to formula (9). Figure 8(a) shows raw fault data, and Figure 8(b) shows standardized fault data. We can see that the 52 unprocessed process variable values vary greatly, while the standardized values fluctuate around 0.

$$x^* = \frac{x_i - \mu}{\sigma}, \quad (6)$$

where x_i is the sample data, μ is the sample mean, σ is standard deviation, and x^* are the normalized data.

When solving the problem of multiclassification, the label needs to be processed with one-hot encoding. The specific operation is to convert it into an array, where the column in the corresponding label is 1, and the rest is 0, such as the one-hot encoding of the fault 3, 21 shown in equation (7).

$$\begin{aligned} &[0001000000000000000000], \\ &[00000000000000000000001]. \end{aligned} \quad (7)$$

4.2.2. Construct Model. Parameter adjustment has always been a challenge in deep learning. When constructing the model, the number of neurons, convolutional kernels, parallel convolutional layers, and other parameters will have a great impact on the results of fault classification. In order to determine the appropriate parameters, 2 to 5 parallel convolutional layers were tried. We use the control variates method to change one model parameter at a time and show some structures in Table 1, comparing the diagnostic performance of different models and selecting the best performing models.

4.2.3. Training Model. After the model is built, the training set is input into the model for training. In the training process, categorical_crossentropy is used as the loss function. It can avoid the problem that the update weight is slower and is beneficial to model convergence. The calculation formula is as follows:

$$C = -\sum_i \log(q(x_i)), \quad (8)$$

where i is the correct class and $q(x_i)$ is the probability of predicting the correct class.

Adam is an optimization algorithm, which is very suitable for solving the optimization problems with large data or parameters and has high computational efficiency. Adam updates the network weight mainly through the following steps. For more details, refer to [32]. The formula for calculating exponential moving averages of the gradient and the square gradient is as follows:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \end{aligned} \quad (9)$$

where g_t is the gradient of timestep t , m_t is the exponential moving average of the gradient, v_t is the exponential moving average of the square gradient, and β_1 and β_2 are the exponential decay rates.

Since m_0 and v_0 are initialized to 0 at the beginning of training, the exponential moving average will be biased towards 0. Therefore, it is necessary to perform deviation correction for m_t and v_t , and the calculation is expressed in the following equations:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t}, \end{aligned} \quad (10)$$

where $\hat{m}_t$ is the deviation correction of m_t and $\hat{v}_t$ is the deviation correction of v_t .

Then, the weight update can be calculated by the following equation:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t, \quad (11)$$

where θ_t is the weight of timestep t , θ_{t+1} is the weight updated of timestep $t + 1$, η is the learning rate, and ε defaults to 10^{-8} .

The accuracy is used as an index to measure the quality of the model. The number of epochs is set to 40, and the batch size is set to 6. By adding EarlyStopping, the training is stopped when the monitoring value is no longer improved to prevent overfitting due to overly training epochs. Finally, we use the testing set to evaluate the performance of different MCLSTM-CNN models, and the results are shown in Figure 9.

It can be seen from Figure 9 that the fault classification accuracy of model 8 is the highest, so we finally choose model 8 as the research model. The model includes three parallel convolutional layers, and its network parameters are as follows. The input size of the sample data is 5×52 , and the number of neurons in the LSTM is 50. Conv₁, Conv₂, and Conv₃ all contain 128 convolutional kernels, whose sizes are 6, 4, and 2, respectively. The neurons generated by the concatenated function are 50, and the number of channels is 384. The rejection rate of the dropout layer is 0.1, and the kernel of the pooling layer is 2. Then, the generated data size is 25×384 . Conv has 256 convolutional kernels of size 2, and the generated data are 24×256 . Through the flatten layer processing, the size is 6144. The output length of the FC₁ layer is set to 300, and the output length of the FC₂ layer is 22. The activation function before the FC₂ layer uses the Rectified Linear Unit (Relu), and the activation function of the FC₂ layer uses softmax.

Figure 10 shows the accuracy of model 8 in training process. We can see that the accuracy converges faster in the early stage and keeps improving. With the increase of training epochs, accuracy gradually tends to be stable and eventually reaches 96.74%.

Since the output of each layer of the MCLSTM-CNN is high-dimensional data, it cannot be visualized. In order to understand its feature learning and classification process, a method of t-distributed stochastic neighbor embedding (t-SNE) is adopted. This method is a derivative of SNE, which can project high-dimensional data into a two-dimensional plane for visualization. Figure 11 shows the learning process of training data, where each point is a sample, and the actual class of each data is marked on the point. To make the classification effect more intuitive, each fault uses a different color. It can be observed from Figures 11(a)–11(g) that the raw sample data are mixed but are gradually separated by the processing of the MCLSTM-CNN.

FIGURE 8: Raw and processed data. (a) Raw data. (b) Processed data.

TABLE 1: Architecture of MCLSTM-CNN model.

Model	Architecture
Model 1	LSTM(50)-Conv ₁ (64)-Conv ₂ (64)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 2	LSTM(50)-Conv ₁ (64)-Conv ₂ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 3	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 4	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 5	LSTM(50)-Conv ₁ (64)-Conv ₂ (64)-Conv ₃ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 6	LSTM(50)-Conv ₁ (64)-Conv ₂ (128)-Conv ₃ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 7	LSTM(50)-Conv ₁ (64)-Conv ₂ (128)-Conv ₃ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 8	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 9	LSTM(50)-Conv ₁ (64)-Conv ₂ (64)-Conv ₃ (128)-Conv ₄ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 10	LSTM(50)-Conv ₁ (64)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 11	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 12	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 13	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv ₅ (128)-Conv(128)-FC ₁ (300)-FC ₂ (22)
Model 14	LSTM(50)-Conv ₁ (64)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv ₅ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 15	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv ₅ (128)-Conv(256)-FC ₁ (300)-FC ₂ (22)
Model 16	LSTM(50)-Conv ₁ (128)-Conv ₂ (128)-Conv ₃ (128)-Conv ₄ (128)-Conv ₅ (256)-Conv(256)-FC ₁ (300)-FC ₂ (22)

4.3. Analysis of Results. In order to visualize the fault classification results of the MCLSTM-CNN model, we introduce a confusion matrix. Due to space constraints, 21 faults are split into two parts. Figure 12(a) shows the fault classification results of 1–10, and Figure 12(b) shows the fault classification results of 11–21. In the confusion matrix, the row is the class of the prediction, and the column is the

FIGURE 9: Classification accuracy of different models.

FIGURE 10: Accuracy of training.

actual situation. The diagonal in the figure is the class where the true value is equal to the predicted value, and the rightmost column is the corresponding classification accuracy.

We found that the classification accuracy of different faults varied greatly. The accuracy of faults 3 and 9 is lower than 80%, the accuracy of fault 15 is lower than 70%, and the

FIGURE 11: Visualization of training process using t-SNE: (a) the input data, (b) the output of the LSTM, (c) the output of the concatenate, (d) the output of the pooling, (e) the output of Conv, (f) the output of FC_1 , and (g) the output of FC_2 .

accuracy of faults 10, 16, and 20 is less than 90%. In addition, the classification accuracy of other faults is above 90%, among which the classification accuracy of faults 1, 5, 6, 7, and 14 is 100%. Therefore, most faults can be effectively separated by MCLSTM-CNN, and only a few faults perform poorly. By analyzing the data in the table, it is found that the lower accuracy of the faults 3, 9, and 15 is caused by the higher degree of confusion between them. For example, for fault 3, 22 samples

are misclassified to fault 9, and 16 samples are misclassified to fault 15. It is found that both faults 3 and 9 are related to the temperature change of material D. The difference is that the type of fault 3 is a step, while fault 9 is a random variable. Fault 15 is a condenser cooling water valve, which is also related to the material temperature change of fault 3 and fault 9.

To verify the effectiveness of the proposed method, six different models of LSTM-CNN, LSTM, CNN, RF,

Predict Lables	1	2	3	4	5	6	7	8	9	10	11	13	15	16	17	18	19	20	21
1	227	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
2	0	229	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	98.28
3	0	0	182	1	0	0	0	0	22	1	0	3	16	8	1	5	1	2	73.68
4	0	0	0	243	0	0	0	0	0	0	1	0	0	1	0	0	0	0	99.18
5	0	0	0	0	222	0	0	0	0	0	0	0	0	0	0	0	0	0	100.0
6	0	0	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	100.0
7	0	0	0	0	0	0	229	0	0	0	0	0	0	0	0	0	0	0	100.0
8	0	1	0	0	0	0	0	253	0	0	0	0	0	1	0	0	0	0	99.22
9	0	0	28	0	0	0	0	0	172	5	1	0	21	1	0	3	1	7	71.37
10	0	0	8	0	5	0	0	0	8	204	1	2	7	10	1	0	0	0	82.26

(a)

Predict Lables	3	4	5	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
11	1	8	0	0	3	0	227	0	0	0	1	0	0	3	1	1	0	92.65
12	0	0	4	6	0	0	1	206	3	0	0	0	0	0	0	0	0	93.64
13	0	0	0	2	1	0	0	1	258	0	0	0	0	0	0	0	0	98.47
14	0	0	0	0	0	0	0	0	0	249	0	0	0	0	0	0	0	100.0
15	27	0	0	0	25	3	0	0	4	0	166	5	0	3	3	2	2	69.17
16	2	0	0	1	5	7	0	0	3	0	3	205	0	0	0	0	0	85.40
17	0	0	0	0	0	0	0	0	0	0	0	0	229	0	0	0	1	99.57
18	3	0	0	1	1	1	3	9	0	0	1	0	0	248	0	1	0	92.54
19	2	0	0	0	3	0	2	0	0	0	0	0	0	2	227	0	0	96.19
20	5	0	0	3	6	0	0	0	0	0	13	0	0	1	0	200	1	87.34
21	12	0	0	0	6	0	0	0	0	0	1	0	0	0	2	1	201	90.13

(b)

FIGURE 12: Confusion matrix: (a) confusion matrix for faults 1–10; (b) confusion matrix for faults 11–21.

KPCA + SVM, and MCLSTM-CNN are established, respectively, and their fault diagnosis classification performance in the TE process is compared. The test accuracy of the MCLSTM-CNN model is as high as 92.06%, while the test accuracy of LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM is 89.05%, 85.12%, 75.84%, 87.8%, and 64.2%, respectively.

The ROC curve is widely applied as a metric to evaluate the classification effect. The horizontal coordinate is FPR (false-positive rate), and the vertical coordinate is TPR (true-positive rate). The samples are classified by setting a threshold value. The closer the ROC curve is to the upper left corner, the better the diagnostic effect will be. In order to more intuitively indicate the overall classification results of the four models, the ROC curve is shown in Figure 13, where Figures 13(a)–13(f) show the ROC curves of MCLSTM-CNN, LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM, respectively. It can be seen from Figure 13 that although six models have different diagnostic performances for various faults, the overall diagnosis results based on the MCLSTM-CNN model is the best. There is no significant difference between LSTM-CNN and RF, followed by LSTM, and the diagnostic results of CNN and KPCA + SVM are worst.

Precision and recall can evaluate the diagnostic performance of the model for a single fault, but in general, they are often contradictory. High precision leads to low recall, while high recall leads to low precision, so it is difficult to

compare different models. F1 score considers both the precision and the recall and can evaluate different models more effectively. The calculation formulas are as follows:

$$\begin{aligned}
 P &= \frac{TP}{TP + FP}, \\
 R &= \frac{TP}{TP + FN}, \\
 F &= \frac{2 \times P \times R}{P + R},
 \end{aligned} \tag{12}$$

where R is the recall, P is the precision, F is the F1 score, TP is the sample for which the predicted label matches the real label, FN is the sample for which the actual label is mispredicted, and FP is the sample for which the predicted label is wrong. Figure 14 shows the F1 Score of the six models.

The average F1 scores of MCLSTM-CNN, LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM models are 92.03%, 89.06%, 85.09%, 76.23%, 88.18%, and 65.37%, respectively. This indicates that MCLSTM-CNN has the best overall diagnosis effect. For the MCLSTM-CNN model, the F1 scores of faults 3, 9 and 15 are less than 80%, and the F1 scores of faults 10, 16 are less than 90%. In addition, the F1 scores of other faults are higher than 90%. However, the F1 score of LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM models for single fault is mostly lower than that of the

FIGURE 13: Continued.

FIGURE 13: ROC curves for different models. (a) MCLSTM-CNN. (b) LSTM-CNN. (c) LSTM. (d) CNN. (e) RF. (f) KPCA + SVM.

FIGURE 14: F1 score of the six models.

MCLSTM-CNN model. Therefore, MCLSTM-CNN can separate individual faults more effectively than other models. Also, we found that for faults 3, 9, and 15, the F1 score of different models is quite different. However, the MCLSTM-CNN model has the highest F1 score, which significantly improves the diagnostic accuracy of these three faults. Based on the above analysis results, we can see that the MCLSTM-

CNN model is more suitable for diagnosing faults in chemical process.

5. Conclusion

This paper proposes a chemical process fault diagnosis method based on MCLSTM-CNN. The fault diagnosis model

is constructed with the LSTM, convolutional layer, pooling layer, dropout layer, and FC layer, which can effectively extract the time domain and spatial features of data. This method is applied to the fault diagnosis of the TE chemical process, and the fault classification results of LSTM-CNN, LSTM, CNN, RF, and KPCA + SVM are compared. The experimental results show that the diagnostic accuracy of MCLSTM-CNN is 92.06%, which is obviously superior to other methods. And for faults 3, 9, and 15, which are difficult to diagnose, MCLSTM-CNN has been greatly improved.

The MCLSTM-CNN fault diagnosis method proposed in this paper has high fault diagnosis accuracy and good reliability, and it is easier to separate different fault types. It can effectively monitor the production process so as to reduce or even avoid accidents. Therefore, the MCLSTM-CNN model is preferably applied for fault diagnosis of chemical process. In this work, there are still some limitations. As far as the model is concerned, the changes of any parameters in the network will affect the quality of the model, and it is difficult to find the optimal model. In addition, when the parameters are constant, the result of each training may also have errors.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

B. S. provided the research idea. X. H. conceived and designed the experiments. G. B. contributed to the validation and visualization of the model. Y. Z. contributed to the article's organization and drafted the manuscript.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant nos. 61672416, 61272458, and 61872284).

References

- [1] T. Rato, M. Reis, E. Schmitt, M. Hubert, and B. De Ketelaere, "A systematic comparison of PCA-based statistical process monitoring methods for high-dimensional, time-dependent processes," *AIChE Journal*, vol. 62, no. 5, pp. 1478–1493, 2016.
- [2] S. Yin, X. D. Steven, A. Naik, P. Deng, and A. Haghani, "On PCA-based fault diagnosis techniques," in *Proceedings of the 2010 Conference on Control and Fault-Tolerant Systems*, Nice, France, October 2010.
- [3] S. Yin, X. Zhu, and O. Kaynak, "Improved PLS focused on key-performance-indicator-related fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1651–1658, 2015.
- [4] R. Leardi, "Application of genetic algorithm-PLS for feature selection in spectral data sets," *Journal of Chemometrics*, vol. 14, pp. 643–655, 2010.
- [5] G. Chen, J. Liang, and J. Qian, "ICA and its application to chemical process monitoring and fault diagnosis," *Journal of Chemical Industry Engineering*, vol. 54, pp. 1474–1477, 2003.
- [6] J.-M. Lee, C. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *Journal of Process Control*, vol. 14, no. 5, pp. 467–485, 2004.
- [7] Z. Gao, Z. Chen, J. Gao et al., "Status monitoring of chemical system based on improved LFDA," *Computer Integrated Manufacturing System*, vol. 22, pp. 1097–1103, 2016.
- [8] B. Li, P.-L. Zhang, D.-S. Liu, S.-S. Mi, and P.-Y. Liu, "Classification of time-frequency representations based on two-direction 2DLDA for gear fault diagnosis," *Applied Soft Computing*, vol. 11, no. 8, pp. 5299–5305, 2011.
- [9] X. G. Deng and X. M. Tian, "Nonlinear process fault diagnosis method using kernel principal component analysis," *Journal of Shandong University (China)*, vol. 35, no. 3, pp. 103–106, 2005.
- [10] G. Stefatos and A. Ben Hamza, "Dynamic independent component analysis approach for fault detection and diagnosis," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8606–8617, 2010.
- [11] Y. Dong and S. J. Qin, "A novel dynamic PCA algorithm for dynamic data modeling and process monitoring," *Journal of Process Control*, vol. 67, pp. 1–11, 2018.
- [12] Y. E. Shao, C.-J. Lu, and Y.-C. Wang, "Hybrid ICA-SVM approach for determining the quality variables at fault in a multivariate process," *Mathematical Problems in Engineering*, vol. 2012, Article ID 284910, 12 pages, 2012.
- [13] L. L. Ma, Z. Zhang, and J. Z. Wang, "Combination method of support vector machine and fisher discriminant analysis for chemical process fault diagnosis," in *Proceedings of the 29th Chinese Control Conference*, Beijing, China, July, 2010.
- [14] K. Niu, F. Zhao, and S. Zhang, "A fast classification algorithm for big data based on KNN," *Journal of Applied Sciences*, vol. 13, no. 12, pp. 2208–2212, 2013.
- [15] G. Wang, J. Liu, and Y. Li, "Fault diagnosis using KNN reconstruction on MRI variables," *Journal of Chemometrics*, vol. 29, no. 7, pp. 399–410, 2015.
- [16] W. U. Ping, S. B. Zhang, Z. B. Yan et al., "Fault isolation method based on Bayesian Lasso for industrial process," *Journal of Optoelectronics-Laser*, vol. 29, pp. 627–633, 2018.
- [17] J. T. Ren, S. D. Lee, X. L. Chen et al., "Naive Bayes classification of uncertain data," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, Miami, FL, USA, December, 2009.
- [18] J. Wang, P. Li, R. Ran, Y. Che, and Y. Zhou, "A short-term photovoltaic power prediction model based on the gradient boost decision tree," *Applied Sciences*, vol. 8, no. 5, pp. 689–702, 2018.
- [19] S. He, G. A. Wang, M. Zhang, and D. F. Cook, "Multivariate process monitoring and fault identification using multiple decision tree classifiers," *International Journal of Production Research*, vol. 51, no. 11, pp. 3355–3371, 2013.
- [20] K. Zhang, K. Qian, Y. Chai, Y. Li, and J. Liu, "Research on fault diagnosis of Tennessee eastman process based on KPCA and SVM," in *Proceedings of the 2014 Seventh International Symposium on Computational Intelligence and Design*, Hangzhou, China, December 2014.
- [21] Y. Zhang, J. Chen, and X. F. Wang, "Application of random forest on rolling element bearings fault diagnosis," *Computer Engineering and Applications*, vol. 54, no. 6, pp. 100–104, 2018.

- [22] J. C. Hu, G. Y. Huang, Z. K. Shao et al., "Fault diagnosis of chemical process instrument based on ICA-PCA," *Computer and Applied Chemistry*, no. 7, pp. 823–826, 2013.
- [23] Y. Zhang, "Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM)," *Industrial & Engineering Chemistry Research*, vol. 47, no. 18, pp. 6961–6971, 2008.
- [24] E. P. D. Moura, C. R. Souto, A. A. Silva, and M. A. S. Irmão, "Evaluation of principal component analysis and neural network performance for bearing fault diagnosis from vibration signal processed by RS and DF analyses," *Mechanical Systems & Signal Processing*, vol. 25, no. 5, pp. 1765–1772, 2011.
- [25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [27] W. Sun, S. Y. Shao, and R. Q. Yan, "Induction motor fault diagnosis based on deep neural network of sparse auto-encoder," *Journal of Mechanical Engineering*, vol. 52, no. 9, pp. 65–71, 2016.
- [28] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Computers & Chemical Engineering*, vol. 107, pp. 395–407, 2017.
- [29] X. Zhang, Z. Cui, Y. X. Dong et al., "Application of fault classification method based on VAE-DBN in chemical process," *Journal of Process Engineering*, vol. 18, pp. 590–594, 2018.
- [30] H. Zhao, S. Sun, and B. Jin, "Sequential fault diagnosis based on LSTM neural network," *IEEE Access*, vol. 6, pp. 12929–12939, 2018.
- [31] H. Wu and J. Zhao, "Deep convolutional neural network model based chemical process fault diagnosis," *Computers & Chemical Engineering*, vol. 115, pp. 185–197, 2018.
- [32] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, CA, USA, May 2015.

Submit your manuscripts at
www.hindawi.com