



Incipient Fault Diagnosis of Batch Process Based on Deep Time Series Feature Extraction

Min Zhang^{1,2} · Xianjun Li¹ · Ruiqi Wang¹

Received: 20 August 2020 / Accepted: 18 January 2021 / Published online: 3 February 2021
© King Fahd University of Petroleum & Minerals 2021

Abstract

Intelligence modeling interpretable for incipient fault diagnosis of batch process represents a serious challenge. For the multivariate, nonlinear, and high-dimensionality characteristics of process data, existing fault diagnosis solutions are easily concealed by noise while neglect the low amplitude and noise interference of the incipient faults. In this paper, an intelligent incipient fault diagnosis model based on deep time series feature extraction network is proposed, which integrates denoising autoencoder (DAE), layer normalization and dropout layer added LSTM (LD-LSTM), and stacked autoencoder (SAE) to obtain the efficient features. DAE is applied to restore the data and eliminate noise interference. LD-LSTM is utilized to extract time series features, in which the layer normalization and the dropout layer are added between the layers of the LSTM to solve the problems of slow model convergence and over-fitting. SAE performs the feature extraction to obtain deep time series features and learns more efficient expressions. The softmax function is applied to achieve the incipient fault diagnosis based on the deep time series features. A case study on a fed-batch penicillin fermentation process suggests that the proposed method can obtain an accuracy rate of 98.97%, which has increased an average of 20% than the traditional shallow machine learning. The accuracy of fault recognition and comparative experiments demonstrate the effectiveness and feasibility of this model, and provide a solution for the application of intelligent fault diagnosis in batch industries.

Keywords Batch process · Incipient fault diagnosis · Deep time series feature extraction · LSTM

1 Introduction

Batch process has been accounted for an increasing proportion in complex industrial fields such as food, medical, and chemical industries. With the safety and stability of batch process gain more and more attention, intelligent fault diagnosis as an important technical method in ensuring reliability has become a popular research topic in academia and industry [1]. Due to the discontinuous operation and complex production environment, the process data of incipient faults have the characteristics of the multivariate, nonlinear, and high-dimensionality. When an incipient fault occurs, it is easy to be covered by noise because of its low

fault amplitude, and it will be extremely difficult to diagnose the fault effectively [2–4]. How to effectively diagnose and identify incipient faults of batch process has grown up to be an increasingly serious problem.

With the increase in complexity of batch process, it is difficult to establish an accurate analytical mathematical model or an efficient expert knowledge system. In the past decades, fault diagnosis methods based on data-driven have been widely used in the batch process, which can be typically divided into two categories: multivariate statistical models and machine learning models. Many multivariate statistical methods have been applied to obtain the incipient fault features, like the cumulative sum, principal component analysis, kernel principal component analysis, and kernel entropy component analysis [5–9]. Cumulative sum as an effective way of data analysis is utilized to detect small changes in the process. As the correlation between the variables of the complex production process is closely related, the traditional variable cumulative sum control charts may generate the false alarm and the missed report [5]. Principal component analysis and its improved methods have used to

✉ Min Zhang
zhmzhangmin16@126.com

¹ School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China

² Technology and Equipment of Rail Transit Operation and Maintenance Key Laboratory of Sichuan Province, Chengdu 610031, China



research batch production process monitoring [6, 7]. However, these approaches do not suitable for solving nonlinear problems. Kernel principal component analysis and kernel entropy component analysis are used to realize the nonlinear fault problems [8, 9]. Statistical methods have made great achievements in fault detection and diagnosis of batch process. However, with the wide application of distributed control systems and the continuous development of data storage, transmission, and processing technologies, intelligent fault diagnosis method based on machine learning has become more suitable. It can effectively analyze and judge multi-source and high-dimensional data [10].

The machine learning method can overcome the shortcomings of traditional detection methods that cannot judge the exact fault type. It can use process data to learn the fault mode features and directly judge the fault model to realize the fault identification [11]. Support vector machine (SVM) as a typical machine learning method has increased attention in fault diagnosis of the batch process because it can obtain remarkable results [12]. Ji et al. [13] put forward an algorithm based on ensemble empirical mode decomposition and support vector machine in the sensor fault detection and classification. However, SVM belongs to the shallow network in structure, which limits the classifier's ability to learn the complex nonlinear and non-stationary relationships contained signals. With developing computing power and algorithms, many deep learning technologies have marched rapidly [14]. Novel convolutional neural networks (CNN) [15], recurrent neural networks (RNN) [16], long short-term memory (LSTM) [16], and other deep neural network methods are growing fast. It can be seen that the deep learning methods have a precedent in the application of batch process fault diagnosis. Wang and Wang [17] introduced LSTM to perform feature learning to obtain the deep feature expression of the data and realized the fault diagnosis of batch process. Zhang et al. [18] established a weighted autoregressive LSTM network to extract in-deep features and realize effective fault diagnosis in the TE process. Despite the progress made by state-of-the-art methods on intelligent fault diagnosis, the focus of incipient fault diagnosis in the batch process is how to obtain the effective characteristics of the data; the deep neural network still suffers from the following three limitations:

- (1) The incipient faults have low amplitude and are susceptible to noise interference, and are easily concealed by noise when processing with machine learning methods.
- (2) The data of batch process present the characteristics of nonlinearity, high-dimensionality, and mutual coupling of various variables, which can be regarded as a multivariate time series. Traditional machine learning

methods cannot effectively obtain in-deep characteristics.

- (3) Traditional deep neural network models, such as original LSTM, do not sufficiently and properly handle the incipient faults information inherent in batch process data.

In this paper, we propose an intelligent incipient fault diagnosis model based on the deep time series feature extraction network. Firstly, denoising autoencoder (DAE), layer normalization and dropout layer added between the layers of the LSTM (LD-LSTM), and stacked autoencoder (SAE) are integrated to obtain the efficient features. Then, the softmax function is applied to achieve the incipient fault diagnosis based on the deep time series features. Finally, a fed-batch penicillin fermentation process is utilized to compare and analyze the accuracy of different diagnosis models according to five evaluation metrics most used in fault diagnosis fields. The main contributions of this paper are summarized as follows:

- (1) An interpretable data-driven incipient fault diagnosis model is proposed based on deep time series feature extraction network, which integrates DAE, LD-LSTM, and SAE to obtain the efficient features of batch process time series data.
- (2) For solving the problems of slow model convergence and over-fitting, layer normalization and dropout layers are added between the layers of the LSTM (LD-LSTM) to improve the performance.
- (3) The proposed incipient fault diagnosis model tends to obtain better fault diagnosis accuracy and achieves more stable results in the batch process when compared to several other models.

The remainder of this paper is organized as follows. Section 2 describes the main concepts used in this paper. In Sect. 3, the detailed procedures of the incipient fault diagnosis model based on the deep time series feature extraction network are introduced. The experimental evaluation is given in Sect. 4. Section 5 presents conclusions and suggestions for future work.

2 Preliminaries

2.1 Autoencoder Network and its Variants

2.1.1 Basic Autoencoder

Autoencoder (AE) as a type of artificial neural network is employed in unsupervised learning or semi-supervised learning. It is primarily used for data dimensionality

reduction and feature extract for the efficient representation of input data [19]. AE network usually consists of an encoder and a decoder. The encoder is used to get the representation y of input vector x . And the decoder uses this representation to reconstruct the original input vector by minimizing the residual vector between the input vector x and output vector \hat{x} . Because the reconstruction loss is small enough, the representation y could retain the most information of the input vector x [20]. The number of neurons in the hidden layer of the AE network is smaller than the input and output layers, which can be regarded as an efficient expression of data. The encoder receives the input data and passes to the hidden layer for obtaining a compressed representation of the data. The decoder passes the low-dimensional data to the output layer for obtaining the original data reconstruction. Figure 1 shows the classic AE network structure.

In the AE network training process, the input data x is mapped and transferred to the hidden layer to obtain the encoding y , and then reverse decoding is performed through the encoding to obtain the input reconstruction \hat{x} . The dimension of the input vector x is m . And the number of the input samples is n . The dimension of output vector y is m' .

Encoder: The encoder is formulated as:

$$y = f_{\text{encoder}}(W_e x + b_e) \tag{1}$$

where f_{encoder} is an activation function, such as sigmoid, tanh, identity, and so on, W_e is a $m' \times m$ weight matrix, and b_e is an offset vector whose dimension is m' .

Decoder: The decoder is formulated as:

$$\hat{x} = f_{\text{decoder}}(W_d y + b_d) \tag{2}$$

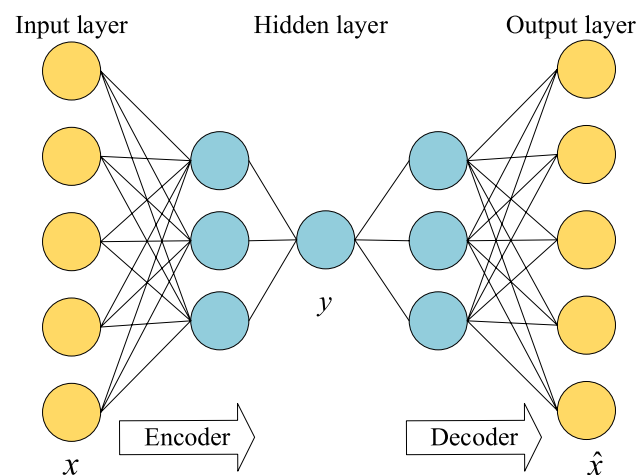


Fig. 1 The classic AE network structure

where f_{decoder} is also an activation function, W_d is a $m \times m'$ weight matrix, and b_d is an offset vector whose dimension is m .

Loss function: Autoencoder obtains a suitable parameter $\theta = (W_e, W_d, b_e, b_d)$ through minimizing the loss function. The loss function is formulated as:

$$J(\theta) = L(x, \hat{x}) + \lambda \|W\|^2 = \sum_{i=1}^n \|x_i - \hat{x}_i\|^2 + \lambda (\|W_e\|^2 + \|W_d\|^2) \tag{3}$$

where $\lambda \|W\|^2$ is a regularized item to avoid over-fitting by minimizing the L2 norm of parameters, and $L(x, \hat{x})$ is reconstruction loss.

The goal of AE network training is to minimize the error of the input range reconstruction value. The intermediate coding y is the focus, which can be regarded as an efficient expression of the input data.

2.1.2 Denoising Autoencoder (DAE)

Denoising autoencoder (DAE) is an extension of autoencoder. It adds noise to the input data, and the training enables the autoencoder to obtain the data that is not polluted by noise from the input data [21]. The training process makes the encoder to learn to extract the most significant features of the input data, and its generalization ability is better than that of general encoders. The DAE model avoids the shortcoming of that the general AE model may learn the identity function and the limitation of sample number required greater than the sample dimension. By minimizing the noise reduction reconstruction error, the true original input is reconstructed from the data containing random noise. The added noise to the input data is equivalent to neurons losing data with a certain

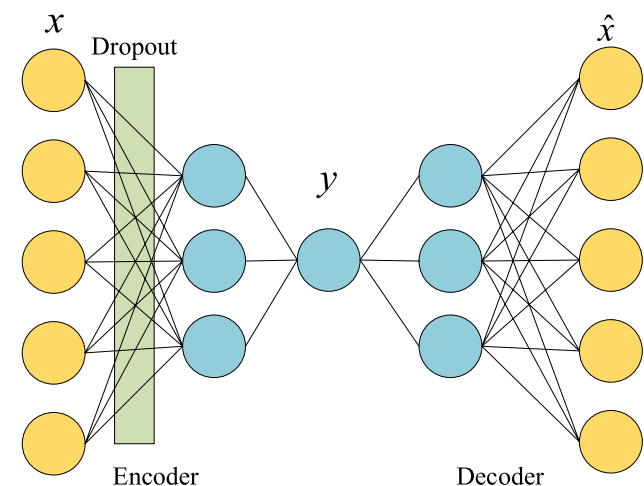


Fig. 2 The schematic diagram of the DAE structure

probability, which can be regarded as a common drop-out operation in neural networks. The model structure is shown in Fig. 2.

2.1.3 Stacked Autoencoder (SAE)

When handling the complex process datasets, the representation extracted by a single autoencoder may not meet the requirement. And the stacked autoencoder (SAE) could generate a more appropriate representation, which is constructed by stacking several individual autoencoders, and each layer of the network is trained in turn using the layer-by-layer greedy training method [22].

Given a stacked autoencoder containing l autoencoders, the process of encoding is formulated as:

$$y = f_{\text{encoder}_l}(f_{\text{encoder}_{l-1}}(\dots f_{\text{encoder}_1}(x))) \tag{4}$$

where f_{encoder_i} is the activation function of the i th encoder. The process of decoding is formulated as:

$$\hat{x} = f_{\text{decoder}_l}(f_{\text{decoder}_{l-1}}(\dots f_{\text{decoder}_1}(y))) \tag{5}$$

where f_{decoder_i} is the activation function of the i th decoder. The autoencoders in the stacked autoencoder can be trained sequentially.

The steps of training an SAE are to use the original input data to train the first AE network, apply the hidden layer output of the first AE network as the input of the second AE network, and then use the same method to train, repeat the above steps several times to obtain the output of the last hidden layer as an efficient expression, which can be regarded as the deep characteristics of the original data (see Fig. 3).

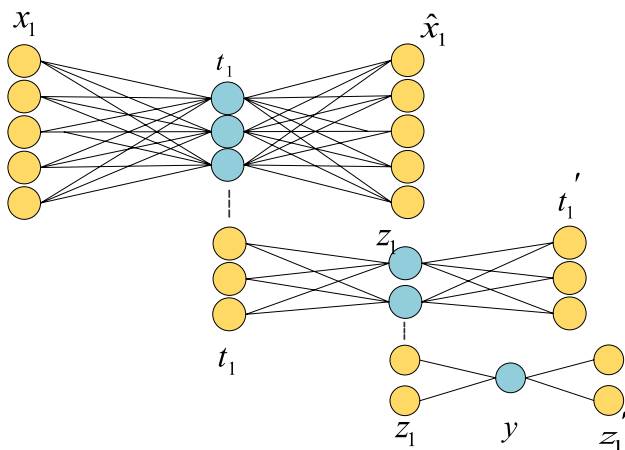


Fig. 3 The schematic diagram of SAE structure

2.2 Long Short-Term Memory

The long short-term memory network is essentially an improved recurrent neural network (RNN). The neurons in the same layer of the fully connected network are independent of each other and are subject to greater constraints when dealing with sequence problems [23]. RNN is suitable for solving sequence problems, it adds a loop structure to the hidden layer of the traditional neural network, the neuron at the current moment is connected with the neuron at the previous moment, and then, the information of the previous time step can be stored. Specifically, the output of the RNN at the current moment depends not only on the input at the current moment but also on the intermediate state at the previous moment. Through this dependency transmission, the RNN ensures the correlation between neurons in the same layer, and the structure is as shown in Fig. 4.

Where x_t denotes the input at time t , s_t denotes the intermediate state at time t , which can be understood as the ‘memory’ of the neuron at the current time. o_t represents the output at time t , U, W, V, b_1, b_2 are the weight parameters that need to be learned, and the weights are shared in the same layer. The schematic diagram of the RNN structure, we can obtain the following equations:

$$s_t = f_1(Ux_t + Ws_{t-1} + b_1) \tag{6}$$

$$o_t = f_2(Vs_t + b_2) \tag{7}$$

The input sequence of RNN can be infinitely long, but in practical applications, it is found that RNN can only establish strong connections with neurons in a relatively close range [24]. When RNN is used to deal with problems, only relatively recent information is needed. However, if the time interval becomes particularly long, the RNN may lose its effect and cause problems such as gradient disappearance and gradient explosion. The connection weight within the RNN network decreases as the interval between prediction information and related information increases. The LSTM

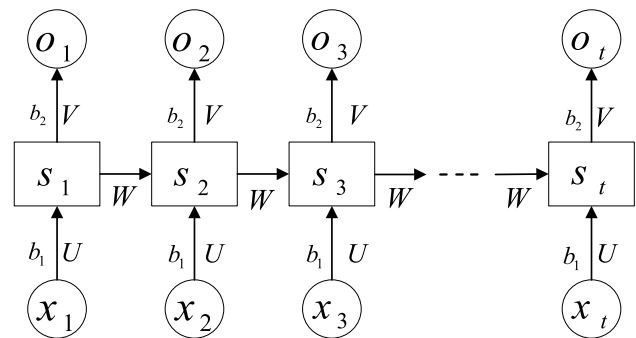


Fig. 4 Schematic diagram of the RNN structure

network is proposed to solve the problem of long-term dependence.

On the basis of the RNN network, the LSTM network introduces a cell control mechanism inside the cell, which effectively solves the problem of gradient disappearance and gradient explosion [25]. The transmission of state information is like walking along a conveyor belt; the information passes through the entire cell after a few linear operations. This structure allows the information to pass through the entire cell without change, and its structure is shown in Fig. 5.

In Fig. 5, the upper horizontal line has no way to add or delete the cells' information, so the gate structure is introduced to process information. The gate structure can change the degree of information retention through a sigmoid neural layer and point-by-point multiplication. The sigmoid function transforms the input data into the [0,1] interval, which represents the probability of the corresponding information passing. When it is 0, it means that the information cannot pass, and when it is 1, it means

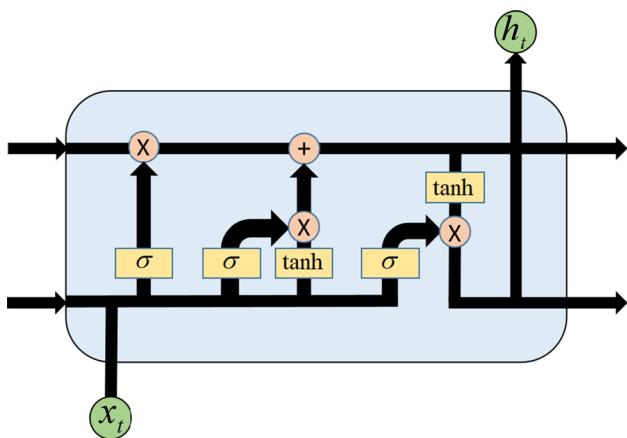


Fig. 5 LSTM cell structure

that the information can all pass. Each LSTM cell unit has three gate structures to control the passage of information, which are forget gate, input gate, and output gate [26].

3 Proposed Deep Time Series Feature Extraction Model

In actual industrial production, equipment operating status data collected by sensors are often disturbed, which increases the difficulty of feature extraction and fault diagnosis. This paper proposes a deep time series feature extraction model to realize intelligent fault diagnosis. The deep time sequence features of the fault data are obtained by a three parts network, which integrates DAE, LD-LSTM, and SAE. The structure of the model is shown in Fig. 6.

The first part of the model in this paper is the DAE layer with denoising and data restoration capabilities. The original data should be standardized and substituted into the DAE network. The DAE network is composed of 7 layers of AE networks. The number of neurons in the input layer is the total number of original data n . The number of neurons in the second layer is 900, the number of neurons in the third layer is 300, and the number of neurons in the hidden layer is 64. Due to the symmetry of the AE neural network structure, the number of subsequent neurons is symmetrically equal to the first three layers, 300, 900, and n , respectively. It adds a dropout layer after the input layer to randomly deactivate some neurons so that to achieve the effect of restoring the real data after training. The output data of the layer are used as the input of the next layer.

In terms of time series processing, LSTM can obtain the essential features from complex time series. Besides, LSTM with layer normalization and dropout layer (LD-LSTM) can solve the problems of slow model convergence and overfitting. Therefore, the second part of the model uses the LD-LSTM network to perform feature extraction. The LSTM network layer adopts a double-layer cyclic structure, the

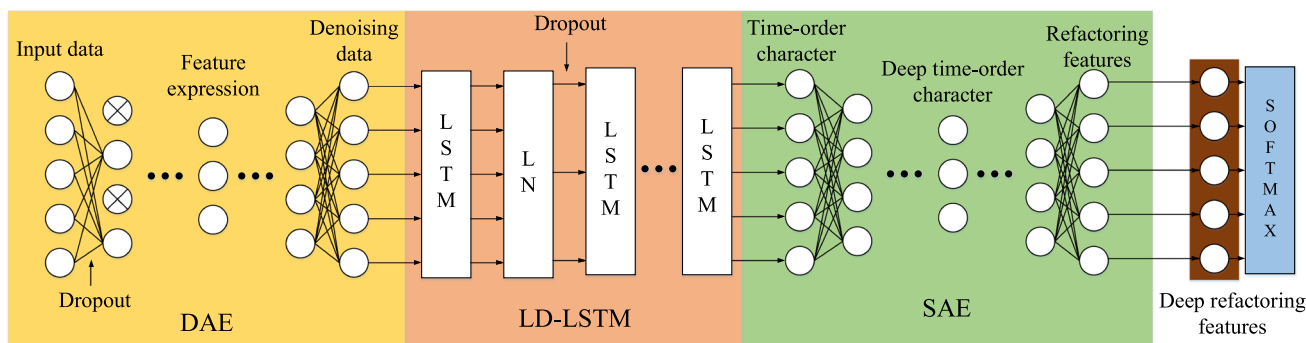


Fig. 6 Deep time series feature extraction model

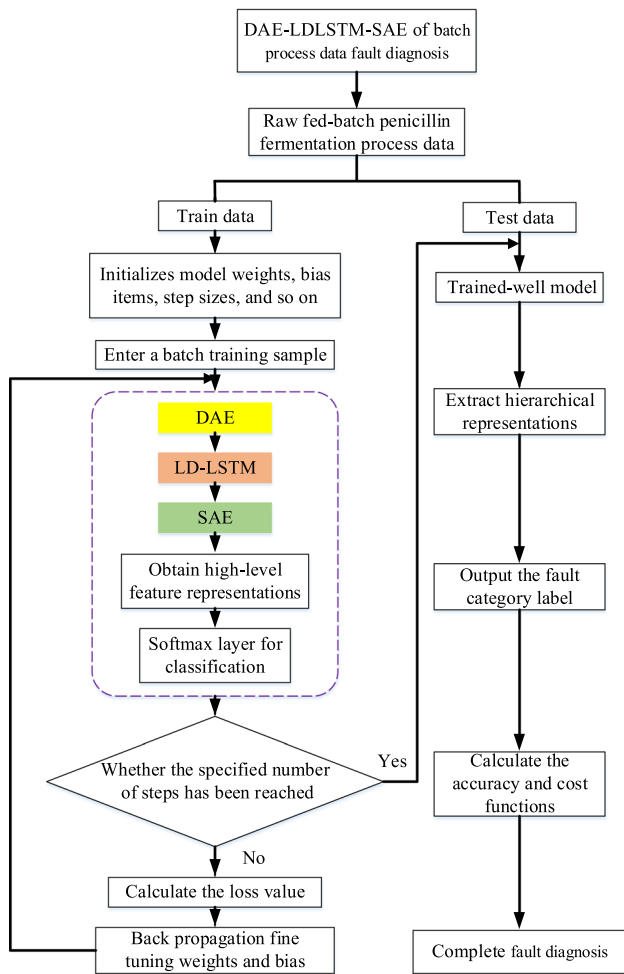


Fig. 7 The flowchart of the proposed deep time series feature extraction model

number of hidden layers is set to 128, and the value at the last moment is taken as the output time series features.

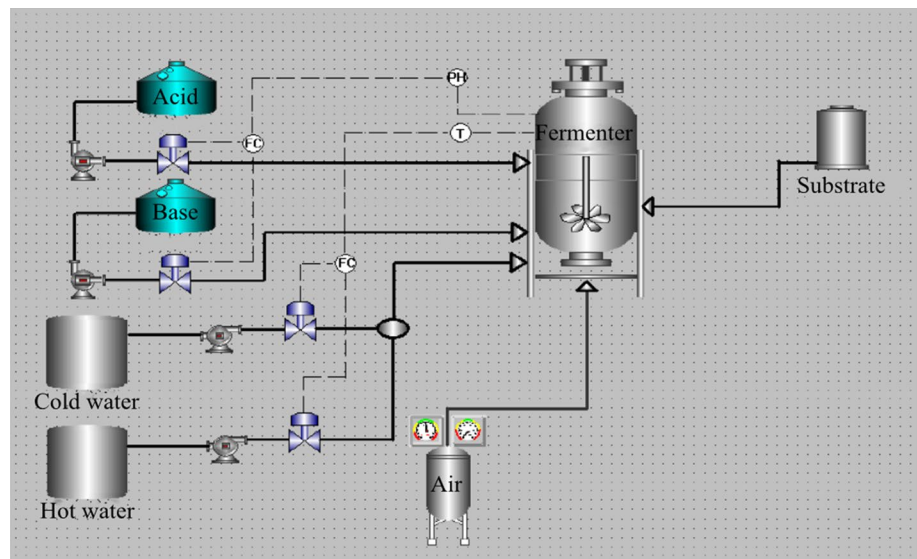
However, the time series features extracted through the LD-LSTM network still have irrelevant and redundant problems, which are not conducive to the incipient fault diagnosis. Therefore, the third part of the model uses the SAE layer to perform feature compression and dimensionality reduction, and extract the most critical feature of the time series feature. In this paper, SAE with three layers is used, in which the numbers of neurons are 128, 64, and 128, respectively. The input layer of AE is equal to the number of hidden layers of LD-LSTM. Through the SAE network layer, the time sequence features obtained from the LD-LSTM network layer are extracted layer by layer, and then, the deep time sequence features are flattened through the output layer and used as the input of the softmax classifier, and finally to achieve the purpose of fault diagnosis. The flowchart of the proposed deep time series feature extraction model is shown in Fig. 7.

4 Experimental Research

4.1 Data Source

The penicillin fermentation process is the metabolic activity of the penicillin-producing bacteria to synthesize antibiotics under appropriate conditions of medium, temperature, pH, and stirring. The schematic diagram of the fermentation process is shown in Fig. 8. The penicillin fermentation process is a typical multi-stage biochemical reaction process. From the perspective of operating technology, it is mainly divided into two stages: batch operation stage and semi-batch feeding operation stage. In the initial batch operation stage, most

Fig. 8 Schematic diagram of penicillin simulation experiment platform



of the bacteria are produced in this stage, and the initially added substrate (glucose) is also consumed by the bacteria. When the reaction enters the second semi-batch feeding operation stage, the growth rate of the bacteria needs to be less than a certain minimum value to ensure a higher yield. Therefore, it is necessary to continuously add glucose to the fermentation tank at this stage. The stages include the synthesis period of penicillin and the death period of the bacteria. Therefore, the penicillin fermentation process is a typical batch industrial process.

Pensim 2.0 is utilized to generate experimental data in this paper; it can simulate the process variables under various conditions and provide a standard platform for fault diagnosis of batch process. The core of the software is the improved Birol model based on the Bajpai mechanism, in which the changes of numerous variables can output in the form of data and simulate the changes of various parameters during the fermentation process [27]. In the penicillin fermentation model, there are 5 input variables to control the process parameters, 9 process variables in the process of bacterial synthesis and growth, and 5 quality variables can affect the yield of penicillin; the parameters are shown in Fig. 9. The temperature variation can be controlled using the flow rate of cold and hot water, which is controlled by the PID controller. The pH value can be controlled by the flow rate of the acid–base solution through the PID controller [28]. The parameters of the software initial conditions in the experiment can be set as Tables 1 and 2; Table 1 is the initial reaction condition setting range and Table 2 is the reaction process parameter setting range.

In the penicillin fermentation process, abnormal situations such as valve leaks or pump failures often occur. To simulate the actual production situation, Pensim2.0 software usually generates three common faults: air flow (Fault 1), agitation power (Fault 2), and bottom flow rate (Fault 3), each fault has two types of faults (step signal and ramp signal) in every fault type [29]. Set the three types

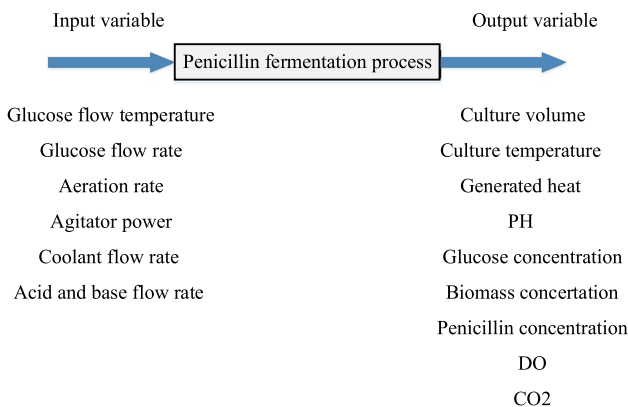


Fig. 9 Input and output of penicillin fermentation process

Table 1 Pensim2.0 initial condition setting

Variable	Unit	Ranges
Substrate concentration	G/L	14–18
Dissolved oxygen concentration	G/L	1–1.2
Bacteria concentration	G/L	0
Penicillin concentration	G/L	0
Reactor volume	L	100–104
Carbon dioxide concentration	G/L	0.5–1
Hydrogen ion concentration	MOL/L	10 ^{-5.1}
PH value		4.5–5.5
Temperature	K	295–301

of faults as small-amplitude step signals, the amplitude of which is -2% of the initial set point, and the fault leading into time is 50 h. To ensure the diversity of data, the initial values and parameters are randomly generated, and the normal state and the three types of faults are repeated 1000 times, 1000 batches of 4000 batches of fault data are generated. Add random noise with a mean value of 0 and a variance of 1 to imitate the actual data disturbed in production.

4.2 Model Simulation Experiments

This experiment uses Python 3.7 as the programming language and PyTorch 1.0 as the deep learning framework. The experimental data are encapsulated into the data loading function by PyTorch, set the first 800 of each type as the training data, and the last 200 as the test data. The batch_size parameter is set as 32, where the network training takes 32 for training every time.

In the obtained experimental data, each batch has 17 variables, which usually have different dimensions, different sizes, and different magnitudes. If you directly calculate the original data, the role of high-dimensional indicators in the comprehensive analysis will be highlighted, the role of low-value indicators will be weakened, and many orders of magnitude will affect the network training speed and effect. Therefore, it is necessary to standardize the

Table 2 Pensim2.0 parameter settings

Variable	Unit	Ranges
Ventilation rate	L/H	8–9
Stirring power	W	29–31
Base material flow rate	L/H	0.039–0.045
Bottom stream Temperature	K	295–296
PH value		4.5–5.5
Reactor temperature	K	297–298

data before training. In this paper, minmax normalization is used to perform a linear transformation on the original data. This transformation will not change the distribution of the data; the original data are mapped to the interval $[0,1]$. The calculation process as follows.

Transform the sequence x_1, x_2, \dots, x_n :

$$y_i = \frac{x_i - \min_{1 \leq j \leq n} \{x_j\}}{\max_{1 \leq j \leq n} \{x_j\} - \min_{1 \leq j \leq n} \{x_j\}} \quad (8)$$

The new sequence $y_1, y_2, \dots, y_n \in [0, 1]$ and dimensionless.

The standardized training group data are substituted into the established deep time series feature extraction model. The number of iterations Epoch is set as 200; the learning rate lr is set as 0.01. *Adam* is chosen as the optimization function, *CrossEntropyLoss* is the loss function for multi-classification, and *Relu* is chosen as the activation functions. The loss function during the training process and the accuracy of the training set are output; the results are shown in Fig. 10.

It can be seen from Fig. 10 that the deep time series extraction model obtained by the preliminary training does not perform well. The loss function shows that the loss function fluctuates near a larger value, and the accuracy of the training set is not stable, which oscillates around 80%. The accuracy rate is only 81.73%; it should be optimized.

4.3 Model Optimization

(1) Deep LSTM.

From the perspective of the basic model, the training performance of the model gradually converges, but

the effect is not good. One is that the convergence curve oscillates violently, and the accuracy of the training set is only about 80%. The reason for the poor effect may be the insufficient fitting ability of the network to the data set, which occurs the problem of under-fitting. The first thing to consider is the problem of network structure, and the most critical part is the LSTM in the deep time series feature extraction model, which is used to extract the time series features of the data. The pros and cons of the LSTM determine whether the model can extract useful time features, and then affect the diagnostic effect of the model. In the basic model, the structure of LSTM adopts the one-way, single-layer structure, and the number of hidden layer units is 64. Due to the insufficient fitting ability of the model, consider deepening the LSTM network structure, from a one-way two-layer structure to a two-way four-layer structure, and the number of hidden layer units is set to 128, then observe the training performance. The performance of the training set in deep LSTM model is shown in Fig. 11.

It can be seen from Fig. 11 that the deep time series extraction model obtained by the deep LSTM model does not perform well, the accuracy rate and loss function fluctuate greatly, and the convergence is slow. At the same time, the accuracy of the training model is close to 100%, but the performance accuracy of testing set has only 85.65%, and the model has over-fitting problem.

(2) LSTM with added Layer normalization (LN) and Dropout layer (LD-LSTM)

For solving the problems of slow model convergence and over-fitting, the deep model should be optimized, and the direction mainly includes the structure and the hyper-parameters of the deep learning model. In this paper, the

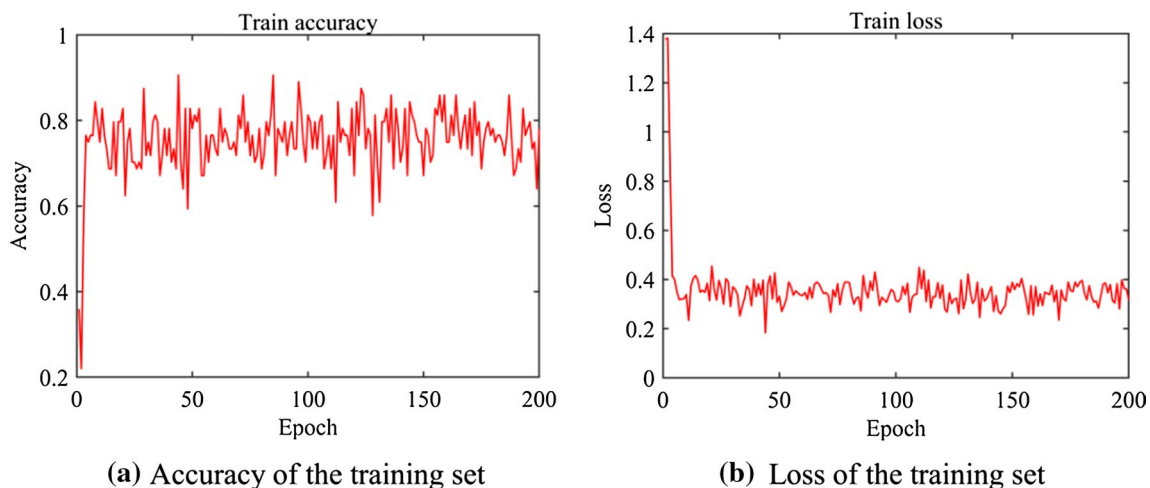


Fig. 10 The performance of training set

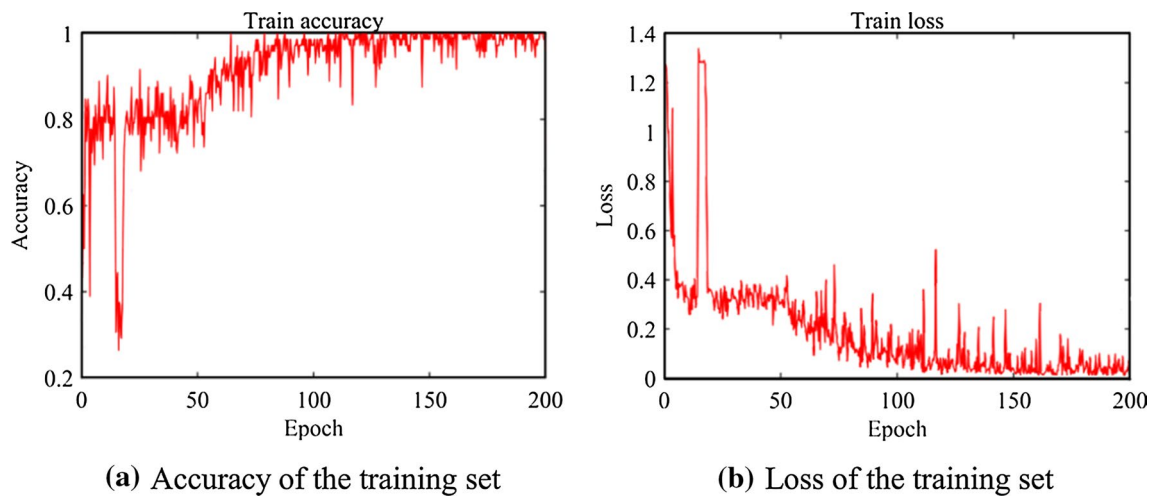


Fig. 11 The performance of training set in deep LSTM model

layer normalization (LN) and the dropout layer are added between the layers of the LSTM (LD-LSTM) to improve the performance. The LN normalizes the activation value of neurons, reduces network training time, speeds up convergence, and enhances network stability. The dropout operation randomly inactivates neurons to avoid excessive network capabilities and can effectively solve the problem of over-fitting. The performance of the training set of LD-LSTM model and fault diagnosis results are shown in Figs. 12 and 13.

Figure 12 shows the results of the accuracy and the loss of training set. It can be seen that the improved model using the LN and dropout layers has converged when it runs to 150 times. The convergence speed of the model has been significantly improved. Besides, the prediction results of testing

samples can be seen in Fig. 13, and the diagnosis accuracy rate is 98.87%, which explains that the improved model is more effective.

In order to visually see the effect of the optimization method, Fig. 14 shows the accuracy and loss value results with different algorithms. It can be seen that the deep LSTM and LD-LSTM has a better fitting effect than the model without optimization. The deep LSTM and LD-LSTM can finally reach a higher accuracy rate and the loss value that the convergence speed of LD-LSTM is faster. In summary, coupled with the over-fitting problem of the previous deep LSTM, the effectiveness of the proposed model in this paper can be preliminarily obtained.

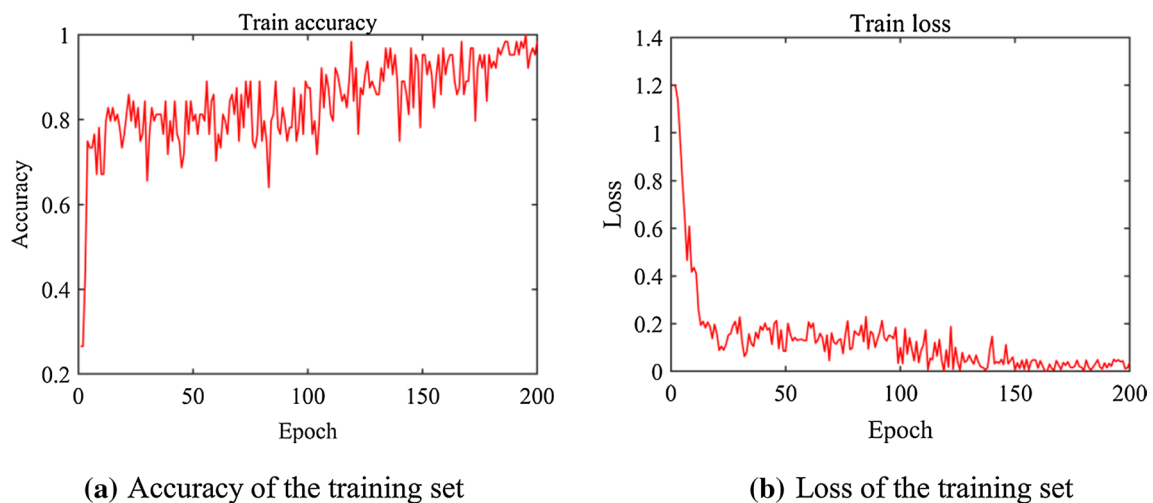


Fig. 12 The performance of training set of LD-LSTM model

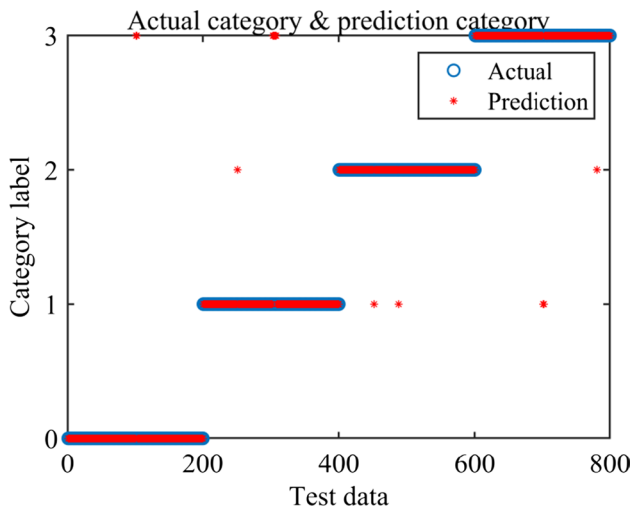


Fig. 13 Fault diagnosis performance of penicillin fermentation process

4.4 Model Stability

As we know, the deep learning model has a certain degree of randomness during training; the model obtained from each training is not the same. To fully explain the general performance of the model in this paper, we conduct 20 repeated experiments and record the results to observe the stability. The average of the 20 experiments is used to illustrate the stability of the model. The results of 20 experiments are shown in Fig. 15.

In Fig. 15, the red points represent the accuracy of the testing set, which is obtained from each training model. The blue straight line represents the average result of the 20 times experiments. It can be seen that the best accuracy

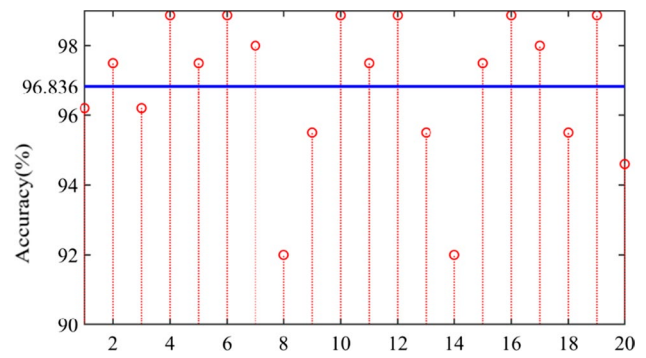
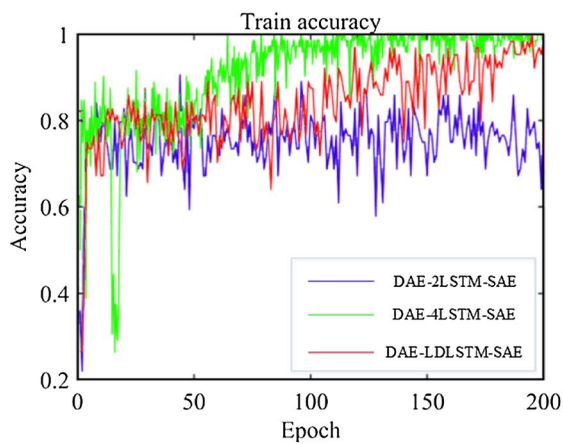


Fig. 15 Accuracy performance of repeated experiment model

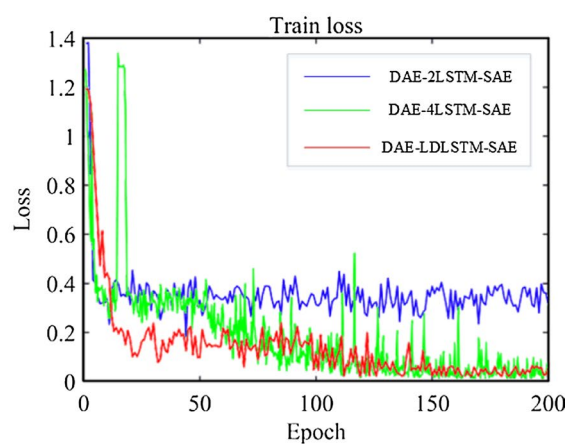
rate that the model can achieve is 98.87%, the lowest is 92.05%, and the average value can reach 96.84%. These indicators demonstrate that the model is effective in terms of fault diagnosis accuracy. However, it still has some unstable situation; we should optimize the model to improve the stability of fault diagnosis in the future.

4.5 Comparative Verification

An incipient fault diagnosis model for batch process is proposed based on deep time series feature extraction; the LN layer and the dropout layer are utilized to optimize the network structure. The effectiveness of the model is verified through the penicillin fermentation process. Firstly, the original data are preprocessed, map-minmax normalization is used for data processing in this paper, which can eliminate the influence of dimensions, speed up the model calculation speed, and increase model accuracy. Then, substitute the pre-processed data into three models for training. To investigate



(a) Accuracy of the training set



(b) Loss of the training set

Fig. 14 Performance before and after model optimization

the effectiveness of the proposed models, we have used six other models. The support vector machine (SVM) uses fixed constants by experiences and studies (C is 100 and γ is 0.5). The kernel extreme learning machine (KELM) parameter takes [0.5,100], and the data are flattened. The BP structure is set to a 5-layer fully connected network; the number of neurons in each layer is 800*17, 256, 128, 64, and 4, respectively. Besides, LSTM and deep LSTM models are utilized to compare. The structure parameters of different machine learning and deep learning are shown in Table 3. The average recognition training and prediction accuracies of the five models are presented in Table 4, where *Epoch*, *lr*, and other hyperparameters are consistent under the data preprocessing.

It can be seen from Table 4 that the traditional machine learning models like SVM and KELM have poor results with accuracy rates of only about 76%, while the classic BP network has an accuracy rate of only 72.56%. It shows that the typical network cannot effectively diagnose incipient faults of batch process. When the LSTM network is used for feature extraction without data noise reduction and deep feature extraction, the effect of the model is still not good. Both the training set and the test set have a fault recognition rate of about 85%. However, it indicates that the LSTM network has a better effect than the traditional shallow networks in time series problems.

When the deep LSTM with two-ways four-layers is used for training, the accuracy of the training set can reach 99.33%. However, due to the complexity of the model, over-fitting occurs and results in poor generalization ability; the accuracy rate is only 85.65% in test data. In this paper, the LN layer and dropout layer are added to the LSTM (LD-LSTM), which introduces the over-fitting and network standardization methods. The over-fitting phenomenon is effectively solved, and the generalization ability and robustness are improved. The DAE and SAE are added to the model to realize time series data purification and key feature extraction. The final accuracy rate has reached 98.87%, which is 13.22% higher than before the improvement. The

Table 4 Diagnostic accuracy of different models

Model	Training accuracy (%)	Prediction accuracy (%)
SVM	80.34	76.20
KELM	75.62	76.80
BP	70.34	72.56
LSTM	88.12	84.38
Deep LSTM	99.33	85.65
DAE + LD-LSTM + SAE	99.78	98.87

comparative experiment fully demonstrates the effectiveness and superiority of the proposed deep time-series feature extraction network.

5 Conclusions and Future Work

In this paper, we propose an incipient fault diagnosis model of batch process based on deep time series feature extraction. LSTM network as a basic model introduces DAE and SAE to make the network have denoising and deeper feature extraction abilities. Besides, LN and dropout layers are introduced to improve the network and accelerate the network convergence, which can solve the problems of over-fitting and improve the accuracy of diagnosis. The penicillin fermentation process is utilized as the experiment example; we conduct 20 repeated experiments and record the results to observe the stability. Finally, to investigate the effectiveness of the proposed models, five other models (SVM, KELM, BP, LSTM, and deep LSTM) are designed and evaluated. The proposed model with deep time series features extraction (LD-LSTM) performs best among the models used, which reflects the benefits of the proposed incipient fault diagnosis model. Currently, the proposed model is quite effective in diagnosing the incipient faults of batch process.

Table 3 The structure parameters of different machine learning and deep learning methods

Model	Layers type	Number of layers	Learning rate	Number of iterations	Ratio of training set to test set	The remaining
SVM	/	/	/	200	0.8:0.2	$C = 100 \gamma = 1$
KELM	/	/	/	200	0.8:0.2	[0.5,100]
BP	Fully connected layer	5	0.01	200	0.8:0.2	Batch_size = 32
LSTM	LSTM cell	2	0.01	200	0.8:0.2	Batch_size = 32
Deep LSTM	LSTM cell	4	0.01	200	0.8:0.2	Batch_size = 32
LD-LSTM	DAE + LD-LSTM cell + SAE	12	0.01	200	0.8:0.2	Batch_size = 32

Currently, the proposed model is quite effective in diagnosing the incipient faults of batch process. However, getting field data for validation of the approach is very difficult, but we generate the simulated data using a fed-batch penicillin fermentation process, then to analyze the incipient fault diagnosis problem of batch process. Studying real data is undoubtedly our next research task. Besides, our next research would be to improve the model stability, analyze specific failure causes, and predict fault in a few time steps in advance. More works will be contributed to promoting the widespread use of the proposed algorithm in the actual batch production process.

Acknowledgements This work is partially supported by China Post-doctoral Science Foundation (No.2020M673279), National Natural Science Foundation of China (NSFC) (No. 51675450), Sichuan Science and Technology Program (No. 2020JDTD0012), and MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 18YJC630255).

Data Availability Statement All data generated or analyzed during the study are included in the submitted paper or supplemental materials files

Compliance with Ethical Standards

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- Gao, X.J.; Xu, Z.D.; Li, Z., et al.: Batch process monitoring using multiway laplacian autoencoders. *Can. J. Chem. Eng.* **98**(6), 1269–1279 (2020)
- Franz, D.B.; Oscar, A.P.; Jakob, K.H.: Discrete-continuous dynamic simulation of plantwide batch process continuous dynamic simulation of plantwide batch process systems in MATLAB. *Chem. Eng. Res. Des.* **159**(7), 66–77 (2020)
- Zhu, J.L.; Yao, Y.; Gao, F.R.: Multiphase two-dimensional time-slice dynamic system for batch process monitoring. *J. Process Control* **85**(1), 184–198 (2020)
- Walid, A.; Abdelkader, K.; Noureddine, L.: Neural observer-based small fault detection and isolation for uncertain nonlinear systems. *Int. J. Adapt. Control Signal Process.* **34**(5), 677–702 (2020)
- Hussain, S.; Wang, X.G.; Ahmad, S., et al.: On a class of mixed EWMA-CUSUM median control charts for process monitoring. *Quality Reliab. Eng. Int.* **36**(3), 910–946 (2020)
- Nomikos, P.; MacGregor, J.: Multivariate statistical process control charts for batch monitoring of transesterification reactions for biodiesel production based on near-infrared spectroscopy. *Comput. Chem. Eng.* **94**, 343–353 (2016)
- Jiang, Q.; Yan, X.: Parallel PCA-KPCA for nonlinear process monitoring. *Control Eng Practice* **80**, 17–25 (2018)
- Zhang, H.; Qi, Y.; Wang, L., et al.: Fault detection and diagnosis of chemical process using enhanced KECA. *Chem. Intell. Lab. Syst.* **161**, 61–69 (2017)
- Zhang M, Wang R Q, Cai Z Y, et al. Phase partition and identification based on KECA and MSVM_FWA for multi-phase batch process fault diagnosis. *Transactions of the Institute of Measurement and Control*, First online, <https://doi.org/https://doi.org/10.1177/0142331220910885>.
- Li, H.M.; Huang, J.Y.; Ji, S.W.: Bearing fault diagnosis with a feature fusion method based on an ensemble convolutional neural network and deep neural network. *Sensors* **19**(9), 1–18 (2019)
- Zhang, C.L.; He, Y.G.; Du, B.L., et al.: Transformer fault diagnosis method using IoT based monitoring system and ensemble machine learning. *Fut. Gener. Comput. Syst.* **108**, 533–545 (2020)
- Hamideh, R.; Blue, J.; Claude, Y.: Automatic equipment fault fingerprint extraction for the fault diagnostic on the batch process data. *Appl. Soft Comput.* **68**(7), 972–989 (2018)
- Ji, J.J.; Qu, J.F.; Chai, Y., et al.: An algorithm for sensor fault diagnosis with EEMD-SVM[J]. *Trans. Instit. Meas. Control* **40**(6), 1746–1756 (2018)
- Shao, H.D.; Jiang, H.K.; Li, X.Q.: Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine. *Knowledge Based Syst.* **140**, 1–14 (2018)
- Zhai M , Xiang X , Zhang R , et al. Optical flow estimation using dual self-attention pyramid networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, pp (99):1–1.
- Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D* **404**(3), 132306 (2020)
- Wang, S.; Wang, P.L.: Fault detection method for batch process based on deep long short-term memory network and batch normalization. *J. Comput. Appl.* **39**(02), 370–375 (2019)
- Zhang, X.; Zou, Y.Y.; Li, S.Y., et al.: A weighted auto regressive LSTM based approach for chemical processes modeling. *Neurocomputing* **367**(20), 64–74 (2019)
- Pan, Y.T.; He, F.Z.; Yu, H.P.: A correlative denoising autoencoder to model social influence for Top-N recommender system. *Front. Comput. Sci.* **14**(3), 143301 (2020)
- Chow, J.K.; Su, Z.; Wu, J., et al.: Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inform.* **45**(8), 101–105 (2020)
- Kensuke, H.; Masahiro, W.: Application of the DAE approach to the nonlinear sloshing problem. *Nonlinear Dyn.* **99**(3), 2065–2081 (2020)
- Su, C.H.; Jiang, M.S.; Liang, J.Y., et al.: Damage assessments of composite under the environment with strong noise based on synchrosqueezing wavelet transform and stack autoencoder algorithm. *Measurement* **156**(5), 107587 (2020)
- Li, F.; Gui, Z.P.; Zhang, Z.Y., et al.: A hierarchical temporal attention-based LSTM encoder-decoder model for individual mobility prediction[J]. *Neurocomputing* **403**(25), 153–166 (2020)
- Erdenebayar, U.; Jong-Uk, P.; Kyoung-Joung, L.: Automatic detection of sleep-disordered breathing events using recurrent neural networks from an electrocardiogram signal. *Neural Comput. Appl.* **32**(9), 4733–4742 (2020)
- Kang S Q, Zhou Y, Wang Y J, et al. RUL prediction method of a rolling bearing based on improved SAE and Bi-LSTM[J/OL]. *Acta Automatica Sinica*:1–11[2020–05–18]. [https://doi.org/https://doi.org/10.16383/j.aas.c190796\(in Chinese\)](https://doi.org/https://doi.org/10.16383/j.aas.c190796(in Chinese)).
- Lyu, P.Y.; Chen, N.; Mao, S.J., et al.: LSTM based encoder-decoder for short-term predictions of gas concentration using multi-sensor fusion[J]. *Process Saf. Environ. Prot.* **137**, 93–105 (2020)
- Max, S.; Murat, K.: Monitoring batch processes with dynamic time warping and K-nearest neighbours. *Chem. Intell. Lab. Syst.* **183**, 102–112 (2018)
- Sun, W.; Meng, Y.; Ahmet, P., et al.: A method for multiphase batch process monitoring based on auto phase identification. *J. Process Control* **21**(4), 627–638 (2010)
- Li, K.Q.; Feng, J.: Grouping multi-rate sampling fault detection method for penicillin fermentation process. *Can. J. Chem. Eng.* **98**(6), 1319–1327 (2020)

