Process Safety and Environmental Protection 185 (2024) 1145–1159

Contents lists available at ScienceDirect

Process Safety and Environmental Protection

journal homepage: www.journals.elsevier.com/process-safety-and-environmental-protection

Dual-attention LSTM autoencoder for fault detection in industrial complex dynamic processes

Lei Zeng [a], Qiwen Jin [a], Zhiming Lin [a], Chenghang Zheng [a,b], Yingchun Wu [a], Xuecheng Wu [a,b,*], Xiang Gao [a]

[a] State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China
[b] Zhejiang Key Laboratory of Clean Energy and Carbon Neutrality, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Keywords:
Complex dynamic process
Fault detection
Autoencoder
LSTM networks
Attention mechanism

ABSTRACT

Complex dynamic characteristics resulting from multi-system coupling and closed-loop control are ubiquitous in modern industrial process data, presenting significant challenges for process fault detection. However, conventional data-driven fault detection methods assume the data to be static or slightly dynamic. Addressing the complex dynamic characteristics and nonlinearity inherent in industrial processes, this paper proposes a dual-attention long short-term memory autoencoder (DALSTM-AE) for fault detection in dynamic processes. Long short-term memory (LSTM) and autoencoder (AE) are combined into a special encoder-decoder LSTM architecture to learn both dynamic features and deep representations of variables in an unsupervised manner. Then, a dual-attention module is embedded in the decoder to properly learn the temporal dependencies associated with long input sequences and retain the most critical information. In addition, based on the reconstruction results of the DALSTM-AE model, two monitoring statistics are designed for fault detection. Finally, the effectiveness and superiority of the proposed method are fully demonstrated through case studies on a numerical simulation example, the Tennessee Eastman (TE) benchmark process, and practical coal pulverizing systems in power plants.

1. Introduction

The importance of industrial process monitoring in ensuring the safety and reliability of equipment operation has gained widespread recognition due to the increasing complexity and magnitude of contemporary industrial systems (Zhang and Zhao, 2022). Moreover, owing to the widespread advancement and application of advanced sensor technologies, data-driven process monitoring methods are facilitated by the abundance of historical data (Wang et al., 2022). Compared to model-based and experiential knowledge-based methods, data-driven detection methods offer enhanced flexibility, simpler implementation, and reduced reliance on physical mechanisms and prior knowledge (van de Sand et al., 2021). Therefore, extensive research has been conducted on data-driven methods for industrial process monitoring and fault detection.

Traditional data-driven fault detection methods include multivariate statistical process monitoring (MSPM) methods, slow feature analysis (SFA), and artificial neural networks (ANN). Common MSPM methods, such as independent component analysis (ICA), principal component analysis (PCA), and partial least squares (PLS), can linearly project the original data into a low-dimensional space for feature extraction (de de de Carvalho Michalski and Martha de Souza, 2022; Gao et al., 2022; Zhu et al., 2022). Researchers have also explored the interconnections between risk assessment and data-driven fault diagnosis, reviewing the safety framework for the process industry (Arunthavanathan et al., 2021; Deng et al., 2023; Liu et al., 2021b). Although these conventional methods do improve the detection performance to some extent, they typically assume that data samples are independent of each other and neglect the temporal correlation within the data.

In practical industrial processes, the resulting data typically exhibits dynamic characteristics due to fluctuations in raw materials, multi-system coupling, and the implementation of closed-loop feedback control (Zheng et al., 2022). Furthermore, the coupling of complex systems results in diverse temporal correlations among different variables, thereby leading to intricate dynamic characteristics. Several process monitoring methods that consider temporal correlation and dynamic characteristics have been proposed, achieving favorable fault detection performance. Dynamic extensions of MSPM methods, such as dynamic

* Corresponding author at: State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China.
E-mail address: wuxch@zju.edu.cn (X. Wu).

https://doi.org/10.1016/j.psep.2024.02.042
Received 7 August 2023; Received in revised form 18 January 2024; Accepted 14 February 2024
Available online 17 February 2024
0957-5820/

PCA (DPCA), dynamic ICA (DICA), and dynamic PLS (DPLS), augment the process data matrix with multiple historical samples and subsequently apply the traditional MSPM model directly to the augmented matrix for monitoring (Kong et al., 2022; Liu et al., 2021a; Tao et al., 2020). However, the dimensions and parameters of the augmented matrix increase dramatically with the degree of dynamics, rendering these methods ineffective in extracting process dynamic changes. Other dynamic approaches, including dynamic inner principal component analysis (DiPCA), dynamic Bayesian network (DBN), and recursive exponential slow feature analysis (ESFA), have been proposed to extract dynamic information from process data (Dong and Qin, 2018; Meng et al., 2023; Yu and Zhao, 2019). Nonetheless, when confronted with extensive volumes of intricate industrial data, these methods tend to exhibit shallow dynamic feature representations, making it challenging to effectively capture potential dynamic features from long time series data and impeding the attainment of satisfactory fault detection results in dynamic processes.

In recent decades, the rapid advancement of artificial intelligence has led to the emergence of deep learning, which shows significant potential for process safety monitoring and fault detection tasks. Deep neural networks, such as convolutional neural networks (CNN), deep belief networks (DBN), and autoencoder (AE) have been successfully applied to fault detection and fault diagnosis due to their ability to adaptively capture representative information and learn features from input data (Qian et al., 2021; Wang et al., 2020; Yu and Yan, 2021; Zeng et al., 2023; Zhang and Zhao, 2022; Zhang et al., 2020; Zhou et al., 2020). To handle the dynamic characteristics of industrial process data, researchers have extensively investigated deep learning methods capable of handling temporal correlation (Lin et al., 2022; Zhu et al., 2019). Particularly, long short-term memory (LSTM) is widely used for fault detection and safety assessment in dynamic processes, as it can adaptively learn the dynamic patterns in time series data using its internal nonlinear gating units (Osarogiagbon et al., 2020; Yin et al., 2021). Recently, networks combining AE and LSTM have achieved favorable applications in extracting dynamic features (Zhang and Qiu, 2022). Aghaee et al. (Aghaee et al., 2023) developed an unsupervised multilayer model combining LSTM and AE networks to improve fault detection accuracy in industrial pharmaceutical processes. Amini et al. (Amini and Zhu, 2022) designed a source-aware AE fault detection network based on bidirectional LSTM architecture, which is capable of detecting unseen faults and dealing with imbalanced datasets. In addition, attention mechanism (AM) has gained widespread attention due to its ability to focus on key features. Xiang et al. (Xiang et al., 2021) embedded an AM in the CNN-LSTM model to improve wind turbine fault detection accuracy. Yang et al. (Yang et al., 2022) integrated AM and deep multiple autoencoders to develop a dynamic domain adaptation method for rotary machine fault diagnosis under different operating conditions. However, existing methods often have limitations in capturing complex and uneven dynamic characteristics of industrial data, especially when the degree of dynamics is high. The network's ability to convey information regarding the dynamics of past time steps is diminished when confronted with long sequence inputs (Sehovac and Grolinger, 2020). Therefore, it is essential to establish an adaptive monitoring model specifically designed for complex multivariate dynamic processes.

In this work, we develop a dual-attention long short-term memory autoencoder (DALSTM-AE) to extract dynamic nonlinear features efficiently. The DALSTM-AE combines LSTM and AE networks, capturing long-term temporal features and potential nonlinear representations of dynamic process variables. Additionally, a dual-attention module is integrated into the model to adaptively assign weights, enabling the decoder to flexibly leverage the most crucial information from the input sequence during the decoding process. The proposed method can effectively handle long sequence data containing complex dynamic information. The contributions of this paper are summarized as follows:

a) DALSTM-AE integrates LSTM and AE networks into an encoder-decoder LSTM architecture that captures the long-term temporal features and potential nonlinear representations of dynamic process variables.
b) Dual attention module is introduced to enhance the decoder's ability to capture different dynamic features of variables, which can effectively solve the information loss problem induced by overly complex and long sequences.
c) The proposed method is based on an end-to-end unsupervised learning paradigm, directly obtaining the system state using the original unlabeled samples of system variables and thereby acquiring optimal features for fault detection and risk warning.
d) The performance of the proposed method for fault detection and process safety monitoring is verified by studies on a numerical simulation example, the Tennessee Eastman (TE) benchmark process, and practical coal pulverizing systems in power plants.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts and fundamentals of AE and LSTM. Section 3 presents the proposed DALSTM-AE algorithm and explains its details. Then the dynamic fault detection procedure based on DALSTM-AE is described. Section 4 illustrates the applicability and effectiveness of the proposed method through a numerical case, TE benchmark process, and two practical industrial cases. Finally, the conclusions of this paper are given in Section 5.

2. Preliminaries

2.1. Autoencoder

Autoencoder (AE) is a special type of deep learning network that has gained widespread interest and application in feature learning, data compression, and fault detection. A typical AE comprises an encoder and a decoder, which collaborate to learn the low-dimensional feature representation, as shown in Fig. 1. The encoder is responsible for mapping the original input data into hidden feature representations. Subsequently, the decoder takes these hidden features as input and endeavors to reconstruct the input data at the output layer. Typically, the number of neural units of the input layer and output layer is the same. Therefore, AE can be trained in an unsupervised manner without requiring additional labels or supervision. For a given input data $\boldsymbol{x} \in \mathbb{R}^m$, the encoding and decoding process can be calculated as

$$\boldsymbol{h} = f(\boldsymbol{x}) = s(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{1}$$

$$\widehat{\boldsymbol{x}} = g(\boldsymbol{h}) = s(\boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}') \tag{2}$$

where $s(\cdot)$ represents the nonlinear activation function. $\boldsymbol{h}$ denotes the

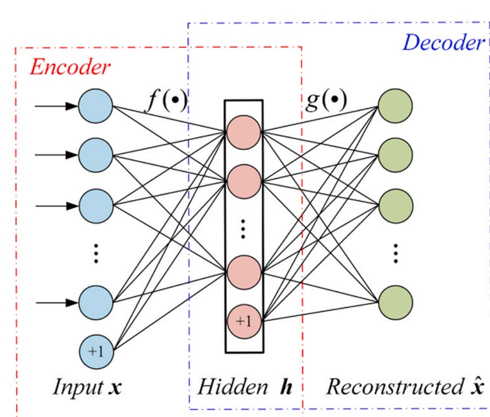

Fig. 1. Structure diagram of AE network.

encoded feature representation, and $\hat{x}$ is the reconstruction of the input data. $\{W, W'\}$ and $\{b, b'\}$ are weight matrices and corresponding bias vectors, respectively.

The loss function L_{AE} and the objective function J can be expressed as follows:

$$L_{AE} = \frac{1}{2}\|x_i - \hat{x}_i\|^2 \tag{3}$$

$$J = \frac{1}{n}\sum_{i=1}^{n} L_{AE}(x_i, \hat{x}_i) \tag{4}$$

where n denotes the number of samples.

2.2. LSTM neural network

Recurrent neural network (RNN) is a deep learning network specifically designed for processing sequential data or time series data by computing recursively along the sequence evolution or time. RNN is characterized by memory ability, parameter sharing, and Turing completeness, making it suitable for nonlinear feature learning of time series problems. The RNN model utilizes information from prior inputs to influence the current input and output, enabling the network to retain memory of past content. However, RNN uses the backpropagation through time (BPTT) algorithm to determine the gradients, making it tend to run into problems of gradients vanishing and exploding in the training process. Therefore, RNN is not suitable for dealing with "long memory" problems.

LSTM is a variant of RNN with a more elaborate internal "gating" mechanism, which allows it to remember valid information for a long time and forget invalid information. LSTM is effective for feature extraction when dealing with time series data with substantial dynamic relationships between variables. The LSTM hidden layer consists of multiple LSTM neurons, with the same number of neurons as the time step. The structure of a single LSTM cell is shown in Fig. 2. The hidden layer inputs include internal self-looping memory units c_{t-1} in addition to external inputs x_t and recurrent outputs h_{t-1}. Inside the LSTM are the gating units that control the flow of information, including input gate i_t, forget gate f_t, and output gate o_t. The input gate controls the calculation of new states and determines what information should be updated into the memory unit. The forget gate controls how much is forgotten in the current computation. And the output gate controls how much of the current output depends on the cell state c_t. Given an input time series $X = \{x_1, x_2, ., x_t\}$, where $x_t \in R^m$ represents an m-dimensional vector at time-instance t, the LSTM neuron memory unit c_t and output $h_t$are calculated as follows:

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i h_{t-1} + \mathbf{b}_i) \tag{5}$$

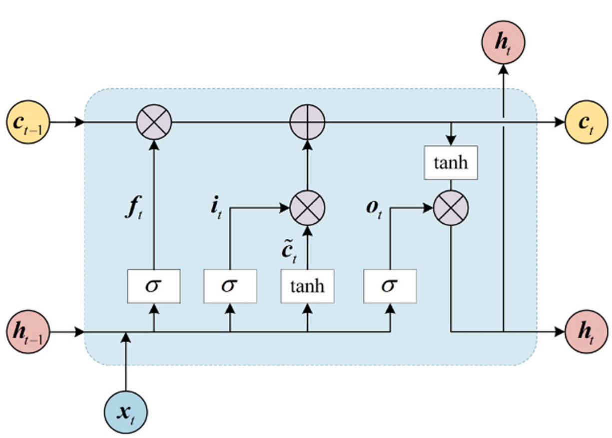

Fig. 2. Basic architecture of LSTM cell.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{6}$$

$$\begin{aligned} \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \end{aligned} \tag{7}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{8}$$

$$h_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{9}$$

where $\tilde{c}_t$ denotes the updated new state. W and U represent the appropriate weight matrices, and b represents the bias vector. $\sigma(\cdot)$ denotes the sigmoid activation functions, and $\odot$ represents the element-wise multiplication.

3. Methodology

3.1. Problem statement and motivation

In modern industrial processes, the application of massive closed-loop control compensates for system disturbances and guarantees economic benefits. However, the closed loops formed are coupled with each other, and the propagation path of the influence is complex. Consequently, the process data exhibits different dynamic characteristics, leading to the collected samples no longer being independent but instead forming multivariate time series. Time series data contain more complex coupling information and features than single time point data, posing great challenges in feature learning. The LSTM autoencoder (LSTM-AE) is an encoder-decoder LSTM architecture integrated by LSTM and AE networks, which allows the model to encode the original inputs as fixed-dimension vectors and decode them into target sequences (Srivastava et al., 2015). LSTM is capable of effectively capturing the dynamic features and nonlinear relationships among variables. The AE neural network, on the other hand, demonstrates exceptional abilities in learning complex nonlinear information among time series variables. The LSTM-AE combining the two networks can extract nonlinear feature representations and learn the dynamic characteristics of the input data sequence, which has been proven to be suitable for industrial dynamic process monitoring (Chen et al., 2021; Li et al., 2020). As shown in Fig. 3, the conventional LSTM-AE model consists of two parts: the encoder LSTM and the decoder LSTM.

In practice, the temporal correlation of different variables in multivariate time series data from industrial processes is different, resulting in complex and heterogeneous dynamic characteristics.

The dynamic degree of variables is denoted by the number of time lags, defined as s. The time lag steps of the variables are related to the process operation mechanism and can be determined by analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the variables. To ensure that the time series dataset used for model training encompasses dynamic information for all variables, the length of the dynamic time window L (i.e., the order of dynamic process) needs to be greater than the maximum time-lag step $s_{\max}$ in the variables. This allows us to more comprehensively capture the dynamic features and dependencies among variables over time. Therefore, it is assumed that $L = s_{\max} + 1$ without loss of generality. Industrial production processes often encompass certain highly dynamic variables, leading to long and complex time series datasets. In this case, the LSTM-AE model experiences information loss while encoding fixed-dimension vectors, which restricts the decoder's access to the information provided by the input and results in underperformance when dealing with long sequence data. The model also faces the problem of increased computational burden. Therefore, it is necessary to introduce a weight assignment method that allows the model to adaptively select the most important information from long time series data based on the current moment.

Therefore, a dual-attention LSTM autoencoder (DALSTM-AE) algorithm and its monitoring scheme are presented in this work to solve the complex dynamic process fault detection problem. A detailed

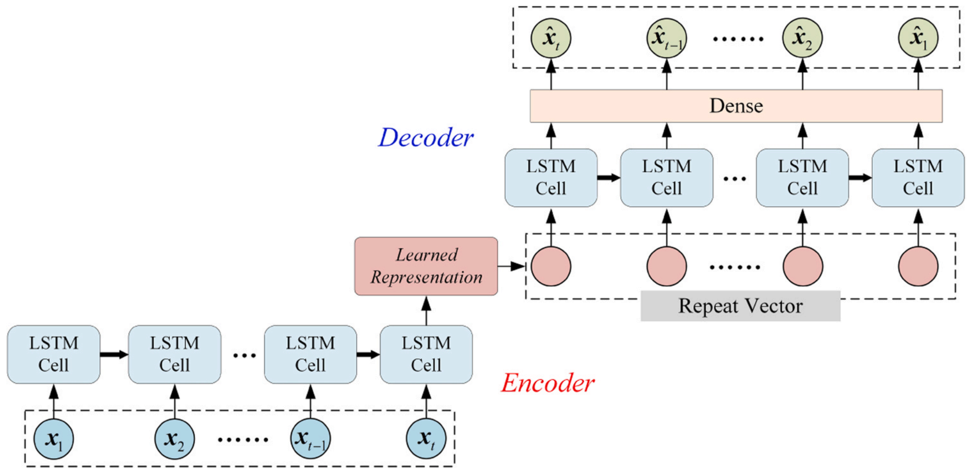

Fig. 3. The conventional LSTM-AE architecture.

explanation of the DALSTM-AE model and its corresponding monitoring procedure are presented in the subsequent subsections.

3.2. The proposed DALSTM-AE model

The attention mechanism (AM) has been widely used in deep learning as a technique inspired by cognitive attention. The essence of data processing by AM is to construct an attention matrix by calculating the probability distribution of attention, thus assigning more weight to important information and highlighting the impact of key features on the results.

As the length and complexity of the input sequence increase in conventional LSTM-AE architecture, the model's ability to learn dynamic features decreases, leading to a decline in performance. Therefore, a dual attention module with input global attention and self-attention is introduced to improve the performance of the LSTM-AE model for industrial fault detection. The schematic diagram of DALSTM-AE is illustrated in Fig. 4.

The input global attention is to permit the decoder to utilize the most critical parts of the input sequence flexibly, by weighting the combination of all encoded input vectors and attributing the highest weight to the most relevant vectors. Given an input time series $X = \{x_1, x_2, ., x_L\}$ with length L, where $x_t \in \mathbb{R}^m$ represents an m-dimensional vector at time-instance t, the proposed DALSTM-AE model's encoder maps the input into a hidden state vector at each time step. Particularly, the hidden state $h_s^L$and cell state c_s^L generated by the last LSTM cell of the encoder are taken as the initial states of the decoder. In the attention

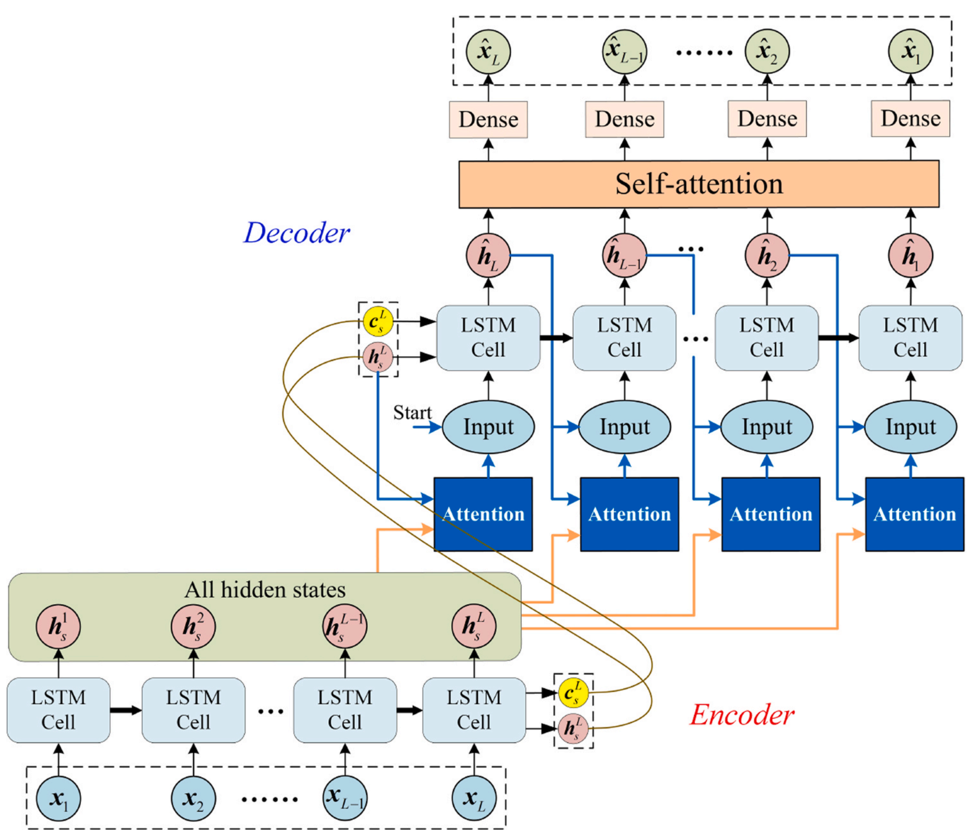

Fig. 4. Network structure diagram of DALSTM-AE.

layer, the encoder's hidden states at all consecutive time steps are utilized to enhance the decoder's ability to capture the dynamic features of diverse variables. The calculation process can be defined as follows:

$$\mathrm{score}(\widehat{\boldsymbol{h}}_t, \overline{\boldsymbol{h}}_s) = \widehat{\boldsymbol{h}}_t^{\top} \overline{\boldsymbol{h}}_s \tag{10}$$

$$\boldsymbol{\alpha}_{ts} = softmax(score(\widehat{\boldsymbol{h}}_t, \overline{\boldsymbol{h}}_s)) \tag{11}$$

$$F(\widehat{\boldsymbol{h}}_t) = \sum_s \boldsymbol{\alpha}_{ts} \overline{\boldsymbol{h}}_s \tag{12}$$

where $\widehat{\boldsymbol{h}}_t$ is the current target hidden state of the decoder, and $\overline{\boldsymbol{h}}_s$ is each hidden state of the encoder. $\boldsymbol{\alpha}_{ts}$ denotes the attention weight, and $F(\widehat{\boldsymbol{h}}_t)$ represents the output vector after attention. In the decoder, the target hidden state$\widehat{\boldsymbol{h}}_L$ is obtained from the first decoder LSTM cell, whose input is the concatenation of the initial attention output $F(\boldsymbol{h}_s^L)$and a fixed start vector. Then, the first hidden state$\widehat{\boldsymbol{h}}_L$ will be taken as the target query vector for the global attention layer to obtain the output $F(\widehat{\boldsymbol{h}}_L)$. More generally, the output $F(\widehat{\boldsymbol{h}}_t)$ is concatenated with the decoder's hidden state $\widehat{\boldsymbol{h}}_t$ to generate the input for the LSTM cell at the next time step.

The reconstructed time series data obtained by the decoder is in reverse order, which means that the decoding operation is recursive in the inverse time direction during training. Therefore, a self-attention layer is used before the output layer to adaptively adjust the temporal weights and select more critical dynamic features to fit the objective function. The process of self-attention is as follows:

$$\mathbf{Q} = \mathbf{W}_q \widehat{\boldsymbol{h}}, \mathbf{K} = \mathbf{W}_k \widehat{\boldsymbol{h}}, \mathbf{V} = \mathbf{W}_v \widehat{\boldsymbol{h}} \tag{13}$$

$$F_s(\widehat{\boldsymbol{h}}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{14}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ represent the trainable parameter matrices, and $\widehat{\boldsymbol{h}}$ denotes the feature outputs of the decoder LSTM layer. d_k is the dimension of the key vector, and $F_s(\widehat{\boldsymbol{h}})$means the output vector after self-attention.

The last layer of the decoder consists of multiple dense layers, also known as fully connected layers or linear layers. The same dense (fully connected) operations are applied to every time step of the input 3D feature tensor. In our work, the last layer receives the output features from the self-attention layer and independently maps the features for each time step to the corresponding reconstructed output. The target sequence reconstructed by the decoder is the reverse order of the input sequence. The reconstructed vector $\widehat{\boldsymbol{x}}_t$ at time point t can be calculated as

$$\widehat{\boldsymbol{x}}_t = \boldsymbol{w}_t F_s(\widehat{\boldsymbol{h}}_t) + \boldsymbol{b}_t \tag{15}$$

where $\boldsymbol{w}_t$ and $\boldsymbol{b}_t$ denote the weight matrix and bias, respectively.

The DALSTM-AE model is trained by minimizing the error between the reconstructed data and the original input sequences. The loss function is expressed in mean square error (MSE)

$$Loss = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2. \tag{16}$$

3.3. Process fault detection scheme

The flowchart of fault detection based on DALSTM-AE is illustrated

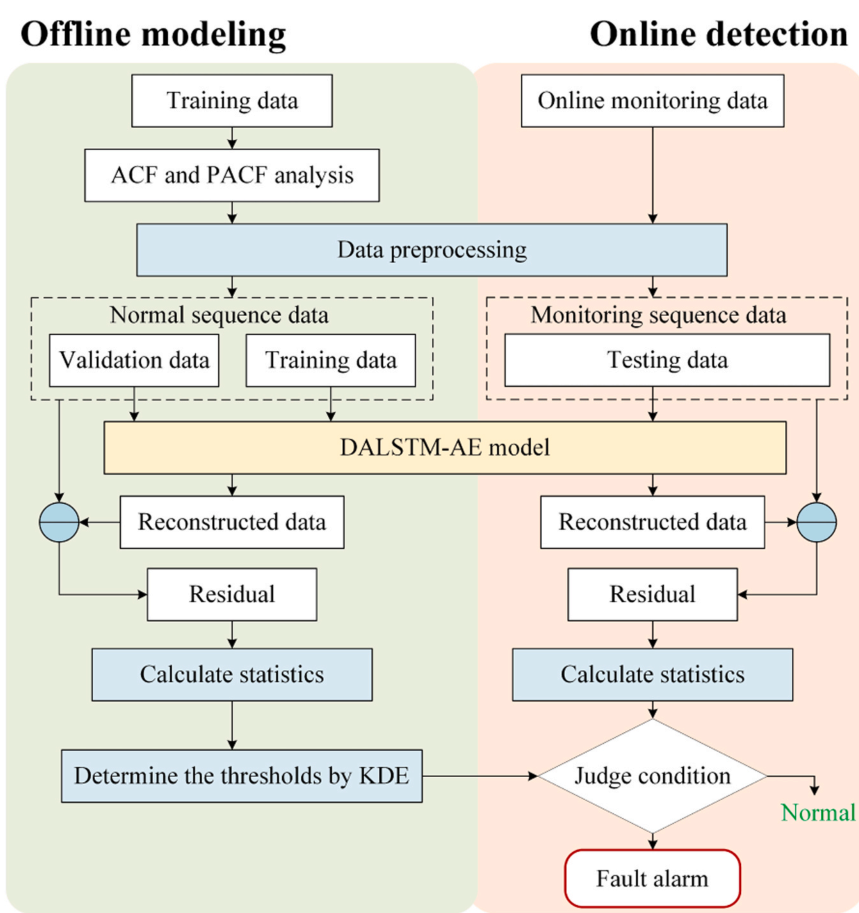

Fig. 5. Flowchart of fault detection based on DALSTM-AE.

in Fig. 5, consisting of two modules: offline modeling and online detection. The purpose of offline modeling is to fully learn the dynamic and internal characteristics of the normal system and set appropriate control limits for statistical indicators. In online fault detection, the well-trained model is utilized to obtain real-time statistical indicators and determine whether the process is abnormal. The specific application details of DALSTM-AE are as follows.

3.3.1. Offline modeling

Original process data $\mathbf{X} \in \mathbb{R}^{n\times m}$ is collected under the normal operation condition, where n is the number of samples and m is the number of process variables. The ACF and PACF of the original variables are calculated to analyze the dynamic correlation degree of the variables. The length of the time window L (i.e., the length of the sequence) is determined by the maximum time lag observed in the variables. The original data is standardized by the z-score normalization to a matrix with zero mean and unit variance. Finally, the processed data matrix is serialized and transformed into a time series $\mathbf{X} \in \mathbb{R}^{(n-L+1)\times L\times m}$ using the sliding window technique, as shown in Fig. 6.

The DALSTM-AE model is trained on the preprocessed training dataset to extract the dynamic and nonlinear features of the long sequences. The reconstruction error of the normal condition is obtained by importing all normal data into the trained model. Anomalies can cause significant changes in the dynamic and nonlinear relationships between variables, leading to substantial reconstruction errors. Therefore, it is possible to determine whether a failure has occurred by designing reasonable statistical indicators. In this work, the MD^2 and RE^2 statistical indicators are used to monitor the abnormal changes in the dynamic process. The MD^2 indicator is based on the Mahalanobis distance, reflecting the degree of deviation of the current reconstruction error. The MD_i^2 of the ith sequence sample can be calculated as

$$MD_{i,t}^2 = (\Delta \boldsymbol{x}_{i,t} - \mu_t)^T \boldsymbol{\Sigma}^{-1} (\Delta \boldsymbol{x}_{i,t} - \mu_t) \tag{17}$$

$$\Delta \boldsymbol{x}_{i,t} = \boldsymbol{x}_{i,t} - \widehat{\boldsymbol{x}}_{i,t} \tag{18}$$

$$MD_i^2 = \max\left[MD_{i,1}^2, MD_{i,2}^2, \dots, MD_{i,L}^2\right] \tag{19}$$

where $MD_{i,t}^2$ represents the statistical indicator of the sample at time step t, and $\Delta x_{i,t}$ is the residual space. $\boldsymbol{\Sigma}^{-1}$ represents the inverse covariance matrix of normal samples and μ_t denotes the mean of the reconstruction error for normal samples.

The RE^2 is defined as the Euclidean norm of the residual vector, which can be calculated as

$$RE_i^2 = \max\left[\left\|\boldsymbol{x}_{i,1} - \widehat{\boldsymbol{x}}_{i,1}\right\|^2, \left\|\boldsymbol{x}_{i,2} - \widehat{\boldsymbol{x}}_{i,2}\right\|^2, \dots, \left\|\boldsymbol{x}_{i,t} - \widehat{\boldsymbol{x}}_{i,t}\right\|^2\right]. \tag{20}$$

When the monitoring statistical indicators of the normal sequence are obtained, kernel density estimation (KDE) is used to calculate the corresponding control limits. The KDE method is a nonparametric statistical technique that does not require any prior assumptions about the data distribution. Take the MD^2 as an example. Assuming that a total of p MD^2 statistics are calculated for normal sequence data, the probability density function $\widehat{f}(MD^2)$ can be expressed as follows (Li et al., 2021):

$$\widehat{f}\left(MD^2\right) = \frac{1}{pB}\sum_{k=1}^{p} K\left(\frac{MD^2 - MD_k^2}{B}\right) \tag{21}$$

where MD_k^2 is monitoring indicator of the kth data sample. B represents the bandwidth, calculated by the empirical equation $B = p^{-0.2}$ (Scott, 2015). $K(\cdot)$ represents the kernel function that satisfies $\int_{-\infty}^{\infty} K(x)dx = 1$. The Gaussian kernel function is adopted in this work:

$$K\left(\frac{MD^2 - MD_k^2}{B}\right) = \frac{1}{B\sqrt{2\pi}}\exp\left(-\frac{\left(MD^2 - MD_k^2\right)^2}{2B^2}\right). \tag{22}$$

$\widehat{f}(MD^2)$ represents the probability density distribution of MD^2 statistic. The probability of samples falling within a confidence interval called confidence level, usually expressed as a percentage. Given a certain confidence level of α, the upper limit of the confidence interval, selected as the control limit MD_θ^2, can be calculated by the cumulative density function (Chen et al., 2020):

$$\alpha = P\left(MD^2 < MD_\theta^2\right) = \int_{-\infty}^{MD_\theta^2} \widehat{f}\left(MD^2\right) d\left(MD^2\right). \tag{23}$$

The control limit for the RE^2 statistic can be calculated similarly.

3.3.2. Online detection

The fault test data is collected in real time. The same data preprocessing procedure is performed as on the training data to obtain the normalized and serialized test dataset X_f. Then X_f is fed into the well-trained DALSTM-AE model to obtain the corresponding reconstructed sequence $\widehat{X}_f$. The statistical indicators MD_f^2 and RE_f^2 of the test data are calculated by Eqs. (19) and (20), respectively. The corresponding control limits MD_θ^2 andRE_θ^2 are calculated by the KDE method. If the current statistical indicator is greater than the control limit, the process is determined to be abnormal. Otherwise, the process is considered normal. Two common evaluation metrics, fault detection rate (FDR) and false alarm rate (FAR), are used to evaluate the performance of the proposed method. Specifically, the FDR and FAR are defined as follows:

$$\text{FDR} = \frac{\text{number of fault samples detected}}{\text{total samples (faulty)}} \tag{24}$$

$$\text{FAR} = \frac{\text{number of normal samples detected as fault}}{\text{total samples (normal)}}. \tag{25}$$

Additionally, the overall FDR and FAR of two statistical indicators

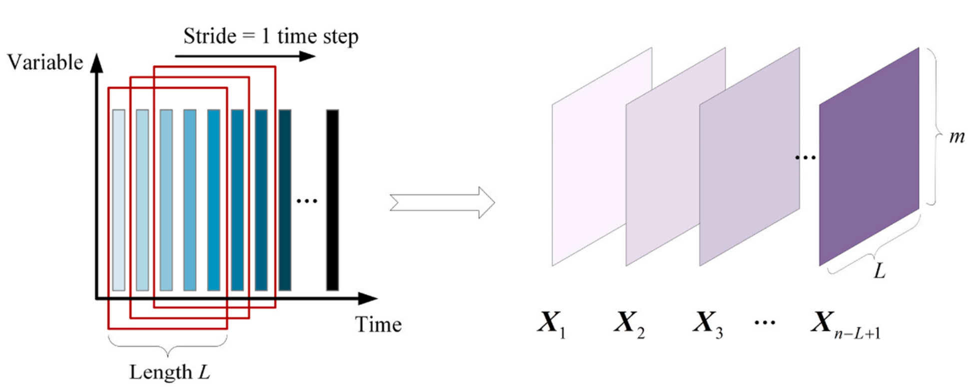

Fig. 6. Schematic of data serialization.

are introduced to facilitate a more comprehensive assessment. Specifically, we integrate two monitoring indicators to obtain the overall discrimination (OD) metrics: if neither RE^2 nor MD^2 exceeds the control limits, the test sample is considered normal. The detection logic satisfies.

$RE_f^2 \le RE_\theta^2$ and $MD_f^2 \le MD_\theta^2 \Rightarrow$fault free, otherwise faulty.

4. Case studies

In this section, the effectiveness and superiority of DALSTM-AE are illustrated through three case studies. The experiments in this study are conducted in Python 3.7, using Keras 2.3.1 and Tensorflow 2.1.0, on a computer with an i5–9400 F CPU @ 2.90 GHz and 16 GB RAM.

4.1. Simulation case

The numerical simulation case simulates a complex dynamic process where x_{k+1} is generated from a 5th-order vector autoregressive process and y_k is generated from a latent variable model.

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{x}_{k-1} + \boldsymbol{C}\boldsymbol{x}_{k-2} + \boldsymbol{D}\boldsymbol{x}_{k-3} + \boldsymbol{E}\boldsymbol{x}_{k-4} + \boldsymbol{F}\boldsymbol{u}_k \tag{26}$$

$$\boldsymbol{y}_k = \boldsymbol{G}\boldsymbol{x}_k + \boldsymbol{e}_k + \boldsymbol{\mu}_k \tag{27}$$

$$\boldsymbol{A} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.67 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.67 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{E} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.33 \end{bmatrix}$$

$$\boldsymbol{F} = \begin{bmatrix} 0.66 & 1.55 \\ 1.97 & 2.38 \\ 4.32 & -0.73 \\ -2.64 & 3.26 \\ 1.16 & 2.01 \end{bmatrix} \boldsymbol{G} = \begin{bmatrix} -0.67 & -2.66 & 0.61 & 1.74 & -0.5 \\ 2.51 & -1.28 & 0.25 & -1.4 & 0.04 \\ -0.39 & 3.27 & -1.5 & 0.76 & 1.08 \\ -1.33 & 1.43 & 1.26 & -0.93 & 0.22 \\ 0.45 & 0.38 & -0.32 & 0.67 & 0.22 \end{bmatrix}$$

where process input $\boldsymbol{u}_k \in \mathbb{R}^2 \sim N(0,\ 0.2^2)$, and noise $\boldsymbol{e}_k \in \mathbb{R}^5 \sim N(0,\ 0.1^2)$. $\boldsymbol{x}_k \in \mathbb{R}^5$ and $\boldsymbol{y}_k \in \mathbb{R}^5$are state and process output, respectively. Seven process variables (five outputs and two inputs) are measured as simulated dynamic process data. The vector $\boldsymbol{\mu}_k \in \mathbb{R}^5$ represents the operating center of the process output, whose value is assumed to be **1** for normal process operation (Jeng, 2010).

To construct the detection model, a dataset comprising 5000 normal samples is generated, with 4000 used for training and 1000 for testing. Two types of faults are simulated for the validation of the detection methods, each containing 1000 samples. Fault 1 directly changes the value of the process output. Fault 2 alters the dynamic characteristics of the variables by changing the coefficient matrix, which is difficult to be detected by general methods. The specific fault cases are designed as follows:

Fault 1: The process output variable $y_{3,k}$ is multiplied by 1.8 from time point $k = 501$.

Fault 2: A process fault is introduced such that the system matrix **C** changes to –**C** from time point $k = 501$.

The high-order autoregression makes simple numerical simulation data have relatively complex and uneven dynamic features. The ACF and PACF analyses for five variables of the process state outputs $\boldsymbol{x}_k$ are illustrated in Fig. 7. It can be seen that each variable has different degrees of autocorrelation and different time lags of correlation. Among them, variable 5 exhibits the largest time-lag step, leading to an extended time dimension in the serialized sequence data.

The time lag $s_{\max}$ is set to 5 since this process is a 5th-order autoregressive system, which is consistent with the results of the analysis performed on the ACF and PACF of the state variables. Consequently, the normalized data is serialized through a sliding window of length $L = s_{\max} + 1$. Hyperparameter optimization is performed using a grid search method based on cross-validation. The number of hidden units in the DALSTM-AE model is determined to be 12. The batch size is set to 32, and the training epochs are set to 300. Training is performed using the Adam optimization algorithm with an initial learning rate of 0.001.

Four other methods are adopted for comparison, including DPCA (Russell et al., 2000), DiPCA (Dong and Qin, 2018), LSTM-AE (Pota et al., 2023), and AE-BiLSTM (Lee et al., 2022). In DPCA, the number of principal components is determined by both empirical selection and comparative experiments. By comparing different cumulative variance contribution rates (from 80% to 95%), we ultimately chose to retain 12 principal components with a cumulative contribution rate of 95% to achieve optimal monitoring performance. In DiPCA with a fixed time lag of $s_{\max}$, the number of dynamic latent variables is determined by the cross-validation method, resulting in 5 for this case. LSTM-AE is an autoencoder composed of two LSTMs with 12 hidden units. AE-BiLSTM is an encoder-decoder network using bi-directional LSTM cells with 12 hidden units. The batch size is set to 32. For a fair comparison, the confidence levels for calculating the control limits are set to 99% across all methods.

The detection results of the two fault cases are listed in Table 1. For DPCA, the FDRs of both statistical indicators T^2 and SPE are considerably low on the two faults, indicating that DPCA fails to extract effective features to distinguish faulty samples from normal samples. This is mainly because DPCA is difficult to extract nonlinear features, and the simple dynamic augmentation matrix processing cannot adapt to complex dynamic features. The dynamic statistical metric Φ_v of DiPCA is effective in monitoring dynamic relationships, achieving FDRs greater than 90% with acceptable FARs. Nonetheless, the FARs of the static statistical metric Φ_s are notably high.

To better compare the model performance, the monitoring metrics of LSTM-AE and AE-BiLSTM also utilize RE^2, MD^2 and OD. For fault 1, the FDRs of the OD indicator for both LSTM-AE and AE-BiLSTM are over 90%. However, when examining the FDRs of the individual RE^2 indicator, they are found to be below 50%, indicating shortcomings in both methods. For fault 2, the dynamic relationship of simulated fault samples has changed. The conventional LSTM-AE cannot detect the fault accurately with an overall FDR of 64.6%. The FDR of the OD metrics for AE-BiLSTM is 77.8%, superior to LSTM-AE, but it still falls short of the requirements. The proposed DALSTM-AE algorithm can accurately detect faulty samples and achieve the optimum on both the FDR and FAR performance metrics. Specifically, the FDRs for RE^2, MD^2 and OD in both faults are higher than 98%, while the FARs are around 1%. Due to space constraints, only the detection charts for fault 2 are displayed in Fig. 8. The pink vertical line denotes the fault occurrence time, while the dashed red line indicates the control limit set at a 99% confidence level. The proposed DALSTM-AE algorithm exhibits superior capability in distinguishing faulty samples compared to other methods.

4.2. Tennessee Eastman process case

The TE process is a simulation based on real chemical reaction processes developed by the control department of Eastman Chemicals. This benchmark process simulates various characteristics of real industrial systems and is therefore widely used to test control and fault diagnosis models of complex industrial processes. The TE process consists of five main units: reactor, condenser, compressor, separator, and stripper. The variables generated by the TE process under closed-loop conditions

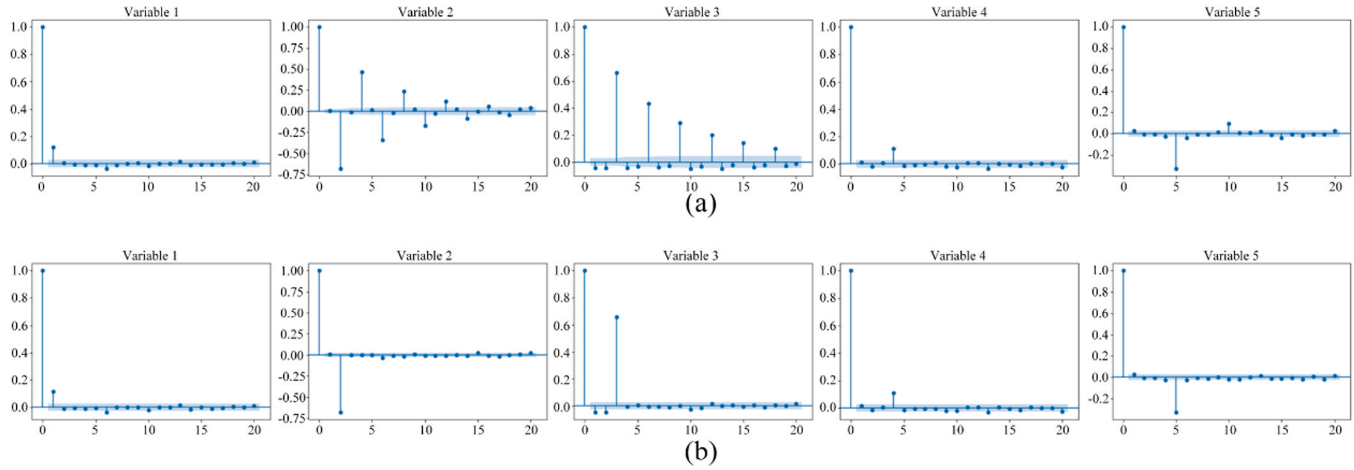

Fig. 7. Autocorrelation and partial autocorrelation of x_k in numerical simulation process. (a) autocorrelation, (b) partial autocorrelation.

Table 1
Evaluation indices of numerical case studies.

Fault number	Indices	DPCA		DiPCA		LSTM-AE			AE-BiLSTM			DALSTM-AE		
		T^2	*SPE*	Φ_v	Φ_s	RE^2	MD^2	*OD*	RE^2	MD^2	*OD*	RE^2	MD^2	*OD*
Fault 1	FDRs	0.04	0.22	0.95	0.99	0.45	0.96	0.96	0.49	0.93	0.93	**0.99**	**1.00**	**1.00**
	FARs	0.00	0.02	0.01	0.96	0.01	0.01	0.02	0.01	0.00	0.01	**0.00**	**0.00**	**0.00**
Fault 2	FDRs	0.01	0.11	0.95	0.98	0.32	0.65	0.65	0.63	0.77	0.78	**0.99**	**0.99**	**0.99**
	FARs	0.01	0.01	0.02	0.98	0.02	0.01	0.02	0.02	0.01	0.03	**0.01**	**0.01**	**0.01**
AVG	FDRs	0.03	0.17	0.94	0.98	0.39	0.80	0.80	0.56	0.85	0.86	**0.99**	**0.99**	**0.99**
	FARs	0.01	0.02	0.01	0.97	0.02	0.01	0.02	0.02	0.01	0.02	**0.01**	**0.01**	**0.01**

exhibit complex and uneven dynamic features, which are often overlooked by other methods. Therefore, we choose the TE process to validate the monitoring performance of the proposed method. TE process contains 52 process variables, including 11 manipulated variables (excluding the stirring rate of the reactor) and 41 measured variables (22 continuous and 19 component variables). The manipulated variables and the first 22 measured variables are sampled at intervals of 3 minutes. The remaining 19 measurement variables have sampling intervals of 6 or 15 minutes. The process consists of a dataset for normal operating conditions and 21 fault datasets for different fault conditions. The processed dataset can be accessed from the website: http://web.mit.edu/braatzgroup/links.html. In this downloaded dataset, each type of fault contains 960 samples, where the fault is introduced from the 8th hour, corresponding to sample 161.

Here we select 22 measurement variables and 11 manipulated variables for the fault detection experiment. The training dataset comprises normal operation data with 960 samples, and an additional 500 samples of normal data are allocated for testing. By analyzing the ACF and PACF of the variables, it is determined that the maximum time lag is 9, implying that the serialization preprocessing uses a sliding window of length $L = 10$. In the experiment of this case, the number of hidden units of DALSTM-AE is set to 20. The batch size and training epochs are 16 and 550, respectively. Through comparative validation, the number of principal components of DPCA is set at 139, determined by 95% cumulative variance contribution of the best monitoring performance. The dynamic latent variable of DiPCA is 9. In both LSTM-AE and AE-BiLSTM, the number of hidden units in the encoder and decoder is set to 20.

The monitoring results of the proposed DALSTM-AE model on unseen normal test data are shown in Fig. 9. The proposed method demonstrates a low incidence of false alarms when monitoring unseen normal data. The FAR for RE^2 is 1% and the FAR for MD^2 is 1.4%. The monitoring results indicate that the proposed method exhibits excellent generalization, effectively capturing the dynamic characteristics and nonlinear relationships of normal data.

The detection results of DALSTM-AE and the comparison methods are summarized in Table 2. Faults 3, 9, and 15 are not included in the calculation of the average results because they are extremely difficult to detect for process monitoring methods. DPCA has the worst detection performance, with an average FDR of less than 70% for T^2 and an average FAR of over 60% for *SPE*. The static statistical metric Φ_s of the DiPCA method also has a high average FAR. Meanwhile, its dynamic metric Φ_v cannot effectively identify all fault samples, with an average FDR of 71%. Compared with the LSTM-AE and AE-BiLSTM methods, the proposed DALSTM-AE achieves the best performance with the two statistical indicators, especially on IDV [5, 10, 12, 16, 19]. For RE^2, the average FDR of DALSTM-AE is 87.6%, which is higher than that of LSTM-AE at 78.2% and AE-BiLSTM at 82.7%. Moreover, the average FAR of the proposed method is 1.4%, which is lower than the other two methods. As for the statistical indicator MD^2, the average FARs of LSTM-AE and AE-BiLSTM are 14.1% and 10.2%, respectively. The *OD* indicator combining the two statistics also shows similar results, with high overall FAR and FDR. This suggests that the two methods are prone to misidentifying normal fluctuating samples as faults, especially on IDV [1, 12, 14, 16, 17, 18]. The proposed DALSTM-AE can effectively detect faults while maintaining a low FAR, indicating its ability to distinguish between normal and abnormal samples. For the *OD* indicator, the average overall FAR is 3.6% and the average overall FDR is 95.4%. The results indicate that the dual-attention mechanism and the special encoder-decoder architecture significantly enhance the fault detection capability of DALSTM-AE.

In addition, the detection charts of IDV(5) are listed in Fig. 10 to intuitively illustrate the effectiveness of the proposed method. Fault 5 is a step fault with a change in the internal temperature of the compressor condensate. As the simulation process continues, the system's variations gradually stabilize, forming a new steady-state system that is difficult to detect by conventional methods. As shown in Fig. 10, DPCA, DiPCA, LSTM-AE, and AE-DiLSTM methods cannot effectively distinguish between normal operation samples and faulty samples. In contrast, both

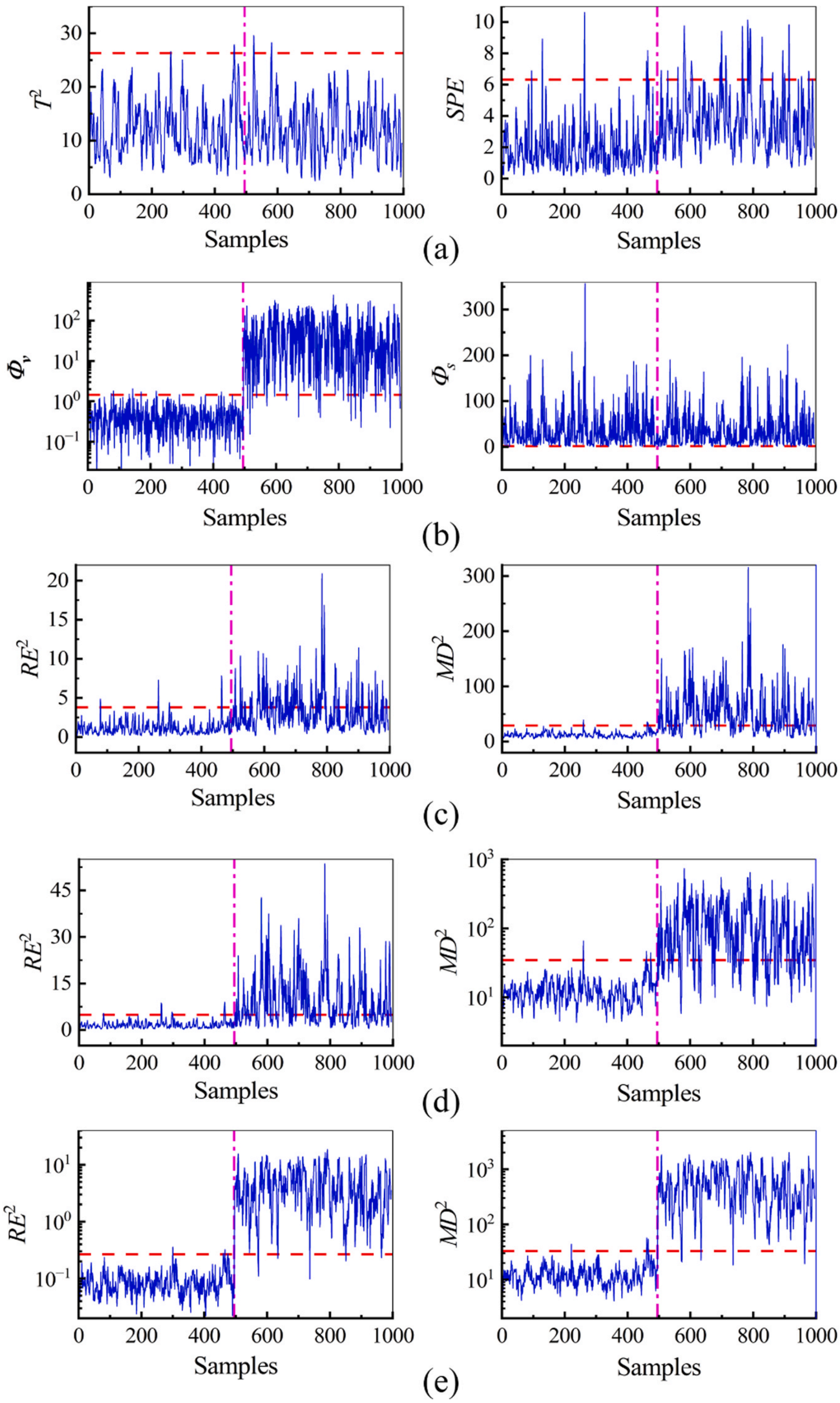

Fig. 8. Monitoring charts for fault 2 of numerical simulation. (a) DPCA, (b) DiPCA, (c) LSTM-AE, (d) AE-BiLSTM, (e) DALSTM-AE.

statistical indicators RE^2 and MD^2 of the proposed DALSTM-AE method successfully identify all fault samples. DALSTM-AE demonstrates the highest fault detection performance with FDR of up to 100% and FAR below 1%. The experiment demonstrates the applicability of the DALSMT-AE model in fault detection of complex industrial dynamic processes.

4.3. Practical cases of coal pulverizing system

We also investigate two practical cases of coal pulverization systems to illustrate the effectiveness of the proposed method. The coal pulverizing system plays a crucial role in coal-fired power plants, as it supplies pulverized coal of optimum temperature and fineness for combustion in the furnace. It contains raw coal bunker, coal feeder, coal mill, rotary separator, and fan system, as illustrated in Fig. 11. In practice, the coal feed flow of the coal feeder is adjusted in real time according to the changes in unit load, and the primary air flow of the coal mill is automatically adjusted according to the set air-to-coal ratio. Consequently, the coal mill outlet temperature deviates from the desired set value, leading to an adjustments of the hot and cold primary airflow. The coupled closed-loop control system of the pulverizing system makes its process variables have significant dynamic characteristics and

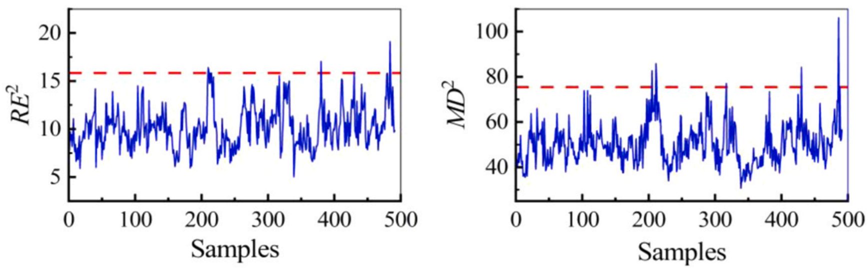

Fig. 9. Monitoring charts of DALSTM-AE for TE process normal test data.

Table 2
FAR/FDR performance evaluation indices of TE process.

Fault	DPCA		DiPCA		LSTM-AE			AE-BiLSTM			DALSTM-AE		
	T^2	SPE	Φ_v	Φ_s	RE^2	MD^2	OD	RE^2	MD^2	OD	RE^2	MD^2	OD
IDV(1)	0.00/0.99	0.71/1.00	0.11/1.00	0.55/1.00	0.01/0.99	0.11/1.00	0.12/1.00	0.03/1.00	0.11/1.00	0.13/1.00	0.00/1.00	0.01/1.00	0.01/1.00
IDV(2)	0.00/0.98	0.72/1.00	0.03/0.99	0.38/0.99	0.02/0.98	0.05/0.99	0.06/0.99	0.03/0.98	0.03/0.98	0.05/0.98	0.01/0.98	0.03/0.98	0.03/0.98
IDV(3)	0.00/0.00	0.70/0.71	0.03/0.06	0.46/0.59	0.00/0.03	0.42/0.34	0.42/0.34	0.01/0.06	0.43/0.26	0.43/0.26	0.07/0.03	0.09/0.09	0.13/0.11
IDV(4)	0.00/0.30	0.74/1.00	0.10/1.00	0.36/1.00	0.03/1.00	0.13/1.00	0.13/1.00	0.03/1.00	0.02/1.00	0.05/1.00	0.01/1.00	0.01/1.00	0.01/1.00
IDV(5)	0.00/0.24	0.74/1.00	0.10/0.13	0.38/0.78	0.03/0.28	0.13/1.00	0.13/1.00	0.03/0.43	0.02/1.00	0.05/1.00	0.01/1.00	0.01/1.00	0.01/1.00
IDV(6)	0.00/0.98	0.65/1.00	0.09/1.00	0.33/1.00	0.01/1.00	0.03/1.00	0.03/1.00	0.04/1.00	0.03/1.00	0.07/1.00	0.00/1.00	0.00/1.00	0.00/1.00
IDV(7)	0.00/1.00	0.77/1.00	0.08/0.72	0.40/1.00	0.03/1.00	0.01/1.00	0.04/1.00	0.03/1.00	0.03/1.00	0.06/1.00	0.00/1.00	0.01/1.00	0.01/1.00
IDV(8)	0.00/0.97	0.62/0.99	0.03/0.96	0.38/1.00	0.04/0.98	0.12/0.99	0.16/0.99	0.01/0.97	0.04/0.98	0.05/0.98	0.00/0.97	0.01/0.98	0.01/0.98
IDV(9)	0.00/0.00	0.75/0.71	0.06/0.06	0.71/0.61	0.03/0.04	0.38/0.29	0.38/0.29	0.07/0.08	0.26/0.26	0.30/0.28	0.01/0.02	0.09/0.06	0.10/0.08
IDV(10)	0.00/0.08	0.56/0.95	0.07/0.08	0.44/0.87	0.01/0.40	0.05/0.98	0.05/0.98	0.00/0.60	0.03/0.97	0.03/0.97	0.00/0.62	0.02/0.94	0.02/0.94
IDV(11)	0.00/0.75	0.68/1.00	0.09/0.86	0.46/0.97	0.01/0.95	0.11/0.99	0.11/0.99	0.04/0.96	0.05/0.97	0.08/0.98	0.06/0.95	0.00/0.95	0.06/0.96
IDV(12)	0.01/1.00	0.60/1.00	0.07/0.95	0.38/1.00	0.03/1.00	0.21/1.00	0.21/1.00	0.11/1.00	0.20/1.00	0.25/1.00	0.01/1.00	0.03/1.00	0.03/1.00
IDV(13)	0.00/0.94	0.68/0.99	0.04/0.92	0.39/0.99	0.00/0.94	0.09/0.96	0.09/0.96	0.00/0.95	0.03/0.96	0.03/0.96	0.00/0.95	0.01/0.95	0.01/0.95
IDV(14)	0.00/1.00	0.70/1.00	0.14/1.00	0.43/1.00	0.00/1.00	0.11/1.00	0.11/1.00	0.03/1.00	0.14/1.00	0.15/1.00	0.03/1.00	0.00/1.00	0.03/1.00
IDV(15)	0.01/0.00	0.69/0.69	0.08/0.05	0.36/0.53	0.01/0.02	0.23/0.36	0.23/0.37	0.03/0.06	0.16/0.33	0.19/0.34	0.00/0.04	0.02/0.15	0.02/0.17
IDV(16)	0.01/0.03	0.75/0.97	0.09/0.10	0.67/0.90	0.03/0.23	0.42/1.00	0.42/1.00	0.07/0.49	0.45/0.99	0.45/0.99	0.02/0.64	0.13/0.98	0.15/0.98
IDV(17)	0.00/0.91	0.70/0.99	0.08/0.95	0.44/0.98	0.01/0.97	0.24/0.98	0.24/0.98	0.04/0.97	0.13/0.98	0.15/0.98	0.05/0.98	0.04/0.98	0.07/0.98
IDV(18)	0.00/0.89	0.69/0.96	0.05/0.91	0.39/0.95	0.04/0.90	0.19/0.91	0.20/0.91	0.09/0.90	0.16/0.92	0.21/0.92	0.02/0.90	0.01/0.90	0.02/0.91
IDV(19)	0.00/0.45	0.62/1.00	0.08/0.33	0.35/0.89	0.00/0.48	0.03/1.00	0.03/1.00	0.00/0.56	0.08/1.00	0.08/1.00	0.00/0.73	0.04/1.00	0.04/1.00
IDV(20)	0.00/0.44	0.75/0.97	0.04/0.65	0.32/0.89	0.01/0.58	0.12/0.92	0.13/0.92	0.00/0.67	0.03/0.92	0.03/0.92	0.02/0.64	0.01/0.92	0.03/0.92
IDV(21)	0.00/0.41	0.68/0.83	0.08/0.21	0.60/0.56	0.01/0.41	0.40/0.73	0.40/0.93	0.07/0.42	0.27/0.69	0.30/0.69	0.03/0.41	0.12/0.58	0.12/0.58
AVG (excluding 3, 9, 15)	0.00/0.69	0.68/0.98	0.08/0.71	0.42/0.93	0.02/0.78	0.14/0.97	0.15/0.97	0.04/0.83	0.10/0.96	0.12/0.96	**0.01/0.88**	**0.03/0.95**	**0.04/0.95**

nonlinearity.

Two typical faults are considered, namely, abnormal loading pressure (case 1) and coal leakage (case 2). The two fault cases are respectively recorded from a 660 MW unit and a 330 MW unit located in Huzhou, Zhejiang Province, China. To enhance detection accuracy and minimize false alarms, variables are selected to construct a fault detection model based on expert knowledge and practical experience. Additionally, we remove variables with low correlation and lack of dynamic information to reduce the computational burden. The data information is outlined in Table 3.

By analyzing the ACF and PACF of the variables, the maximum time lags of case 1 and case 2 are determined to be 14 and 26, respectively. The data from the normal operation of the practical system is divided into training data and test data, both of which are serialized through data preprocessing. The critical hyperparameters of the model are determined by the grid search method based on cross-validation to obtain the best reconstruction and detection performance simultaneously. For the proposed DALSTM-AE, the number of hidden units is set to 30 for case 1 and 20 for case 2. The number of principal components for DPCA is 13 and 6, corresponding to case 1 and case 2. The number of

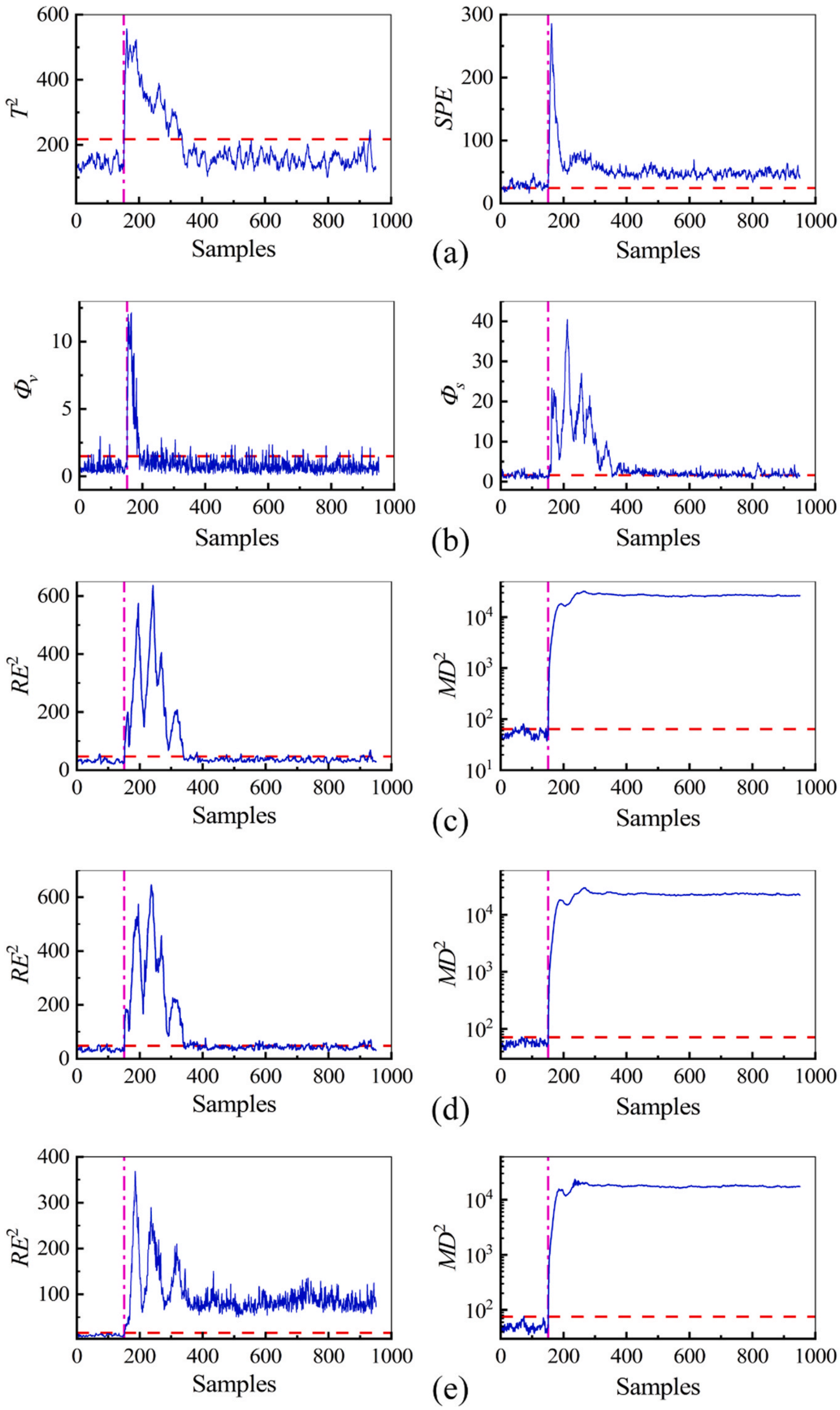

Fig. 10. Monitoring charts for IDV(5) of TE process. (a) DPCA, (b) DiPCA, (c) LSTM-AE, (d) AE-BiLSTM, (e) DALSTM-AE.

dynamic latent variables for DiPCA is 13 and 11, respectively. The number of neurons in LSTM-AE and AE-BiLSTM is consistent with the proposed method in this work.

In this experiment, FDR and FAR are similarly used as detection performance evaluation metrics. However, the exact start time of a fault is often difficult to determine in actual industrial processes. The time of fault recording in the operation log often lags behind the actual fault occurrence time. However, for this study, the FDR is computed based on the recording time, which indicates that the system is indeed in a fault state. In addition, the time when an algorithm can continuously detect the fault is considered to be the actual start time of a fault.

The detection results are summarized in Table 4. The detection charts of five methods for case 1 are listed in Fig. 12, and their analysis is discussed below. For DPCA, the T^2 statistic shows a detection delay with significant missed detections in the early stage of this fault. The FAR of the *SPE* statistic is 30.6%, which fails to identify normal process state changes. For DiPCA, the FAR of the static statistical metric Φ_s is particularly high, and normal variations caused by unit load changes are mistaken for faults. The FDR of the dynamic metric Φ_v is only 63.8%, which cannot detect the fault accurately. As illustrated by the green

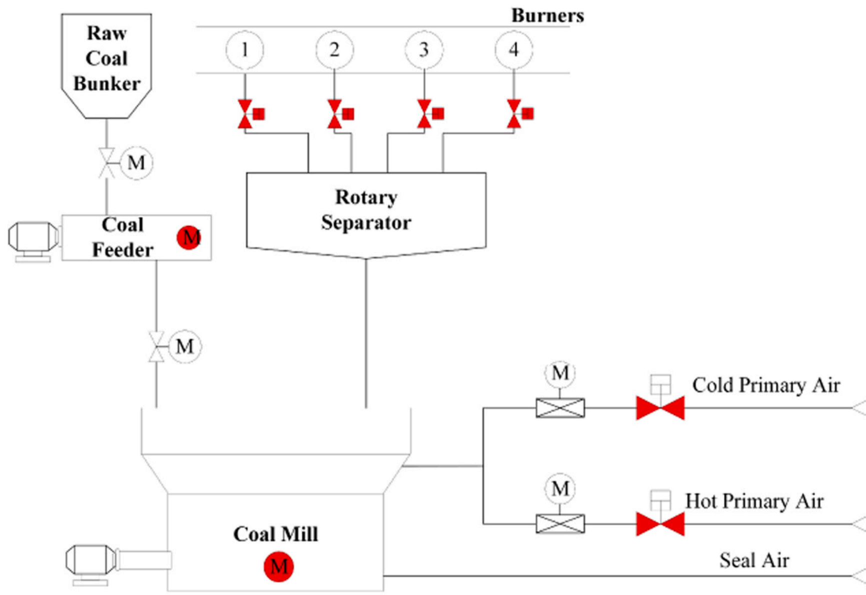

Fig. 11. Schematic diagram of coal pulverizing system.

Table 3
Data information of the practical pulverizing system.

Practical case number	Fault type	Key variables	Sampling interval	Number of training data/ test data/ fault test data	Fault recording point	Fault cause
Case 1	Abnormal loading pressure	29 variables: Unit load, current of coal mill, coal feed flow, primary air pressure, differential pressure of coal mill inlet and outlet, hydraulic pump current, loading oil pressure, etc.	1 min	5130/570/654	383	Internal leakage of hydraulic cylinder
Case 2	Coal leakage	13 variables: Unit load, current of coal mill, coal feed flow, outlet temperature, differential pressure between seal air and primary air, differential pressure of coal mill inlet and outlet, etc.	20 s	3207/400/1257	573	Coal mill carbon seal ring leaking powder

Table 4
Evaluation indices of practical case studies.

Practical case number	Indices	DPCA		DiPCA		LSTM-AE			AE-BiLSTM			DALSTM-AE		
		T^2	SPE	Φ_v	Φ_s	RE^2	MD^2	OD	RE^2	MD^2	OD	RE^2	MD^2	OD
Case 1	FDRs	0.89	1.00	0.64	0.96	0.89	1.00	1.00	0.92	1.00	1.00	**1.00**	**1.00**	**1.00**
	FARs	0.01	0.31	0.07	0.65	0.00	0.36	0.36	0.04	0.28	0.28	**0.02**	**0.02**	**0.04**
Case 2	FDRs	0.63	0.89	0.38	1.00	0.93	1.00	1.00	0.94	1.00	1.00	**0.93**	**1.00**	**1.00**
	FARs	0.00	0.03	0.08	0.95	0.22	0.25	0.38	0.14	0.23	0.33	**0.03**	**0.00**	**0.03**
AVG	FDRs	0.76	0.95	0.51	0.98	0.91	1.00	1.00	0.97	1.00	1.00	**0.97**	**1.00**	**1.00**
	FARs	0.00	0.17	0.08	0.80	0.11	0.31	0.37	0.12	0.32	0.31	**0.03**	**0.01**	**0.04**

vertical line in Fig. 12(e), the MD^2 statistic of DALSTM-AE first detects the fault at sequence sample point 330, which is considered to be the start time of this fault. The time detected is approximately 39 minutes earlier than the recorded time, which is valuable for incipient fault detection. The MD^2 of LSTM-AE also has similar results, detecting faults at an early stage. However, it fails to effectively distinguish between normal process state changes and fault-induced changes in dynamic systems, resulting in more than 35% of FARs. AE-BiLSTM suffers from the same problem with an average overall FAR of 30.8%. In contrast, the proposed DALSTM-AE method has fewer false alarms, with FARs in the OD indicator below 4%. These detection results further demonstrate the effectiveness and superiority of DALSTM-AE in real industrial dynamic processes.

4.4. Sensitivity analysis to sequence length

To further validate the applicability of the proposed DALSTM-AE model to complex long sequence data, we investigate the effect of sequence length L on the model's reconstruction performance. Specifically, the root mean square error (RMSE) (the square root of the loss function) is used to evaluate the accuracy of model reconstruction. The results of the TE process data and practical case 1 under different sequence lengths are shown in Fig. 13. The findings show that the proposed DALSTM-AE model exhibits robustness to sequence length as its RMSE varies slowly with the increase in sequence length. Additionally, DALSTM-AE consistently achieves the smallest RMSE at various sequence lengths, demonstrating optimal reconstruction performance.

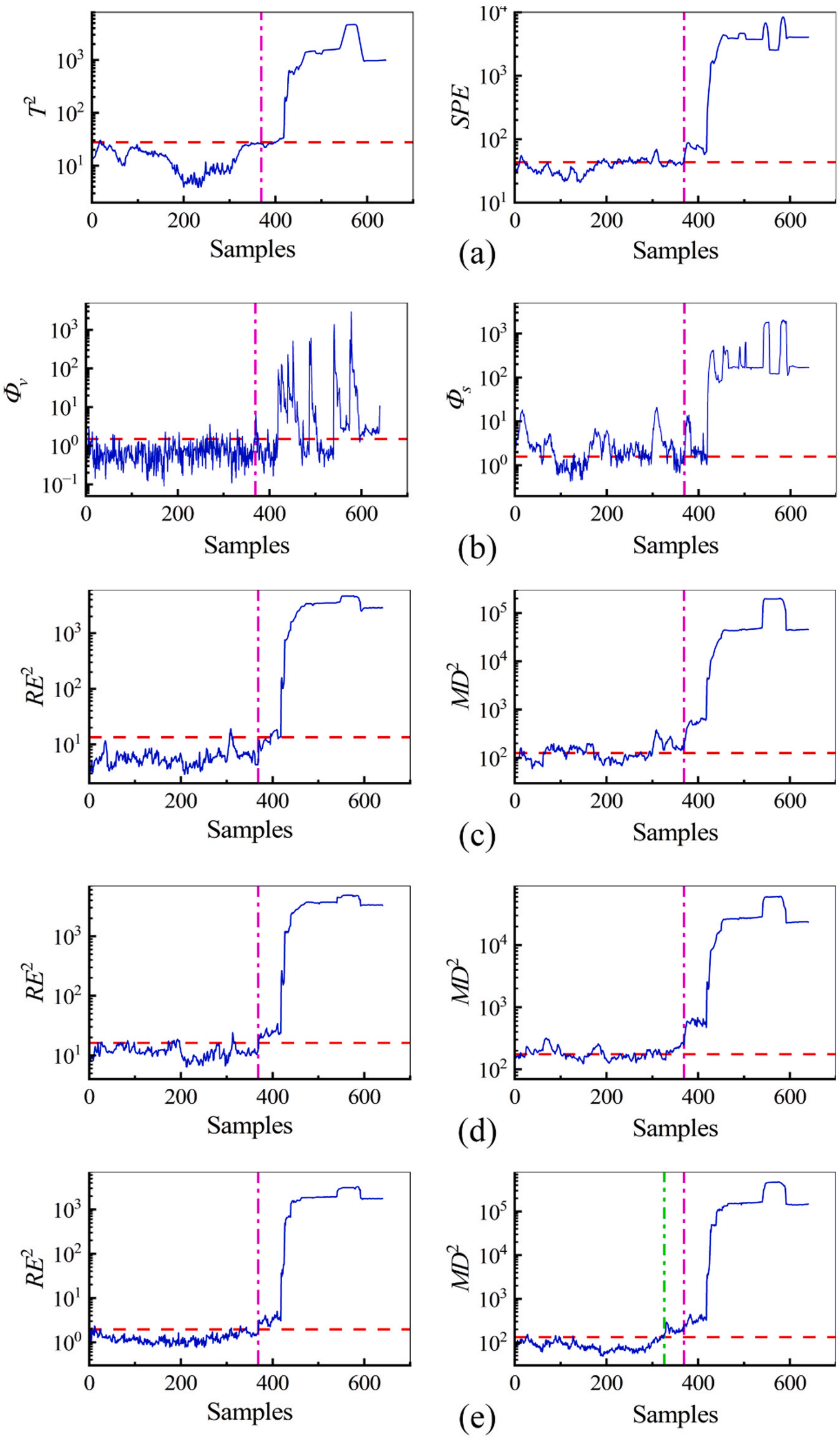

Fig. 12. Monitoring charts for case 1 of the practical pulverizing system. (a) DPCA, (b) DiPCA, (c) LSTM-AE, (d) AE-BiLSTM, (e) DALSTM-AE.

In contrast, the reconstruction performance of LSTM-AE and AE-BiLSTM deteriorates significantly with increasing sequence length. The experiment results highlight that DALSTM-AE can effectively learn temporal features and latent representations, addressing the issue of degradation in reconstruction performance when dealing with complex and lengthy sequences. As a result, the proposed method learns the effective features of the complex industrial data and performs effectively in process monitoring and fault detection.

5. Conclusions

In this paper, a novel unsupervised deep learning model, DALSTM-AE, is proposed for fault detection in industrial complex dynamic processes. Specifically, the LSTM and AE networks are integrated into a special encoder-decoder LSTM architecture that can capture the dynamic relationships and deep representations of variables in an unsupervised manner. The dual attention module is integrated into the decoder, enabling the selection of critical information and effective

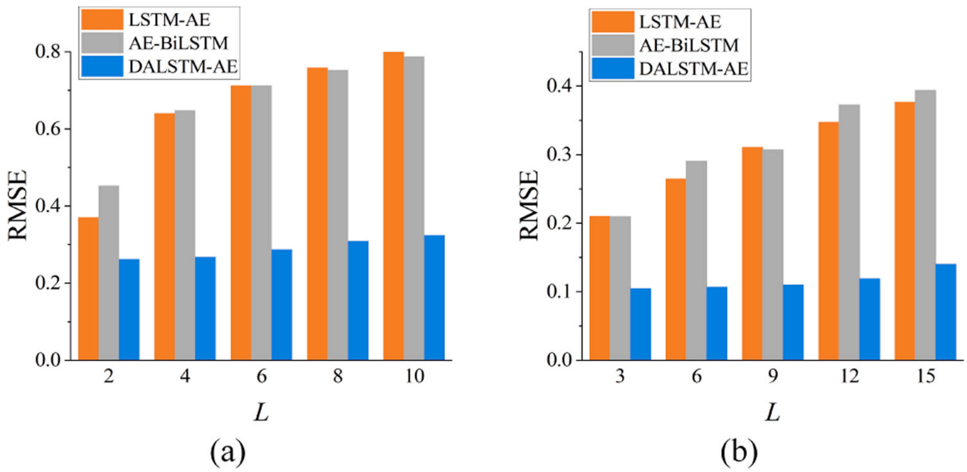

Fig. 13. Sensitivity analysis of the effect of sequence length on model reconstruction. (a) TE process, (b) case 1 of the practical pulverizing system.

extraction of dynamic features from complex time series data, thus addressing the problem of information loss in long time series. In addition, the monitoring statistics RE^2 and MD^2 are designed, and the corresponding control limits are calculated by KDE. The introduction of the *OD* indicator, considering both statistical metrics, provides a comprehensive evaluation of performance. The effectiveness of the proposed method is demonstrated by case studies on a numerical simulation example, the TE benchmark process, and practical coal pulverizing systems in power plants. The results of the comparative analysis indicate that the proposed DALSTM-AE exhibits superior monitoring capabilities in complex dynamic processes. It effectively detects anomalies at an early stage and provides timely warnings for various types of safety risks. The proposed method has a significant reference value for safety and risk assessment of industrial dynamic processes. However, it should be noted that DALSTM-AE is designed for single-mode dynamic process and multi-mode non-stationary dynamic process monitoring needs to be considered in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities (2022ZFJH004) and the Program of Introducing Talents of Discipline to University (BP0820002).

References

Aghaee, M., Krau, S., Tamer, M., Budman, H., 2023. Unsupervised fault detection of pharmaceutical processes using long short-term memory autoencoders. Ind. Eng. Chem. Res. 62, 9773–9786.
Amini, N., Zhu, Q., 2022. Fault detection and diagnosis with a novel source-aware autoencoder and deep residual neural network. Neurocomputing 488, 618–633.
Arunthavanathan, R., Khan, F., Ahmed, S., Imtiaz, S., 2021. An analysis of process fault diagnosis methods from safety perspectives. Comput. Chem. Eng. 145, 107197.
de Carvalho Michalski, M.A., Martha de Souza, G.F., 2022. Comparing PCA-based fault detection methods for dynamic processes with correlated and Non-Gaussian variables. Expert Syst. Appl. 207, 117989.
Chen, H., Liu, H., Chu, X., Liu, Q., Xue, D., 2021. Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network. Renew. Energy 172, 829–840.
Chen, J., Li, J., Chen, W., Wang, Y., Jiang, T., 2020. Anomaly detection for wind turbines based on the reconstruction of condition parameters using stacked denoising autoencoders. Renew. Energy 147, 1469–1480.
van de Sand, R., Corasaniti, S., Reiff-Stephan, J., 2021. Data-driven fault diagnosis for heterogeneous chillers using domain adaptation techniques. Control Eng. Pract. 112, 104815.
Deng, X., Zhang, X., Liu, X., Cao, Y., 2023. Incipient fault detection of nonlinear chemical processes based on probability-related randomized slow feature analysis. Process Saf. Environ. Prot. 169, 797–807.
Dong, Y., Qin, S.J., 2018. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. J. Process Control 67, 1–11.
Gao, L., Li, D., Liu, X., Liu, G., 2022. Enhanced chiller faults detection and isolation method based on independent component analysis and k-nearest neighbors classifier. Build. Environ. 216, 109010.
Jeng, J.-C., 2010. Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms. J. Taiwan Inst. Chem. Eng. 41, 475–481.
Kong, X., Yang, Z., Luo, J., Li, H., Yang, X., 2022. Extraction of reduced fault subspace based on KDICA and its application in fault diagnosis. IEEE Trans. Instrum. Meas. 71, 1–12.
Lee, S., Jin, H., Nengroo, S.H., Doh, Y., Lee, C., Heo, T., Har, D., 2022. Smart Metering System Capable of Anomaly Detection by Bi-directional LSTM Autoencoder, 2022 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6.
Li, D., Li, L., Li, X., Ke, Z., Hu, Q., 2020. Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation. Neurocomputing 411, 351–363.
Li, Z., Tian, L., Jiang, Q., Yan, X., 2021. Distributed-ensemble stacked autoencoder model for non-linear process monitoring. Inf. Sci. 542, 302–316.
Lin, Z., Hu, Z., Peng, J., Zhao, H., 2022. Dynamic-scale graph neural network for fault detection. Process Saf. Environ. Prot. 168, 953–970.
Liu, H., Yang, J., Zhang, Y., Yang, C., 2021a. Monitoring of wastewater treatment processes using dynamic concurrent kernel partial least squares. Process Saf. Environ. Prot. 147, 274–282.
Liu, Z., Ma, Q., Cai, B., Liu, Y., Zheng, C., 2021b. Risk assessment on deepwater drilling well control based on dynamic Bayesian network. Process Saf. Environ. Prot. 149, 643–654.
Meng, H., Yang, Q., Zio, E., Xing, J., 2023. An integrated methodology for dynamic risk prediction of thermal runaway in lithium-ion batteries. Process Saf. Environ. Prot. 171, 385–395.
Osarogiagbon, A., Muojeke, S., Venkatesan, R., Khan, F., Gillard, P., 2020. A new methodology for kick detection during petroleum drilling using long short-term memory recurrent neural network. Process Saf. Environ. Prot. 142, 126–137.
Pota, M., De Pietro, G., Esposito, M., 2023. Real-time anomaly detection on time series of industrial furnaces: a comparison of autoencoder architectures. Eng. Appl. Artif. Intell. 124, 106597.
Qian, Q., Qin, Y., Wang, Y., Liu, F., 2021. A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis. Measurement 178, 109352.
Russell, E.L., Chiang, L.H., Braatz, R.D., 2000. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemom. Intell. Lab. Syst. 51, 81–93.
Scott, D.W., 2015. Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley & Sons.
Sehovac, L., Grolinger, K., 2020. Deep learning for load forecasting: SEquence to Sequence Recurrent Neural Networks with Attention. IEEE Access 8, 36411–36426.
Srivastava, N., Mansimov, E., Salakhudinov, R., 2015. Unsupervised Learning of Video Representations Using Lstms, International Conference on Machine Learning. PMLR, pp. 843–852.
Tao, Y., Shi, H., Song, B., Tan, S., 2020. A novel dynamic weight principal component analysis method and hierarchical monitoring strategy for process fault detection and diagnosis. IEEE Trans. Ind. Electron. 67, 7994–8004.
Wang, P., Ren, S., Wang, Y., Zhu, B., Fan, W., Si, F., 2022. Quality-related nonlinear process monitoring of power plant by a novel hybrid model based on variational autoencoder. Control Eng. Pract. 129.

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 Trans. 96, 457–467.

Xiang, L., Wang, P., Yang, X., Hu, A., Su, H., 2021. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. Measurement 175, 109094.

Yang, S., Kong, X., Wang, Q., Li, Z., Cheng, H., Xu, K., 2022. Deep multiple auto-encoder with attention mechanism network: a dynamic domain adaptation method for rotary machine fault diagnosis under different working conditions. Knowl. Based Syst. 249, 108639.

Yin, Q., Yang, J., Tyagi, M., Zhou, X., Hou, X., Cao, B., 2021. Field data analysis and risk assessment of gas kick during industrial deepwater drilling process based on supervised learning algorithm. Process Saf. Environ. Prot. 146, 312–328.

Yu, J., Yan, X., 2021. A new deep model based on the stacked autoencoder with intensified iterative learning style for industrial fault detection. Process Saf. Environ. Prot. 153, 47–59.

Yu, W., Zhao, C., 2019. Recursive exponential slow feature analysis for fine-scale adaptive processes monitoring with comprehensive operation status identification. IEEE Trans. Ind. Inform. 15, 3311–3323.

Zeng, L., Zhang, X., Yu, K., Jin, Q., Wu, Y., Chen, L., Wu, X., 2023. Incipient fault detection and process monitoring of thermal power plant pulverizing system based on deep representation learning. Trans. Inst. Meas. Control.

Zhang, J., Zhao, C., 2022. Condition-driven probabilistic adversarial autoencoder with nonlinear Gaussian feature learning for nonstationary process monitoring. J. Process Control 117, 140–156.

Zhang, S., Qiu, T., 2022. Semi-supervised LSTM ladder autoencoder for chemical process fault diagnosis and localization. Chem. Eng. Sci. 251.

Zhang, Y., Ji, J., Ma, B., 2020. Fault diagnosis of reciprocating compressor using a novel ensemble empirical mode decomposition-convolutional deep belief network. Measurement 156, 107619.

Zheng, J., Zhao, C., Gao, F., 2022. Retrospective comparison of several typical linear dynamic latent variable models for industrial process monitoring. Comput. Chem. Eng. 157.

Zhou, D., Yao, Q., Wu, H., Ma, S., Zhang, H., 2020. Fault diagnosis of gas turbine based on partly interpretable convolutional neural networks. Energy 200, 117467.

Zhu, J., Shi, H., Song, B., Tan, S., Tao, Y., 2019. Deep neural network based recursive feature learning for nonlinear dynamic process monitoring. Can. J. Chem. Eng. 98, 919–933.

Zhu, P.-F., Chen, X.-F., Meng, F.-Q., He, X.-B., Chen, W.-H., 2022. Thermal risk modeling and safety optimization of an arylamine diazo reaction based on PLS algorithm. Process Saf. Environ. Prot. 166, 108–112.