Computers and Chemical Engineering 192 (2025) 108887

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

# Industrial Process Fault Detection Based on Siamese Recurrent Autoencoder

Cheng Ji [a], Fangyuan Ma [a,b], Jingde Wang [a], Wei Sun [a,*], Ahmet Palazoglu [c]

[a] College of Chemical Engineering, Beijing University of Chemical Technology, North Third Ring Road 15, Chaoyang District, 100029 Beijing, China
[b] Center of Process Monitoring and Data Analysis, Wuxi Research Institute of Applied Technologies, Tsinghua University, 214072 Wuxi, China
[c] Department of Chemical Engineering, University of California, Davis, CA 95616, USA

ARTICLE INFO

*Keywords:*
Chemical process monitoring
Process safety
Siamese autoencoder
Long short-term memory unit
Contrastive loss
Wax oil hydrogenation reactor

ABSTRACT

Although deep autoencoders excel at extracting intricate features, their application in process monitoring is limited by the requirement for large sample sizes and interpretability of latent representations. This work presents a special deep learning structure named Siamese network to detect abnormal deviations in nonlinear dynamic processes. By leveraging the capability of Siamese architecture to process multiple inputs simultaneously, the training sample size expands exponentially, which enhances the learning potential of the model. Furthermore, a long short-term memory unit is integrated to enable the capture of long-term process dynamics. To refine the distribution of latent features extracted from diverse data types, a contrastive loss function is proposed, which strengthens the model's fault detection capabilities and enhances its interpretation of latent representations. Then $T^2$ statistic is established on the latent space to perform fault detection. The effectiveness of the method is demonstrated through case studies on simulation processes and an industrial process.

## 1. Introduction

Abnormal situations occur in process industries often cause huge losses in terms of economic and physical security. Generally, an abnormal situation arises from the emergence and development of faults during process operation (Amin et al., 2018). A fault can be defined as at least one measurement deviates from its pre-defined acceptable operating regions (R.Isermann and P.Ballé, 1997). As one of the tools available in the Process Engineering System (PSE) toolbox, process monitoring techniques not only serve as an important guarantee for production safety and reliability but also provide a prerequisite platform for the development of process optimization and control systems to improve profitability (Modak et al., 2020; Pistikopoulos et al., 2021). Data-driven fault detection and diagnosis, which relies on process measurements that do not necessitate process knowledge (Alauddin et al., 2018), have been attracting attention from both academia and industry as the level of factory digitization and automation evolves (Ji and Sun, 2022; Qin, 2012; Yin et al., 2014).

Traditional Multivariate Statistical Process Monitoring (MSPM) methods, underpinned by Principal Component Analysis (PCA), Partial Least Squares (PLS), and Independent Component Analysis (ICA) have been applied to monitoring chemical processes through linear projection (He and Wang, 2018; Kresta et al., 1991; Qin, 2012). By examining the variance-covariance of process variables, MSPM methods transform original data into a principal component subspace and a residual subspace where the Hotelling's $T^2$ statistic and squared prediction error statistic are calculated respectively to perform fault detection (Nomikos and MacGregor, 1995). However, with the development scale of modern industrial plants, the production process is characterized by strong nonlinearity and process dynamics. Over the past three decades, these methods have been extensively studied and modified regarding the extraction of various complex features of industrial process data (Ku et al., 1995; Lee et al., 2004; Li et al., 2014; Li et al., 2022). As a result, numerous novel MSPM methods have been proposed and achieved satisfactory performance in the application of the Tennessee Eastman Process (TEP), which is a well-known benchmark platform. Several studies also reported the application of MSPM methods to monitor the operating status of certain specific units (Ji et al., 2022; Kumar et al., 2020).

Furthermore, the MSPM methods have also been combined with probabilistic models to achieve fault detection and root cause diagnosis simultaneously (Amin et al., 2018). Yu et al. proposed a probabilistic multivariate method for fault detection and diagnosis of industrial processes, in which Gaussian copula based on rank correlation was

* Corresponding author at: College of Chemical Engineering, Beijing University of Chemical Technology, 100029 North Third Ring Road 15, Chaoyang District, Beijing, China.
*E-mail address:* sunwei@mail.buct.edu.cn (W. Sun).

https://doi.org/10.1016/j.compchemeng.2024.108887
Received 4 July 2024; Received in revised form 3 September 2024; Accepted 30 September 2024
Available online 3 October 2024

employed for modeling of dependencies to distinguish between normal and faulty samples (Yu et al., 2015b). Amin et al. combined the probabilistic model R-vine copula with MSPM to propose a risk-based fault detection and diagnosis method (Amin et al., 2021). As another effective probabilistic model, Bayesian Networks (BN) excel at mapping causal relations and conditional dependencies while capturing uncertainty. It has been proven to exhibit strength in reliability and risk analysis compared to related methods such as Markov chains and fault trees (Khakzad et al., 2013). Khakzad et al. demonstrated the superior performance of BN in safety analysis through a comparison with the fault tree (Khakzad et al., 2011). Alauddin et al. integrated physics-based models within a BN framework to handle data variabilities in dynamical process systems (Alauddin et al., 2024). Furthermore, BN also shows promising applications in hybrid modeling for simultaneously achieving fault detection and propagation path analysis. Amin et al. proposed a hybrid model that combined PCA and BN to identify the root cause and fault propagation pathway (Amin et al., 2018). Yu et al. utilized a modified Independent Component Analysis (ICA) and Bayesian Network (BN) to effectively detect process faults and trace the fault propagation path (Yu et al., 2015a). Gharahbagheri et al. further proposed a hybrid method that incorporated Kernel PCA and BN to enhance its performance in nonlinear processes (Gharahbagheri et al., 2017). Galagedarage Don and Khan proposed a hybrid method based on hidden Markov model and BN for fault prognosis(Galagedarage Don and Khan, 2019). Amin et al. further conducted a comparative study to verify that the integration of multivariate fault probability into fault detection and diagnosis could lead to performance improvement (Amin et al., 2020). Overall, the hybrid modeling of BN and traditional fault detection and diagnosis methods utilizes the advantages of each method to overcome the limitations of an individual method, exhibiting promising application prospects. However, with the increasing scale and complexity of equipment in modern process industries, sensors are liberally placed to capture a wide range of dynamic behavior, the consequence of which is the strong correlations among them. Coupled with various random factors during operation, multivariate statistical methods often fail to effectively capture the inherent information embedded in process data.

As an alternative, deep neural networks are beginning to be applied more widely in process monitoring, relying on their powerful feature processing capabilities derived from advanced model structures and massive learnable parameters. As Industry 4.0 emerges, instrumentation has significantly advanced, resulting in a pervasive availability of sensor data. Simultaneously, the process industries are witnessing a surge in the scale and complexity of the equipment. In response, Artificial Intelligence (AI) methodologies have demonstrated their ability in modeling complex nonlinear relationships among process variables. Consequently, the chemical industry is progressively integrating AI into its digital transformation strategies. Deep Neural Networks (DNNs) have gained prominence among AI techniques, with the emergency of big models represented by the ChatGPT exemplifying this trend. These AI-based models demonstrate promising application prospect in various engineering applications such as fault detection (Bhakte et al., 2024; Bi et al., 2022). Under this scope, Autoencoders (AEs) have been the most commonly used methods. Generally, the AEs consist of two fully connected layers called encoder and decoder. The encoder aims to extract features from original data and the decoder is applied for data reconstruction. The operating status can be monitored by observing the reconstruction error (Sakurada and Yairi, 2014). The depth of the encoder and the decoder can be adjusted according to the complexity of data characteristics. In addition, the model structure of the encoder and the decoder can also be adjusted to handle different types of process data. Yu and Zhao proposed a denoising AE to improve the robustness of the model (Yu and Zhao, 2020). Zhang and Qiu proposed a dynamic-inner convolutional AE to monitor nonlinear and dynamic processes (Zhang and Qiu, 2022). Cheng et al. proposed a Variational Recurrent Autoencoder (VRAE), where the gate recurrent unit is adopted to extract the long-term time dependency of process data (Cheng et al., 2019). To further consider the graph structure of processes, a graph dynamic AE was proposed by Liu et al. for fault detection (Liu et al., 2022). Aiming to capture the spatial distribution information of measurements in the equipment, Ma et al. proposed a three-dimensional convolutional neural network to extract the spatial features of chemical process data (Ma et al., 2023). Regarding the issue of data imbalance, Jiang et al. proposed generative adversarial network-based method for monitoring industrial time series (Jiang et al., 2019). Jia et al. introduced graph structure into deep learning models to consider the topology relationship among variables (Jia et al., 2023). Liu et al. proposed a graph dynamic autoencoder to model the process dynamics (Liu et al., 2022). Lv et al. further proposed a special graph network named causality-embedded reconstruction network for high-resolution fault identification in chemical process (Lv et al., 2024). Liu and Jafarpour incorporated Granger causality map into graph attention network to perform fault detection and root cause diagnosis (Liu and Jafarpour, 2024). With the remarkable achievements of attention mechanisms in domains such as natural language processing and computer vision, they have also been introduced into fault detection in large-scale industrial processes (Lv et al., 2022; Zhou et al., 2023). Bi and Zhao proposed a orthogonal self-attentive VAE for fault detection, in which the orthogonal attention was introduced into the VAE to model the correlation and temporal dependency among process variables in different time steps (Bi and Zhao, 2021). Bi et al. further proposed a deep learning-based fault diagnosis method that utilized Transformer to discover causal relationships in large-scale chemical process (Bi et al., 2023). One can witness the rapid development of deep learning models in the field of process monitoring.

Theoretically, given a certain depth with sufficient neurons, a neural network model can capture almost any complex relationship among measurements. However, process safety requires a high degree of interpretability and model generalization ability, which presents a substantial challenge to process monitoring methods based on black-box deep learning models. Firstly, the more complex the model is, the more training samples are required, or the model will not be generalized to new samples in test data, resulting in poor monitoring performance in real-time operation. Moreover, the reconstruction error of the AE is utilized as the training target and the monitoring indicator in most present methods, such that the latent features extracted by the model are not interpretable or even involved in fault detection.

The main goal of this work is to contribute to advancing the state-of-the-art in deep learning-based process monitoring by introducing a novel and effective Siamese Recurrent Autoencoder (SRAE) that addresses key challenges and limitations of existing methods. The original code of SRAE is available at https://github.com/Cheng960724. Compared with most relevant works on this topic that utilized conventional deep autoencoders, the strengths of this study are mainly reflected in the following aspects:

(1) It utilizes a Siamese recurrent autoencoder architecture, which allows it to process multiple inputs simultaneously, and therefore enhance the learning potential of the model. Specifically, A Siamese neural network (SNN) with a long short-term memory (LSTM) unit is employed to conduct feature extraction from data with few available samples. This strength stems from the multi-input structure of the Siamese network. In this work, training data are input into the model in pairs. Compared with traditional neural networks with a single input, the available training samples can be increased exponentially, by which the generalization ability of the model is improved for a given sample size. Such a structure is mainly applied in the field of computer vision represented by face recognition, while has not ever been applied in fault detection and diagnosis.
(2) Most existing works that applied autoencoders for fault detection are trained using the reconstruction error, in which the distribution of latent representations has not been effectively regularized. Comparatively, a contrastive loss function is adopted in the training

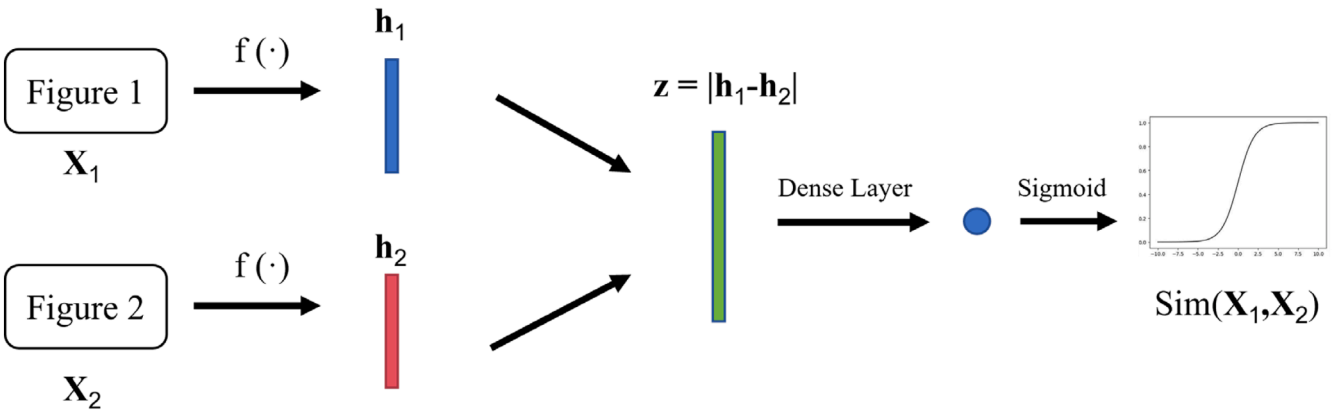


**Fig. 1.** Schematic diagram of SNN.

process of this study to compress the distribution of latent features extracted from data collected under normal operating conditions, which significantly strengthens the model's fault detection capabilities and enhances its interpretation of latent representations. The contrastive loss is developed based on the idea that the latent features extracted from data belonging to the same category, i.e., data under normal operating conditions in this work, should be similar. If a pair of inputs both belong to normal operating conditions, the distance between their latent features will be minimized, and if only one of them belongs to normal operating conditions, the distance will be maximized. This way, the latent features extracted from normal data will be regularized into a small region, by which the dissimilarity between normal data and fault data can be magnified and the fault detectability of the model is improved.

(3) Fault detection in existing works that utilize autoencoders is mostly performed through the mean square errors between input data and reconstructed data, while the fault detection of this work is performed based on the Siamese recurrent autoencoder with T2 statistics in the latent subspace. The main training objective of this study is to minimize the distance between latent features extracted from normal samples, and therefore a compressed region in the latent subspace that represents normal operating conditions can be determined to perform fault detection. The association between feature extraction and fault detection is established, by which the fault detection results are also more easily understood and interpreted.

The rest of this paper is organized as follows: the basic knowledge for the proposed method is introduced in Section 2. Section 3 presents the model design and procedures of the proposed SRAE-based process monitoring method. In Section 4, the proposed method is validated through three case studies, and results are discussed and compared with other related methods. The conclusions are drawn in Section 5.

## 2. Preliminaries

In this section, several basic methods adopted in the proposed method are briefly reviewed.

### 2.1. Auto Encoders (AEs)

AEs have been the most widely used deep learning model to implement fault detection tasks. As a special form of the neural network, AEs employ an encoder to obtain a compressed representation to convert original data into a latent space, and then utilize a decoder to reconstruct data with the information in the latent space. A primary AE can be described as follows,

$$z = f(\boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}) \tag{1}$$

$$\widetilde{\boldsymbol{x}} = f'(\boldsymbol{w}'\boldsymbol{z} + \boldsymbol{b}') \tag{2}$$

where $\boldsymbol{x}, \boldsymbol{z}, \widetilde{\boldsymbol{x}}$ represent the input data, latent features extracted by the encoder, and the reconstruction outputs obtained by the decoder, respectively. $f(\cdot)$ and $f'(\cdot)$ are nonlinear representations of the encoder and the decoder. $\boldsymbol{w}$ and $\boldsymbol{w}'$ are weights and $\boldsymbol{b}$ and $\boldsymbol{b}'$ represent the bias.

The training target of AEs is to find optimal parameters to minimize the reconstruction error between $x$ and $\widetilde{x}$. The reconstruction error is usually adopted as the monitoring indicator in AE-based process monitoring methods. The faulty data could not be reconstructed by the feature extractor established with data from normal operating conditions. As a result, the reconstruction error increases accordingly, by which the fault can be detected. To improve the process monitoring performance, the structure of neural networks in AE should be adjusted according to the data characteristics of different processes.

### 2.2. LSTM neural networks

Process dynamics are prevalent in industrial process data, which should be considered explicitly to improve the process monitoring performance. For this purpose, Recurrent Neural Networks (RNNs) are commonly applied to capture the process dynamics through the transition of hidden states. However, as the time step increases, the long-term time dependency of the process could be lost by conventional RNNs. LSTM unit is a well-known variant of RNNs to extract long-term time dependency. Its strength stems from the memory transitions controlled by a gate structure, which can be expressed as follows,

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{3}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f\right) \tag{4}$$

$$\boldsymbol{g}_t = tanh\left(\boldsymbol{W}_g \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_g\right) \tag{5}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{C}_t) \tag{8}$$

where $\boldsymbol{i}_t, \boldsymbol{f}_t, \boldsymbol{o}_t$ stand for the input gate, forget gate, and output gate, respectively, $\boldsymbol{x}_t$ is the input at the current time step, $\boldsymbol{g}_t$ is the current candidate state, $\boldsymbol{C}_t, \boldsymbol{C}_{t-1}$ are the memory cells of the current and last time step, $\boldsymbol{h}_t, \boldsymbol{h}_{t-1}$ are the hidden states of the current and last time step, $\boldsymbol{W}_\cdot, \boldsymbol{b}_\cdot$ are the weight and bias, and $\sigma, tanh$ are sigmoid and tanh activation functions. With the gate structure, irrelevant information of hidden

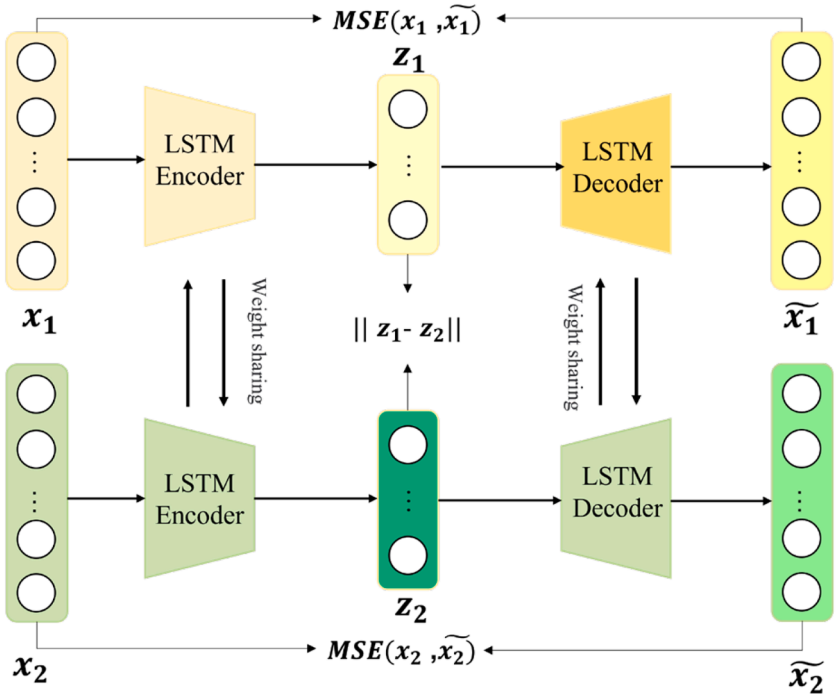


**Fig. 2.** The structural design of the proposed SRAE model.

states from the previous time step can be forgotten and useful information from the current time step can be retained by the memory cell, making the LSTM unit more effective in extracting long-term time dependency. Therefore, the LSTM unit is adopted in the proposed SRAE-based process monitoring method.

### 2.3. Siamese neural network (SNN)

SNN is a type of neural network architecture that involve two (or more) parallel neural networks that share the same architecture and weights. The name "Siamese" comes from the idea of two Siamese twins who are connected but operate independently. It was first introduced by Bromley et al. to perform signature verification (Bromley et al., 1993). The SNN is trained jointly on pairs (or tuples) of input data to compare their similarities or learn a relationship between them.. It shows significant strengths in few shots learning or even one-shot learning to solve image verification as an image matching problem (Koch et al., 2015). A Schematic diagram of SNN is shown in Fig. 1.

It can be shown that there are two parallel networks within the SNN. Each of the colored bars (blue and red) can be metaphorically regarded as latent representations obtained from separate but identical neural networks within the Siamese architecture. They are "parallel" in the sense that they process data independently but share the same underlying structure and parameters. The special characteristic is that each Siamese network would receive a different piece of data as input. For example, in a typical application of Siamese networks for face verification, one network might receive an image of a known face, and the other an image of a query face. The shared weights allow the networks to collaborate in learning a common representation of the data. The similarity between two inputs can be measured by the objective function, which is usually the Euclidean distance to measure the similarity in distance or Kullback-Leibler divergence to measure the similarity in their distributions.

If the different parts of input data come from various sources and exhibit different forms but there exist certain associations among them, the weights and the structures of each Siamese networks also could be different accordingly to better capture latent features from different parts of input data. In this situation, the model is called a pseudo-Siamese network. In this study, the SNN model shares the same weights and structure as both inputs are of the same type of data in the fault detection task, and the LSTM unit is adopted given the presence of process dynamics in industrial process data, as discussed before.

## 3. The SRAE-based process monitoring method

In this section, the proposed SRAE-based process monitoring method is presented, including the design of the model structure and loss function, the implementation procedures of the process monitoring method, and the indicators adopted to evaluate process monitoring performance.

### 3.1. Model design of the SRAE

To improve the generalization ability of deep learning-based process monitoring models, a novel SRAE model is proposed, whose structure is shown in Fig. 2.

The SNN is adopted as the basic structure of the proposed method,

$$\boldsymbol{z}_1 = LSTM(\boldsymbol{x}_1) \tag{9}$$

$$\boldsymbol{z}_2 = LSTM(\boldsymbol{x}_2) \tag{10}$$

where $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ are two inputs of the SNN, $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ are corresponding latent features extracted by the LSTM encoder. In most situations, only one type of data, time series, is considered in industrial process monitoring. As a result, the two inputs are from the same source with significant similarity between them. Therefore, the SNN model in the proposed SRAE shares the same weights and structure. On the other hand, through the introduction of the SNN, the generalization ability of the model trained with the same sample size can be significantly improved. This stems from the multi-input structure of the SNN, as noted before. Given $n$ training samples, the sample size can be expanded to $n(n-1)/2$ by an SNN with dual inputs.

The selection and processing of training data for neural network inputs significantly influences the model's performance. Data shuffling is a vital aspect of neural network training in various contexts, contributing to overfitting prevention, enhancing data independence,

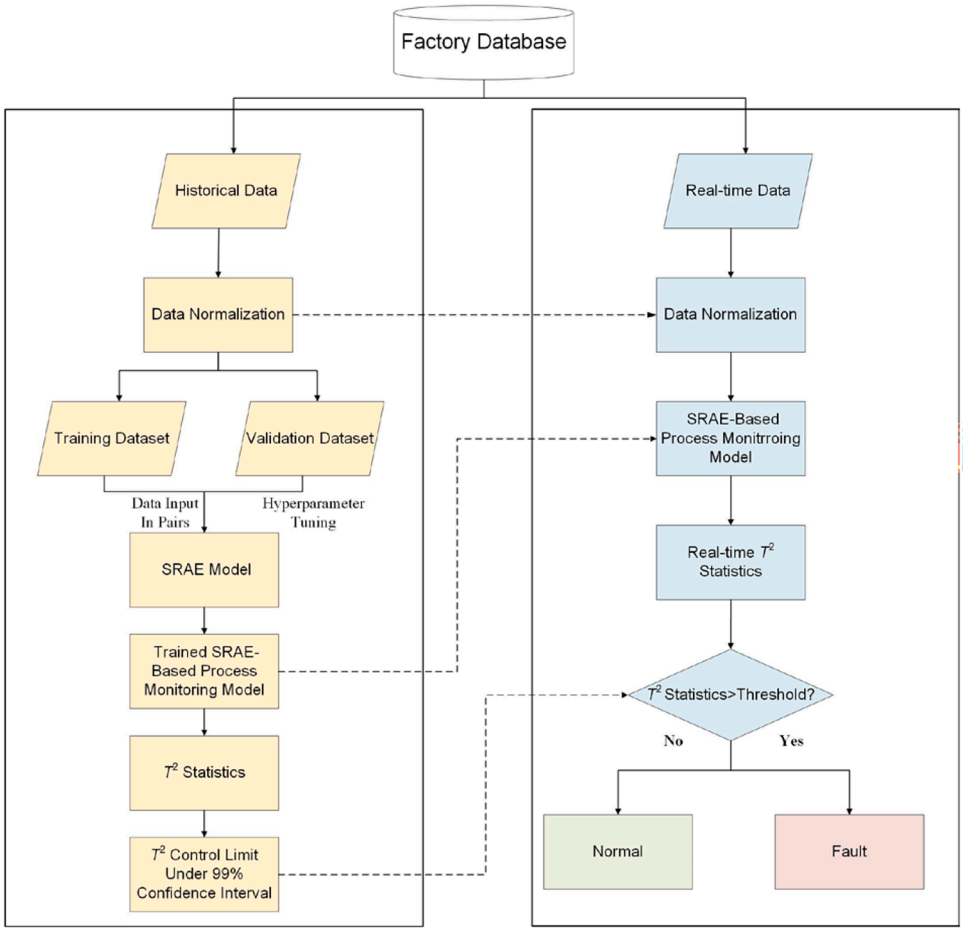


**Fig. 3.** The implementation procedures of the proposed SRAE-based process monitoring method.

**Table 1**
The pseudo code for the training of the proposed Siamese recurrent autoencoder.

| |
|---|
| # Initialize parameters |
| - Set the input data dimension: input_size<br>- Define the hidden layer sizes: hidden_size<br>- Set the output data dimension: output_size = input_size<br>- Initialize weights $\boldsymbol{W}$. and biases $\boldsymbol{b}$.<br>- Set the training parameters: learning rate, batch size, dropout rate, the number of training epochs |
| # Initialize the Siamese recurrent autoencoder (essentially two instances of the same autoencoder with shared parameters, as shown in Fig. 2) |
| - Create the autoencoder structure: Autoencoder(input_size, hidden_size, output_size) |
| # Define the loss function |
| - Define a contrastive loss function ContrastiveLoss(encoded_x1, encoded_x2, y)<br>- If $\boldsymbol{y} = 1$ (similar samples), the loss is a function of the distance between the encodings, as shown in Eq. (11)<br>- If $\boldsymbol{y} = 0$ (dissimilar samples), the loss function is adopted using the one presented in Eq. (12) |
| # Training process |
| - Divide datasets into $\boldsymbol{x}_1, \boldsymbol{x}_2$ and obtain their labels $\boldsymbol{y}$<br>For each epoch from 1 to epochs:<br>For each batch from the datasets:<br>- Extract current batch of samples x1_batch from $\boldsymbol{x}_1$ and x2_batch from $\boldsymbol{x}_2$<br>- Extract current batch of labels y_batch from $\boldsymbol{y}$<br>- Encode x1_batch and x2_batch using the SRAE to get encoded_x1 $\boldsymbol{z}_1$ and encoded_x2 $\boldsymbol{z}_1$<br>- Calculate the contrastive loss for the current batch: loss = Sum(ContrastiveLoss(encoded_x1[i], encoded_x2[i], y_batch[i])) / batch_size<br>- Use Adam optimization algorithm with learning_rate to update the weights and biases of the SRAE<br>- Compute gradients of the loss with respect to each parameter<br>- Update each parameter: param = param - learning_rate * gradient |
| # Training complete, save the model parameters |
| - Calculate $T^2$ statistics of latent representations of normal samples using Eq. (13)<br>- Calculate thresholds using Eqs. (14) and 15 |

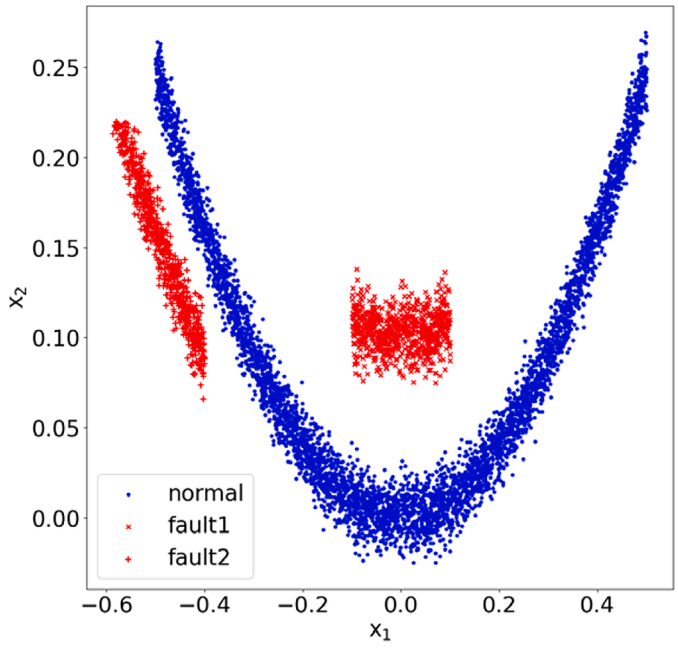


**Fig. 4.** The scatter plot of normal data and two faults of the simulation process.

improving generalization capabilities, and potentially evading local minima. For time-series data, such as those encountered in dynamic processes like chemical processes, shuffling the dataset is generally not recommended due to the inherent temporal dependencies within the data. In these cases, the order of the observations matters, as future values often depend on past values. Shuffling would disrupt these dependencies, making it difficult for the model to capture the temporal patterns and dynamics of the process. Consequently, this study opts to preserve the original sequence of time-series data collected under normal operating conditions, ensuring that the model can effectively learn and leverage these temporal dependencies. Given a time series representing normal operating conditions, the division of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as inputs is performed as follows.

The Siamese Neural Network requires two inputs, while the core characteristic of time-series data lies in the temporal order relationship among its data points. Therefore, when processing time-series data in this study, it is crucial to ensure that this sequential relationship remains intact as much as possible. The two inputs of the Siamese Neural Network can be different segments from the same time series, or segments from two related but distinct time series. For the fault detection task in this study, we adopt an approach where the first half of the time series serves as the first input $\boldsymbol{x}_1$, and the second half as the second input $\boldsymbol{x}_2$. Although there is a temporal difference between the two inputs, they both belong to normal operating conditions. By employing the Siamese Neural Network and contrastive loss, their representations in feature space are as similar or closely aligned as possible, thereby ensuring the effectiveness of the model in fault detection.

Moreover, to further study the distribution of latent features extracted by the deep learning model, a contrastive loss is proposed as the loss function to regularize the distribution of latent features,

$$loss = w_1\frac{\| \boldsymbol{x}_1 - \widetilde{\boldsymbol{x}_1}\|_2^2}{n} + w_2\frac{\| \boldsymbol{x}_2 - \widetilde{\boldsymbol{x}_2}\|_2^2}{n} + w_3\| \boldsymbol{z}_1 - \boldsymbol{z}_2 \|,\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in P \tag{11}$$

$$loss = w_1\frac{\| \boldsymbol{x}_1 - \widetilde{\boldsymbol{x}_1}\|_2^2}{n} + w_2(\max(margin - \| \boldsymbol{z}_1 - \boldsymbol{z}_2\|, 0)),\ \boldsymbol{x}_1 \in P, \boldsymbol{x}_2 \in N \tag{12}$$

where $\widetilde{\boldsymbol{x}_1}, \widetilde{\boldsymbol{x}_2}$ are the reconstruction of the decoder, $w_1, w_2, w_3$ are weights of each term of the loss function, $P$ denotes the input belonging to normal operating conditions in this fault detection task, while $N$ denotes the input belonging to abnormal operating conditions, and *margin* is a user-defined parameter.

In most existing AE-based process monitoring methods, only the reconstruction error is adopted as the training target and monitoring indicator. The extracted latent features are simply available for obtaining better reconstruction of original data, which is rarely considered in process monitoring. In practice, it is more effective and reasonable to realize process monitoring directly in the feature space. Therefore, the proposed loss function in Eqs. (11) and 12 is designed by the idea that the latent features of two inputs of the same category should be similar to each other, while the latent features of two inputs of different categories should be far away from each other. In this work, the Euclidean distance is employed to measure the similarity between latent features. The mean square error between the input and output of the AE is also retained to extract more representative features. The dominance of each item is adjusted through the weights. In this work, the contrastive loss is set as the dominant objective of model training with a weight of 0.8, as the fault detection is performed in the feature space in the SRAE model. This way, the latent features extracted from normal training data could be regularized into a small region, and $T^2$ statistics are established as follows,

$$T^2 = \mathbf{z}_t \Lambda_z^{-1} \mathbf{z}_t^T \tag{13}$$

where $\mathbf{z}_t$ is the latent feature extracted at step $t$, and $\Lambda_z$ is the covariance matrix of the eigenvectors. A control limit can be determined by the Kernel Density Estimation (KDE) method (Silverman, 1986), by which the online process monitoring can be realized. As a non-parametric estimation method, KDE can provide reliable threshold estimation without requiring any assumptions about the underlying data distribution. Given the monitoring statistics $T^2(x)$ calculated from normal operating conditions, a threshold under the 99% confidence interval can be determined as follows through KDE,

$$\rho\left(T^2(x)\right) = \frac{1}{\sqrt{2\pi}dt}\sum_{i=1}^{t} e^{-\frac{\left(T^2(x) - T^2(x_i)\right)^2}{2d^2}} \tag{14}$$

$$\int_{-\infty}^{threshold} \rho\left(T^2(x)\right) dT^2(x) = 0.99 \tag{15}$$

where $d$ is the window width of the Gaussian kernel function employed in this work, which is usually determined using Silverman's method (Silverman, 1986).

In summary, the strengths of the proposed SRAE-based process monitoring are reflected in the following aspects. Firstly, the generalizing ability of the process monitoring model is enhanced because the sample size can be expanded exponentially. Then the distribution region of the latent features of normal data is compressed by the proposed loss function, by which the fault detectability of the process monitoring method has been improved. Combined with the monitoring statistics established on the feature space, the transparency of process monitoring has also been enhanced. The effectiveness of the proposed method will be demonstrated in the next section.

### 3.2. Procedures of the SRAE-based process monitoring methods

Based on the proposed SRAE model presented in Section 3.1, the implementation procedures, which can be divided into offline training and online monitoring phases, are shown in Fig. 3. The specific description is as follows:

Offline training:

(1) Historical data from normal operating conditions are collected.
(2) Historical data are normalized to zero-mean and unit-variance.

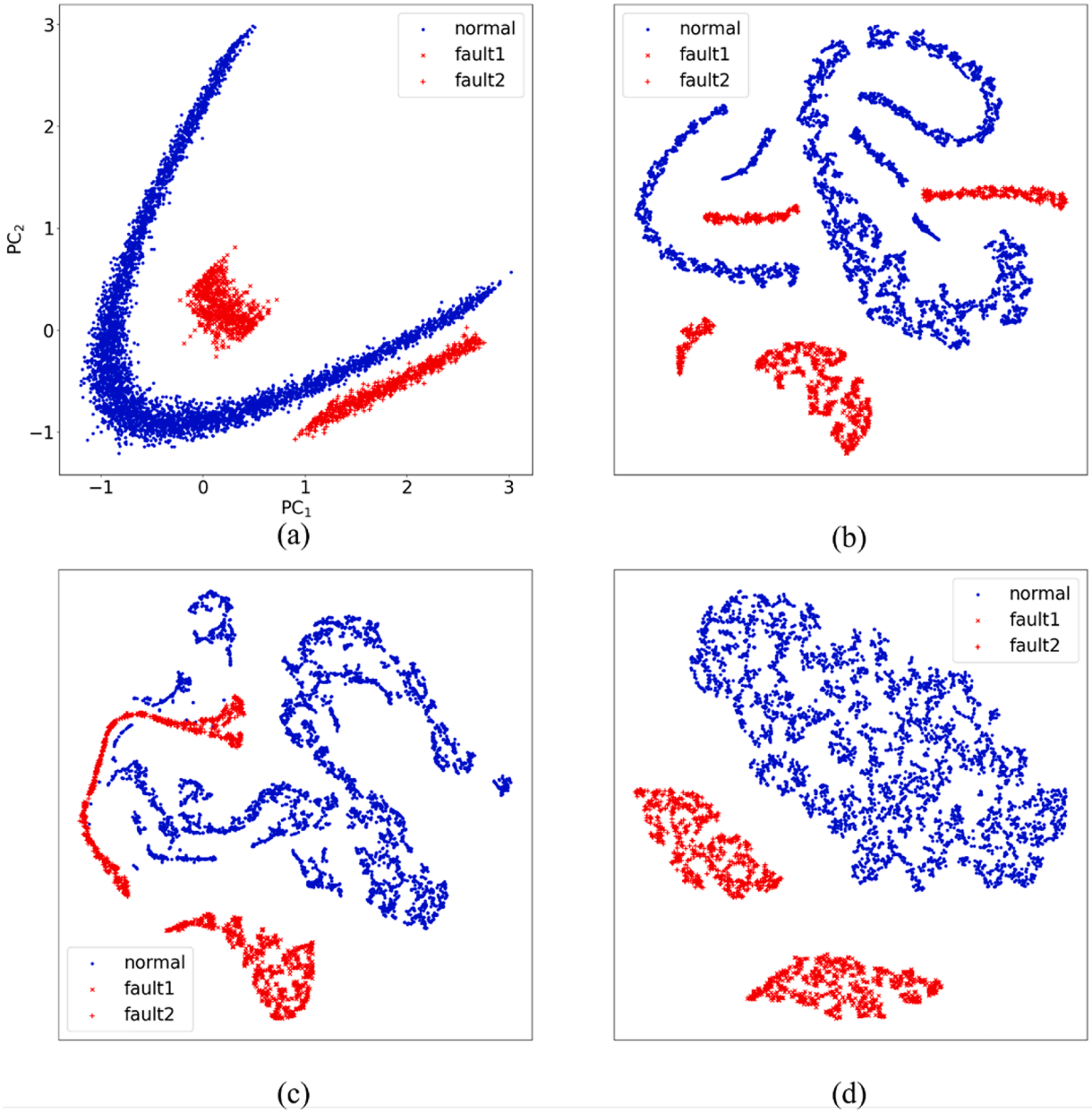


**Fig. 5.** Latent features extracted by (a) PCA; (b) AE; (c) LSTM-AE; (d) SRAE.

(3) Normalized data are divided into a training dataset and a validation dataset.
(4) Training data are input in pairs to train the proposed SRAE model.
(5) Validation data are applied for hyperparameter tuning.
(6) $T^2$ statistics are calculated using data under normal operating conditions and the corresponding control limit is determined by KDE.

Online monitoring:

(1) Real-time test data are collected and normalized based on the mean and variance of historical data.
(2) Normalized data are input into the SRAE model to obtain latent features.
(3) Real-time $T^2$ statistics are calculated and compared with the control limit to determine the operating status of the process.

To refine the description of experimental settings in the proposed SRAE model, the pseudo code for the training of the SRAE is shown in Table 1.

### 3.3. Evaluation indicators for process monitoring performance

To evaluate the process monitoring performance of different methods, two commonly used quantitative indicators, Fault Detection Rate (FDR) and False Alarm Rate (FAR), are adopted in this work,

$$FDR = \frac{TP}{TP + FN} \tag{16}$$

$$FAR = \frac{FP}{FP + TN} \tag{17}$$

where $TP$ is the abbreviation for true positives, which means the number of fault samples that are correctly detected. $FN$ is the abbreviation for false negatives and represents the number of fault samples that failed to

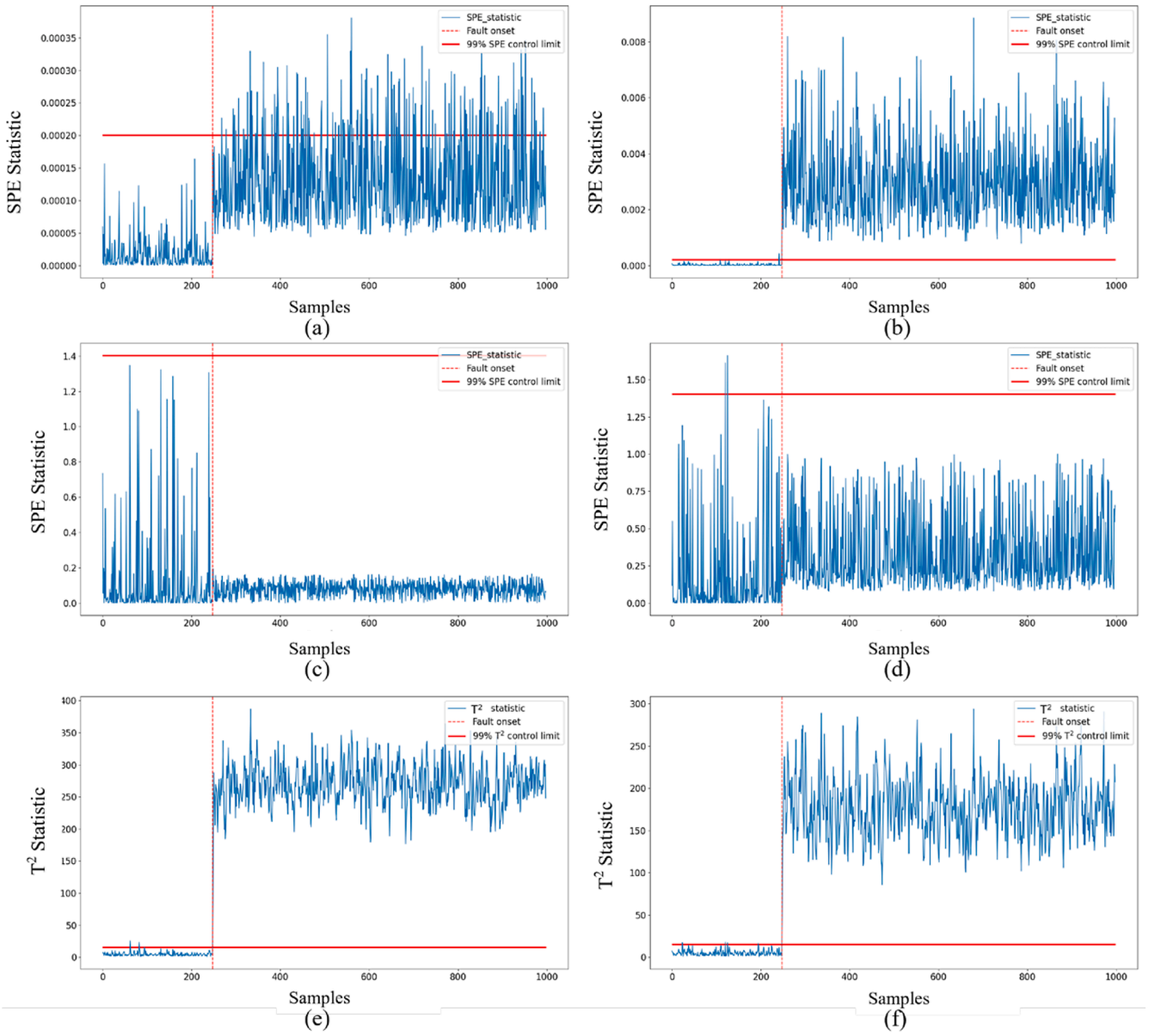


**Fig. 6.** Fault detection results for (a) AE, fault 1; (b) AE, fault 2; (c) LSTM-AE, fault 1; (d) LSTM-AE, fault 2; (e) SRAE, fault 1; (f) SRAE, fault 2.

be detected. *FP* corresponds to false positives, representing the number of normal samples that are wrongly detected as a fault, and *TN* is the abbreviation for true negatives and represents the number of normal samples that are correctly identified to be normal.

Moreover, considering that the early identification of abnormal process deviations during process operation is important, the Fault Detection Delay (FDD), which represents the time it takes for the model to detect a fault, is also used in this work.

## 4. Case studies

In this section, the proposed SRAE-based process monitoring method is tested through three case studies, including a numerical process, the TEP, and an industrial wax oil hydrogenation reactor unit.

### 4.1. A nonlinear numerical process

A bivariate nonlinear process is first investigated to illustrate the effectiveness of the proposed method. Data are simulated according to the following equation (He and Wang, 2007),

$$y = \boldsymbol{x}^2 + \varphi \tag{18}$$

where $\boldsymbol{x}$ is a random variable that conforms to a uniform distribution between $-0.5$ and 0.5, $\varphi$ denotes random noise with $N(0, 0.01^2)$. A total of 5000 normal samples are generated as the training dataset, and two datasets with 1000 samples and a fault introduced at the 250th sample are simulated as the test datasets. The distribution of normal data and two faults are shown in Fig. 4.

It can be shown that the data show significant process nonlinearity and non-Gaussianity, and the magnitude of the faults is small, even within the range of normal fluctuations, which poses a significant challenge for fault detection. The performance of the proposed method is compared with those of several other methods, including PCA, AE, and LSTM-AE. The latent features extracted by different methods are shown in Fig. 5.

As a linear method, the nonlinear and non-Gaussian features of the original dataset cannot be extracted by PCA. Therefore, the faults cannot be distinguished from normal data. For deep learning-based methods,

**Table 2**
FDR results for 18 process faults of the TEP.

| Fault No. | PCA T2 | PCA SPE | AE | SPA | VRAE | LSTM-AE | SRAE |
|---|---|---|---|---|---|---|---|
| 1 | 0.991 | 0.998 | 0.995 | 0.990 | 0.993 | 0.994 | **0.999** |
| 2 | 0.984 | 0.955 | 0.980 | 0.973 | 0.985 | 0.981 | **0.986** |
| 4 | 0.201 | 1 | 0.748 | 0.999 | 1 | 0.805 | **1** |
| 5 | 0.240 | 0.191 | 0.935 | 0.980 | 0.939 | 0.22 | **1** |
| 6 | 0.990 | 1 | 1 | 1 | 0.998 | 1 | **1** |
| 7 | 1 | 1 | 1 | 0.998 | 0.999 | 1 | **1** |
| 8 | 0.969 | 0.846 | 0.973 | 1 | 0.976 | 0.969 | 0.984 |
| 10 | 0.299 | 0.229 | 0.319 | 0.795 | 0.931 | 0.299 | 0.904 |
| 11 | 0.406 | 0.731 | 0.528 | 0.991 | 0.993 | 0.549 | 0.863 |
| 12 | 0.984 | 0.893 | 0.984 | 1 | 1 | 0.983 | **1** |
| 13 | 0.936 | 0.951 | 0.944 | 0.923 | 0.945 | 0.943 | **0.958** |
| 14 | 0.993 | 1 | 0.999 | 0.998 | 0.998 | 1 | 0.999 |
| 16 | 0.139 | 0.231 | 0.333 | 0.769 | 0.973 | 0.119 | 0.943 |
| 17 | 0.768 | 0.946 | 0.86 | 0.986 | 0.999 | 0.858 | 0.976 |
| 18 | 0.894 | 0.900 | 0.894 | 1 | 0.961 | 0.893 | 0.905 |
| 19 | 0.088 | 0.163 | 0.013 | 0.959 | 0.984 | 0.010 | 0.976 |
| 20 | 0.313 | 0.485 | 0.449 | 0.935 | 0.900 | 0.374 | 0.906 |
| 21 | 0.355 | 0.485 | 0.384 | 0.999 | 0.668 | 0.346 | 0.656 |

**Table 3**
A summary of hyperparameters needed for the training of neural network.

| Hyperparameter | Value |
|---|---|
| Number of layers | 1 |
| Maximum number of units neurons | 64 |
| Activation function | Sigmoid, Tanh |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Number of epochs | 1000 |
| Batch size | 256 |
| Dropout rate | None |

**Table 4**
FDD results for 18 process faults of the TEP (min).

| Fault No. | PCA T2 | PCA SPE | AE | VRAE | LSTM-AE | SRAE |
|---|---|---|---|---|---|---|
| 1 | 21 | 6 | 12 | 18 | 15 | **3** |
| 2 | 42 | 129 | 48 | 36 | 48 | **36** |
| 4 | 444 | 0 | 0 | 0 | 0 | **0** |
| 5 | 0 | 0 | 0 | 6 | 0 | **0** |
| 6 | 24 | 0 | 0 | 6 | 0 | **0** |
| 7 | 0 | 0 | 0 | 3 | 0 | **0** |
| 8 | 75 | 57 | 60 | 18 | 66 | 24 |
| 10 | 291 | 150 | 156 | 87 | 285 | **36** |
| 11 | 15 | 15 | 30 | 18 | 30 | **15** |
| 12 | 18 | 6 | 6 | 0 | 18 | **0** |
| 13 | 144 | 129 | 135 | 132 | 138 | **111** |
| 14 | 0 | 0 | 3 | 6 | 0 | 3 |
| 16 | 930 | 585 | 105 | 0 | 921 | 21 |
| 17 | 84 | 63 | 72 | 24 | 69 | 57 |
| 18 | 261 | 246 | 258 | 45 | 258 | 231 |
| 19 | / | 1230 | / | 39 | / | **0** |
| 20 | 255 | 258 | 252 | 225 | 252 | **195** |
| 21 | 1554 | 768 | 1482 | 0 | 1545 | 747 |

the T-distributed Stochastic Neighbor Embedding (T-SNE) is adopted for visualization, as the dimension of latent features is higher than that of original data. The faults still cannot be effectively distinguished by AE and LSTM-AE in the feature space because their main objective is to minimize the reconstruction error of training data, while the distribution of latent features is not regularized, resulting in poor fault sensitivity to faults with small magnitudes. Comparatively, the proposed method effectively compresses the feature space of normal data, and the distance between fault data and normal data can be effectively amplified in the feature space, by which the fault detection performance can be significantly improved. As shown in Fig. 4(d), both faults can be clearly distinguished from normal data. The fault detection results of AE, LSTM-AE, and the proposed SRAE are shown in Fig. 6.

It can be seen from the process monitoring charts that AE and the LSTM-AE do not perform well in detecting faults. AE-based process monitoring is successful in detecting fault 2, but fault 1 cannot be detected promptly. LSTM-AE model fails to detect both faults, as the magnitude of the faults is too small, obscured by the normal fluctuations of data. By contrast, both faults can be effectively detected by the proposed method as soon as the fault occurs. This is attributed to the compression of the region of normal data in the feature space. The results in Fig. 4 and Fig. 5 illustrate that the proposed SRAE-based process monitoring method achieves better performance than other methods.

To facilitate replication of the results, the key parameters of the models are provided below. The depth of the latent layer and the number of latent nodes of all three models are 1 and 5, respectively. The number of trainable parameters of AE is 27, and the number of trainable parameters of LSTM-AE and SRAE are both 224. It demonstrates that the performance of the model in fault detection is significantly improved with no increase in the number of parameters and model complexity. To further verified the proposed method with published studies, the results obtained by studies of Wang and He (2010), and Cheng et al. (2019) are also investigated and compared(Cheng et al., 2019; Wang and He, 2010). The Statistics Pattern Analysis (SPA) by Wang and He is known as a more advanced statistical process monitoring method, but there are still certain false alarms and missed alarms. For other published methods including Kernel Principal Component Analysis (KPCA), Sequence Autoencoder (SA), and Variational Autoencoder (VAE), the faults cannot be detected. Only the Variational Recurrent Autoencoder (VRAE) is applicable to detect the faults of this case study. Although both the proposed SRAE method in this study and VRAE in Cheng et al.'s study shows promising fault detection performance, the trainable parameters and complexity of this study are much less than VRAE. For complex industrial processes, the strengths of this study regarding fault detection performance and generalization capability can be demonstrated, as verified in other case studies.

### 4.2. *The Tennessee Eastman Process (TEP)*

The TEP is selected as the next case study because it is a well-known benchmark platform for evaluating process monitoring performance. While a total of 52 process variables are measured in simulations, in this work, 33 process variables are used because the remaining 19 composition variables are sampled less frequently. A more detailed description of the process can be found in (Downs and Vogel, 1993). Data in TEP are sampled every 3 min, and 21 types of process faults can be introduced by the simulator. In this work, the public datasets provided by Chiang et al. are adopted (Chiang et al., 2000). The training data contains 960 samples collected under normal operating conditions, and the test data consists of 18 fault datasets, where faults 3, 9, and 15 are excluded because they have been widely reported to be difficult to observe due to the small fault magnitude.

The generalization ability of the model is an issue here, as there are a large number of parameters to be trained in deep learning models with only 960 normal training samples being available. On the other hand, there are several complex features contained in data, e.g., process nonlinearity, dynamics, and non-Gaussianity, which pose significant challenges to the model to effectively detect all types of process faults. As argued earlier regarding the strengths of the proposed SRAE-based process monitoring method, we expect notable improvement in the process monitoring performance.

The process monitoring results for all 18 faults are summarized in Table 2. Among these methods, the PCA performs the worst because it is insufficient to extract various features of the data, primarily because it is a linear transformation. As expected, the performance of deep learning-based methods is better than PCA. The hyperparameters needed and adopted for the training of deep learning models in this work are summarized in Table 3. In this case study, the depth of all deep learning

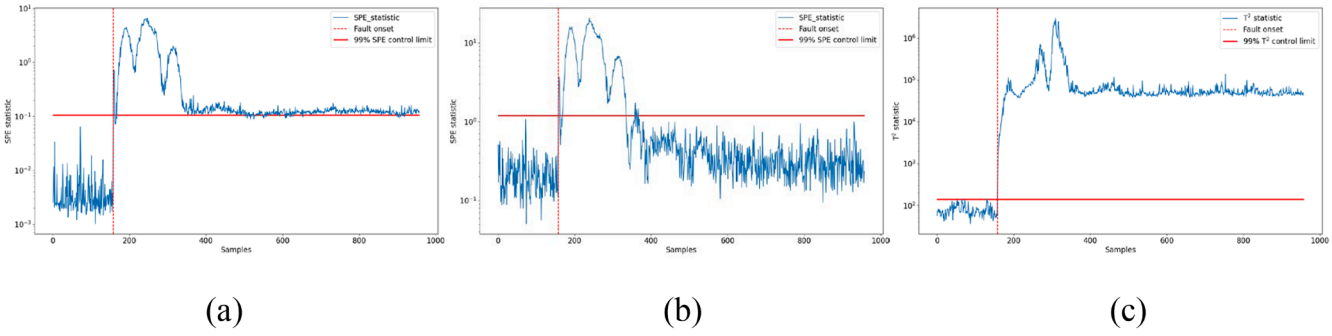


**Fig. 7.** Process monitoring charts for fault 5: (a) AE, (b) LSTM-AE, (c) SRAE.

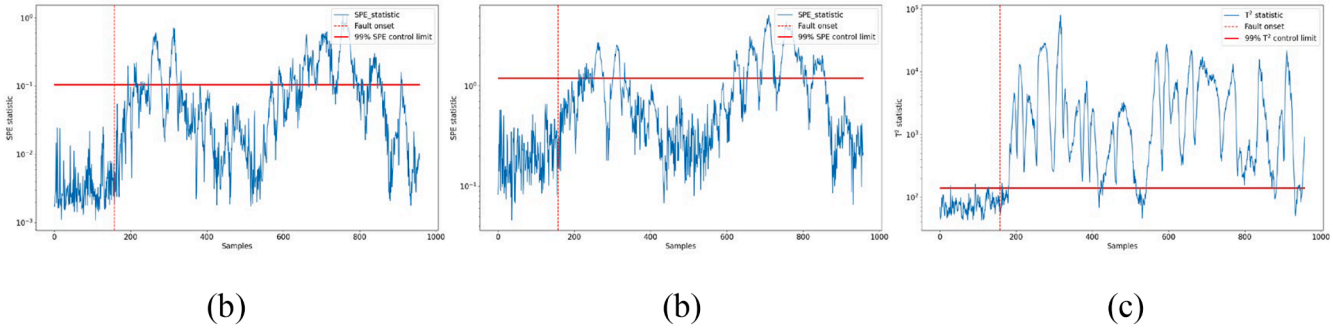


**Fig. 8.** Process monitoring charts for fault 10: (a) AE, (b) LSTM-AE, (c) SRAE.

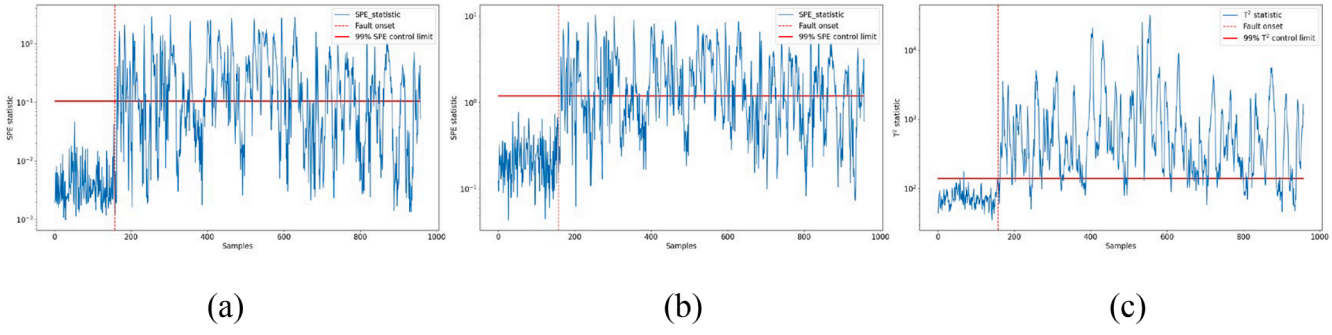


**Fig. 9.** Process monitoring charts for fault 11: (a) AE, (b) LSTM-AE, (c) SRAE.

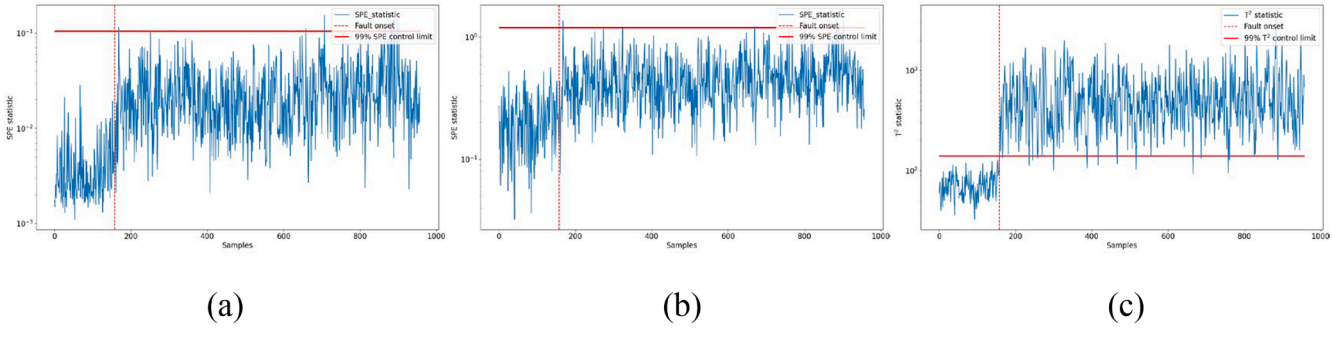


**Fig. 10.** Process monitoring charts for fault 19: (a) AE, (b) LSTM-AE, (c) SRAE.

models is 1, and the number of latent nodes is 64. The number of trainable parameters in AE, LSTM-AE, and SRAE are 4321, 38,024, and 38,024, respectively. Although there are many more learnable parameters in LSTM-AE, its performance in fault detection is similar to AE. The reason for this result can be attributed to the generalization ability of the models, as there are only 960 samples available to train the model. It is natural that the model will overfit, indicating that the model cannot be generalized readily to validation datasets and test datasets. As a result, the mean square error of normal data in the validation dataset will increase, ending up with a higher control limit. This adversely affects fault

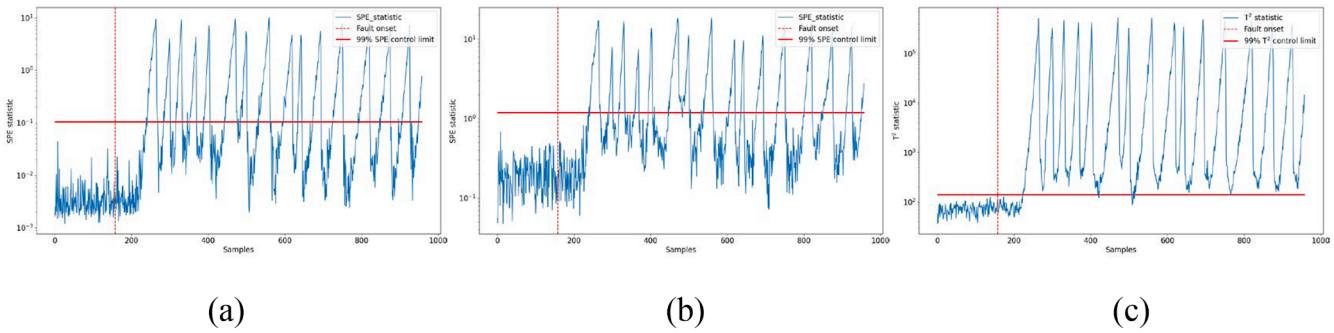


**Fig. 11.** Process monitoring charts for fault 20: (a) AE, (b) LSTM-AE, (c) SRAE.

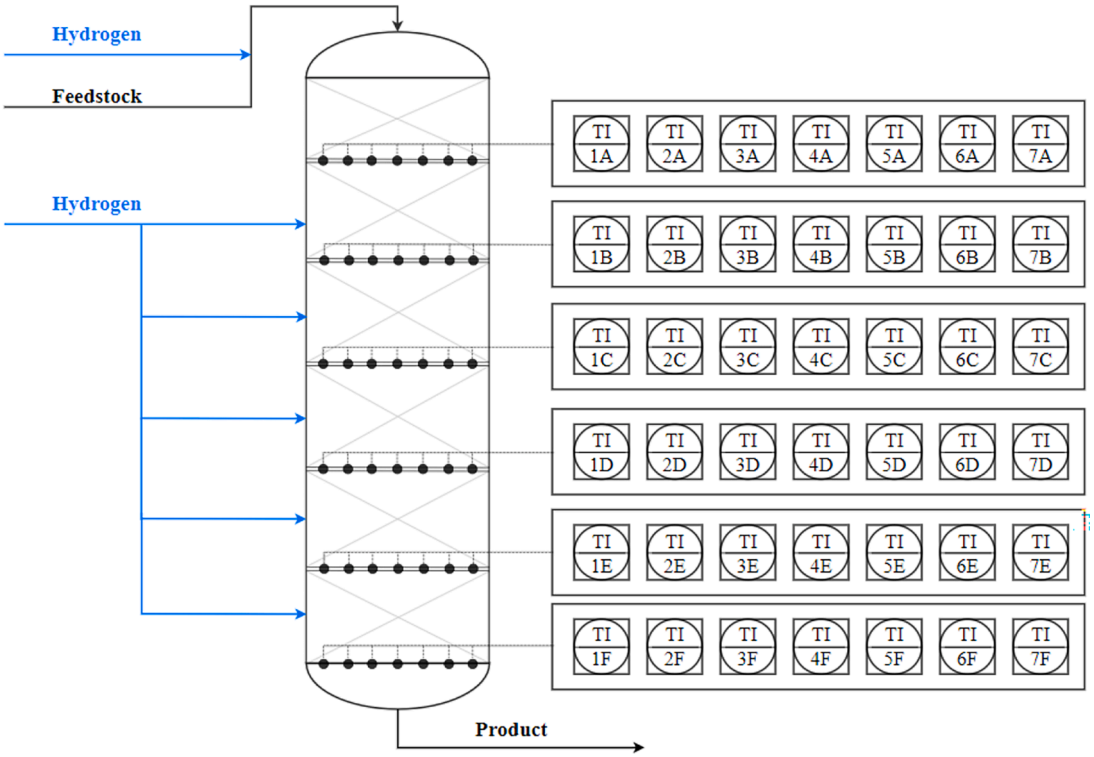


**Fig. 12.** Structure and sensor placement of the wax oil hydrogenation reactor.

detectability, leading to a lower FDR. On the other hand, the proposed SRAE achieves the best performance (at least as good) in almost all 18 faults. It is worth noting that the same number of parameters and training samples as the LSTM-AE has been adopted by the proposed SRAE method, but the SRAE shows a much better generalization ability. Benefitting from the multiple input form, the sample size for model training has been widely expanded, which contributes to the better performance of the proposed SRAE-based method.

The effectiveness of the proposed method can also be verified through the FDD. The FDD results for the 18 faults are compared in Table 4. It is demonstrated that the proposed SRAE shows superior performance in monitoring all these faults at their early stage. Especially for a fault with a minor magnitude, for example, fault 19, the proposed method can detect it as soon as the fault occurs, while other methods fail even to detect it. Meanwhile, the FAR of SRAE for fault 19 is 0, which indicates that the fault detection result is reliable. The process monitoring charts of AE, LSTM-AE, and SRAE for faults 5, 10, 11, 19, and 20 are shown in Figs. 7, 8, 9, 10, and 11. These faults have either small magnitudes or trigger the response of the control system. As a result, they cannot be effectively detected by conventional methods. For faults 10, 11, and 20, the monitoring statistics of AE and LSTM-AE fluctuate around their control limits, which will greatly interfere with operators' judgment of process operating status. Comparatively, benefitting from its special multi-input structure and the contrastive loss, the proposed SRAE-based process monitoring method shows powerful ability in the extraction of discriminative features in the latent space when the faults occur. As shown in the process monitoring charts, the fault detectability of the proposed method is significantly improved compared with other methods. Moreover, it can also be observed from the process monitoring charts that there are barely any false alarms triggered by the SRAE model, which further strengthens its application prospects in monitoring practical industrial processes. We further verified the proposed method by comparing with published studies. Similarly, the fault detection performance of PCA, KPCA, AE, SA, and VAE is significantly worse than that of the proposed SRAE. Regarding SPA and VRAE, the FDR results are compared in Table 1. It can be shown that the proposed SRAE performs better than SPA and VRAE in most faults, especially for the first 14 faults. The strengths of the SRAE can also be further verified through the FDD results shown in Table 3.

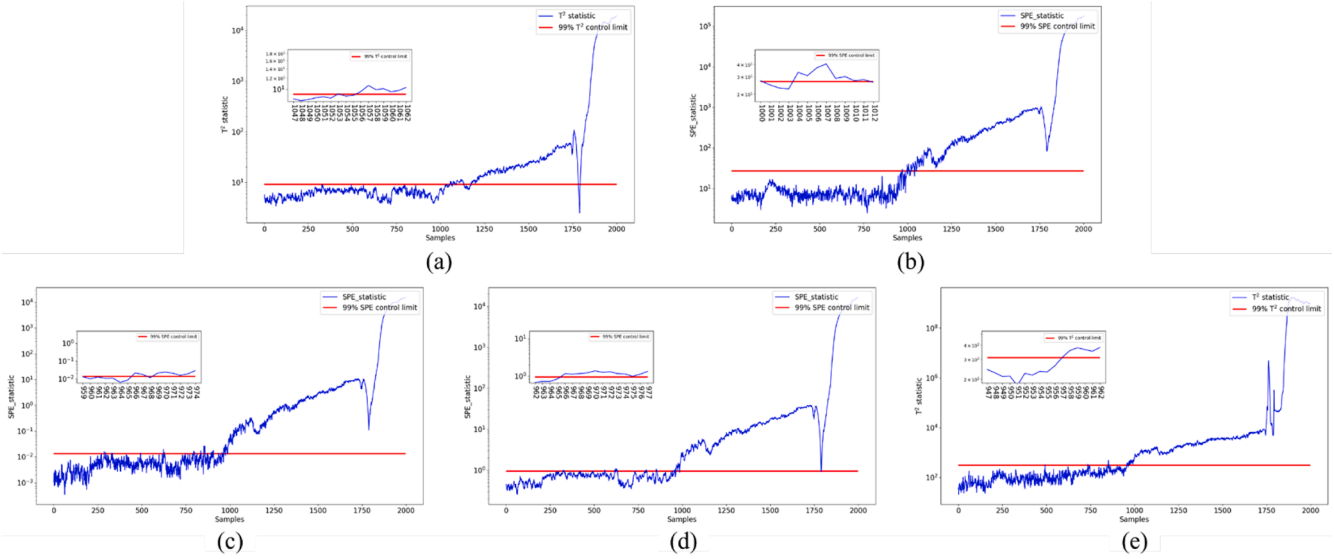


**Fig. 13.** Process monitoring charts for the temperature spike fault: (a) PCA $T^2$, (b) PCA SPE, (c) AE, (d) LSTM-AE, (e) SRAE.

**Table 5**
Fault alarm time results of different methods.

| Method | DCS | PCA $T^2$ | PCA SPE | AE | LSTM-AE | SRAE |
|---|---|---|---|---|---|---|
| Fault alarm time | 1209 | 1056 | 1004 | 969 | 966 | **957** |

### 4.3. The wax oil hydrogenation reactor unit

For the third case study, industrial process data from a wax oil hydrogenation reactor unit of an actual chemical factory in China are further applied to validate the effectiveness of the proposed method. There are 42 temperature measurement points distributed at 6 different reactor bed heights, which are shown in Fig. 12. A total of 8000 samples with a sampling frequency of 1 min collected from normal operating conditions are employed to train the models, and then the fault detection performance of different models is compared through a test dataset, which contains 2000 samples. According to the operating records, there is a temperature spike fault in the test dataset. The temperature spike fault will result in serious damage to the product quality and process safety. Therefore, it is of great significance to detect the fault at its early stage.

In this case study, the depth of all deep learning models is 1, and the number of latent nodes is 64. The number of trainable parameters in AE, LSTM-AE, and SRAE is 5482, 45,368, and 45,368. The process monitoring chart for this temperature spike fault is shown in Fig. 13. Meanwhile, the fault alarm time of different methods is summarized in Table 5. It can be shown that the alarm is triggered by the Distributed Control System (DCS) at the 1209th sample. Comparatively, this fault can be detected much earlier by data-driven process monitoring methods, which could provide operators more time to take action to minimize the impact of this fault. Among these methods, PCA performs worse than deep learning methods because of its lack of ability to extract complex nonlinear features of industrial process data. For deep learning methods, the fault is detected by the proposed SRAE method at the 957th sample with scarce false alarms, which is 9 min and 12 min earlier than AE and LSTM-AE, respectively. The results demonstrate the better fault detection performance of the proposed SRAE method. Given the multi-input structure and the contrastive loss in latent space, the generalization ability and fault detectability of the deep learning method can be effectively improved with equivalent parameters to the baseline method. This case study further illustrates the industrial application prospect of the proposed method in monitoring chemical processes. The effectiveness of the SRAE demonstrated from this work may benefit researchers working on related topics, such as industrial process control, fault detection and diagnosis, and predictive maintenance.

## 5. Conclusions

This work presents a highly generalized deep learning model, named SRAE, for process monitoring. With the multiple-input structure of the Siamese network, the sample size of the training dataset is significantly expanded, by which the generalization performance of the deep learning model can be improved. Moreover, a contrastive loss is proposed to measure the distance between latent features extracted from the two inputs of the SNN, by which the region of normal data can be effectively compressed in the feature subspace. Accordingly, the fault detection performance of the model is enhanced through monitoring solely the feature subspace, which also brings a certain degree of transparency to the deep learning-based process monitoring methods. Overall, the proposed method provides a useful strategy to motivate the development of further studies on the generalization performance of deep learning models for process monitoring. The strengths of this work in fault detection also demonstrate potential applicability in process industries, where it can be interfaced or integrated with the factory's distributed control system to significantly enhance the overall reliability and robustness of the monitoring infrastructure, ultimately contributing to more effective predictive maintenance practices within the factory. In future work, we plan to extend the proposed method to multiple sources of industrial process data using a pseudo-Siamese network, and historical fault data can also be applied by the proposed method to further improve the process monitoring performance.

## CRediT authorship contribution statement

**Cheng Ji:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Fangyuan Ma:** Validation, Resources, Methodology, Data curation. **Jingde Wang:** Supervision, Project administration, Formal analysis. **Wei Sun:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization. **Ahmet Palazoglu:** Writing – review & editing, Supervision.

## 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.

## Data availability

Data will be made available on request.

## Acknowledgments

The authors would like to acknowledge the financial support from the National Natural Science Foundation of China (Grant # 22278018).

## References

Alauddin, M., Khan, F., Imtiaz, S., Ahmed, S., 2018. A bibliometric review and analysis of data-driven fault detection and diagnosis methods for process systems. Ind. Eng. Chem. Res. 57 (32), 10719–10735.
Alauddin, M., Khan, F., Imtiaz, S., Ahmed, S., Amyotte, P., Vanberkel, P., 2024. A hierarchical Bayesian network-based semi-mechanistic model for handling data variabilities in dynamical process systems. Comput. Chem. Eng. 185.
Amin, M.T., Imtiaz, S., Khan, F., 2018. Process system fault detection and diagnosis using a hybrid technique. Chem. Eng. Sci. 189, 191–211.
Amin, M.T., Khan, F., Ahmed, S., Imtiaz, S., 2020. A novel data-driven methodology for fault detection and dynamic risk assessment. Can. J. Chem. Eng. 98 (11), 2397–2416.
Amin, M.T., Khan, F., Ahmed, S., Imtiaz, S., 2021. Risk-based fault detection and diagnosis for nonlinear and non-Gaussian process systems using R-vine copula. Process Saf. Environ. Prot. 150, 123–136.
Bhakte, A., Kumar Kumawat, P., Srinivasan, R., 2024. Explainable AI methodology for understanding fault detection results during Multi-Mode operations. Chem. Eng. Sci. 299.
Bi, X., Qin, R., Wu, D., Zheng, S., Zhao, J., 2022. One step forward for smart chemical process fault detection and diagnosis. Comput. Chem. Eng. 164.
Bi, X., Wu, D., Xie, D., Ye, H., Zhao, J., 2023. Large-scale chemical process causal discovery from big data with transformer-based deep learning. Process Saf. Environ. Prot. 173, 163–177.
Bi, X., Zhao, J., 2021. A novel orthogonal self-attentive variational autoencoder method for interpretable chemical process fault detection and identification. Process Saf. Environ. Prot. 156, 581–597.
Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R., 1993. Signature verification using a "Siamese" time delay neural network. Adv. Neural Inf. Process Syst. 7 (04), 669–688.
Cheng, F., He, Q.P., Zhao, J., 2019. A novel process monitoring approach based on variational recurrent autoencoder. Comput. Chem. Eng. 129, 106515.
Chiang, L.H., Russell, E.L., Braatz, R.D., 2000. Fault Detection and Diagnosis in Industrial Systems. Springer Science & Business Media.
Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17 (3), 245–255.
Galagedarage Don, M., Khan, F., 2019. Process Fault Prognosis Using Hidden Markov Model–Bayesian Networks Hybrid Model. Ind. Eng. Chem. Res. 58 (27), 12041–12053.
Gharahbagheri, H., Imtiaz, S.A., Khan, F., 2017. Root Cause Diagnosis of Process Fault Using KPCA and Bayesian Network. Ind. Eng. Chem. Res. 56 (8), 2054–2070.
He, Q.P., Wang, J., 2007. Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. IEEE Trans. Semicond. Manuf. 20 (4), 345–354.
He, Q.P., Wang, J., 2018. Statistical process monitoring as a big data analytics tool for smart manufacturing. J. Process Control 67, 35–43.
Ji, C., Ma, F., Wang, J., Sun, W., Zhu, X., 2022. Statistical method based on dissimilarity of variable correlations for multimode chemical process monitoring with transitions. Process Saf. Environ. Prot. 162, 649–662.
Ji, C., Sun, W., 2022. A Review on Data-Driven Process Monitoring Methods: characterization and Mining of Industrial Data. Processes 10 (2), 335.
Jia, M., Hu, J., Liu, Y., Gao, Z., Yao, Y., 2023. Topology-Guided Graph Learning for Process Fault Diagnosis. Ind. Eng. Chem. Res. 62 (7), 3238–3248.
Jiang, W., Hong, Y., Zhou, B., He, X., Cheng, C., 2019. A GAN-Based Anomaly Detection Approach for Imbalanced Industrial Time Series. IEEE Access 7, 143608–143619.
Khakzad, N., Khan, F., Amyotte, P., 2011. Safety analysis in process facilities: comparison of fault tree and Bayesian network approaches. Reliab. Eng. Syst. Saf. 96 (8), 925–932.
Khakzad, N., Khan, F., Amyotte, P., 2013. Risk-based design of process systems using discrete-time Bayesian networks. Reliab. Eng. Syst. Saf. 109, 5–17.
Koch, G., Zemel, R., Salakhutdinov, R., 2015. Siamese Neural Networks For One-shot Image Recognition. ICML deep learning workshop, Lille, France, pp. 1–30. Lille, France.
Kresta, J.V., MacGregor, J.F., Marlin, T.E., 1991. Multivariate statistical monitoring of process operating performance. Can. J. Chem. Eng. 69 (1), 35–47.
Ku, W., Storer, R.H., Georgakis, C., 1995. Disturbance detection and isolation by dynamic principal component analysis. Chemom. Intell. Lab. Syst. 30 (1), 179–196.
Kumar, A., Bhattacharya, A., Flores-Cerrillo, J., 2020. Data-driven process monitoring and fault analysis of reformer units in hydrogen plants: industrial application and perspectives. Comput. Chem. Eng. 136, 106756.
Lee, J.-M., Yoo, C., Choi, S.W., Vanrolleghem, P.A., Lee, I.-B., 2004. Nonlinear process monitoring using kernel principal component analysis. Chem. Eng. Sci. 59 (1), 223–234.
Li, G., Qin, S.J., Zhou, D., 2014. A New Method of Dynamic Latent-Variable Modeling for Process Monitoring. IEEE Trans. Ind. Electron. 61 (11), 6438–6445.
Li, Y., Ma, F., Ji, C., Wang, J., Sun, W., 2022. Fault detection method based on global-local marginal discriminant preserving projection for chemical process. Processes 10, 122.
Liu, L., Zhao, H., Hu, Z., 2022. Graph dynamic autoencoder for fault detection. Chem. Eng. Sci. 254.
Liu, Y., Jafarpour, B., 2024. Graph attention network with Granger causality map for fault detection and root cause diagnosis. Comput. Chem. Eng. 180.
Lv, F., Bi, X., Xu, Z., Zhao, J., 2024. Causality-embedded reconstruction network for high-resolution fault identification in chemical process. Process Saf. Environ. Prot. 186, 1011–1033.
Lv, H., Chen, J., Pan, T., Zhang, T., Feng, Y., Liu, S., 2022. Attention mechanism in intelligent fault diagnosis of machinery: a review of technique and application. Measurement 199.
Ma, F., Ji, C., Xu, M., Wang, J., Sun, W., 2023. Spatial Correlation Extraction for Chemical Process Fault Detection Using Image Enhancement Technique aided Convolutional Autoencoder. Chem. Eng. Sci.
Modak, N.M., Lobos, V., Merigó, J.M., Gabrys, B., Lee, J.H., 2020. Forty years of computers & chemical engineering: a bibliometric analysis. Comput. Chem. Eng. 141.
Nomikos, P., MacGregor, J.F., 1995. Multivariate SPC charts for monitoring batch processes. Technometrics 37 (1), 41–59.
Pistikopoulos, E.N., Barbosa-Povoa, A., Lee, J.H., Misener, R., Mitsos, A., Reklaitis, G.V., Venkatasubramanian, V., You, F., Gani, R., 2021. Process systems engineering – The generation next? Comput. Chem. Eng. 147.
Qin, S.J., 2012. Survey on data-driven industrial process monitoring and diagnosis. Annu. Rev. Control 36 (2), 220–234.
R.Isermann, P.Ballé, 1997. Trends in the application of model-based fault detection and diagnosis of technical processes. Control Eng. Pract. 5 (5), 709–719.
Sakurada, M., Yairi, T., 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis—MLSDA'14, Gold Coast, Australia, 2 December 2014, pp. 4–11.
Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. Chapman & Hall, London, UK.
Wang, J., He, Q.P., 2010. Multivariate statistical process monitoring based on statistics pattern analysis. Ind. Eng. Chem. Res. 49 (17), 7858–7869.
Yin, S., Ding, S.X., Xie, X., Luo, H., 2014. A review on basic data-driven approaches for industrial process monitoring. IEEE Trans. Ind. Electron. 61 (11), 6418–6428.
Yu, H., Khan, F., Garaniya, V., 2015a. Modified Independent Component Analysis and Bayesian Network-Based Two-Stage Fault Diagnosis of Process Operations. Ind. Eng. Chem. Res. 54 (10), 2724–2742.
Yu, H., Khan, F., Garaniya, V., 2015b. A probabilistic multivariate method for fault diagnosis of industrial processes. Chem. Eng. Res. Des. 104, 306–318.
Yu, W., Zhao, C., 2020. Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. IEEE Trans. Control Syst. Technol. 28 (3), 1083–1091.
Zhang, S., Qiu, T., 2022. A dynamic-inner convolutional autoencoder for process monitoring. Comput. Chem. Eng. 158, 107654.
Zhou, K., Tong, Y., Li, X., Wei, X., Huang, H., Song, K., Chen, X., 2023. Exploring global attention mechanism on fault detection and diagnosis for complex engineering processes. Process Saf. Environ. Prot. 170, 660–669.