兰州理工大学

科研成果汇总

学　　号：	221081104003
研 究 生：	张妍
导　　师：	曹洁 教授　　赵小强 教授
研究方向：	故障检测与质量预测
论文题目：	基于数据驱动的间歇过程故障检测与质量预测方法研究
学　　科：	控制理论与控制工程
学　　院：	电气工程与信息工程学院
入学时间：	2022 年 9 月

目 录

.

图书馆 文献检索报告

兰州理工大学图书馆 LUTLIB 报告编号：R2025-0232

机构：兰州理工大学

姓名：张妍 [221081104003]

著者要求对其在国内外学术出版物所发表的科技论著被以下数据库收录情况进行查证。

检索范围：

- 科学引文索引（Science Citation Index Expanded）： 1900年-2025年

检索结果：

检索类型	数据库	年份范围	总篇数	第一作者篇数
SCI-E 收录	SCI-EXPANDED	2023 - 2025	4	4

End

委托人声明：

本人委托兰州理工大学图书馆查询论著被指定检索工具收录情况，经核对检索结果，附件中所列文献均为本人论著，特此声明。

作　　者（签字）：张妍

完 成 人（签字）：安宗玉

完 成 日 期 ：2025年3月24日

完成单位（盖章）：兰州理工大学图书馆信息咨询与学科服务部

（本检索报告仅限校内使用）

图书馆

文献检索报告
SCI-E 收录

兰州理工大学图书馆 LUTLIB

报告编号：R2025-0232 SCI-E 收录

数据库：科学引文索引 (Science Citation Index Expanded) 时间范围：2023年至2025年	作者姓名：张妍 作者单位：兰州理工大学	检索人员： 检索日期：2025年3月24日

检索结果：被 SCI-E 收录文献 4 篇

#	作者	地址	标题	来源出版物	文献类型	入藏号
1	**Zhang, Y**; Cao, J; Zhao, XQ; Hui, YY	[Zhang, Yan; Cao, Jie; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Zhang, Yan; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Key Lab Gansu Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.; [Cao, Jie] Mfg Informatizat Engn Res Ctr Gansu Prov, Lanzhou, Peoples R China.	Batch process quality prediction based on denoising autoencoder-spatial temporal convolutional attention mechanism fusion network	*APPLIED INTELLIGENCE* 2025, 55 (6): 515.	J Article	WOS:001439610700002
2	**Zhang, Y**; Cao, J; Zhao, XQ; Hui, YY	[Zhang, Yan; Cao, Jie; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Zhang, Yan; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Key Lab Gansu Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.; [Cao, Jie] Mfg Informatizat Engn Res Ctr Gansu Prov, Lanzhou, Peoples R China.	Deep quality-related stacked isomorphic autoencoder for batch process quality prediction	*MEASUREMENT SCIENCE AND TECHNOLOGY* 2024, 35 (11): 116202.	J Article	WOS:001287502400001
3	**Zhang, Y**; Zhao, XQ; Cao, J; Hui, YY	[Zhang, Yan; Zhao, Xiaoqiang; Cao, Jie; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn,	Batch process monitoring based on sequential phase division multiway sparse weighted neighborhood preserving embedding	*MEASUREMENT SCIENCE AND TECHNOLOGY* 2024, 35 (3): 035704.	J Article	WOS:001121396700001

		Lanzhou, Peoples R China.; [Zhang, Yan; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Key Lab Gansu Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.; [Cao, Jie] Mfg Informatizat Engn Res Ctr Gansu Prov, Lanzhou, Peoples R China.				
4	**Zhang, Y**; Cao, J; Zhao, XQ; Hui, YY	[Zhang, Yan; Cao, Jie; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou 730050, Peoples R China.; [Cao, Jie] Mfg Informatizat Engn Res Ctr Gansu Prov, Lanzhou 730050, Peoples R China.; [Zhang, Yan; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Key Lab Gansu Adv Control Ind Proc, Lanzhou 730050, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou 730050, Peoples R China.	Nonlinear multiphase batch process monitoring and quality prediction using multi-way concurrent locally weighted projection regression	***CHEMOMETRICS AND INTELLIGENT LABORATORY SYSTEMS*** 2023, 240: 104922.	J Article	WOS:001049378000001
					合计	4

Applied Intelligence (2025) 55:515
https://doi.org/10.1007/s10489-025-06368-7

Batch process quality prediction based on denoising autoencoder-spatial temporal convolutional attention mechanism fusion network

Yan Zhang[1,2] · **Jie Cao**[1,4] · **Xiaoqiang Zhao**[1,2,3] · **Yongyong Hui**[1,2,3]

Accepted: 14 February 2025

Abstract
In batch processes, the accurate prediction of quality variables plays a crucial role in smooth production and quality control. However, various sources of noise in the production environment cause abnormal data fluctuations that deviate from the real value. Coupled with the dynamic nonlinearity of batch processing and the complex spatiotemporal relationship of variables, which greatly increase the difficulty of prediction and pose a severe challenge to prediction performance. Therefore, a denoising autoencoder-Spatial Temporal Convolution Attention Fusion Network (DAE-STCAFN) prediction method is proposed. Firstly, combining DAE and maximum information coefficient (MIC), multi-level data features are extracted to prepare high-quality input data for the quality prediction model. DAE is used to denoise the original data, and relevant variables are selected through MIC. Then, an augmented matrix is constructed to eliminate the autocorrelation of the selected variables in the time series. Secondly, a spatial temporal convolutional attention fusion mechanism is created to extract the spatial temporal fusion features between the input and output variable sequences. Thirdly, to further enhance the learning ability of the model, a batch attention module is constructed to automatically learn the relationship among sample in small batch. Finally, experiments were carried out on the simulation platform of penicillin fermentation and hot tandem rolling process. In the prediction process of penicillin concentration, RMSE and MAE of the proposed method were 0.0099 and 0.0077, respectively. In the prediction of strip thickness, the RMSE and MAE are 0.0008 and 0.0003 respectively. The results show that the proposed method is effective both in simulation experiment and in actual industrial production in terms of prediction accuracy, stability and generalization ability.

Keywords Batch processes · Quality prediction · Denoising-Autoencoder · Maximum Information Coefficient · Spatiotemporal convolutional attention

1 Introduction

As one of the main production ways in modern industry, batch process is widely used in the fields of fine chemicals, biotechnology, pharmaceuticals and specialty polymers. Unlike continuous process, batch process data have the characteristic of batch, the production process is cyclical and dynamic, and the product quality is easily affected by the factors such as the production environment and the state of the equipment [1, 2]. With the increasing emphasis on batch process safety and product quality, as an important strategy to ensure production reliability, quality prediction has become a hot research topic in both academia and industry [3, 4].

In recent years, data-driven methods [5, 6] have gradually attracted the attentions with the wide applications of big

✉ Jie Cao
zhyan0423@163.com

1 College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China

2 Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou, China

3 National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China

4 Manufacturing Informatization Engineering Research Center of Gansu Province, Lanzhou, China

Published online: 08 March 2025

data and the improvement of computing power. Data-driven quality prediction methods can be broadly categorized into multivariate statistical methods and deep learning methods [7–9]. The multivariate statistical methods mainly include principal component analysis [10], partial least squares [11] and other methods. Considering the nonlinear characteristic of real data, further kernel-based statistical learning methods [12–14] and support vector machine [15] are widely used. Compared to multivariate statistical methods, deep neural networks(DNN) [16, 17] have multiple hidden layers, which can fit nonlinear systems through hierarchical nonlinear characterization and can handle complex network training problems through unsupervised pretraining and supervised fine-tuning. Typical deep learning methods include Deep Belief Network (DBN), Stacked Autoencoder (SAE). For the past few years, deep learning using multilayer nonlinear mapping has achieved good results in the study of quality prediction for real industrial production. Shang et al. [18] proposed an industrial soft sensor based on Deep Belief Network (DBN) to predict 95% of the cut points of heavy diesel fuel. Yan et al. [19] proposed a denoising autoencoder (DAE) integrated with neural network method to improve the soft sensor's prediction performance and robustness. In order to analyze the relationship between input and target variables, Yuan et al. [20] proposed a variable weighted stacked autoencoder (VW-SAE) to achieve a linear correlation metric between input and target variables to extract the features related to the output. Liu et al. [21] proposed a new stacked neighborhood maintaining autoencoder for extracting hierarchical neighborhood maintaining features in the accurate quality prediction for industrial hydrocracking process. Sun et al. [22] proposed a gated stacked target correlation autoencoder (GSTAE) that utilized gated neurons to extract and control the features of different hidden layers to improve the soft measurement performance.

However, the above methods are mainly used for static network modeling. Industrial processes have complex nonlinear dynamic characteristics due to the presence of feedback control as well as complex physicochemical reactions and dynamic noise. Therefore, industrial process data can be viewed as time series with highly nonlinear temporal correlation. In order to capture the temporal correlation in production data, Recurrent Neural Network (RNN) was proposed to achieve the transfer of information so as to effectively deal with simple time series [23]. For long time series modeling, the gradient vanishing problem occurs by using recurrent neural networks. In order to solve the gradient vanishing problem, long-short-term memory network (LSTM) was proposed [24]. LSTM has special memory units and gating mechanisms, which can utilize past outputs and current inputs to process long sequence data and solve the gradient vanishing problem. However, for multivariate time series, LSTM cannot focus on different variables at different time steps. In industrial process quality prediction, the data samples of previous moment always have various impacts on the current data. Yuan et al. [25] proposed a variable attention mechanism to adaptively select the variables related to quality variables for dynamic modeling of industrial processes. Xiang et al. [26] proposed a long and short-term memory neural network with weight amplification by using the attention mechanism for predicting gear life. For highly redundant data in industry, Ren et al. [27] proposed a wide-deep sequence model that integrated LSTM with a multilayer perceptron network for quality prediction. Yuan et al. [28] further explored and constructed a spatial–temporal attention-based LSTM network for soft sensor modeling, which improved the prediction performance by comprehensively considering the spatial–temporal mass interactions. These methods have certain applicability, end-to-end prediction comes at the expense of a certain level of model interpretability. A recurrent long short-term memory networks (RNN-LSTM) architecture is proposed for early diagnosis of diabetic retinopathy [29]. LSTM has some advantages in dealing with long-term dependency problems, but it has some problems such as high computational resource consumption and easy layer disappearance or explosion. Through in-depth analyses of industrial soft sensor model, we find the following challenges that may be encountered in practical applications: 1) The problem of data quality: the accuracy and reliability of industrial soft sensor depend heavily on data quality [30]. The problems such as missing data, noise interference, outliers, etc. may seriously affect the prediction accuracy of soft sensor models. 2) Insufficient model generalization ability: Although the soft sensor model performs well on the training set, it limits the generalization ability on new data sets. Especially when dealing with complex and changing industrial processes, the model may have the difficulty in adapting to new data samples. 3) Difficulty in feature selection: feature selection and feature extraction are crucial steps when constructing soft sensor model [31, 32]. However, due to the complexity of industrial processes, certain key features may be difficult to capture, which further increases the difficulty of feature selection and negatively affects model performance.

Based on the above analysis, this paper proposes a batch quality prediction method based on the fusion network of denoising autoencoder and spatiotemporal convolutional attention mechanism. The main innovations are as follows:

1) We innovatively combined DAE and MIC to provide high-quality feature input for subsequent model training through data noise reduction and high-correlation feature extraction. At the same time, in order to effectively eliminate the autocorrelation between variables, we construct an augmentation matrix, which enhances the robustness of the model.

2) In order to capture the dynamic characteristics of batch processes more accurately, we propose a new spatial temporal feature fusion strategy. Specifically, we input spatial convolution attention and temporal convolution attention into the encoder and decoder of the Transformer network, respectively. The multi-head attention mechanism in transformer network is used for deep spatial temporal feature fusion.
3) A DAE-STAFN is proposed for quality prediction of batch process. In order to enhance the adaptability of the model to the batch process, we further proposed to construct batch attention. Use batch attention to model the relationship among samples in small batch. The extracted same sample features and small batch sample features were input into the full connection layer respectively for prediction.

The rest of the paper is organized as follows. Section 2 reviews the basics of the DAE and Transformer. Section 3 specifically describes the proposed denoising autoencoder-spatial–temporal convolutional attention mechanism fusion network model. Experimental validation results are shown in Sect. 4. Section 5 gives the conclusion and outlook.

2 Preliminaries

2.1 Denoising autoencoder (DAE)

As a neural network architecture, autoencoder is used to encode the input data to obtain a low-dimensional representation, followed by decoding this low-dimensional representation to reconstruct the original input. DAE [33], as an extension of AE, further introduces noise into the input data and trains the model to recover the original noise-free data from these noisy inputs. This training process not only allows the encoder to capture the most salient features in the input data, but also significantly improves its generalization ability, making it better at handling general encoders. DAE is unique in its ability to process high-dimensional redundant input data. By deeply learning the intrinsic dependencies and rules of data, DAE can show strong robustness to partially damaged inputs at the intermediate presentation layer, that is, it can extract and reconstruct effective features from damaged inputs. This is in contrast to traditional filtering techniques, which mainly focus on noise removal and have relatively limited contributions to feature extraction. DAE can effectively extract features while removing noise. The schematic diagram of DAE is shown in Fig. 1, DAE consists of input data x, corrupted data $\widetilde{x}$, intermediate features h and reconstructed data z. The corrupted data $\widetilde{x}$ are used as input data, DAE learns mapping from input to representation and reconstruction from representation to input to effectively capture the essential features of data.

The d-dimensional input x is mapped to the hidden representation by a deterministic mapping encoding function as shown in Eq. (1).

$$h = f_\theta(\widetilde{x}) = s(W\widetilde{x} + b) \tag{1}$$

where, s represents the sigmoid function, W is a weight matrix and b is a bias vector.

The intermediate feature representation h is then mapped back into the input space, z is the reconstructed vector of x, and the decoding function is shown in Eq. (2).

$$z = f_\theta\prime(h) = s(W\prime h + b\prime) \tag{2}$$

where, $W\prime$ and $b\prime$ represent the weight matrix and the bias vector, respectively. $W\prime = W^T$.

If the input x is a continuous value vector, the reconstructed loss function is traditional squared error, as shown in Eq. (3).

$$L(x, z) = \|x - z\|^2 \tag{3}$$

2.2 Transformer

Transformer is a deep learning model widely used in the field of natural language processing. It was originally proposed by Vaswani et al. [34] in 2017. Transformer consists

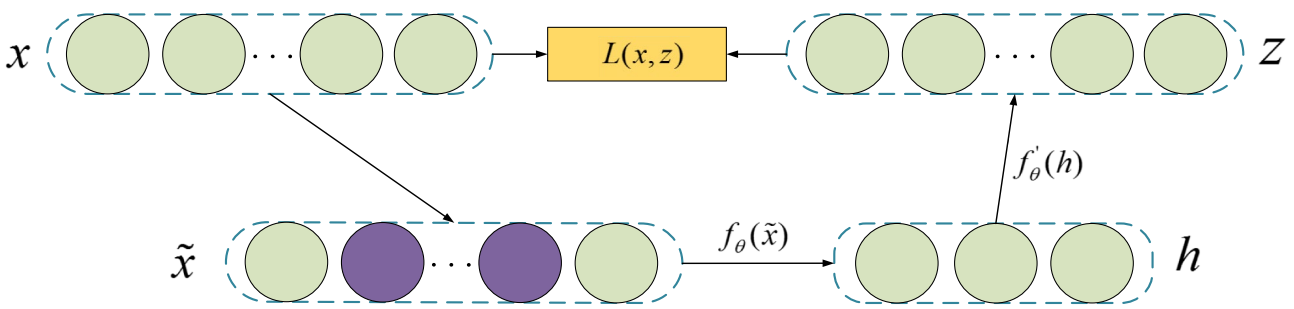


Fig. 1 Schematic diagram of DAE

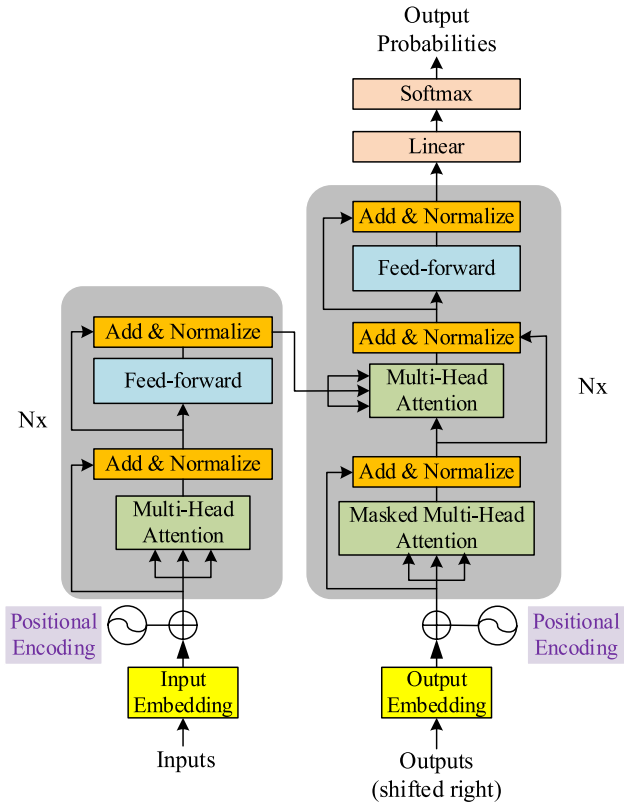


Fig. 2 Transformer Structure

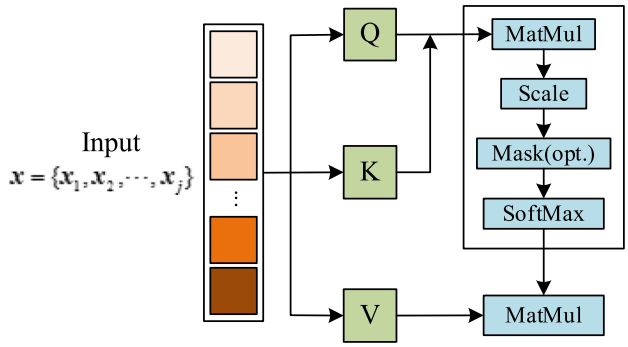


Fig. 3 Self-Attention Calculation Flow

of an encoder and a decoder. The encoder is responsible for transforming the input sequence into a series of hidden states, and the decoder generates the output sequence based on these hidden states. Both the encoder and decoder are stacked with multiple identical layers, each containing a self-attention mechanism and a feed-forward neural network, the structure of which is shown in Fig. 2.

As a key feature of the Transformer, Self-Attention Mechanism is capable of weighting and aggregating various parts of the input sequence $x = \{x_1, x_2, \cdots, x_j\}, x_i \in R^n$. The calculation process is shown in Fig. 3.

Firstly, for each input feature, Query, Key and Value vectors are generated through three different linear layers. These vectors are obtained by multiplying the input elements with a set of learnable parameters. The Query vector is used to compute the similarity to all the Key vectors to determine which input element should be focused on when generating the output.

$$\begin{aligned} q_i &= W^q x_i \in R^{d_n}, \\ k_i &= W^k x_i \in R^{d_n}, \\ v_i &= W^v x_i \in R^{d_n}, i = 1, 2, ..., j \end{aligned} \tag{4}$$

where, W^q, W^k and W^v are the learning matrices of the linear mapping, $Q = [q_1, \cdots, q_j], K = [k_1, \cdots, k_j]$ and $V = [v_1, \cdots, v_j]$.

Then attention weights are calculated. The similarity between each query vector and all key vectors is calculated, often by using methods such as dot product or cosine similarity. These similarity scores are normalized to weights by using the softmax function, which are assigned to the corresponding value vectors when generating the output.

$$\begin{aligned} Similarity(Q, k_i) &= Q^T \cdot k_i \\ Similarity(Q, k_i) &= \frac{Q^T \cdot k_i}{\|Q^T\| \cdot \|k_i\|} \end{aligned} \tag{5}$$

Finally, each value vector is multiplied with its corresponding weight and these weighted vectors are then summed to obtain the output of the self-attention mechanism. This output contains the information about all relevant elements in the input sequence, allowing the model to better understand the relationships between the sequences.

$$Self\_Attention(Q, K, V) = \sum_{i=1}^{n} softmax(Similarity(Q, k_i)) * V^T \tag{6}$$

3 Predictive model construction based on DAE-spatial temporal convolutional attention mechanism fusion network

This section discusses the principles of the proposed DAE-STCAFN model applied in batch process quality prediction and the implementation process. Firstly, the raw data are preprocessed, including data unfolding, noise reduction, variable selection, construction of augmentation matrix and data normalization. Secondly, spatial convolutional attention is calculated for the time series of process and quality variables, and temporal convolutional attention is calculated for the quality variable series. And the spatial–temporal fusion features are obtained by multi-head attention mechanism in Transformer. Finally, the sample relationship within small batch is learned automatically by batch attention. And the quality prediction

model are built before and after the batch attention module to achieve quality prediction.

3.1 Batch process data pre-processing

The batch process data are three-dimensional data $X(I \times J \times L)$ which consist of variable (J), sampling time (L), and batch (I). Therefore, the process data $X(I \times J \times L)$ are expanded along the direction of the variables in the order of increasing batches, resulting in a two-dimensional matrix $X(IL \times J)$, as shown in Fig. 4. As the number of batches increases, the batch process data show time series characteristics.

In the actual industrial data acquisition process, it is often disturbed by noise. Firstly, the unfolded data $X(IL \times J)$ are processed to reduce noise using DAE. In this process, DAE can remove part of noise, but due to the differences of the noise level between variables, it may lead to a complex nonlinear relationship between the variables. On the other hand, MIC can still reflect the real association between variables more accurately when analyzing the relationship between variables, even if there is a certain degree of noise in the data. On the other hand, MIC can simplify the model and improve the prediction accuracy by screening the process variables that have a significant effect on the target variables. Therefore, this paper uses MIC to calculate the correlation between input and target variables. MIC is a feature selection method based on mutual information, which has been widely used in the literature[35, 36] on quality prediction in recent years for relevant feature extraction, and has achieved relatively satisfactory results. The mutual information of X, Y is calculated as shown in Eq. (7).

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{7}$$

where, $p(x, y)$ denotes the joint probability distribution function of X and Y, while $p(x)$ and $p(y)$ denote the marginal probability distribution function of X and Y, respectively.

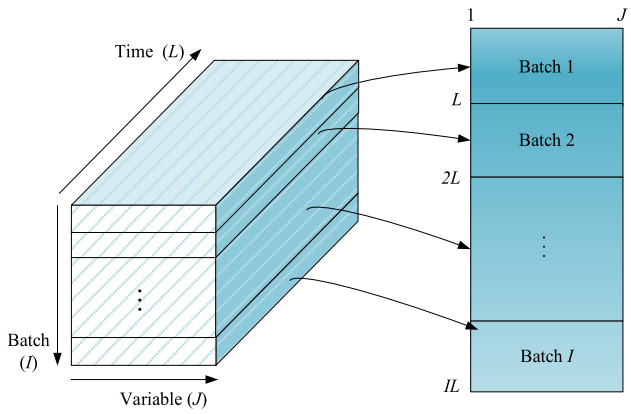


Fig. 4 Schematic of the 3D data expansion

The MIC formula is shown in Eq. (8):

$$MIC(X,Y) = \max_{a*b<B} \frac{I(X,Y)}{\log_2 \min(a,b)} \tag{8}$$

where, a, b are the number of lattice divisions in the x, y direction. B is a variable with a size of about 0.6 times of data.

To fully consider the dynamic characteristics of batch processes, we construct an augmented matrix based on time series for the screened features. By constructing this matrix, more variable information is introduced to eliminate the autocorrelation of time series. In addition, the augmented matrix provides more abundant variables, enabling the model to analyze and predict time series from multiple angles, thereby enhancing the model's time series prediction ability. For the process data $X^i \in R^{L\times J_x}$ of the i-th batch, the augmentation matrix is built with window width m, as shown in Eq. (9).

$$\hat{X}^i = \begin{bmatrix} x^i_{1,1} & x^i_{1,2} & \cdots & x^i_{1,m} & \cdots & x^i_{J_x,1} & \cdots & x^i_{J_x,m} \\ x^i_{1,2} & x^i_{1,3} & \cdots & x^i_{1,m+1} & \cdots & x^i_{J_x,2} & \cdots & x^i_{J_x,m+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^i_{1,L-m+1} & x^i_{1,L-m+2} & \cdots & x^i_{1,L} & \cdots & x^i_{J_x,L-m+1} & \cdots & x^i_{J_x,L} \end{bmatrix} \tag{9}$$

where, $\hat{X}^i \in R^{(L-m+1)\times mJ_x}$.

For better data analysis and training of machine learning models, data normalization is performed on the augmented process variables. Here, the max–min method is utilized to linearly transform the data to the range of [0, 1], as shown in Eq. (10).

$$x_{value}{}' = \frac{x_{value} - x_{min}}{x_{max} - x_{min}} \tag{10}$$

where, x_{value} is the original data value, x_{min} and x_{max} are the minimum and maximum values in the data set respectively.

3.2 Quality prediction model based on spatial–temporal convolution attention fusion network

3.2.1 Spatial convolution attention mechanism

Supposed the input and output sequences are $x = (x^1, x^2, \cdots, x^j) = (x_{(1)}, x_{(2)}, \cdots, x_{(L)}) \in R^{j\times L}$, $y = (y_{(1)}, y_{(2)}, \cdots, y_{(L)}) \in R^L$ respectively, for the complex spatial dependence between multivariate time series, we propose a new spatial convolutional attention, whose computational process is shown in Fig. 5.

For multivariate time series $\hat{X} = (x^1, x^2, \cdots, x^j, y)^T \in R^{(j+1)\times L}$, x_n represents the n-th time series. Considering that local contextual information in the time dimension may be ignored when applying the point-by-point attention mechanism to capture the relationship

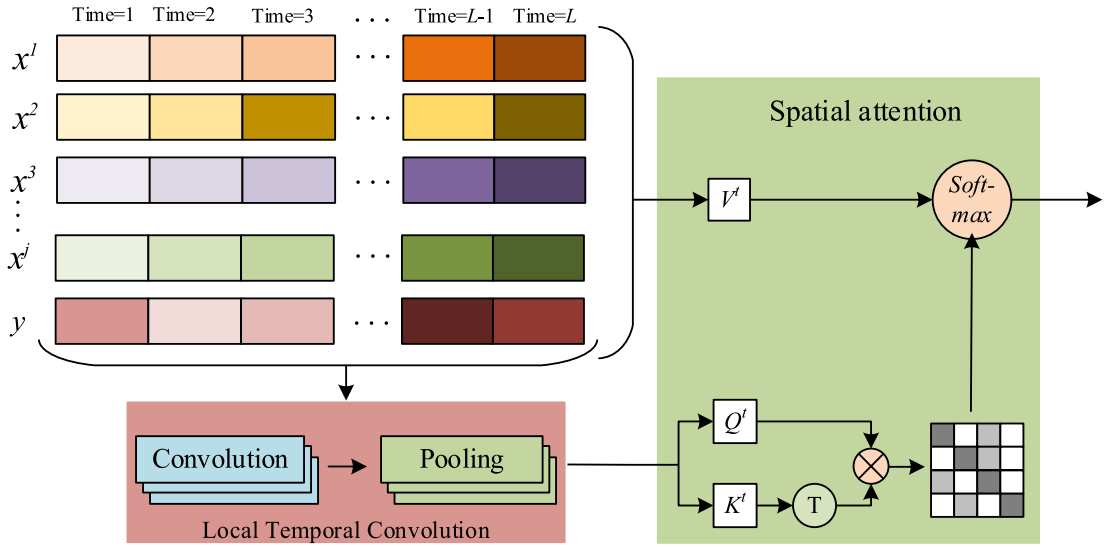


Fig. 5 Spatial Convolutional Attention Computation Process

between multivariate time series, local temporal convolution is introduced for the existence of complex spatial correlations between different time series, which consists of a convolutional layer and a pooling layer.

The convolutional layer consists of a 1-D CNN filter of length ω (ω<L). For each univariate time series, the h-th CNN filter scans along the time dimension and generates the vector $m_h^j = \mathrm{ReLu}(\omega_h * x^n + b_k) \in \mathrm{R}^{L-\omega+1}$, corresponding to the matrix $M^j \in R^{d_{\mathrm{model}} \times (L-\omega+1)}$, where d_{model} represents the number of filters. Then, a 1-D maximum pooling layer on each row of M^j is used to reduce the output size and prevent overfitting. The final feature extracted from each univariate time series by local time convolution is $p^j = \mathrm{MaxPool}(M^j) \in d^{\mathrm{model}}$. Thus, the time series feature matrix $P = (p^1, p^2, \cdots, p^{j+1})^{\mathrm{T}} \in R^{(j+1) \times d_{\mathrm{model}}}$ is obtained. Each row of the feature matrix can be considered as a learned representation of the univariate time series.

Next, each univariate time series feature learned is fed into the spatial self-attention mechanism to capture the spatial dependence between different sequences. The feature matrix P and the input matrix $\widehat{X}$ are converted into p different query matrices Q_i^s, key matrix K_i^s and value matrix V_i^s, as shown in Eq. (11).

$$\begin{aligned} Q_i^s &= PW_i^{qs} \\ K_i^s &= PW_i^{ks} \\ V_i^s &= \widehat{X}W_i^{vs} \end{aligned} \tag{11}$$

where, i is the serial number of the attention head. W_i^{qs}, W_i^{ks} and W_i^{vs} represent learnable parameter matrices.

After that, a series of outputs are computed as shown in Eq. (12) using the scaled dot product self-attention function. Finally, these outputs are concatenated and linearly projected again to produce a representation matrix as shown in Eq. (13).

$$Head_i^s = Attention(Q_i^s, K_i^s, V_i^s) = \mathrm{softmax}(\frac{Q_i^s (K_i^s)^{\mathrm{T}}}{\sqrt{d_k}}) V_i^s \tag{12}$$

$$Z = Concat(Head_1^s, \cdots, Head_h^s) W^{os} \tag{13}$$

where, $\sqrt{d_k}$ denotes the scaling factor.

This new spatial convolutional attention mechanism combines local temporal convolution, pooling layer, and spatial self-attention mechanism, which can better capture the complex spatial dependencies among multivariate time series, and improve the modeling ability and prediction performance for time series data.

3.2.2 Time convolution attention mechanism

Time attention is calculated as shown in Eqs. (14)-(16).

$$\begin{aligned} Q_i^t &= YW_i^{qt} \\ K_i^t &= YW_i^{qt} \\ V_i^t &= YW_i^{qt} \end{aligned} \tag{14}$$

$$Head_i^t = Attention(Q_i^t, K_i^t, V_i^t) = soft\mathrm{max}(\frac{Q_i^t (K_i^t)^T}{\sqrt{d_k}}) V_i^t \tag{15}$$

$$U = Concat(Head_1^t, Head_2^t, \cdots, Head_h^t) W^{ot} \tag{16}$$

For time series data, the effect of local similarity since observations at the current moment can only reflect the

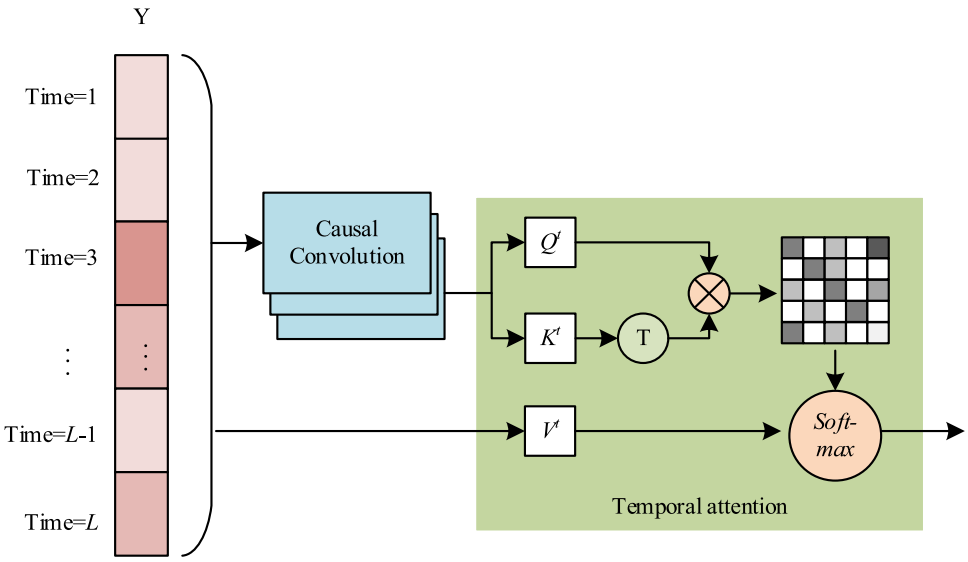


Fig. 6 Temporal Convolutional Attention Computation Process

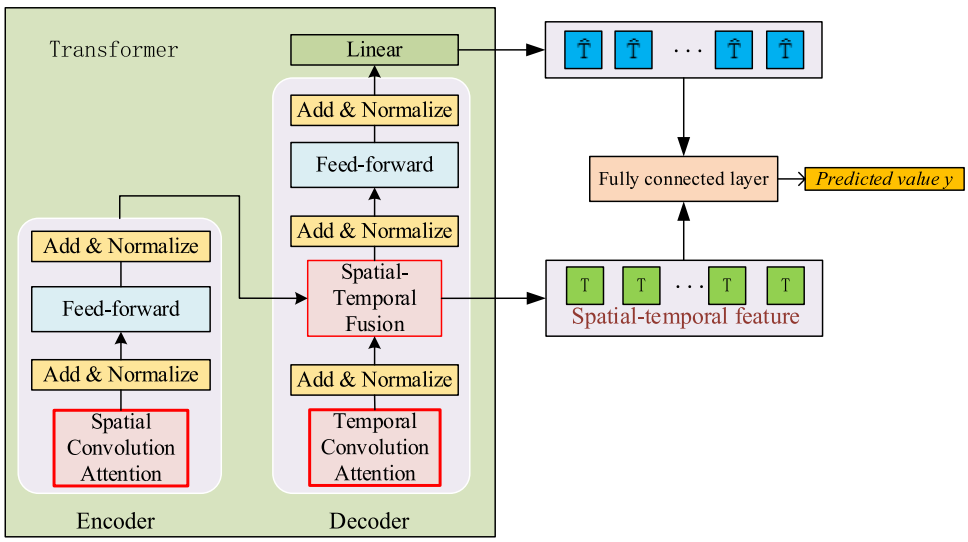


Fig.7 Flowchart for calculating the batch attention mechanism

current state and cannot predict future information. By employing causal convolution, query values and key values for information retrieval are generated. The unique feature of temporal convolutional attention is the ability to correlate different positions of the target sequence in the time dimension, which enables the model to learn long-term temporal dependencies and thus exhibiting enhanced performance in handling complex time series analysis tasks. Figure 6 illustrates the computational process of temporal convolutional attention.

3.2.3 Spatial–temporal convolution attention fusion mechanism

To better capture the spatial–temporal relationships between the variables, in the encoder part of the Transformer, we use the computed spatial convolutional attention as the input, which helps the model better capture the spatial correlation in the input data. Meanwhile, we use the temporal convolutional attention as the input to the decoder part. As shown in Fig. 7, the outputs of the encoder and decoder are more efficiently fused by the multi-head attention mechanism in Transformer to generate a representation with rich spatial–temporal map ξ^{st}, as shown in Eq. (17).

$$\xi^{st} = soft\max\left(\frac{(ZW^{qst})(UW^{kst})^{T}}{\sqrt{d_k}}\right) \tag{17}$$

where, $W^{qst}, W^{kst} \in R^{d_{model} \times d_k}$ denote the learned parameters. d_{model} is the dimension of the features.

The temporal-spatial attention fusion features used for prediction can be expressed as $T^{st} = \xi^{st}(ZW^{vst})$. Input the

spatial–temporal fusion features T^{st} into the fully connected regression layer to predict the quality output:

$$\widetilde{y} = \varepsilon(W^T T^{st} + u) \tag{18}$$

To further enhance the learning capability of the model, we propose a batch attention mechanism module for the generated spatial–temporal fusion features $T_1^{st}, T_2^{st}, \cdots, T_M^{st}$, which helps the model understand the underlying structure of the data more thoroughly by learning the feature $\widehat{T}$ between samples within a batch. Batch attention is mainly composed of multi-head attention (Multi-Head), layer normalization (Layer-Norm) and feed-forward network (FFN). The batch attention calculation process is shown in Eq. (18).

$$\begin{aligned} T\prime &= LayerNorm(MultiHead(T) + T) \\ \widehat{T} &= LayerNorm(FFN(T\prime) + T\prime) \end{aligned} \tag{19}$$

Input the spatial–temporal fusion features $\widehat{T}$ into the fully connected regression layer to predict the quality output:

$$\widehat{y} = \varepsilon(W^T \widehat{T} + u) \tag{20}$$

where, W^T and u are prediction weights and bias vectors, which belong to prediction sharing parameters. ε represents regression coefficient.

Finally, the network parameter set $\{\theta^1_{STCAFN};\theta^2_{STCAFN};\ldots;\theta^n_{STCAFN};W^T,u\}$ is obtained by minimizing the unconstrained optimization problem shown in formula (21).

$$L(\theta_{STCAFN}) = \frac{1}{2n}\sum_{i=1}^{n} \|\widetilde{y}(i) - y(i)\|^2 + \frac{1}{2n}\sum_{i=1}^{n} \|\widehat{y}(i) - y(i)\|^2 \tag{21}$$

where, $y(i)$ is the truth value.

The overall time complexity of DAE-STCAFN is the sum of the time complexity of the noise reduction DAE and the spatial–temporal convolutional fusion network, including the forward propagation time of the input data in the DAE through the autoencoder and the convolution time and fusion operation time in the spatial–temporal convolutional network.

As the size of batch data increases, the DAE-STCAFN can improve processing efficiency through distributed computing and parallel processing. In addition, the structure of the network can be adjusted according to the size of the data. For large-scale data, the dimensions of hidden layers or the number of convolutional kernels can be increased to improve the expressive power of the model, but this would also increase the computational time. Therefore, a trade-off needs to be made between model performance and computational efficiency. The number of parameters of the network mainly depends on the hidden layer dimensions of the autoencoder, the size and number of spatial–temporal convolutional kernels, and the complexity of the fusion operation. These parameters can be adjusted to suit different application needs. So the DAE-STCAFN can improve the accuracy of quality control by denoising and feature extraction of data and can meet the real-time requirements.

The batch process quality prediction model based on DAE-STCAFN is divided into two parts: offline modeling and online prediction, and the flow chart is shown in Fig. 8.

Offline modeling:

Step 1: Collect historical data of batch processes, pre-process the data: noise reduction, variable selection, constructing augmentation matrices and normalization.
Step 2: Compute local temporal attention for univariate time series of input variables and spatial convolutional attention between multivariate time series.
Step 3: Compute temporal convolutional attention for the quality-variable time series.
Step 4: In the Transformer model, spatial convolutional attention is used as the input of the encoder.
Step 5: The temporal convolutional attention is used as the input of the decoder, and the spatial-temporal fusion features are obtained through the spatial-temporal fusion mechanism.
Step 6: Establish a batch attention mechanism to automatically learn the correlation characteristics between variables.
Step 7: Separate regression predictions were made for the features before and after the batch attention calculation, respectively.

Online prediction:

Step 1: Collect batch process online data and pre-process the data.
Step 2: Compute spatial convolutional attention.
Step 3: Compute temporal convolutional attention.
Step 4: The computed spatial convolutional attention and temporal convolutional attention are put into the trained quality prediction model for quality prediction.
Step 5: The predicted values are inversely normalized to obtain the final quality prediction.

Furthermore, the three criteria are used to evaluate the performance of the proposed method. They are Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and determinate coefficient (R^2), which are shown in Eq. (23). RMSE can well reflect the deviation between the predicted value and the actual value. The value of MAE visually represents the mean absolute deviation between the predicted value and the true value, making it easier to understand and interpret. R^2 is mainly used to compare the ability of different models to fit the same dataset. When there are multiple

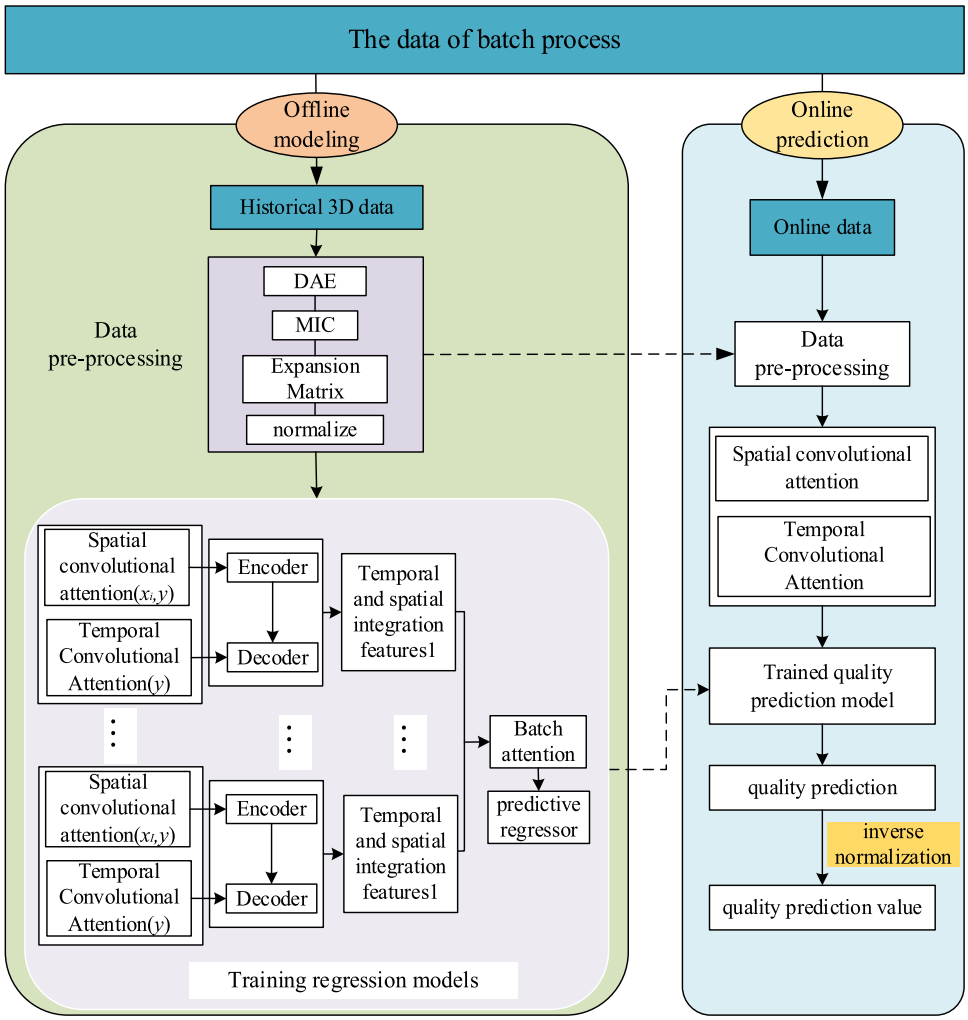


Fig.8 Flowchart of quality prediction based on DAE-STCAFN

models to choose, it is possible to determine which model performs better in fitting the data by comparing their R^2 values.

$$\begin{aligned} MAE &= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \\ RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \\ R^2 &= 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{Y}_i)^2} \end{aligned} \tag{22}$$

where, is the true value, $\hat{y}_i$ represents the model predicted value, and N represents the number of samples in the test set.$\overline{Y}_i$ is the mean of Y_i.

4 Experimental verification

To validate the performance of DAE-STCAFN, it is applied for the penicillin fermentation simulation process and hot strip mill process. These computations are done in Python 3.7. The hardware is an Intel (Intel) Core (TM) i7-8700 GPU @ 3.20 GHz 16.00G RAM.

We employ a variety of regularization techniques in our model to avoid overfitting. First, $L2$ regularization in the layers of neural networks is used to limit the complexity of the model and prevent overfitting by penalizing the weights. Specifically, we add a weight decay parameter to the optimizer to regularize the weights of the model. In addition, we also employ Dropout technology to randomly discard some nerve cells during training, increasing the generalization ability of the model. Dropout can effectively reduce co-adaptation between nerve cells and prevent overfitting, which is used in multiple layers of the model, and we experimentally adjust the probability of Dropout to obtain the best performance.

To ensure that the model does not overfit, we employ a rigorous validation procedure. We divide the dataset into a training dataset, a validation set, and a test set. During training, we use a training dataset to train the model, a validation set to monitor the model's performance, and adjust

the model's hyperparameters based on the validation set's performance. When the model's performance on the validation set no longer improves, we stop training to prevent overfitting. We also use an early stop technique, where we stop training when the model's performance on the validation set does not improve for several consecutive epochs to prevent overfitting. In order to verify the generalization ability of the model, we divide the test set into several independent datasets for verification, and take the average of each performance index as the verification result of the final test set.

4.1 Penicillin fermentation simulation platform

In this section, the proposed model is validated by performing experiments on a penicillin fermentation simulation platform. The production of penicillin is a typical batch process, which requires suitable conditions such as medium environment, pH value, aeration rate, stirring power, etc., to ensure that the bacteria can carry out antibiotic growth and anabolism normally. The whole process is characterized by nonlinearity, time-varying and uncertainty. Considering the complexity of the intrinsic reaction mechanism, the key parameters cannot be directly measured online, so predictive modelling of the key parameters is very important. In 2002, Cinar led a group at the Illinois Institute of Technology to develop the Pensim2.0 simulation software[37], which could simulate the batch process more realistically and comprehensively, so it has been widely used in research fields such as batch process monitoring and quality control[38, 39]. The flow chart of penicillin fermentation process is shown in Fig. 9.

In this paper, the time period of the penicillin reaction process is set to 400 h and the sampling time is set to 0.5 h, and then 10 batches of data under normal working conditions with different initial conditions and within the allowable range are obtained. Among 18 process variables, 10 process variables are selected. This specific value ranges of the variables are presented in Table 1. Additionally, the quality variables are penicillin concentration and biomass concentration.

Random noise with a mean value of 0 and variances of 0.01, 0.1, and 1 is added to all measured data to simulate the actual data in production. The mean square error between the original data and the reconstructed data under three different noises is shown in Fig. 10. As can be seen from Fig. 10, after adding random noise with a mean value of 0 and a variance of 1, the mean square error between the original data and the reconstructed data is the smallest. Therefore, in this section, penicillin concentration and biomass concentration are predicted in a noise environment with (0,1).

As shown in Fig. 11, the data distribution histograms of the original and reconstructed data are presented under the (0, 1) noise environment. The horizontal axis represents the value range of the data after the data noise reduction process by DAE. For the penicillin fermentation simulation data, it encompasses the 10 features listed in Table 1. The horizontal axis covers the interval from the minimum to the maximum value in the data and is divided into several intervals, each corresponding to a specific data range. The vertical axis

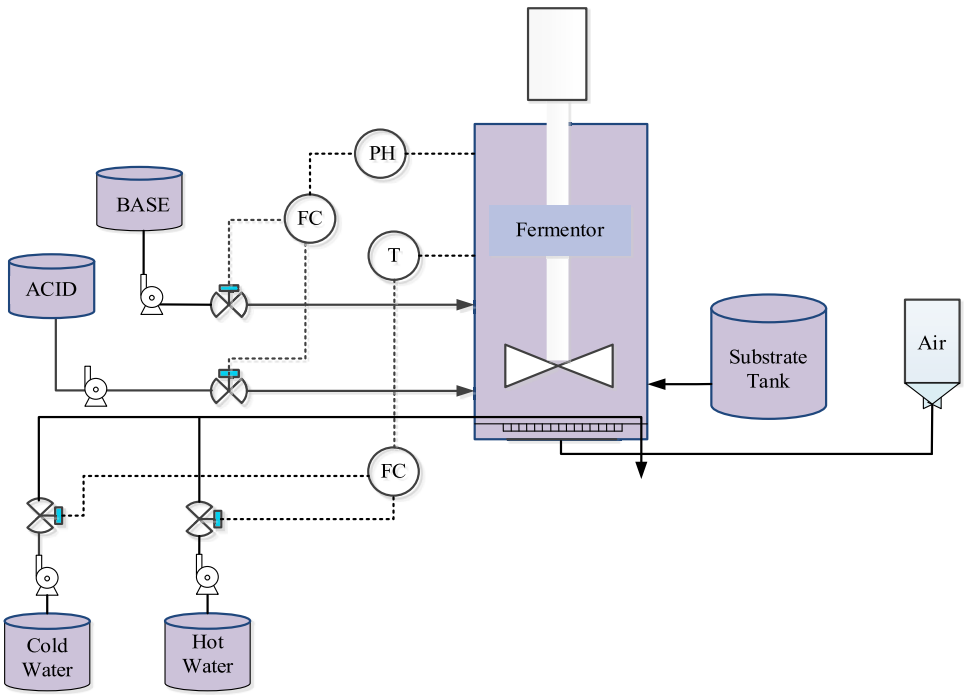


Fig.9 Flow chart of penicillin fermentation process

Table 1 Ten main variables and initial value settings for penicillin fermentation process

Variable	Description	Default setting	Setting range
×1	Aeration rate (L/h)	8.6	8 ~ 9
×2	Agitator power (r/min)	30	29 ~ 31
×3	Substrate feed flow rate (L/h)	0.042	0.039 ~ 0.045
×4	Substrate feed temperature (K)	296	295 ~ 296
×5	Substrate concentration (g/L)	15	14 ~ 18
×6	O_2 (%)	1.16	1.0 ~ 1.2
×7	Culture volume (L)	100	100 ~ 104
×8	CO_2 (%)	0.5	0.5 ~ 1.0
×9	PH	5.0	4.5 ~ 5.5
×10	Temperature (K)	298	295 ~ 301

represents the number or frequency of data points within each interval, reflecting the number of data points in a particular data range. A higher value on the vertical axis indicates more data points in the corresponding interval, while a lower value indicates fewer data points. By observing the histograms of the original and reconstructed data, we can visually compare their data distributions within different value ranges and then assess that the data distributions are less altered after DAE processing, thus better preserving the original characteristics and meaning of the data.

The results of quality-related feature selection using MIC are shown in Fig. 12, where the blue bars represent the variables associated with the penicillin concentration variables. It can be seen that the variables $\times 3, \times 4, \times 5, \times 6, \times 7, \times 8$ have a greater effect on penicillin concentration. The orange bar graph represents the variables associated with the biomass

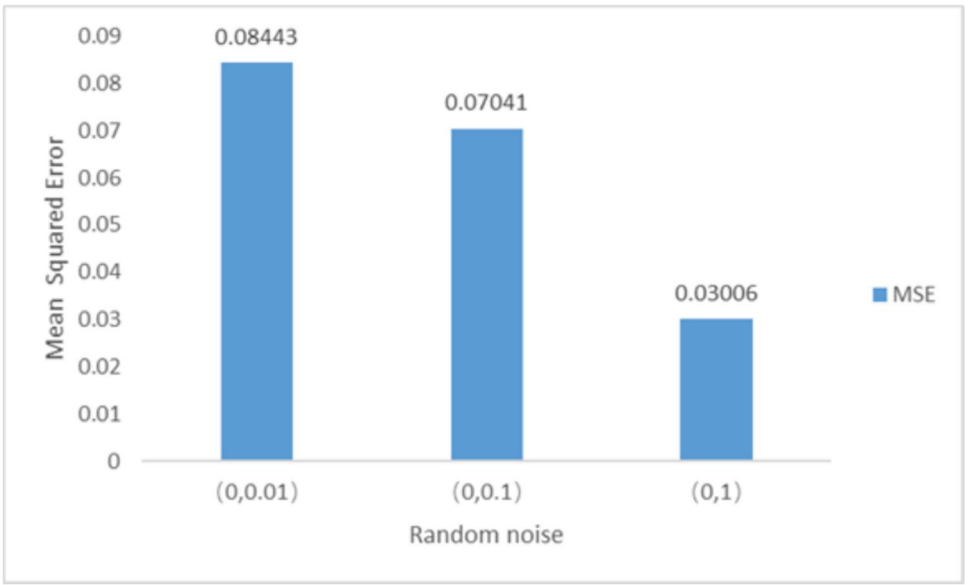


Fig. 10 Mean square error (MSE) of original data source and reconstructed data in different noise environments

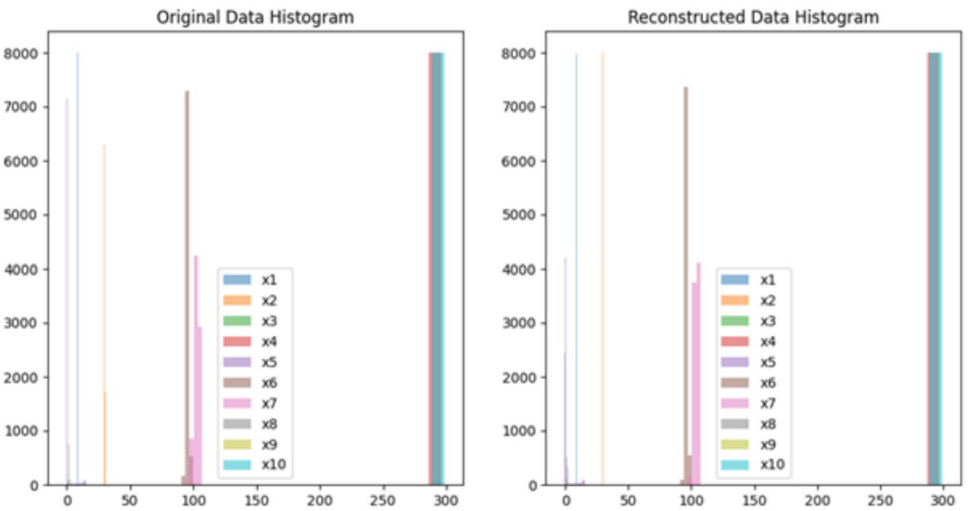


Fig. 11 Distribution histogram of original data source and reconstructed data

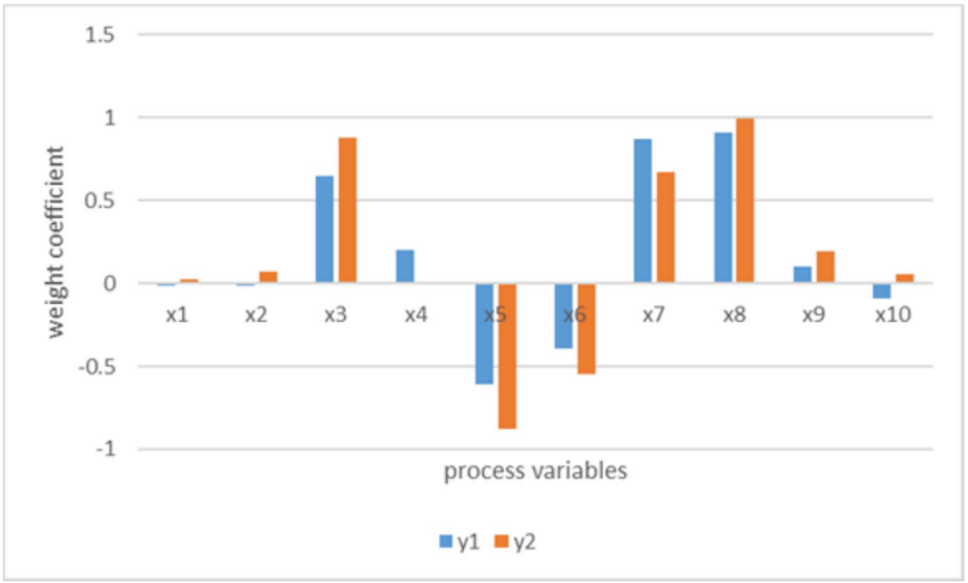


Fig. 12 Variable selection results

Table 2 Comparison of Ablation Experiment Results of Concentration Model in Penicillin Production Process

Model	RMSE	MAE	R^2
Non-DAE-STAFN	0.0363	0.0102	0.9931
Non-DAE-STCAFN	0.0346	0.0050	0.9938
DAE-STAFN	0.0345	0.0215	0.9970
DAE-STCAFN	0.0099	0.0077	0.9990

concentration variables, it can be seen that the variables $\times 3, \times 5, \times 6, \times 7, \times 8, \times 9$ have a large effect on the concentration of bacteriophage. By selecting the key features, it helps to provide a basis for decision making in the model on one hand, and on the other hand, it allows the practitioner to visualize which factors are most important in the input data.

In order to analyze the rationality of the model structure and the help of each component to improve the prediction accuracy of the model, an ablation experiment was carried out on the DAE-STCAFN model. The Non-DAE-STAFN, DAE-STAFN, Non-DAE-STCAFN and DAE-STCAFN models were experimentally compared. As can be seen from Table 2, the RMSE and MAE values of the DAE-STCAFN model are the smallest, with smaller errors. Further analysis of the results shows that the model Non-DAE structure has insufficient data feature extraction ability, resulting in poor model prediction performance. Models without a convolutional structure cannot extract enough important time information from the time step, which affects the prediction effect of the model.

Predicted results of penicillin concentration

Firstly, the penicillin concentration is taken as the prediction target, and the prediction is carried out with the existing models such as LSTM, S-LSTM [40], GSTAE, STA-LSTM, DAE-STAFN and the DAE-STCAFN model proposed in this paper. To ensure the objectivity of the validation results, the data are divided into training set, validation set and test set according to 6:2:2. As shown in Table 3, the proposed model has the best performance in all the metrics compared to the other five models. The results of the six models for penicillin concentration prediction are shown in Table 3. It can be seen from the table that compared with the other five models, the proposed model has the smallest RMSE and MAE values and the largest R^2 value, which shows that the proposed model has the best prediction performance.

In order to show more visually the prediction effect of each model on penicillin concentration, Fig. 10 shows the prediction fit curves after inverse normalization for each model, separately, the red color curve represents the true value and the blue color curve represents the predicted value. As shown in Fig. 13 (a), this is the prediction effect of LSTM, it can be seen that there is large error between

Table 3 Performance metrics of six models for predicting penicillin concentrations

Model						
Metrics	S-LSTM	LSTM	GSTAE	DAE-STAFN	STA-LSTM	DAE-STCAFN
RMSE	0.1693	0.0568	0.0258	0.0345	0.0163	0.0099
MAE	0.1541	0.0460	0.0205	0.0201	0.0117	0.0077
R^2	0.7508	0.9719	0.9960	0.9970	0.9988	0.9990

the predicted value of LSTM and the actual value, this is because, in the calculation of LSTM, the information at each time step needs to rely on the state of the previous time step and the current input. As the sequence increases, the information transmission path becomes longer and the risk of information loss also increases. Figure 13(b) is the prediction fitting curve of S-LSTM model. It realizes the simultaneous use of quality variables and input variables to learn more relevant dynamic hidden states. However, S-LSTM has the problem of large computational resource consumption. Figure 13(c) shows the prediction effect of GSTAE on penicillin concentration, as a target correlation auto-encoder with a gating structure, GSTAE is capable of controlling the validity of the historical information. Although GSTAE has its unique features in processing spatiotemporal data, it has shortcomings in dealing with complex dynamic changes and noise interference. Figure 13(d) shows the predictive fitting effect of SAT-LSTM model. SAT-LSTM is a long-term

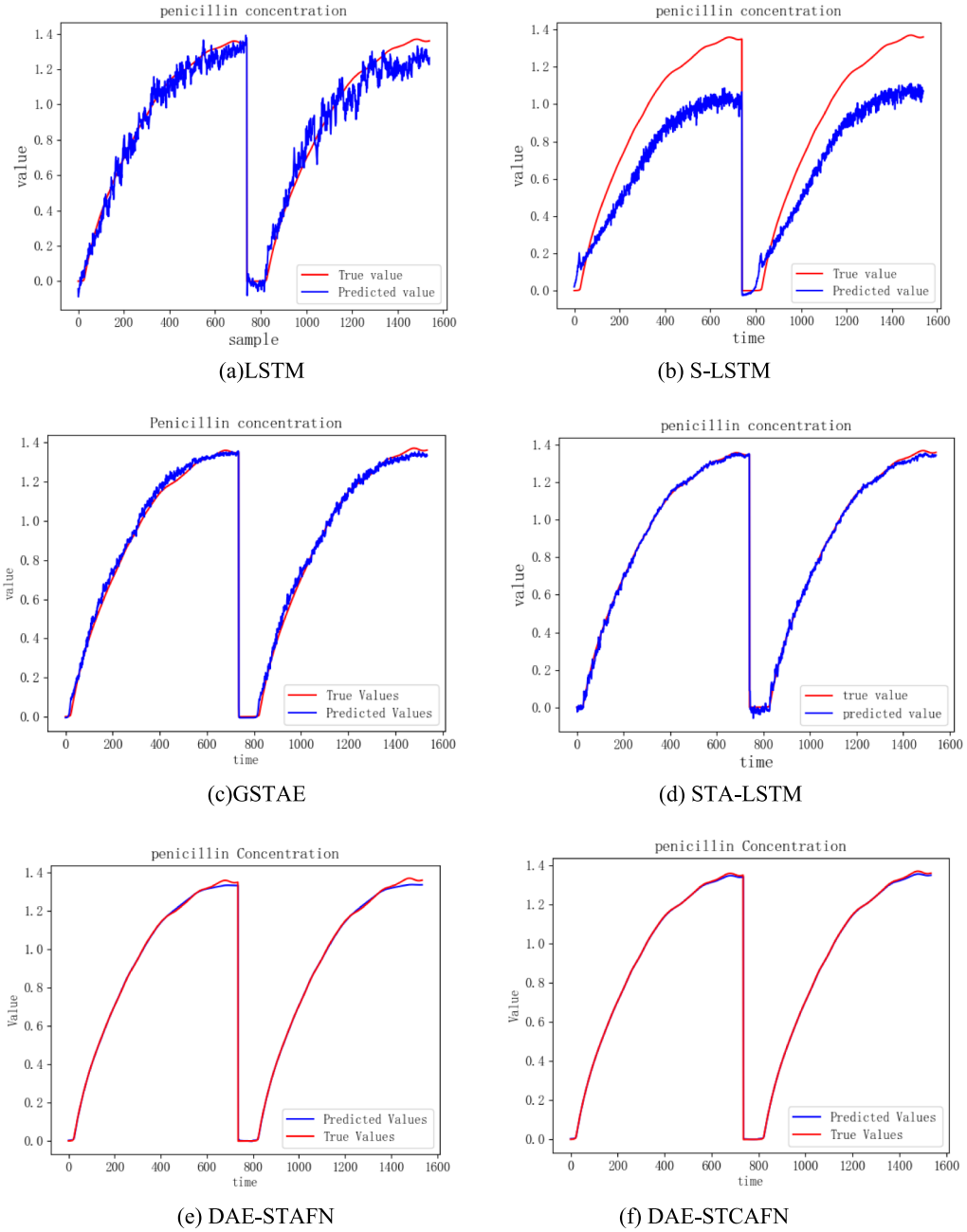


Fig.13 Prediction results of penicillin concentration by (**a**) S-LSTM, (**b**) LSTM, (**c**) GSTAE, (**d**) STA-LSTM, (**e**) DAE-STAFN, (**f**) DAE-STCAFN

and short-term prediction network based on spatiotemporal attention computation. However, STA-LSTM only uses spatial–temporal attention to filter out useful spatial–temporal location information, and does not suppress redundant information through convolution, so its performance is affected. Figure 13(e) shows the prediction effect of DAE-STAFN on penicillin concentration. It can be seen from the figure that its prediction accuracy R^2 is 0.9970 without the convolution operation. It can be seen from Fig. 13(f) that the prediction effect of DAE-STCAFN is better than the other five prediction models. The DAE-STCAFN model reduces the noise of the data on the one hand and ensures that the data are not disturbed by noise during model training. On the other hand, the dynamic spatial–temporal relationship between variables is deeply explored through the spatial–temporal convolutional attention mechanism, and the learned spatial–temporal relationship is fused by transformer to extract more accurate spatial–temporal relationship features, so as to improve the prediction performance of the proposed model.

Figure 14 shows the scatter plots of penicillin concentration predicted by six models, in which the blue asterisk the true value and the predicted value, and the red solid line represents the reference line. We find that the S-LSTM model has a large deviation throughout the quality prediction process. The LSTM and GSTAE models has a large deviation at the beginning of the batch. At the beginning of the reaction, the real value of STA-LSTM model is slightly different from the predicted value. In the DAE-STAFN model, there is also a slight deviation between the predicted value and the real value at the beginning of the reaction. It can be seen that the DAE-STCAFN has the best prediction results in the quality prediction process except for a few samples with bias. This further demonstrates the effectiveness and good generalization ability of the DAE-STCAFN model in the quality prediction of batch process.

Predicted results of biomass concentration

In the prediction of biomass concentration, the average evaluation indexes of each model after 10 repetitions of prediction are shown in Table 4, it can be seen that in the prediction of biomass concentration, RMSE of the proposed model is 0.0052 and MAE is 0.0036, compared to the other five models, the proposed model has the best prediction performance.

Figure 15 demonstrates the inverse normalization results of the six models for the prediction of biomass concentration, it can be seen that the other five models have larger error between the true value and the predicted value throughout the prediction process, while the proposed model has a better prediction performance. It means that the time step feature extraction of individual variables can help the model better solve the spatial–temporal relationship between variables, and improve the generalization and effectiveness of the proposed model. In addition, the DAE-STCAFN model has a p-value of 8.624576457542713e-10 and a confidence interval of (−0.058337165430181186, −0.05154930202663644). A

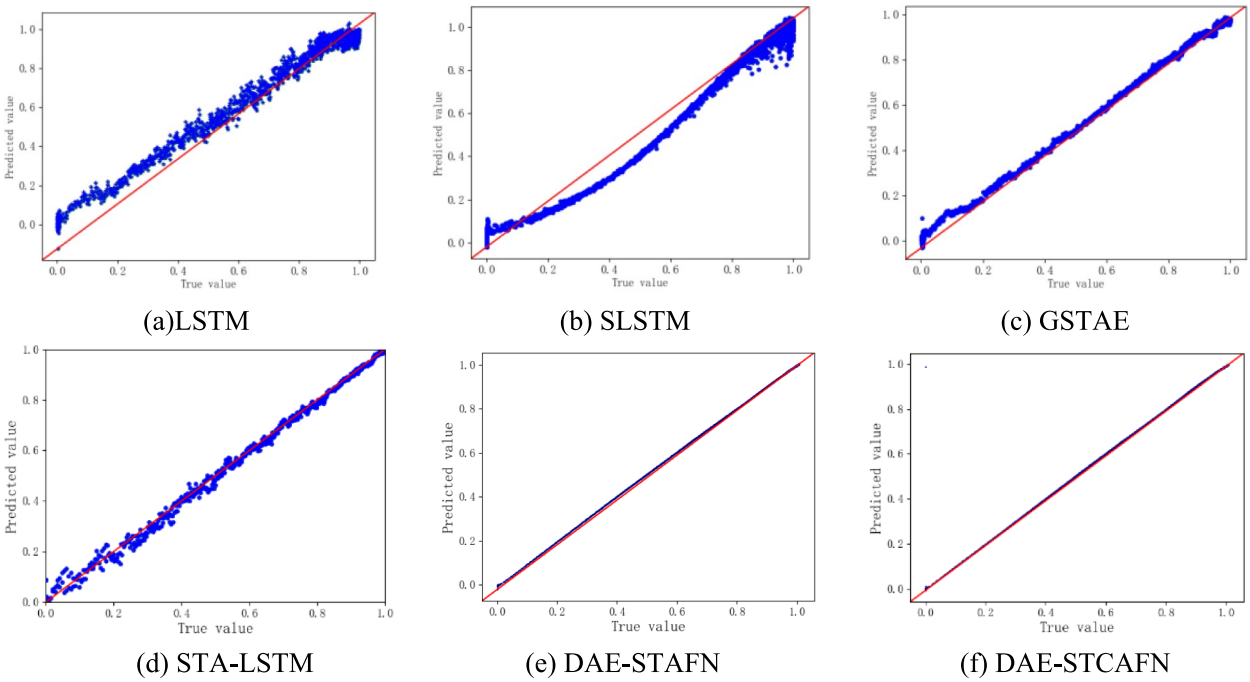


Fig. 14 Scatter plots of predicted penicillin concentration by (**a**) LSTM, (**b**) S-LSTM, (**c**) GSTAE, (**d**) STA-LSTM, (**e**) DAE-STAFN, (**f**) DAE-STCAFN

Table 4 Performance metrics of the six models for predicting biomass concentration

Model						
Metrics	S-LSTM	LSTM	GSTAE	DAE-STAFN	STA-LSTM	DAE-STCAFN
RMSE	0.2623	0.0458	0.0235	0.0218	0.0152	0.0052
MAE	0.2470	0.0383	0.0179	0.0165	0.0108	0.0036
R^2	0.0727	0.9717	0.9924	0.9933	0.9968	0.9990

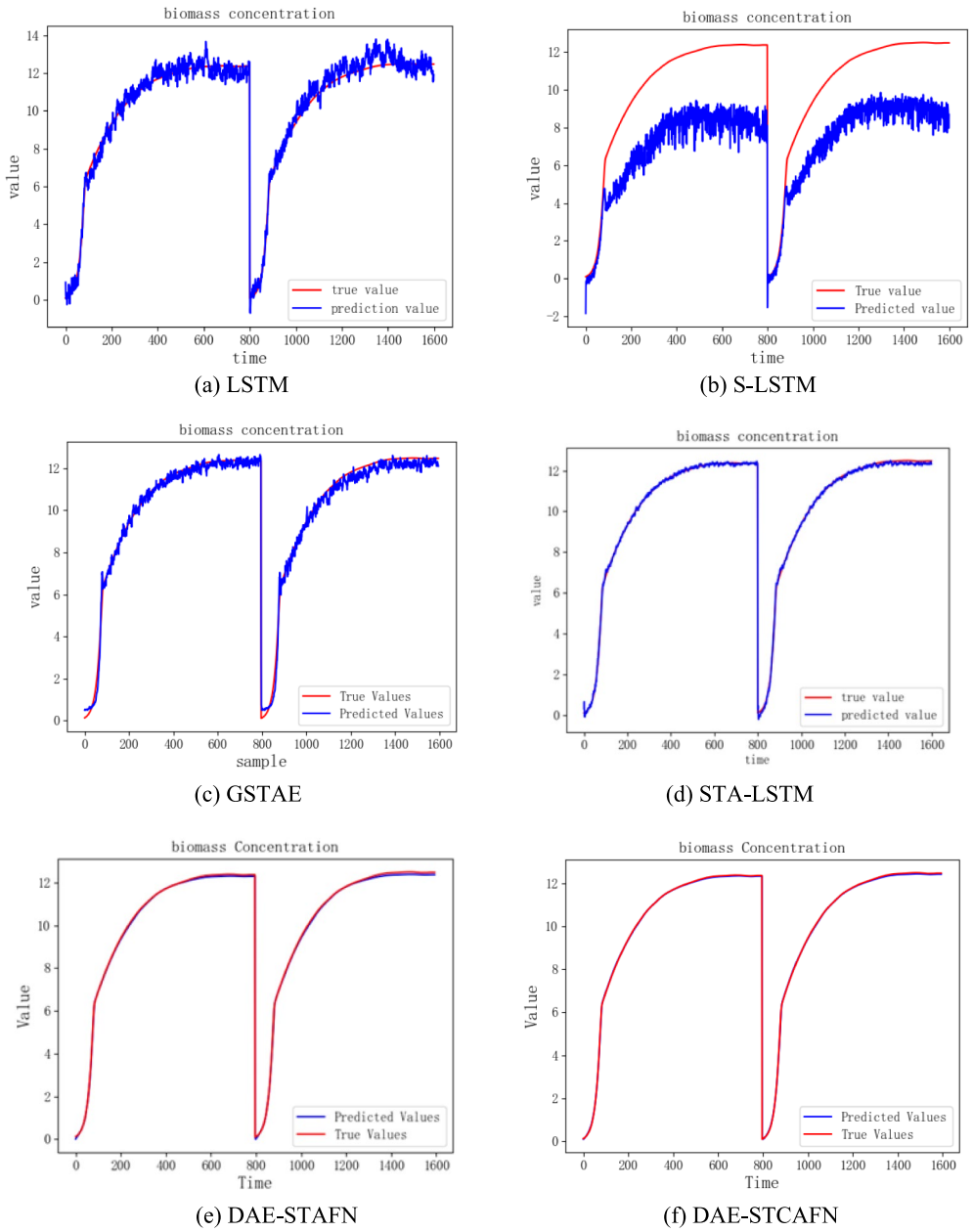


Fig.15 Prediction results of biomass concentration by (**a**) S-LSTM, (**b**) LSTM, (**c**) GSTAE, (**d**) STA-LSTM, (**e**) DAE-STAFN, (**f**) DAE-STCAFN

low p-value means that the model's predictions are statistically significant, while a narrow confidence interval reflects a higher accuracy of the model's predictions.

The scatter plots of biomass concentrations predicted by the six models are shown in Fig. 16. It can be seen that the S-LSTM, LSTM and GSTAE models have large deviation throughout the quality prediction process, The STA-LSTM model has some deviation in the beginning part of the batch. The DAE-STAFN model has bias in the whole process. The DAE-STCAFN model has the best prediction results in the quality prediction process except for a few samples with bias. This further demonstrates the effectiveness and good generalization ability of the DAE-STCAFN model in the quality prediction of batch processes.

4.2 Case study on hot strip mill process (HSMP)

The experimental data for this section are obtained from a 1700 mm hot strip mill process at a steel company [41]. The field data collected from this line are for the production of hot strip with a thickness of 2.70 mm. Figure 17 presents the configuration of the hot strip mill. As can be seen in Fig. 17, the industrial HSMP basically consists of 6 main sections in sequence: heating furnace, roughing mill, conveyor table and shear, finishing mill, cooling table, and coiler. During the roughing stage, the thickness of the hot rolled steel slab is roughly reduced, while its length increases proportionally to the reduction in thickness. After being conveyed through the transfer table, the head and tail of the strip are sheared to avoid damage to the work rolls. Next comes the finishing cut, the centerpiece of the process, which further reduces the thickness so that the exact desired thickness is achieved.

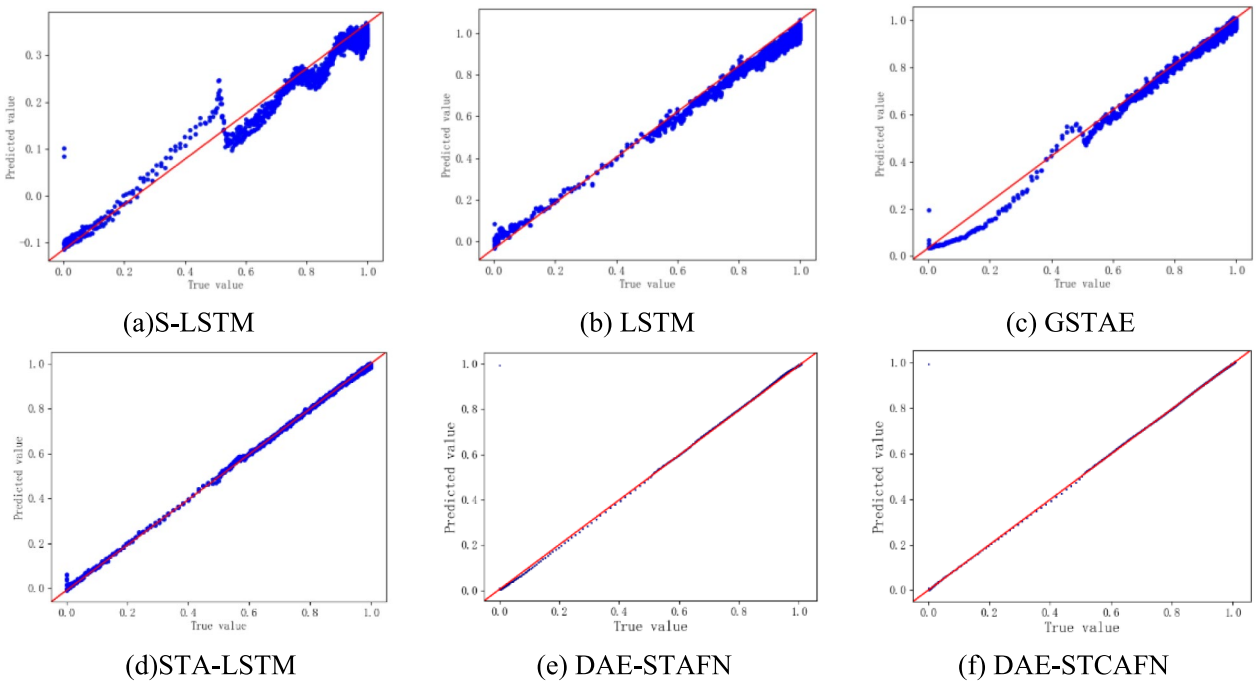


Fig.16 Scatter plots of predicted biomass concentration by (**a**) S-LSTM, (**b**) LSTM, (**c**) GSTAE, (**d**) STA-LSTM, (**e**) DAE-STAFN, (**f**) DAE-STCAFN

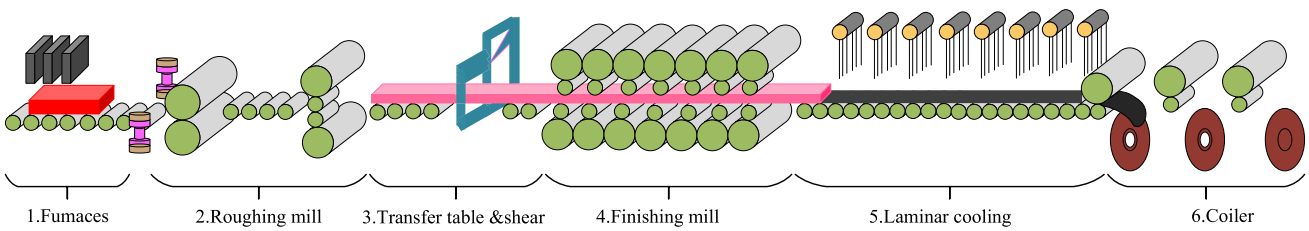


Fig.17 Schematic layout of the hot strip mill

Finally, the hot rolled strip is rolled into the desired product through laminar flow cooling equipment. There are 20 process variables, including roll gap, rolling force and roll bending force for the seven stands of the finishing mill (no roll bending force for the first stand). The quality variable is the finishing mill exit thickness. The process and quality variables are described in Table 5. In addition, the sampling time is 30 s and the sampling period is 10 ms, so the total number of samples is 3000[42].

LSTM, S-LSTM, GSTAE, DAE-STAFN, STA-LSTM and DAE-STCAFN proposed in this paper were used to predict strip thickness. Table 6 shows the index values of the prediction results. It can be seen that compared with other models, the DAE-STCAFN model has the smallest RMSE and MAE values. It shows that the DAE-STCAFN model has the smallest prediction error and better prediction performance.

Figure 18(a) shows the fitting curve of the prediction of the strip thickness by the LSTM model, it can be seen that there is a large error between the true value and the predicted value during the whole prediction process. This is because the information lost by LSTM gradually increases as the sequence increases. It can be seen from the prediction effect of S-LSTM on the strip thickness shown in Fig. 18(b) that although S-LSTM extracts the quality-related features of the process variables before forecasting, the prediction accuracy of the later prediction is not high, compared with the LSTM model, the prediction accuracy is slightly improved. The GSTAE model shown in Fig. 18(c) not only extracts the quality-related features, but also makes full use of the features of each hidden layer by gating nerve cells, so the prediction effect is better than that of the S-LSTM model. Figure 18(d) shows the prediction effect of STA-LSTM on strip thickness, and the prediction effect is generally better than that of the previous models. Figure 18(e) shows the fitting effect of the DAE-STAFN model to predict the thickness of the strip. It can be seen that there is no local information extraction of spatial–temporal features, and its prediction effect is not ideal. Figure 18(f) shows the prediction fitting curve of the model proposed in this paper, it can be seen that the prediction accuracy R^2 of the DAE-STCAFN model reaches 94.72%. Therefore, the proposed model has better prediction accuracy. In addition, the proposed model has a p-value of 1.14032569423156e-06 and a confidence interval of (0.02006123456111642, 0.02331562348956123). The p-value is much less than the common significance level of 0.05, this indicates that the difference between the model's prediction value and the true value is highly statistically significant. The lower bound of the confidence interval is 0.0200 and the upper bound is 0.0233. Therefore, a low P-value means that the model's predictions are statistically significant, while a narrow confidence interval reflects a higher accuracy of the model's predictions.

Table 5 Description of process and quality variables in finishing mill

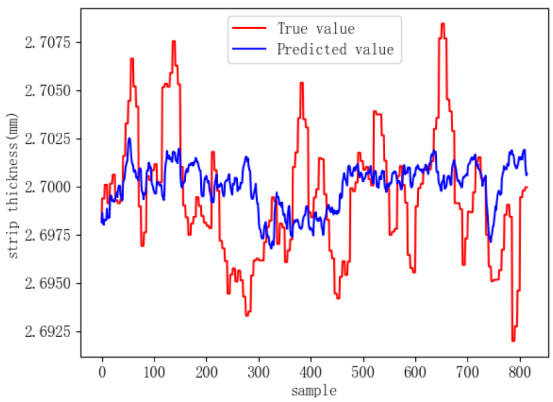
Variable	Type	Description	Unit
$v1 \sim v7$	Measured	Average gap of *Fi* stand, $i = 1, \ldots, 7$	mm
$v1 \sim v14$	Measured	Total force of *Fi* stand, $i = 1, \ldots, 7$	MN
$v15 \sim v20$	Measured	Work roll bending force of *Fi* stand, $i = 1, \ldots, 7$	MN
y	Quality	Finishing mill exit thickness of strip	mm

5 Conclusion

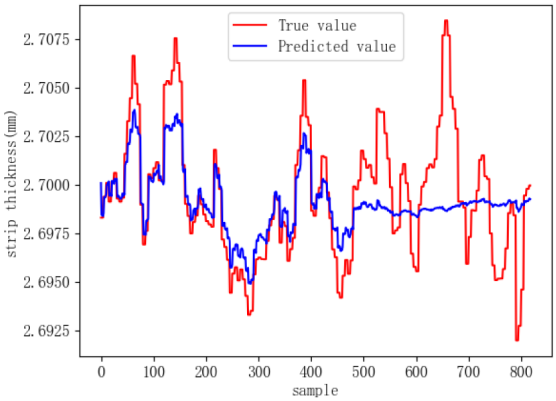
This paper proposes a batch process quality prediction method based on the DAE-STCAFN. Data quality is first considered and noise reduction is performed using DAE, then variable selection is used to eliminate the variables that are not related to the quality variables. Autocorrelation of the variables in the time series is eliminated by constructing an augmentation matrix. In order to effectively learn the spatial–temporal relationship between the variables, the spatial–temporal convolutional attention mechanism is proposed. Temporal convolutional attention is computed to enhance the attention to dynamic temporal dependencies. Spatial convolutional attention is computed to attend to complex spatial correlations. And the computed temporal attention and spatial attention are input into the Transformer model separately to obtain more accurate spatial–temporal fusion features. The batch attention module is proposed to learn the features for each small batch of samples and build the regressor before and after the batch attention module for quality prediction. Finally, it was verified in the simulation process of penicillin fermentation and hot strip mill rolling, and the prediction performance was compared with S-LSTM, LSTM, GSTAE, STA-LSTM, DAE-STAFN. We found that the R^2 of the proposed model was 0.9990 in the prediction of penicillin concentration. In the prediction of

Table 6 Performance metrics of the six models for predicting strip thickness

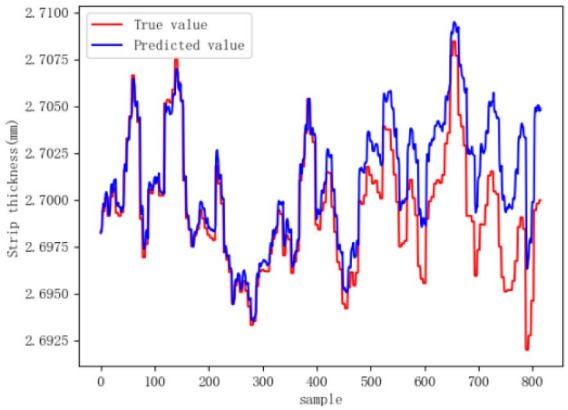
Model						
Metrics	LSTM	S-LSTM	GSTAE	STA-LSTM	DAE-STAFN	DAE-STCAFN
RMSE	0.1485	0.1233	0.0359	0.0030	0.0010	0.0008
MAE	0.1687	0.1482	0.0258	0.0025	0.0007	0.0003
R^2	0.5660	0.6019	0.8524	0.8996	0.9055	0.9472



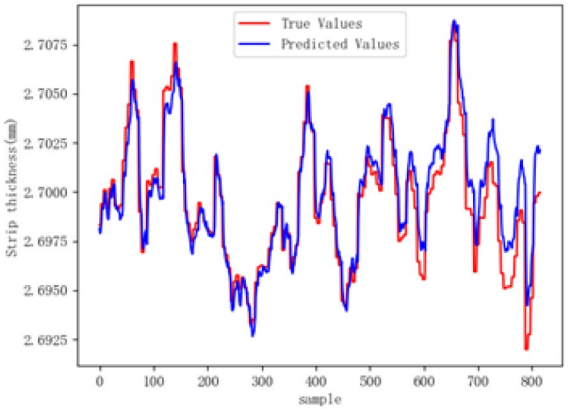
(a) LSTM



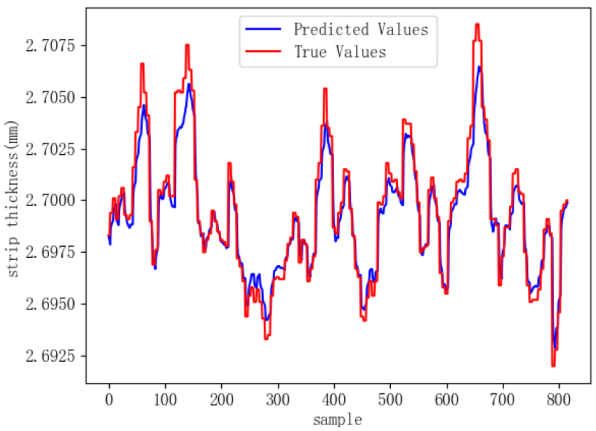
(b) S-LSTM



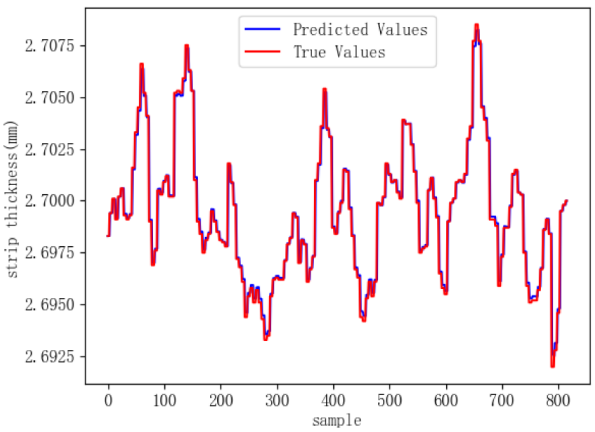
(c)GSTAE



(d) STA-LSTM



(e) DAE-STAFN



(f) DAE-STCAFN

Fig.18 Prediction results of strip thickness by (**a**) S-LSTM, (**b**) LSTM, (**c**) GSTAE, (**d**) STA-LSTM, (**e**) DAE-STAFN, (**f**) DAE-STCAFN

strip thickness, the R^2 of the proposed model is 0.9472, indicating that the proposed model has significant advantages in noise reduction effect, prediction performance and generalization. However, DAE-STCAFN model involves the learning of sample relationships, which is suitable for the learning of small sample data, and may takes a long time for large quantities of data. In the following research, the learning ability of the model should be improved.

Acknowledgements This work was supported by the Key Project of Natural Science Foundation of Gansu Province (24JRRA173), Gansu Province University young doctor support project (2024QB-037), Lanzhou youth science and technology talent innovation project (2023-QN-36), National Natural Science Foundation of China (No.62263021), the Science and Technology Project of Gansu Province (21JR7RA206), and the College Industrial Support Project of Gansu Province (2023CYZC-24).

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Conflict of interest There is no conflict of interest.

References

1. Li W, Zhao C (2017) Latent variable based concurrent multi-trends analysis method for monitoring batch processes with irregular and limited batches, The. Can J Chem Eng 95:1817–1829
2. Gao X, Xu Z, Li Z, Wang P (2020) Batch process monitoring using multiway Laplacian autoencoders, The. Can J Chem Eng 98:1269–1279
3. Wang K, Gopaluni RB, Chen J, Song Z (2018) Deep learning of complex batch process data and its application on quality prediction. IEEE Trans Industr Inf 16:7233–7242
4. Yao H, Zhao X, Li W, Hui Y (2023) Multi-stage fusion regression network for quality prediction of batch process, The. Can J Chem Eng 101:6977–6994
5. Ren L, Meng Z, Wang X, Zhang L, Yang LT (2020) A data-driven approach of product quality prediction for complex production systems. IEEE Trans Industr Inf 17:6457–6465
6. Md AQ, Jha K, Haneef S, Sivaraman AK, Tee KF (2022) A review on data-driven quality prediction in the production process with machine learning for industry 4.0. Processes 10(10):1966
7. Yan W, Guo P, Li Z (2016) Nonlinear and robust statistical process monitoring based on variant autoencoders. Chemometr Intell Lab 158:31–40
8. Wang Y, Luo J, Liu C, Yuan X, Wang K, Yang C (2022) Layer-wise residual-guided feature learning with deep learning networks for industrial quality prediction. Ieee T Instrum Meas 71:1–11
9. Akande T, Alabi O, Ajagbe S (2022) A deep learning-based CAE approach for simulating 3D vehicle wheels under real-world conditions. In Artificial Intelligence and Applications
10. Wang Y, Zheng Y, Fang H, Wang Y (2014) ARMAX model based run-to-run fault diagnosis approach for batch manufacturing process with metrology delay. Int J Prod Res 52:2915–2930
11. Jiang Q, Yan X, Yi H, Gao F (2019) Data-driven batch-end quality modeling and monitoring based on optimized sparse partial least squares. IEEE T Ind Electron 67:4098–4107
12. Sun Y, Qin W, Zhuang Z, Xu H (2021) An adaptive fault detection and root-cause analysis scheme for complex industrial processes using moving window KPCA and information geometric causal inference. J Intell Manuf 32:2007–2021
13. García V, Sánchez JS, Rodríguez-Picón LA, Méndez-González LC, Ochoa-Domínguez HDJ (2019) Using regression models for predicting the product quality in a tubing extrusion process. J Intell Manuf 30:2535–2544
14. Lughofer E, Zavoianu A-C, Pollak R, Pratama M, Meyer-Heye P, Zörrer H, Eitzinger C, Radauer T (2019) Autonomous supervision and optimization of product quality in a multi-stage manufacturing process based on self-adaptive prediction models. J Process Control 76:27–45
15. Leong WC, Bahadori A, Zhang J, Ahmad Z (2021) Prediction of water quality index (WQI) using support vector machine (SVM) and least square-support vector machine (LS-SVM). Int J River Basin Manag 19:149–156
16. Sun Q, Ge Z (2018) Probabilistic sequential network for deep learning of complex process data and soft sensor application. IEEE Trans Industr Inf 15:2700–2709
17. Wang Z, Feng L, Li Y et al (2024) Prediction of dust emissions in highway subgrade-filling construction based on deep neural network. HighTech Innov J 5(2):259–271
18. Shang C, Yang F, Huang D, Lyu W (2014) Data-driven soft sensor development based on deep learning technique. J Process Control 24:223–233
19. Yan W, Tang D, Lin Y (2016) A data-driven soft sensor modeling method based on deep learning and its application. IEEE T Ind Electron 64:4237–4245
20. Yuan X, Huang B, Wang Y, Yang C, Gui W (2018) Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. IEEE Trans Industr Inf 14:3235–3243
21. Liu C, Wang Y, Wang K, Yuan X (2021) Deep learning with non-local and local structure preserving stacked autoencoder for soft sensor in industrial processes. Eng Appl Artif Intel 104:104341
22. Sun Q, Ge Z (2020) Gated stacked target-related autoencoder: a novel deep feature extraction and layerwise ensemble method for industrial soft sensor application. IEEE Trans Cybernetics 52:3457–3468
23. Li S, Li W, Cook C, Zhu C, Gao Y (2018) Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5457-5466)
24. Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput 31:1235–1270
25. Yuan X, Li L, Wang Y, Yang C, Gui W (2020) Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network. Can J Chem Eng 98:1377–1389
26. Xiang S, Qin Y, Zhu C, Wang Y, Chen H (2020) Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. Eng Appl Artif Intel 91:103587
27. Ren L, Meng Z, Wang X, Lu R, Yang LT (2020) A wide-deep-sequence model-based quality prediction method in industrial process analysis. IEEE Trans Neural Netw Learning Syst 31:3721–3731
28. Yuan X, Li L, Shardt YA, Wang Y, Yang C (2020) Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. IEEE T Ind Electron 68:4404–4414
29. Özçelik YB, Altan A (2023) Overcoming nonlinear dynamics in diabetic retinopathy classification: a robust AI-based model with chaotic swarm intelligence optimization and recurrent long short-term memory. Fractal Fract 7(8):598

30. Lebedev IS, Sukhoparov ME (2024) Improving the quality indicators of multilevel data sampling processing models based on unsupervised clustering. Emerg Sci J 8(1):355–371
31. Razak SFA, Yogarayan S, Ullah A (2024) Preventing impaired driving using IoT on steering wheels approach. HighTech Innov J 5(2):400–409
32. Chen H, Ru J, Long H et al (2024) Semi-supervised adaptive pseudo-label feature learning for hyperspectral image classification in internet of things. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2024.3412925
33. Zhang M, Li X, Wang R (2021) Incipient fault diagnosis of batch process based on deep time series feature extraction. Arab J Sci Eng 46:10125–10136
34. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need, Advances in neural information processing systems, 30
35. Ji C, Ma F, Wang J, Sun W (2023) Profitability related industrial-scale batch processes monitoring via deep learning based soft sensor development. Comput Chem Eng 170:108125
36. Tuo B, Zhao X, Sun K et al (2024) Soft sensor model for nonlinear dynamic industrial process based on GraphSAGE-IMATCN. Process Saf Environ Prot 191:1131–1147
37. Birol G, Ündey C, Çinar A (2002) A modular simulation package for fed-batch fermentation: penicillin production. Comput Chem Eng 26:1553–1565
38. Hui Y, Zhao X (2018) Batch process monitoring based on WGNPE–GSVDD related and independent variables. Chinese J Chem Eng 26:2549–2561
39. Yao H, Zhao X, Li W, Hui Y (2022) Quality-related fault monitoring for multi-phase batch process based on multiway weighted elastic network. Chemometr Intell Lab 223:104528
40. Yuan X, Li L, Wang Y (2019) Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. IEEE Trans Industr Inf 16:3168–3176
41. Li S, Wang J, Chen S (2021) Quality prediction of strip in finishing rolling process based on GBDBN-ELM. J Sensors 2021(1):9943153
42. Ma L, Dong J, Peng K et al (2017) A novel data-based quality-related fault diagnosis scheme for fault detection and root cause diagnosis with application to hot strip mill process. Control Eng Pract 67:43–51

Yan Zhang is the main research is fault detection and quality prediction of batch process.

Jie Cao is the Doctoral Supervisor, Professor.

Xiaoqiang Zhao is the Doctoral Supervisor, Professor.

Yongyong Hui is the PhD, Associate Professor.

PAPER

Deep quality-related stacked isomorphic autoencoder for batch process quality prediction

IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **35** (2024) 116202 (18pp) https://doi.org/10.1088/1361-6501/ad6684

Deep quality-related stacked isomorphic autoencoder for batch process quality prediction

Yan Zhang[1,2], **Jie Cao**[1,4,*], **Xiaoqiang Zhao**[1,2,3] **and Yongyong Hui**[1,2,3]

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China
[2] Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou, People's Republic of China
[3] National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China
[4] Manufacturing Informatization Engineering Research Center of Gansu Province, Lanzhou, People's Republic of China

E-mail: zhyan0423@163.com

Received 22 March 2024, revised 3 July 2024
Accepted for publication 23 July 2024
Published 9 August 2024

Abstract

Batch processes play an important role in modern chemical industrial and manufacturing production, while the control of product quality relies largely on online quality prediction. However, the complex nonlinearity of batch process and the dispersion of quality-related features may affect the quality prediction performance. In this paper, a deep quality-related stacked isomorphic autoencoder for batch process quality prediction is proposed. Firstly, the raw input data are reconstructed layer-by-layer by isomorphic autoencoder and the raw data features are obtained. Secondly, the quality-related information is enhanced by analyzing the correlation between the isomorphic feature of each layer of the network and the output target, and constructing a correlation loss function. Thirdly, a deep quality-related prediction model is constructed to predict the batch process quality variables. Finally, experimental validation was carried out in penicillin fermentation simulation platform and strip hot rolling process, and the experimental results demonstrated the feasibility and effectiveness of the model proposed in this paper for the quality prediction of the batch process.

Keywords: batch processes, quality prediction, isomorphic autoencoder, quality-related information, maximum information coefficient

1. Introduction

Batch processes are highly flexible and adaptable and have been widely used in industrial production, such as bioferment-ation, energy, and semiconductors. To improve the product quality and economic benefits of batch processes, it is essential to realize online monitoring of key variables [1, 2]. However, unlike continuous production, batch processes are segmented and discontinuous due to the existence of batch production characteristics, resulting in the collection of segmented and discontinuous data. Therefore, the three-dimensional characteristics of the production data of batch processes need to be taken into account in quality prediction, and the construction of its quality prediction models is more complex than that of continuous processes [3]. At the same time, in large-scale industrial production systems, the safety of batch processes is largely challenged by the complex equipment operating environments and production conditions that make it difficult to measure some important quality variables directly online, while offline measurements lead to serious delays [4, 5].

* Author to whom any correspondence should be addressed.

With the development of digital and intelligent production, a large amount of process data containing useful information have been collected and stored. How to extract the information from process data and build data-driven quality prediction models has become a growing concern in industry and academia [6, 7]. The main idea of data-driven soft-based measurement is to extract the representations from available historical process variable data, then to establish the relationships between the representations and the target variables, and finally to measure hard-to-obtain quality variables by using available variables in an online program. Multivariate statistical machine learning techniques, such as principal component analysis (PCA) [8], partial least squares regression (PLSR) [9], principal component regression (PCR) [10] and partial least squares (PLS) [11] have been developed to enable soft sensor techniques through low-dimensional data embedding. Nomikos and MacGregor extended traditional PLS to batch processes for quality prediction and developed a multi-way partial least squares method (MPLS) [12]. At the same time, some scholars have explored kernel-based methods for dealing with nonlinear relationships between process variables and quality variables. The kernel methods, such as kernel partial least squares (KPLS) [13, 14], kernel principal component regression (KPCR) [15] and support vector regression (SVR) [16], map the nonlinear variables into a high-dimensional space and then perform feature extraction and regression. However, they still have the problem of excessive computation and are difficulty in determining a reasonable kernel function for dealing with complex nonlinear batch process data.

Although data-driven soft-sensor-based methods have the ability to collect a large amount of process data, and establish a regression model between the process variable and the quality variable to realize the prediction of difficult-to-measure quality variables indirectly [17]. However, these methods basically perform shallow feature extraction, which are not sufficient to extract complex features for large-scale and complex industrial production processes, so it is difficult to realize high-precision and reliable prediction. In recent years, as a powerful machine learning method with the ability to process complex data and extract advanced features, deep learning has opened up new possibilities for industrial process quality prediction [18–20]. Stacked autoencoder (SAE), as a deep learning method, has the ability to automatically learn feature representation and dimensionality reduction, and is widely used in the field of quality prediction [21]. Considering that SAE reconstruction is not fully accurate enough, Yuan *et al* [22] proposed a new deep stacked isomorphic autoencoder (SIAE) to obtain a better feature representation of the raw input data. However, SIAE only seeks to learn deep feature representations of the raw input data, when it is used for quality prediction modeling, it is difficult to learn the features related to quality. To realize quality-related feature representations, Yuan *et al* [23] proposed a variable weighted stacked autoencoder (VW-SAE), which achieved the selection of quality variables that were important for its input layer, but the VW-SAE ignored the deep representations of input variables. Yuan *et al* [24] proposed stacked quality-driven autoencoders that use quality data to guide the learning of quality-related features. Jiang *et al* [25] proposed a soft sensor model for deep quality-related representation learning via mutual information and use it for penicillin production process prediction. Wang *et al* [26] used LSTM method to extract comprehensive quality-related hidden features from the long time series of each stage, which were further integrated by SAE to predict the chain-growth batch copolymerisation process. However, those methods did not fully consider the importance of input data reconstruction for deep feature extraction. Since input features and quality-related features are of equal importance in the quality prediction model, we should consider both raw input features and quality-related features comprehensively when performing batch process quality prediction.

This paper proposes a deep quality-related stacked isomorphic autoencoder for batch process quality prediction model. The batch process data are expanded as two-dimensional data along the variable direction, and the expanded input data are constructed through the isomorphic autoencoder, so as to reduce the error accumulation in the training process. At the same time, the correlation between input variables and quality variables is calculated layer-by-layer, and the variables with higher correlation are retained to reduce the amount of redundancy. The correlation error loss function is designed to pre-train the model, then the extracted deep quality-related features are fed into the regression network for batch process quality variable prediction. The main contributions are as follows:

1. The raw input data features are constructed by stacked isomorphic autoencoder layer-by-layer.
2. For the inputs to each layer of the network, the correlation between inputs and target outputs is calculated by Maximal information coefficient- General Jaccard Coefficient (MIC-GJC). The GJC is originally used to compare the similarity of the sample sets, and combined with MIC for similarity metrics to further access deep quality-related features.
3. The correlation error function is constructed and the network structure is optimized. The acquired deep quality-related features are fed into a regression network for batch process quality variable prediction.

The reminder of this paper is arranged as follows. Section 2 reviews SAE. The proposed quality prediction model for Batch Process is described in detail in section 3. Section 4 describes the experimental studies conducted on the penicillin fermentation process and the hot strip mill process. Finally, Section 5 provides the conclusion of this paper.

2. Autoencoder and stacked autoencoder

Autoencoder (AE) is a neural network model. Its network structure is shown in figure 1. It consists of an encoder and a decoder, which work together to reconstruct the input data. The encoder compresses the m-dimensional input data $X = \{x_1, x_2, \cdots, x_m\}$ into a lower n-dimensional representation $h =$

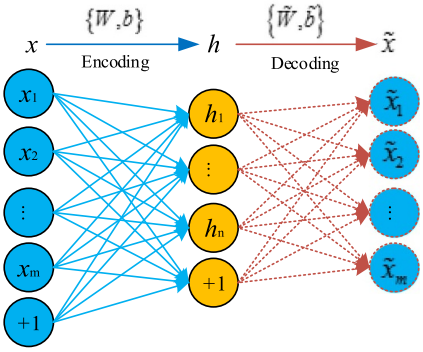


Figure 1. AE structure diagram.

$\{h_1, h_2, \cdots, h_n\}$, called the latent space, while the decoder reconstructs the original input from this latent representation as $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_m\}$. The nonlinear mapping is shown in equation (1).

$$\begin{aligned} \mathrm{h} &= f_e\left(\mathrm{Wx} + \mathrm{b}\right) \\ \tilde{\mathrm{x}} &= f_d\left(\tilde{\mathrm{W}}\mathrm{h} + \tilde{\mathrm{b}}\right) \end{aligned} \tag{1}$$

Where f_e and f_d, W and $\tilde{\mathrm{W}}$, b and $\tilde{\mathrm{b}}$ are the activation functions, weight matrixes, bias vectors of the encoder and decoder, respectively.

The reconstruction error is calculated by comparing the reconstructed output with the original input. The model parameter set $\theta_{\mathrm{AE}} = \left\{W, b; \tilde{W}, \tilde{b}\right\}$. is updated by using back-propagation and gradient descent to minimize the reconstruction error. The reconstruction error is expressed as follows:

$$L\left(\theta_{\mathrm{AE}}\right) = \frac{1}{M}\sum_{m=1}^{M}\left\|\tilde{x}_m - x_m\right\|^2. \tag{2}$$

Multiple iterations are made until the model converges and the reconstruction error is minimized.

Stacked Autoencoder (SAE) is an extension of the basic AE, where the output of each auto-encoder is used as an input to the next autoencoder, and each autoencoder can be trained to map from the raw data to a low-dimensional representation. Figure 2 shows the schematic diagram of SAE structure composed of k AE structures. The learning process of SAE network consists of unsupervised pre-training and supervised fine-tuning. The red arrows represent the pre-training process and the green arrows represent the supervised fine-tuning process. The individual training for each AE process is realized through pre-training. Following the direction of the red arrow in figure 2, the raw input vector $x \in R^{d_x}$ is fed to the input layer of the AE1, and its parameter set $\{W_1, b_1\}$ is trained by solving equation (1), and the hidden layer feature h_1 is obtained. After the AE1 is trained, h_1 is used as input for the AE 2. By optimizing the loss function shown in equation (3), the parameter set $\{W_2, b_2\}$ and the hidden layer feature h_2 can be obtained. After layer-by-layer training to the kth AE, the parameter set $\{W_k, b_k\}$ and the hidden layer feature h_k are obtained, and the pre-training of the whole SAE is completed.

$$L\left(\theta_{AE\text{-}k}\right) = \frac{1}{M}\sum_{i=1}^{M}\left\|\tilde{h}_{k-1}(i) - h_{k-1}(i)\right\|^2 \tag{3}$$

where k represents the number of hidden layers.

After completing the pre-training of the SAE, a fully connected layer is added to the top of the SAE in order to perform regression. Supervised fine-tuning of the network is performed in the direction shown by the green arrow in the figure 2. Through forward propagation, the estimated quality output value $\hat{y}$ can be obtained:

$$\hat{y} = f_o\left(\mathrm{W}_O\mathrm{h}_k + \mathrm{b}_o\right) \tag{4}$$

where f_o is the activation function of the output layer, W_O and b_o represent the weight matrix and bias vector of the output layer, respectively.

3. The quality prediction model for batch process

The training of SAE-based neural networks consists of unsupervised pre-training and supervised fine-tuning processes. It is important to note that each layer during the pre-training process may result in the loss of original data information. In addition, the hidden features extracted at each layer may contain the information that is irrelevant to the output prediction. As the pre-training proceeds, the irrelevant information propagates forward to the next layer until it reaches the last layer. This results in a large amount of redundant information in the extracted high-level features, degrading the prediction performance. In order to change the impact of this unsupervised reconstruction, it is important to ensure that the amount of raw information lost is minimized on the one hand, and on the other hand, to make the learned features highly correlated with the output variables, the reconstruction is more accurate. Based on the above analysis, a deep quality-related stacked isomorphic autoencoder for batch process quality prediction is proposed and efficient quality prediction model based on the learned correlation representation is constructed.

3.1. Batch process data preprocessing

The batch process data composition has a batch dimension in addition to the variable and time dimensions: $X(K \times I \times J)$, K represents the samples, I represents the batch, and J represents the variable. The process data are first unfolded along the variables. Figure 3 shows the schematic diagram of data processing.

Since each variable has a different data range, the values are not of the same order of magnitude, so in the training process of neural network, it may cause the problem that the data convergence speed is too slow. Therefore, the data of different orders of magnitude are unified to the same order of magnitude by normalization process, this makes the data more

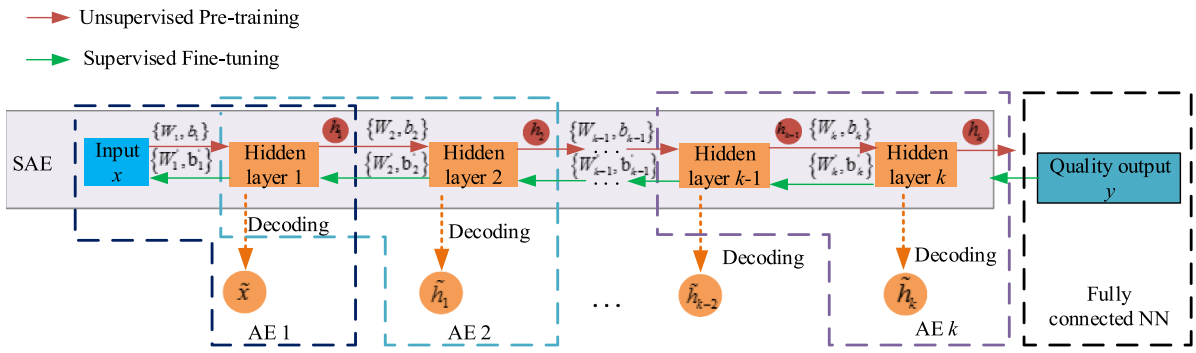


Figure 2. SAE structure diagram.

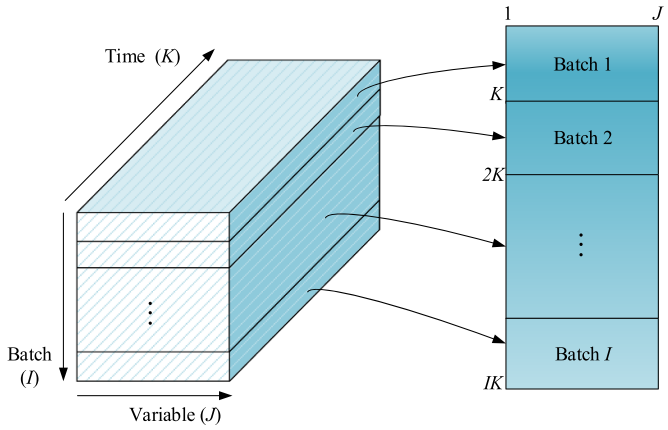


Figure 3. The schematic diagram of 3D-data processing.

meaningful and can better reflect the characteristics of the data. By applying a linear transformation to the original data, the results of the transformation fall into the interval [0,1]. The maximum-minimum function is:

$$x_i^* = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

where, x_i^* represent the normalized value, x_i represent the feature value corresponding to the ith sample, $x_{\min}$ and $x_{\max}$ represent the maximum value and minimum value of the characteristic corresponding to the samples, respectively.

3.2. Deep quality-related feature extraction

Quality irrelevant features do not provide valid information for the prediction of quality variables. Therefore, for each layer of input variables, quality-related variables are retained and quality-irrelevant information is eliminated. Maximal information coefficient (MIC) [27] can detect not only the linear relationship between two variables, but also the nonlinear relationship between them. MIC further considers higher order relationships between variables compared to MI. The formula for calculating the maximum information coefficient is shown as equation (6)

$$\mathrm{MIC} = \max\left\{I\left(x\left(u\right),y\right)/\log_2 \min\left\{n_{x(u)},n_y\right\}\right\} \tag{6}$$

where, $n_{x(u)} \cdot n_y < B(n)$ represents the upper bound of the grid division, $B(n) = n^{0.6}$, $n_{x(u)}$ and n_y are the number of cells along the $x(u)$ and y axes, respectively. $I(x(u),y)$ is the value of the mutual information along $x(u)$ and y, which can be obtained by equation (7)

$$\begin{aligned} I\left(x\left(u\right),y\right) &= H\left(x\left(u\right)\right) + H\left(y\right) - H\left(x\left(u\right),y\right) \\ &= \sum_{k=1}^{n_{x(u)}} \rho\left(x^k\left(u\right)\right) \log_2 \frac{1}{\rho\left(x^k\left(u\right)\right)} + \sum_{s=1}^{n_y} \rho\left(y^s\right) \log_2 \frac{1}{\rho\left(y^s\right)} \\ &\quad - \sum_{k=1}^{n_{x(u)}} \sum_{s=1}^{n_y} \rho\left(x^k\left(u\right)y^s\right) \log_2 \frac{1}{\rho\left(x^k\left(u\right)y^s\right)} \end{aligned} \tag{7}$$

where, $H(x(u))$ and $H(y)$ are the entropies of $x(u)$ and y, respectively, and $H(x(u),y)$ is the joint entropy of $x(u)$ and y.$\rho(x^k(u)$ and $\rho(y^s)$ are the probability density values of $x^k(u)$and y^s, respectively. $\rho(x^k(u)y^s)$ is the joint probability density value of the two variables $x(u)$ and y. The range of MIC values is [0,1].

After data preprocessing, the spatial differences in batch processing data are considered more important than numerical differences. Therefore, MIC can be combined with the General Jaccard Coefficient (GJC) [28] to construct a hybrid correlation measure to assess the correlation between variables. GJC is used to compare the similarity and diversity of samples. For finite sample sets U and Y, $J(U,\mathrm{Y})$ for computing the correlation between U and V is defined as equation (8):

$$J(U,Y) = \frac{|U \cap Y|}{|U \cup Y|} = \frac{|U \cap Y|}{|U| + |Y| - |U \cap Y|} \tag{8}$$

where U, Y denote two sample sets respectively. If both U and Y are empty sets, then let$J(U,Y) = 1$. The numerator of GJC is the inner product of the vectors and the denominator of GJC is the quadratic sum of the vectors minus the inner product of the vectors. GJC is shown in equation (9)

$$\mathrm{GJC}\left(x\left(u\right),y\right) = \frac{\sum_{i=1}^{l} x_i\left(u\right) y_i}{\sum_{i=1}^{l} \left(x_i\left(u\right)\right)^2 + \sum_{i=1}^{l} \left(y_i\right)^2 - \sum_{i=1}^{l} x_i\left(u\right) y_i}. \tag{9}$$

It is worth noting that the value of GJC ranges from 0 to 1, and it has the characteristics of normalization and symmetry. When the variables are independent of each other, the value of GJC is 0. Therefore, $\mathrm{sim}(x(u),y)$ is the mixed correlation measure for variables $x(u)$ and y, which is shown in equation (10)

$$\mathrm{sim}\left(x\left(u\right),y\right)=\omega\mathrm{MIC}\left(x\left(u\right),y\right)+\left(1-\omega\right)\mathrm{GJC}\left(x\left(u\right),y\right) \tag{10}$$

where, ω denotes the mixing coefficient, which takes the value between 0 and 1. $\omega=0.5$ is chosen in this paper [29].

3.3. Establishment of quality prediction models

The structure of the isomorphic autoencoder (IAE) is described in the literature [22]. As an autoencoder, IAE also consists of input, hidden and output layers. Its input vectors are usually raw input data or feature states learned in one of the hidden layers. The output layer of AE learns the same function for its input vectors. In contrast, the output layer of IAE is fixed and isomorphic to the observed raw data. That is, the output layer of an IAE always seeks to reconstruct the original input data. In order to extract deep quality-related features, supervised and semi-supervised layered pre-training strategies are introduced to extract relevant features layer-by-layer. Assumed that the variable vectors of the input and hidden layers are x and h, respectively, the output layer aims to obtain the reconstructed raw data $\tilde{x}$ from the learned abstract features h. We still use $\{W,b\}$ and $\left\{\tilde{W},\tilde{b}\right\}$ to denote the parameters of the hidden and output layers, respectively. Also, the corresponding activation functions are f and $\tilde{f}$. Thus, the computation of $\tilde{x}$ is shown in equation (11)

$$\tilde{x}=\tilde{f}(f(x)). \tag{11}$$

It is assumed that all training data consists of labeled and unlabeled data, where the labeled data are$\{X_l,Y_l\}=\{(x_1,y_1),\cdots,(x_{M_l},y_{M_l})\}$, the unlabeled data are$\{X_u\}=\{x_1,x_2,\cdots,x_{M_u}\}$, where M_l and M_u are the numbers of labeled and unlabeled samples, respectively. After expanding and normalizing the batch process data, the process variables are considered as the inputs to IAE1. The mixed correlation between each variable and the output target variable is calculated by using equation (10) for labeled data. The higher the correlation, the more relevant it is to the target output variable. The correlation index can be calculated by equation (12):

$$s_i=\mathrm{sim}\left(i\right)/\sum_{l=1}^{M_l}\mathrm{sim}\left(l\right). \tag{12}$$

The schematic diagram of deep quality related feature extraction for stacked isomorphic autoencoder by MIC-GJC is shown in figure 4, the pink circle indicates the quality-related variable in each layer of the network. Blue, green and light blue circles represent the uncorrelated variables eliminated in each layer of the network, respectively. During the pre-training process, quality-related variables are selected for each hidden layer feature. The variables with higher correlation are retained as the input information for the next layer of the network to extract the deep quality-related features.

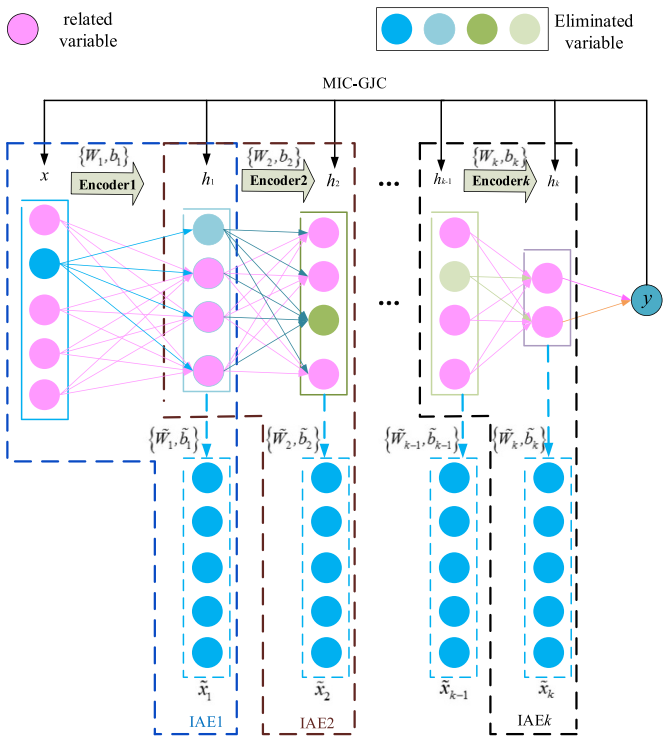


Figure 4. Schematic diagram of deep quality-related feature extraction.

In order to optimize the model parameters, the relevant loss function $J(W,\tilde{W},b,\tilde{b})$ is constructed and minimized between the raw input data and reconstructed one as shown in equation (13)

$$\begin{aligned}J\left(W,\tilde{W},b,\tilde{b}\right)&=s_i\sum_{i=1}^{N}J\left(W,\tilde{W},b,\tilde{b}\right)/2N\\&=s_i\sum_{i=1}^{N}\left\|\tilde{x}_i-x_i\right\|^2/2N.\end{aligned} \tag{13}$$

Features h^1 is as its input vector to IAE 2. After mapping h^1 to the hidden layer of IAE 2 by the encoder, h^2 is generated by a nonlinear activation function. After that, the output layer of IAE2 seeks to reconstruct the raw data $\tilde{x}_2$ by decoder. By minimizing the correlation loss function implements the pre-training of IAE 2. Similarly, all k IAEs are sequentially pre-trained and then stacked one by one to obtain k IAEs. Thus, the features of the raw input x can be gradually learned from IAE 1 to IAE k from the lowest hidden layer h^1 to the highest hidden layer h^k. In each IAE, the variable correlation index is computed from the labeled data. IAEs are then trained using the relevant loss function to make the hidden features more relevant to the target variables. The final extracted deep quality-related features are fed into the regression network for quality variable prediction. The structure of the whole network is shown in figure 5.

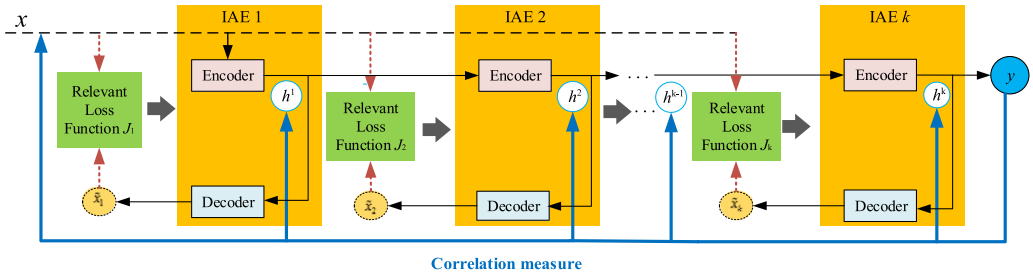


Figure 5. Quality prediction model structure diagram.

3.4. The process of quality prediction model

Overall, the main process of quality prediction model includes offline modeling and online prediction, which are shown in figure 6. The specific steps of offline modeling and online prediction are as follows:

(1) Offline modeling:
Step1: Collect training data from batch process;

Step2: Expand and normalize the training data along the variable directions;

Step3: For the labeled data $\{X_l, Y_l\}$, calculate the correlation index between the input data and the target data by using MIC-GJC;

Step4: Retain the data with larger correlation indices as input to IAE1, and generated isomorphic data by the decoder. The correlation error loss function is constructed according to equation (13) to train IAE1;

Step5: In a similar way, pre-training is performed layer-by-layer until the *k-th* IAE module is obtained. At the same time, depth quality-related features are obtained;

Step6: After pre-training is complete, a regression neural network is added to the top layer of the network based on deep quality-related features and quality variables;

Step7: The parameters of the whole network were optimized through a fine-tuning process. Then, a quality prediction model is established for the batch process.

(2) Online prediction:
Step1: Collect real-time data samples;

Step2: Expand and normalize the real-time data along the variable directions;

Step3: The processed data are fed into the established quality prediction model to predict the quality variables.

Step4: The obtained values of quality prediction variables are inversely normalized to obtain quality prediction values.

4. Experimental analysis

In this section, the proposed quality prediction model is applied to two industrial processes to verify its prediction performance. One is the penicillin fermentation simulation process, which is a typical nonlinear batch process, and the other is the actual strip hot rolling process, which is based on the production data from the hot rolling plant of a steel company.

4.1. Penicillin fermentation process

The proposed model is applied on batch penicillin fermentation process for the validation of penicillin concentration and biomass concentration prediction. The production process of penicillin, a potent antibiotic, consists mainly of bacterial culture, fermentation, extraction, crystallization and purification processes. Usually, the reaction and production of each batch need to last for hundreds of hours. Figure 7 is the main flow chart of the penicillin production process. The experimental data applied in this paper are obtained through the Pensim simulation platform [30]. The platform is a commonly used simulation platform that can comprehensively reflect the penicillin fermentation process with the Birol model as the core, fully considering various physical quantities and biomass, with high authority and practicability. The effectiveness of the proposed model is demonstrated by comparing with some existing methods: Stacked Autoencoder (SAE), Stacked isomorphic autoencoder (SIAE), Gated Stacked Target-Related Autoencoder (GSTAE) [31], Variable-Wise Weighted SAE (VW-SAE) and Target-Related Stacked Autoencoder (TSAE). Where TSAE is a quality prediction model that extracts the correlation between the output features of the trained stacked autoencoder and the target variable.

To test the performance of the proposed model, the model evaluation indicators such as the mean absolute error (MAE), the root mean squared error (RMSE) and the coefficient of determination (R^2) are introduced to quantitatively and comprehensively evaluate the quality prediction effectiveness of each model. MAE represents the mean value of the absolute error and reflects the actual situation of the prediction error. Its expression is shown in equation (14):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right| \tag{14}$$

where, n represents the total number of samples, y_i represents the actual value, $\widehat{y}_i$ represents the predicted value.

RMSE denotes the prediction error for the entire validation or test set and it is defined as shown in equation (15). The smaller the RMSE, the better the predicted performance of the model

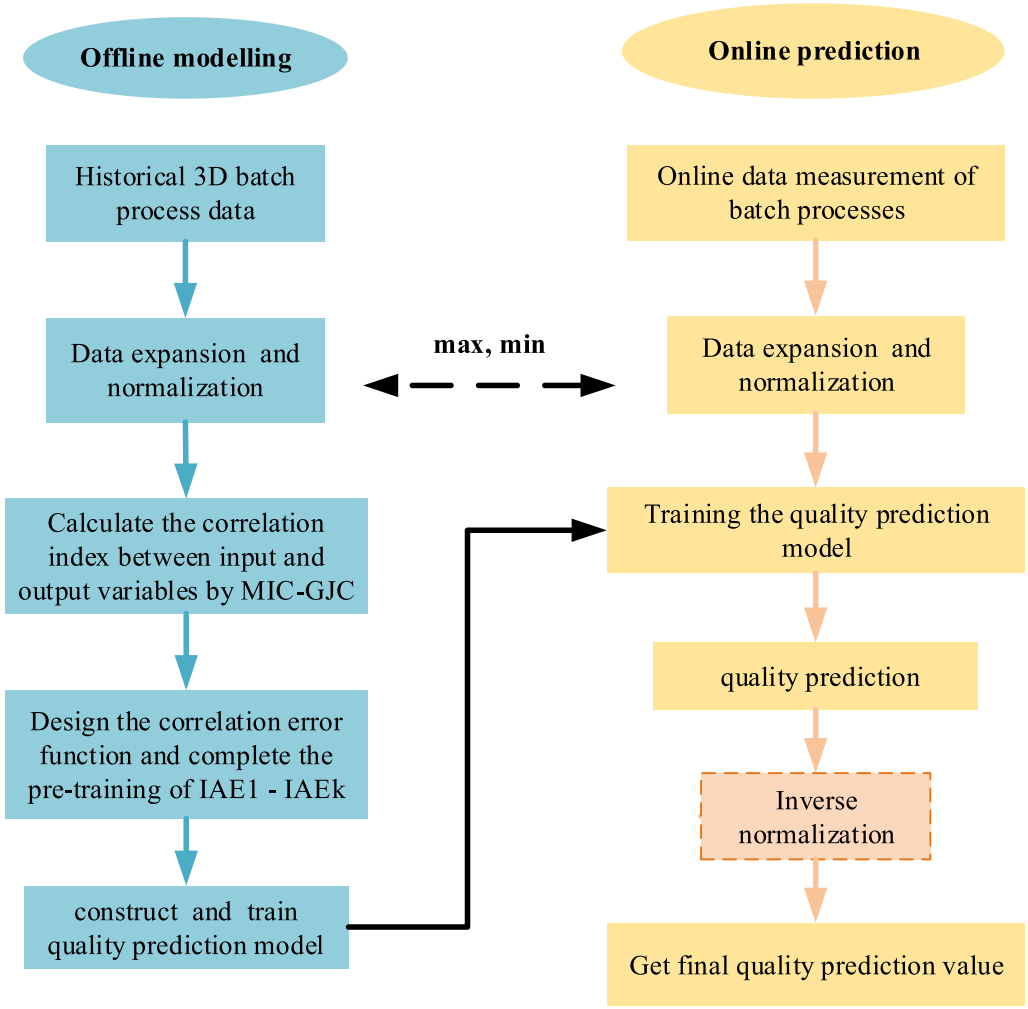


Figure 6. Processes for quality prediction modeling.

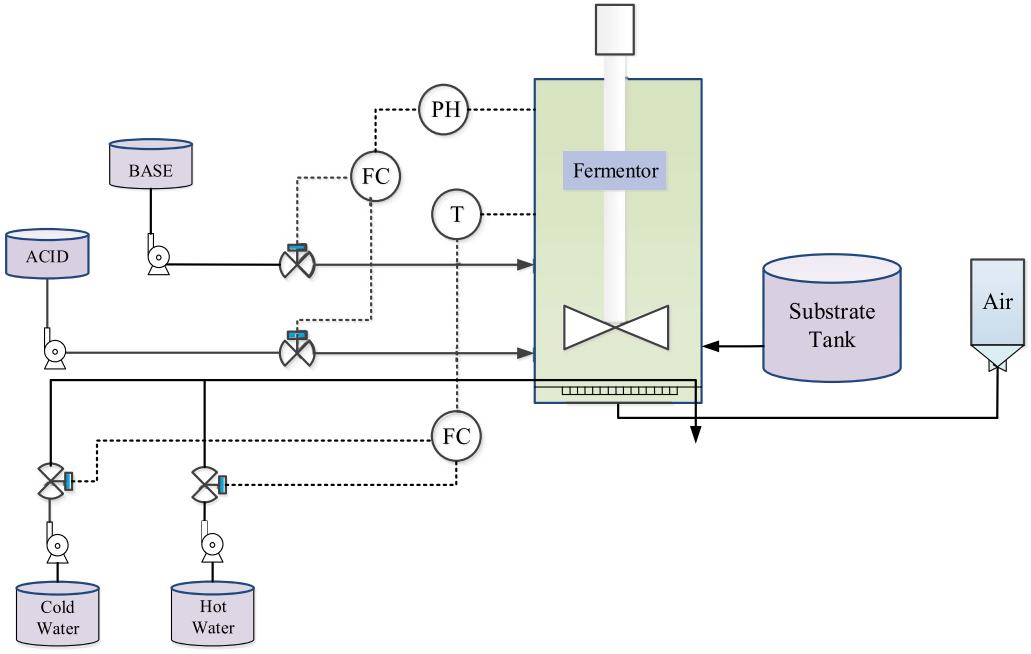


Figure 7. Penicillin fermentation flow chart.

Table 1. Process variables and quality variables.

Variable		Description	Unit
Measurements	$x1$	Aeration rate	L/h
	$x2$	Agitator power	W
	$x3$	Substrate feed flow rate	L/h
	$x4$	Substrate feed temperature	K
	$x5$	Substrate concentration	g/L
	$x6$	O_2	%
	$x7$	Culture volume	L
	$x8$	CO_2	%
	$x9$	PH	1
	$x10$	Temperature	K
Quality	$y1$	Penicillin concentration	g/L
	$y2$	Biomass concentration	g/L

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2} \tag{15}$$

R^2 is used to measure the fitting degree between the actual value and the predicted value. R^2 is defined as equation (16)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \hat{y}\right)^2} \tag{16}$$

where $\hat{y}$ is the mean of the actual value. The ability of the model to explain the variance of the output data can be seen by the value of R^2. $R^2 \leqslant 1$, the closer R^2 is to 1, the better the model fit.

Ten variables during penicillin fermentation were selected as process variables, as shown in table 1, and the quality variables were penicillin concentration and biomass concentration. The fermentation cycle was set to be 400 h, the sampling interval was 1 h, and a total of 40 batches of penicillin fermentation process data were collected. The fermentation process data were divided into training set (60%), validation set (20%) and test set (20%).

The setting of hyperparameters is a key issue in the process of setting network parameters. In this paper, the combination of random search and trial-and-error method is chosen to determine the optimal hyperparameters [31]. First, several sets of hyperparameters are generated and run for several times to obtain experimental data. Then, the roughly optimal intervals are found by comparison. Finally, the detailed set of optimal hyperparameters is determined. In addition, the number of hidden layers is decided according to the complexity of the processed data. Too few layers would result in poor model performance, and too many layers would increase the amount of computation and there is no significant improvement for model performance. The number of hidden layers for several models involved in this paper are 11, 7, 5, 3.

4.1.1. Penicillin concentration prediction. The penicillin concentration during penicillin fermentation was predicted using SAE, SIAE, GSTAE, VW-SAE, TSAE and the model proposed in this paper, respectively. Figure 8 shows the loss curves for the fine-tuning stage of each model, it can be seen that the initial loss value of the fine-tuning process of the proposed model is 0.0055 at the end of the pre-training, and after 150 epochs of training, the final loss value of the proposed model is smaller than that of the other models and tends to be stable. This indicates that the proposed model has good convergence. This also further indicates that the proposed model can extract more relevant features for quality prediction.

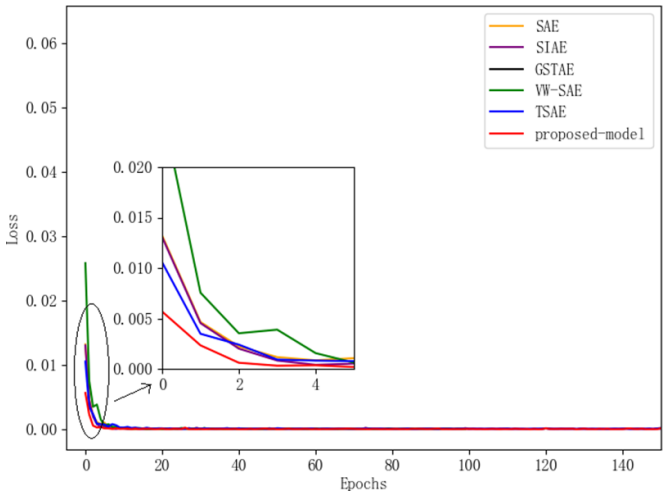
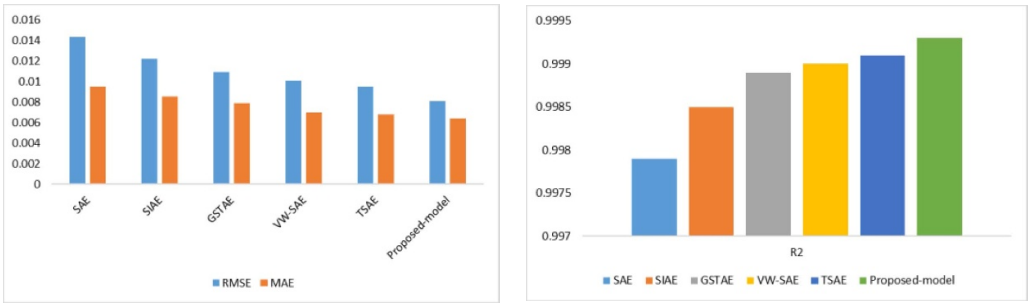


Figure 8. Loss curves of the training process.

The SAE, SIAE, GSTAE, VW-SAE, TSAE, and proposed models are used to predict penicillin concentrations, respectively. Considering the generalization performance of the models, the test sets of 8 batches are divided equally into 4 sub-test sets. The average RMSE, MAE and R^2 values predicted for each of the four sub-test sets individually are shown in table 2, it can be clearly seen that the proposed model in this paper has lower RMSE and MAE than other models, which indicates that the proposed model has better robustness. Also, R^2 of the proposed model is the largest, which indicates that it has the highest accuracy. The bar charts of RMSE and MAE for the six models are shown in figure 9(a), and it can be seen that the proposed model has the smallest RMSE value and MAE value. Figure 9(b) shows the bar chart of the R^2 values of the six models and it can be seen that the proposed model has the largest R^2 value. In addition, in order to show the prediction effect of each model more visually, the prediction error boxplots of the six models are given in figure 10, in each boxplot, the red marker line in the center of the box indicates the median of the error dataset, and the top and bottom edges of the box are the upper and lower quartiles of the dataset, respectively. The points outside the maximum and minimum values represent outliers. Boxplots can reflect the statistical properties of the data set. The error boxplots show that SAE, SIAE, VW-SAE and TSAE have wider error boxes and the median lines are all deviating from the zero value, which mean that the overall prediction errors of the compared models are large. Meanwhile

Table 2. Performance index of five models for predicting penicillin concentration.

Model	RMSE	MAE	R^2
SAE	0.0143	0.0095	0.9979
SIAE	0.0122	0.0085	0.9985
GSTAE	0.0109	0.0079	0.9989
VW-SAE	0.0101	0.0070	0.9990
TSAE	0.0095	0.0068	0.9991
Proposed-model	0.0081	0.0064	0.9993



(a) (b)

Figure 9. (a) RMSE and MAE bar charts for six models (b) R^2 bar charts for six models.

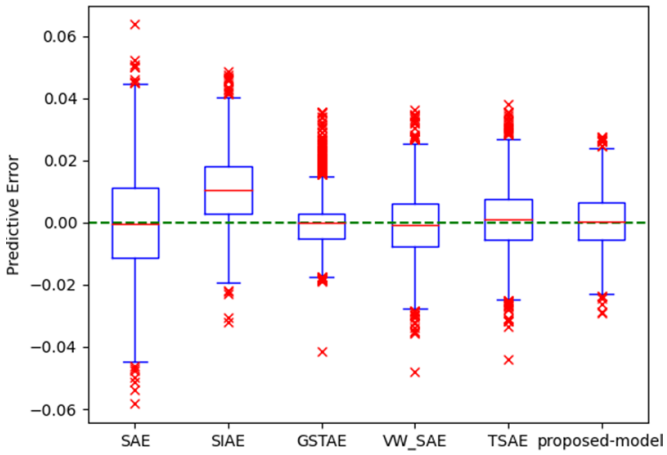


Figure 10. Boxplots of prediction error of penicillin concentration for six models.

the residuals of SIAE show a positive bias of overestimation. There are more outliers in each model. The GSTAE model has a smaller error box with the most outlier points and more high outliers than low outliers, which indicates that the model prediction errors are unevenly distributed and biased towards positive bias. Whereas, the proposed model has a narrower error box, its median line is located at the zero value, and there are fewer outliers, which indicates that the proposed model has the smallest prediction error and is evenly distributed around the zero value. This further indicates that the method proposed in this paper has good robustness.

To show the prediction effect of each model more intuitively, the predicted values of all models are subjected to the inverse normalization operation. The prediction results of each model are shown in figure 11. The prediction effect of the SAE model is shown in figure 11(a), it can be seen that the SAE model has a large difference between the predicted values and the true values throughout the process, which is because the reconstruction error is accumulated layer by layer during the training process. Figure 11(b) shows the prediction results of SIAE, it can be seen that its prediction effect is better than that of SAE model. This is because the SIAE model reconstructs the raw data in the output layer and preserves the structural features of the original data during the training process. The predictive effect of the GSTAE model on penicillin concentration is shown in figure 11(c), and it can be seen that there is a large error between the true and predicted values in the second half of each batch of reactions. The prediction results of the VW-SAE model are shown in figure 11(d) it can be seen that the predicted values of the two back-end batches differ from the actual values, which is due to the fact that the VW-SAE model constructs the quality-related features layer

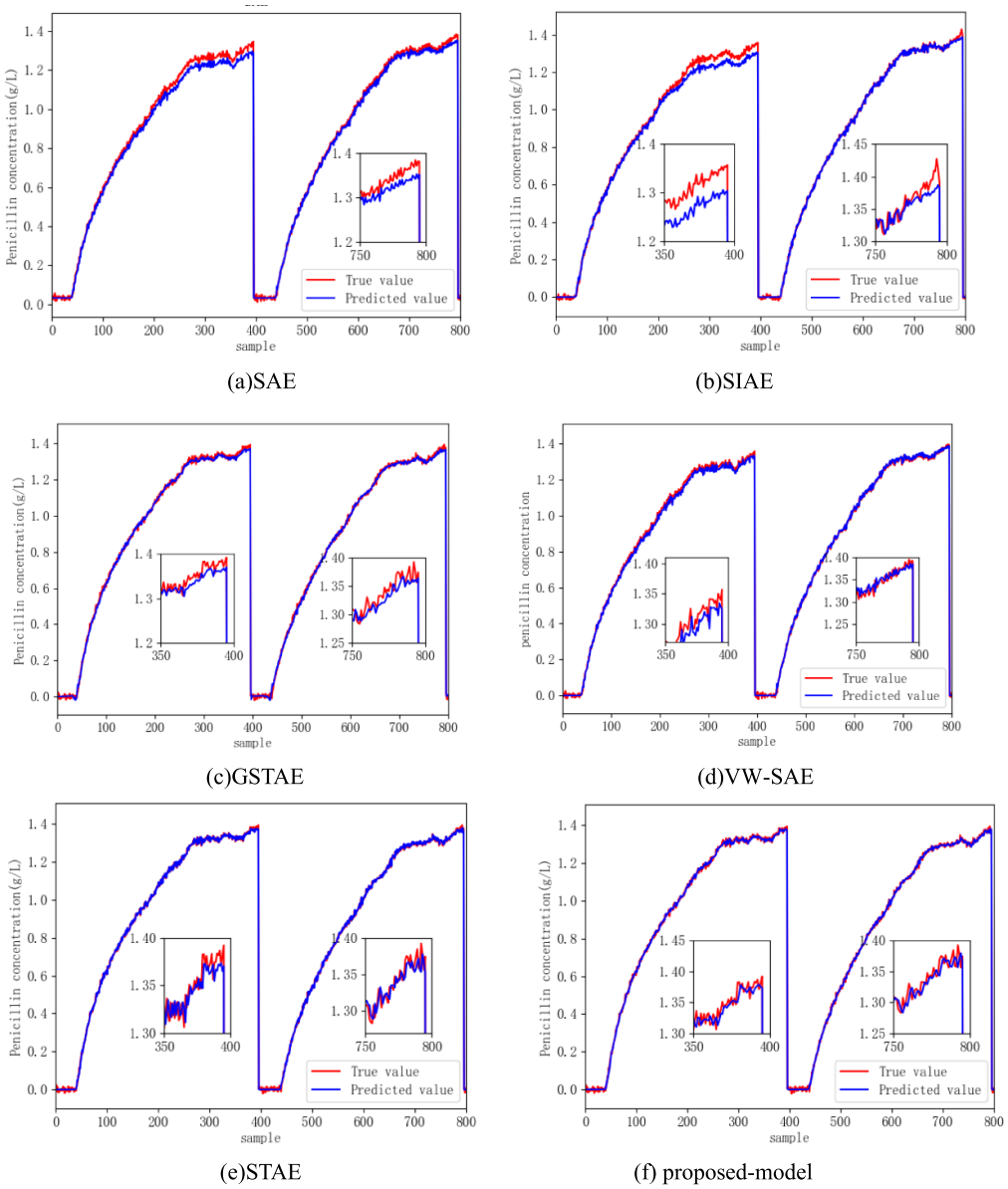


Figure 11. Predicted results of penicillin concentration by (a) SAE, (b) SIAE, (c) GSTAE, (d) VW-SAE, (e) TSAE, and (f) proposed-model.

by layer and ignores the deeper representation of the input features. Figure 11(e) shows the prediction effect of TSAE, it can be seen that the TSAE model has a small error between the true and predicted values at the last moment of each batch of responses. This is due to the fact that TSAE considers the correlation with the target variable only after extracting features layer by layer. The prediction effect of the proposed model in this paper is shown in figure 11(f), and it can be seen that the prediction effect is better than other models, this is because the proposed model not only considers the problem of isomorphization of the raw data structure of each layer of the network, but also considers the correlation between the representations and outputs of each layer.

Figure 12 is the scatter plot of the results of the six models for the prediction of penicillin concentration, the horizontal axis represents the true values and the vertical axis represents the predicted values. Both true and predicted values are normalized. The red solid line represents the reference line, and the closer the blue point is to the reference line, the better the predictive performance of the model.

4.1.2. Biomass concentration prediction. In order to further verify the validity and generalization performance of the proposed model, six models were used in this section to predict the biomass concentration during penicillin fermentation. Figure 13 shows the loss curves for the fine-tuning stage of each model, it can be seen that the initial loss value of the fine-tuning process of the proposed model is 0.004 at the end of the pre-training, and after 150 epochs of training, the final loss value of the proposed model is smaller than that of the other models and tends to be stable.

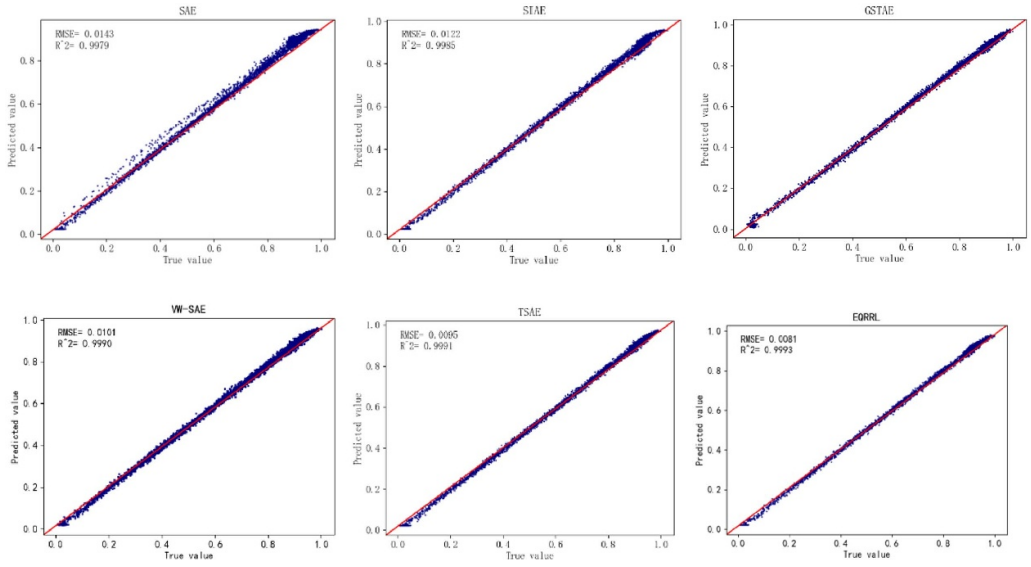


Figure 12. Scatter plot of predicted penicillin concentrations.

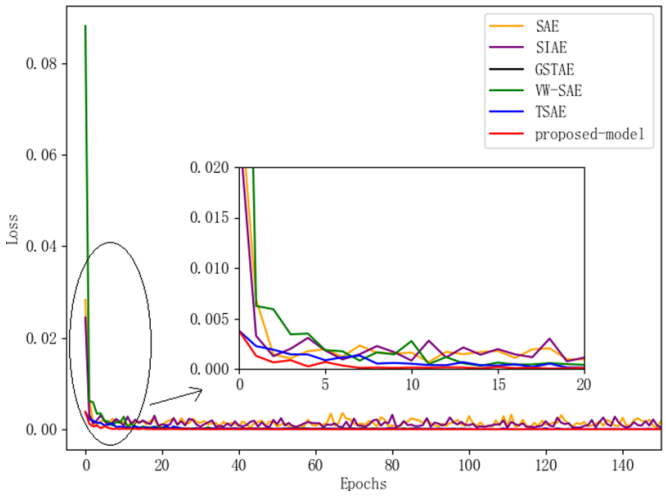


Figure 13. Loss curves of the training process.

Table 3. Performance index of six models for predicting biomass concentration.

Model	RMSE	MAE	R^2
SAE	0.0581	0.0410	0.9538
SIAE	0.0492	0.0313	0.9641
GSTAE	0.0162	0.0116	0.9960
VW-SAE	0.0088	0.0060	0.9988
TSAE	0.0067	0.0051	0.9993
Proposed-model	0.0049	0.0039	0.9996

Table 3 is the values of RMSE, MAE and R^2 of the six models on the testing dataset, it can be seen that the propose model has smaller RMSE and MAE than the other models, while R^2 reaches 0.9996, which indicate that the proposed model has a better accuracy for the prediction of biomass concentration.

The prediction error boxplots of six models are shown in figure 14, which show that the proposed model has a better error distribution than other models. The effectiveness of the six models in predicting the concentration of biomass respectively is shown in figure 15, it can be seen that SAE and SIAE have great fluctuations in the prediction process. Figure 15(c)

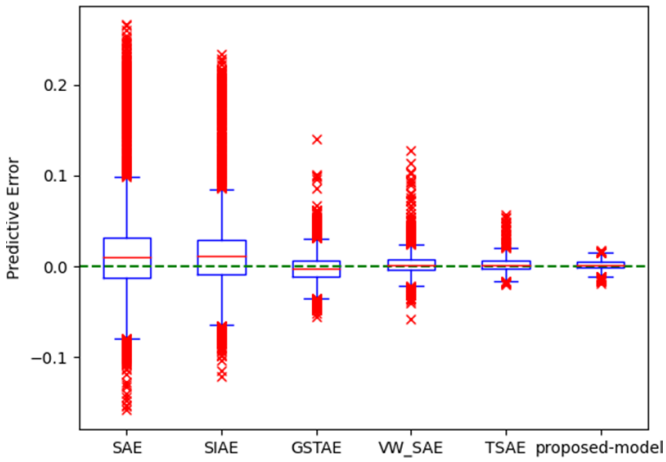


Figure 14. Boxplots of prediction error of biomass concentration for six models.

shows the prediction effect of the GSTAE model on the concentration of biomass, and it can be seen that the error between the true and predicted values is large at the beginning of the first and second batch of reactions. The predicted results of VW-SAE are shown in figure 15(d), and it can be seen that during the first batch of reactions, the actual values differ significantly from the predicted values. The predicted results of TSAE are given in figure 15(e), and it can be seen that there is some errors between the actual and predicted values during the first batch of responses. Figure 15(f) shows the prediction effect of the proposed model on the concentration of the biomass, and it can be seen that the prediction effect is better than that of the other models by considering the correlation between the representation and the output of each layer based on isomorphizing the input variables of each layer.

Figure 16 is the scatter plots of the predicted values of biomass concentration for each model, it can be seen that the quality prediction of SAE and SIAE has large deviations throughout the batch. GSTAE, VW-SAE and TSAE have large deviations at the beginning part. Compared with the above models, the proposed model has a smaller deviation, which illustrates its effectiveness and generalization ability in batch process quality prediction.

4.2. The hot strip mill process (HSMP)

In order to further verify the effectiveness and generalization of the proposed model in practical industrial applications, the strip thickness of the hot strip rolling process is used as the target for validation. As an extremely complex process in the steelmaking industry, strip hot strip rolling is a rolling production project that requires high investment, high quality and generates high efficiency. The process usually consists of a heating furnace, roughing mill, intermediate delay rolls, flying shears, finishing mill units, hot output rolls and laminar flow cooling and coiling machines [32]. Its basic arrangement is shown in figure 17. Firstly, the heating furnace ensures that the slabs are heated to about 1200 °C and fed into the roughing mill. In the roughing mill, slabs with a thickness of 100–200 mm are rolled in a number of passes into intermediate billets with a thickness of 28–45 mm, which are output at the temperature of 1050 °C. Then, the intermediate delay roller conveyor would quickly transport the intermediate billet through the flying shear to be sheared into the finishing mill unit for seven racks of continuous rolling. The width, thickness, plate shape and final rolling temperature of the strip meet the requirements of continuous rolling through the finishing unit. Subsequently, the organizational properties of the steel are further improved by laminar flow cooling, and the steel is finally wound into coils by an underground coiler. The thickness of the strip is a key quality indicator of the final product, and depending on the customer's requirements, the strip hot rolling process can produce strips with export thicknesses ranging from 1.5 mm to 12.7 mm [33].

In this paper, the 1700 mm strip hot rolling line of a steel company was used as the background of the study, and field data on the production of finished 2.70 mm thick strips were collected for experimental validation. The specific descriptions of process variables and quality variables are shown in table 4. There are 20 process variables, including the roll gap, rolling force, and bending roll force of the seven stands of the finishing mill, in which the first stand is not equipped with bending rolls. The quality variable is the exit thickness of the finishing mill. The sampling time is 30 s and the sampling period is 10 ms. The total number of samples is 3000.

The SAE, SIAE, GSTAE, VW-SAE, TSAE, and proposed-models are used to predict strip thickness, respectively. The loss curves of each model fine-tuning stage are shown in figure 18. From the figure, it can be seen that at the end of pre-training, the initial loss value of the fine-tuning process of the model proposed in this paper is 0.0052, which is smaller than that of the other models, and it tends to stabilize after 120 training sessions. This indicates that the model has good convergence.

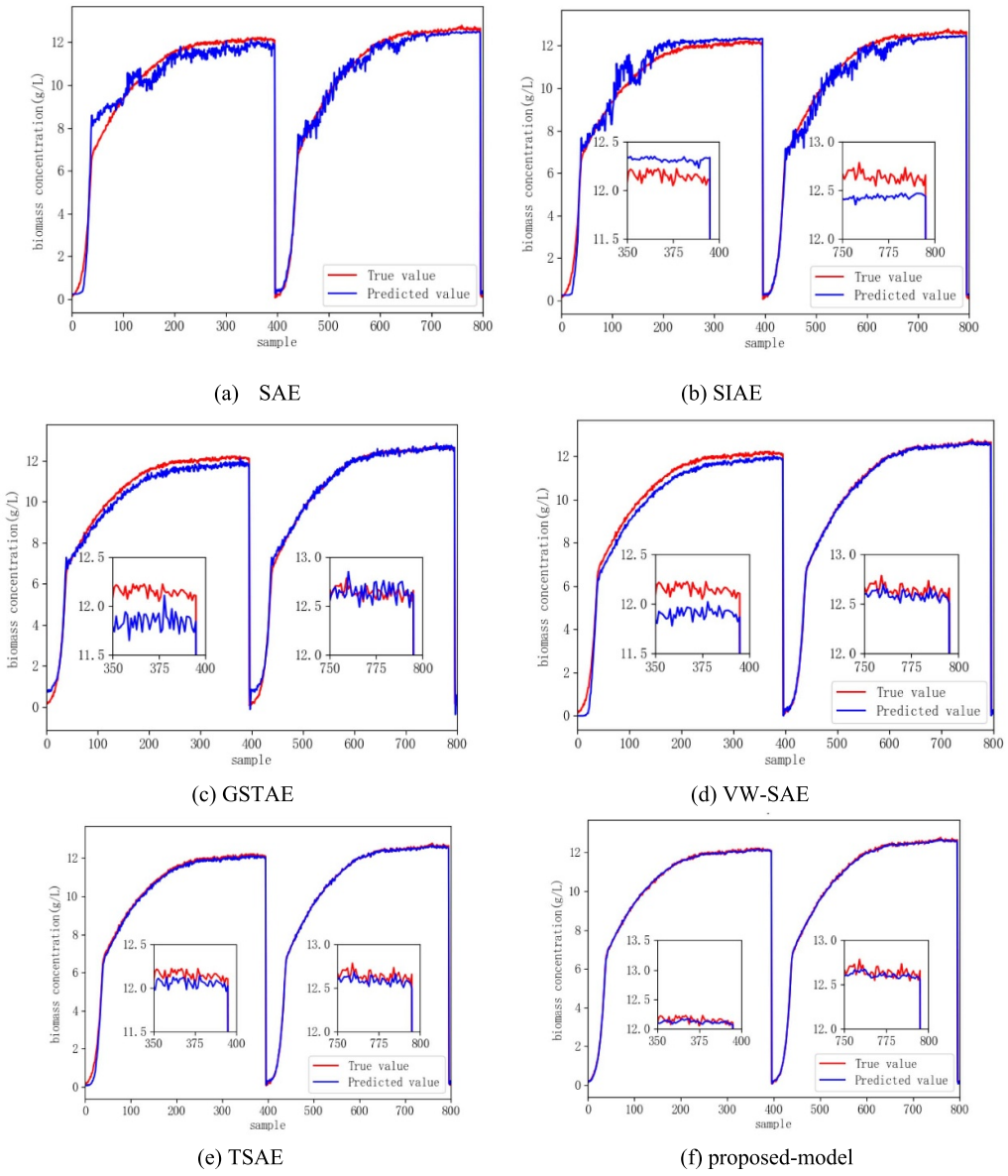


Figure 15. Predicted results of biomass concentration by (a) SAE, (b) SIAE, (c) GSTAE, (d) VW-SAE, (e) TSAE, and (f) proposed-model.

Considering the generalization performance of the models, the test set is divided equally into four sub-test sets. The average RMSE, MAE and R^2 values predicted for each of the four sub-test sets individually are shown in table 5. From the figure, it can be seen that in the prediction of strip thickness, the proposed model has the smallest RMSE and MAE values and the R^2 is 0.9413.

The prediction error boxplots of six models are shown in figure 19. From the figure, it can be seen that the error box of the proposed model in this paper is narrower and the median is located at zero value compared to the other models, which indicates that the model has the least prediction error and has better robustness.

The fitted curves of the six models for predicting the thickness of the strip are shown in figure 18, where figures 20(a) and (b) are the prediction effects of SAE and SIAE, respectively, it can be seen that in the late stage of prediction, there is a large error between the real value and the predicted value, and the models do not get convergence. The predicted results of GSTAE are shown in figure 20(c), it can be seen that there is a large error between the true and predicted values throughout the prediction process. Figure 20(d) shows the prediction effect of the VW-SAE model, and it can be seen that there is a difference between the actual value and the predicted value at the later stage of the prediction. The prediction results of TSAE are shown in figure 20(e), and it can be seen that there is an error between the actual and predicted values throughout the prediction process. However, the overall prediction results are better than the other four models. Figure 20(f) shows the fitting curve of the proposed model for strip thickness

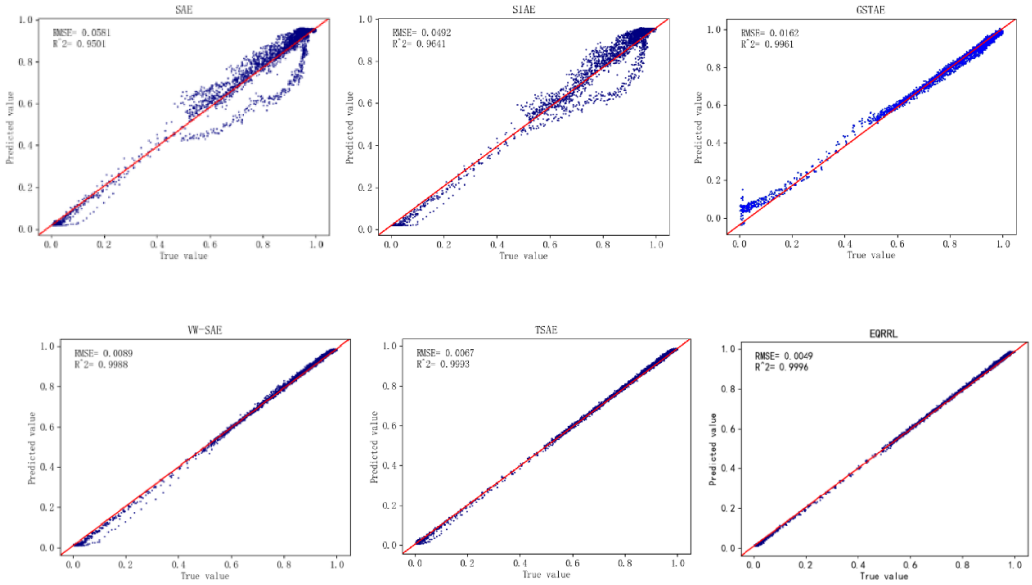


Figure 16. Scatterplot of predicted biomass concentration.

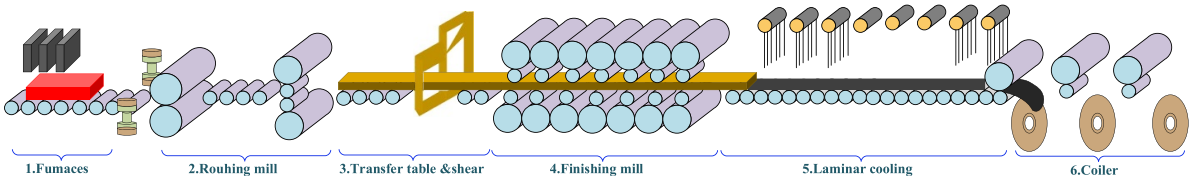


Figure 17. Schematic layout of the hot strip mill.

Table 4. Process and quality variables in finishing mill.

Variable	Type	Description	Unit
1–7	Measured	Average roll gap	mm
8–14	Measured	Rolling force	MN
15–20	Measured	Roll bending force	MN
21	Quality	Finishing mill exit strip thickness	mm

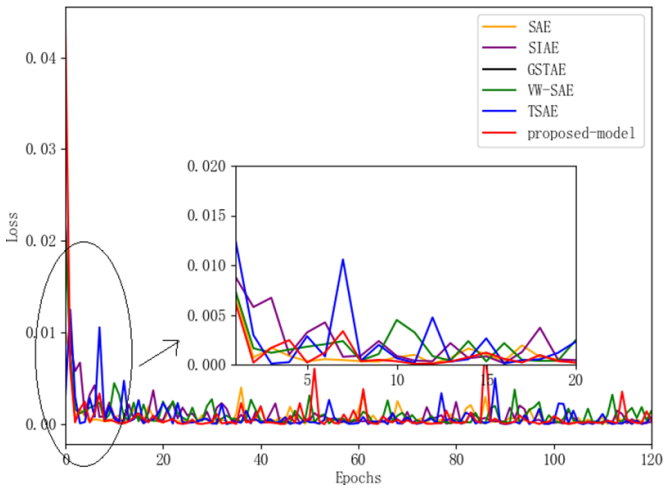


Figure 18. Loss curves of the training process.

Table 5. Performance index of six models for predicting strip thickness.

Model	RMSE	MAE	R^2
SAE	0.0776	0.0471	0.3503
SIAE	0.0924	0.0631	0.0785
GSTAE	0.0390	0.0316	0.8360
VW-SAE	0.0374	0.0255	0.8486
TSAE	0.0333	0.0231	0.8807
Proposed-model	0.0233	0.0119	0.9413

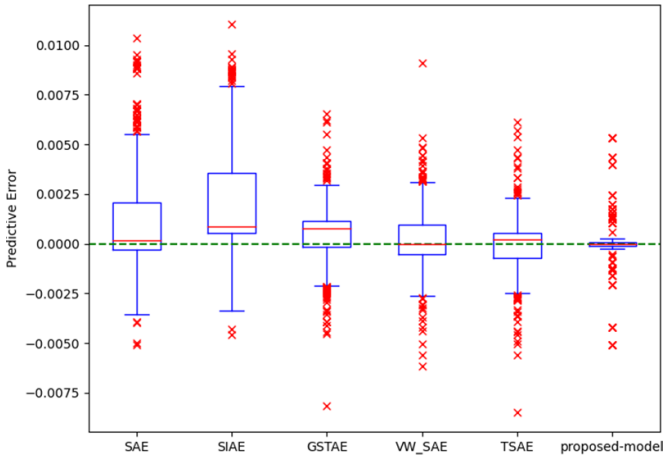


Figure 19. Boxplots of prediction error of biomass concentration for six models.

prediction, it can be seen that the difference between the actual value and the predicted value during the whole process of prediction is small, which means the proposed model has a better performance of quality prediction than the other five models.

The scatter plot of strip thickness prediction of each model is shown in figure 21, it can be seen that the predicted values of the proposed model are basically close to the reference red line, while most of the predicted values of the other models are far away from the reference line, i.e. they have larger deviations from the true values. It can be seen that the proposed model has more superior quality prediction capability.

5. Conclusion

This paper proposes a deep quality-related stacked isomorphic autoencoder for batch process quality prediction. First, the three-dimensional data of batch processes are expanded and normalized along the variable direction. Then, the raw input data are reconstructed layer-by-layer by using isomorphic autoencoders. Correlation calculations are performed layer-by-layer between numerous input variables and target variables through MIC-GJC to retain quality-relevant variables. The correlation error function is constructed to optimize the network structure layer-by-layer. Deep quality-related

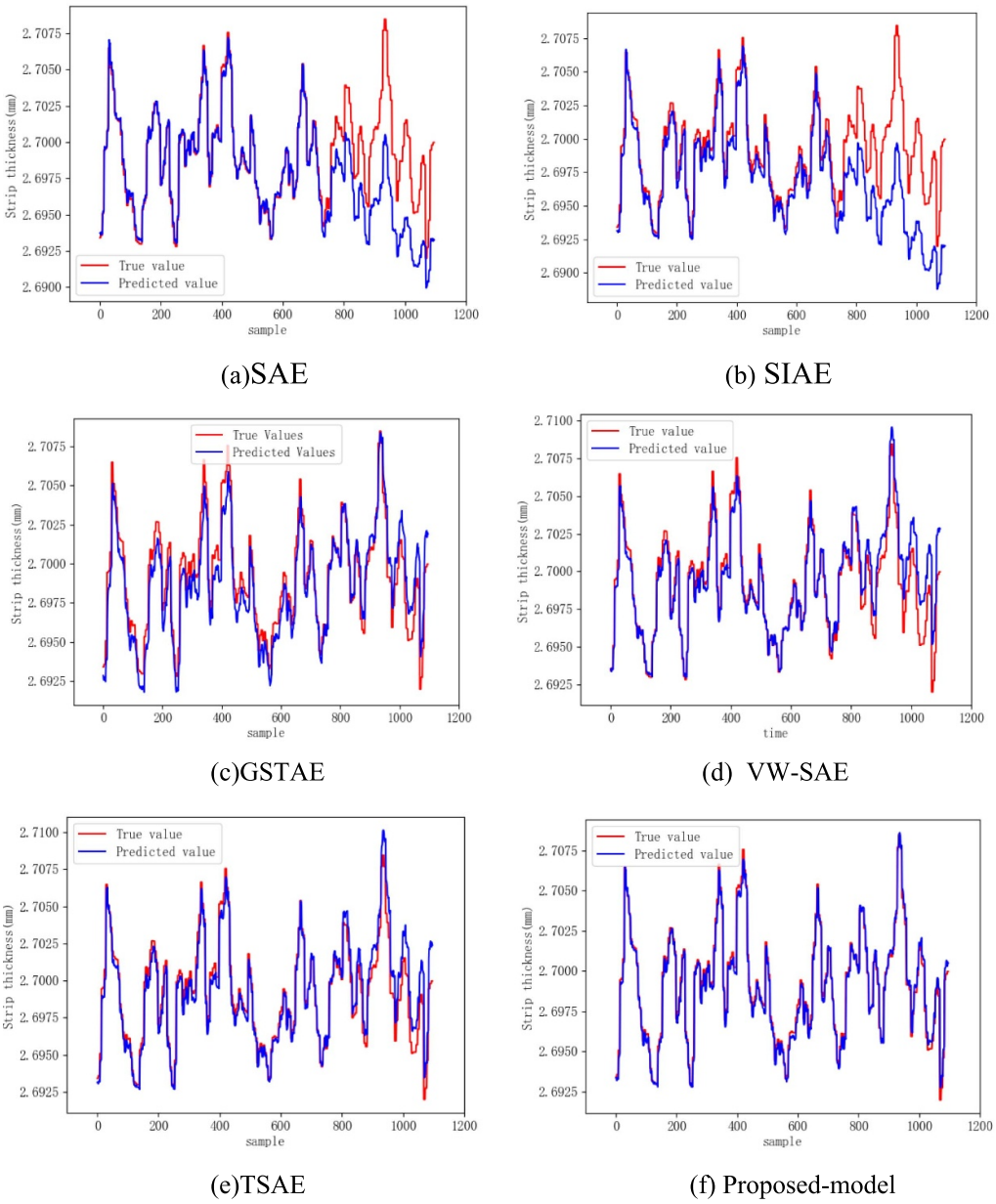


Figure 20. Predicted results of strip thickness by (a) SAE, (b) SIAE, (c) GSTAE, (d) VW-SAE, (e) TSAE, and (f) proposed-model.

features that facilitate quality modeling are extracted and fed into the regression network to obtain quality predictions. Finally, SAE, SIAE, GSTAE, VW-SAE, STAE and the proposed-model are applied to the penicillin fermentation process and the hot strip mill process, the experimental results show the proposed model has better quality prediction performance. Since the proposed quality prediction model needs to reconstruct the raw inputs of each layer in the pre-training phase, while extracting the quality related variables. It is considered that the pre-training time may be increased when dealing with very high dimensional raw datasets. Therefore, in the next research, it needs to find ways to optimize the model parameters and reduce the model training time.

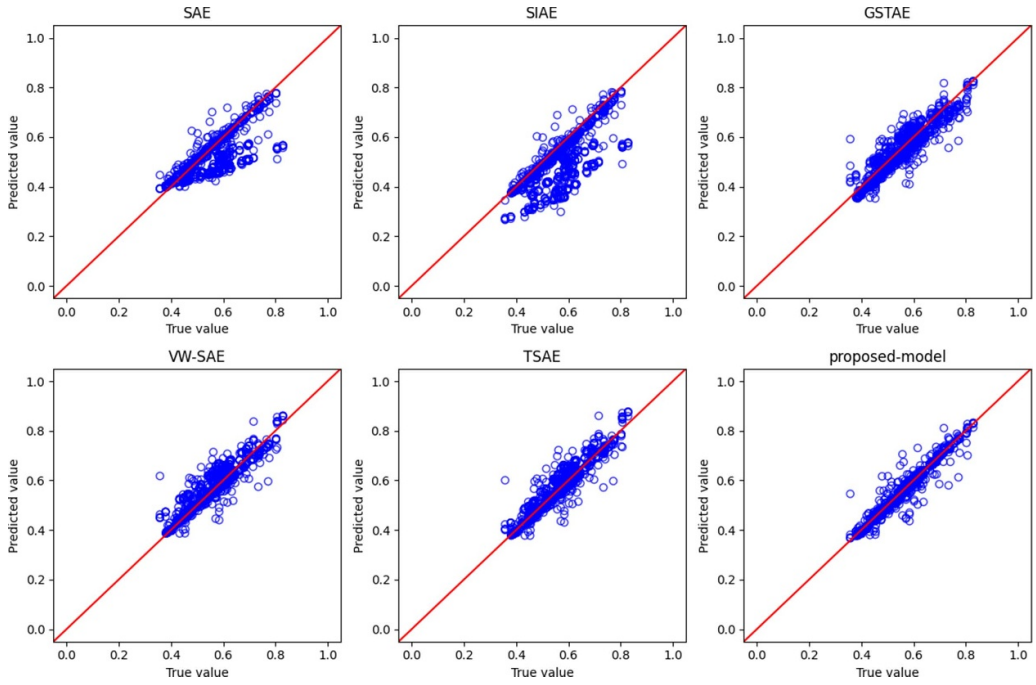


Figure 21. Scatter plots of predicted strip thickness by SAE, SIAE, GSTAE, VW-SAE, TSAE, and proposed-model.

Data availability statement

The data that support the findings of this study are available at www.chbe.iit.edu/∼cinar/. These data were derived from the following resources available in the public domain: 10.1016/S0098-1354(02)00127-8 [30].

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62263021), the Science and Technology Project of Gansu Province (21JR7RA206), Key Project of Natural Science Foundation of Gansu Province (24JRRA173), and the College Industrial Support Project of Gansu Province (2023CYZC-24). Gansu Province University young doctor support project (2024QB-037); Lanzhou youth science and technology talent innovation project (2023-QN-36).

Author contributions

Yan Zhang: Software; validation; writing-original draft. Jie Cao: Supervision; writing-review and editing. Xiaoqiang Zhao: Conceptualization; methodology. Yongyong Hui: Data curation; investigation.

ORCID iDs

Yan Zhang https://orcid.org/0000-0002-3577-6752
Xiaoqiang Zhao https://orcid.org/0000-0001-5687-942X
Yongyong Hui https://orcid.org/0000-0001-5014-887X

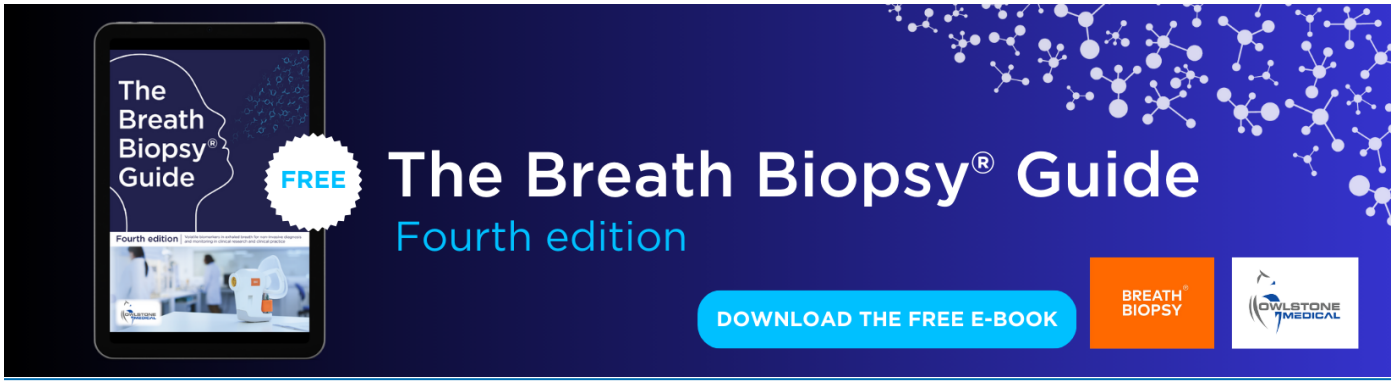
References

[1] Wei C, Shao W and Song Z 2020 Virtual sensor development for multioutput nonlinear processes based on bilinear neighborhood preserving regression model with localized construction *IEEE Trans. Ind. Inform.* **17** 2500–10
[2] Li H, Wang S, Shi H, Su C and Li P 2023 Two-dimensional iterative learning robust asynchronous switching predictive control for multiphase batch processes with time-varying delays *IEEE Trans. Syst. Man Cybern. Syst.* **53** 6488–502
[3] Yao H, Zhao X, Li W and Hui Y 2023 Multi-stage fusion regression network for quality prediction of batch process *Can. J. Chem. Eng.* **101** 6977–94
[4] Yuan X, Li L, Wang Y, Yang C and Gui W 2020 Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network *Can. J. Chem. Eng.* **98** 1377–89
[5] Ren J-C, Liu D and Wan Y 2022 VMD-SEAE-TL-Based Data-Driven soft sensor modeling for a complex industrial batch processes *Measurement* **198** 111439
[6] Kadlec P, Gabrys B and Strandt S 2009 Data-driven soft sensors in the process industry *Comput. Chem. Eng.* **33** 795–814
[7] Qiu K, Wang J, Wang R, Guo Y and Zhao L 2021 Soft sensor development based on kernel dynamic time warping and a relevant vector machine for unequal-length batch processes *Expert Syst. Appl.* **182** 115223
[8] Keithley R B, Wightman R M and Heien M L 2009 Multivariate concentration determination using principal component regression with residual analysis *TRAC Trends Anal. Chem.* **28** 1127–36
[9] Facco P, Doplicher F, Bezzo F and Barolo M 2009 Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process *J. Process. Control* **19** 520–9
[10] Yuan X, Ge Z, Huang B, Song Z and Wang Y 2016 Semisupervised JITL framework for nonlinear industrial soft sensing based on locally semisupervised weighted PCR *IEEE Trans. Ind. Inform.* **13** 532–41

[11] Hazama K and Kano M 2015 Covariance-based locally weighted partial least squares for high-performance adaptive modeling *Chemometr. Intell. Lab.* **146** 55–62
[12] Nomikos P and MacGregor J F 1995 Multi-way partial least squares in monitoring batch processes *Chemometr. Intell. Lab.* **30** 97–108
[13] Zhang Y, Teng Y and Zhang Y 2010 Complex process quality prediction using modified kernel partial least squares *Chem. Eng. Sci.* **65** 2153–8
[14] Lughofer E, Zavoianu A-C, Pollak R, Pratama M, Meyer-Heye P, Zörrer H, Eitzinger C and Radauer T 2019 Autonomous supervision and optimization of product quality in a multi-stage manufacturing process based on self-adaptive prediction models *J. Process. Control* **76** 27–45
[15] Yuan X, Ge Z and Song Z 2014 Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes *Ind. Eng. Chem. Res.* **53** 13736–49
[16] Yan W, Shao H and Wang X 2004 Soft sensing modeling based on support vector machine and Bayesian model selection *Comput. Chem. Eng.* **28** 1489–98
[17] Zhao C 2013 A quality-relevant sequential phase partition approach for regression modeling and quality prediction analysis in manufacturing processes *IEEE Trans. Autom. Sci. Eng.* **11** 983–91
[18] Zheng W, Liu Y, Gao Z and Yang J 2018 Just-in-time semi-supervised soft sensor for quality prediction in industrial rubber mixers *Chemometr. Intell. Lab.* **180** 36–41
[19] Yuan X, Ou C, Wang Y, Yang C and Gui W 2020 A novel semi-supervised pre-training strategy for deep networks and its application for quality variable prediction in industrial processes *Chem. Eng. Sci.* **217** 115509
[20] Yan W, Tang D and Lin Y 2016 A data-driven soft sensor modeling method based on deep learning and its application *IEEE Trans. Ind. Electron.* **64** 4237–45
[21] Liu C, Wang Y, Wang K and Yuan X 2021 Deep learning with nonlocal and local structure preserving stacked autoencoder for soft sensor in industrial processes *Eng. Appl. Artif. Intell.* **104** 104341
[22] Yuan X, Wang Y, Yang C and Gui W 2020 Stacked isomorphic autoencoder based soft analyzer and its application to sulfur recovery unit *Inf. Sci.* **534** 72–84
[23] Yuan X, Huang B, Wang Y, Yang C and Gui W 2018 Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE *IEEE Trans. Ind. Inform.* **14** 3235–43
[24] Yuan X, Zhou J, Huang B, Wang Y, Yang C and Gui W 2019 Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy *IEEE Trans. Ind. Inform.* **16** 3721–30
[25] Jiang Q, Wang Z, Yan S and Cao Z 2022 Data-driven soft sensing for batch processes using neural network-based deep quality-relevant representation learning *IEEE Trans. Artif. Intell.* **4** 602–11
[26] Wang K, Gopaluni R B, Chen J and Song Z 2018 Deep learning of complex batch process data and its application on quality prediction *IEEE Trans. Ind. Inform.* **16** 7233–42
[27] Ji C, Ma F, Wang J and Sun W 2023 Profitability related industrial-scale batch processes monitoring via deep learning based soft sensor development *Comput. Chem. Eng.* **170** 108125
[28] Li Z, Zheng X, Chen G, Wei Y and Lu K 2023 A matching pursuit algorithm for sparse signal reconstruction based on Jaccard coefficient and backtracking *Circuit Syst. Signal Proc.* **42** 6210–27
[29] Zhang C, Peng K and Dong J 2019 A novel plant-wide process monitoring framework based on distributed Gap-SVDD with adaptive radius *Neurocomputing* **350** 1–12
[30] Birol G, Ündey C and Cinar A 2002 A modular simulation package for fed-batch fermentation: penicillin production *Comput. Chem. Eng.* **26** 1553–65
[31] Sun Q and Ge Z 2020 Gated stacked target-related autoencoder: a novel deep feature extraction and layerwise ensemble method for industrial soft sensor application *IEEE Trans. Cybern.* **52** 3457–68
[32] Ma L, Dong J, Hu C and Peng K 2021 A novel decentralized detection framework for quality-related faults in manufacturing industrial processes *Neurocomputing* **428** 30–41
[33] Ma L, Dong J, Peng K and Zhang K 2017 A novel data-based quality-related fault diagnosis scheme for fault detection and root cause diagnosis with application to hot strip mill process *Control Eng. Pract.* **67** 43–51

PAPER

Batch process monitoring based on sequential phase division multiway sparse weighted neighborhood preserving embedding



IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **35** (2024) 035704 (17pp) https://doi.org/10.1088/1361-6501/ad1154

Batch process monitoring based on sequential phase division multiway sparse weighted neighborhood preserving embedding

Yan Zhang[1,2], **Xiaoqiang Zhao**[1,2,3,*], **Jie Cao**[1,4] **and Yongyong Hui**[1,2,3]

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China
[2] Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou, People's Republic of China
[3] National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China
[4] Manufacturing Informatization Engineering Research Center of Gansu Province, Lanzhou, People's Republic of China

E-mail: xqzhao@lut.edu.cn

Received 27 June 2023, revised 5 November 2023
Accepted for publication 30 November 2023
Published 11 December 2023

Abstract

Batch processes are often characterized by multiphase and different batch durations, which vary from phase to phase presenting multiple local neighborhood features. In this paper, a sequential phase division-multiway sparse weighted neighborhood preserving embedding method is proposed for monitoring batch processes more sensitively. First, batches with uneven durations are synchronized, and the phases are automatically determined in chronological order. Secondly, the nearest neighbors are computed at each phase and the optimal sparse representation (SR) is obtained based on the nearest neighbors. This improves the robustness of the algorithm to noise and outliers, and solves the problem of computational difficulties associated with global SR based. Thirdly, the distance values of the neighbor elements are considered to fully extract the neighbor structure when the optimal SR is calculated. Finally, after dimension reduction, T^2 and squared prediction error statistics are established in feature space and residual space respectively for fault detection. The effectiveness of the method is verified by a multiphase numerical simulation example and the penicillin fermentation process.

Keywords: process monitoring, multiphase, neighborhood preserving embedding, sparse representation, distance weighted

1. Introduction

With the rapid development of modern detection and sensing technology, a large amount of data reflecting the production status can be stored. How to effectively use these data and extract valuable information has become a research hotspot in process monitoring [1]. Data-driven fault detection and diagnosis technology has inherent advantages, so it is becoming more and more popular in the field of process monitoring [2]. As an important process monitoring method, multivariate statistical methods transform process data as process operation knowledge and use historical data to monitor the process online, which play an

* Author to whom any correspondence should be addressed.

important role for ensuring process safety and product quality [3, 4].

As typical multivariate statistical methods, multiway principal component analysis (MPCA) and multiway partial least squares [5, 6] have been widely used in process monitoring. Batch process plays an important role in modern industrial production, which has the nonlinear, dynamic, and other characteristics, some improved methods are proposed to improve the effectiveness of process monitoring [7, 8]. As most nonlinear dynamic monitoring methods cannot extract the potential driving forces of slow changes in nonlinear batch processes, Zhang *et al* [9] proposed a monitoring scheme based on two-directional dynamic kernel slow feature analysis to realize the global modeling of batch-wise dynamics. Independent component analysis and related improvement methods are used to solve the non-Gaussian feature of the process [10, 11]. The actual industrial process does not completely satisfy a Gaussian or non-Gaussian distribution but usually satisfies a mixture of Gaussian distribution and non-Gaussian distribution. The literature [4] solved the mixed distribution problem by establishing a twin space; however, it ignored the local structure that was critical to the representation of process information. The manifold learning algorithm is a local information modeling method, which maintains the original manifold structure of the data while extracting the features of process data, and reflects the process information more accurately [12–15].

Multiphase and uneven batch duration are the typical characteristics of batch process in practice, and different phases may exhibit diverse underlying behaviors. MPCA and related methods treat the whole process as a single object which cannot reveal the multiplicity of information. The variable correlation characteristics and trajectories vary from one phase to another; therefore, it is necessary to establish multiphase partition methods for batch process to develop phase-based models [16, 17]. Zhang *et al* [18] proposed a new index called the phase recognition factor to divide the phase automatically. Zhu *et al* [17] proposed a multiphase batch process method based on a 2D time-slice dynamic system which characterized the batch-wise and variable-wise dynamics simultaneously. Chunhao *et al* [19] proposed an affinity propagation phase partition method to divide the whole process into several steady and transition phases. However, most of these multiphase methods assume that each batch has the same length and that critical events occur at the same time. Furthermore, the local neighbor features in each phase cannot be adequately characterized. Recent directions in data-driven methods have focused on applying neural networks in fault detection and diagnosis tasks. However, the unequal length of batch process has rarely been studied [20, 21].

As a typical manifold learning algorithm, neighborhood preserving embedding (NPE) [22] has been widely used in process monitoring [23]. Compared to other manifold learning algorithms such as locally linear embedding and Laplacian Eigenmaps (LE), NPE has several advantages: (1) NPE can explicitly preserve the local neighborhood relationships of data points. Its purpose is to maintain the pairwise distances between neighboring points in high-dimensional space and to ensure that nearby points remain close to each other in low-dimensional embedding. This makes NPE particularly effective in capturing the local structure of data. (2) NPE is computationally efficient, which makes it suitable for large-scale datasets. It avoids the computationally expensive eigendecomposition required by the algorithm like LE, and it is more scalable and applicable to real scenarios. (3) The low-dimensional representation obtained by NPE retains the local neighborhood relationships of the original data, making it easier to understand and analyze the data structure.

When the NPE algorithm constructs a neighborhood graph based on the Euclidean distance in each phase, it is susceptible to process outliers and noise, resulting in a distortion of the potential manifold structure of process data, which cannot effectively reflect the features of the process data. Sparse representation (SR) is used for data classification and image processing due to its feature selection and task simplification [24, 25], which solves the optimal SR to remove the variables that have no contribution or small contribution to the output. For batch process, the number of process variables is numerous, and how to realize the feature selection is very important for process monitoring [26]. SR can obtain the sparse coefficients by solving the L1 norm; therefore, the obtained sparse coefficients can be used to construct the neighborhood graph, and the variables that can reflect the feature information of the process are selected to remove the influence of outliers and noise. Compared with the Euclidean distance neighborhood graph, the neighborhood graph constructed by SR can better reflect the spatial structure of the process data. However, SR is a global optimization method, when the data dimension is large, the calculation amount is very large, which causes the calculation speed to be slower or even exceed the calculation memory. It also ignores the local characteristics of process data when the process data are globally optimized, and cannot fully characterize the process characteristics.

Therefore, a batch process monitoring method for sequential phase division multiway sparse weighted NPE (SPD-MSWNPE) is proposed in this paper. The relaxed greedy time warping (RGTW) is used to accomplish online synchronization of the entire batches. After synchronization, the local neighbor features are synchronized in different batches, and the phases are recognized based on local neighbor features in the time sequence automatically, which are defined to reflect the local feature variation among the whole batch process. For the divided phases, SR is used to achieve the local optimal SR after finding the local neighbors, which can avoid the problem of increasing calculation amount caused by the global optimization. When the optimal SR is calculated, the distance between the data points in the neighborhood is introduced as the weight value, which can be used to construct an enhanced local sparse structure. Based on the proposed method, T^2 and squared prediction error (SPE) statistics are established to realize fault detection. The contributions of the proposed method are as follows:

(1) A phase division model with uneven batches in a time sequence is established.

(2) The optimal SR and distance weight values are obtained based on the nearest neighbors to make it more robust to noise and outliers and avoid the computational difficulty caused by global SR.
(3) The distance values of the neighbor elements are calculated to construct an enhanced local sparse structure.

The remaining parts of this paper are arranged as follows. In section 2, NPE algorithm and SR are briefly introduced. The proposed method is described in detail in section 3. In section 4, the main process and steps of fault detection with the proposed method are described. The proposed model is validated on a multiphase nonlinear numerical example and penicillin fermentation process in section 5. The conclusion of this paper is given in section 6.

2. Preliminaries

2.1. Neighborhood preserving embedding (NPE)

In NPE algorithm, the neighbor point reconstruction is realized by the linear combination of the neighbor points, and this reconstruction relationship is maintained during the dimension reduction process. For the process data $X=\{x_1,x_2,\ldots,x_n\}\subseteq R^{m\times n}$, NPE achieves dimension reduction by solving the projection matrix $A\in R^{m\times p}(p<m)$. The specific solution process is as follows:

k nearest neighbors of a data point are sought. If two points are within k nearest neighbors, the two points are connected by an edge. Suppose that the weight matrix is W, if i and j have an edge connected, the weight is W_{ij}, otherwise, the weight is 0. It can be obtained by equation (1) ($i_1,\ldots,i_k$ are obtained by the neighborhood graph),

$$\begin{aligned}&\min_{W}\sum_{i=1}^{n}\left\|x_i-\sum_{j=i_1}^{i_k}W_{ij}x_j\right\|_2^2\\&s.t.\ \sum_{j=i_1}^{i_k}W_{ij}=1,\quad j=1,2,\ldots,m.\end{aligned}\tag{1}$$

The projection matrix A can be obtained by solving equation (2),

$$\min_{A\in R^{m\times d}}\left\|\sum_{i=1}^{n}A^{\mathrm{T}}x_i-A^{\mathrm{T}}\sum_{j=i_1}^{i_k}W_{ij}x_j\right\|_2^2.\tag{2}$$

By introducing generalized eigenvectors, the column vectors of projection matrix A can be obtained by equation (3):

$$\begin{aligned}&XMX^{\mathrm{T}}a=\lambda XX^{\mathrm{T}}a\\&M=(I-W)^{\mathrm{T}}(I-W)\end{aligned}.\tag{3}$$

The eigenvectors $(a_1,a_2,\ldots,a_p)$ corresponding to the smallest p eigenvalues $(\lambda_1\leqslant\lambda_2\leqslant\cdots\leqslant\lambda_p)$ in equation (3) constitute the projection matrix A.

2.2. Sparse representation (SR)

For a given data set $X=[x_1,x_2,\ldots,x_n]\in R^{m\times n}$, the SR uses other vectors in X to reconstruct x_i. Suppose that the n-dimensional coefficient vector of x_i is $s_i=[s_{i,1},s_{i,2},\ldots,s_{i,i-1},0,s_{i,i+1},\ldots,s_{i,n}]^{\mathrm{T}}$, and $s_{i,j}(i\neq j)$ represents the contribution value of x_j reconstructed to x_i, and the sparse coefficient can be obtained by solving equation (4),

$$\min_{S}\|X-X\cdot S\|_2^2+\lambda\|S\|_0\tag{4}$$

where $\|\|_0$ represents the L_0 norm. The solution of the L_0-norm is a NP problem. Therefore, based on the compressed sensing theory, the L_1-norm is used to replace the L_0-norm to solve the problem, as shown in equation (5),

$$\min_{S}\|X-X\cdot S\|_2^2+\lambda\|S\|_1\tag{5}$$

where $\|\|_1$ is the L_1 norm, which is used to solve the sum of absolute values in the matrix, and λ is the regularization parameter.

3. The proposed method

3.1. Sequential phase division (SPD) with uneven lengths

The uneven length problem in batch process would result in inconsistent batch profiles for different batches and similar events that appear in different time instants. To divide the batch process phases reasonably and make each phase have similar local neighbor characteristics, it needs to synchronize the batch trajectories and make the key events occur at the same time.

Different batches with uneven lengths are collected under normal operation condition. Then, the RGTW method is used to synchronize the ongoing batches [27, 28]. For batches of data with uneven-lengths, a batch close to the middle length of the duration is selected as the reference batch F_r. The Euclidean distance between the row vectors of F_r and new sample $f_{\mathrm{new},t}(1\times J)$ at time t is computed. A set of cumulative weighted distances $D_{t,j}$ at t sampling time is estimated, where $j=\hat{l}_t,\ldots,\hat{u}_t$, $\hat{l}_t$ and $\hat{u}_t$ are the upper and lower boundaries at the t sampling time, and the optimal path at first β sampling time can be derived by following the method of Kassidas *et al* [29].

For the window $\eta_{t-\beta+1,t}$ at the interval $[t-\beta+1,t]$, the optimal path can be defined as equation (6),

$$g_{t-\beta+1,t}^{\mathrm{T}}=\{w(1),\ldots,w(K_{t-\beta+1,t})\}\tag{6}$$

where $K_{t-\beta+1,t}$ is the number of points for the optimal path within the window $\eta_{t-\beta+1,t}$, the alignment would continue by the sliding window within $\eta_{t-\beta+1,t}$.

If the endpoint e_t^* at the current time is located at one of the extremes of the band, the lower and upper boundaries must be

updated according to the position of e^*, which can be calculated by equation (7),

$$u_i = \min\left(u_i + 2, K_r\right) \quad i = t+1, \ldots, \max\left(K_n\right)$$
$$l_i = \begin{cases} l_t & i = \{t+1, t+2\} \\ l_i - \left(l_{t+2} - l_t\right) & i = t+3, \ldots, \max\left(K_n\right) \\ n = 1, \ldots, N \end{cases}. \quad (7)$$

If the ongoing batch exceeds the maximum duration of a normal batch, the boundaries are extended by taking the value of the last boundary belonging to the upper and lower boundaries when measuring each new value. After alignment, the samples need to be synchronized. Depending on whether the algorithm compresses or expands the samples, an update of the last monitored sample or some new values can be obtained. Therefore, the whole batch is accessible and can be monitored in real-time.

After synchronization, the batch data forms a three-dimensional data matrix $X(I \times J \times K)$, where I represents the batch, J represents the number of variables, and K represents the sample time. To depict the multiphase characteristics and highlight the local neighbor features, the whole process is partitioned according to the local neighbor features and the time sequence [30].

The three-dimensional data matrix $X(I \times J \times K)$ is unfolded as $X(I \times JK)$ along the batch direction. Then, the time slice matrix $X(I \times J)_k$ is normalized by equation (8),

$$\bar{X}_k = \frac{X(I \times J)_k - \text{mean}\left(X(I \times J)_k\right)}{\text{std}\left(X(I \times J)_k\right)} \quad (8)$$

where $\text{mean}(X(I \times J)_k)$ and $\text{std}(X(I \times J)_k)$ are the mean and standard deviation of $X(I \times J)_k$.

For the normalized time slice matrix $\bar{X}(I \times J)_k$, NPE is used to extract the neighbor structure features and get the initial time slice model as equation (9),

$$M_k = A_k \bar{X}_k^{\mathrm{T}} \quad (9)$$

where A_k is the projection matrix that can be calculated by equation (2).

The first h time slices are arranged as $\bar{X}_c(Ih \times J)$, NPE is used to calculate the projection matrix $\hat{A}_c$ based on $\bar{X}_c(Ih \times J)$, the phase model is shown as equation (10),

$$M_c = \hat{A}_c \bar{X}_c^{\mathrm{T}}. \quad (10)$$

Then, we can evaluate the similarity between the kth ($k = h+1, h+2, \ldots$) time slice model A_k and the phase mode $\hat{A}_c$, as shown in equation (11),

$$M_{k,c} = \exp\left(-\frac{\left\|A_k - \hat{A}_c\right\|}{2J}\right) \quad (11)$$

where h is the shortest duration time of the phase, J is the number of variables.

When $M_{k,c}$ in d successive time slices starting at time k^* is below the control limit γ, the time slice before k^* can be represented as a phase with the same local neighbor feature. $M_{k,c}$ in the same phase is larger than that in different phases. Therefore, if d successive values are lower than the control limit, the next phase is indicated. The first phase is determined and the remaining data are updated as the new input in equation (10), then equations (10) and (11) are repeated to find the following phases. For each phase, key features reflecting the state of the batch operation need to be extracted. Figure 1 shows the schematic diagram of uneven-length data processing and phase division.

3.2. Multiway sparse weighted neighborhood preserving embedding (MSWNPE)

For each divided phase $X(I \times Jp)$ (p is the length of the divided phase), it is normalized along the batch direction and is rearranged as $X(Ip \times J)$. Then, the next vital step is to extract key features of each phase that reflect the operational state of the batch process. The NPE algorithm is susceptible to outliers and noise when the process data features are extracted, and it treats the data points in the neighborhood equally, which does not consider the impact of different distance values in the neighborhood. Therefore, a SWNPE algorithm is proposed for extracting process operation features. SWNPE calculates the optimal SR in the obtained neighbors, thereby reducing the number of calculations and maintaining the local sparse relationship. In the neighborhood, the points that are close to each other are more important to characterize the local structure than the points that are far away. So, the distance weight is introduced to the elements in the neighborhood, and the enhanced objective function shown in equation (12) is established to obtain the local optimal sparse structure,

$$\min_{s_i} \left\|x_i - X_{T(x_i)} s_i\right\|_2^2 + \lambda \|d_i \otimes s_i\|_1 \quad (12)$$

where $X_{T(x_i)} = [x_{i_1}, \ldots, x_{i_j}, \ldots, x_{i_k}] \in R^{m \times k}$ is the k nearest neighbors of x_i, the value of k can be determined by the [31]. $\lambda > 0$ is a regularization parameter that balances reconstruction error and sparsity, $d_i = [d_{i1}, d_{i2}, \ldots, d_{ik}] \in R^k$ is the distance weighted vector between x_i and its neighbors, $s_i = [s_i^1, s_i^2, \ldots, s_i^k]^{\mathrm{T}} \in R^k$ is the optimal sparse reconstruction coefficient of x_i, $d_i \otimes s_i = [d_{i1} \cdot s_{i1}, d_{i2} \cdot s_{i2}, \ldots, d_{ik} \cdot s_{ik}] \in R^k$.

Since the distance value represents the degree of proximity between two elements, a smaller distance value indicates that the two points are closer, which is more important for the feature of the local structure. Therefore, a smaller distance value d_{ij} corresponds to a larger sparsity coefficient s_{ij}. The distance value between two elements can be obtained by equation (13),

$$d_{ij} = \exp\left(\left\|x_i - x_{i_j}\right\|_2 / \sigma\right) \quad (13)$$

where $\sigma > 0$ is used to adjust the decay rate of the distance weight. The sparse coefficient matrix $S = [s_1, s_2, \ldots, s_n] \in R^{n \times n}$ is introduced and equation (12) can be rewritten as equation (14),

$$\min_S \|X - X \cdot S\|_2^2 + \lambda \|D \otimes S\|_1 \quad (14)$$

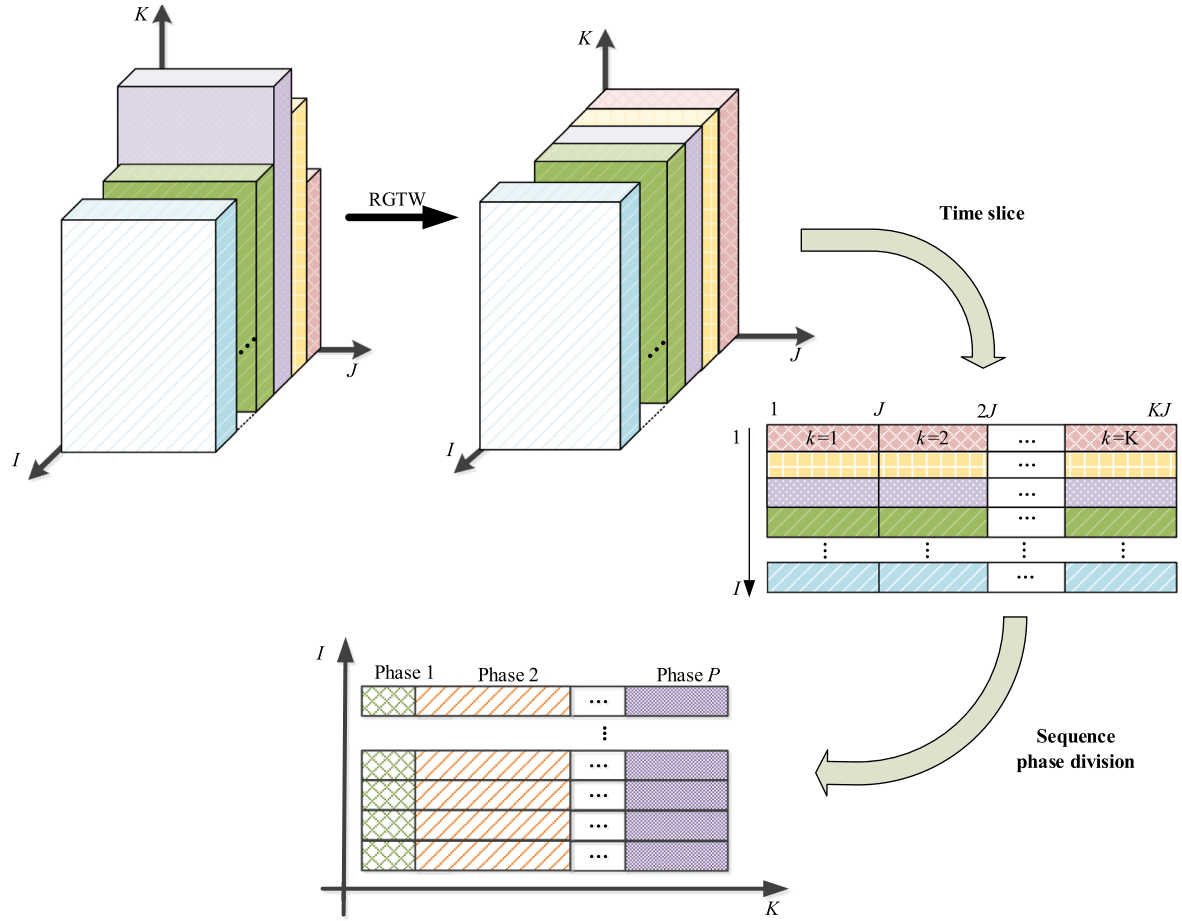


Figure 1. Schematic diagram of uneven length data processing and phase division.

where $D=[d_1,d_2,\ldots,d_n]\in R^{n\times n}$ is the distance matrix of the neighbors, and D_{ij} is the distance between x_i and x_j in the neighborhood obtained by equation (13). When $D=[d_1,d_2,\ldots,d_n]\in R^{n\times n}$ is not a distance matrix of the neighbors, $D_{ij}=0$. Similarly, $S=[s_1,s_2,\ldots,s_n]\in R^{n\times n}$ is a sparse matrix of neighbors, when it does not belong to neighbors, $S_{ij}=0$. Equation (14) is solved by the method of the optimal toolbox CVX [32].

After obtaining the optimal local sparse matrix S, the goal of SWNPE algorithm is to obtain the projection matrix $A\in R^{m\times d}(d<m)$, so that the local sparse structure of the data is maintained during the dimension reduction process. SWNPE algorithm can be written in the form of equation (15),

$$J_{\mathrm{SWNPE}}=\min_{A\in R^{m\times d}}\left\|\sum_{i=1}^{n}A^{\mathrm{T}}x_i-A^{\mathrm{T}}\sum_{j=i_1}^{i_k}S_{ij}x_j\right\|_2^2 \tag{15}$$

where S_{ij} is the optimal local sparse matrix obtained by equation (14). The method is robust to outliers and noise by introducing a locally optimal SR and fully extracts the local structure by weighting the distance values of neighboring points. Compared with the global SR, SWNPE algorithm can reduce the calculation time, and equation (15) can be calculated by equation (16),

$$\begin{aligned}J_{\mathrm{SWNPE}}&=\min_{A\in R^{m\times d}}\left\|\sum_{i=1}^{n}A^{\mathrm{T}}x_i-A^{\mathrm{T}}\sum_{j=i_1}^{i_k}S_{ij}x_j\right\|_2^2\\&=\min_{A\in R^{m\times d}}A^{\mathrm{T}}X(I-S)^{\mathrm{T}}(I-S)X^{\mathrm{T}}A\\&=\min_{A\in R^{m\times d}}A^{\mathrm{T}}X\widehat{M}X^{\mathrm{T}}A\end{aligned} \tag{16}$$

where, $\hat{M}=(I-S)^{\mathrm{T}}(I-S)$, to avoid degenerate solution, the constraint $A^{\mathrm{T}}XX^{\mathrm{T}}A=I$ is introduced; then, equation (16) can be transformed as:

$$J_{\mathrm{SWNPE}}=\min_{A^{\mathrm{T}}XX^{\mathrm{T}}A=I}A^{\mathrm{T}}X\hat{M}X^{\mathrm{T}}A. \tag{17}$$

The projection matrix A can be obtained by solving the generalized eigenvalue problem of equation (18),

$$X\hat{M}X^{\mathrm{T}}a=\lambda XX^{\mathrm{T}}a. \tag{18}$$

The projection matrix A is composed of eigenvectors $(a_1,a_2,\ldots,a_d)$ corresponding to the smallest d eigenvalues $(\lambda_1\leqslant\lambda_2\leqslant\cdots\leqslant\lambda_d)$. The optimal value of d should be the inherent structure that can effectively reflect the raw dynamic data [33].

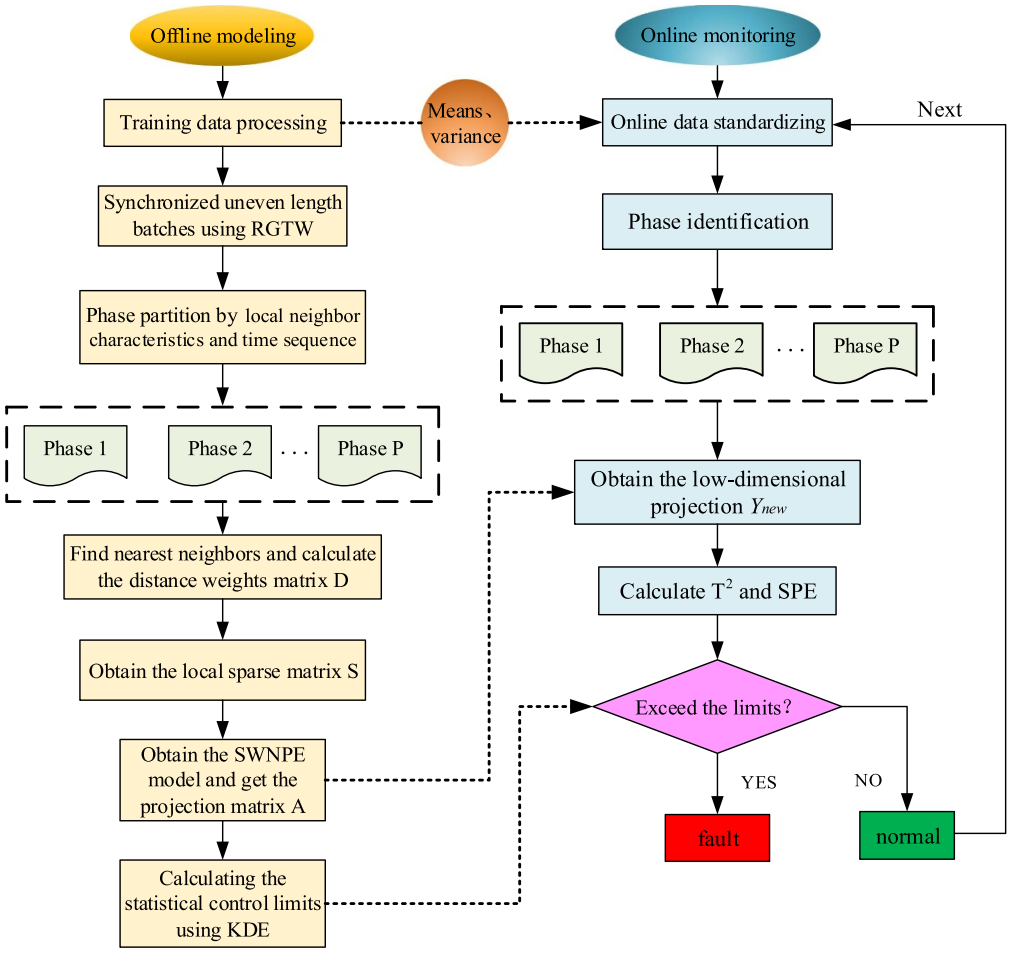


Figure 2. Flow chart of fault detection of SPD-MSWNPE.

4. Fault detection

When the batch process phases with uneven length are divided according to the local neighbor characteristics, the process local features reflecting the state of the batch operation process are extracted by SWNPE, T^2 and SPE statistics are established for process fault detection. $Y(y_1,\ldots,y_n)\in R^{n\times d}$ is the low-dimensional representation of the original variable $X(x_1,\ldots,x_m)\in R^{n\times m}$, where $Y=A^{\mathrm{T}}X$, the feature variables and residual variables have the following relationships:

$$\begin{aligned} X&=\hat{X}+\bar{X}=BY+E\\ Y&=A^{\mathrm{T}}X=\left(B^{\mathrm{T}}B\right)^{-1}B^{\mathrm{T}}X\\ E&=X-BY \end{aligned} \tag{19}$$

where $A^{\mathrm{T}}=\left(B^{\mathrm{T}}B\right)^{-1}B^{\mathrm{T}}$ is the conversion matrix. $\bar{X}$ represent the residual variables, and E represents the residual matrix.

T^2 statistic in the feature space is established to measure the fluctuations of the process data, as shown in equation (20),

$$T^2=\mathrm{y}_{\mathrm{new}}\Lambda^{-1}y_{\mathrm{new}}^{\mathrm{T}} \tag{20}$$

where $\Lambda^{-1}=((Y)^{\mathrm{T}}Y/(n-1))$ is the sample covariance matrix of Y. An SPE statistic as shown in equation (21) is created for the residual space to measure the random variation of the process,

$$\mathrm{SPE}=\|x_{\mathrm{new}}-\hat{x}_{\mathrm{new}}\|^2=e^{\mathrm{T}}e \tag{21}$$

where $\hat{x}_{\mathrm{new}}$ is the reconstruction vector of the low-dimensional projection y_{new}, that is $\hat{x}_{\mathrm{new}}=By_{\mathrm{new}}$.

The control limits of T^2 and SPE statistics are obtained by kernel density estimation (KDE) [34].

Figure 2 shows the monitoring process of fault detection based on SPD-MSWNPE, which includes two parts: offline modeling and online monitoring.

4.1. Offline modeling

(1) Pretreat the training data;
(2) RGTW is used for synchronization of uneven-length batches;
(3) Partition the phases based on local neighborhood features and time sequence;
(4) Seek nearest neighbors through k nearest neighbors;
(5) Calculate the distance weight matrix D by equation (13);
(6) Obtain the local sparse matrix S by equation (14);
(7) Obtain the projection matrix A by equation (16);

(8) Obtain the monitoring model and use KDE method to calculate the statistical control limits.

4.2. Online monitoring

(1) Standardize the online data;
(2) Identify the phases based on offline modeling;
(3) Obtain the low-dimensional projection Y_{new} of the online data through the projection matrix A;
(4) Calculate T^2 and SPE statistics of the online data through equations (20) and (21);
(5) Determine whether the statistics exceed the limits. If the statistics exceed the limits, it means that a fault has occurred.

5. Simulation verification

In this part, the performance of the proposed algorithm is verified through a multiphase nonlinear numerical process and the well-known penicillin fermentation process. The results are compared and analyzed using MPCA, multi-way NPE (MNPE), multi-way dynamic NPE (MDNPE), SPD multi-way global NPE (SPD-MGNPE), and SPD-MSWNPE. For SPD-MSWNPE, the phases are identified based on local neighbor features in the time sequence automatically, and the sparse weight matrix is calculated in each phase, which avoids the increase in computational complexity caused by global optimization. The proposed method calculates optimal SR and eigenvalue decomposition in divided phases and local structures, which can improve the computational efficiency of batch processes.

5.1. Multiphase nonlinear numerical example

A multiphase nonlinear numerical example is employed to test the monitoring effectiveness of the proposed method [35, 36]. The numerical example consists of two phases and seven variables $x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7]^{\mathrm{T}}$, the duration of each batch varies from 95 to 105 sampling points. For the first phase, the process data are generated as follows:

$$\begin{aligned} x_1(k) &= 50 + k + \zeta_1 \\ x_2(k) &= 50 - 0.5 * k + \zeta_2 \\ x_3(k) &= 45 - 0.2 * k + \zeta_3 \\ x_4(k) &= 0.1 * x_2(k) + 0.5 * x_3(k) + \zeta_4 \\ x_5(k) &= x_2(k) - 0.2 * x_3(k) + \zeta_5 \\ x_6(k) &= 10 + k + \zeta_6 \\ x_7(k) &= 0.1 * x_6(k) + k + \zeta_7 \end{aligned} \quad . \tag{22}$$

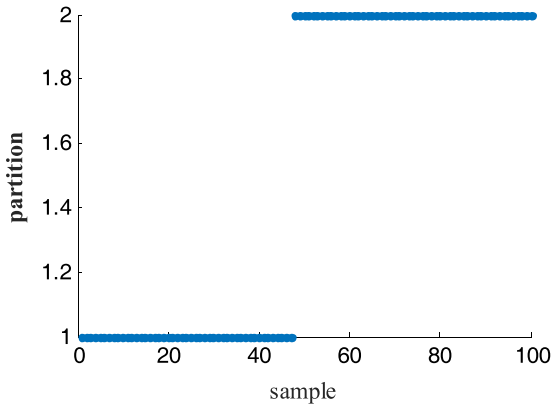


Figure 3. Phase division results of the numerical simulation process.

For the second phase, the process data are generated as follows:

$$\begin{aligned} x_1(k) &= 60 + k + \zeta_1 \\ x_2(k) &= 50 - 0.4 * k + \zeta_2 \\ x_3(k) &= 45 - 0.2 * k + \zeta_3 \\ x_4(k) &= 0.2 * x_2(k) + 0.5 * x_3(k) + \zeta_4 \\ x_5(k) &= x_2(k) - 0.2 * x_3(k) + \zeta_5 \\ x_6(k) &= 10 + k + \zeta_6 \\ x_7(k) &= 0.1 * x_6(k) + k + \zeta_7 \end{aligned} \tag{23}$$

where k is the number of sampling points. $[\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6, \zeta_7]$ is the independent random noise with Gaussian distribution of $N(0, 0.1)$. The train data are composed of 20 batches with different lengths, and the whole process varies from 95 to 105. To verify the effectiveness of the proposed algorithm, variable 1 is introduced a $0.05 * (k - 30)$ ramp fault from the 31th sampling point to the 80th sampling point for fault detection and verification. The training data are synchronized and unfolded, then the proposed algorithm is used for phase division, and the partition results are shown in figure 3.

We can see from figure 3 that the whole process is divided into two phases: 0–52 and 53–100 respectively, this is consistent with the numerical batch process, which proves that the local neighbor features are similar in the same phase and are different between phases.

Figures 4–8 show the comparison results of fault detection of MPCA, MNPE, MDNPE, SPD-MGNPE, and the proposed algorithm. In these figures, the red dashed line represents the control limit, and the solid line marked with a circle represents the test results. Figure 4 shows the fault detection results of MPCA, and it can be seen that T^2 and SPE detect the fault at the 57th and 52th sampling points, respectively. We can see that both T^2 and SPE of MNPE detect the fault at the 49th sampling points in figure 5. As can be seen in figure 6, T^2

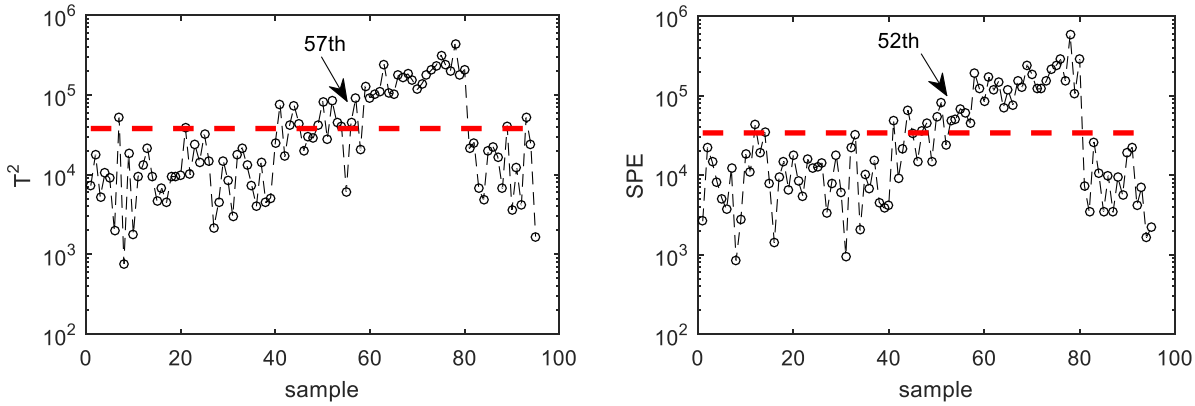


Figure 4. T^2 and SPE monitoring charts of MPCA.

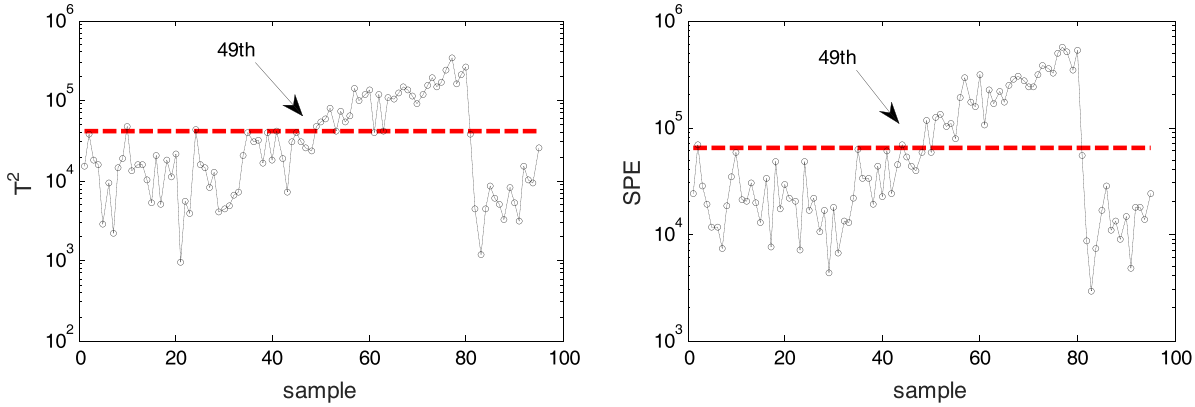


Figure 5. T^2 and SPE monitoring charts of MNPE.

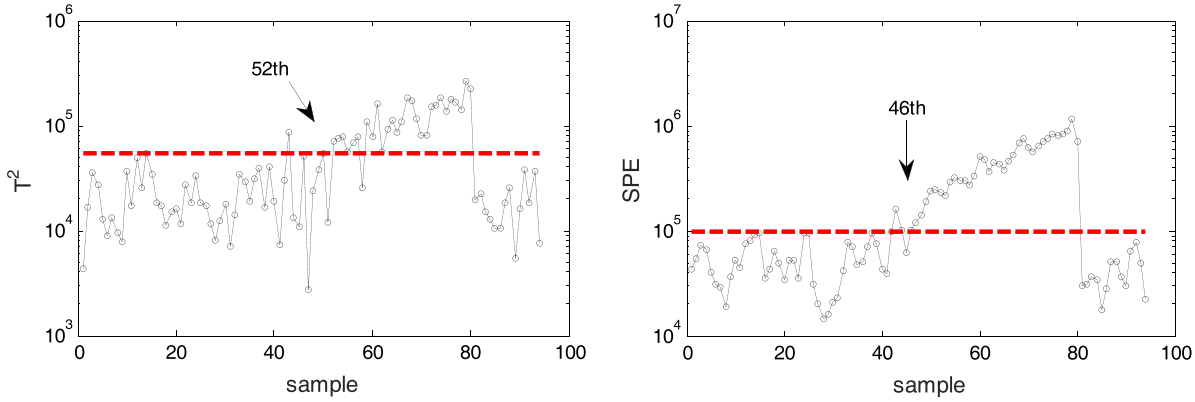


Figure 6. T^2 and SPE monitoring charts of DNPE.

and SPE of MDNPE method detects the fault at the 52th and 46th sampling points. T^2 and SPE fault detection results of SPD-MGNPE method are shown in figure 7, where the faults are detected at the 47th and 48th sampling points respectively, which can detect the fault earlier because the statistical model is established based on the partition phases and the multiplicity information can be revealed. The monitoring results for the proposed method are presented in figure 8, it can be seen that T^2 and SPE detect the faults earlier than other methods and have lower false alarms, because the proposed method mainly synchronizes batch trajectories, which can fully extract multiphase batch operation features and is more sensitive to a fault.

5.2. *Penicillin fermentation process*

The penicillin fermentation process is a typical batch process with multiple operation phases, and it is widely used to evaluate the monitoring and fault detection of a multiphase batch process. To facilitate the application of advanced process modeling, monitoring, and control methods to the

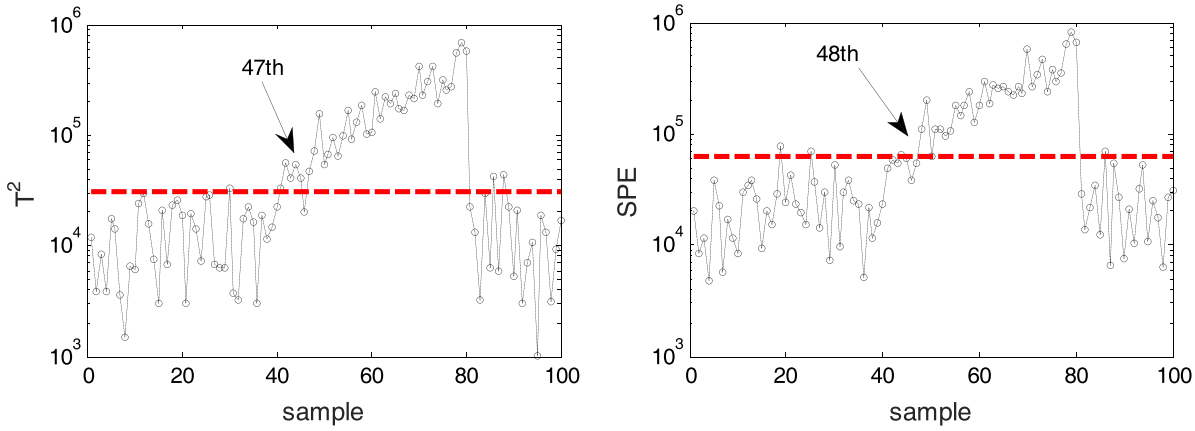


Figure 7. T^2 and SPE monitoring charts of SPD-MGNPE.

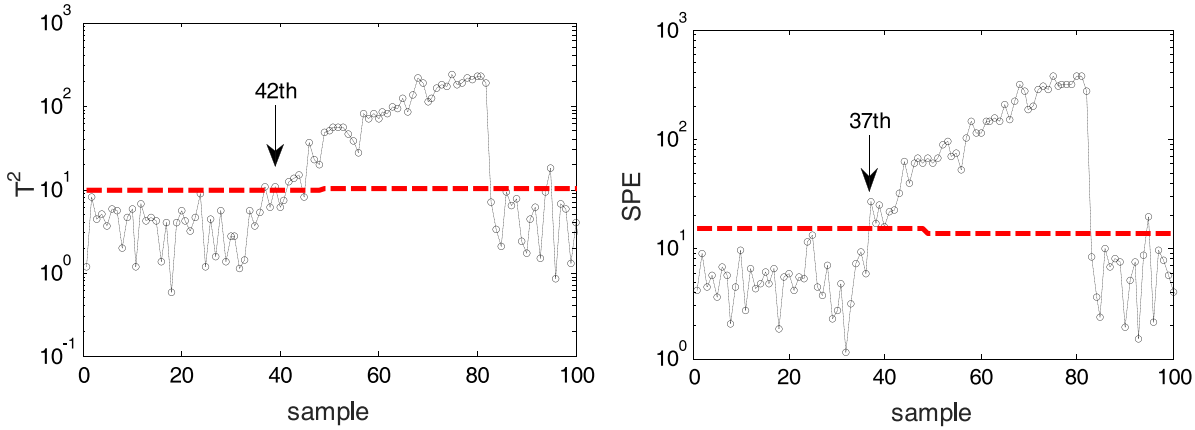


Figure 8. T^2 and SPE monitoring charts of the proposed algorithm.

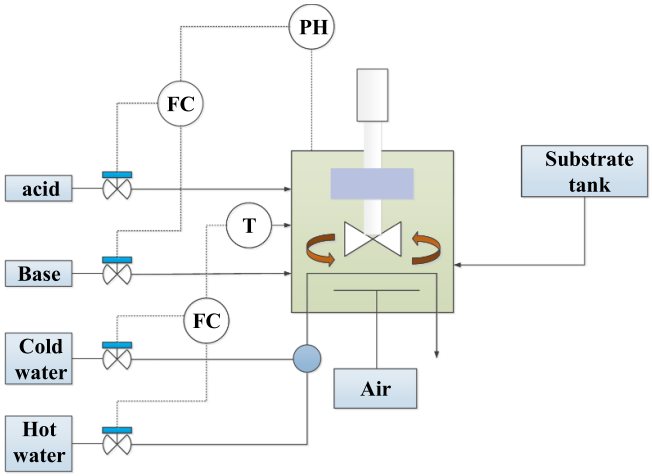


Figure 9. Penicillin fermentation process.

Table 1. Process variables for the penicillin fermentation.

Variable number	Process variable	Unit
1	Aeration rate	l h^{-1}
2	Agitator power	r min^{-1}
3	Substrate feed flow rate	l h^{-1}
4	Substrate feed temperature	K
5	Substrate concentration	g l^{-1}
6	DO	%
7	Culture volume	l
8	CO_2	g l^{-1}
9	pH	
10	Temperature	K
11	Penicillin concentration	g l^{-1}

complex reaction process of penicillin, the Illinois Institute of Technology team developed the Pensim2.0 simulation platform in 2002 [37], shown in figure 9.

This simulation platform can fully simulate batch processes and reflect the characteristics of time-varying and multi-phase batch processes. Therefore, this paper would use Pensim2.0 to generate penicillin fermentation process data to verify the effectiveness of the proposed algorithm.

The running time of each batch is set to 395–505 h, and the sampling time interval is set to 1 h. By setting different initial conditions and parameters, the data of 20 normal conditions are obtained. The initial conditions and parameters for all settings are within normal limits. All variables have noise. Eleven process variables (see table 1) are selected from 18 generated process variables as monitoring variables to form the process

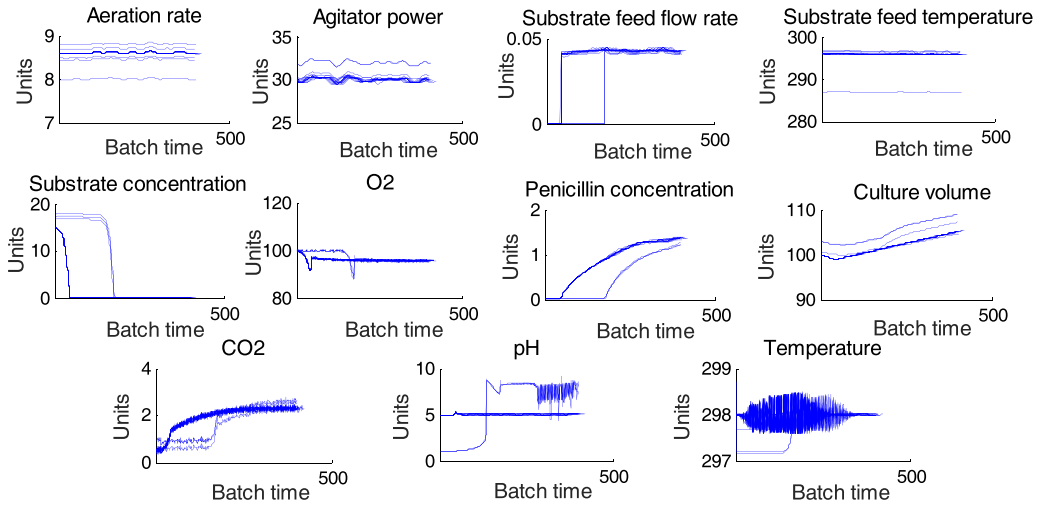


Figure 10. The measure profiles of the selected 11 variables before synchronization.

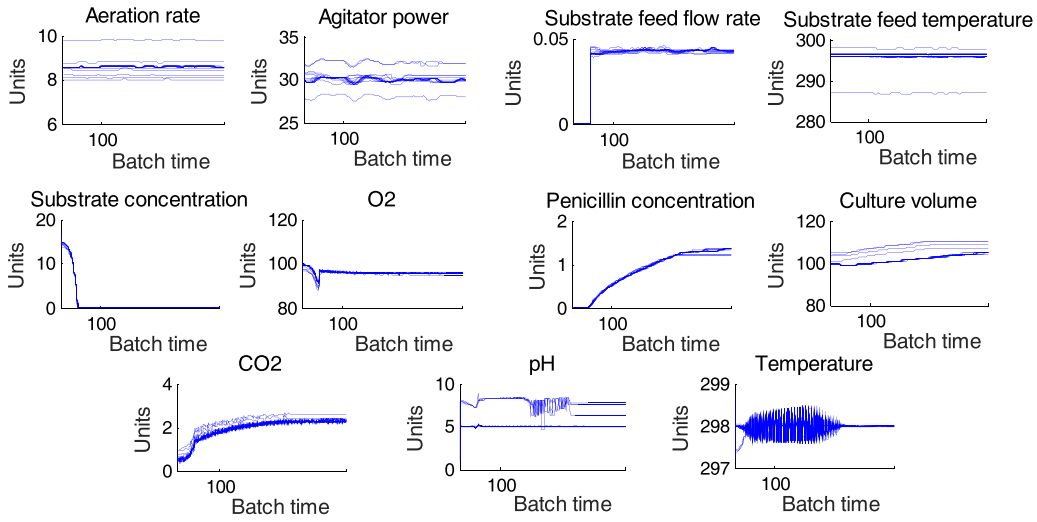


Figure 11. The measure profiles of the selected 11 variables after synchronization.

dataset. In Pensim2.0, the variables 1, 2, 3 in table 1 are set to the fault variables. Fault types include ramp fault and step fault.

Figure 10 shows the measure profiles of the 11 variables selected from the 20 normal batches before synchronization, we can see that the variables of different batches have different measure profiles and most of the process variables have multiphase characteristic, but the phases have not happened simultaneously, especially the differences between batches of substrate feed flow rate, substrate concentration, O2, penicillin concentration, CO2, PH, temperature, etc. Therefore, the variables of different batches need to be synchronized.

In this paper, we use RGTW to synchronize the original data of 20 normal batches with uneven lengths, and the results of the synchronized data are shown in figure 11. As can be seen from figure 11, RGTW synchronizes the uneven-length data while retaining the features of the variables. The measure profiles of substrate feed flow rate, substrate concentration, O2, penicillin concentration, CO2, PH, and temperature show the same trend.

The obtained 20 batches of process dataset are synchronized by RGTW. By analyzing the trajectories of process variables, each batch can be divided into several phases precisely, and the results of phase partition based on local neighbor features in time sequence are illustrated in figure 12. From figure 12, we can see the penicillin process can be divided into five phases by using the proposed algorithm, namely 0–22, 23–42, 43–138, 139–256 and 257–400, respectively. The phase division can not only highlight the local neighbor features, but also enhance the process state understanding and analysis.

The performance of the proposed algorithm is verified by setting different fault types and magnitudes. Six faulty batches in table 2 are obtained by setting different fault variables and fault signals.

Fault detection rate (FDR) and false alarm rate (FAR) are usually used as evaluation indexes in the field of fault

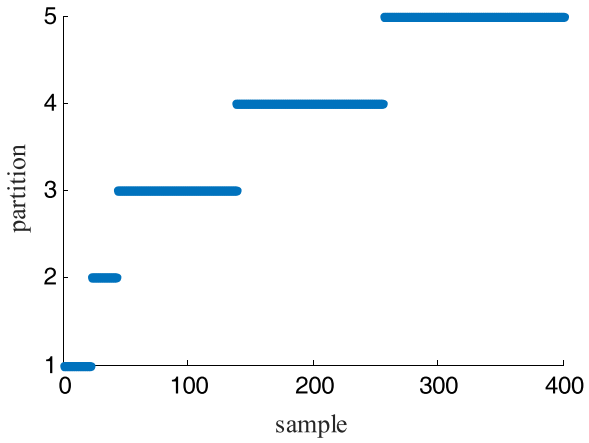


Figure 12. Phase partition results of the penicillin fermentation process.

Table 2. Fault batches in the penicillin fermentation process.

Fault No.	Variable name	Fault type	Amplitude	Occurrence time (h)	End time (h)
1	Aeration rate	Step	3	200	400
2	Aeration rate	Ramp	0.4	200	400
3	Agitator rate	Step	2	100	300
4	Agitator rate	Ramp	0.1	200	400
5	Substrate feeding rate	Step	1	200	400
6	Substrate feeding rate	Ramp	0.005	200	400

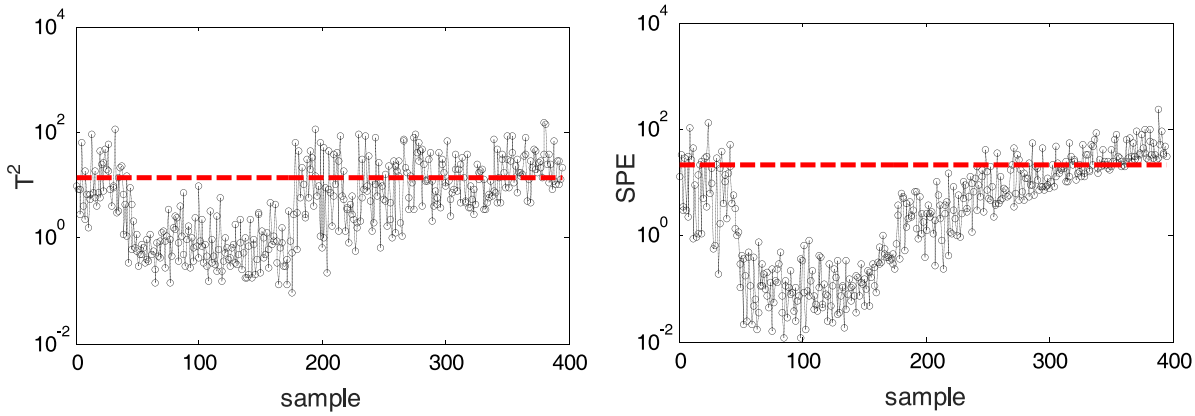


Figure 13. T^2 and SPE monitoring charts of the MPCA method under fault 2.

monitoring. FDR refers to the ratio of the number of correctly detected faults to the total number of faults, and FAR refers to the ratio of the number of incorrectly detected faults to the sum of the normal operating data, which are shown in equation (24),

$$\begin{aligned} \mathrm{FDA} &= \frac{J_1}{J_2} \times 100\% \\ \mathrm{FAR} &= \frac{J_3}{J_4} \times 100\% \end{aligned} \tag{24}$$

where J_1 indicates the number of fault detected successfully; J_2 indicates the total number of fault; J_3 the number of incorrectly detected faults; J_4 represents the total number of samples.

As shown in table 2, fault 2 is a ramp fault with an amplitude of 0.4 added to the aeration rate from the 200th sampling point to the end. The detection effects of the five methods under fault 2 are shown in figures 13–17. Figure 13 shows the T^2 and SPE detection results of the MPCA method, where we can see that MPCA cannot detect fault 2 in time. Figure 14 is the T^2 and SPE monitoring charts of MNPE, where it can be seen that T^2 of MNPE cannot detect the fault, and SPE detects the fault at the 331th sampling point. Figure 15 shows the T^2 and SPE monitoring charts of MDNPE under fault 2, where we can see that T^2 and SPE of MDNPE detect the fault at the 299th and 337th sampling points, respectively, it can detect the fault earlier than MPCA and MNPE. Figure 16 is the T^2 and SPE monitoring charts of the SPD-MGNPE method show that T^2 and SPE of SPD-MGNPE detect the fault at the 258th and

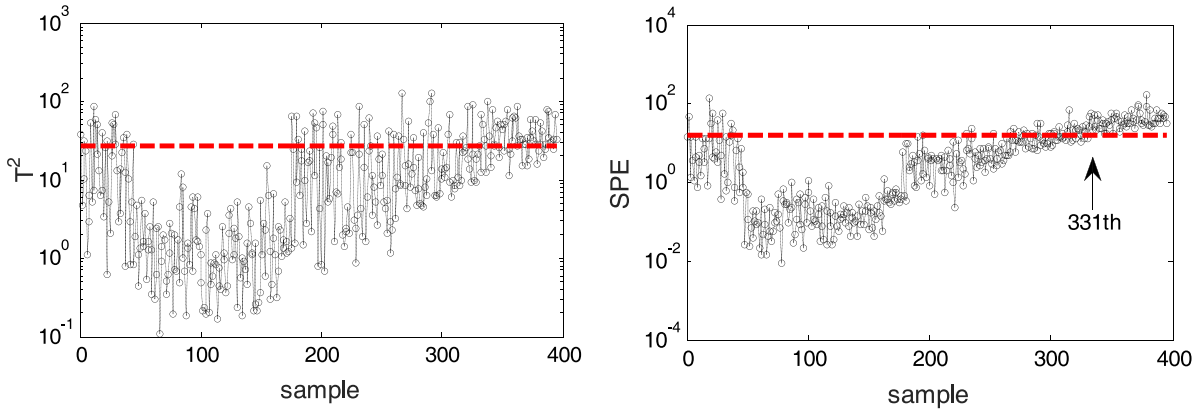


Figure 14. T^2 and SPE monitoring charts of the MNPE method under fault 2.

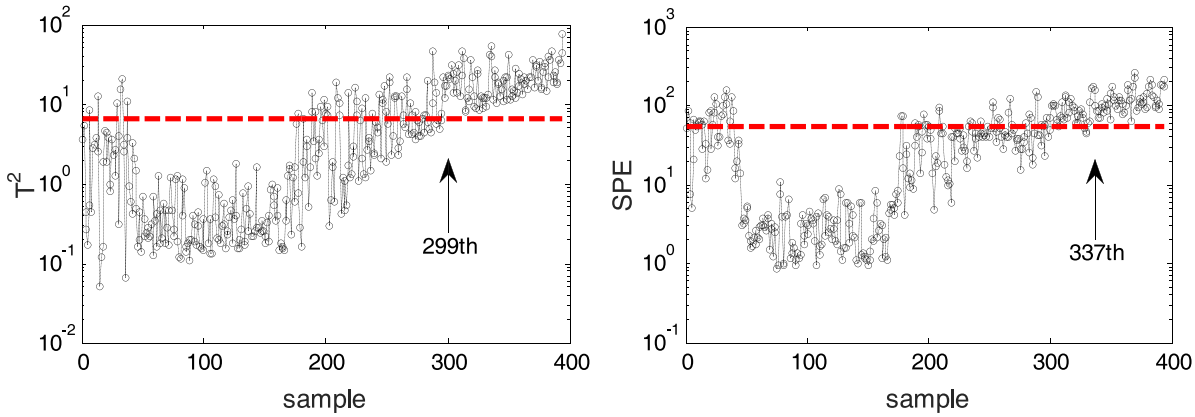


Figure 15. T^2 and SPE monitoring charts of the MDNPE method under fault 2.

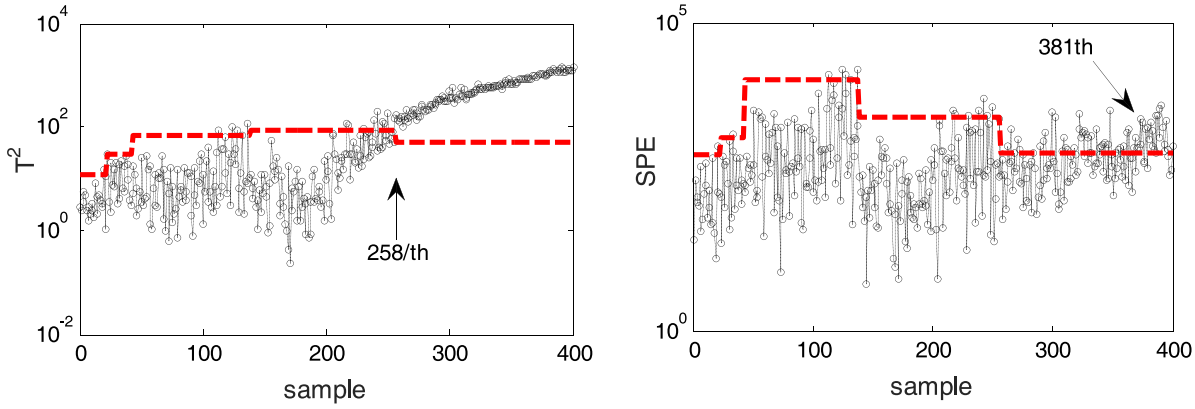


Figure 16. T^2 and SPE monitoring charts of the SPD-MGNPE method under fault 2.

381th sampling points, respectively, the effect of fault detection is better than MPCA, MNPE and MDNPE. T^2 and SPE of the SPD-MSWNPE are shown in figure 17, we can see that T^2 and SPE of SPD-MSWNPE detect the fault at the 220th and 214th sampling points, respectively. Compared with MPCA, MNPE, MDNPE and SPD-MGNPE, SPD-MSWNPE has a higher FDR and a lower FAR. This is because SPD-MSWNPE can not only realize synchronization of multiple phase batches but also maintain the local neighborhood structure of the process for process monitoring.

Fault 3 is a step fault with an amplitude of 2 added to the variable agitator rate from the 100th sampling point to the 300th sampling point. Figures 18–22 show the fault detection results of MPCA, MNPE, MDNPE, SPD-MGNPE, and SPD-MSWNPE under fault 3. It can be seen that T^2 and SPE of MPCA in figure 18 cannot detect the fault effectively. Figure 19 is the T^2 and SPE monitoring charts of MNPE, we can see that both T^2 and SPE can detect the fault in time when the fault occurs, but there are some false alarms before the 50 sampling points. The T^2 and SPE monitoring charts of

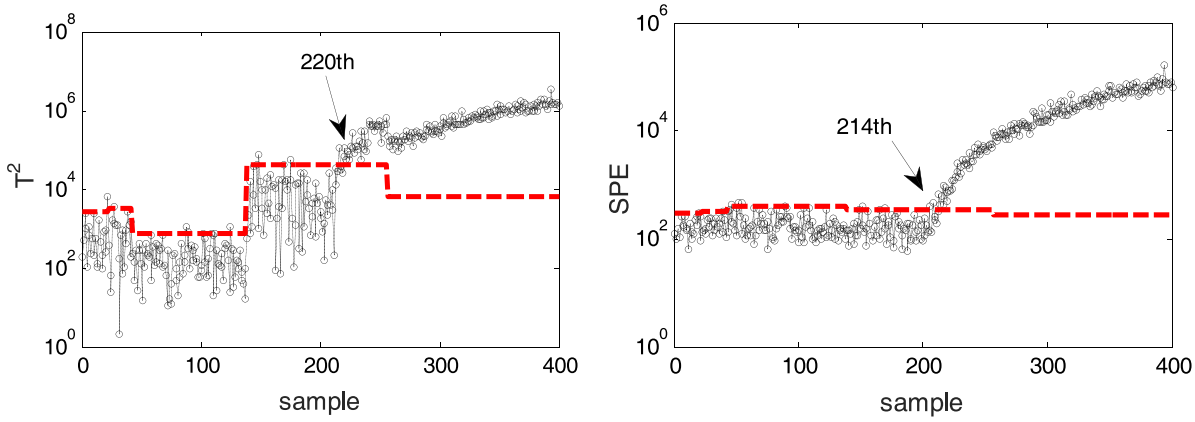


Figure 17. T^2 and SPE monitoring charts of the SPD-MSWNPE method under fault 2.

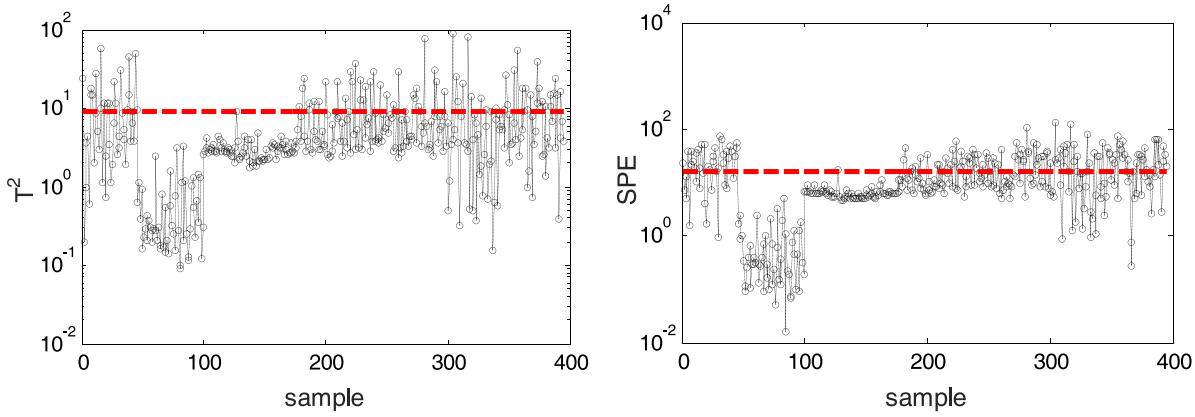


Figure 18. T^2 and SPE monitoring charts of the MPCA method under fault 3.

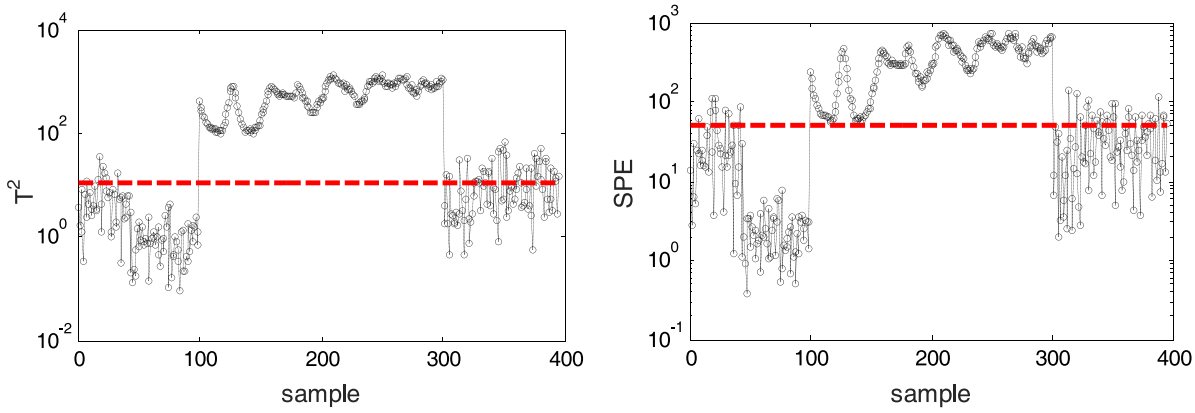


Figure 19. T^2 and SPE monitoring charts of the MNPE method under fault 3.

MDNPE are shown in figure 20, we can see that T^2 cannot detect the fault and SPE can detect the fault, but there are some false alarms before the 50 sampling points. Figure 21 is the T^2 and SPE monitoring charts of the SPD-MGNPE method, it can be seen that T^2 can detect the fault in time; however, there are some false alarms before the 100 sampling points, and SPE cannot detect the fault effectively. The fault detection results of the proposed algorithm are shown in figure 22, SPD-MSWNPE can detect the fault at the first time of occurrence and has the lowest FAR when the fault occurs.

Tables 3 and 4 show the comparison results of FDR and FAR monitored by MPCA, MNPE, MDNPE, SPD-MGNPE, and SPD-MSWNPE for six faults. The proposed algorithm has the highest FDR and lowest FAR, the whole process is divided into phases and a process model is built for each phase, so that the operation characteristics of the process can

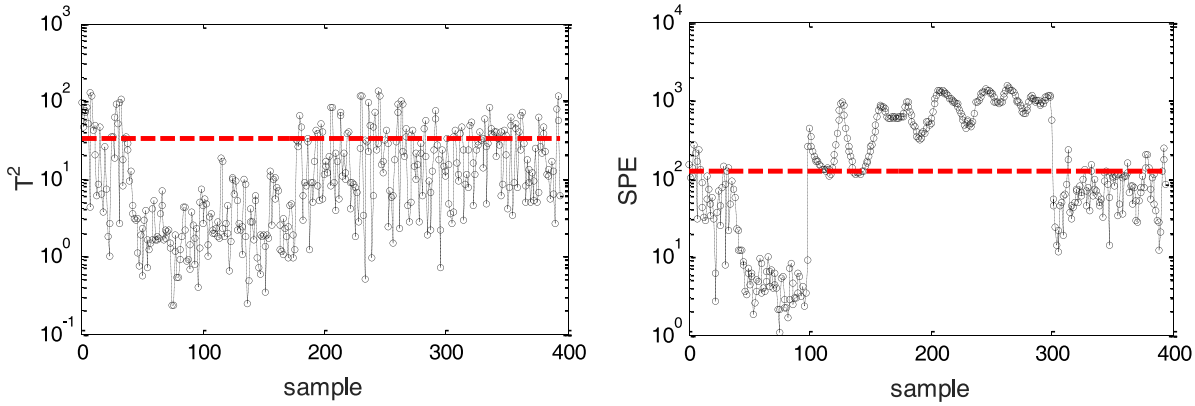


Figure 20. T^2 and SPE monitoring charts of the MDNPE method under fault 3.

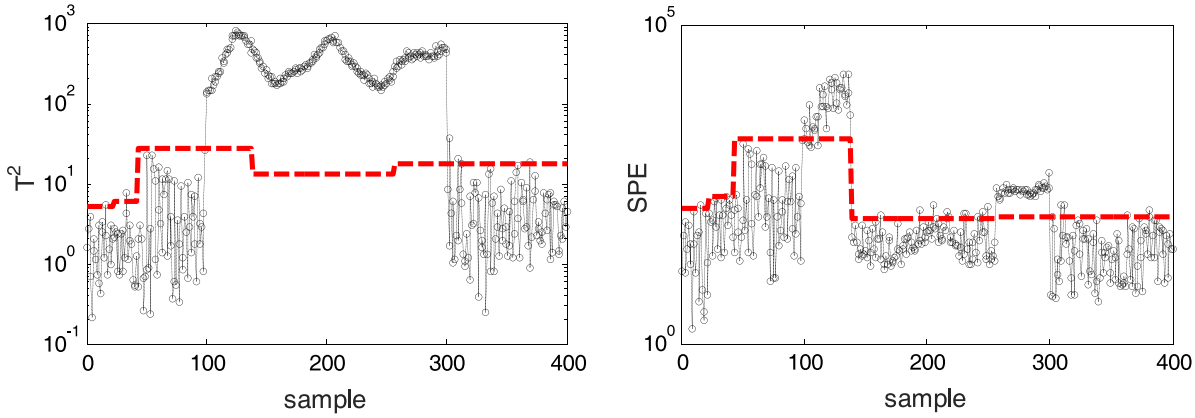


Figure 21. T^2 and SPE monitoring charts of the SPD-MGNPE method under fault 3.

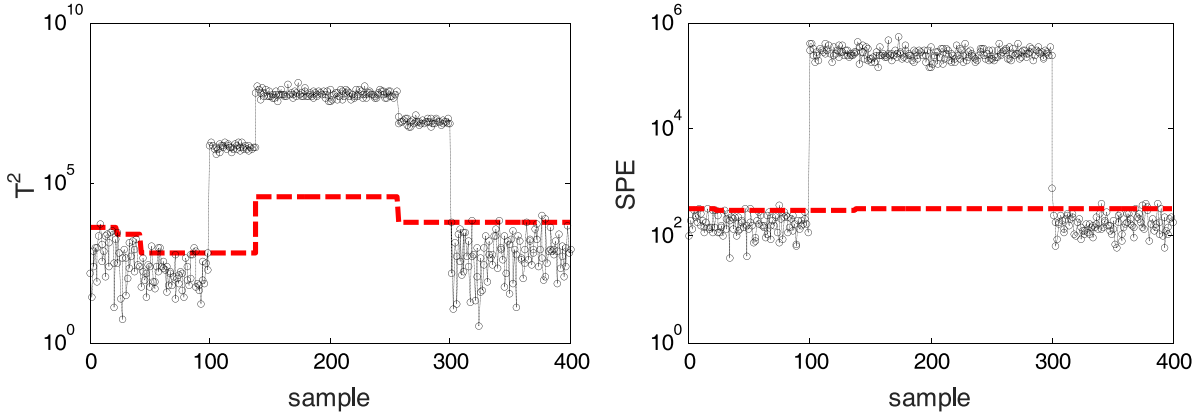


Figure 22. T^2 and SPE monitoring charts of the SPD-MSWNPE method under fault 3.

be effectively represented. Step faults can be detected by SPD-MGNPE and SPD-MSWNPE immediately like fault 1 and fault 3, but SPD-MSWNPE has a lower FAR. Because the glucose substrate feed rate propagates slowly through the correlated variables, fault 5 cannot be detected effectively. In contrast to step faults, ramp faults 2, 4, 6 are more difficult to detect due to their slow variation. In the detection of ramp faults, SPD-MSWNPE detects the faults earlier than those four methods, which first detects the faults with less FAR, this is because SPD-MSWNPE can synchronize the batches with several phases and extract the neighbor structure by considering the weighted distance value of the nearest neighbor elements.

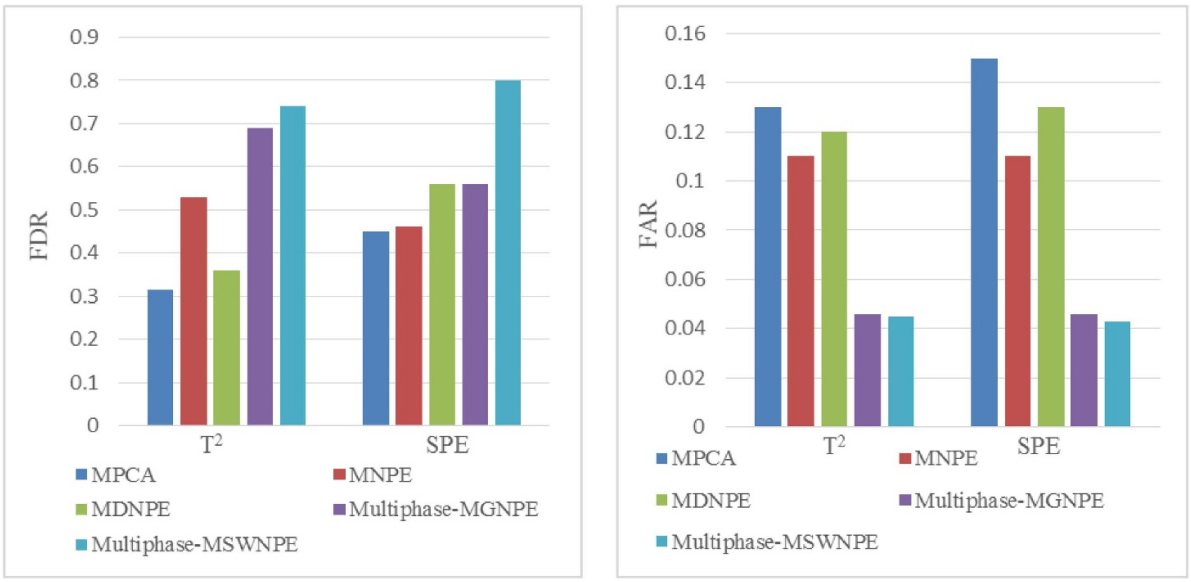
Figure 23 illustrates the average FDRs and average FARs of five methods under six faults. The monitoring performance is presented using a bar chart to explain the fault detection effect of each method more intuitively. It can be seen from figure 23(a) that FDR of the proposed method is higher than

Table 3. Fault detection rates (FDRs).

	MPCA		MNPE		MDNPE		Multiphase–MGNPE		SPD-MSWNPE	
Fault No.	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
1	0.5000	0.6800	0.3400	0.6100	0.6700	0.4200	1	1	1	1
2	0.4450	0.4650	0.3650	0.5800	0.6900	0.5700	0.7650	0.3450	0.9100	0.9500
3	0.1550	0.2700	1	1	0.1650	0.9550	1	0.4500	1	1
4	0.4950	0.5550	0.7850	0.0300	0.3250	0.6900	0.7450	0.7900	0.7750	0.8100
5	0.0300	0.0150	0.0050	0.0100	0.0250	0.0050	0.0100	0.0700	0.0700	0.0500
6	0.2590	0.7150	0.7050	0.5600	0.3000	0.7200	0.6150	0.6600	0.6950	0.9700

Table 4. False alarm rates (FARs).

	MPCA		MNPE		MDNPE		Multiphase–MGNPE		SPD-MSWNPE	
Fault No.	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
1	0.1500	0.1650	0.1150	0.1650	0.1500	0.1850	0.0500	0.0400	0.0500	0.0350
2	0.1250	0.0450	0.1100	0.1250	0.0700	0.1550	0.0350	0.0350	0.0500	0.0450
3	0.2500	0.4200	0.2150	0.1850	0.2500	0.1200	0.0350	0.0350	0.0250	0.0450
4	0.1300	0.1350	0.0850	0.0550	0.0850	0.1150	0.0350	0.0450	0.050	0.0450
5	0.0500	0.0500	0.0450	0.0550	0.0600	0.0900	0.0650	0.0750	0.0450	0.0550
6	0.1000	0.1250	0.0950	0.0700	0.1100	0.1050	0.0550	0.0500	0.0500	0.0350



(a) (b)

Figure 23. Comparison charts of (a) average FDR and (b) average FAR, under the 5 fault batches.

that of other methods. Figure 23(b) shows that FAR of SPD-MSWNPE is lower than that of other methods. By comparing the average FDR and average FAR under six faults, we can conclude that SPD-MSWNPE is superior to MPCA, MNPE, MDNPE, and SPD-MGNPE methods.

6. Conclusion

This paper proposes a process fault detection algorithm based on SPD-MSWNPE. The batch process data of uneven length are processed by RGTW method, and the sequential phase partition method is used to effectively divide the phases. Feature selection of different phases is achieved through SR, thereby filtering out irrelevant information that affects process data modeling, such as noise and outliers. For the problem of excessive calculation by using SR to find the global optimal solution, the local sparse structure is established to reduce the calculation amount and maintain the local manifold structure. To maintain the data structure and accurately reflect the operating characteristics of the process , the distance weights between the neighboring points are introduced to construct an enhanced objective function, and the optimal local sparse structure is obtained. The effectiveness of the proposed

algorithm is verified through a numerical simulation process and the Penicillin fermentation process.

Data availability statement

The data that support the findings of this study are available at www.chee.iit.edu/∼control/software.html.

These data were derived from the following resources available in the public domain: Birol, Ündey and Cinar 2002 A modular simulation package for fed batch fermentation: penicillin production *Comput. Chem. Eng.* **26** 1553–65.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62263021), the College Industrial Support Project of Gansu Province (2023CYZC-24) and Lanzhou Youth Science and Technology Talent Innovation Project (2023-QN-36).

Author contributions

Yan Zhang: Software; validation; writing-original draft. Xiaoqiang Zhao: Supervision; writing-review and editing. Jie Cao: Conceptualization; methodology. Yongyong Hui: Data curation; investigation.

ORCID iDs

Yan Zhang https://orcid.org/0000-0002-3577-6752
Xiaoqiang Zhao https://orcid.org/0000-0001-5687-942X
Yongyong Hui https://orcid.org/0000-0001-5014-887X

References

[1] Lu S X, Yue Y Q, Liu X Y, Wu J and Wang Y Q 2023 A novel unbalanced weighted KNN based on SVM method for pipeline defect detection using eddy current measurements *Meas. Sci. Technol.* **34** 014001
[2] Mou M, Zhao X Q, Liu K, Cao S Y and Hui Y Y 2023 A latent representation dual manifold regularization broad learning system with incremental learning capability for fault diagnosis *Meas. Sci. Technol.* **34** 075005
[3] Ge Z and Song Z 2007 Process monitoring based on independent component analysis−principal component analysis (ICA−PCA) and similarity factors *Ind. Eng. Chem. Res.* **46** 2054–63
[4] Huang J and Yan X 2015 Gaussian and non-Gaussian double subspace statistical process monitoring based on principal component analysis and independent component analysis *Ind. Eng. Chem. Res.* **54** 1015–27
[5] Nomikos P and Macgregor J F 1994 Monitoring batch process using multiway principal component analysis *AIChE J.* **40** 1361–75
[6] Nomikos P and Macgregor J F 1995 Multi-way partial least squares in monitoring batch processes *Chemometr. Intell. Lab. Syst.* **30** 97–108
[7] Zhang S and Bao X 2023 Slow-varying batch process monitoring based on canonical variate analysis *Can. J. Chem. Eng.* **101** 400–19
[8] Deng X G and Du K Y 2022 Efficient batch process monitoring based on random nonlinear feature analysis *Can. J. Chem. Eng.* **100** 1826–37
[9] Zhang H, Deng X, Zhang Y, Hou C and Li C 2021 Dynamic nonlinear batch process fault detection and identification based on two-directional dynamic kernel slow feature analysis *Can. J. Chem. Eng.* **99** 306–33
[10] Lakshmi Priya Palla G and Kumar Pani A 2023 Independent component analysis application for fault detection in process industries: literature review and an application case study for fault detection in multiphase flow systems *Measurement* **209** 112504
[11] Xu C, Huang D, Cai B, Chen H and Liu Y 2023 A complex-valued slow independent component analysis based incipient fault detection and diagnosis method with applications to wastewater treatment processes *ISA Trans.* **135** 213–32
[12] Peng Y and Lu B-L 2017 Discriminative extreme learning machine with supervised sparsity preserving for image classification *Neurocomputing* **261** 242–52
[13] Hui Y and Zhao X 2018 Multi-phase batch process monitoring based on multiway weighted global neighborhood preserving embedding method *J. Process Control* **69** 44–57
[14] Liu Y, Hu Z and Zhang Y 2022 Symmetric positive definite manifold learning and its application in fault diagnosis *Neural Netw.* **147** 163–74
[15] Xu X and Ding J 2021 Decentralized dynamic process monitoring based on manifold regularized slow feature analysis *J. Process Control* **98** 79–91
[16] Zhang S and Bao X 2022 Two-dimensional multiphase batch process monitoring based on sparse canonical variate analysis *J. Process Control* **116** 185–98
[17] Zhu J, Yao Y and Gao F 2020 Multiphase two-dimensional time-slice dynamic system for batch process monitoring *J. Process Control* **85** 184–98
[18] Zhang H, Tian X, Deng X and Cao Y 2018 Multiphase batch process with transitions monitoring based on global preserving statistics slow feature analysis *Neurocomputing* **293** 64–86
[19] Chunhao D, Peng C and Kang O 2020 Enhanced high-order information extraction for multiphase batch process fault monitoring *Can. J. Chem. Eng.* **98** 2187–204
[20] Agarwal P, Aghaee M, Tamer M and Budman H 2022 A novel unsupervised approach for batch process monitoring using deep learning *Comput. Chem. Eng.* **159** 107694
[21] Vijayan S V, Mohanta H K, Rout B K and Pani A K 2023 Adaptive soft sensor design using a regression neural network and bias update strategy for non-linear industrial processes *Meas. Sci. Technol.* **34** 085012
[22] Hu K and Yuan J 2008 Statistical monitoring of fed-batch process using dynamic multiway neighborhood preserving embedding *Chemometr. Intell. Lab. Syst.* **90** 195–203
[23] Xiaoqiang Z and Tao W 2017 Tensor dynamic neighborhood preserving embedding algorithm for fault diagnosis of batch process *Chemometr. Intell. Lab. Syst.* **162** 94–103
[24] Yang Y, Wang Y and Xue X 2017 Discriminant sparse locality preserving projection for face recognition *Multimedia Tools Appl.* **76** 2697–712
[25] Qiao L, Chen S and Tan X 2010 Sparsity preserving projections with applications to face recognition *Pattern Recognit.* **43** 331–41
[26] Hui Y and Zhao X 2020 Sparse representation preserving embedding based on extreme learning machine for process monitoring *Trans. Inst. Meas. Control* **42** 1895–907
[27] González-Martínez J M, Ferrer A and Westerhuis J A 2011 Real-time synchronization of batch trajectories for on-line

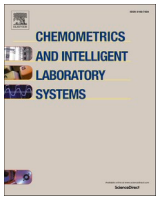
multivariate statistical process control using dynamic time warping *Chemometr. Intell. Lab. Syst.* **105** 195–206
[28] Zhang Y, Zhao X Q, Hui Y Y and Liu K 2022 Online monitoring and fault diagnosis for uneven length batch process based on multi-way orthogonal enhanced neighborhood preserving embedding *Asia-Pac. J. Chem. Eng.* **17** e2763
[29] Kassidas A, Macgregor J F and Taylor P A 1998 Synchronization of batch trajectories using dynamic time warping *AIChE J.* **44** 864–75
[30] Zhang S and Zhao C 2019 Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly *IEEE Trans. Ind. Electron.* **66** 3773–83
[31] Liou J W and Liou C Y Neighborhood selection and eigenvalues for embedding data complex in low dimension *Intelligent Information and Database Systems (ACIIDS 2012)* pp 413–22
[32] Grant M and Boyd S 2014 CVX: Matlab software for disciplined convex programming (version 2.1)
[33] Miao A, Ge Z, Song Z and Zhou L 2013 Time neighborhood preserving embedding model and its application for fault detection *Ind. Eng. Chem. Res.* **52** 13717–29
[34] Luo L, Bao S, Mao J and Tang D 2016 Nonlocal and local structure preserving projection and its application to fault detection *Chemometr. Intell. Lab. Syst.* **157** 177–88
[35] Jiang Q, Gao F, Yi H and Yan X 2019 Multivariate statistical monitoring of key operation units of batch processes based on time-slice CCA *IEEE Trans. Control Syst. Technol.* **27** 1368–75
[36] Qin Y, Zhao C and Gao F 2016 An iterative two-step sequential phase partition (ITSPP) method for batch process modeling and online monitoring *AIChE J.* **62** 2358–73
[37] Birol G, Ündey C and Çinar A 2002 A modular simulation package for fed-batch fermentation:penicillin production *Comput. Chem. Eng.* **26** 1553–65

Chemometrics and Intelligent Laboratory Systems 240 (2023) 104922

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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



Nonlinear multiphase batch process monitoring and quality prediction using multi-way concurrent locally weighted projection regression

Yan Zhang [a,c], Jie Cao [a,b,*], Xiaoqiang Zhao [a,c,d], Yongyong Hui [a,c,d]

[a] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, China
[b] Manufacturing Informatization Engineering Research Center of Gansu Province, Lanzhou, 730050, China
[c] Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou, 730050, China
[d] National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, 730050, China

ARTICLE INFO

Keywords:
Batch process
Nonlinearity
Multiphase
Concurrent locally weighted projection regression
Quality prediction

ABSTRACT

The batch process has the characteristics of nonlinear and multiphase due to variation operation conditions. Nonlinear and multiphase modeling of the batch process is very important to understand the running state of a process and improve the monitoring effect. In this work, a multiphase multi-way concurrent locally weighted projection regression algorithm is proposed. Firstly, the entire process is partitioned into phases according to local quality-related characteristics and time sequence. Secondly, the nonlinear process is modeled with locally linear models in each partitioned phase, the global approximation results are obtained by weighting all the local models. Thirdly, the complete monitoring indices of quality-related and process related are built, and the quality variables are predicted while exploiting the regression structure for quality and process monitoring. Finally, a nonlinear numerical process and the penicillin fermentation process are used to verify the effectiveness of the proposed algorithm.

1. Introduction

As one of the most important production modes in the modern process industry, batch process is widely used in many fields, such as biopharmaceutical, fine chemical, food processing, and semiconductor production due to its advantages of high-value, multi-variety, and low-volume production [1]. Unlike continuous process that runs in a steady state, the batch process has the characteristics of dynamic, diversity, uncertainty, and quality variables are difficult to be measured in real-time, which greatly increases the difficulty of fault detection and diagnosis for the batch process [2]. However, timely fault detection and accurate diagnosis of faults have become important parts of the safe and smooth operation of the production process [3]. Therefore, it is urgent to establish a process monitoring system. In the past few decades, many statistical methods, such as principal component analysis (PCA) and partial least squares (PLS), have been developed for process monitoring [4,5]. For the extensive multiphase characteristic of the batch process, if we build a global model, missing alarms and false alarms may occur inevitably.

For the multiphase characteristic of the batch process, it is desirable to establish multiphase models that capture the inherent behaviors [6, 7]. Each phase has specific local behaviors, multiphase modeling is beneficial to the representation of process characteristics and can improve the process monitoring effect. Lv et al. [8] proposed a multiphase partition algorithm to monitor batch process online, in which Euclidean radius and k-means were used to determine the phase number and phase data set, however, the retained components could not be determined substantiated. To consider the varying multiphase characteristic of batch processes, Liu et al. [9]proposed a sequential local-based Gaussian mixture model, but the nonlinearity of the batch process could not be solved effectively. To characterize the dynamics of variable-wise and batch-wise batch processes, Zhu et al. [10] proposed a batch process monitoring method based on a multiphase two-dimensional time-slice dynamic system. However, the above methods still have the following problems: first, during the phase partition, there is need to identify the phases by clustering methods and complex post-processing is required. Second, each search for the optimal shard point requires to traversing all possible shard points, which have a high time complexity of order squared with sequence length. Third, the final product quality of batch process does not take into account when dividing the phases, because reasonable phase division can effectively improve the effect of batch process monitoring and product quality

* Corresponding author. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, China.
E-mail address: huiyy@lut.edu.cn (J. Cao).

https://doi.org/10.1016/j.chemolab.2023.104922
Received 11 April 2023; Received in revised form 16 June 2023; Accepted 24 July 2023
Available online 25 July 2023

prediction.

In some cases, the appropriate fluctuations in process variables do not affect product quality, so it shouldn't trigger the disturbance alarm. However, some unsupervised methods only consider the abnormal in the whole process space, the output variables are not used. The supervised methods utilize the quality data to extract the related information and it is suitable for quality-related process monitoring [11]. PLS is a typical supervised method that considers the quality variables when conducting information extraction. Zhang et al. [12] proposed a multi-scale Kernel PLS method in nonlinear process monitoring, this method could describe not only the global relations of the whole scale but also the local features within each scale. Botre et al. [13] proposed Kernel PLS-based GLRT method for quality-related process monitoring. However, the use of the kernel method to deal with nonlinearity would make the computational complexity rise sharply, especially for batch processes, it would lead to the deterioration of real-time process monitoring. For batch process monitoring, Luo et al. [14] proposed a multilinear PLS method, which avoided the batch data unfolding and had better modeling accuracy, but the determination of parameters needs further study. To detect the output unrelated and related faults, the total PLS was proposed to decompose the input space into four subspaces [15], but it could not monitor the portion of the quality variations unpredictable from inputs. To decompose the input data into the input-related and output-related spaces, Qin et al. [16] proposed a concurrent projection to latent structures for process-related and quality-related fault detection, however, the nonlinear characteristic of the process was not considered. For the nonlinear and multiple subspace monitoring, Zhang et al. [17] proposed a nonlinear CPLS method. To consider the dynamic and nonlinear characteristics of the industrial processes, Zhang et al. [18] proposed the Recurrent Kalman Variational Autoencoder method for process modeling and fault detection, it showed a remarkable performance in many hard-to-detect faults. Considering the variable importance in the projection, Yang et al. [19] proposed a new dynamic concurrent partial least square method for quality-related monitoring. However, For the nonlinear and multiphase quality-related process monitoring, it is desirable to divide the process phase reasonably according to the local quality-related characteristics and time sequence. Each phase has a specific local nonlinear characteristic, it needs to extract nonlinear features that capture the inherent behaviors. The statistics that can effectively improve the effect of batch process monitoring should be established.

In this work, a quality-related nonlinear multi-phase batch process monitoring and quality prediction algorithm is presented. To divide the multiphase reasonably, a step-wise sequential quality-related phase division model is developed. Moreover, locally weighted projection regression is used to monitor each local model, the global approximation results are obtained by weighting all the local models in the divided phases. Then, the complete monitoring indices of quality-related and process related are built, and the process-related space is decomposed into quality-related and unrelated subspaces. Besides, the quality variables are predicted while exploiting the regression structure for quality and process monitoring.

The contributions of the proposed fault detection and quality prediction method are as follows.

(1) It provides a step-wise sequential quality-related phase division model according to local quality-related characteristics and time sequence.
(2) A nonlinear phase is modeled with global approximation weighted local models.
(3) The complete monitoring indices of quality-related and process related are built, and the quality variables are predicted while exploiting the regression structure for quality and process monitoring.

2. Partial least squares (PLS)

PLS algorithm [20]gives a pair of input and output data matrices X and Y, where X represents the input variables and Y is the output variables, respectively, the details are shown in equation (1).

$$X = \begin{bmatrix} x_1^{\mathrm{T}} \\ \vdots \\ x_N^{\mathrm{T}} \end{bmatrix} \in R^{N\times m} \quad Y = \begin{bmatrix} y_1^{\mathrm{T}} \\ \vdots \\ y_N^{\mathrm{T}} \end{bmatrix} \in R^{N\times l} \tag{1}$$

where, N is the number of samples, m and l are the numbers of input variables and output variables respectively. The original variable space is projected into an uncorrelated latent factor space as equation (2).

$$T = [t_1 \cdots t_A] \in R^{N\times A} \tag{2}$$

where, T is the score matrix, because the score matrix T cannot be obtained from the input X, there need to introduce the weight $H = W(P^T W)^{-1}$, T can be expressed by equation (3).

$$T = XH \tag{3}$$

Based on the above transformation, the original variable X can be divided into principal subspace $\widehat{X}$ and residual subspace $\widetilde{X}$. Y is divided into predictable parts $\widehat{Y}$ and unpredictable parts $\widetilde{Y}$. They can be expressed by equation (4).

$$\begin{cases} X = \widehat{X} + \widetilde{X} = TP^{\mathrm{T}} + \widetilde{X} \\ Y = \widehat{Y} + \widetilde{Y} = TQ^{\mathrm{T}} + \widetilde{Y} \end{cases} \tag{4}$$

where, $P \in R^{m\times A}$ and $Q \in R^{l\times A}$ are the loading matrixes of X and Y. A is the number of principles, it can be obtained by cross-validation. The nonlinear iterative partial least squares (NIPALS) algorithm is used to calculate each parameter of PLS. In NIPALS iteration process, the eigenvectors of Y are used to regression the eigenvectors of X. And the eigenvectors of X are used to regress Y. Finally, the algorithm gradually increases the high correlation convergence by rotating the eigenvectors to obtain the model parameters. Although the method of solving model parameters by standard PLS has the problem of a large amount of calculation, its biggest disadvantage is that it uses skew decomposition for process variable X, which leads to the loss of quality-related information, thus resulting in a low-quality-related fault detection rate and affecting the performance of its process monitoring.

3 Multiphase Multiway Concurrent Locally Weighted Projection Regression (Multiphase-MCLWPR) for process monitoring and quality prediction.

For the nonlinear and multiphase characteristics of the batch process, the entire process is divided into the phases according to the local quality-related characteristic and time sequence. Then, for the specific local nonlinear characteristic in each phase, the nonlinear models are established to capture the inherent behaviors. Monitoring space is decomposed into quality-related and process related, and obtained complete monitoring scheme.

2.1. Data preprocessing and step-wise sequential quality-related phase division

To describe the time sequence multiphase characteristic and highlight the local quality-related features, the entire process is partitioned into several phases according to local quality-related characteristics and time sequences [21,22].

The batch process three-dimensional data matrix $X(I\times J\times K)$ is unfolded as $\mathrm{X}(I\times JK)$ along the batch direction. Then, the time slice matrix $\mathrm{X}_k(I\times J)$ is normalized by equation (5).

$$\overline{X}_k = \frac{\mathrm{X}_k(I\times J) - \mathrm{mean}(\mathrm{X}_k(I\times J))}{\mathrm{std}(\mathrm{X}_k(I\times J))} \tag{5}$$

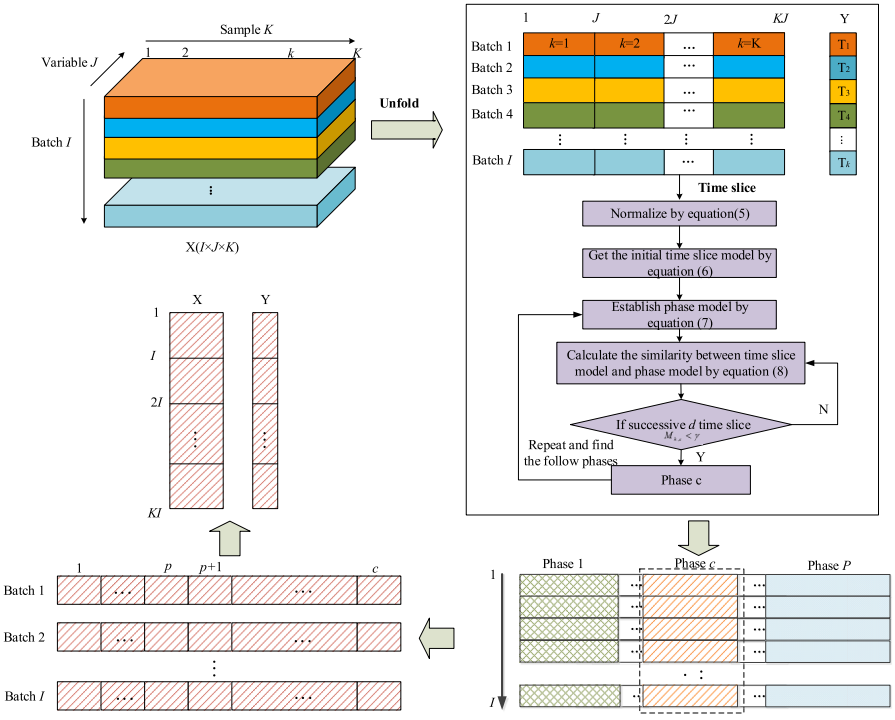


Fig. 1. The flow chart of data preprocessing and step-wise sequential quality-related phase division.

where, mean($X_k(I \times J)$) and std($X_k(I \times J)$) are the mean and standard deviations of $X_k(I \times J)$.

For the normalized time slice matrix $\overline{X}_k(I \times J)$, PLS is used to extract the quality-related features and get the initial time slice model as equation (6).

$$G_k = Q_k \overline{X}_k^T \tag{6}$$

where, Q_k is the loading matrix of Y that can be obtained by equation (4).

The first h consecutive time slices constitute the matrix $\overline{X}_c(Ih \times J)$, PLS is used to calculate the loading matrix $\widetilde{Q}_c$ based on $\overline{X}_c(Ih \times J)$, the phase model shown in equation (7).

$$G_c = \widetilde{Q}_c \overline{X}_c^T \tag{7}$$

Then, the similarity between the kth ($k = h+1, h+2, \cdots$) time slice model Q_k and the phase mode $\widetilde{Q}_c$ can be evaluated by equation (8).

$$M_{k,c} = \exp\left(-\frac{\|Q_k - \widetilde{Q}_c\|}{2J} \right) \tag{8}$$

Where, J is the number of variables, h is the shortest duration time of the phase.

When the successive d time slice from time k_c that $M_{k,c}$ lower than control limit γ, it is considered that a phase is formed. If d consecutive $M_{k,c}$ are higher than the control limit γ, it means that the quality-related features are changed and move on to the next phase. When the first phase is determined, the remaining time slice data are updated and used as the new input in equation (7), then equations (7) and (8) are repeated to find the following phases. For the divided phases, it is necessary to extract the key information reflecting the running state of the process for process monitoring and quality variable prediction. The flow chart of data preprocessing and step-wise sequential quality-related phase division is shown in Fig. 1.

2.2. Concurrent locally weighted projection regression algorithm

In each divided phase, the phase-related data generally covers high nonlinearity. For nonlinear function approximation problems, Vijayakumar et al. [23] proposed a locally weighted projections regression (LWPR) algorithm. To ensure the high efficiency of the algorithm, LWPR uses a univariate PLS regression model in each phase, and the global approximation results are obtained by the weight of all local models.

For the nonlinear function $y_i = f(x_i) + e$, f is a nonlinear function, x_i and y_i are the input and output respectively, and e is the noise of zero means. LWPR approximates the nonlinear function f by using several local linear models. To measure the locality of the data point, a Gaussian kernel is used to calculate the weight w, as shown in equation (9).

$$w_i = \exp\left(-\frac{(x_i - x_c)^T D (x_i - x_c)}{2} \right) \tag{9}$$

where, x is the data point, x_c is the data center of the local model, D is a positive semidefinite matrix used to determine the size and shape of local model. Then, the weighted mean of the current local model is:

$$\overline{x} = \sum_{i=1}^{N} w_i x_i \Big/ \sum_{i=1}^{N} w_i, \overline{y} = \sum_{i=1}^{N} w_i y_i \Big/ \sum_{i=1}^{N} w_i \tag{10}$$

The main task of LWPR in nonlinear function approximation is to determine the number of local models, the regression model is calculated for each local model and the local model's receptive field (RF) is adjusted automatically. The predicted output $\widehat{y}_q$ of LWPR is normalized as the weighted mean of all the predicted outputs $\widehat{y}_k$, as shown in equation (11).

$$\widehat{y}_q = \sum_{k=1}^{K} w_k \widehat{y}_k \Big/ \sum_{k=1}^{K} w_k \tag{11}$$

For each local model, locally weighted PLS is used to determine the parameters. The details of LWPR can be found in literature [23,24]. When the first training data sample (x_1, y_1) is entered, the new RF would be created. Then, the Gaussian kernel weights would be calculated for all

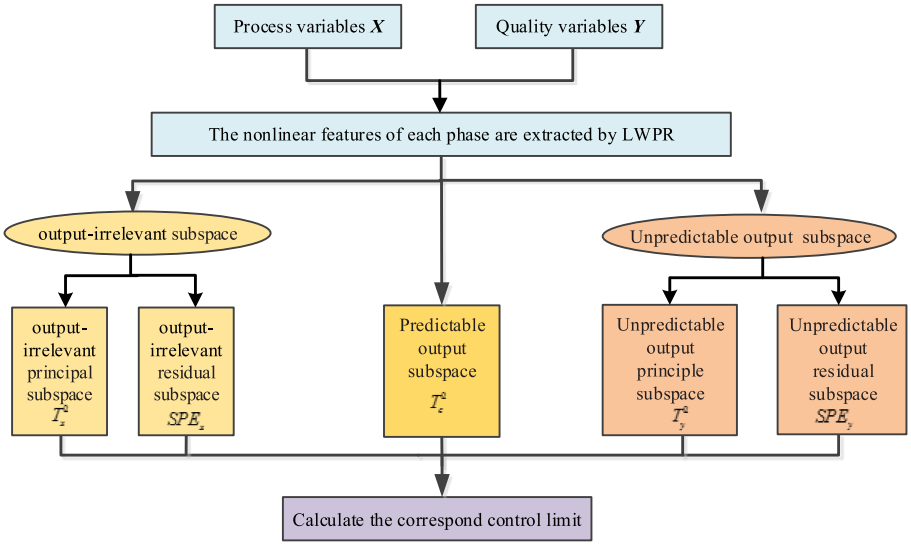


Fig. 2. MCLWPR process monitoring flow chart.

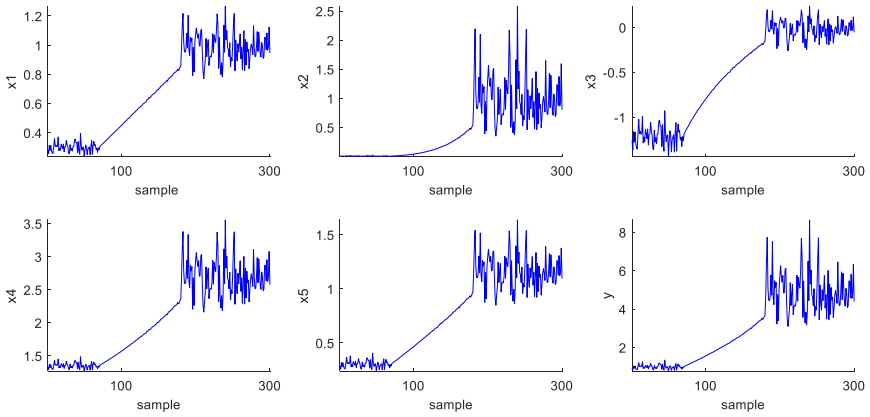


Fig. 3. Trajectories of six numerical process variables under a normal batch.

the existing RFs. Afterward, each local model is updated by the sample with its weight. LWPR algorithm with locally linear models is obtained.

A complex nonlinear problem is turned into a weighted combination of a few simple linear problems. After the normal sample $(x_n, y_n)(n = 1, \cdots, N)$ is trained by LWPR, the LWPR model has K local linear models RF_k $(k = 1, \cdots, K)$. To ensure the inputs and outputs have zero means, the weighted mean $\bar{x}_k, \bar{y}_k$ from equation (10) should be subtracted from the training data samples (x_n, y_n), as shown in equation (12).

$$\begin{aligned} x_{mz,n} &= x_n - \bar{x}_k \\ y_{mz,n} &= y_n - \bar{y}_k \end{aligned} \tag{12}$$

Thus, the input and output matrices are $X_k = [x_{mz,1}, \cdots, x_{mz,N_k}]^{\mathrm{T}}$ and $Y_k = [y_{mz,1}, \cdots, y_{mz,N_k}]^{\mathrm{T}}$, respectively. The load vectors are p_r and q_r, weight vector u_r denotes the projection from the deflated input matrix to the latent score vector and would be updated from each training sample. After training, the loading matrix and weight matrix can be expressed as $P_k = [p_1, \cdots, p_A]$, $Q_k = [q_1, \cdots, q_A]$ and $U_k = [u_1, \cdots, u_A]$.

The score matrix T_k cannot be directly calculated from X_k, there constructed intermediate matrix Rj, as shown in (13).

$$\begin{aligned} r_1 &= u_1 \\ r_2 &= \left(I - u_1 p_1^T\right) u_2 \\ &\vdots \\ r_A &= \left(I - u_1 p_1^T\right)\left(I - u_2 p_2^T\right) \cdots \left(I - u_{A-1} p_{A-1}^T\right) u_A \end{aligned} \tag{13}$$

We can obtain the matrix $R_k = [r_1, r_2, \cdots, r_A]$, T_k is calculated by equation (14).

$$T_k = X_k R_k \tag{14}$$

To further refine the statistical analysis space, and achieve a complete monitoring scheme of quality-related data and process-related data, the monitoring statistics are defined as predictable output subspace, unpredictable output residual subspace and unpredictable output principle subspace, output irrelevant principal subspace and output irrelevant residual subspace [16].

From equation (14) and the PLS model, we can obtain $\widehat{Y}_k = T_k Q_k$ and perform singular value decomposition (SVD).

$$\widehat{Y}_k = T_k Q_k = U_k^c D_k^c \left(V_k^c\right)^{\mathrm{T}} = U_k^c \left(Q_k^c\right)^{\mathrm{T}} \tag{15}$$

where, $Q_k^c = V_k^c D_k^c$, V_k^c is orthonormal, U_k^c can be expressed as:

$$U_k^c = \widehat{Y}_k V_k^c \left(D_k^c\right)^{-1} = X_k R_k Q_k V_k^c \left(D_k^c\right)^{-1} = X_k R_k^c \tag{16}$$

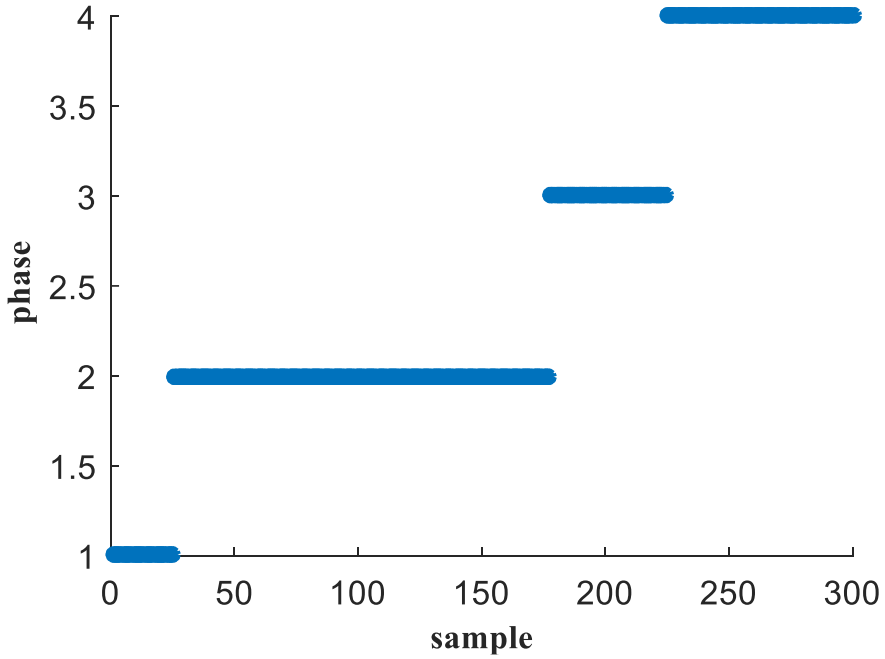


Fig. 4. Numerical phase partition result.

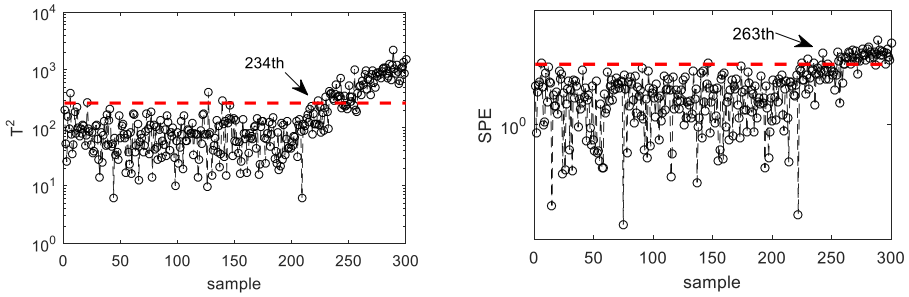


Fig. 5. Monitoring chart of MPLS.

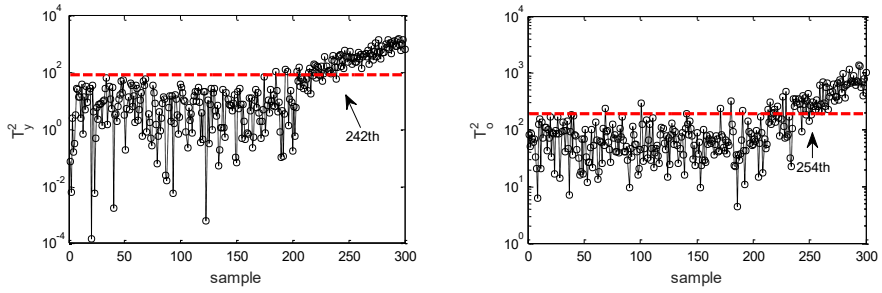


Fig. 6. Monitoring chart of MTPCR.

where, $R_k^c = R_k Q_k V_k^c (D_k^c)^{-1}$,the unpredictable output $\widetilde{Y}_k^c = Y_k - U_k^c (Q_k^c)^{\mathrm{T}}$ and PCA with l_y principle components are performed.

$$\widetilde{Y}_k^c = T_k^y (P_k^y)^{\mathrm{T}} + \widetilde{Y}_k \tag{17}$$

where, T_k^y is unpredictable output principle score, $\widetilde{Y}_k$ is unpredictable output residual.

The output-irrelevant subspace is represented as $\widetilde{X}_k^c = X_k - U_k^c M_k^c$, $M_k^c = ((R_k^c)^{\mathrm{T}} R_k^c)^{-1} (R_k^c)^{\mathrm{T}}$ and PCA with l_x principle components is performed, as shown in equation (18).

$$\widetilde{X}_k^c = T_k^x (P_k^x)^{\mathrm{T}} + \widetilde{X}_k \tag{18}$$

where, T_k^x is the input principal score, $\widetilde{X}_k$ is input residual. The input data matrix X_k and output data matrix Y can be decomposed as equation (19) by the MCLWPR model.

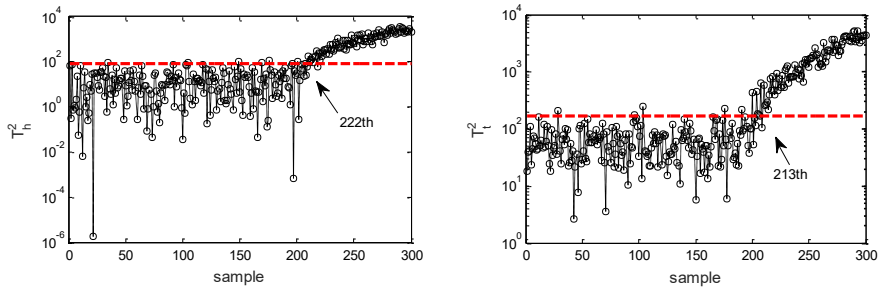


Fig. 7. Monitoring chart of MMPLS.

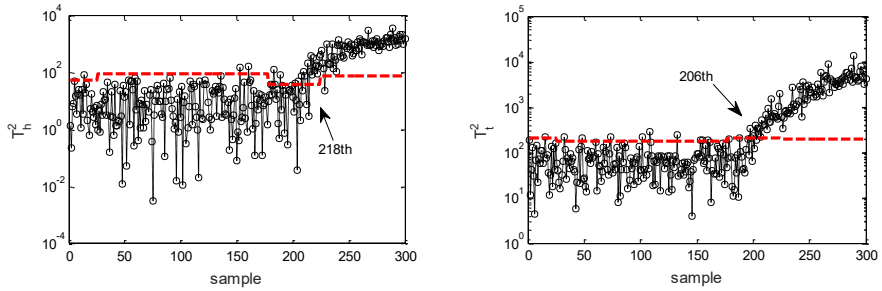


Fig. 8. Monitoring chart of multiphase-MEMPRM.

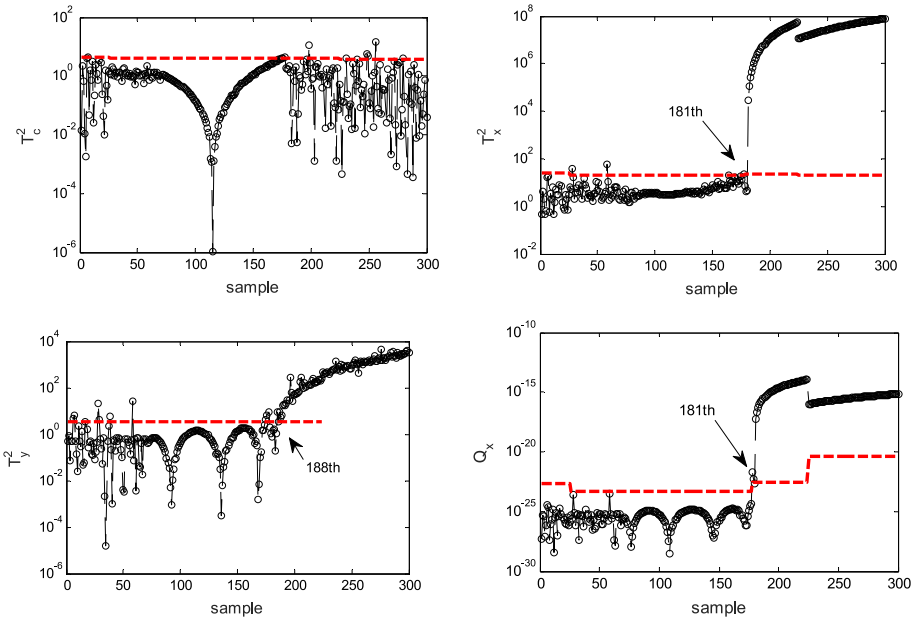


Fig. 9. Monitoring chart of multiphase-MCLWPR.

$$X_k = U_k^c M_k^c + T_k^x \left(P_k^x\right)^{\mathrm{T}} + \widetilde{X}_k$$
$$Y_k = U_k^c Q_k^c + T_k^y (P_k^y)^{\mathrm{T}} + \widetilde{Y}_k \tag{19}$$

In MCLWPR model, M_k^c, P_k^x, Q_k^c and P_k^y are the loading matrixes, U_k^c is the scores matrix that denotes the covariations in X_k associated with the predictable part $\widehat{Y}$. T_k^x denotes the variations in X_k that is not relevant to predicting Y_k, and T_k^y represents the variations in Y_k that X_k cannot predict.

2.3. MCLWPR for process monitoring

From the MCLWPR model, it needs to design the monitoring statistics based on equations 15–19. The output-related scores are orthonormalized, hence, the monitoring statistics of predictable output subspace can be followed T^2 statistic, as shown in equation (20).

$$\left(T_k^c\right)^2 = (n-1)\left(u_k^c\right)^{\mathrm{T}} u_k^c = (n-1) x_{mz}^{\mathrm{T}} R_k^c \left(R_k^c\right)^{\mathrm{T}} x_{mz} \tag{20}$$

where, x_{mz} is the zero mean of input x, which can be obtained by equation (12).

The monitoring statistics of output-irrelevant principal subspace and

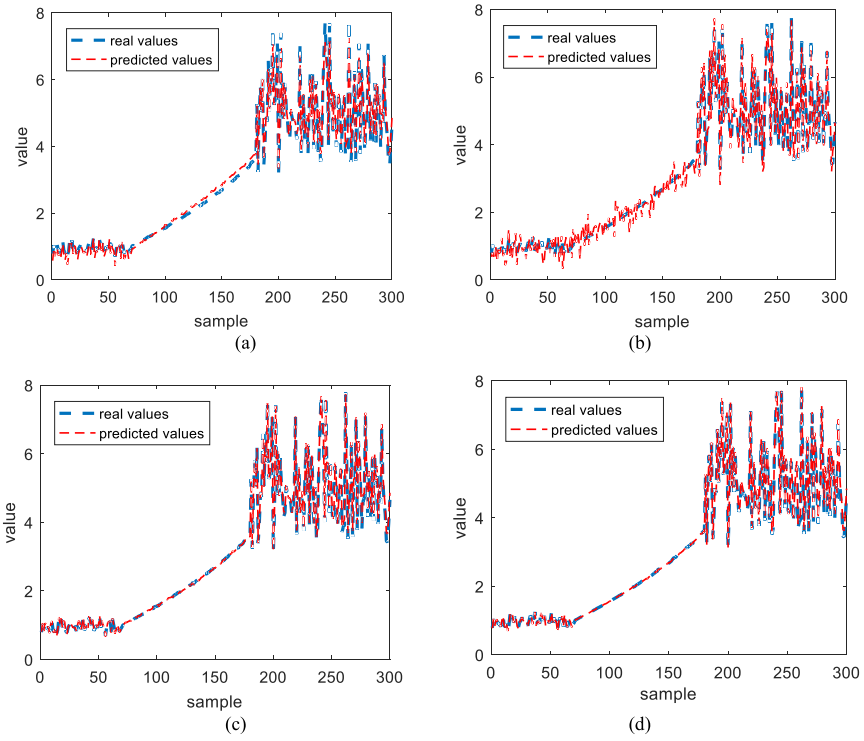


Fig. 10. The quality variable prediction results: (a) MPLS, (b)MTPCR, (c)multiphase-MEMPRE, (d)multiphase-MCLWPR.

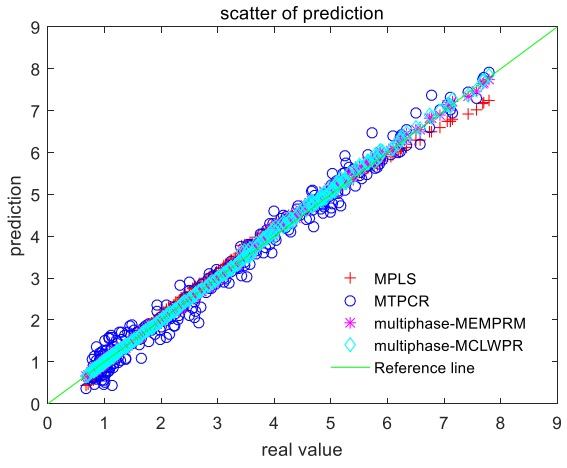


Fig. 11. The scatters of prediction.

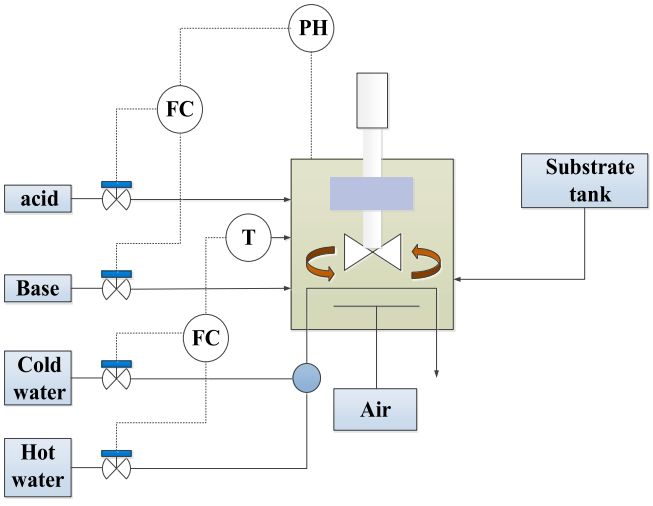


Fig. 12. Penicillin fermentation process.

Table 1
RMSE of different methods.

Methods	MPLS	MTPCR	Multiphase–MEMPRM	Multiphase-MCLWPR
RMSE	0.1637	0.2360	0.0730	0.0466

output-irrelevant residual subspace can be followed T^2 statistics and SPE statistics, as shown in equations 21 and 22.

$$\left(T_k^x\right)^2 = \left(t_k^x\right)^{\mathrm{T}} \left(\Lambda_k^x\right)^{-1} t_k^x = \left(\widetilde{x}_k^c\right)^{\mathrm{T}} P_k^x \left(\Lambda_k^x\right)^{-1} \left(P_k^x\right)^{\mathrm{T}} \widetilde{x}_k^c \tag{21}$$

$$\mathrm{SPE}_k^x = \left\|\widetilde{x}_k\right\|^2 = \left(\widetilde{x}_k^c\right)^{\mathrm{T}} \left(I - P_k^x \left(P_k^x\right)^{\mathrm{T}}\right) \widetilde{x}_k^c \tag{22}$$

where, t_k^x is the row vector of T_k^x, $\widetilde{x}_k^c = x_{mz} - M_k^c u_k^c$, u_k^c is the row vector of U_k^c, $\Lambda_k^x = \frac{1}{n-1}(T_k^x)^{\mathrm{T}} T_k^x$.

The monitoring statistics of unpredictable output principal subspace and unpredictable output residual subspace can be followed T^2 statistics and SPE statistics, as shown in equations 23 and 24.

$$\left(T_k^y\right)^2 = \left(t_k^y\right)^{\mathrm{T}} \left(\Lambda_k^y\right)^{-1} t_k^y = \left(\widetilde{y}_k^c\right)^{\mathrm{T}} P_k^x \left(\Lambda_k^x\right)^{-1} \left(P_k^x\right)^{\mathrm{T}} \widetilde{y}_k^c \tag{23}$$

$$\mathrm{SPE}_k^y = \left\|\widetilde{y}_k\right\|^2 = \left(\widetilde{y}_k^c\right)^{\mathrm{T}} \left(I - P_k^x \left(P_k^x\right)^{\mathrm{T}}\right) \widetilde{y}_k^c \tag{24}$$

where, t_k^y is the row vector of T_k^y, $\widetilde{y}_k^c = y_{mz} - Q_k^c u_k^c$, $\Lambda_k^y = \frac{1}{n-1}(T_k^y)^{\mathrm{T}} T_k^y$, y_{mz} is the zero mean of input y that can be obtained by equation (12).

The global statistics of MCLWPR for each data point (x_k, y_k) can be calculated as equations 25–29.

Table 2
Process variables.

Variable number	Process Variable	Unit
1	Aeration	L/h
2	Agitator	W
3	Substrate feed flow rate	L/h
4	Substrate feed temperature	K
5	Substrate concentration	g/L
6	O_2	%
7	Penicillin concentration	%
8	Culture volume	L
9	CO_2	g/L
10	PH	1
11	Temperature	K

$$T_c^2 = \sum_{k=1}^{K} w_k \left(T_k^c\right)^2 \Big/ \sum_{k=1}^{K} w_k \tag{25}$$

$$T_x^2 = \sum_{k=1}^{K} w_k \left(T_k^x\right)^2 \Big/ \sum_{k=1}^{K} w_k \tag{26}$$

$$SPE_x = \sum_{k=1}^{K} w_k SPE_k^x \Big/ \sum_{k=1}^{K} w_k \tag{27}$$

$$T_y^2 = \sum_{k=1}^{K} w_k \left(T_k^y\right)^2 \Big/ \sum_{k=1}^{K} w_k \tag{28}$$

$$SPE_y = \sum_{k=1}^{K} w_k SPE_k^y \Big/ \sum_{k=1}^{K} w_k \tag{29}$$

To monitor the above indexes, the control limits of corresponding statistics should be calculated from the statistics of the normal data. Kernel density estimation (KDE) [25] is used to evaluate the corresponding control limit. KDE is a non-parametric method that estimates the probability density function of a random variable, it does not need to

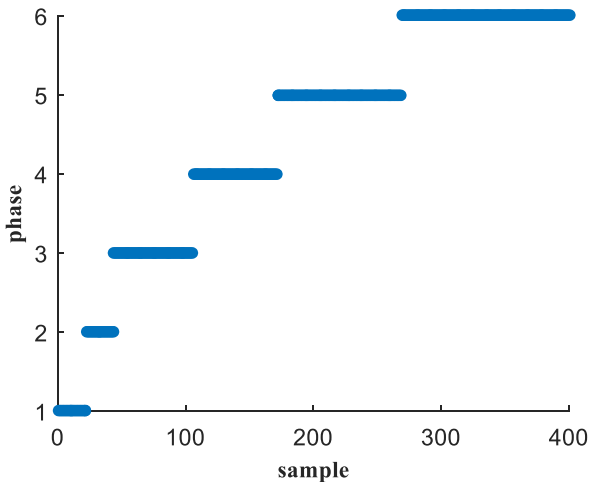


Fig. 14. Penicillin fermentation process phase partition results.

Table 3
Fault batches in the penicillin fermentation process.

Fault No.	Variable name	Fault type	Amplitude	Occurrence time/h	End time/h
1	Aeration rate	step	4	100	400
2	Aeration rate	ramp	0.5	100	400
3	Agitator rate	step	5	100	300
4	Agitator rate	ramp	0.7	200	400
5	Substrate feeding rate	step	1.8	200	400
6	Substrate feeding rate	ramp	0.007	200	400

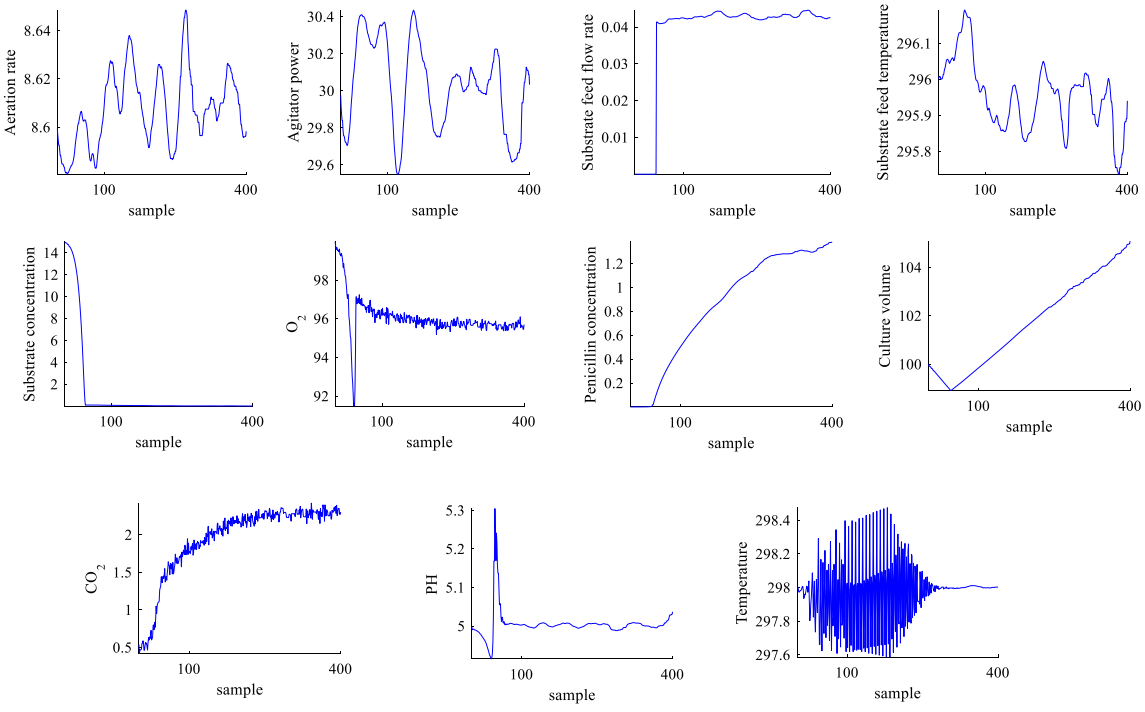


Fig. 13. Trajectories of 11 penicillin fermentation process variables under a normal batch.

Table 4
Fault detection rate (FDR).

Fault No.	MPLS		MTPCR		MMPLS		Multiphase-MEMPRM		Multiphase-MCLWPR			
	T^2	SPE	T_y^2	T_o^2	T_h^2	T_t^2	T_h^2	T_t^2	T_c^2	T_x^2	T_y^2	Q_x
1	0.6412	0.0266	0.0365	0.7176	0.9535	0.1728	0.5482	1	0	1	0.0664	0.9900
2	0.6080	0.6445	0.5349	0.6844	0.8040	0.6977	0.0764	0.4452	0	0.9900	0.0299	0.7608
3	1	1	0.0149	1	0.4023	1	0.6020	1	0.0746	1	0.8010	0.4826
4	0.4328	0.5423	0.5224	0.5771	0.6219	0.6517	0.7413	0.2388	0.1791	0.8955	0.7363	0.2338
5	0.0050	0	0.0050	0.0050	0.0945	0.0199	0.3035	0	0.0398	0.6119	0.2438	0.5572
6	0.4229	0.5970	0.4627	0.5473	0.4726	0.5970	0.6219	0.2935	0.1244	0.9403	0.8806	0.8458

Table 5
False alarm rate (FAR).

Fault No.	MPLS		MTPCR		MMPLS		Multiphase-MEMPRM		Multiphase-MCLWPR			
	T^2	SPE	T_y^2	T_o^2	T_h^2	T_t^2	T_h^2	T_t^2	T_c^2	T_x^2	T_y^2	Q_x
1	0.1100	0.0100	0.0400	0.0900	0	0.1200	0.0200	0.0400	0.0300	0	0.0200	0.0560
2	0	0.0600	0.0200	0.0300	0.0800	0.0600	0	0.0450	0.0200	0.0100	0.0200	0.0330
3	0.0250	0.0100	0.0050	0.0250	0.0350	0.0350	0.0050	0	0.0050	0.0350	0.0400	0
4	0.0350	0.0150	0.0350	0.0450	0.0700	0.1050	0.3250	0.0050	0.0100	0.0200	0.0600	0
5	0.0400	0.0350	0.1100	0.0250	0.0850	0.0650	0.2750	0.0050	0.0100	0.0300	0.0650	0.0400
6	0.0150	0.0850	0.0050	0.0550	0.0200	0.1150	0.2050	0	0.0100	0.0300	0.0650	0.0400

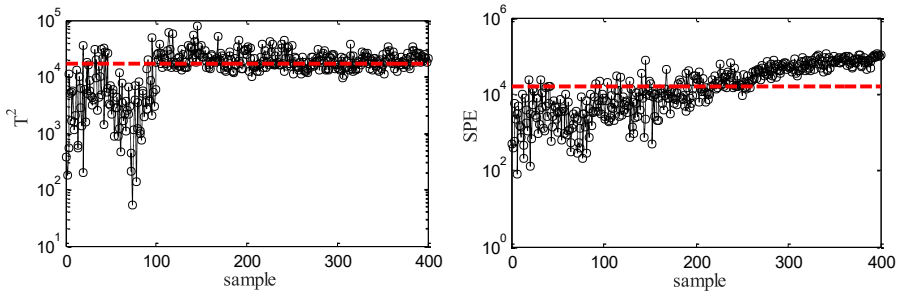


Fig. 15. Monitoring chart of MPLS under Fault 1.

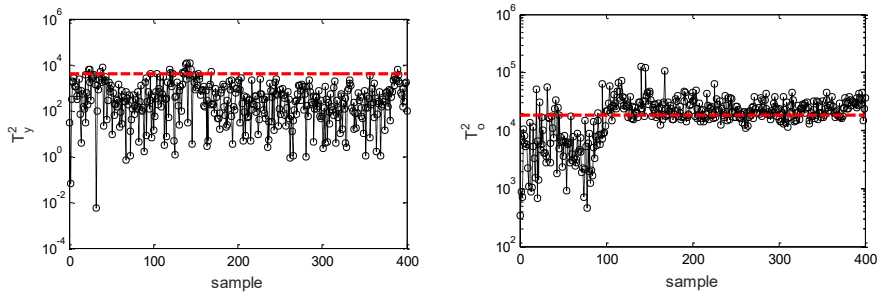


Fig. 16. Monitoring chart of MTPCR under Fault 1.

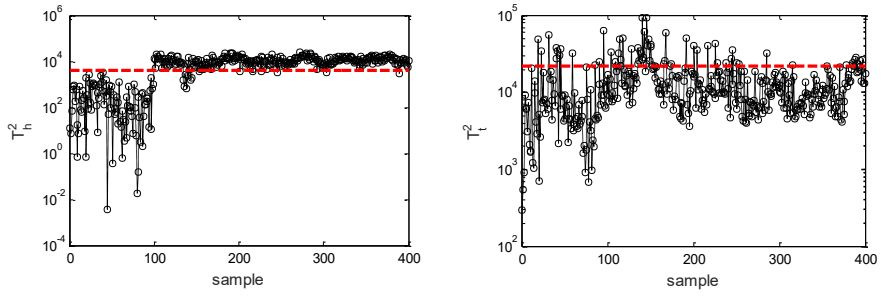


Fig. 17. Monitoring chart of MMPLS under Fault 1.

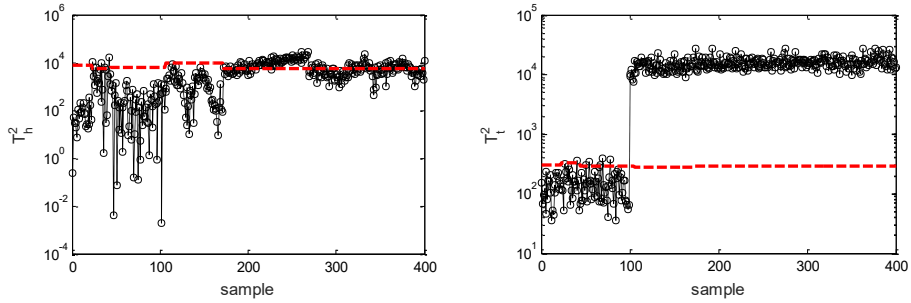


Fig. 18. Monitoring chart of Multiphase-MEMPRM under Fault 1.

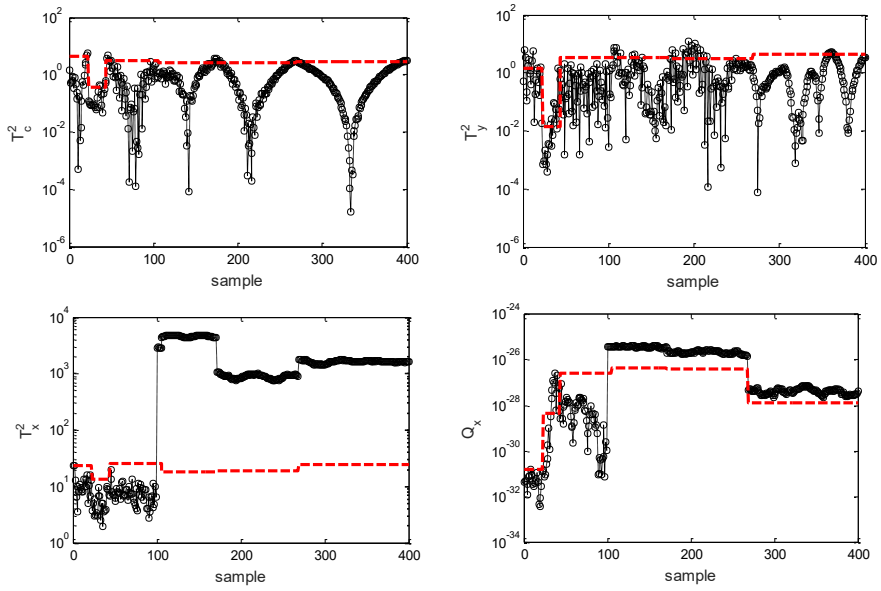


Fig. 19. Monitoring chart of Multiphase-MCLWPR under Fault 1.

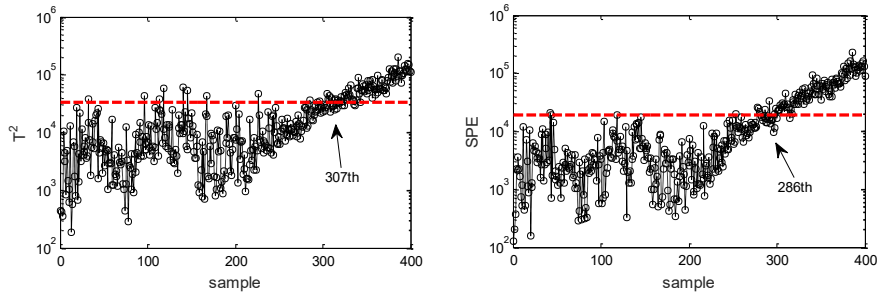


Fig. 20. Monitoring charts of MPLS under Fault 4.

follow Gaussian distribution. The MCLWPR process monitoring flow chart is shown in Fig. 2.

2.4. Quality prediction

Because multiphase MCLWPR has a strong capacity to extract the multiphase batch process nonlinear complex features, it can integrate the phase features and predict the quality variables. For the K local linear models RF_k $(k=1,\cdots,K)$ of MCLWPR, the test sample $(x_n^t, y_n^t)(n=1,\cdots,N)$ should be normalized as $(x_{mz,n}^t, y_{mz,n}^t)(n=1,\cdots,N)$ by equation (10), then, the predicted value can be obtained by equation (30).

$$\hat{y}_k = t_k q_k = x_{mz,n}^t r_k q_k \tag{30}$$

The global predicted value of MCLWPR is the normalized weighted mean of all the predicted outputs in equation (31).

$$\hat{y} = \sum_{k=1}^{K} w_k \hat{y}_k \Big/ \sum_{k=1}^{K} w_k \tag{31}$$

The prediction accuracy is used to evaluate the performance of the method. Root mean squared error (RMSE) is usually used to evaluate the prediction accuracy. A smaller RMSE value indicates that the prediction accuracy is higher. RMSE is defined as:

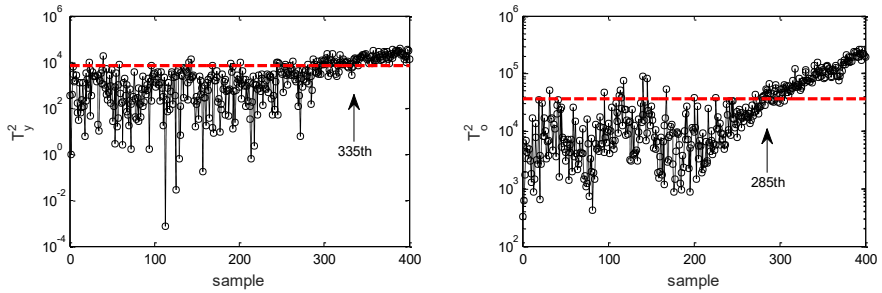


Fig. 21. Monitoring charts of MTPCR under Fault 4.

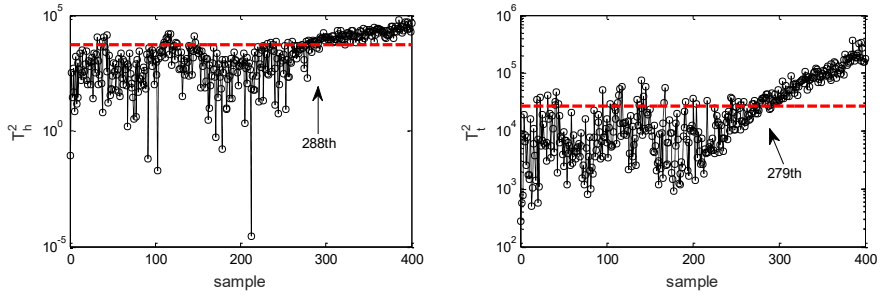


Fig. 22. Monitoring charts of MMPLS under Fault 4.

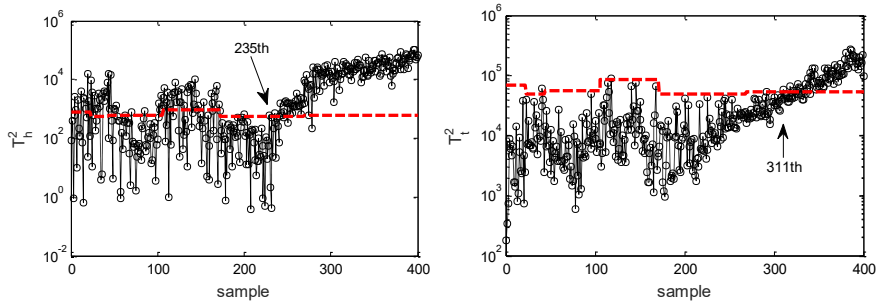


Fig. 23. Monitoring charts of Multiphase-MEMPRM under Fault 4.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}(y_i - \hat{y}_i)^2} \tag{32}$$

where, y_i and $\hat{y}_i$ are the actual value and predicted value. $\bar{y}$ is the mean value of actual value y_i, N_t is the length of testing sample.

3. Monitoring and quality prediction steps

3.1. Offline modeling

Step 1: Collect the normal batch process data and normalize the data.
Step 2: Step-wise sequential quality-related phase division.
Step 3: Unfold the three-dimension data in each phase and build the MCLWPR model.
Step 4: Refine the monitoring statistics and provide complete monitoring of the output variations.
Step 5: Calculate T_c^2, T_x^2, T_y^2 and Q_x statistics.
Step 6: Compute the control limits of normal data by kernel density estimation.

3.2. Online monitoring

Step 1: Standardize new batch data.
Step 2: Divide the new batch as offline phase division.
Step 3: Calculate T_c^2, T_x^2, T_x^2 and Q_x statistics for the new samples.
Step 4: Judge if T_c^2, T_x^2, T_x^2 and T_x^2 statistics exceed the control limits. If there exceeds the control limit, it indicates a fault occurs; Otherwise, return to **Step 1** for the next monitoring.

3.3. Quality prediction

The quality variables can be predicted by equation (31), and then the prediction accuracy is evaluated by equation (32).

4. Case study and discussion

To verify the process monitoring and quality prediction effectiveness of the proposed algorithm, a nonlinear multiphase numerical process and penicillin fermentation process are selected to test. For comparative analysis, the performance of the proposed algorithm is compared with MPLS, MTPCR, MMPLS, and multiphase-MEMPRM algorithms. Q_y is the unpredicted output residual statistic and it is null in most training re-

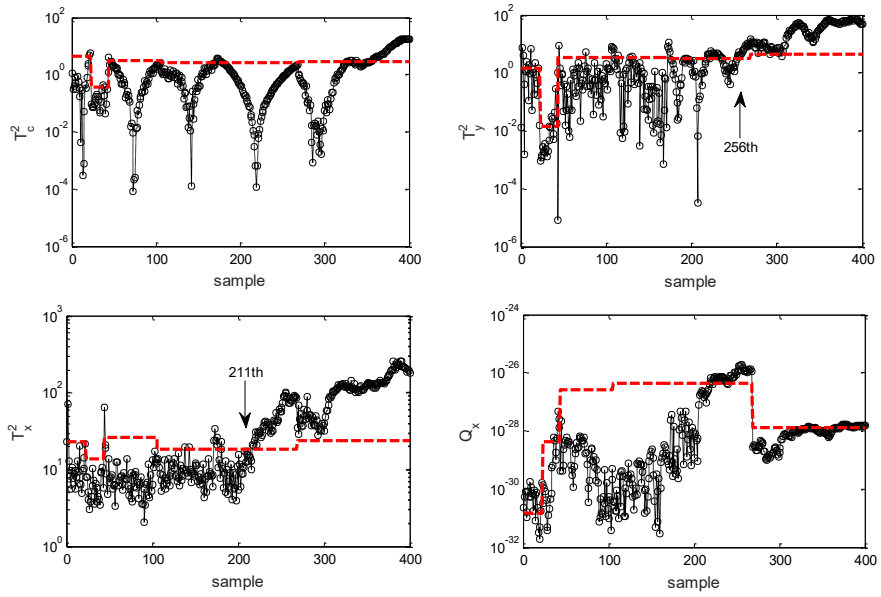


Fig. 24. Monitoring charts of Multiphase-MCLWPR under Fault 4.

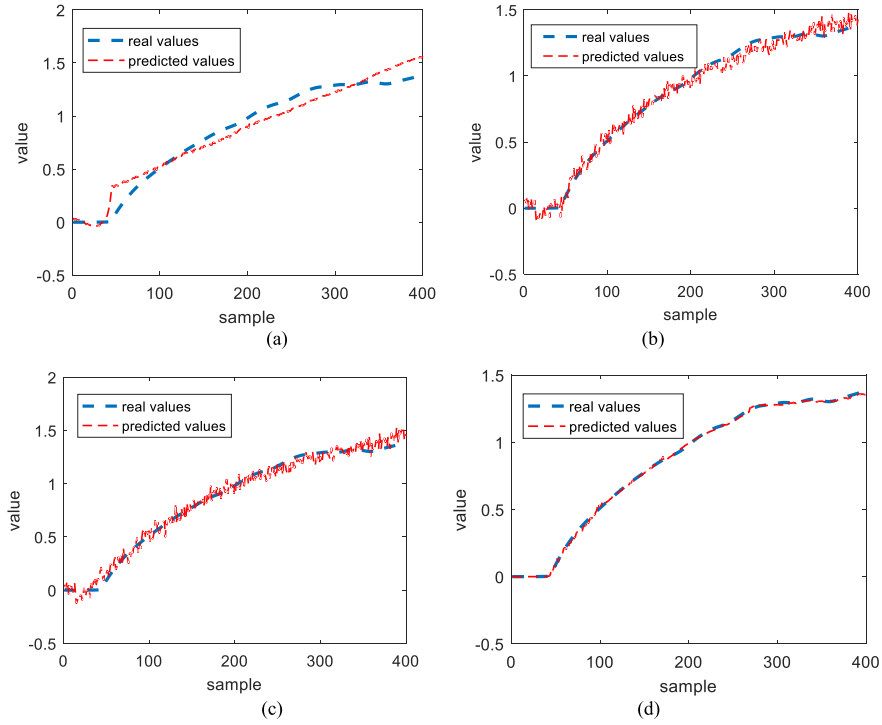


Fig. 25. Product concentration prediction results: (a) MPLS; (b) MTPCR; (c) multiphase-MEMPRE; (d) multiphase-MCLWPR.

sults. Therefore, four statistics T_c^2, T_x^2, T_y^2 and Q_x are shown in process monitoring [16,19].

4.1. Numerical system

A numerical example is generated to simulate a nonlinear dynamic multiphase batch process [26,27]. Six variables are used as monitored variables, where 300 sample points are collected for each batch (20 batches in total).

For the first 70 sample points, the training data of each sample point are generated as equations (33) and (34):

$$\begin{cases} x_{11} = normrnd(0.3, 0.03) + e_1 \\ x_{12} = x_1{}^4 + e_2 \\ x_{13} = \log(x_1) + e_3 \\ x_{14} = \exp(x_1) + e_4 \\ x_{15} = \sinh(x_1) + e_5 \\ y_{11} = x_{11} + 2x_{12}x_{13} + x_{11}x_{14} + x_{15} \end{cases} \tag{33}$$

$$X_{train1} = [x_{11}\ \ x_{12}\ \ x_{13}\ \ x_{14}\ \ x_{15}\ \ y_{11}] \tag{34}$$

From the 71st to 180th sampling points, the training data are generated as follows:

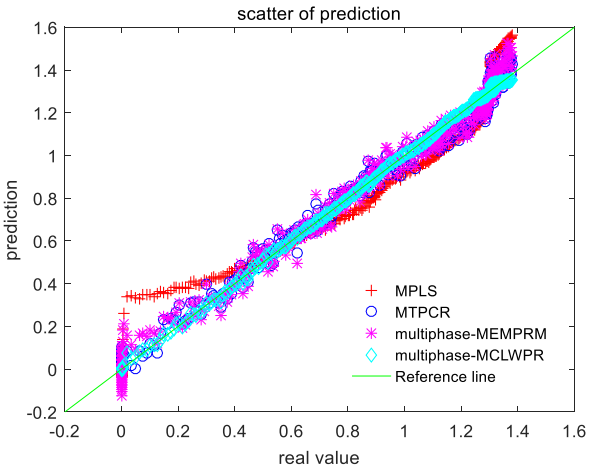


Fig. 26. The scatter chart of prediction.

Table 6
RMSEs of different methods.

Methods	MPLS	MTPCR	Multiphase-MEMPRM	Multiphase-MCLWPR
RMSE	0.1017	0.0508	0.0588	0.0074

$$\begin{cases} x_{21} = normrnd(0.3 + 0.005 * i, 0.01 * (0.03 + 0.003 * i) + e_1 \\ x_{22} = {x_{21}}^4 + e_2 \\ x_{23} = \log(x_{21}) + e_3 \\ x_{24} = \exp(x_{21}) + e_4 \\ x_{25} = \sinh(x_{21}) + e_5 \\ y_{21} = x_{21} + 2x_{22}x_{23} + x_{21}x_{24} + x_{25} \end{cases} \tag{35}$$

$$X_{train2} = [x_{21} \;\; x_{22} \;\; x_{23} \;\; x_{24} \;\; x_{25} \;\; y_{21}] \tag{36}$$

where $i = 1, 2, \cdots, 110$, i represents the *ith* sample point.

From the 181th to 300th sampling point, the training data can be obtained as follows:

$$\begin{cases} x_{31} = normrnd(1, 0.1) + e_1 \\ x_{32} = {x_{31}}^4 + e_2 \\ x_{33} = \log(x_{31}) + e_3 \\ x_{34} = \exp(x_{31}) + e_4 \\ x_{35} = \sinh(x_{31}) + e_5 \\ y_{21} = x_{31} + 2x_{32}x_{33} + x_{31}x_{34} + x_{35} \end{cases} \tag{37}$$

$$X_{train3} = [x_{31} \;\; x_{32} \;\; x_{33} \;\; x_{34} \;\; x_{35} \;\; y_{31}] \tag{38}$$

where, $[e_1, e_2, e_3, e_4, e_5]$ is white noise with a variance of 0.01. The initial values of the batches are different in the numerical process to simulate the stochastic variation between batches. The train data are $X_{train}(20 \times 6 \times 300)$. In addition, the variable x_{32} is added a $0.002\times(k-180)$ ramp fault from the 181th point to verify the effect of the proposed algorithm. A normal batch is selected to illustrate the multiphase and nonlinear characteristics of the numerical process, as shown in Fig. 3, the variables clearly exhibit the multiphase and nonlinear characteristics. Therefore, to realize effective process monitoring and quality prediction, it is necessary to divide the whole numerical process into multiphases and extract nonlinear features of the running process.

For the obtained 20 normal batches, the step-wise sequential quality-related phase division method is used to divide the phases. The phase partition is shown in Fig. 4, we can see that the whole process is divided into four phases. The divided results are inconsistent with the numerical process, this is because step-wise sequential quality-related phase division method focuses on the local quality-related model to extract data features.

The ramp fault is added in variable x_{32} from the 181th point to verify the effect of the proposed algorithm. The monitoring charts of MPLS, MTPCR, MMPLS, multiphase-MEMPRM, and the proposed algorithm for the fault batch are presented in Fig. 5-Fig. 9, respectively. In Fig. 5, the T^2 and SPE monitoring charts of MPLS detect the fault at the 234th and 263rd sample points, there is a large delay in fault detection. Fig. 6 is the monitoring chart of MTPCR, it shows that ${T_y}^2$ and ${T_o}^2$ detect the fault at the 242nd and 254th sample points, the performance is similar with MPLS. In Fig. 7, the ${T_h}^2$ and ${T_t}^2$ monitoring charts of MMPLS detect the fault at the 222nd and 213rd sample points, it can detect the fault earlier than MPLS and MTPCR. Fig. 8 is the monitoring chart of multiphase-MEMPRM, ${T_h}^2$ and ${T_t}^2$ detect the fault at the 218th and 206th sample points, the performance is better than MPLS, MTPCR and MMPLS. By contrast, the monitoring chart of multiphase-MCLWPR is shown in Fig. 9, ${T_x}^2$, ${T_y}^2$ and Q_x can detect the fault when it occurs, this indicates that the fault is a potential output-relevant fault, and the detection effect is better than MPLS, MTPCR, MMPLS, and multiphase-MEMPRM. This is because multiphase-MCLWPR can extract the time sequence multiphase characteristic and the local quality-related features effectively.

The quality variable y is used to evaluate the prediction performance. The prediction results of quality variables are shown in Fig. 10, where the red dotted line predicted by multiphase-MCLWPR coincides well with the actual value trajectory. To compare the prediction results visually, the scatters of prediction are shown in Fig. 11, we can see that the predictive values of multiphase-MCLWPR are closer to the reference line, that is to say, the proposed multiphase-MCLWPR shows a better performance than MPLS, MTPCR, and multiphase-MEMPRM. Also, RMSE values are listed in Table .1, it can be seen that the RMSE value of multiphase-MCLWPR is smaller than that of MPLS, MTPCR, and multiphase-MEMPRM. It indicates that the prediction accuracy of multiphase-MCLWPR is higher.

4.2. Penicillin fermentation process

As a typical multiphase batch process, the penicillin production process has been widely used in process monitoring and quality prediction [28]. Birol et al. [29] developed a simulation platform, which simulated the penicillin production mechanism and provided a series of process variables. Fig. 12 is the flow chart of the penicillin fermentation process. The durations of each batch are set to 400h. 20 batches under normal operating conditions are produced. We select 11 variables as monitoring variables (see Table .2).

The obtained 20 normal batches are composed the training data $X_{train}(20 \times 11 \times 400)$.A normal batch is selected to illustrate the multiphase and nonlinear characteristics of the process. As shown in Fig. 13, the variables clearly exhibit phase and nonlinear characteristics. Therefore, to realize effective process monitoring and quality prediction, it is necessary to divide the whole process into multiphase and extract nonlinear features of the running process.

The step-wise sequential quality-related phase division method is used to divide the process. The phase partition results are shown in Fig. 14, we can see that the whole process is divided into six phases. The divided results are inconsistent with the pensim process, this is because the step-wise sequential quality-related phase division method focuses on the local quality-related model to explain the data. It is more suitable for the statistical analysis of local processes.

Pensim benchmark can produce fault batches caused by variable ramp or step changes, which include aeration rate, substrate feeding rate, and agitator rate. We introduce 6 different fault batches, as shown in Table .3. The effectiveness of the proposed algorithm is verified by setting different fault magnitudes and types.

Fault Detection Rate (FDR) and False Alarm Rate (FAR) for all fault batches are quantitatively summarized in Table .4 and Table .5 to evaluate the effect of process monitoring. From Tables 4 and 5, we can see that Multiphase-MCLWPR has a higher FDR compared with MPLS,

MTPCR, MMPLS, and Multiphase-MEMPRM. The Multiphase-MCLWPR and Multiphase-MEMPRM can detect step Fault 1 without delay, but Multiphase-MCLWPR has a lower FAR. Step Fault 3 can be detected immediately by these five methods, this is because Fault 3 has a large fault amplitude, which is easy to detect. Step Fault 5 cannot be detected effectively because the glucose substrate feeding rate is diffused slowly through the correlated variables. In contrast to the step faults, ramp faults 2, 4, and 6 are more difficult to detect due to their slow variation. Compared with MPLS, MTPCR, MMPLS, and Multiphase-MEMPRM, Multiphase-MCLWPR detects a fault earlier than others, this is because Multiphase-MCLWPR considers the time sequence multiphase characteristic and extracts the local quality-related features effectively. By comprehensive analysis of FDR and FAR, Multiphase-MCLWPR can effectively distinguish abnormal and normal states.

Fault 1 is the aeration rate step change from the 100th sample point to the end. Fig. 15 is the monitoring chart of MPLS, which shows T^2 and SPE cannot detect the fault when the fault occurs, there is a serious delay. T_y^2 and T_o^2 of MTPCR are shown in Fig. 16, it also cannot detect the fault when the fault occurs. The monitoring chart of MMPLS in Fig. 17, T_h^2 can detect the fault timely, but there are several points under the control limit when the fault exists. Fig. 18 is the monitoring chart of Multiphase-MEMPRM, it shows that T_t^2 can detect the fault immediately, but there are false alarms between 30 to 80 sample points. The Multiphase-MCLWPR monitoring chart is Fig. 19, which shows that T_x^2 and Q_x can detect the fault in time when the fault occurs, and there are no false alarms. It indicates that the fault is an output-irrelevant but input-relevant fault. We can conclude that Multiphase-MCLWPR is better than MPLS, MPTCR, MMPLS, and Multiphase-MEMPRM for output-irrelevant fault.

Fault 4 is a ramp fault with a 0.7 increase in agitator rate, which is introduced from the 200th sample point to the end. The monitoring charts of MPLS, MTPCR, MMPLS, Multiphase-MEMPRM, and the proposed algorithm are presented in Fig. 20-Fig. 24, respectively. In Fig. 20, the T^2 and SPE monitoring charts of MPLS detect the fault at the 307th and 286th sample points, there is a large delay in fault detection. Fig. 21 is the monitoring chart of MTPCR, it shows that T_y^2 and T_o^2 detect the fault at the 335th and 285th sample points, the performance is similar with MPLS. In Fig. 22, the T_h^2 and T_t^2 monitoring charts of MMPLS detect the fault at the 288th and 279th sample points, it can detect the fault earlier than MPLS and MTPCR. Fig. 23 is the monitoring chart of multiphase-MEMPRM, T_h^2 and T_t^2 detect the fault at the 235th and 311th sample points, but there are false alarms between 0 to 200 sample points. The monitoring chart of multiphase-MCLWPR is in Fig. 24, ${T_c}^2$, ${T_y}^2$, T_x^2 and Q_x can detect the fault, especially for T_x^2, it detects the fault at the 211th sample point, ${T_c}^2$ detects the fault which delays a period of time, because when the ramp fault in agitator rate occurs, it propagates slowly for the quality variable under the closed-loop control. By contrast, the detection results of multiphase-MCLWPR are better than that of MPLS, MTPCR, MMPLS, and multiphase-MEMPRM. This is because multiphase-MCLWPR can describe the time sequence multiphase characteristic and extract the local quality-related features effectively.

In the simulation process of penicillin fermentation, the product concentration was selected to evaluate prediction performance. The quality prediction results are shown in Fig. 25, in which the predicted red dotted line of multiphase-MCLWPR tracks the real value line pretty well. To compare the prediction results visually, the scatters of prediction are shown in Fig. 26, we can see that the predicted values of multiphase-MCLWPR are closer to the reference line, that is to say, the proposed multiphase-MCLWPR shows a better performance than that of MPLS, MTPCR, and multiphase-MEMPRM. Also, RMSE values are listed in Table .6, it can be seen that the RMSE value of multiphase-MCLWPR is smaller than that of MPLS, MTPCR, and multiphase-MEMPRM. It indicates that the prediction accuracy of multiphase-MCLWPR is higher.

5. Conclusion

This paper proposes a multiphase concurrent locally weighted projection regression algorithm to monitor the nonlinear multiphase batch process running state and predict the quality variables. This algorithm first divides the entire process into multi-phases according to local quality-related characteristics and time sequence. The statistical characteristics are similar in each phase, which is convenient to describe the nonlinear characteristics. Furthermore, the nonlinear process is modeled with locally linear models in each partitioned phase, it can effectively represent the running state of the process. The complete monitoring indices of quality-related and process-related are built, and the quality variable is predicted while exploiting the regression structure for quality and process monitoring. The feasibility and efficiency of the proposed algorithm are verified by a nonlinear numerical case and penicillin fermentation process.

Author statement

Zhang Yan: Writing, Software, Validation.
Cao Jie: Conceptualization, Methodology.
Zhao Xiaoqiang: Reviewing and Editing, Supervision.
Hui Yongyong: Investigation, Data curation.

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.

References

[1] Y. Qin, Process Monitoring and Quality Control for Batch Process Based on Data-Driven Methods [D], Zhejiang University, 2018.
[2] C. Zhang, K. Peng, J. Dong, An extensible quality-related fault isolation framework based on dual broad partial least squares with application to the hot rolling process [J], Expert Syst. Appl. (2020), 114166.
[3] Z. Zhao, P.X. Liu, J. Gao, Fault detection for non-Gaussian stochastic distribution systems based on randomized algorithms[J], IEEE Trans. Instrum. Meas. 71 (2022) 1–9.
[4] Q. Jiang, X. Yan, B. Huang, Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes[J], Ind. Eng. Chem. Res. 58 (29) (2019) 12899–12912.
[5] J. Liu, D. Wang, J. Chen, Monitoring framework based on generalized tensor PCA for three-dimensional batch process data[J], Ind. Eng. Chem. Res. 59 (22) (2020) 10493–10508.
[6] C. Zhao, Y. Sun, Step-wise sequential phase partition (SSPP) algorithm based statistical modeling and online process monitoring, J]. Chemometrics and Intelligent Laboratory Systems 125 (2013) 109–120.
[7] P. Agarwal, M. Aghaee, M. Tamer, et al., A novel unsupervised approach for batch process monitoring using deep learning[J], Comput. Chem. Eng. 159 (2022), 107694.
[8] Z. Lv, X. Yan, Q. Jiang, Batch process monitoring based on multiple-phase online sorting principal component analysis[J], ISA (Instrum. Soc. Am.) Trans. 64 (2016) 342–352.
[9] J. Liu, T. Liu, J. Chen, Sequential local-based Gaussian mixture model for monitoring multiphase batch processes[J], Chem. Eng. Sci. 181 (2018) 101–113.
[10] J. Zhu, Y. Yao, F. Gao, Multiphase two-dimensional time-slice dynamic system for batch process monitoring[J], J. Process Control 85 (2020) 184–198.
[11] T.J. Rato, M.S. Reis, An integrated multiresolution framework for quality prediction and process monitoring in batch processes[J], J. Manuf. Syst. 57 (2020) 198–216.
[12] Y. Zhang, Z. Hu, Multivariate process monitoring and analysis based on multi-scale KPLS[J], Chem. Eng. Res. Des. 89 (12) (2011) 2667–2678.
[13] C. Botre, M. Mansouri, M. Nounou, et al., Kernel PLS-based GLRT method for fault detection of chemical processes[J], J. Loss Prev. Process. Ind. 43 (2016) 212–224.
[14] L. Luo, S. Bao, J. Mao, et al., Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes[J], Chemometr. Intell. Lab. Syst. 150 (2016) 9–22.
[15] D. Zhou, G. Li, S.J. Qin, Total projection to latent structures for process monitoring [J], AIChE J. 56 (1) (2010) 168–178.

[16] S.J. Qin, Y. Zheng, Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures[J], AIChE J. 59 (2013).
[17] Y. Zhang, R. Sun, Y. Fan, Fault diagnosis of nonlinear process based on KCPLS reconstruction[J], Chemometr. Intell. Lab. Syst. 140 (2015) 49–60.
[18] Z. Zhang, J. Zhu, Y. Liu, et al., Industrial Process Modeling and Fault Detection with Recurrent Kalman Variational Autoencoder [C]//2020 IEEE 9th Data-Driven Control and Learning Systems Conference, DDCLS), 2020.
[19] J. Yang, J. Wang, J. Sha, et al., Quality-related monitoring of distributed process systems using dynamic concurrent partial least squares[J], Comput. Ind. Eng. 164 (2022), 107893.
[20] P. Nomikos, J.F. Macgregor, Multi-way partial least squares in monitoring batch processes[J], Chemometr. Intell. Lab. Syst. 30 (1) (1995) 97–108.
[21] S. Zhang, C. Zhao, Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly[J], IEEE Trans. Ind. Electron. 66 (5) (2019) 3773–3783.
[22] Y. Zhang, X.Q. Zhao, Y.Y. Hui, et al., Online monitoring and fault diagnosis for uneven length batch process based on multi-way orthogonal enhanced neighborhood preserving embedding[J], Asia Pac. J. Chem. Eng. 17 (3) (2022), e2763.
[23] S. Vijayakumar, A. D'Souza, S. Schaal, Incremental online learning in high dimensions[J], Neural Comput. 17 (12) (2005) 2602–2634.
[24] G. Wang, S. Yin, O. Kaynak, An LWPR-based data-driven Fault Detection approach for nonlinear process monitoring[J], IEEE Trans. Ind. Inf. 10 (4) (2014) 2016–2023.
[25] E.B. Martin, A.J. Morris, Non-parametric confidence bounds for process performance monitoring charts[J], J. Process Control 6 (6) (1996) 349–358.
[26] R. Guo, K. Guo, J. Dong, Phase partition and online monitoring for batch process based on multiway BEAM[J], IEEE Trans. Autom. Sci. Eng. 14 (4) (2017) 1582–1589.
[27] Y. Hui, X. Zhao, Multi-phase batch process monitoring based on multiway weighted global neighborhood preserving embedding method[J], J. Process Control 69 (2018) 44–57.
[28] D. Hematillake, D. Freethy, J. McGivern, et al., Design and optimization of a penicillin fed-batch reactor based on a deep learning Fault Detection and diagnostic model[J], Ind. Eng. Chem. Res. 61 (13) (2022) 4625–4637.
[29] G. Birol, C. Ündey, A. Çinar, A modular simulation package for fed-batch fermentation: penicillin production[J], Comput. Chem. Eng. 26 (11) (2002) 1553–1565.