

兰州理工大学

科研成果汇总

学 号：221081101001

研 究 生：刘凯

导 师：赵小强 教授

研究方向：故障诊断

论文题目：基于数据驱动的间歇过程故障检测
与诊断方法研究

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

1. 检索报告.....	1
2. Liu Kai , Zhao Xiaoqiang*. An adaptive imbalance robust graph embedding broad learning system fault diagnosis for imbalanced batch processes data (SCI).....	8
3. Liu Kai , Zhao Xiaoqiang*. Quality prediction of multi-stage batch process based on integrated ConvBiGRU with attention mechanism (SCI).....	21
4. Liu Kai , Zhao Xiaoqiang*. Siamese DeNPE network framework for fault detection of batch process (SCI).....	43
5. Liu Kai , Zhao Xiaoqiang*. Nonlinear Quality - Related Fault Detection Using Neighborhood Embedding Neural Orthogonal Mapping Algorithm for Batch Process (SCI).....	64
6. Liu Kai , Zhao Xiaoqiang*. An adaptive strategy for time - varying batch process fault prediction based on stochastic configuration network (SCI).....	79
7. Liu Kai , Zhao Xiaoqiang*. A generalized zero-shot semantic learning model for batch process fault diagnosis (SCI).....	99
8. 刘凯, 赵小强*. 基于 S-RNPAE 算法的间歇过程早期故障监测 (EI).....	116
9. Zhao Xiaoqiang*, Liu Kai . Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm (SCI)....	126
10. Zhao Xiaoqiang*, Liu Kai. WOPCA-EGNPE algorithm based on variable division for fault monitoring of batch process (SCI).....	141
11. 刘凯, 赵小强*. 一种基于变分自编码器的无线网络中继选择方法, 发明专利 (已授权, 专利号: ZL202310971760.6)	156
12. 刘凯, 赵小强*. 基于孪生深度邻域保持嵌入网络的间歇过程故障检测方法, 发明专利 (已授权, 专利号: ZL202410220859.7)	157
13. 刘凯, 赵小强*. 基于自适应随机配置网络的间歇过程故障预测方法, 发明专利 (已授权, 专利号: ZL202410220859.7)	158



机构: 兰州理工大学 电气工程与信息工程学院

姓名: 刘凯 [221081101001]

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): 2023年-2025年
- 工程索引 (Engineering Index): 2023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	第二作者篇数
SCI-E 收录	SCI-EXPANDED	2023 - 2025	8	6	2
EI 收录	EI-Compendex	2023 - 2025	9	7	2

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): 刘凯

完成人(签字): 郑玲

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: R2025-0244 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: 刘凯 作者单位: 兰州理工大学 电气工程与信息工程学院		检索人员: 郑玲 检索日期: 2025年3月24日		
检索结果: 被 SCI-E 收录文献 8 篇						
#	作者	地址	标题	来源出版物	文献类型	入藏号
1	Liu, K; Zhao, XQ; Hui, YY; Jiang, HM	[Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Lanzhou Univ Technol, Coll Elect & Informat Engn, Qilihe St, Lanzhou 730050, Peoples R China.; [Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Lanzhou Univ Technol, Gansu Key Lab Adv Control Ind Proc, Qilihe St, Lanzhou 730050, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Qilihe St, Lanzhou 730050, Peoples R China.	Nonlinear Quality-Related Fault Detection Using Neighborhood Embedding Neural Orthogonal Mapping Algorithm for Batch Process	CHEMICAL ENGINEERING & TECHNOLOGY 2024, 47 (5): 764-778.	J Article	WOS:0011 935209000 01
2	Liu, K; Zhao, XQ; Mou, M; Hui, YY	[Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong] Gansu Key Lab 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.	A generalized zero-shot semantic learning model for batch process fault diagnosis	MEASUREMENT SCIENCE AND TECHNOLOGY 2025, 36 (1): 016228.	J Article	WOS:0013 770342000 01
3	Liu, K; Zhao, XQ; Mou, M; Hui, YY	[Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Liu, Kai; Zhao,	Quality prediction of multi-stage batch process based on integrated ConvBiGRU with attention mechanism	APPLIED INTELLIGENCE 2025, 55 (2): 123.	J Article	WOS:0013 741498000 10

		Xiaoqiang; Mou, Miao; Hui, Yongyong] Gansu Key Lab 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.				
4	Liu, K; Zhao, XQ; Hui, YY; Jiang, HM	[Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Gansu Key Lab Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.	An adaptive strategy for time-varying batch process fault prediction based on stochastic configuration network	<i>JOURNAL OF CHEMOMETRICS</i> 2024, 38 (9): 20.	J Article	WOS:0012 094229000 01
5	Liu, K; Zhao, XQ; Hui, YY	[Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong] Gansu Key Lab 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.	An adaptive imbalance robust graph embedding broad learning system fault diagnosis for imbalanced batch processes data	<i>PROCESS SAFETY AND ENVIRONMENTAL PROTECTION</i> 2024, 192: 694-706.	J Article	WOS:0013 482183000 01
6	Liu, K; Zhao, XQ; Mou, M; Hui, YY	[Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong] Gansu Key Lab Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui,	Siamese DeNPE network framework for fault detection of batch process	<i>CANADIAN JOURNAL OF CHEMICAL ENGINEERING</i> 2024, 102 (3): 1167-1187.	J Article	WOS:0010 720038000 01

		Yongyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.				
7	Zhao, XQ; Liu, K; Hui, YY	[Zhao, Xiaoqiang; Liu, Kai; Hui, Yonyong] Lanzhou Univ Technol, Coll Elect & Informat Engn, 287 Langongping Load, Lanzhou 730050, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yonyong] Gansu Key Lab Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yonyong] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.	Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm	<i>TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT AND CONTROL</i> 2023, 45 (1): 89-103.	J Article	WOS:0008 307639000 01
8	Zhao, XQ; Liu, K; Hui, YY; Jiang, HM	[Zhao, Xiaoqiang; Liu, Kai; Hui, Yonyong; Jiang, Hongmei] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Liu, Kai; Hui, Yonyong; Jiang, Hongmei] Gansu Key Lab Adv Control Ind Proc, Lanzhou, Peoples R China.; [Zhao, Xiaoqiang; Hui, Yonyong; Jiang, Hongmei] Lanzhou Univ Technol, Natl Expt Teaching Ctr Elect & Control Engn, Lanzhou, Peoples R China.	WOPCA-EGNPE algorithm based on variable division for fault monitoring of batch process	<i>BRAZILIAN JOURNAL OF CHEMICAL ENGINEERING</i> 2023, 40 (3): 827-841.	J Article	WOS:0008 546864000 01
合计						8



数据库: 工程索引 (Engineering Index)
时间范围: 2023年至2025年作者姓名: 刘凯
作者单位: 兰州理工大学 电气工程
与信息工程学院检索人员: 郑玲
检索日期: 2025年3月24日

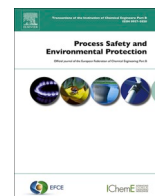
检索结果: 被 EI 收录文献 9 篇

#	作者	地址	标题	来源出版物	文献类型	入藏号
1	Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei	College of Electrical and Information Engineering, Lanzhou University of Technology, Qilihe Street, Lanzhou; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou University of Technology, Qilihe Street, Lanzhou; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Qilihe Street, Lanzhou	Nonlinear Quality-Related Fault Detection Using Neighborhood Embedding Neural Orthogonal Mapping Algorithm for Batch Process	<i>Chemical Engineering and Technology</i> 2024, 47 (5): 764-778.	Journal article (JA)	202414158 31682
2	Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Zhang, Yan	College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou; Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou	Incipient fault monitoring of batch process based on S-RNP AE algorithm 基于S-RNP AE算法的间歇过程早期故障监测	<i>Kongzhi yu Juece/Control and Decision</i> 2024, 39 (5): 1577-1586.	Journal article (JA)	202418159 97015
3	Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Centre of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	A generalized zero-shot semantic learning model for batch process fault diagnosis	<i>Measurement Science and Technology</i> 2025, 36 (1): 016228.	Journal article (JA)	202510180 05910

4	Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Centre of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	Quality prediction of multi-stage batch process based on integrated ConvBiGRU with attention mechanism	<i>Applied Intelligence</i> 2025, 55 (2): 123.	Journal article (JA)	202450175 15937
5	Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong; Jiang, Hongmei	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	An adaptive strategy for time-varying batch process fault prediction based on stochastic configuration network	<i>Journal of Chemometrics</i> 2024, 38 (9): e3555.	Journal article (JA)	202418160 25252
6	Liu, Kai; Zhao, Xiaoqiang; Hui, Yongyong	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	An adaptive imbalance robust graph embedding broad learning system fault diagnosis for imbalanced batch processes data	<i>Process Safety and Environmental Protection</i> 2024, 192: 694-706.	Journal article (JA)	202444172 72915
7	Liu, Kai; Zhao, Xiaoqiang; Mou, Miao; Hui, Yongyong	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Centre of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	Siamese DeNPE network framework for fault detection of batch process (Open Access)	<i>Canadian Journal of Chemical Engineering</i> 2024, 102 (3): 1167-1187.	Journal article (JA)	202339148 12687

8	Zhao, Xiaoqiang; Liu, Kai; Hui, Yonyong	College of Electrical and Information Engineering, Lanzhou University of Technology, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, China; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, China	Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm	<i>Transactions of the Institute of Measurement and Control</i> 2023, 45 (1): 89-103.	Journal article (JA)	202231124 70688
9	Zhao, Xiaoqiang; Liu, Kai; Hui, Yonyong; Jiang, Hongmei	College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China; Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China; National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, China	WOPCA-EGNPE algorithm based on variable division for fault monitoring of batch process	<i>Brazilian Journal of Chemical Engineering</i> 2023, 40 (3): 827-841.	Journal article (JA)	202238127 78708
合计						9





An adaptive imbalance robust graph embedding broad learning system fault diagnosis for imbalanced batch processes data

Kai Liu^{a,b}, Xiaoqiang Zhao^{a,b,c,*}, Yongyong Hui^{a,b,c}

^a College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China

^b Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, China

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

ARTICLE INFO

Keywords:

Batch process
Fault diagnosis
Broad learning system
Imbalanced data
Adaptive modified

ABSTRACT

Modern batch processes develop toward higher levels, and devices usually work at normal conditions with generally very few faults. Therefore, fewer fault data are collected than normal data, which makes the normal and fault modes imbalanced, and the fault diagnosis model is incapable focusing on the minority of fault samples as much as on most normal samples, consequently leading to insufficient generalization ability for the model. For that, we propose an adaptive imbalance-robust graph embedding broad learning system (AI-RGEBLS) in this paper. Firstly, it achieves adaptive correction of unbalanced samples by niche technique and synthetic minority over-sampling technique (SMOTE) with improved Mahalanobis distance. Then, broad learning system (BLS) is regularized by graph embedding, and further introduced the $L_{2,1}$ norm constraint, which makes it possible to enhance the robustness of the model while considering the local manifold information of the data in the feature extraction process. Finally, the incremental learning approach is applied to the model to avoid the disastrous forgetting problem caused by training the whole model from zero. The effectiveness of the proposed method is verified by penicillin fermentation process and semiconductor etching process. It can effectively improve the speed of model training and provide better fault diagnosis for imbalanced batch processes compared with the existing methods.

1. Introduction

As the demand for multi-standard, refined, and individual products increases, batch production processes are focusing the attention in industry and academics. It is essential for the safe and reliable process requirements of the batch process (Arunthavanathan et al., 2021a; Chang et al., 2023; Jiang and Ge, 2020; Khan et al., 2015; Qian et al., 2022). Modern batch process the development was intelligent and complex, which led once the equipment fault maintenance cost is very high. And after the fault occurred, due to the equipment stopping caused great economic losses for the enterprise, the heavy is very easy to accompany the major casualty problem (Gu et al., 2023; Ji et al., 2023; Khan and Yairi, 2018; Zhao et al., 2023). Therefore, it is very crucial for the effective fault diagnosis for the batch process on the safe production.

The data driven approach is much appreciated because it does not require precise modeling as well as empirical knowledge. Such methods like multivariate statistics (Yoon and MacGregor, 2001), representation learning (Fan et al., 2021), reinforcement learning (Zhang et al., 2023b),

and shallow learning (Cirrincione et al., 2020) have been applied to industrial process fault diagnosis with favorable results. In many complex scenes, the variety of fault categories and the complex properties among the variables caused not to achieve satisfactory results. The deep learning approach is recognized as the preferred fault diagnosis method for its unique ability for data interpretation. Such networks include, Pan et al., 2021 proposed a novel classification-driven neuron grouped stacked autoencoder (CG-SAE) for hierarchical fault related feature representation. Wang et al. (2020) proposed an extended deep belief network (EDBN) to fully utilize the useful information in the original data, and constructed a fault classifier based on the dynamic EDBN by taking the dynamic characteristics of process data into full consideration. Chen et al. (2022) proposed a fusion model (CS-IMLSTM) based on convolutional neural network (CNN), squeeze excitation (SE) attention mechanism, and improved long short-term memory network (IMLSTM) for chemical process fault diagnosis, which significantly improved the fault accuracy. Arunthavanathan et al. (2021b) proposed a combined CNN and LSTM approach for early potential fault detection.

* Corresponding author at: College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China.

E-mail addresses: lutlk196@163.com (K. Liu), xqzhao@lut.edu.cn (X. Zhao), hyylut@163.com (Y. Hui).

<https://doi.org/10.1016/j.psep.2024.10.043>

Received 26 July 2024; Received in revised form 1 October 2024; Accepted 12 October 2024

Available online 15 October 2024

0957-5820/© 2024 Institution of Chemical Engineers. Published by Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Zheng and Zhao (2020) proposed an unsupervised data mining method based on convolutional stacked autoencoder (CSAE) of deep learning that is used to isolate different states of chemical processes, which include normal operation and malfunctions, and construct fault diagnostic models. Wang et al. (2024c) proposed a novel fault diagnosis network based on a refined prototype and correlation weighting Manhattan distance (RPCMN). Wang et al. (2024b) proposed a multi-scale cross feature extraction module (MSCM), which is constructed to mine key classification information under noise interference to improve fault identifiability. Wang et al. (2024a) proposed an innovative Neural-Transformer to realize high-precision robust fault diagnosis with low computational cost. Nonetheless, these deep learning networks usually require frequent parameter tuning and multilayer stacking to achieve the desired accuracy. However, such process can be extremely time-consuming, which runs counter to the goal of building a model with both efficiency and accuracy.

In view of the problem of training and updating time consuming for deep learning, Chen and Liu (2017) proposed a network structure called broad learning system (BLS). As alternative methods for learning deep structures, BLS transforms the problem with finding complex nonlinear relationships between the input and output layers of multilayer deep networks into the problem of finding linear relationships between two-layer-breadth stochastically mapped layers and the output layer, which has gained significant research interest due to its unique appeal, and led to the emergence of a numbers of improved methods based on BLS. Pu and Li (2021) proposed an online semi-supervised broad learning system (OSSBLS) for fault diagnosis in situations where data labels are difficult to obtain. The proposed method not only efficiently constructs and incrementally updates the model, but also improves the diagnostic performance of the model by utilizing unlabeled data. Lu et al. (2023) proposed an improved BLS fault diagnosis method based on data enhancement and multidomain feature fusion, which improved BLS effectively solves the problem of sample imbalance and greatly improves diagnostic accuracy. Hu et al. (2024) proposed a multi-feature fault diagnosis method based on a weighted temporal broad learning system, which incorporates multiple features extracted from raw process data to improve the fault diagnosis performance, and makes the diagnostic model suitable for dynamic fault diagnosis problems by combining with BLS.

Notably, from the perspective of global structure, BLS deals with the data and seldom considers the geometric structure. Graph neural network (Yin et al., 2024) shows great potential in dealing with complex relationships and structured data. However, the main advantage of graph embedding over graph neural network is that it is computationally efficient and versatile, and a fixed representation of the nodes can be generated through a single embedding, which can be quickly used in subsequent tasks. Local metrics and neighborhood information can fully reveal the intrinsic geometric features of the data, and with the above analysis, scholars have proposed several graph embedding methods. Such as, locality preserving projection (LPP) (Shah et al., 2022; Sobhani-Tehrani et al., 2014) and neighborhood preserving embedding (NPE) (He et al., 2019; Sobhani-Tehrani et al., 2014) and its extended models which aim to transform data by preserving the local neighborhood structure while preserving the intrinsic geometric properties of the original data. With the advantage that NPE can maintain the local near neighbor structure of data while prevents model overfitting. Many scholars have improved the generalization ability and stability of embedding by NPE regularization model. Liu et al. (2021) proposed a novel Stacked Neighborhood Preserving Autoencoder (S-NPAE) to extract hierarchical neighborhood preserving features, which regularizes the AE by NPE in to reconstruct the input data while preserving the neighborhood structure of the input data. Zhang et al. (2023a) proposed a dual-preserving integrated neighborhood localization projection (DPNLP) algorithm for fault diagnosis. When solving the singular matrix problem, that approach introduces regularization into the DPNLP to preserve both the neighborhood similarity and the dual preservation

weights of local linear reconstruction, which enhances the ability of dimensionality reduction. However, when it comes to modeling time series with noise and outliers, only the above regularization may not be sufficient to eliminate the adverse effects. Park and Jo (2016) proposed a novel multilayer perceptron (MLP) regularization method that learns a regression function in the presence of noise, regardless of how smooth the function is, and fits nonsmoothed functions more accurately. In the training process of neural networks, it is common to introduce noise to enhance the generalization performance, and adding noise to the input data is a common method to prevent overfitting in deep learning (Wang and Chen, 2018). In addition, it regularizes the neural network by adding noise to its hidden units, which accelerates the convergence performance of the neural network and improves the model robustness and diagnostic accuracy (Feng et al., 2019). Zhu et al. (2023) proposed a new incremental learning method using classification and feature-level information, which aimed to improve the robustness of the model under noisy conditions.

There are available fault diagnosis approaches which assume of balanced distribution of data across different classes in the dataset. In actual industrial production, systems are usually run under normal operating conditions. Therefore, fault data is very limited when compared to data from normal operation. How to eliminate data imbalance for the establishment of accurate fault diagnosis model, which is very important issue. Nowadays, there are two main approaches to solve the sample imbalance problem: algorithm and data-based approach (Jing et al., 2019). Algorithm-based methods better handle the classification of imbalanced data by adjusting the algorithm. For example, cost-sensitive learning (Khan et al., 2017) and integration learning (Razavi-Far et al., 2017). In contrast, data-based methods have the advantage of dealing with imbalanced samples independently of the data-driven model and without interfering with each other (Dong et al., 2024c). Dong et al. (2024a) proposed a dynamic normalization supervised contrastive network (DN SCN) with a multiscale compound attention mechanism to recognize imbalanced faults. Dong et al. (2024b) proposed a multi-sensor data fusion-enabled lightweight convolutional double regularization contrast transformer for small samples fault diagnosis. Tan et al. (2023) proposed a deep adversarial learning system for fault diagnosis in the Fused deposition modeling (FDM) process, based on captured upper layer images during the manufacturing process. Conditional generative adversarial network is adopted to augment the original dataset and solve the between-class data imbalance problem. Compared to other methods, SMOTE directly balances the data distribution by generating a small number of class samples, which is simple and does not require the modification of the model structure or training mechanism. Zhang et al. (2022) proposed a sequence oversampling discriminative method for imbalanced batch process fault detection, where the whole batch of sequences is transformed into multiple fixed length sequences through sliding windows and robust time varying dynamic features are extracted from each batch of sequences. Moreover, oversampling neural network is used to balance the sequences of the minority class and the majority class. Li et al., (2021) proposed a synthetic minority oversampling technique with natural neighbors (NaNSMOTE). In NaNSMOTE, the random difference between the selected base samples and their natural neighborhood samples is used to generate synthetic samples to improve the generalization ability of the synthetic samples and reduce the error of the synthetic samples. Men and Zhao, (2023) proposed an adaptive imbalanced data stream correction method for fault diagnosis under imbalanced chemical process data by combining the niche technique, oversampling technique, and manifold regularization technique. However, in the above mentioned SMOTE-based oversampling algorithms, clustering is usually performed by using the Euclidean distance, which does not consider the possible coupling relationship between the features, and thus the features cannot be recognized effectively.

Motivated here, we propose an adaptive imbalance robust graph embedding broad learning system fault diagnosis. which dynamically

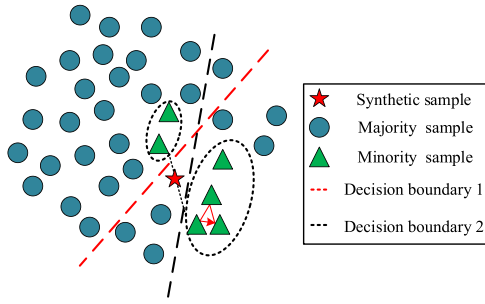


Fig. 1. Illustration of class imbalanced problem.

corrects for imbalance diagnostic data and generates representative samples. The construction of the AI-RGEBS model involves both manifold regularization and the model robustness for noise, while AI-RGEBS can train and update the fault diagnosis model efficiently in an incremental manner.

The main contributions of this paper are as follows:

- (1) The traditional SMOTE algorithm with Euclidean distance does not consider the coupling relationship between the features. To avoid this issue, the Mahalanobis distance is introduced, as it does not depend on scale and improves the SMOTE algorithm by eliminating the influence of different dimensions.
- (2) It dynamically generates samples through niche technique and continuously adjusts the oversampling process of SMOTE, thus increasing the amount of information and realizing adaptive modification of imbalanced samples. While enhancing the representativeness of the data, it improves the performance and stability of the model in dealing with imbalanced data.
- (3) Broad learning system (BLS) is regularized by graph embedding, so that the local structure information of the data can be considered in the process of feature extraction. BLS can better capture and utilize the local geometric relationship of the data to improve the effect of feature extraction.
- (4) To enhance the robustness of the model, introduce $L_{2,1}$ norm bounds to construct the robustness matrix. In addition, the incremental learning method is utilized to gradually add enhancement nodes in the incremental window, which avoid the disastrous forgetful problem caused due to training whole model since zero.

The rest of this paper is organized as follows. The overview of the base model and problem description are illustrated in Section 2. The proposed adaptive imbalance robust graph embedding broad learning system modeling (AI-RGEBS) is shown in Section 3. Fault diagnosis of imbalanced batch processes based on AI-RGEBS which is shown in Section 4. The experiments of the proposed method based on the penicillin fermentation process and the semiconductor etching process are validated in Section 5. And Section 6 summarizes this paper.

2. Overview of the base model and problem description

In this section, we focus on introducing the theory of the basic model BLS and descriptions of the class imbalance problem.

2.1. Problem description

In complex batch process fault diagnosis and quality prediction etc. tasks, there is usually most normal state data and less data on faults or abnormal states. It is because the production process is always working in a normal state most of the time, and faults and abnormal conditions are relatively rare. Meanwhile, data collection for abnormal states is not timely or complete, which results in a severely insufficient number of

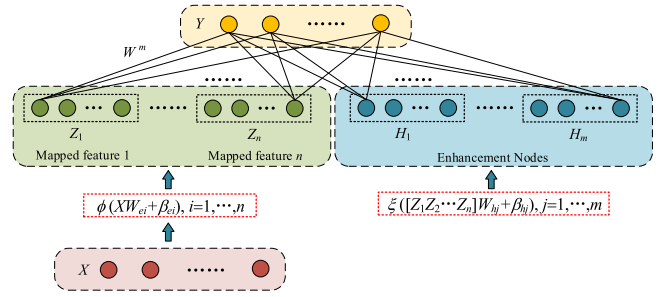


Fig. 2. The linear network structure of the BLS.

data samples. Illustration of class imbalanced problem is shown in Fig. 1.

In the Fig. 1, where different colors and shapes of points indicate different classes of samples, the challenge of imbalanced classification is demonstrated by the distribution and classification boundaries in the figure. Normal samples (majority samples) occupy most of the area of the graph, while abnormal samples (minority samples) are very few and scattered in distribution. Due to the small number of abnormal samples, the classification model tends to prefer normal samples, which results in classification boundaries cannot accurately distinguish abnormal samples. Under extreme imbalanced data, the model may either overfit the normal samples and not recognize the abnormal samples correctly, or it may be underfitted because there are too few abnormal samples. In addition, it can be seen from the Fig. 1 that a few samples are in different subspaces, the densities of different subspaces are inconsistent and the distribution density of a few samples is high.

For imbalanced samples, SMOTE is usually used for oversampling, SMOTE makes use of the spatial relationships between samples to synthesize new samples, which means it can be suitable for all sampling schedules. Typically, SMOTE algorithm adopts Euclidean distance-based clustering method for oversampling, but Euclidean distance does not consider the coupled relationship between features. Therefore, it is necessary to carry out the traditional SMOTE algorithm distance measure, and find a way that can make the SMOTE algorithm to consider more feature relationships when oversampling. In addition, synthetic oversampling methods are usually limited by information loss and adaptivity. To fully consider the dynamic characteristics of the fault modes by improving SMOTE technique to recognize the samples with more information and adaptive modification of the number of samples for the corresponding fault modes.

When the process data scales are large, the deep learning model will encounter the problems of weight adjustment being time-consuming and easy to fall into the local optimal solution. To solve these problems, BLS came into being, which not only has a simple structure, fast training speed and high accuracy, but also has the advantage of incremental learning.

2.2. Broad learning system (BLS)

BLS is an efficient incremental learning system. It is different from traditional deep neural networks without constructing deep network structure. The linear network structure of the BLS is shown in Fig. 2.

In BLS, firstly the original input data is converted into random features in the feature nodes, secondly a series of enhancement nodes are randomly generated through the feature nodes to extend the width of the network, then all the feature nodes and enhancement nodes are combined and imported into the output layer, finally the corresponding output coefficients are obtained by computing the pseudo-inverse of the linear system equations. The BLS model has an effective time performance in training, and hence is more valid than the deep learning methods in some occasions. The typical BLS modeling specific steps are as follows:

- (1) For given training data X , the input data is transformed into

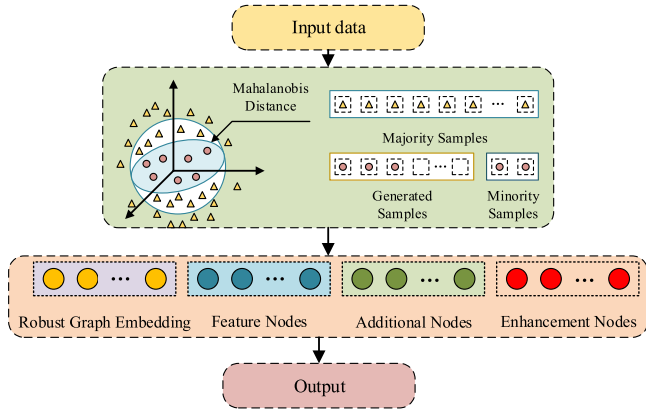


Fig. 3. The network structure of AI-RGEBLS.

random features on the feature nodes with linear mapping, and the random feature is represented as shown in formula (1):

$$Z_i = \phi_i(XW_{ei} + \beta_{ei}) \in R^{N \times k}, i = 1, \dots, n \quad (1)$$

where $\phi(\cdot)$ is the activation function of the linear transformation, W_{ei} and β_{ei} are the random weights and biases of the appropriate dimensions, N denotes the number of samples, k is the number of feature nodes in each group, and n is the number of feature node groups. For n groups of feature nodes can be expressed: $Z^n \triangleq [Z_1, Z_2, \dots, Z_n]$.

(2) The enhancement nodes are transformed from the feature nodes, which is obtained from formula (2):

$$H_j = \xi_j(ZW_{hj} + \beta_{hj}) \in R^{N \times 1}, j = 1, \dots, m \quad (2)$$

where ξ_j is the nonlinear activation function, W_{hj} and β_{hj} are the random weights and biases of the appropriate dimensions, and m is the number of enhancement nodes. The set of augmented nodes is denoted as: $H^m \triangleq [H_1, H_2, \dots, H_m]$.

(3) All the feature nodes Z^n and enhancement nodes H^m are combined into $A = [Z^n | H^m]$ and fed into the output layer. The generalized model output Y_o of BLS is given by formula (3):

$$\begin{aligned} Y_o &= AW \\ &= [Z^n | H^m] W \\ &= [Z_1, Z_2, \dots, Z_n, H_1, H_2, \dots, H_m] W \end{aligned} \quad (3)$$

where W is the output coefficient of the model. The coefficient W is calculated by minimizing the sum of the squared losses of the prediction errors shown in formula (4):

$$\min_W (\|AW - Y\|^2 + \lambda \|W\|^2) \quad (4)$$

where Y denotes the true value of the target.

Formula (4) can be solved by calculating violation, as shown in formula (5):

$$W = (\lambda I + A^T A)^{-1} A^T Y \quad (5)$$

where λ is the adjustment coefficient, I is the unit matrix.

In this study, we aim to design an imbalanced batch process fault diagnosis method that involves combined SMOTE and BLS to identify the type of faults accurately and quickly. As mentioned before, the main problem is adaptive imbalance modification for process imbalance data while generate infinite number of representative samples and make full use of these sample features to construct the fault diagnosis model.

3. Adaptive imbalance robust graph embedding broad learning system modeling (AI-RGEBLS)

In this section, we describe in detail for the modeling of the proposed adaptive imbalance robust graph embedding broad learning system. The structure of the proposed AI-RGEBLS model as shown in Fig. 3.

The structure contains two parts: (1) Adaptive modified for imbalanced samples: Representative samples are generated by improved SMOTE based on Mahalanobis distance and niche technique; (2) Robust graph embedding broad learning system modeling (RGEBLS): Based on BLS, robust graph embedding is utilized to enhance the feature extraction capability of the model, while the incremental learning ability of BLS is enhanced by adding enhancement nodes.

3.1. Adaptive modified for imbalanced samples

For the imbalance problem, we combine niche technique and the Mahalanobis distance improved SMOTE algorithm to deal with it. The specific steps of adaptive modified for imbalanced samples is as follows. The historical training data D_{htd} and new training data D_{ntd} is merged into dataset union $X = D_{htd} \cup D_{ntd}$. For $X = (x_1, x_2, \dots, x_n) \in R^{n \times m}$ with the mean value is $\mu = (\mu_1, \mu_2, \dots, \mu_m)^T$ and the covariance matrix is Σ , where n is the number of samples, and m is the dimension of the sample.

The niche account $m(x)$ is defined as the sum of the similarities between sample x and other samples in the neighborhood. Calculate the niche account as shown in formula (6):

$$m(x) = \sum_{x' \in N_k, x' \neq x} sf(x, x') \quad (6)$$

where, the neighborhood is defined as: $\forall x' \in N_k \wedge x' \neq x$, The similarity is measured by Mahalanobis distance instead of the traditional Euclidean distance, and the Mahalanobis distance is calculated as shown in formula (7):

$$D_m(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (7)$$

The niche account threshold χ is shown in formula (8):

$$\chi = 0.3 \times \frac{\sum_{x \in D_{htd}} m(x)}{|D_{htd}|} \quad (8)$$

where, $|D_{htd}|$ is the size of historical training data.

The imbalance dataset was modified by niche account threshold χ to add the appropriate samples. When $m(x) < \chi$, select the k nearest neighbor samples and record them as $x_j, j = 1, 2, \dots, k$. The new sample generated based on SMOTE is shown in formula (9):

$$x_{new} = x_i + rand(0, 1) \times (x_j - x_i) \quad (9)$$

When the training fault model output label is the same as the actual label, $m(x)$ is calculated as shown in formula (10):

$$m(x) = m(x) + sf(x, x_{new}) \quad (10)$$

The modified training data D_{ld} is obtained from generated new samples x_{new} . Update fault diagnosis models with dataset union $D_{ld} \cup D_{ntd}$. After one iteration of the model, historical training data were updated: $D_{htd} = D_{htd} \cup D_{ntd}$.

3.2. Robust graph embedding broad learning system modeling (RGEBLS)

In industrial production system modeling, the samples collected by sensors contain various noises and outliers. we use BLS does not consider the influence of noise and has poor robustness. At the same time, simply excluding outliers from modeling may lead to the neglect of critical states. For this, we regularize the BLS by neighborhood preserving embedding (NPE), which is a graph embedding approach that allows the local structural information of the data to be considered in the feature

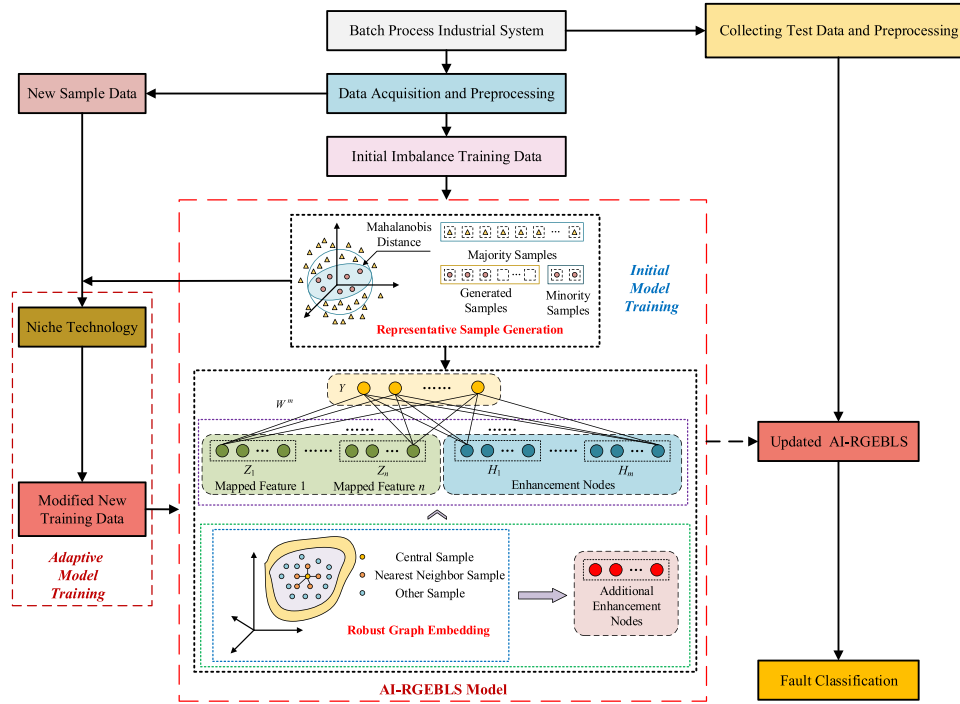


Fig. 4. Flowchart of AI-RGEBS based imbalanced batch process fault diagnosis.

extraction process. Further $L_{2,1}$ norm is introduced to enhance the robustness of the model. At the same time, the incremental learning pattern allows additional enhancement nodes to be quickly merged into the incremental window of the model without the need to start the training process from scratch, which enables timely adjustment of the structure of the model and the accuracy of the fault diagnosis to achieve

$$\|W_e\|_{2,1} = \sum_{i=1}^u \sqrt{\sum_{j=1}^v W_{ij}^2} = \sum_{i=1}^u \|W^i\|_2 \quad (13)$$

Therefore, according to our main considerations, the overall objective function of RGEBS is shown in formula (14):

$$\min_W \left(\|AW_{BLS} - Y_{BLS}\|^2 + \lambda \|W_{BLS}\|^2 + \alpha \|W_e\|_{2,1} + \beta \text{Tr}(Y_{NPE}(I - W_{NPE})^T(I - W_{NPE})Y_{NPE}^T) \right) \quad (14)$$

the desired results.

Neighborhood preserving embedding (NPE): NPE represents the local linear graph structure of the manifold by local linear reconstruction in the form of mean square error. The loss function of NPE is shown in formula (11):

$$\min \sum_{i=1}^n \left| y_i - \sum_{j \in Q(i)} W_{ij} y_j \right|^2 \quad (11)$$

where, $Q(i)$ is the set of neighboring samples; $y_j \in R^{D \times n}$ is the local manifold feature extracted from X , D is the dimension of the feature vector, $W \in R^{n \times n}$ should satisfy the normalization constraint.

The modeling for formula (11) can be transformed into the form of matrix as shown in formula (12):

$$\text{Tr}(Y(I - W)^T(I - W)Y^T) \quad (12)$$

where, $Y \in R^{D \times n}$ is the matrix form of the local manifold features, $I = \text{diag}(1, \dots, 1)$.

Transformation of the low-rank constant of the matrix rank W allows the selection of the features most relevant to the low-dimensional manifold. The $L_{2,1}$ norm constraint is applied to improve the robustness of the model, as shown in formula (13):

where, α and β are balance parameters. The first term is the loss function of the BLS model. W_{BLS} and Y_{BLS} are the weights and outputs of the BLS model. W_{NPE} and Y_{NPE} are the weights and outputs of the NPE model. The second is the $L_{2,1}$ norm constraint on the transformation matrix W_e , which reduces the effect of noise on the modeling and improves the robustness of the model. The third is the graph embedding regularization, which is used to maintain the local manifold structure of the data.

When new samples or classes appear, it is necessary to retrain the model to achieve better diagnostic results, and retraining is often time-consuming. Therefore, we propose the incremental learning method with AI-RGEBS. When adding new samples or classes, the additional enhancement nodes are quickly merged into the incremental window of AI-RGEBS, which eliminates the need to start the training process from scratch.

By adding a new group of enhancement nodes, the AI-RGEBS model can be efficiently updated. When new samples X_{new} appear, new training samples can represent as follows:

$$X^* = \begin{bmatrix} X \\ X_{new} \end{bmatrix} \quad (15)$$

Then the output representation matrix of AI-RGEBS is shown in the formula (16):

$$A_{mn}^a = \begin{bmatrix} A_{mn} \\ A_a \end{bmatrix} \quad (16)$$

where, A_{mn} is the original BLS output representation matrix. A_a is derived from formula (17):

$$A_a = [Z_a | \xi(Z_a W_h + \beta_h)] \quad (17)$$

where, $Z_a = X_a W$. The pseudo-inverse of A_{mn}^a is calculated from formula (18):

$$(A_{mn}^a)^+ = [(A_{mn})^+ - BD^T | B] \quad (18)$$

where, for formula (18) is solved according to the pseudo-inverse update from formula (19) to formula (21) which are obtained:

$$D = ((A_{mn}^a)^T)^+ A_a^T \quad (19)$$

$$B = \begin{cases} C^+ \text{if } C \neq 0 \\ (1 + D^T D)^{-1} D^T (A_{mn})^T \text{if } C = 0 \end{cases} \quad (20)$$

$$C = A_a^T - (A_{mn})^T D \quad (21)$$

where, $(\cdot)^+$ is the pseudo-inverse of matrix. The updated output weight is shown in formula (22):

$$W_{mn}^a = W_{mn} + B(Y_{new} - A_a W_{mn}) \quad (22)$$

where, Y_{new} is the label of the new sample X_{new} .

4. Fault diagnosis of imbalanced batch processes based on AI-RGEBS

In this section, we perform fault diagnosis based on the AI-RGEBS model for imbalanced batch process and give specific steps. The fault diagnosis flowchart of the imbalanced batch process based on AI-RGEBS is shown in Fig. 4.

The fault diagnosis of batch process based on AI-RGEBS consists of two parts: model training and fault diagnosis. In the model training, samples collected from the batch process industrial system are pre-processed. The initial AI-RGEBS model is trained by processing the collected imbalanced samples to generate balanced samples through the SMOTE algorithm with improved Mahalanobis distance. The new sample data dynamically adjust the oversampling process of the improved smote through the niche technique to realize the adaptive modification of the imbalanced samples, and get the modified new training data to update the AI-RGEBS model. In fault diagnosis, the new collected test samples are input into the updated AI-RGEBS model to obtain the fault diagnosis results.

To further evaluate the effectiveness of the model for fault diagnosis, we measure the effectiveness of fault diagnosis with relevant evaluation indicators. This study mainly involves a strongly imbalanced proportion of samples from different classes, so that the class with a large proportion tends to be the most important factor that affects the accuracy rate. We adopt the $F1$ –score and G –mean indicators to evaluate the fault diagnosis performance of the model, which considers the effect of the imbalance rate of the data. The indicators are calculated as shown in formula (23) and formula (24):

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (23)$$

$$G - mean = \sqrt{\frac{TP}{TP + FN} + \frac{TN}{TN + FP}} \quad (24)$$

where, TP denotes the number of positive classes predicted to be positive, TN denotes the number of negative classes predicted to be negative, FP denotes the number of negative classes predicted to be positive, i.e.,

Table 1

Parametric description of different comparison models.

Model	Description
NPE	Nearest neighbors: 10, Dimensionality reduction: 2
DNN (1)	1024–600–200–50–7 with learning rate: 0.05
DNN (2)	2000–500–150–50–8 with learning rate: 0.05
BLS	Feature nodes: $n=7$, Enhancement nodes: $K_m=2000$, Mapped feature nodes: $m=1$
DSSAE (1)	Layer 1: 80, Layer 2: 40, MaxEpochs:50, Sparsity Proportion: 0.1
DSSAE (2)	Layer 1: 100, Layer 2: 50, MaxEpochs:50, Sparsity Proportion: 0.1
SNN (1)	Spike Duration: 2 ms, Time Step: 1 ms, Learning rate: 0.01
SNN (2)	Spike Duration: 1 ms, Time Step: 1 ms, Learning rate: 0.03
MV-LEAP	Data view of size: $n \times m=100 \times 200$, Number of views: $N_v=4$
CGFTSVM	Kernel type: RBF kernel, Learning rate: 0.03
AI-	Random state: $R=42$, k neighbors-SMOTE:5, k neighbors-NPE:10,
RGEBS	Feature nodes: $n=7$, Enhancement nodes: $K_m=2000$, Mapped feature nodes: $m=1$

false positives, and FN denotes the number of positive classes predicted to be negative, i.e., false negatives. *Precision* and *Recall* are computed as shown in formula (25) and formula (26):

$$Precision = \frac{TP}{TP + FP} \quad (25)$$

$$Recall = \frac{TP}{TP + FN} \quad (26)$$

5. Case Studies

In this section, we evaluate the performance of the proposed model for imbalanced data fault diagnosis by two typical batch processes, penicillin fermentation process (Case 1) and semiconductor etching process (Case 2). We compared AI-RGEBS with seven models including Neighborhood preserving embedding (NPE), Deep Neural Networks (DNN), BLS, Dynamic Sparse Stacked Auto-encoders (DSSAE) (Jiang et al., 2017), Spiking Neural Network (SNN) (Zuo et al., 2021), Multi-View Learning Data Proliferator (MV-LEAP) (Graa and Rezik, 2019) and Class probability and Generalized Fuzzy Twin SVM (CGFTSVM) (Kumari et al., 2024) at five different balance ratios.

Meanwhile, we designed ablation experiments by with Adaptive Imbalance-Broad Learning System (AI-BLS), AI-GEBS, SMOTE-Robust Graph Embedding Broad Learning System (S-RGEBS) and Mahalanobis SMOTE-Robust Graph Embedding Broad Learning System (MS-RGEBS), and demonstrate the effectiveness and superiority of the proposed AI-RGEBS model. BLS is performed using the BLS toolkit provided in the literature (Chen and Liu, 2017) with 20 feature nodes and 2000 augmentation nodes. NPE (Mou and Zhao, 2022) use typical structures. DNN are implemented by the Deep Learning toolbox in MATLAB 2023b.

For the case studies that mentioned the imbalance ratio means the ratio of normal samples to fault samples, we set the same number of samples for different fault samples to better illustrate the problem. A description of the parameters for the different models is shown in Table 1. In Table 1, labeled (1) indicates the model parameter details for Case 1 and labeled (2) indicates the model parameter details for Case 2. The unlabeled cases (1) and (2) indicate that the same parameters were used for Case 1 and Case 2.

5.1. Penicillin fermentation process (PFP)

Penicillin is widely used and highly valuable antibiotic in clinical medicine. Its production process involves complex biochemical reactions and usually operates in batch mode. In this case, the penicillin fermentation process data were obtained from the pensim2.0 platform developed by the Illinois Institute of Technology (Biol et al., 2002). A schematic diagram of the penicillin fermentation process is shown in Fig. 5.

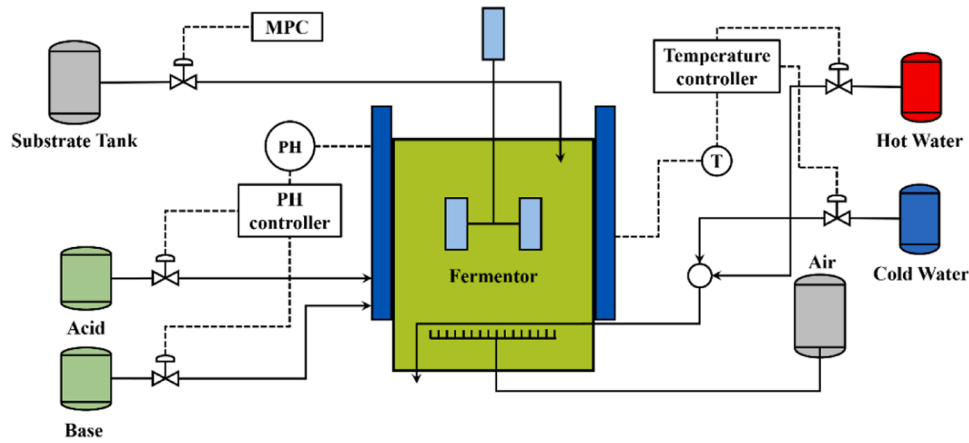


Fig. 5. Schematic of the Penicillin fermentation process.

Table 2
Penicillin fermentation process fault information.

No.	Fault variable	Fault amplitude	Fault signal type	Fault time
1	Aeration rate	+1 %	Step Signal	0–400 h
2	Aeration rate	−0.5	Ramp Signal	0–400 h
3	Agitator power	+1.5 %	Step Signal	0–400 h
4	Agitator power	−1	Ramp Signal	0–400 h
5	Substrate feeding rate	+2 %	Step Signal	0–400 h
6	Substrate feeding rate	−0.01	Ramp Signal	0–400 h

The main equipment of the fermentation system consists of fermenters, agitation motors and aeration equipment, in addition to sections for the addition of reactants such as acids, bases, hot water, cold water and substrates, with corresponding temperature and pH controllers. In penicillin fermentation, specific production bacteria grow and multiply under suitable conditions, and after reaching a certain concentration, it starts to produce penicillin as metabolite, which needs to be constantly supplemented with nitrogen, sugar and other nutrients to ensure the continuous production of penicillin.

To validate the proposed model, we use the penicillin fermentation process simulation platform Pensim 2.0 to generate 50 batches of normal samples and 6 batches of different types faults for model training. In this case, the different fault types are set to have 2400 samples, ensuring the same number of samples for each type of fault.

Based on the imbalance rate, we choose the number of normal samples for training the model at different imbalance rates. During model training, we use a grid search method to adjust the parameters. We focus on the training data imbalance, so we select for testing with 6 types of Faults and Normal samples totally 2800 samples for testing. For each batch samples, the reaction time is 400 h and the sampling time is 1 h, the faults are introduced from the moment of the reaction start to end, the details of the fault batch are shown in Table 2.

The optimization problem shown in formula (14) in AI-RGEBSL exists, and the balanced parameters α and β are taken to be found by grid search. The variation between the balanced parameters α and β chosen

Table 3
The comparison of *F1-score*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
NPE	87.9963	86.4695	84.9291	82.3398	79.0365	3.5428
DNN	88.4487	86.6985	85.2535	83.2698	79.0259	3.6216
BLS	89.0029	87.2598	86.0032	84.2569	80.9965	3.0597
DSSAE	89.2203	88.6889	86.2954	85.0533	83.2264	2.5021
SNN	90.2364	88.9039	87.0282	86.8625	84.7796	2.0898
MV-LEAP	93.4904	93.0069	92.8642	91.0089	89.2658	1.7617
CGFTSVM	93.6889	93.0568	92.8647	91.5987	90.0269	1.4548
AI-RGEBSL	94.6691	94.0058	93.8858	93.5356	93.0859	0.8570

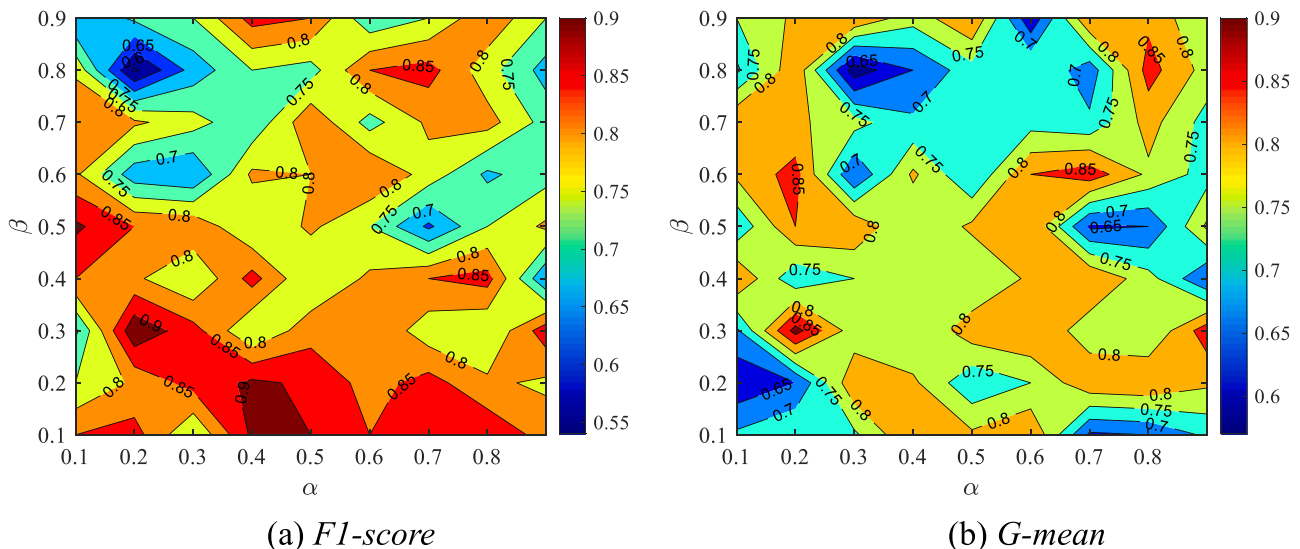


Fig. 6. The *F1-score* and *G-mean* diagrams for the balanced parameters α and β in the PFP.

Table 4
The comparison of *G-mean*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
NPE	85.6325	83.2665	81.2956	80.0026	77.2658	3.1785
DNN	85.2658	83.9964	80.9569	79.6559	76.2658	3.5755
BLS	89.1944	87.5529	85.2658	83.0021	80.2952	3.5457
DSSAE	89.4582	87.5951	85.1944	83.0658	80.9963	3.3939
SNN	90.9361	86.9964	84.2691	83.4583	81.2265	3.7303
MV-LEAP	92.2083	91.3677	91.0024	90.2658	89.1068	1.1724
CGFTSVM	92.9917	91.6658	91.1257	89.9965	89.0026	1.5340
AI-RGEBS	93.2256	93.0013	92.8569	92.0258	91.8659	0.6096

Table 5
The comparison of *F1-score*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
AI-BLS	85.9375	85.2554	83.5214	81.6639	78.2698	3.0872
AI-GEBS	91.2569	90.8561	89.3691	89.6658	89.0036	0.9760
S-RGEBS	87.4545	86.9663	85.2658	84.2658	82.2528	2.1069
MS-RGEBS	91.7115	91.0036	90.3329	91.0258	90.2338	0.6006
AI-RGEBS	94.6691	94.0058	93.8858	93.5356	93.0859	0.8570

Table 6
The comparison of *G-mean*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
AI-BLS	84.2569	83.2269	81.0025	80.221	79.9919	1.9008
AI-GEBS	91.0258	89.9963	89.6621	88.2569	89.2214	1.0168
S-RGEBS	88.2589	87.2698	87.2658	85.2698	83.2598	2.0005
MS-RGEBS	91.2236	90.3552	89.9881	89.7715	89.3698	0.7022
AI-RGEBS	93.2256	93.0013	92.8569	92.0258	91.8659	0.6096

to correspond *F1-score* and *G-mean* is shown in Fig. 6. It is illustrated by Fig. 6(a) that higher *F1-scores* (more than 0.9) are distributed in specific regions, indicated by the red and dark orange colors. where $\alpha = 0.4$ and $\beta = 0.2$, or $\alpha = 0.2$ and $\beta = 0.3$ provide higher *F1-scores*. It is illustrated by Fig. 6(b) that higher *G-mean* (more than 0.85) are distributed in specific regions, indicated by the red and dark orange colors. where $\alpha = 0.7$ and $\beta = 0.6$, or $\alpha = 0.2$ and $\beta = 0.3$ provide higher *G-mean*. The combined Fig. 6(a) and Fig. 6(b) for the balanced parameter choices $\alpha = 0.2$ and $\beta = 0.3$ gives higher diagnostic performance for the overall model.

To validate the effectiveness of the method proposed in this paper, multiple imbalance rates of normal samples and fault samples were set on the training set of the penicillin fermentation process dataset, and then based on the different classification models mentioned above on the test set, *F1-score* and *G-mean* results are shown in Table 3 and Table 4, respectively, where, SD denotes the standard deviation.

From Table 3 and Table 4, when the imbalance rate is increasing, the proposed method in this paper shows a decreasing trend in both *F1-score* and *G-mean* values, which is due to the extreme sample imbalance. Compared with other comparative methods, the accuracy of the proposed method varies less and more stable within a certain range. Even though the imbalance ratio between normal and fault samples reaches 50:1, that accuracy of the fault classification of penicillin fermentation process stays within a small range of variation after utilizing the sample balanced dataset generated by the proposed method in this paper. Meanwhile, more accuracy is achieved for MV-LEAP and CGFTSVM methods for imbalanced samples than other methods. The SD value reflects the dispersion degree of the fault diagnosis rate for a fault diagnosis method at different imbalance rates, i.e., it reflects the robustness of the fault diagnosis method. The proposed AI-RGEBS has higher fault accuracy than other methods and is much more robust under different imbalance rates.

To further validate the effectiveness of AI-RGEBS in improving the accuracy of model fault diagnosis during penicillin fermentation, abscission experiments were conducted on AI-RGEBS. The fault diagnosis accuracy of the model was compared with AI-BLS, AI-GEBS, S-RGEBS and MS-RGEBS. The results are shown in Table 5 and Table 6. MS-RGEBS achieves higher fault diagnosis accuracy than S-RGEBS, thanks to the enhancement of SMOTE using Mahalanobis Distance. The results show that the fault diagnosis accuracy of AI-RGEBS is significantly improved compared to other comparative methods under five imbalance ratios. AI-RGEBS combines the adaptive correction of imbalance data and robust graph embedding to achieve the best fault diagnosis accuracy, which means that the proposed model is effective.

Confusion matrix for the eight model diagnostics in the PFP is shown in Fig. 7 for when imbalance ratio is 5:1. The rows of the matrix represent the predicted labels, the columns represent the actual labels, and the diagonal elements correspond to the classification performance for each category. The fault diagnosis performance is evaluated using accuracy, which measures the proportion of correctly predicted labels out of all predictions made. From Fig. 7, it is evident that the proposed method performs better in terms of accurately diagnosing faults across all categories, despite the class imbalance. This suggests that the method effectively handles imbalanced datasets and ensures reliable fault diagnosis even for underrepresented fault categories.

Fig. 8 shows the training and fault diagnosis times for different models with samples added in the PFP. According to Fig. 8(a), the proposed method takes less time compared to other algorithms as the number of training samples increases. Because the proposed method is an extension of BLS, the training time is a little bit longer compared to BLS, but the additional time consumed is negligible in terms of the overall fault diagnostic performance. In Fig. 8(b), the proposed method is still less time consuming compared to the other methods in overall view due to the small difference in fault diagnosis time. As a result, AI-RGEBS has a very short training time in achieving better fault diagnosis accuracy compared to other methods. AI-RGEBS (1) refers to the version of AI-RGEBS that does not incorporate incremental learning. It is evident that without incremental learning, AI-RGEBS (1) requires more time to complete the task.

5.2. Semiconductor etching process (SEP)

Semiconductor etching is one of the essential steps in the semiconductor manufacturing process and usually performed in batch production. The most popular semiconductor etching method is plasma etching, which utilizes coils or electrodes to ionize gas into plasma, and achieves etching by taking advantage of the physical impact and chemical reaction that occurs between the plasma and the surface of the wafer (Hirai and Kano, 2015). Schematic of the semiconductor etching process is shown in Fig. 9.

In this case, it was performed on a Lam9600 plasma etching tool, which utilizes inductively coupled Bl_3/Cl_2 plasma to etch $\text{TiN}/\text{Al}-0.5\%$ $\text{Cu}/\text{TiN}/\text{oxide}$ stacks. The key parameters include the line width of the etched Al line, uniformity on the wafer, and oxide loss. The metal etcher was equipped with three sensor systems during the experiment: device status sensor, RF monitor, and optical emission spectrometer. In this case, the device status sensor was used to collect data for experimental validation. The device status sensor collects device data during wafer processing, including 40 process setpoints sampled once per second, such as gas flow, chamber pressure, and RF power.

We use 50 batches normal samples and 8 batches with different types of faults in the semiconductor etching process. In this case, the different fault data are set to be 7200 samples and the same number of samples for each type of faults. Based on the imbalance rate, we choose the number of normal samples for training the model at different imbalance rates. During model training, we use a grid search method to adjust the parameters. We select for testing with 8 types of Fault and Normal samples totally 8100 samples for testing. Each batch sample reaction time is 90 h

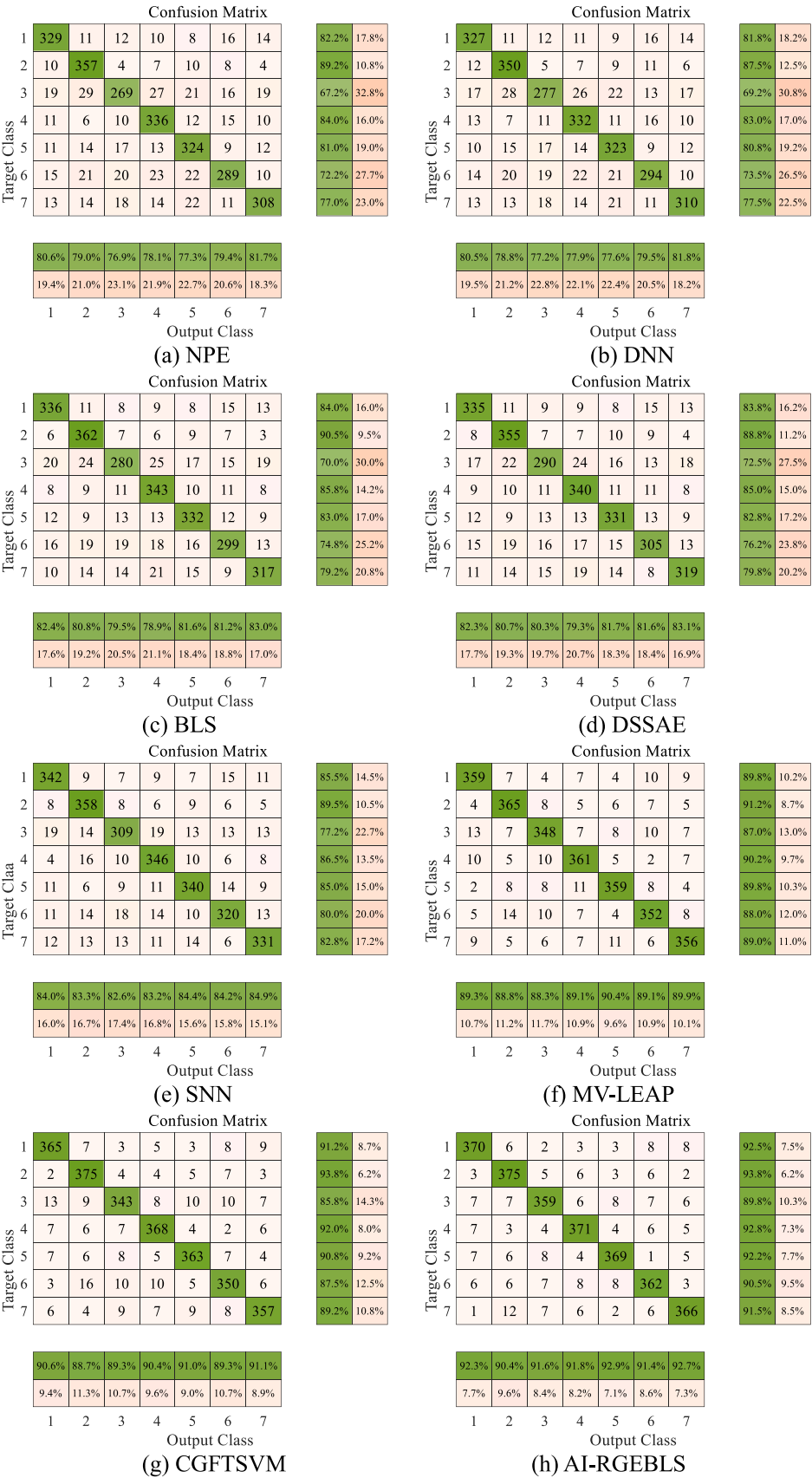


Fig. 7. Confusion matrix for different model diagnostic results in the PFP.

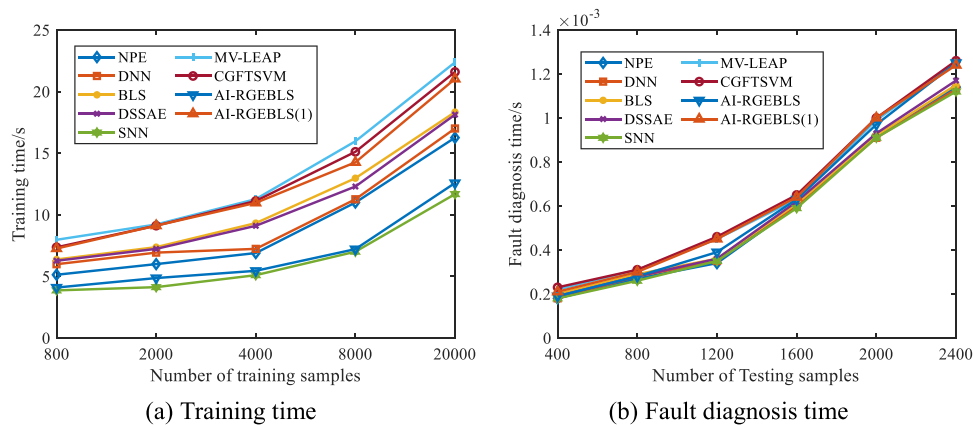


Fig. 8. Training and fault diagnosis times for different models with added samples in PFP.

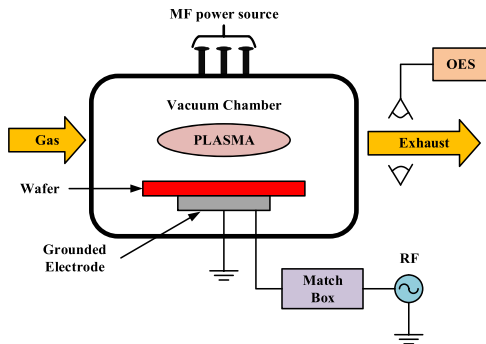


Fig. 9. Schematic of the semiconductor etching process.

Table 7
Semiconductor etching process fault information.

No.	Fault variable	Fault amplitude	Fault time
1	Radio frequency power	−12 W	0–90 h
2	Radio frequency power	+10 W	0–90 h
3	He chuck pressure	+3 Pa	0–90 h
4	Transformer coupled plasma power	+10 W	0–90 h
5	BCl ₃ flow rate	+5 sccm	0–90 h
6	He chuck pressure	−2 Pa	0–90 h
7	Cl ₂ flow rate	−5 sccm	0–90 h
8	Transformer coupled plasma power	−15 W	0–90 h

Table 8
The comparison of *F1-score*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
NPE	73.1258	69.2631	67.2589	65.2665	60.1114	4.8263
DNN	75.2269	71.0023	69.3229	64.4545	61.5598	5.3979
BLS	80.2356	79.2145	77.9865	77.1029	74.2690	2.2866
DSSAE	81.1257	81.1025	79.2154	78.2365	76.2368	2.0637
SNN	82.2568	80.1256	79.6912	77.2369	75.1502	2.7488
MV-LEAP	86.2356	86.0012	85.0015	83.9569	83.1257	1.3265
CGFTSVM	86.6587	85.3256	85.0125	84.1588	82.9636	1.3745
AI-RGEBS	87.3438	86.7256	86.2351	85.8694	85.2568	0.7981

and sampling time is 1 h, the faults are introduced from the reaction beginning to the end, the details of the fault batches are shown in Table 7.

In similar manner to the penicillin fermentation process, the optimization problem shown in formula (14) exists for AI-RGEBS in the semiconductor etching process, where the equilibrium parameters $\alpha = 0.5$ and $\beta = 0.4$ are found through a grid search.

The F1-score and G-mean results of different methods for

Table 9
The comparison of *G-mean*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
NPE	74.2658	72.9963	70.2653	68.7356	65.2213	3.5791
DNN	75.1741	73.2569	72.1543	70.2563	67.8596	2.8073
BLS	77.2569	75.9946	74.2261	72.1654	70.5596	2.7309
DSSAE	81.5866	80.1254	79.2641	77.9687	76.9912	1.7993
SNN	81.9965	80.2655	79.2254	78.3691	77.0021	1.8928
MV-LEAP	85.9014	85.0124	84.7115	83.9375	80.3125	2.1648
CGFTSVM	86.2673	85.2250	84.9691	84.0010	82.3125	1.4906
AI-RGEBS	87.9938	87.3125	86.5564	85.9994	85.0028	1.1576

Table 10
The comparison of *F1-score*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
AI-BLS	81.3654	79.9658	77.2658	76.0254	73.2650	3.2076
AI-GEBS	85.2655	84.2569	83.2561	82.9658	81.0089	1.5913
S-RGEBS	85.9658	84.0257	83.2685	81.9963	80.2658	2.1418
MS-RGEBS	86.0254	85.2654	84.5269	83.9568	81.9587	1.5451
AI-RGEBS	87.3438	86.7256	86.2351	85.8694	85.2568	0.7981

Table 11
The comparison of *G-mean*.

Imbalance ratio	2:1	5:1	10:1	20:1	50:1	SD
AI-BLS	81.2631	80.8859	78.9637	76.9931	76.0012	2.3206
AI-GEBS	86.5896	86.0013	85.6587	84.2654	82.0215	1.8256
S-RGEBS	85.0124	85.6998	85.0032	83.2651	82.0125	1.5177
MS-RGEBS	86.9205	86.9993	85.2689	85.0029	83.6984	1.3948
AI-RGEBS	87.9938	87.3125	86.5564	85.9994	85.0028	1.1576

semiconductor etching process are shown in Table 8 and Table 9.

Through comparing the performance of the algorithms under different imbalance ratios, it is obvious that AI-RGEBS has the best *F1-score* and *G-mean* under all imbalance ratios, which indicates that the method exhibits excellent stability and classification performance in dealing with the category imbalance problem. AI-RGEBS can classify efficiently under various imbalance ratios, which not only performs well at low imbalance ratios, but also maintains excellent classification performance at very high imbalance ratios. In contrast, other algorithms have significantly lower performance when faced with high imbalance ratios, which makes it difficult to maintain an efficient classification capability.

Overall, AI-RGEBS performs well in handling imbalanced data, no matter how the imbalance ratio changes, its *F1-score* and *G-mean* are

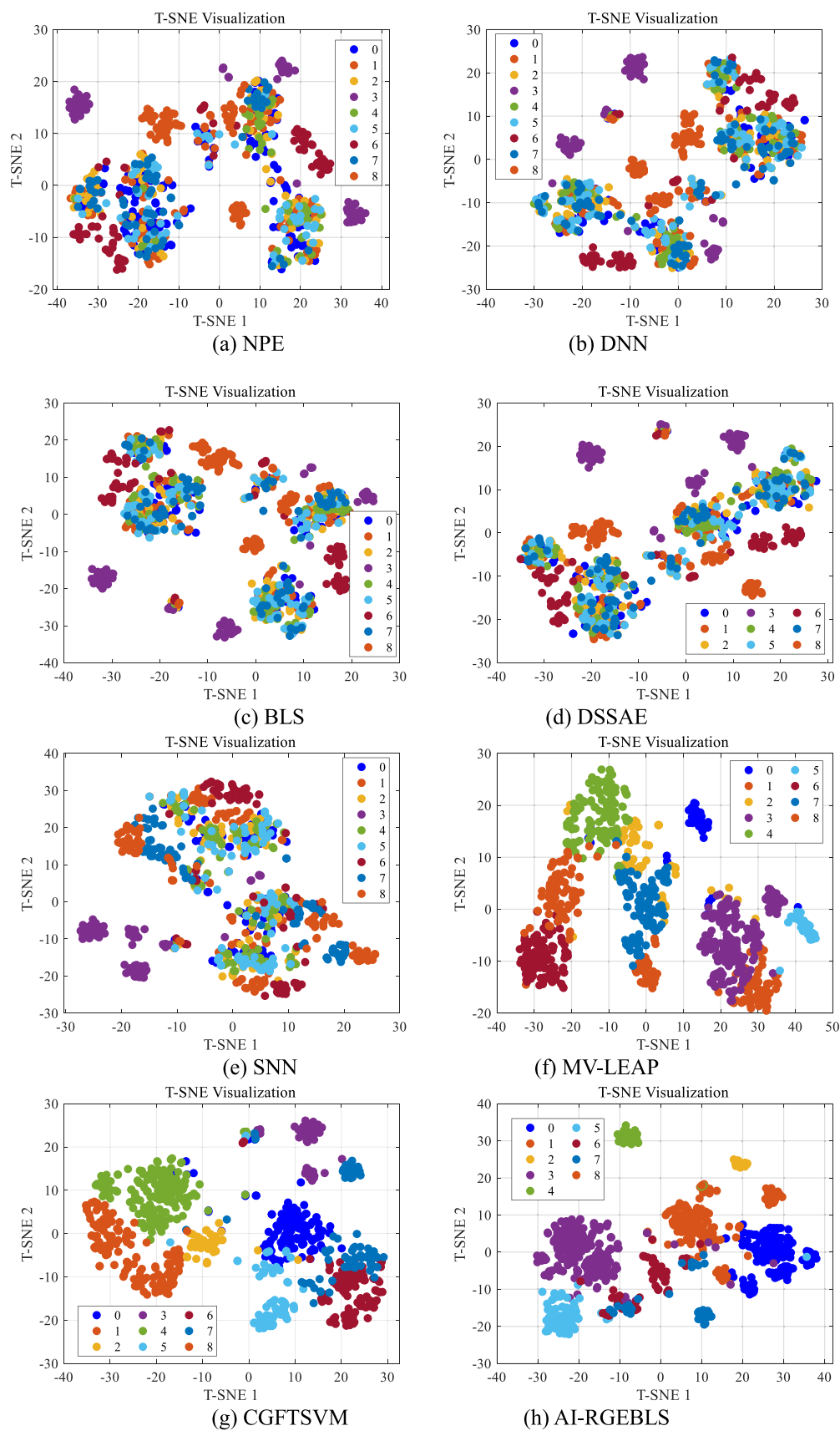


Fig. 10. T-SNE visualization plots of extracted features from different models in the SEP.

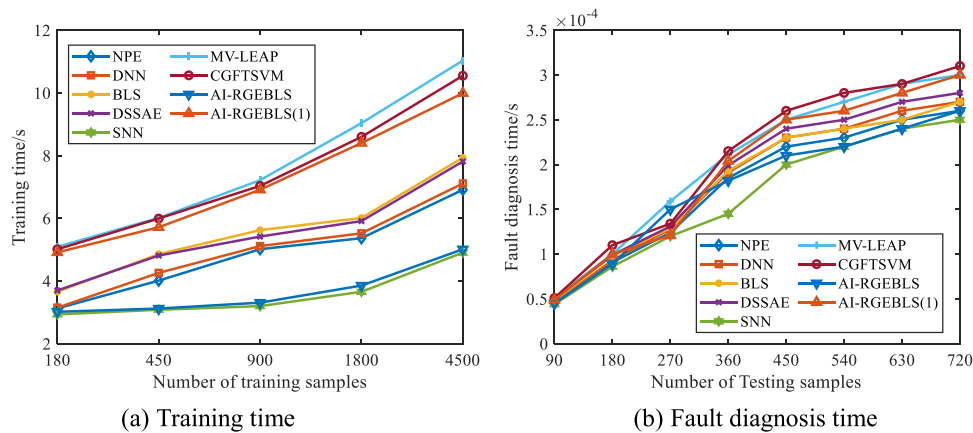


Fig. 11. Training and fault diagnosis times for different models with added samples in SEP.

always kept at a high level, which is significantly better than other algorithms. The superior performance is attributed to the adaptive correction of imbalanced data, which makes AI-RGEBS better able to cope with the challenges of imbalanced datasets.

The ablation experiments of the proposed method are shown in Table 10 and Table 11. The experiment mainly compares the performance of five methods, AI-RGEBS, MS-RGEBS, S-RGEBS, AI-GEBS, and AI-BLS, in terms of *F1-score* and *G-mean*.

The results of the ablation experiment clearly show that AI-RGEBS performs well on imbalanced data, with both *F1-score* and *G-mean* significantly higher than other methods. This shows that our improved algorithm is very effective and significantly improves its classification performance on imbalanced datasets. Specifically, the improved feature selection method enables AI-RGEBS to identify and utilize key features more efficiently, thus improving the classification performance. The optimized sample balancing processing method enables AI-RGEBS to maintain efficient classification performance even under extreme unbalanced conditions. Since MS-RGEBS has higher fault diagnosis accuracy than S-RGEBS due to the improvement of SMOTE by Mahalanobis Distance. The classifier optimization significantly improves the overall performance of AI-RGEBS, so that it performs well under all imbalance ratios.

The T-SNE visualization of the features extracted from the different models is shown in Fig. 10. where 0 indicates normal samples, and 1–8 indicates different fault samples.

Fig. 10 (a) to (e) display the results for NPE, DNN, BLS, DSSAE, and SNN respectively, it can be observed that while some of the faults are distinguishable, the overall performance is limited. Consequently, their abilities to provide reliable fault diagnosis across all categories are compromised. Fig. 10 (f) and (g) illustrate the results for MV-LEAP and CGFTSVM, the performance improves noticeably. The fault diagnosis results are comparatively better, but there are still some limitations in separating fault categories and achieving clear boundaries, especially when a class overlap occurs. Moreover, AI-RGEBS exhibits stronger clustering of the imbalanced data, resulting in better fault diagnosis accuracy across both majority and minority classes. This means that the effectiveness of AI-RGEBS in tackling the imbalance issue compared to other models. By comparing all methods shown in Fig. 10, AI-RGEBS outperforms the others in diagnosing faults within imbalanced data scenarios.

The training and fault diagnosis time for different models with added samples in SEP is shown in Fig. 11. According to Fig. 11(a), the proposed method requires less time than other algorithms as the number of training samples increases. In Fig. 8(b), the proposed method still outperforms other methods in terms of time consumption. In summary, AI-RGEBS not only has an advantage in training time, but also achieves higher fault diagnosis accuracy. Since AI-RGEBS (1) does not involve

incremental learning, its structure requires more time to train and fault diagnosis.

6. Conclusion

In this paper, we proposed an adaptive imbalance robust graph embedding broad learning system (AI-RGEBS). AI-RGEBS utilizes SMOTE with improved Mahalanobis distance and niche technique to adaptively correct imbalance samples. Meanwhile, the graph embedding and $L_{2,1}$ paradigm constraint enables the model to extract local manifold information and enhance the robustness of the model. The model simultaneously considers the accuracy and time consumption of fault diagnosis results in imbalanced batch process data. In addition, AI-RGEBS model with incremental learning dynamically adds enhancement nodes in an incremental window, which avoids the need to re-train the whole AI-RGEBS system from zero, and enables real time adjustments to the network structure and updating parameters. It is confirmed that the AI-RGEBS model has significant advantages in terms of time consumption and better accuracy for fault diagnosis of imbalanced batch process data. Since the introduction of Mahalanobis distance increases the computational cost, in future research, we will focus on how to reduce the overhead of Mahalanobis distance computation through improving or introducing efficient algorithms.

CRedit authorship contribution statement

Xiaoqiang Zhao: Funding acquisition, Formal analysis, Data curation, Conceptualization. **Kai Liu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision. **Yongyong Hui:** Project administration, Methodology, Investigation, Funding acquisition.

Declaration of Competing Interest

No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Acknowledgments

This research work has been awarded by the National Natural Science Foundation of China (62263021), Industrial Support Project of Education Department of Gansu Province (2023CYZC-24), Gansu Province University Young Doctor Support Project (2024QB-037),

Lanzhou Youth Science and Technology Innovation Program (2023-QN-36) and Science and Technology Project of Gansu Province (24JRRA172).

References

- Arunthavanathan, R., Khan, F., Ahmed, S., Imtiaz, S., 2021a. An analysis of process fault diagnosis methods from safety perspectives. *Comput. Chem. Eng.* 145, 107197.
- Arunthavanathan, R., Khan, F., Ahmed, S., Imtiaz, S., 2021b. A deep learning model for process fault prognosis. *Process Saf. Environ. Prot.* 154, 467–479.
- Birol, G., Ündey, C., Cinar, A., 2002. A modular simulation package for fed-batch fermentation: penicillin production. *Comput. Chem. Eng.* 26, 1553–1565.
- Chang, P., Xu, Y., Meng, F., Xiong, W., 2023. Fault detection in wastewater treatment process using broad slow feature neural network with incremental learning ability. *IEEE Trans. Ind. Inform.*
- Chen, C.P., Liu, Z., 2017. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 10–24.
- Chen, H., Cen, J., Yang, Z., Si, W., Cheng, H., 2022. Fault diagnosis of the dynamic chemical process based on the optimized CNN-LSTM network. *ACS Omega* 7, 34389–34400.
- Cirrincione, G., Kumar, R.R., Mohammadi, A., Kia, S.H., Barbiero, P., Ferretti, J., 2020. Shallow versus deep neural networks in gear fault diagnosis. *IEEE Trans. Energy Convers.* 35, 1338–1347.
- Dong, Y., Jiang, H., Jiang, W., Xie, L., 2024a. Dynamic normalization supervised contrastive network with multiscale compound attention mechanism for gearbox imbalanced fault diagnosis. *Eng. Appl. Artif. Intell.* 133, 108098.
- Dong, Y., Jiang, H., Wang, X., Mu, M., Jiang, W., 2024c. An interpretable multiscale lifting wavelet contrast network for planetary gearbox fault diagnosis with small samples. *Reliab. Eng. Syst. Saf.* 251, 110404.
- Dong, Y., Jiang, H., Mu, M., Wang, X., 2024b. Multi-sensor data fusion-enabled lightweight convolutional double regularization contrast transformer for aerospace bearing small samples fault diagnosis. *Adv. Eng. Inform.* 62, 102573.
- Fan, S., Zhang, X., Song, Z., 2021. Imbalanced sample selection with deep reinforcement learning for fault diagnosis. *IEEE Trans. Ind. Inform.* 18, 2518–2527.
- Feng, S., Ren, W., Han, M., Chen, Y.W., 2019. Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective. *Neural Netw.* 117, 179–190.
- Graa, O., Reikik, I., 2019. Multi-view learning-based data proliferator for boosting classification using highly imbalanced classes. *J. Neurosci. Methods* 327, 108344.
- Gu, S., Chen, J., Xie, L., 2023. Automatic segmentation of batch processes into multi-local state-space models for fault detection. *Chem. Eng. Sci.* 267, 118274.
- He, Y.-L., Yan, X., Zhu, Q.-X., 2019. Novel pattern recognition using bootstrap-based discriminant locality-preserving projection and its application to fault diagnosis. *Ind. Eng. Chem. Res.* 58, 17906–17917.
- Hirai, T., Kano, M., 2015. Adaptive virtual metrology design for semiconductor dry etching process through locally weighted partial least squares. *IEEE Trans. Semicond. Manuf.* 28, 137–144.
- Hu, W., Wang, Y., Li, Y., Wan, X., Gopaluni, R.B., 2024. A multi-feature-based fault diagnosis method based on the weighted timeliness broad learning system. *Process Saf. Environ. Prot.* 183, 231–243.
- 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.
- Jiang, L., Ge, Z., Song, Z., 2017. Semi-supervised fault classification based on dynamic Sparse Stacked auto-encoders model. *Chemom. Intell. Lab. Syst.* 168, 72–83.
- Jiang, X., Ge, Z., 2020. Data augmentation classifier for imbalanced fault classification. *IEEE Trans. Autom. Sci. Eng.* 18, 1206–1217.
- Jing, X.-Y., Zhang, X., Zhu, X., Wu, F., You, X., Gao, Y., Shan, S., Yang, J.-Y., 2019. Multiset feature learning for highly imbalanced data classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 139–156.
- Khan, F., Rathnayaka, S., Ahmed, S., 2015. Methods and models in process safety and risk management: Past, present and future. *Process Saf. Environ. Prot.* 98, 116–147.
- Khan, S., Yairi, T., 2018. A review on the application of deep learning in system health management. *Mech. Syst. Signal Process.* 107, 241–265.
- Khan, S.H., Hayat, M., Bennamoun, M., Sohail, F.A., Togneri, R., 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 3573–3587.
- Kumari, A., Tanveer, M., Lin, C.-T., 2024. Class Probability and Generalized Bell Fuzzy Twin SVM for Imbalanced Data. *IEEE Trans. Fuzzy Syst.*
- Li, J., Zhu, Q., Wu, Q., Fan, Z., 2021. A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. *Inf. Sci.* 565, 438–455.
- Liu, C., Wang, K., Ye, L., Wang, Y., Yuan, X., 2021. Deep learning with neighborhood preserving embedding regularization and its application for soft sensor in an industrial hydrocracking process. *Inf. Sci.* 567, 42–57.
- Lu, J., Cui, R., Li, S., 2023. An imbalanced sample intelligent fault diagnosis method using data enhancement and improved broad learning system. *Meas. Sci. Technol.* 34, 075106.
- Men, J., Zhao, C., 2023. An adaptive imbalance modified online broad learning system-based fault diagnosis for imbalanced chemical process data stream. *Expert Syst. Appl.* 234, 121159.
- Mou, M., Zhao, X., 2022. Incipient fault detection and diagnosis of nonlinear industrial process with missing data. *J. Taiwan Inst. Chem. Eng.* 132, 104115.
- Pan, Z., Wang, Y., Yuan, X., Yang, C., Gui, W., 2021. A classification-driven neuron-grouped SAE for feature representation and its application to fault classification in chemical processes. *Knowl.-Based Syst.* 230, 107350.
- Park, J.-G., Jo, S., 2016. Approximate Bayesian MLP regularization for regression in the presence of noise. *Neural Netw.* 83, 75–85.
- Pu, X., Li, C., 2021. Online semisupervised broad learning system for industrial fault diagnosis. *IEEE Trans. Ind. Inform.* 17, 6644–6654.
- Qian, J., Song, Z., Yao, Y., Zhu, Z., Zhang, X., 2022. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemom. Intell. Lab. Syst.* 231, 104711.
- Razavi-Far, R., Farajzadeh-Zanjani, M., Saif, M., 2017. An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors. *IEEE Trans. Ind. Inform.* 13, 2758–2769.
- Shah, M.Z.H., Hu, L., Ahmed, Z., 2022. Modified LPP based on Riemannian metric for feature extraction and fault detection. *Measurement* 193, 110923.
- Sobhani-Tehrani, E., Talebi, H.A., Khorasani, K., 2014. Hybrid fault diagnosis of nonlinear systems using neural parameter estimators. *Neural Netw.* 50, 12–32.
- Tan, L., Huang, T., Liu, J., Li, Q., Wu, X., 2023. Deep adversarial learning-system for fault diagnosis in fused deposition modeling with imbalanced data. *Comput. Ind. Eng.* 176, 108887.
- Wang, C., Yang, J., Zhang, B., 2024c. A fault diagnosis method using improved prototypical network and weighting similarity-Manhattan distance with insufficient noisy data. *Measurement*, 114171.
- Wang, C., Yang, J., Jie, H., Tian, B., Zhao, Z., Chang, Y., 2024b. An uncertainty perception metric network for machinery fault diagnosis under limited noisy source domain and scarce noisy unknown domain. *Adv. Eng. Inform.* 62, 102682.
- Wang, C., Tian, B., Yang, J., Jie, H., Chang, Y., Zhao, Z., 2024a. Neural-transformer: A brain-inspired lightweight mechanical fault diagnosis method under noise. *Reliab. Eng. Syst. Saf.* 251, 110409.
- Wang, D., Chen, J., 2018. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 26, 1702–1726.
- Wang, Y., Pan, Z., Yuan, X., Yang, C., Gui, W., 2020. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans.* 96, 457–467.
- Yin, M., Li, J., Shi, Y., Qi, C., Li, H., 2024. Fusing logic rule-based hybrid variable graph neural network approaches to fault diagnosis of industrial processes. *Expert Syst. Appl.* 238, 121753.
- Yoon, S., MacGregor, J.F., 2001. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *J. Process Control* 11, 387–400.
- Zhang, N., Xu, Y., Zhu, Q.-X., He, Y.-L., 2023a. Novel regularization double preserving integrated with neighborhood locality projections for fault diagnosis. *IEEE Trans. Ind. Inform.* 19, 10478–10488.
- Zhang, X., Fan, S., Song, Z., 2023b. Reinforcement learning-based cost-sensitive classifier for imbalanced fault classification. *Sci. China Inf. Sci.* 66, 212201.
- Zhang, Y., Peng, P., Liu, C., Xu, Y., Zhang, H., 2022. A sequential resampling approach for imbalanced batch process fault detection in semiconductor manufacturing. *J. Intell. Manuf.* 1–16.
- Zhao, X., Liu, K., Hui, Y., 2023. Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm. *Trans. Inst. Meas. Control* 45, 89–103.
- Zheng, S., Zhao, J., 2020. A new unsupervised data mining method based on the stacked autoencoder for chemical process fault diagnosis. *Comput. Chem. Eng.* 135, 106755.
- Zhu, J., Wang, Y., Huang, C.-G., Shen, C., Chen, B., 2023. A new incremental learning for bearing fault diagnosis under noisy conditions using classification and feature level information. *IEEE Trans. Instrum. Meas.*
- Zuo, L., Zhang, L., Zhang, Z.-H., Luo, X.-L., Liu, Y., 2021. A spiking neural network-based approach to bearing fault diagnosis. *J. Manuf. Syst.* 61, 714–724.



Quality prediction of multi-stage batch process based on integrated ConvBiGRU with attention mechanism

Kai Liu^{1,2} · Xiaoqiang Zhao^{1,2,3} · Miao Mou^{1,2} · Yongyong Hui^{1,2,3}

Accepted: 30 September 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

It is important for quality prediction and monitoring to ensure the safe operation of the process. When constructing a prediction model, it is crucial to choose appropriate input variables to influence the online prediction performance and quality monitoring. Data-driven techniques have been widely used for prediction and monitoring of quality variables, but there are some difficulties in the application of batch processes, three-dimensional characteristics of data, different initial conditions, and multi-stage characteristics within batches. Therefore, we propose a quality prediction model of multi-stage batch process based on integrated ConvBiGRU with attention mechanism (MI-ConvBiGRU-AM). Firstly, the original 3D data are expanded into 2D time slices by the batch-variable expansion method. Secondly, the 2D time slices are clustered to complete stage identification using the improved affine propagation clustering method based on the design of the Markov chain similarity matrix. At each stage, we select product quality-related modeling variables using the Maximum Relevance Minimum Redundancy (mRMR). Then, the selected variables are used to train a convolutional bi-directional gated recurrent unit with an attention mechanism (ConvBiGRU-AM). Finally, ConvBiGRU-AM model for each stage is integrated together a whole prediction model for the entire process to accomplish quality prediction, and the prediction residuals are utilized for quality monitoring. The validity of the proposed method was verified by Industrial-scale fed-batch fermentation (IFBF) process and the Hot strip mill (HSM) process. For the IFBF process, the model achieved an FDR of 99.73%, FAR of 0.54%, MAE of 0.0043, RMSE of 0.0396, MAPE of 0.0121, and R^2 of 0.9971. For the HSM process, the results were an FDR of 99.95%, FAR of 0.25%, MAE of 0.0053, RMSE of 0.0111, MAPE of 0.1539, and R^2 of 0.9990. These results demonstrate that the proposed method significantly improves prediction accuracy and achieves better quality monitoring compared to existing methods, highlighting its effectiveness for industrial applications.

Keywords Batch process · Quality prediction · Bidirectional gate recurrent unit (BiGRU) · Attention mechanism · Multi-stage

1 Introduction

Batch processes are widely used in industrial production as they have the advantages of small batch sizes, variety, and flexibility in operating batches. In production, batch

processes have a high priority and therefore product quality and process safety must be ensured [1, 2]. Due to variations in raw materials, equipment faults, and the complexity of the process, the quality of the product at the end of each batch varies considerably from the production requirements. The lack of measurement of quality variables during the process and the fact that quality measurements are usually obtained only at the end of a batch, which makes it difficult to measure quality indicators online [3, 4]. Therefore, it is critical to construct quality monitoring and prediction model not only to control the quality well, but also for the safe production of the batch [5–8].

The core of quality prediction is to build a prediction model with high accuracy, easy implementation, and steady performance. Regarding the prediction modeling mainly

✉ Xiaoqiang Zhao
xqzhao@lut.edu.cn

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

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

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

consists of both the fundamental modeling and the data-driven approaches. Due to the large scale and complexity of modern industrial processes, it is difficult to obtain all the process knowledge in a limited time, which limits its promotion and application in engineering practice. The data-driven based approach does not consider the process empirical knowledge and builds models based on process data to accomplish the goal of process monitoring. Traditional multivariate statistical analysis methods are widely used in industrial process quality prediction, such as least squares regression (LSR) [9], principal component regression (PCR) [10], and partial least squares regression (PLSR) [11]. Based on this derived multiway principal component regression (MPCR) [12] and multiway partial least squares regression (MPLSR) [13] are mostly used for batch processes. Although the effectiveness of these linear methods has been demonstrated in a variety of applications, these methods may not be able to capture the nonlinear relationship between process variables and quality variables in complex processes. In contrast, artificial neural network methods are more nonlinear feature extraction than multivariate statistics methods. However, for most shallow artificial neural networks, it may be limited to strongly nonlinear representations of industrial processes.

To achieve better nonlinear representation performance, deep learning methods with multiple hidden layers are introduced, such as deep belief networks (DBN) [14], convolutional neural networks (CNN) [15], and deep auto-encoders (DAE) [16]. However, most of these methods assume that the production process is running in a steady state, which ignores the temporal nature of the production process, and thus it is crucial to build deep learning models with temporal order. Due to Long Short-Term Memory (LSTM) [17] and Gated Recurrent Units (GRU) [18] have excellent abilities to process complex dynamic time series. In addition, compared to LSTM, there are fewer parameters in GRU than LSTM, which enables GRU to store and retrieve dynamic time information efficiently. Yao et al. [19] combined GRU with Granger Causality (GC) to extract causal relationships between process variables and construct a quality prediction model for industrial processes. Ma et al. [20] designed a BiGRU structure for dynamic nonlinear soft sensor modeling that makes full use of historical and future information and related features within industrial time series for quality prediction. Li et al. [21] addressed the temporal mismatch between the input time step and the output target due to inertia of the blast furnace ironmaking process by embedding an attention mechanism in the GRU model to accomplish quality prediction. Based on this, we use GRU as the basic quality prediction method.

In constructing neural net models for quality prediction, the selection of proper input variables or features for training the model is the primary issue. Proper variable selection

can effectively eliminate redundant variables in modeling and significantly reduce the computational complexity of the model. In addition, it is possible to minimize the damage to model performance caused by redundant variables or features. Traditional methods of variable selection that depend on process knowledge or operator experience are clearly less efficient for high-dimensional data and without sufficiently reliable empirical knowledge. Sun et al. [22] proposed a new variable selection method for soft sensor applications based on nonnegative garrote (NNG) and artificial neural networks (ANN). Fujiwara et al. [23] proposed a variable selection method that clusters variables into groups of variables based on the correlation between the variables through Nearest Correlation Spectral Clustering (NCSC) and examines whether each group of variables should be used as an input variable. Yao et al. [24] proposed two enhanced Binary Differential Evolution (BDE) algorithms to select variables for nonlinear process soft sensors. However, all these data-driven variable or feature selection methods are not from the perspective of causal relationships between process and quality variables, which makes it difficult to further improve the performance of the model.

In addition, most of the batch process variables exhibit multi-stage relationships as the process moves forward. For the multi-stage characteristics of the batch process, it is necessary to deal with it in The lack of measurement of quality variables during the process and the fact that quality measurements are usually obtained only at the end of a batch, Zhao et al. [25] proposed a statistical modeling strategy for phase division of batch processes. Luo et al. [26] proposed a stage segmentation method based on the warped K-means (WKM) clustering algorithm which divides the whole batch into several operational stages by clustering the trajectory data of stage-sensitive process variables. Peng et al. [27] proposed a batch process quality related monitoring scheme using Gaussian mixture model (GMM) algorithm for stage division and fuzzy affiliation method for transfer identification. Liu et al. [28] proposed a sequential local-based GMM approach to accomplish industrial process stage division. The above methods need to set parameters manually when segmenting time periods, and these parameters have a large impact on the segmentation results. For some complex batch processes which are difficult to determine the parameter values based on the available information, AP clustering can realize the automated division of stages without manually presetting the parameters. Chang et al. [29] considered that the different stages of the batch process have different production characteristics, adopted affinity propagation (AP) to separate the different stages of the production process, and investigated a multi-stage process monitoring framework with the fusion of AP and broad learning system. Zhao et al. [30] divided the batch process into multi-stages by AP algorithm and proposed a multi-stage optimized regularized

neighborhood preserving embedding (ORNPE) algorithm for process monitoring. However, AP clustering is obviously insufficient to measure similarity only by using Euclidean distance, which may affect the accuracy of stage segmentation. Therefore, it is necessary to consider the temporal information of the samples and the neighborhood information, etc., during AP stage segmentation. Furthermore, for each stage modeling should consider that the model input variables or features are different for each stage, while integrate models for each stage as an overall model.

Regarding these issues, we propose a quality prediction model of multi-stage batch process based on integrated ConvBiGRU with attention mechanism (MI-ConvBiGRU-AM). The main contributions are given as follows:

- (1) To enhance process stage identification, we use an improved AP clustering method that Markov chain-based design of similarity matrix. This approach, which is an enhancement of the standard AP method, considers both the temporal and neighborhood information of the samples. It solves the problem that the original AP uses Euclidean distance to measure the similarity insufficiently and makes the clustering results more accurate.
- (2) The method of maximum correlation-minimum redundancy is used to mine the internal relationship between process variables and quality variables for variable selection, seeking to improve the interpretability of subsequent deep learning methods for quality prediction.
- (3) Under the coexistence of nonlinear and dynamic characteristics of process data, ConvBiGRU extracts the causal relationship between process and quality variables. Through attention mechanism applied to the output layer of ConvBiGRU, which helps the model to better understand the key parts of the input sequence, it focuses the key information on the prediction process.
- (4) For multi-stage batch processes, it is possible to integrate the ConvBiGRU-AM established at each stage into the MI-ConvBiGRU-AM model for the whole production process. Through Industrial-scale fed-batch fermentation process and the Hot strip mill process validation the proposed MI-ConvBiGRU-AM can effectively accomplish quality prediction and monitoring for the batch process.

The structure of the paper is as follows. In Sect. 2, a brief overview of Affinity Propagation (AP) and Gate Recurrent Unit (GRU) is given, while our proposed ConvBiGRU with attention mechanism for multi-stage batch process quality prediction is detailed in Sect. 3. In Sect. 4, two case studies are used to validate the proposed algorithm, and our conclusions are drawn in Sect. 5.

2 Preliminaries

In this section, AP clustering and GRU are briefly introduced as the basis of the proposed integrated quality prediction algorithm in this paper.

2.1 Affinity Propagation (AP)

Affinity Propagation [31, 32] clustering is a clustering algorithm with "information transfer" between data. Compared with traditional clustering algorithms, it has a great improvement in clustering performance and efficiency, and the biggest strength is that there is no need to specify the number of clusters before running the algorithm, thus avoiding the influence of human factors on the accuracy of clustering. AP clustering regards all sample points as nodes of the network and automatically finds representative cluster centers to maximize the sum of similarities from all sample points to the nearest class representative point. It is realized by firstly picking out some special objects that are called exemplars, and then associating each left object to its nearest exemplar. The objective is to maximize formula (1):

$$z = \sum_{i=1}^n s(i, c_i) \quad (1)$$

where, $s(i, c_i)$ denotes the similarity between the sample x_i and its nearest sample x_{c_i} . It is a difficult problem to solve the optimization problem of formula (1). Thus, by introducing a constraint function, which transforms the optimization problem into an unconstrained optimization problem, as shown in formula (2):

$$z = \sum_{i=1}^n s(i, c_i) + \sum_{j=1}^n \delta_j(c) \quad (2)$$

where, $c = (c_1, c_2, \dots, c_i, \dots, c_n)$. $\delta_j(c)$ is the constraint function defined as in formula (3):

$$\delta_j(c) = \begin{cases} -\infty, & \text{if } c_i \neq j, \text{ but } \exists c_i = j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

here, c_i indicates the exemplar of the data point i . A value of $c_i = j$ for $i \neq j$ indicates that object i is assigned to a cluster with object j as its exemplar. A value of $c_i = j$ indicates that object j is an exemplar. The introduction of penalty term $\delta_j(c)$ is to avoid such a situation that object i chooses object j as its exemplar, but object j is not an exemplar at all.

AP clustering achieves the interaction of information through the iteration of responsibility and availability. Responsibility $r(i, j)$ is sent from variable node c_i to function node δ . It indicates how strongly object i wants to choose candidate exemplar j as its exemplar. Responsibility is calculated from formula (4):

$$r(i, j) \leftarrow s(i, j) - \max_{j', j' \neq j} \{a(i, j') + s(i, j')\} \quad (4)$$

where, $s(i, j)$ is similarity between two different data points i and j . Similarity is captured by the negative Euclidean distance: $s(i, j) = -\|x_i - x_j\|^2$.

Availability $a(i, j)$ is sent from variable node c_i to function node δ_j . It reflects the accumulated evidence for how well suited it would be for point i to choose point j as its exemplar. Availability is calculated from formula (4):

$$a(i, j) \leftarrow \min \left\{ r(j, j) + \sum_{i', i' \notin \{i, j\}} \max \{0, r(i', j)\} \right\} \quad (5)$$

Responsibility and availability update until convergence, the clustering result is: $\hat{c}_1 = (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n)$, which can be obtained from formula (6) [33]:

$$\hat{c}_1 = \arg \max_j \{a(i, j) + r(i, j)\} \quad (6)$$

2.2 Gate Recurrent Unit (GRU)

While dealing with temporal data, recurrent neural networks have great advantages. In particular, the output of the current moment in the network depends on the network's memory of previous information, which means that the input of the hidden layer not only contains the output of the input layer at the current moment, but also the output of the hidden layer at the previous moment. Based on RNN, Hochreiter et al. [34] proposed LSTM neural network. It adds three kinds of gate structures besides RNN, and the number of neurons in each gate is the same, which controls the transmission of information flow and mitigates the problem of gradient vanishing or explosion in RNN. Cho et al. [35] proposed gated recurrent unit neural network, GRU achieves equivalent results compared to LSTM at the same time, with fewer gate structures, simpler model, fewer network parameters and faster convergence. GRU network structure basically consists of reset and update gates. A general structure of GRU cell is shown in Fig. 1 [36].

The update formula is shown in (7)–(10) [37]:

$$z_t = \sigma(W^z x_t + U^z h_{t-1}) \quad (7)$$

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \quad (8)$$

$$h_t = z_t \odot \tilde{h}_t + (1 - z_t) h_{t-1} \quad (9)$$

$$\tilde{h}_t = \tanh(W x_t + r_t \odot U h_{t-1}) \quad (10)$$

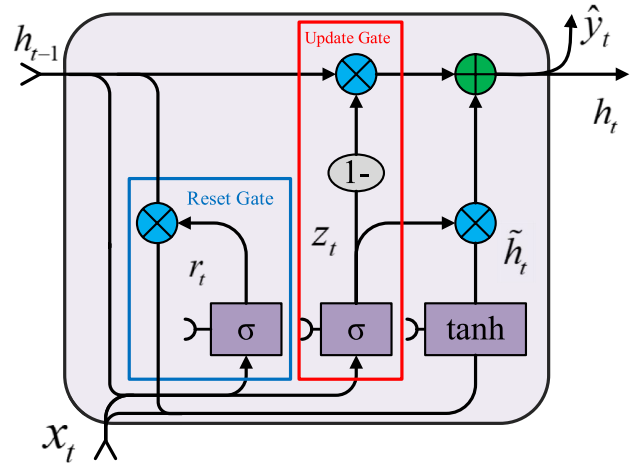


Fig. 1 Structure of the GRU network

where, x_t is the input of the current time step, h_{t-1} is the hidden state of the last time unit, h_t is candidate variable, \tilde{h}_t indicates candidate status, σ is the sigmoid function, W^z , W^r and W is the input weights, U^z , U^r and U is the hidden state weights, and \odot is the Hadamard product.

GRU effectively retains dynamic temporal information, making it more suitable for dynamic, nonlinear process quality prediction. However, to ensure the accuracy of modeling, it is necessary to consider both past historical information and future information. Meanwhile, high-dimensional variables of the process may increase the complexity of model training, thus affecting the efficiency and interpretability of the model.

3 The proposed ConvBiGRU based quality prediction framework for multi-stage batch process

In this section, we illustrate the proposed ConvBiGRU-based multi-stage batch process quality prediction framework, which includes stage identification, variable selection, stage modeling, model evaluation, and quality monitoring.

3.1 Stage identification by Improved Affinity Propagation (IAP)

In the case of stage identification of processes through AP clustering, how the similarity matrix is constructed determines the clustering results. Therefore, it is not sufficient to measure between-sample similarity through Euclidean distance alone. To retain more similarity information between the samples, we proposed an IAP clustering based on the design of similarity with finite

Markov chain, while considering the time information and neighborhood information.

The design of similarity matrices usually involves calculating the similarities between a set of objects and constructing a matrix to represent these similarities. A similarity matrix can be constructed efficiently by utilizing the idea of Markov chains, which capture the relationships between objects by transferring probabilities. Markov chain-based similarity matrix [38] is calculated as follows:

Let A be a matrix whose rows are all equal. Since P denotes a stationary probability distribution, by the nature of the Markov chain being irreducible and ergodic, and the Perron-Frobenius theorem. That is $A = [a^T, a^T, \dots, a^T]$, where $a = [a_1, a_2, \dots, a_m]$, then we have $aP = a$. For a Markov chain determined by P , we can form the fundamental matrix $B = (I - P + A)^{-1}$. For a Markov chain, transition matrix P , limiting vector a , and limiting matrix A . Then:

$$\begin{aligned} PB &= BP \\ aB &= a \\ I - B &= A - PB \end{aligned} \quad (11)$$

The transition matrix P is constructed according to formula (12):

$$P_{ij} = \frac{f(D_{ij}; \beta_1, \beta_2)}{\sum_k f(D_{ik}; \beta_1, \beta_2)} \quad (12)$$

Let $l_{ij} = \exp\left(\frac{-D_{ij}}{\beta_1}\right) \exp\left(\frac{-\|i-j\|^2}{\beta_2}\right)$, $f(D_{ij}; \beta_1, \beta_2)$ is obtained according to formula (13):

$$f(D_{ik}; \beta_1, \beta_2) = \begin{cases} l_{ij} + \epsilon, & \text{if } X_i \text{ is among } k \text{ nearest neighbors of } X_j \\ \epsilon, & \text{otherwise} \end{cases} \quad (13)$$

where, ϵ is a small positive number. $D_{ij} = \|X_i - X_j\|_j^2$ be distance between two states. l_{ij} can be viewed as the integration of the locality weight $\exp\left(\frac{-D_{ij}}{\beta_1}\right)$ and the time weight $\exp\left(\frac{-\|i-j\|^2}{\beta_2}\right)$. It means that l_{ij} reflects both the local neighborhood structure and the time information of the states. β_1 and β_2 are regulators, generally, β_1 takes the square root of the mean value of D_{ij} , and β_2 takes the square root of the average Euclidean distance between all pairs of i and j .

Based on the fundamental matrix B , we can obtain the limiting covariance C_{ij} for the number of times in states X_i and X_j in the first N steps by formula (14):

$$C_{ij} = a_j B_{ij} + a_j B_{ji} - a_i a_j - a_i \delta_{ij} \quad (14)$$

where, $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Calculate the similarity matrix by formula (15):

$$S_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} C_{jj}}} \quad (15)$$

For the similarity matrix S_{ij} calculated above, then responsibility and availability of AP can be further written as shown in formula (16):

$$\begin{aligned} r(i, j) &\leftarrow S_{ij} - \max_{j', j' \neq j} \{a(i, j') + A_{ij'}\} \\ a(i, j) &\leftarrow \min \left\{ r(j, j) + \sum_{it, it \notin \{i, j\}} \max \{0, r(it, j)\} \right\} \end{aligned} \quad (16)$$

$r(i, j)$ and $a(i, j)$ update until convergence, the result of clustering is obtained. The process stage identification is performed by clustering the results. For multiple batches with inconsistent phasing results, the number of batches with the most phasing results shall take precedence and fulfill the characteristics of the production process.

3.2 Variable selection by Max-Relevance and Min-Redundancy (mRMR)

For correlation between variables is characterized using mutual information, for two random variables x and y the mutual information is defined according to the probability density function as shown in formula (17):

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy \quad (17)$$

where, $p(x, y)$ is the joint probability distribution function of x and y . $p(x)$ and $p(y)$ are the marginal probability distribution functions of x and y , respectively.

In variable selection, mRMR not only considers the correlation between the process and the quality variables, but also the correlation between the process variables, through finding the subset of process variables from which the correlation with the quality variable is the highest and the correlation between them and other process variables is the lowest. The mRMR method effectively filters out features that have a significant impact on the target variable by enhancing the correlation between features while reducing the redundancy between them, and this selection helps to reduce the impact of noise and outliers on the model. Even if some features are disturbed by noise or outliers, the mRMR method still better preserves the selection of important features by considering their correlation and redundancy together. The selected subset of features is more likely to contain features that are highly correlated with the target variable, thus reducing the risk of model overfitting and enhancing its robustness to noise and outliers.

Found an optimal subset S_m of process variables with m eigenvectors in the original process variables whose index D of maximum relevance is shown in formula (18):

$$\max D(S, c), D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c) \quad (18)$$

where, I denote mutual information.

The features selected based on Max Relevance may have abundant redundancy. When two features are highly dependent, if one feature is removed, their respective classification discriminative ability will not change much. Therefore, the optimal set of variables can be selected by adding the Min Redundancy R as shown in formula (19).

$$\min R(S), R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \quad (19)$$

Based on combining the two constraints of formulas (18) and (19), the simultaneous optimization of D and R by formulas (20).

$$\max \phi(D, R), \quad \phi = D - R \quad (20)$$

where, ϕ is defined operator. Incremental search methods can be used to obtain an approximate optimal solution. It can be written as an optimization problem as in formula (21) [39]:

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} (x_j; x_i) \right] \quad (21)$$

Through the above mRMR method, we select the process variables that have affected the product quality at each stage for modeling. And the process variables affecting product quality are not the same in each stage.

3.3 The proposed ConvBiGRU prediction model with attention mechanism

After the quality related variables are selected, it is necessary to build the prediction model for quality prediction.

The proposed ConvBiGRU-AM prediction model is shown in Fig. 2.

The framework of the proposed ConvBiGRU-AM includes an Input layer, convolutional neural network (CNN) layer, BiGRU layer, Attention layer, and Output layer. Process data is input to the CNN layer through the input layer. CNN layer is focused on feature extraction. To extract features from time-series sensor data, we employ one-dimensional convolutional neural network (1D CNN) in this paper. Multidimensional time series sensor raw data will be Convolved as feature mapping. The number of feature dimensions extracted after convolutional filtering is dependent on the dimensionality of the sensor data, the size of the filter and the convolution step size. The convolution layer uses filter matrix for feature extraction and pooling layer for feature dimensionality reduction to compress the amount of data and parameters and reduce overfitting. The convolutional operation process of 1D-CNN is shown in formula (22):

$$M_i = f(\text{conv}(w_i^T * X) + b_i) \quad (22)$$

where, w_i denotes i th filter kernel matrix, b_i denotes bias of the feature, f denotes nonlinear activation function, conv denotes the convolution operation.

The pooling layer is scaled mapping of the data from the upper convolutional layer, which can significantly reduce the spatial dimension of the input data by sampling the input data through pooling, and the pooling layer operation is shown in formula (23):

$$D_i = f[\beta_i \text{down}(x) + b_i] \quad (23)$$

where, down denotes downsampling function, β_i denotes i th weight of the feature value.

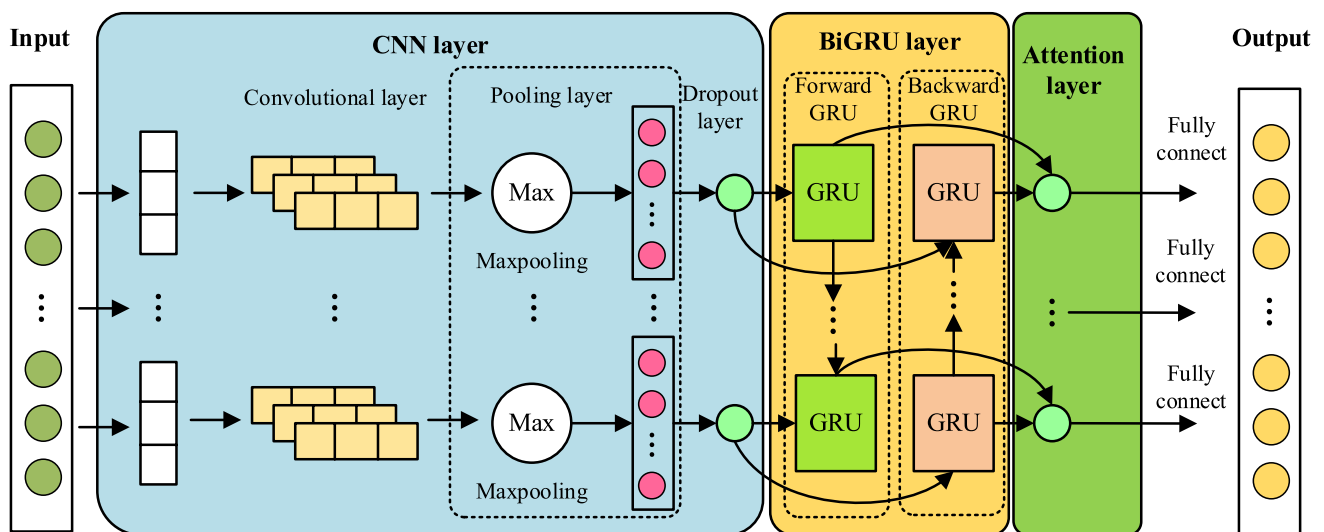


Fig. 2 Flowchart of the proposed framework

The features extracted through the CNN layer are further input to the next layer of the network. Compared to GRU, BiGRU fuses both historical and future information that determines the quality prediction results from state information in both directions. BiGRU consists of two GRUs: a forward GRU model that accepts forward inputs and a backward GRU model that learns the backward inputs. That is described as shown in formulas (24)–(26):

$$h_t^+ = GRU(x_t, h_{t-1}^+) \quad (24)$$

$$h_t^- = GRU(x_t, h_{t-1}^-) \quad (25)$$

$$H_t = h_t^+ \oplus h_t^- \quad (26)$$

where, H_t denotes the output of the BiGRU layer, h_t^+ and h_t^- denote forward and backward outputs. \oplus denotes element-wise sum.

Considering not all the feature vectors output from BiGRU are decisive for quality prediction, the hidden states of different BiGRU layers are normalized and assigned different weights to focus on the important information. For BiGRU output vectors, it is necessary to focus on the importance of different information to predict the model more accurately, therefore through introducing the attention mechanism model [40] as shown in formulas (27)–(29):

$$w_i^t = \frac{\exp(s(H_t, u_{t-1}))}{\sum_{i=1}^T \exp(s(H_i, u_{t-1}))} \quad (27)$$

$$s(H_t, s_{t-1}) = H_t u_{t-1} \quad (28)$$

$$C_t = \sum_{i=1}^T w_i^t H_i \quad (29)$$

where, w_i^t denotes the attention weigh, $s(H_t, u_{t-1})$ represents the scoring function, s represents the weight calculation methods, u_{t-1} denotes the random initialization vector of the previous layer, C_t denotes the output of the model that introduces the attention mechanism.

The attention mechanism helps the model to better focus on important features and ignore unimportant or noisy data, which allows the model to filter out this distracting information more effectively when dealing with outliers. The model that combines the attention mechanism and BiGRU has strong robustness in dealing with noise and outliers.

3.4 The indicators for prediction evaluation and quality monitoring

To evaluate the prediction performance of the method proposed in this paper, we employ several evaluation

indicators to evaluate the model. It includes Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and coefficient of determination R^2 . The specific calculations are shown in formulas (30)–(33):

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \quad (30)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (31)$$

$$MAPE = \frac{100\%}{m} \sum_{i=1}^m \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (32)$$

$$R^2 = 1 - \frac{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}{\frac{1}{m} \sum_{i=1}^m (y_i - \bar{y})^2} \quad (33)$$

where m is the number of prediction points, i is the ordinal number of prediction points, y_i is the actual value, \bar{y} is the average of y_i , \hat{y}_i is the predicted value. MAE and MAPE values are as small as possible and take values greater than 0. RMSE values are smaller to indicate higher accuracy. R^2 describes the ability of the prediction model to fit the actual data curve, the larger the better, and the range of values is $(-\infty, 1)$.

To estimate the operational state of the process variables, it is further monitored by the process variables through the proposed prediction model. Under normal operating conditions, the prediction error of the data obtained by the proposed model should follow normal distribution and can be considered as noise. When faults occur, the residuals will increase due to the correlation of the variables captured by the prediction model being destroyed accordingly. Therefore, the residuals through normal operating conditions according to the 3σ criterion determine a threshold d that can cover almost all the residuals under normal operating conditions. When the real-time residuals of the collected online data exceed this threshold, the process is considered to have faults; otherwise, the process is deemed normal.

For process monitoring, fault detection rate (FDR) is computed as the rate of effective alarmed fault data corresponding to the entire fault dataset. fault false alarm rate (FAR) is the rate of normal being false alarmed as faulty. The higher FDR and lower FAR, the stronger discriminating performance. The calculation of FDR and FAR is as formula (34) and (35):

$$FDR = \frac{\text{Number of fault alarmed data}}{\text{Number of fault data set}} \times 100\% \quad (34)$$

$$\text{FAR} = \frac{\text{Number of fault false alarmed data}}{\text{Number of normal data set}} \times 100\% \quad (35)$$

3.5 The procedure of the proposed quality prediction framework

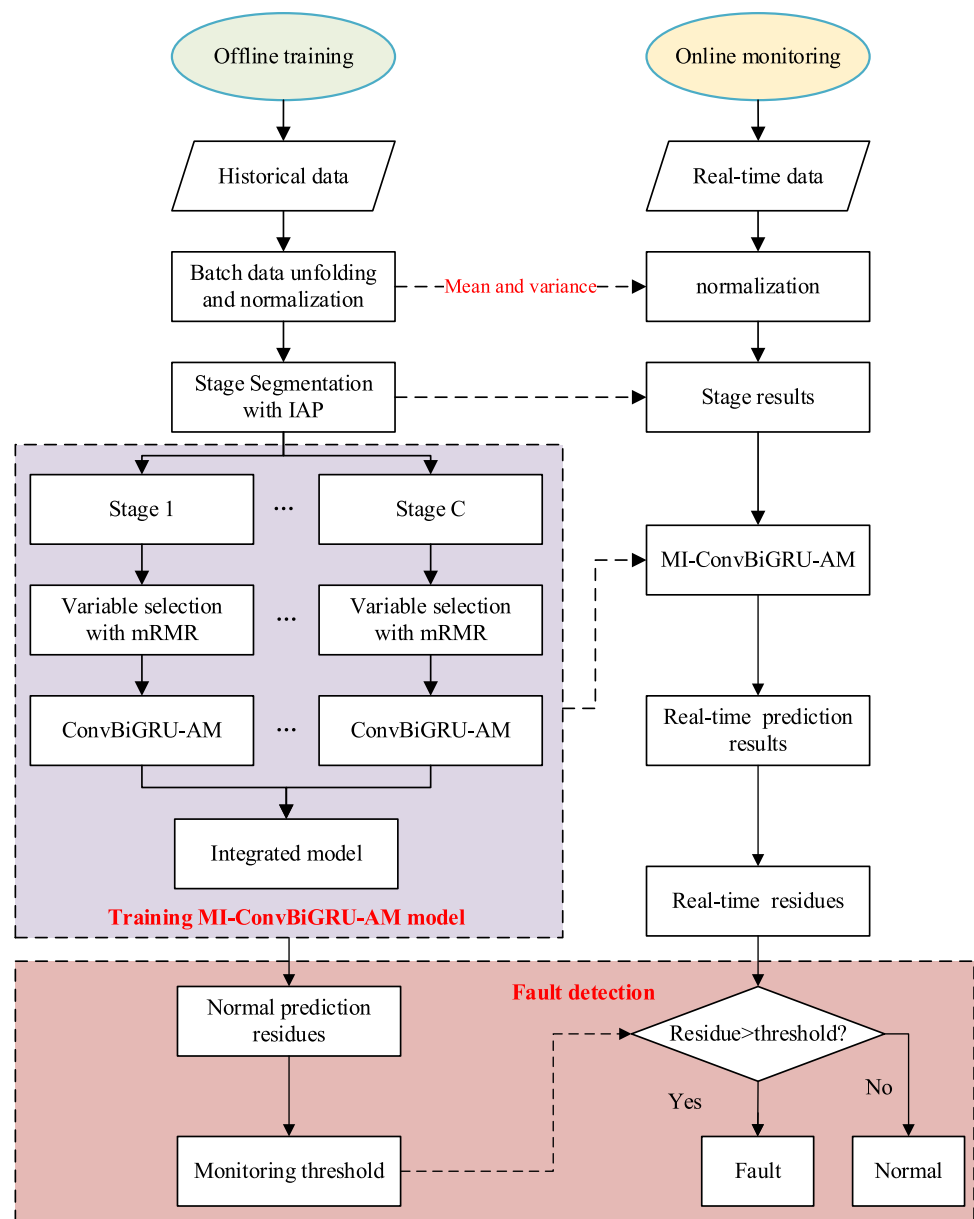
In general, we propose a quality prediction and monitoring framework in this paper as shown in Fig. 3. The framework includes two subsections: A. offline modeling and B. Online monitoring. The specific steps are as follows.

A. Offline training

Step 1: The three-dimensional process data $X (I \times J \times K)$ of the batch process is unfolded and normalized according to the batch-variable unfolding and converted into a two-dimensional time-slice data $X (I \times JK)$, where X denotes the three-dimensional data of batch I , J denotes the variable, and K denotes the sample time. Batch-variable unfolding can attenuate to some extent the nonlinearities of the variables along the direction of the time axis, and highlight the information of the differences along the batch direction. Figure 4 shows the 3D batch data unfolding and normalization.

Step 2: The pre-processed 2D time is divided into C stages of the production process using the IAP in Section 3.1. For multiple batches with inconsistent phasing results,

Fig. 3 The procedures of integrated ConvBiGRU-based quality prediction and monitoring framework



the number of batches with the most phasing results shall take precedence and fulfill the characteristics of the production process.

Step 3: For each stage, it was used to select modeling variables using mRMR in subsection 3.2 and to build ConvBiGRU-AM quality prediction model.

Step 4: The set of ConvBiGRU-AM models constructed at each stage is integrated into a predictive model for the whole process. Calculate the prediction residuals of the offline normal data, and use the 3σ criterion to obtain the control threshold of the prediction residuals.

B. Online monitoring

Step 5: Collect real-time data and normalize them with the mean and variance of the offline historical data.

Step 6: The standardized real-time data is segmented into C stages according to the results of the stages segmented during offline modeling.

Step 7: The real-time data is input into the integrated model MI-ConvBiGRU-AM at stages and is obtained predictions results of product quality. The results of the prediction are evaluated by means of evaluation indicators.

Step 8: Calculate real-time prediction residuals and compare them with offline monitoring thresholds for quality monitoring.

4 Case studies

We conduct two batch process cases involving the Industrial-scale fed-batch fermentation process [41] and the Hot strip mill process [42] to verify the functionality and superiority of our proposed ConvBiGRU with attention mechanism in multi-stage batch process quality prediction. The simulation was achieved with the Python 3.7 software. In this work, the calculation was performed on a computer with the following

specifications: CPU: Intel(R) Core (TM) i5-6200U; RAM: 12.0 GB; GPU: NVIDIA GeForce 930 M.

4.1 Industrial-scale fed-batch fermentation process

Penicillin is an antibiotic widely used in clinical medicine, and its fermentation process is a typical batch process. The fermentation system usually includes major equipment such as fermenters, stirring motors, ventilation equipment, and involves the addition of reactants such as acids, bases, hot water, cold water, substrates, etc., and is equipped with temperature and pH controllers. Under specific production strains and conditions, the penicillin fermentation process leads to the growth and multiplication of the bacteria, and after a certain concentration of the strain is reached, penicillin begins to be produced as a metabolite. The IndPenSim is an industrial-scale penicillin fermentation simulation platform. Compared to traditional benchmark penicillin fermentation platforms, it considers the growth, metabolism, and degradation of large-scale penicillin fermentation besides modeling the necessary online and offline variables. Penicillin production is maximized by specifying multiple control strategies to reduce system fluctuations. The schematic of the Industrial-scale fed-batch fermentation process is shown in Fig. 5. More specific description of the process description and operation can be referenced from corresponding original reference [43, 44].

4.1.1 Description of IndPenSim and datasets

In this case study, it was used 100 batches of dataset as shown in Table 1. In 100 batches, the 90 batches under normal conditions with three different control strategies and 10 batches under fault condition.

In this work, every batch was run for 230 h, sample frequency of 0.2 h. The 22 main process variables were selected as shown in Table 2. Penicillin concentration is a

Fig. 4 Batch data 3D unfolding

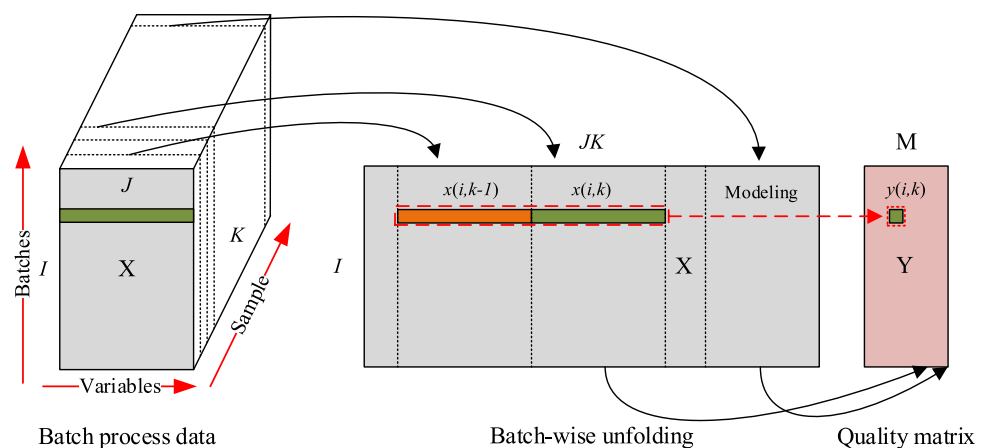
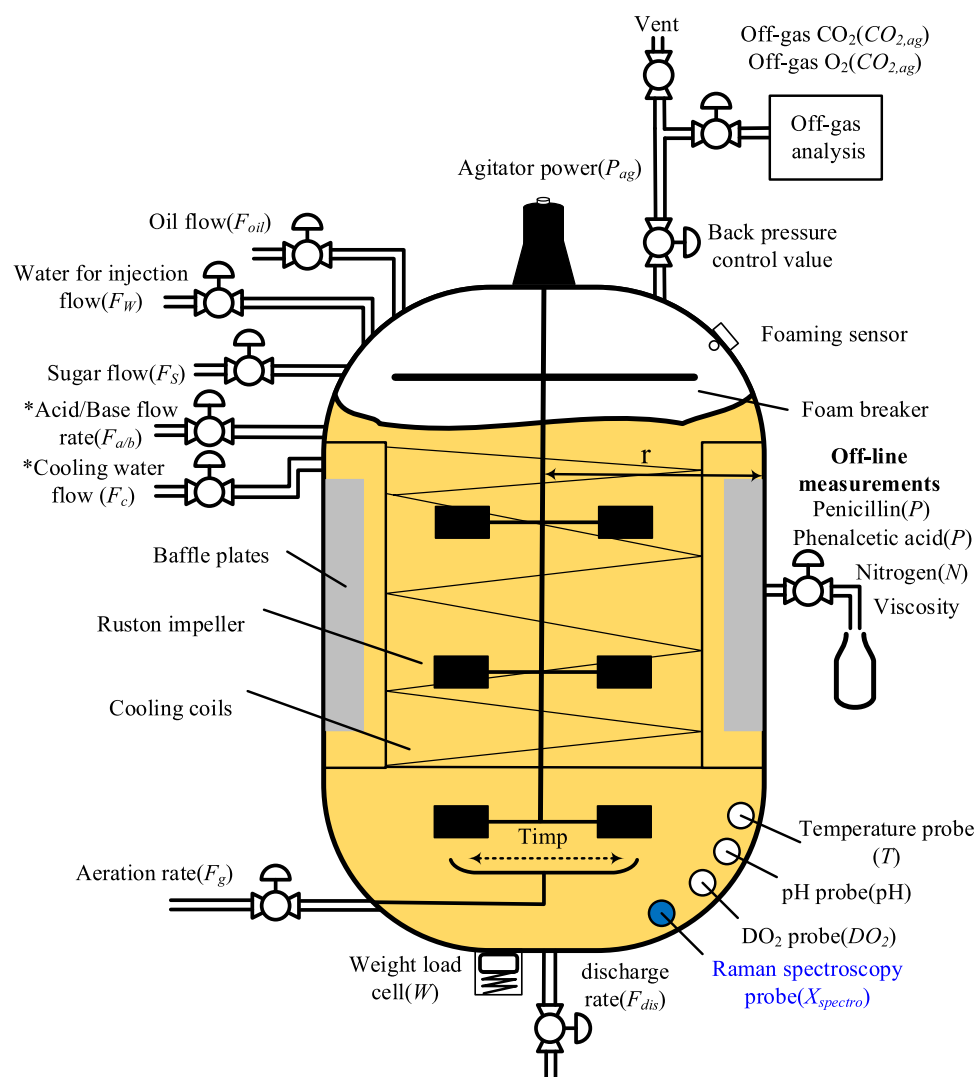


Fig. 5 The schematic of the Industrial-scale fed-batch fermentation process



key indicator of product quality and it is difficult to measure online. 30 batches of normal data were utilized for training and the rest of the batches were used for testing.

Table 1 Description of batch information in industrial-scale penicillin fermentation process

Batch No	Batch type	Control strategy
1–10	Normal batch for training	Recipe driven approach
11–30	Normal batch for test	
31–40	Normal batch for training	Operators
41–60	Normal batch for test	
61–70	Normal batch for training	Advanced Process Control (APC) & Raman spectroscopy
71–90	Normal batch for test	
91–100	Faulty batch	Faulty batch

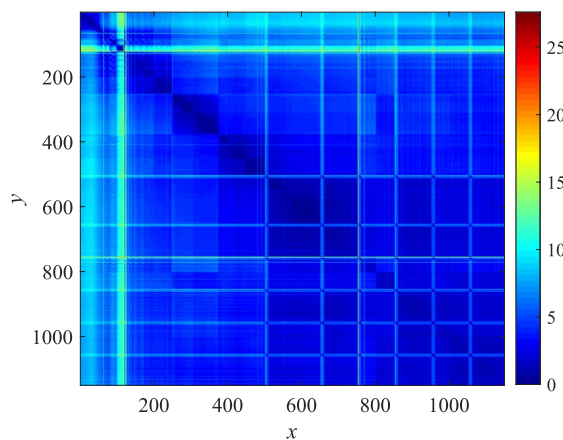
4.1.2 Results of stage identification

As mentioned in the Sect. 3.1 above, when stage identification through AP clustering, it is considered that the similarity matrix has a greater impact on the clustering results. The more similarities between samples are retained based on the finite Markov chain similarity matrix, while considering both temporal and neighborhood information. The heat map of similarity matrix with AP and IAP is shown in Fig. 6, in which different colors correspond to different values. It can be seen from Fig. 6 that the large values in the IAP heatmap are closer to the diagonal, which means that IAP uses a smaller similarity for larger time interval samples compared to AP. It can effectively prevent sample points with large time intervals from having too much influence on the current sample.

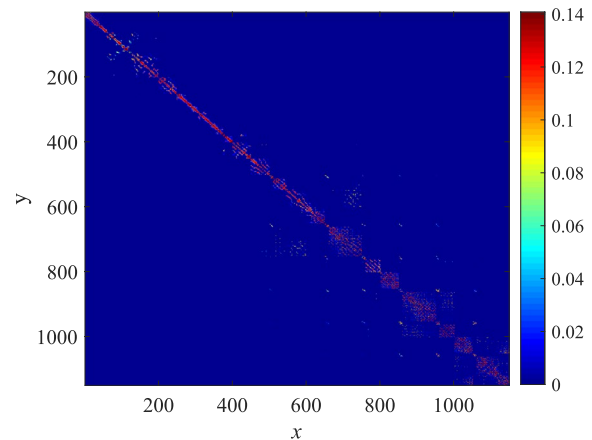
The IAP algorithm was utilized to segment the industrial-scale penicillin fermentation process into three sampling stages, i.e., 0~25 h, 25~120h, and 120~230 h, respectively.

Table 2 Description of process variable in industrial-scale penicillin fermentation process

Variable No	Variable description	Variable No	Variable description
1	Aeration rate (L/h)	12	Dissolved oxygen conc. (mg/L)
2	Agitator RPM (RPM)	13	Vessel Volume (L)
3	Sugar feed rate (L/h)	14	Vessel Weight (Kg)
4	Acid flow rate (L/h)	15	pH (pH)
5	Base flow rate (L/h)	16	Temperature (K)
6	Heating/cooling water flow rate (L/h)	17	Generated heat (kJ)
7	Heating water flow rate (L/h)	18	CO ₂ percent in off-gas (%)
8	Water for injection/dilution (L/h)	19	PAA flow (L/h))
9	Air head pressure (bar)	20	Oil flow (L/hr)
10	Dumped broth flow (L/h)	21	Oxygen in percent in off-gas (%)
11	Substrate concentration (g/L)	22	Carbon evolution rate (g/L)



(a) Similarity matrix of AP



(b) Similarity matrix of IAP

Fig. 6 Heat map of similarity matrices with IAP and AP

The results of the stage segmentation are consistent with the three stages of penicillin fermentation process: bacterial growth, penicillin synthesis and bacterial autolysis.

4.1.3 Results of variable selection

The importance of each process variable for the quality of the final product varies for the different stages. To build effective models at each stage, different process variables are selected for modeling at each stage reducing the complexity of modeling due to redundant variables. As mentioned in Sect. 3.2 above, each stage of the process variables that have a high impact on product quality are selected by mRMR, and the results of the variable selection are shown in Fig. 7. Then the yellow grid represents the selected variable, and the blue grid represents the unselected variable.

For some variables are uncorrelated with the characteristics of the final product quality throughout the production stages of the process, several process variables affect the product quality at specific stages.

4.1.4 The selection of model parameters

To demonstrate the superiority of the methods in this paper through comparison with GRU, BiGRU, BiGRU-AM, Spatial-Temporal Graph Conv-GRU Network (STGCNT) [45] and ConvBiGRU-AM, the choice of parameters for these methods is described as follows. The GRU parameters are chosen according to the Deep Learning toolkit implementation, which has been shown to be able to converge stably, ensuring that the stability of the training can provide better performance. To verify that performance improvement of the model is contributed by the design of the model structure rather than the number of trainable parameters, all comparison methods were designed to have similar number of parameters. Specially, for MI-ConvBiGRU-AM is a model of multiple stages of integration, each stage is set with different parameters, and the number of parameters is not more than that of the other comparison methods during the fairness period. Through the grid search method making the mean square error of the validation results the smallest, we

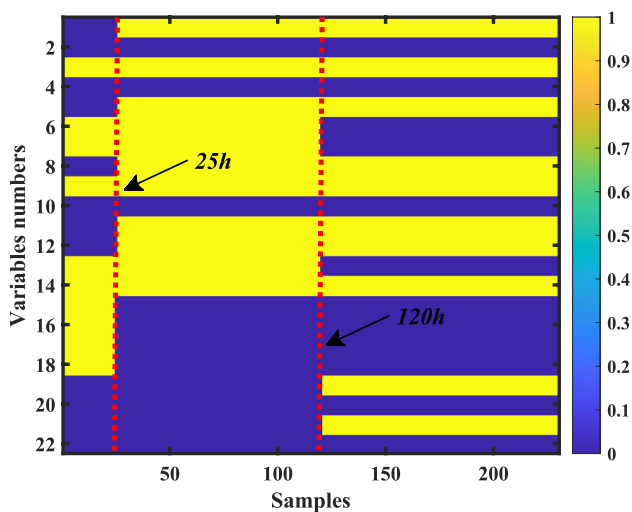


Fig. 7 Results of variable selection (yellow grid: selected variable; blue grid: unselected variable)

choose the suitable stopping epoch and hidden neurons for each of the above stages. The detailed parameter choices for the model are shown in Table 3.

4.1.5 Quality prediction and monitoring results

We use the parameters mentioned in the previous section to construct the MI-ConvBiGRU-AM model, as well as the comparison model. Under normal operating conditions, 30 batches of data with different control strategies are used for model training and validation. Batches of 70 data containing normal working conditions and fault states were used for model testing. For the test batches, the evaluation indexes of the penicillin quality prediction results of 60 batches under normal working conditions were mean values as shown in Table 4.

It can be noted from Table 4 that the model in this paper has better predictive performance. It can monitor the changes

of quality variables in real time. Compared with other models, MI-ConvBiGRU-AM has a smaller variance index and larger R^2 , which indicates that MI-ConvBiGRU-AM model has stronger robustness and fitting ability. Above results demonstrate the prospective performance of the proposed modeling framework in batch process quality prediction.

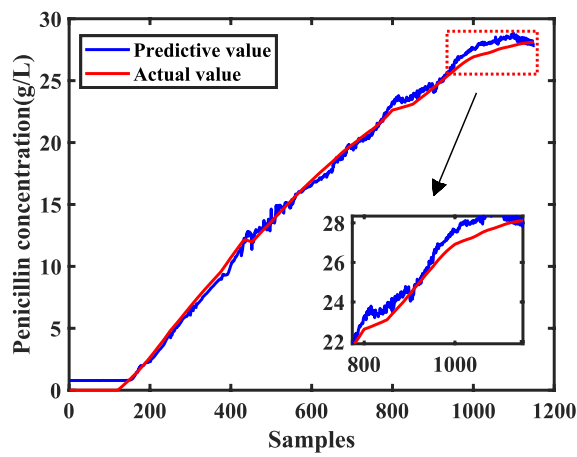
For one of the typical test batches the results of the prediction on penicillin concentration are shown in Fig. 8. Here, blue indicates the predicted value and red is the actual value. It can be seen from Fig. 8(a) that the traditional GRU model can predict the overall direction of the quality trend, but for the local area the gap between the actual value and the predicted value is worse, and it cannot predict the product quality well. Compared with Fig. 8(a) GRU, Fig. 8(b) BiGRU has improved local prediction ability, but there are still many locally inaccurate prediction points. It is shown in Fig. 8(c) BiGRU-AM introduces the output attention mechanism based on Fig. 8(b) BiGRU, and the prediction effect is further improved. To make the BiGRU-AM input features contain more information about the prediction results, Fig. 8(e) shows that ConvBiGRU-AM extracts the original data features by convolution operation, which makes the prediction results more accurate. Compared with Fig. 8(d) STGCNT, ConvBiGRU-AM which introduces the attention mechanism predicts better results. We proposed MI-ConvBiGRU-AM as an integrated structure that has different data and parameters at each stage for training ConvBiGRU-AM such that the

Table 4 Comparison of algorithms in evaluation index

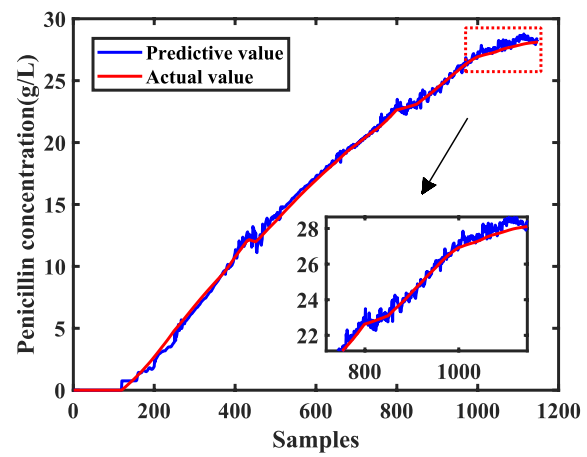
N0	Algorithms	MAE	RMSE	MAPE	R^2
1	GRU	0.0333	0.0692	0.3761	0.9896
2	BiGRU	0.0242	0.0490	0.0805	0.9949
3	BiGRU-AM	0.0144	0.0598	0.1262	0.9923
4	STGCNT	0.0067	0.0580	0.0322	0.9927
5	ConvBiGRU-AM	0.0068	0.0417	0.0266	0.9963
6	MI-ConvBiGRU-AM	0.0043	0.0396	0.0121	0.9971

Table 3 The selection of model parameters

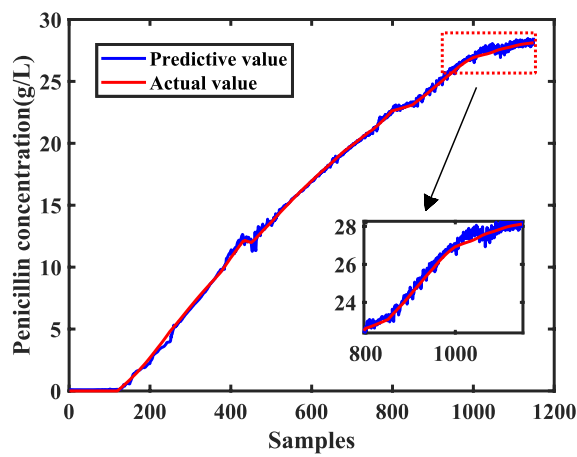
Model	Optimizer	Learning rate	Epochs (stages)	Batch Size	Time Steps	Fully Connected Layer	Hidden Neurons (stages)
GRU	Adam	0.0001	50	16	10	1	90
BiGRU	Adam	0.0001	50	16	10	1	90
BiGRU-AM	Adam	0.0001	50	16	10	1	90
STGCNT	Adam	0.0001	50	16	10	1	90
ConvBiGRU-AM	Adam	0.0001	50	16	10	1	90
MI-ConvBiGRU-AM	Adam	0.0001	20 (1)	16	10	1	40 (1)
	Adam	0.0001	30 (2)	16	10	1	65 (2)
	Adam	0.0001	45 (3)	16	10	1	90 (3)



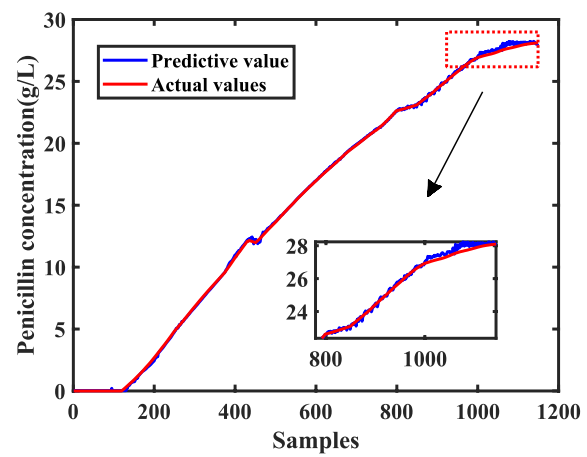
(a) GRU



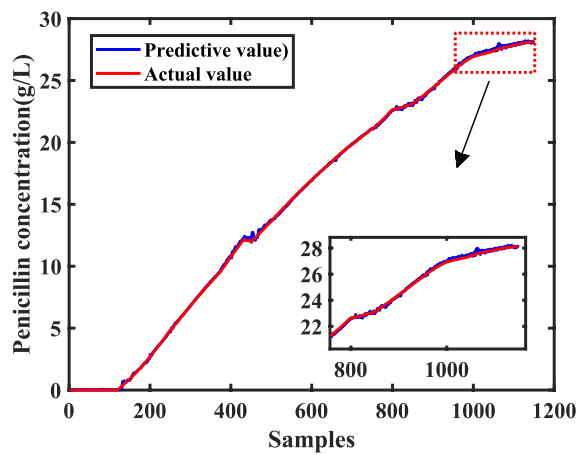
(b) BiGRU



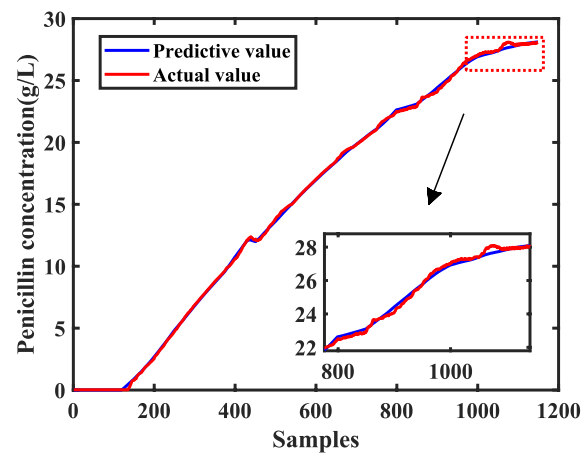
(c) BiGRU-AM



(d) STGCNT



(e) ConvBiGRU-AM



(f) MI-ConvBiGRU-AM

Fig. 8 Prediction results of penicillin concentration

integrated model is more adapted to the quality prediction of the multistage penicillin fermentation process.

As mentioned in Sect. 3.4, prediction residuals were used as statistics for quality monitoring, and thresholds were defined based on the residuals calculated from data under normal operating conditions. The performance of quality monitoring was verified with 20 test batches which contained normal operation and faults, and the mean fault detection rate and mean fault false alarm rate for each model are shown in Table 5. It can be observed from Table 5 that the proposed MI-ConvBiGRU-AM has higher fault detection rate and lower fault false alarm rate.

For typical batches, the quality monitoring results under normal and fault operating conditions are shown in Fig. 9. Where, quality related and unrelated variable faults are selected to be analyzed in the fault batch, and for the normal working condition which refers to under the control strategy of recipe driven approach. Where, Dissolved oxygen concentration, pH, temperature, substrate concentration, agitator speed, and hot/cooling water flow rate faults can directly affect the penicillin quality. Reactor volume, weight, and generated heat fault have a small effect on penicillin mass, which is a quality-unrelated fault variable. Figure 9 shows quality-related faults are substrate concentration faults and quality-unrelated faults are reactor volume faults.

It can be noticed in Fig. 9(a) that it is able to reach the expected value of penicillin concentration at the end of a batch by using the control strategy of recipe driven approach. From Fig. 9(b) we can see that the prediction residuals of the model are kept consistently below the threshold value, which indicates that there are no fault false alarms. In Fig. 9(c) because of the fault occurrence the prediction model is not able to predict the change of penicillin concentration well, but the fault is a quality unrelated fault therefore it does not have much impact on the product quality and it is able to generate the whole batch of penicillin that fulfills the requirements. According to the quality monitoring perspective, no downtime is required for this fault alarm. In Fig. 9(d), it shows the result of fault detection, the fault is detected at the 437th samples, and the actual time of fault occurs at the 435th sample with some detection delay, that is, the fault missed alarm. From Fig. 9(e), the fault has affected the quality of the product, and even if the fault is eliminated, penicillin concentration increases are no longer desirable for

the growth curve. Figure 9(f) reflects the fault detection after the occurrence of the fault, which can be detected at the first time. And since the elimination of this fault does not make it possible to produce the product normally, it is necessary to downtime and reprocess it.

The radar chart of multiple evaluative indicators for the different models is shown in Fig. 10. To enable the different indicators to be visualized in the radar chart, the values of the indicators were quantified. For the indicators FAR, MAE, RMSE, and MAPE, smaller values indicate better model; for FDR and R^2 larger values is better. It can be clearly seen from Fig. 10 that the proposed model in this paper has better results compared to other models.

4.2 Hot strip mill process

The hot strip rolling process is a high investment, high quality and high productivity rolling process. A typical hot strip rolling line includes a heating furnace, roughing mill, transfer table & shear, finishing mill, laminar cooling equipment and coiler. The Schematic of the Hot strip mill process is shown in Fig. 11. The process is as follows: the slab is heated in a heating furnace and then rolled several times through the roughing mill to form an intermediate slab. The intermediate slab is then quickly transported to the finishing mill using a flying shear and intermediate delay roller tables for more precise control to obtain a strip that meets the requirements. Finally, the strip is treated in a laminar flow cooling plant to improve properties and coiled for storage. Strip thickness is one of the key factors in the production of hot-rolled strips. According to customer requirements, the hot rolled strip process can produce strip thicknesses from 1.5 mm to 12.7 mm to meet different requirements.

4.2.1 Description of hot strip mill and datasets

The case study is experimentally validated with live data of strip production from a steel company, which produces thickness of 3.9 mm strips. The production process involves 20 process variables and 1 quality variable which is shown in Table 6. This includes roll gaps, rolling forces and bending roll forces for the 7 stands of the finishing mill (no bending rolls for stand 1), and the quality variable is the thickness at the exit of the finishing mill.

Table 5 Mean monitoring index of different methods for 10 test batches

Algorithms	GRU	BiGRU	BiGRU-AM	STGCNT	ConvBiGRU-AM	MI-ConvBiGRU-AM
FDR/%	93.91	95.65	97.73	96.95	98.34	99.73
FAR/%	1.18	3.25	2.91	2.29	2.13	0.54

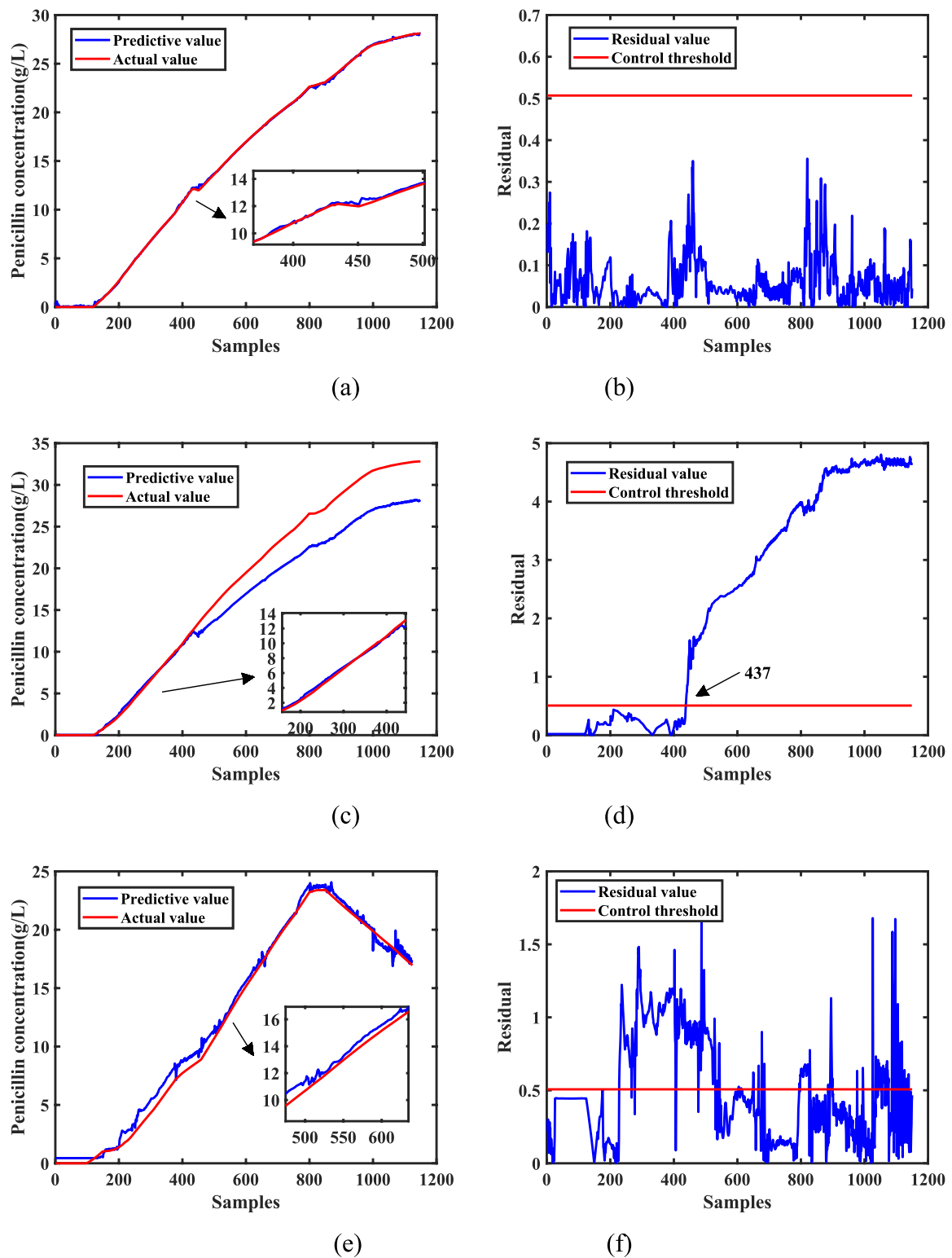


Fig. 9 Quality monitoring results for different test batches. (a) and (b) are prediction and monitoring results under normal operating conditions; (c) and (d) are prediction and monitoring results for quality

unrelated faults; (e) and (f) are prediction and monitoring results for quality related faults

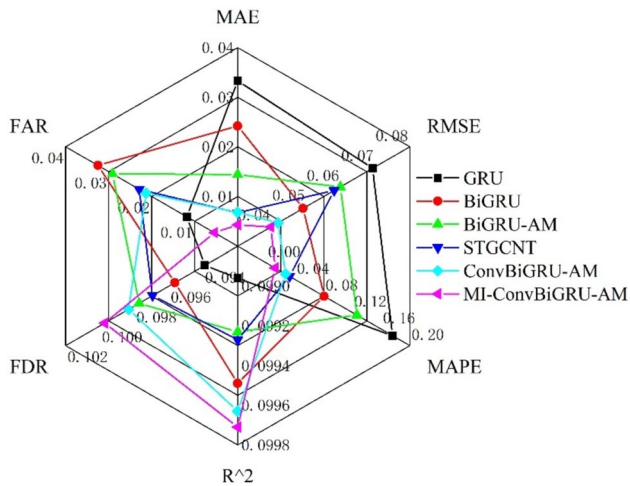


Fig. 10 Radar plot of prediction and monitoring indicators for different algorithms

In this case study, it collects process data for each batch with sampling time of 30 s, sampling period 10 ms, and total 3000 samples.

4.2.2 Results of stage identification

As mentioned in Sect. 3.1, according to the finite Markov chain similarity matrix, more similarity between samples is also kept by considering both temporal and neighborhood information. Figure 12 shows the heatmap of AP and IAP similarity matrices with different colors representing different values.

It can be noticed from Fig. 12 that the larger values in the IAP are closer to the diagonal line, which means that IAP uses smaller similarity for samples from larger time intervals compared to AP.

The IAP algorithm was utilized to segment the hot strip mill process into three sampling stages, i.e., 0~572th, 572~2117th, and 2117~3000th, respectively. The results of the stage segmentation are consistent with the three stages

Table 6 Process and quality variables in finishing mill

Variable No	Type	Description	Unit
1–7	Process variable	Average roll gap of 7 stands	mm
8–14	Process variable	Roll force of 7 stands	MN
15–20	Process variable	Bending roll force of 6 stands	MN
21	Quality variable	Exit thickness at the last finishing mill stand	mm

of hot strip mill process: acceleration period, steady period and deceleration period.

4.2.3 Results of variable selection

For hot strip mill process, as mentioned in Sect. 3.2 above, each stage of the process variables that have a high impact on product quality are selected by mRMR, and the results of the variable selection are shown in Fig. 13.

4.2.4 The selection of model parameters

To verify that performance improvement of the model is contributed by the design of the model structure rather than the number of trainable parameters, all comparison methods were designed to have similar number of parameters. Specially, for MI-ConvBiGRU-AM is a model of multiple stages of integration, each stage is set with different parameters, and the number of parameters is not more than that of the other comparison methods in the interests of fairness. By using the grid search method, the mean square error of the validation results is minimized, we choose the suitable stopping epoch and hidden neurons for each of the above stages. The detailed parameter choices for the model are shown in Table 7.

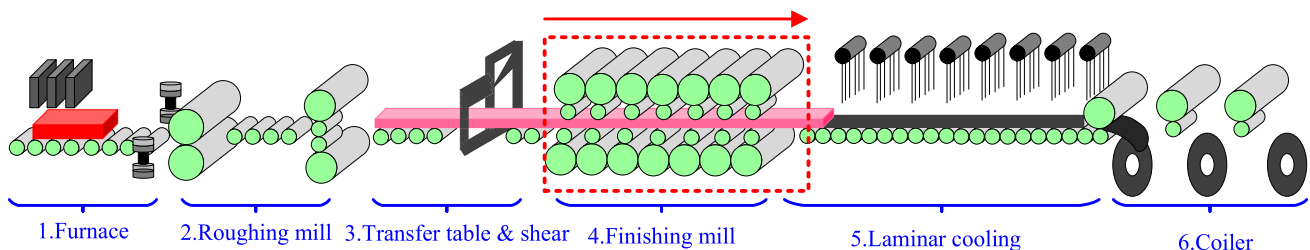


Fig. 11 Schematic of the Hot strip mill process

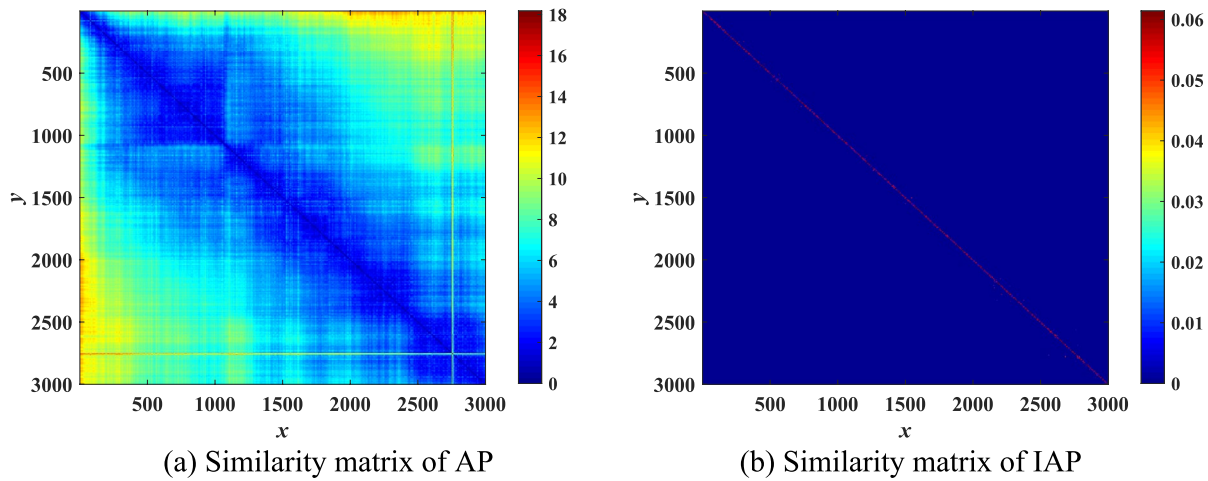


Fig. 12 Heat map of similarity matrices with IAP and AP

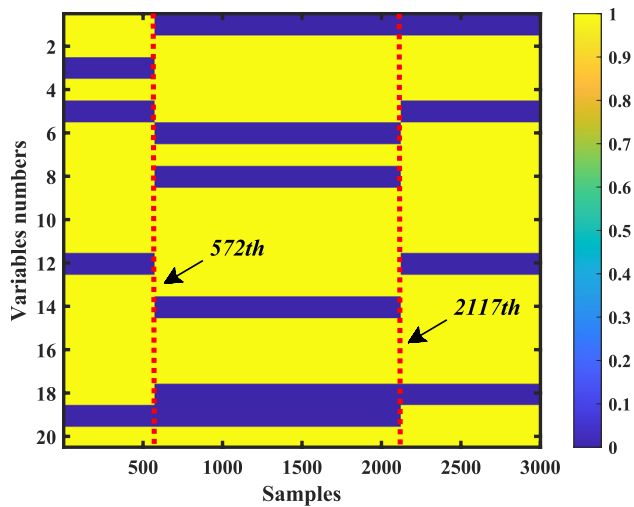


Fig. 13 Results of variable selection (yellow grid: selected variable; blue grid: unselected variable)

4.2.5 Quality prediction and monitoring results

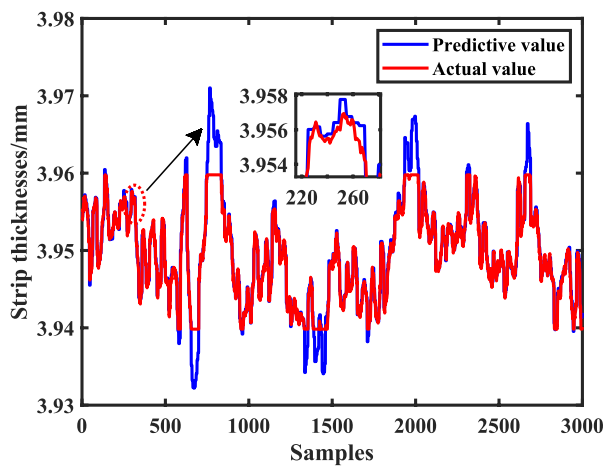
The MI-ConvBiGRU-AM model was constructed with the designed parameters and comparison models. Normal data of 10 batches were used for model training and validation. For the test batches, the evaluation indexes of the hot strip

Table 8 Comparison of algorithms in evaluation index

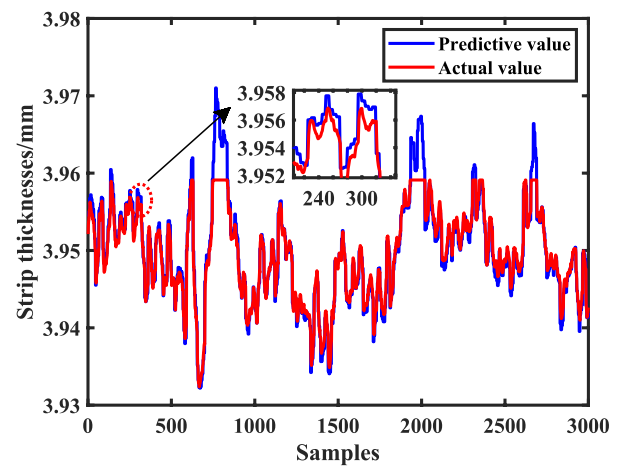
N0	Algorithms	MAE	RMSE	MAPE	R ²
1	GRU	0.0580	0.0996	0.7534	0.9202
2	BiGRU	0.0526	0.0971	0.6635	0.9220
3	BiGRU-AM	0.0404	0.0656	0.6250	0.9653
4	STGCNT	0.0357	0.0592	0.6052	0.9718
5	ConvBiGRU-AM	0.0129	0.0351	0.3432	0.9901
6	M-ConvBiGRU-AM	0.0053	0.0111	0.1539	0.9990

Table 7 The selection of model parameters

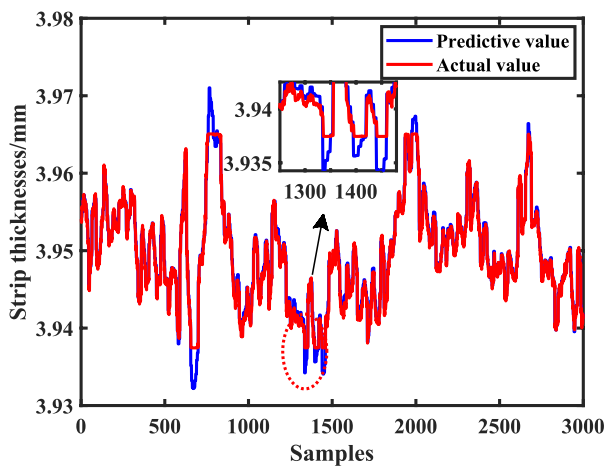
Model	Optimizer	Learning rate	Epochs (stages)	Batch Size	Time Steps	Fully Connected Layer	Hidden Neurons (stages)
GRU	Adam	0.0001	30	8	10	1	48
BiGRU	Adam	0.0001	30	8	10	1	48
BiGRU-AM	Adam	0.0001	30	8	10	1	48
STGCNT	Adam	0.0001	30	8	10	1	48
ConvBiGRU-AM	Adam	0.0001	30	8	10	1	48
MI-ConvBiGRU-AM	Adam	0.0001	25 (1)	8	10	1	24 (1)
	Adam	0.0001	20 (2)	8	10	1	30 (2)
	Adam	0.0001	20 (3)	8	10	1	26 (3)



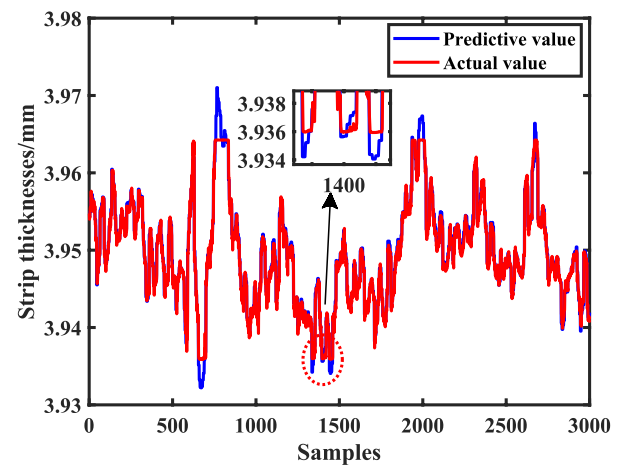
(a) GRU



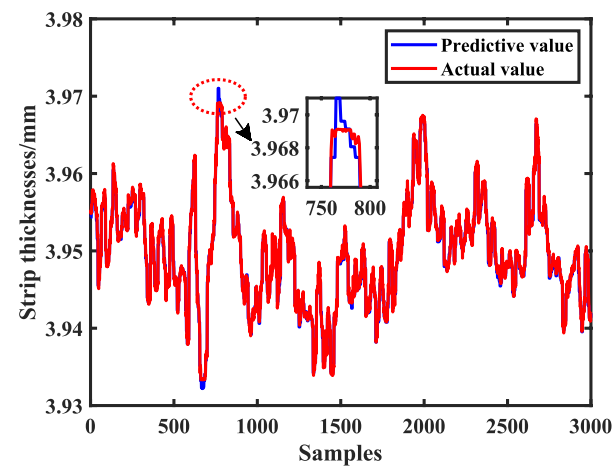
(b) BiGRU



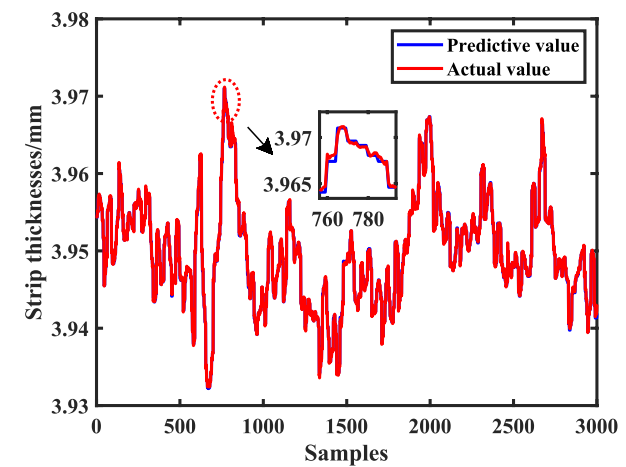
(c) BiGRU-AM



(d) STGCNT



(e) ConvBiGRU-AM



(f) MI-ConvBiGRU-AM

Fig. 14 Prediction results of strip thicknesses

Table 9 Mean monitoring index of different methods for 5 test batches

Algorithms	GRU	BiGRU	BiGRU-AM	STGCNT	ConvBiGRU-AM	MI-ConvBiGRU-AM
FDR/%	94.99	95.23	96.15	97.23	98.01	99.95
FAR/%	3.25	3.23	2.12	2.36	1.22	0.25

mill quality prediction results of 5 batches under normal working conditions were mean values as shown in Table 8.

It can be noticed from Table 8 that the model in this paper has better predictive performance, which can monitor the changes of quality variables in real time. The relevant evaluation indexes indicate that the MI-ConvBiGRU-AM model has stronger robustness and fitting ability.

For one of the typical tests batches, the results of the prediction on strip thickness are shown in Fig. 14. Here, blue indicates the predicted value and red is the actual value.

It can be shown from Fig. 14(a) that the traditional GRU model cannot predict strip thickness trends well in the local area. Compared with Fig. 8(a) GRU, Fig. 8(b) BiGRU has improved local prediction ability, but there are still many locally inaccurate prediction points. Figure 14(c) and Fig. 14(d) introduce the attention mechanism and the convolution part based on BiGRU, respectively, which makes the prediction results more accurate. Compared with Fig. 14(d) STGCNT, ConvBiGRU-AM which introduces the attention mechanism predicts better results. We proposed MI-ConvBiGRU-AM as an integrated structure that has different data and parameters at each stage for training ConvBiGRU-AM such that the integrated model is more adapted to the quality prediction of the multistage hot strip mill process.

The performance of quality monitoring was verified with test batches and the mean fault detection rate and mean fault false alarm rate for each model are shown in Table 9. It can be observed from Table 9 that the proposed MI-ConvBiGRU-AM has higher fault detection rate and lower fault false alarm rate.

For normal operating conditions, the quality related and unrelated fault quality monitoring results are shown in Fig. 15. For hot strip mill process, the fault of the actuator of the press-down system in stand 4 is a quality-related fault affecting the strip thickness, and the fault of the bend roll force control system in stand 5 is a quality-unrelated fault not affecting the strip thickness.

It can be seen from Fig. 15(a) that the method proposed in this paper can accurately predict the strip thickness. From Fig. 15(b) we can see that the prediction residuals of the model are kept consistently below the threshold value, which indicates that there are no fault false alarms.

From Fig. 15(c), the fault affects the quality of the product. Figure 15(d) reflects the fault detection after the fault has occurred, which can be detected at the first time. Since the elimination of this fault would not allow the product to be produced normally, it was necessary to stop the machine and reprocess it. In Fig. 15(e), because of the occurrence of the fault, the prediction model is unable to predict the change in strip thickness well, but the fault is a quality-unrelated fault, so it does not have much impact on the product quality, and is able to generate a whole batch of strip steel that meets the requirements. Figure 15(f), the alarm occurs due to fault. According to the quality monitoring point of view, this fault alarm does not require downtime.

The radar chart of multiple evaluative indicators for the different models is shown in Fig. 16. To enable the different indicators to be visualized in the radar chart, the values of the indicators were quantified. It can be clearly seen from Fig. 16 that the proposed model in this paper has better results compared to other models.

5 Conclusions

In this study, we propose an MI-ConvBiGRU-AM model for batch process quality monitoring and prediction. This model accomplishes process stage identification through IAP and employs mRMR at each stage to select process variables related to final product quality for modeling and to improve the interpretability of the subsequent model. The proposed model effectively extracts predictive features at the input layer, while combining historical and future information to extract dynamic and nonlinear features, and introduces attention mechanisms to ensure that important information is not lost and to reduce the complexity of the model. Two typical batch processes, industrial-scale fed-batch fermentation process and hot strip mill process, demonstrate that the model is effective for monitoring high quality production of batch products by accurately predicting product quality while reducing false alarms for processes that do not affect product quality. Since MI-ConvBiGRU-AM model needs huge computational resources during the training and monitoring process, in future, we will research a lightweight network based on MI-ConvBiGRU-AM.

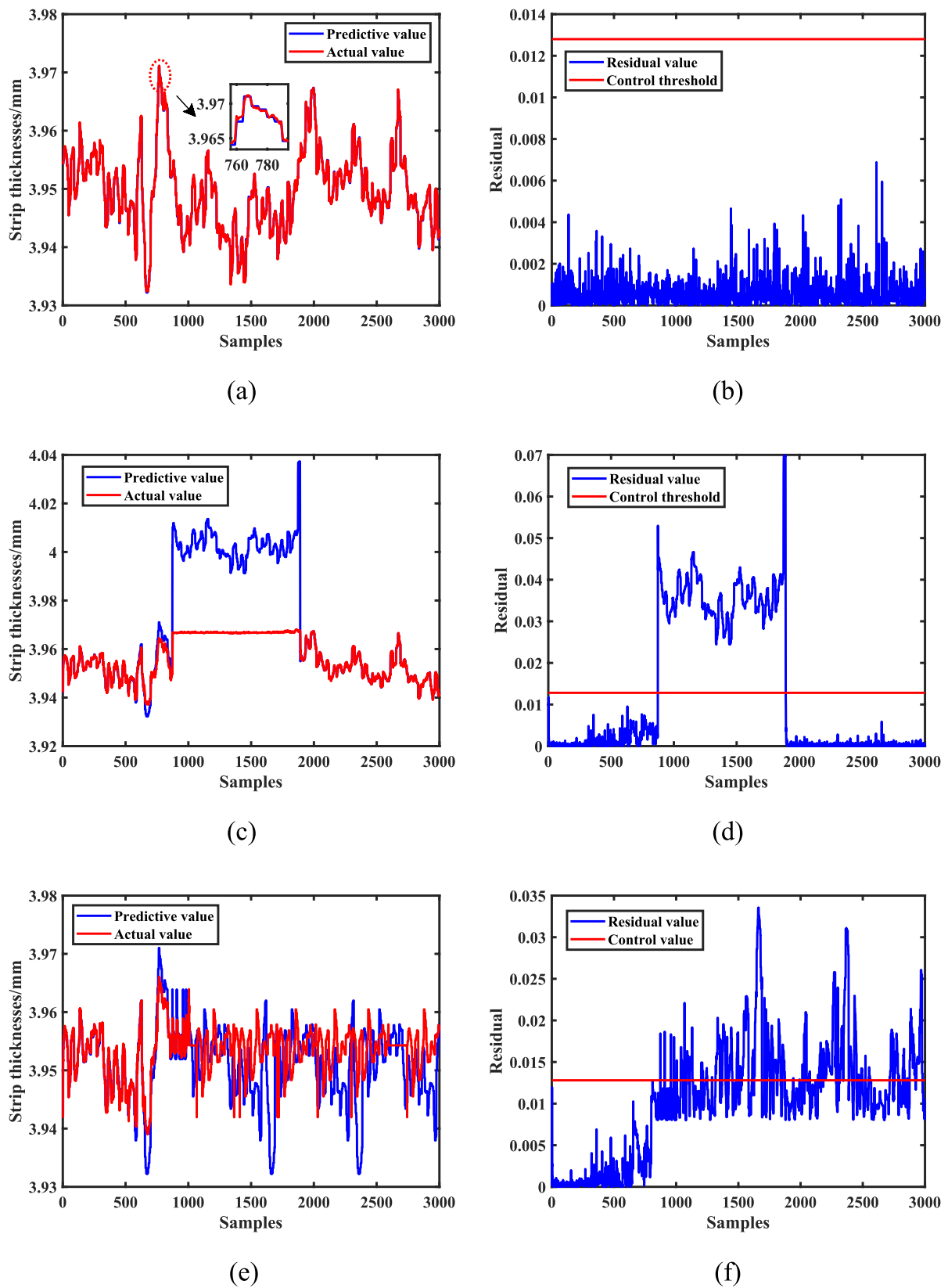


Fig. 15 Quality monitoring results for different test batches. (a) and (b) are prediction and monitoring results under normal operating conditions; (c) and (d) are prediction and monitoring results for quality

related faults; (e) and (f) are prediction and monitoring results for quality unrelated faults

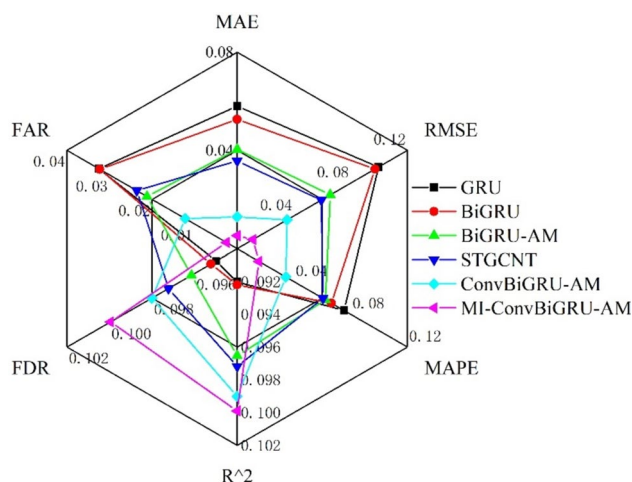


Fig. 16 Radar plot of prediction and monitoring indicators for different algorithms

Acknowledgements This research work has been awarded by the National Natural Science Foundation of China (62263021, 62163023), Industrial Support Project of Education Department of Gansu Province (2023CYZC-24), the Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (2022KX07).

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

Declarations

Conflict of interest There is no conflict of interest.

References

- Fujita H, Fournier-Viger P, Sasaki J, Ali M (2021) Advances in theory and applications of artificial intelligence. *AI Mag* 42(1):86–87
- Chandrasekar A, Radhika T, Zhu Q (2022) Further results on input-to-state stability of stochastic Cohen–Grossberg BAM neural networks with probabilistic time-varying delays. *Neural Process Lett* 1–23
- Radhika T, Chandrasekar A, Vijayakumar V, Zhu Q (2023) Analysis of Markovian jump stochastic Cohen–Grossberg BAM neural networks with time delays for exponential input-to-state stability. *Neural Process Lett* 55(8):11055–11072
- Tamil Thendral M, Ganesh Babu TR, Chandrasekar A, Cao Y (2022) Synchronization of Markovian jump neural networks for sampled data control systems with additive delay components: analysis of image encryption technique, Mathematical methods in the applied sciences
- 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
- Peng C, Chunhao D (2022) Monitoring multi-domain batch process state based on fuzzy broad learning system. *Expert Syst Appl* 187:115851
- Sansana J, Rendall R, Joswiak MN, Castillo I, Miller G, Chiang LH, Reis MS (2023) a functional data-driven approach to monitor and analyze equipment degradation in multiproduct batch processes. *Process Safety Environ Protect*
- Zhang Y, Cao J, Zhao X, Hui Y (2023) Nonlinear multiphase batch process monitoring and quality prediction using multi-way concurrent locally weighted projection regression. *Chemom Intell Lab Syst* 240:104922
- Yu Y (2012) Intelligent quality prediction using weighted least square support vector regression. *Phys Procedia* 24:1392–1399
- Yuan X, Ge Z, Song Z (2014) Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Ind Eng Chem Res* 53(35):13736–13749
- Yu J (2012) Multiway Gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes. *Ind Eng Chem Res* 51(40):13227–13237
- Rong M, Shi H, Tan S (2019) Large-scale supervised process monitoring based on distributed modified principal component regression. *Ind Eng Chem Res* 58(39):18223–18240
- Gins G, Van Impe JF, Reis MS (2018) Finding the optimal time resolution for batch-end quality prediction: MRQP–A framework for multi-resolution quality prediction. *Chemom Intell Lab Syst* 172:150–158
- Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554
- Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9:611–629
- Jiang K, Han Q, Du X, Ni P (2021) A decentralized unsupervised structural condition diagnosis approach using deep auto-encoders. *Computer-Aided Civil Infrastruct Eng* 36(6):711–732
- Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput* 31(7):1235–1270
- Zhao R, Wang D, Yan R, Mao K, Shen F, Wang J (2017) Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans Industr Electron* 65(2):1539–1548
- Yao L, Ge Z (2023) Causal variable selection for industrial process quality prediction via attention-based GRU network. *Eng Appl Artif Intell* 118:105658
- Ma L, Wang M, Peng K (2022) A novel bidirectional gated recurrent unit-based soft sensor modeling framework for quality prediction in manufacturing processes. *IEEE Sens J* 22(19):18610–18619
- Li J, Yang C, Li Y, Xie S (2021) A context-aware enhanced GRU network with feature-temporal attention for prediction of silicon content in hot metal. *IEEE Trans Industr Inf* 18(10):6631–6641
- Sun K, Liu J, Kang J-L, Jang S-S, Wong DS-H, Chen D-S (2014) Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote. *J Process Control* 24(7):1068–1075
- Fujiwara K, Kano M (2015) Efficient input variable selection for soft-sensor design based on nearest correlation spectral clustering and group Lasso. *ISA Trans* 58:367–379
- Yao L, Ge Z (2018) Variable selection for nonlinear soft sensor development with enhanced binary differential evolution algorithm. *Control Eng Practice* 72:68–82
- Zhao C (2014) Concurrent phase partition and between-mode statistical analysis for multimode and multiphase batch process monitoring. *AIChE J* 60(2):559–573
- Luo L, Bao S, Mao J, Tang D (2016) Phase partition and phase-based process monitoring methods for multiphase




- batch processes with uneven durations. *Ind Eng Chem Res* 55(7):2035–2048
27. Peng K, Li Q, Zhang K, Dong J (2016) Quality-related process monitoring for dynamic non-Gaussian batch process with multi-phase using a new data-driven method. *Neurocomputing* 214:317–328
 28. Liu J, Liu T, Chen J (2018) Sequential local-based Gaussian mixture model for monitoring multiphase batch processes. *Chem Eng Sci* 181:101–113
 29. Peng C, Lu R, Kang O, Kai W (2020) Batch process fault detection for multi-stage broad learning system. *Neural Netw* 129:298–312
 30. Zhao X, Liu K, Hui Y (2023) Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm. *Trans Inst Meas Control* 45(1):89–103
 31. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. *Science* 315(5814):972–976
 32. Lerm S, Saeedi A, Rahm E (2021) Extended affinity propagation clustering for multi-source entity resolution
 33. Wei Z, He D, Jin Z, Liu B, Shan S, Chen Y, Miao J (2023) Density-based affinity propagation tensor clustering for intelligent fault diagnosis of train bogie bearing. *IEEE Trans Intell Transp Syst* 24(6):6053–6064
 34. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
 35. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches, arXiv preprint [arXiv:1409.1259](https://arxiv.org/abs/1409.1259)
 36. Zhang X, Tang L, Chen J (2021) Fault diagnosis for electro-mechanical actuators based on STL-HSTA-GRU and SM. *IEEE Trans Instrum Meas* 70:1–16
 37. Xia M, Shao H, Ma X, De Silva CW (2021) A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. *IEEE Trans Industr Inf* 17(10):7050–7059
 38. Zhao H (2018) Dynamic graph embedding for fault detection. *Comput Chem Eng* 117:359–371
 39. Gu X, Guo J, Xiao L, Li C (2022) Conditional mutual information-based feature selection algorithm for maximal relevance minimal redundancy. *Appl Intell* 52(2):1436–1447
 40. Niu Z, Zhong G, Yu H (2021) A review on the attention mechanism of deep learning. *Neurocomputing* 452:48–62
 41. Goldrick S, Ștefan A, Lovett D, Montague G, Lennox B (2015) The development of an industrial-scale fed-batch fermentation simulation. *J Biotechnol* 193:70–82
 42. Ding SX, Yin S, Peng K, Hao H, Shen B (2012) A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *IEEE Trans Industr Inf* 9(4):2239–2247
 43. Mears L, Stocks SM, Sin G, Gernaey KV (2017) A review of control strategies for manipulating the feed rate in fed-batch fermentation processes. *J Biotechnol* 245:34–46
 44. Nadal-Rey G, McClure DD, Kavanagh JM, Cassells B, Cornelissen S, Fletcher DF, Gernaey KV (2021) Development of dynamic compartment models for industrial aerobic fed-batch fermentation processes. *Chem Eng J* 420:130402
 45. Mourchid Y, Slama R (2023) D-STGCNT: a dense spatio-temporal graph Conv-GRU Network based on transformer for assessment of patient physical rehabilitation. *Comput Biol Med* 165:107420

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

RESEARCH ARTICLE

Siamese DeNPE network framework for fault detection of batch process

Kai Liu^{1,2}  | Xiaoqiang Zhao^{1,2,3} | Miao Mou^{1,2}  | Yongyong Hui^{1,2,3} 

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

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

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

Correspondence

Xiaoqiang Zhao, College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China.
 Email: xqzhao@lut.edu.cn

Funding information

Gansu Provincial Education Department Postgraduate Innovation Star Project, Grant/Award Numbers: 2023CXZX-467, 2023CXZX-474; National Natural Science Foundation of China, Grant/Award Number: 62263021; Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process, Grant/Award Number: 2022KX07; Science and Technology Project of Gansu Province, Grant/Award Numbers: 21JR7RA206, 21YF5GA072

Abstract

In batch processes, it is crucial to ensure safe production by fault detection. However, the long batch duration, limited runs, and strong nonlinearity of the data pose challenges. Incipient faults with small amplitudes further complicate the detection process. To achieve safe production, motivated by deep learning strategies, we propose a new fault detection method of batch process called Siamese deep neighbourhood preserving embedding network (SDeNPE). First, the DeNPE network is constructed by means of NPE and kernel functions, which utilizes the different types of kernel functions in the kernel mapping layer to extract diverse deep nonlinear features and overcome strong nonlinearity in the process data. Then, the Siamese network is used to obtain the different features between the data and improve the recognition of incipient faults. In addition, the deep extraction and Siamese network allow for batches of training data reduction without diminishing the performance of fault detection. Finally, we utilize monitoring statistics to complete the fault detection process. Two batch process cases involving the penicillin fermentation process and the semiconductor etching process demonstrate the superior fault detection performance of the proposed SDeNPE over the other comparison methods.

KEYWORDS

batch process, fault detection, kernel method, neighbourhood preserving embedding, Siamese network

1 | INTRODUCTION

The batch process has the advantages of small batches, multiple varieties, and flexible operating batches, and therefore plays an important role in industrial production. Along with improved automation and increasingly complex processes, the safety and reliability of production systems are becoming more and more important. Therefore, it is urgent to develop process monitoring technology to ensure that the operational potential of the process is maximized and that the production is safe and efficient. Fault detection of batch process is an important

research topic.^[1–5] Traditional model-based methods have difficulty achieving better performance of fault detection in complex batch processes, and data-driven methods are gaining more and more attention from scholars. As one of the important branches of data-driven approaches, multivariate statistical process monitoring (MSPM) methods have been successfully used for batch processes with satisfying results.^[6–8]

Typical MSPM methods include principal component analysis (PCA),^[9] independent component analysis (ICA),^[10] partial least squares (PLS),^[11] and neighbourhood preserving embedding (NPE).^[12] Compared to other

MSPM methods, NPE considers the local neighbourhood of data samples by mapping similar samples to neighbourhood locations in the low-dimensional space. The local structure of the original data is maintained in the low-dimensional space, which helps to preserve the important similarities and differences among the data samples. Meanwhile, the batch diversity of the batch process pays more attention to the local structural information, and the preservation of the manifold structure in the feature extraction process is very important for the fault detection model. NPE is considered due to the advantage of preserving the local manifold structure. By considering the serial correlation of the process data, Miao et al.^[13] proposed temporal neighbourhood preserving embedding (TNPE) and applied it to dynamic process fault detection. However, process data are closely related to time series and dynamic characteristic. For the dynamic characteristic of processes, Zhao and Wang^[14] proposed a tensor dynamic neighbourhood preserving embedding (TDNPE) algorithm. Song et al.^[15] constructed an enhanced neighbourhood preserving embedding (ENPE) to solve reconstruction error and distance problems for industrial process monitoring. In order to monitor the processes with multiple operating conditions, the multiple mode NPE method has been developed.^[16,17] Tan et al.^[18] proposed an adaptive neighbourhood preserving embedding (ANPE) algorithm that combined nearly linear dependency conditions with neighbourhood preserving embedding to achieve online fault detection. Zhu et al.^[19] proposed an advanced fault diagnosis method based on discriminative neighbourhood preserving embedding of Mahalanobis distance (DNPE-M), which addressed the problems of classification accuracy and data overlap in process monitoring.

All the above NPE-based methods are designed for linear process monitoring. However, most industrial processes are nonlinear, so researchers have developed nonlinear NPE algorithm, called kernel NPE (KNPE).^[20] Owing to its effectiveness, KNPE has become a widely used algorithm for nonlinear process monitoring. To improve the performance of nonlinear modelling, Miao et al.^[21] proposed an orthogonal KNPE method for fault detection of nonlinear processes. In addition, Miao et al.^[22] employed KNPE for soft sensor modelling, which built KNPE from local variables that was different from previous global modelling and which was applied to industrial cases; the method was simultaneously used to estimate some product qualities or critical variables that were difficult to measure online. Liu et al.^[23] proposed a new anomaly detection method which used KNPE and the double kernel parametrization to explore the similarity relationship of the data and in turn to detect process anomalies. Mou and Zhao^[24] proposed a

hybrid kernel function KNPE method for the timely detection and diagnosis of incipient faults.

Such KNPE-based methods all use KNPE as the core algorithm and have been applied to various nonlinear process monitoring cases. However, the KNPE model is not adequate to extract the inherent features of process data, and KNPE only obtains one layer of nonlinear features for statistical modelling when applying a kernel function to implement a nonlinear transformation. For complex nonlinear process data, it is often difficult to mine the inherent information through a single feature extraction step. In other words, the feature extraction with KNPE is shallow and does not make full use of the inherent data information. In recent years, deep learning has had great success in the field of fault detection.^[25–27] Zhao and Lai^[28] proposed a neighbourhood preserving neural network (NPNN) which adaptively trained a nonlinear neural network for extracting deep features, and considered the local geometric structure of the data. Liu et al.^[29] proposed a novel stacked neighbourhood preserving autoencoder (S-NPAE) to extract hierarchical neighbourhood preserving features, and this network extracted deep features by stacking. This demonstrates that multi-layer feature extraction facilitates the discovery of complex data structures. For small amplitude signals, feature extraction often involves much deeper potential variables to be obtained. With deep feature extraction from raw signals, the fault information contained in early micro signals can be fully extracted. Therefore, it is necessary to develop a multi-layer KNPE feature extraction model in process monitoring.

As mentioned above, latent feature extraction is an important aspect of all these algorithms. Latent variables have the unobserved features that really represent the property of the data. For some processes, the number of batches of process data collected is limited. In general, for fault detection in batch processes, the number of fault batches affects the modelling results and the fault detection results. Abundant modelling batches statistically cover enough batch-to-batch process variations to reveal the potential process information. However, it may be impractical to perform many operation cycles in each state and wait for sufficient available batches. Therefore, the models trained with limited batch data often have poor results. Siamese networks are widely used in process situations with limited data due to their effectiveness in dealing with small samples of data.^[30–32] Qin and Hu^[33] proposed an ANS-Net framework for measuring inherent differences by using a small number of signals, and established a fault diagnosis model. Yang et al.^[34] utilized a Siamese two-dimensional convolutional neural network to extract the feature vectors of the input fault signal couple. For limited training samples, the proposed method

was able to obtain good fault diagnosis accuracy. Therefore, Siamese network is applied to the modelling of batch process, which deals with limited batches of batch process data.

Motivated by the above analysis, we propose an improved NPE method with a deep architecture, referred to as Siamese deep neighbourhood preserving embedding network (SDeNPE), which adopts the Siamese depth feature extraction strategy to obtain the data features for nonlinear process monitoring. In contrast to the traditional shallow NPE and KNPE, SDeNPE constructs a deep feature extraction model that captures the deep nonlinear features of the data. By alternating different kernel functions in the kernel mapping layer, the extracted features are more diverse. Simultaneously, SDeNPE has a Siamese structure, which obtains difference characteristics by measuring the differentiation between data. Normally, the difference between normal samples is small, and when a fault occurs, the difference between the faults and normal samples can increase. The faults with smaller magnitudes can be detected by the structure of this difference metric. In this way, the twin structure allows more meaningful low-dimensional features to be preserved for fault detection. SDeNPE effectively solves the problem of limited batches of a batch process by obtaining difference features for fault detection through the Siamese method based on the nonlinear feature extraction. The effectiveness of SDeNPE is verified by two typical cases of batch process: the penicillin fermentation and the semiconductor etching processes.

In this paper, SDeNPE is used for batch process fault detection. The main contributions lie in the following aspects:

1. In this paper, a new SDeNPE network is proposed for batch process fault detection, motivated by a deep learning strategy. The network improves the fault detection performance of nonlinear batch processes by alternately employing different kernel functions in different kernel mapping layers, resulting in a diversity of obtained deep nonlinear features.
2. The proposed SDeNPE network has a Siamese architecture, which effectively captures the difference characteristics in process data, thereby ensuring better fault detection performance, especially for incipient micro magnitude faults.
3. The SDeNPE network can mine deep features and extract data difference features, which leads to obtaining excellent fault detection performance even with limited batches of training data, making it highly effective for batch processes with a limited data availability.

The structure of the paper is as follows. In Section 2, a brief overview of NPE, KNPE, and Siamese neighbourhood preserving embedding (SNPE) is given, while our proposed SDeNPE network for nonlinear process fault detection is detailed in Section 3. In Section 4, two cases are used to validate the proposed network, and our conclusions are drawn in Section 5.

2 | OVERVIEW OF NPE, KNPE, AND SNPE

The proposed network is based on NPE^[35] and Siamese neural networks.^[36] This section presents these preliminaries.

2.1 | NPE method

Consider a data matrix \mathbf{X} with n samples and m variables for each sample, for which the matrix is assumed to be standardized according to the z-score method. NPE represents the local linear structure of manifold by local linear reconstruction, which is in the form of mean square error. The objective function representing the reconstruction error in the high-dimensional space is shown in formula (1):

$$\Phi(\mathbf{W}) = \min \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j \in Q(i)} \mathbf{w}_{ij} \mathbf{x}_j \right\|^2, \quad (1)$$

where $Q(i)$ is the set of neighbouring samples and \mathbf{W} should satisfy the normalization constraint.

During dimensionality reduction, NPE retains the same local linear reconstruction in the low-dimensional space as in the original space. The objective function in low-dimensional space is shown in formula (2):

$$\Phi(\mathbf{y}) = \sum_{i=1}^n \left(\mathbf{y}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{y}_j \right)^2, \quad (2)$$

where $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ is the linear reconstruction of \mathbf{X} in low-dimensional space. \mathbf{X} and \mathbf{Y} satisfy $\mathbf{Y} = \mathbf{A}^T \mathbf{X}$, and $\mathbf{A}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$ is the reduced projection matrix.

The dimensionality-reduced linear map is computed by solving for the generalized eigenvectors, as in formula (3):

$$\mathbf{X} \mathbf{M} \mathbf{X}^T \mathbf{a} = \lambda \mathbf{X} \mathbf{X}^T \mathbf{a}, \quad (3)$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$ and $\mathbf{I} = \text{diag}(1, \dots, 1)$. The eigenvectors \mathbf{a} are arranged in ascending order of eigenvalues to obtain the matrix \mathbf{A} .

For the test data \mathbf{x}_{new} , low-dimension is represented as \mathbf{y}_{new} after dimensionality reduction through NPE. Statistics T^2 and SPE can be established for fault detection, as shown in formulas (4) and (5):

$$T^2 = \mathbf{y}_{\text{new}}^T \mathbf{\Lambda}^{-1} \mathbf{y}_{\text{new}}, \quad (4)$$

$$\text{SPE} = \|\mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}}\|^2, \quad (5)$$

where $\mathbf{\Lambda}^{-1}$ is the covariance matrix of the samples and $\hat{\mathbf{x}}_{\text{new}}$ is the low-dimensional reconstruction of the test data. Figure 1 shows the schematic for NPE based fault detection. For the NPE method, the T^2 and SPE statistics are constructed based on the linear features of the data. Specifically, the T^2 statistic is used to monitor data changes in the feature space, while the SPE statistic is used to monitor data changes in the residual space. Thus, with these two statistics, it is possible to comprehensively monitor data changes in linear characteristic. The fault detection based on NPE is shown in Figure 1B, which involves the original data layer L1 and the linear feature layer L2.

2.2 | KNPE method

Although NPE is more effective than other dimension reduction methods for pattern recognition, it is still a

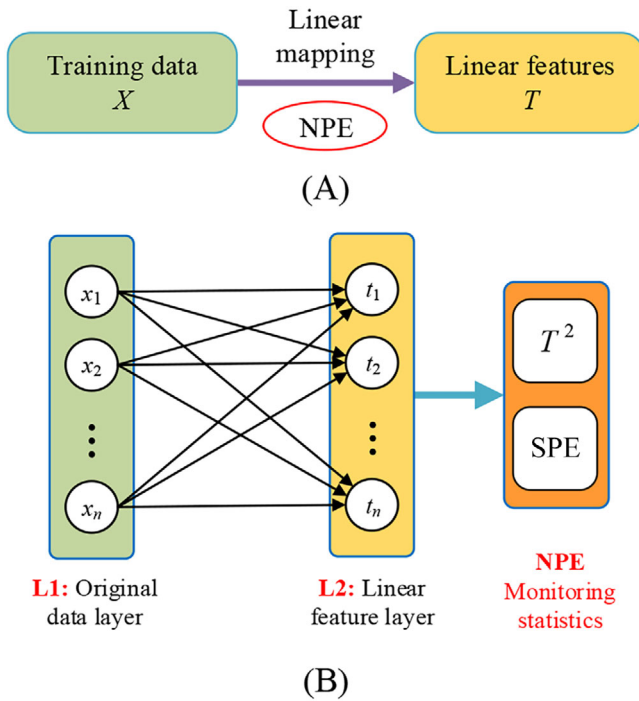


FIGURE 1 Schematic for neighbourhood preserving embedding (NPE) based fault detection. (A) NPE model training. (B) NPE fault detection. SPE, squared prediction error.

linear technique in nature. Therefore, it is not sufficient to describe nonlinearly distributed data. KNPE is obtained by nonlinear mapping to deal with the nonlinearity of the process. The nonlinear mapping $\Phi: \mathbf{x} \in \mathbf{R}^{m \times 1} \rightarrow \Phi(\mathbf{x}) \in \mathbf{F}$ maps the process data into a linear feature space \mathbf{F} . In the feature space, NPE builds the next objective function as shown in formula (6):

$$\Phi(\mathbf{X}) \mathbf{M} \Phi^T(\mathbf{X}) \mathbf{a} = \lambda \Phi(\mathbf{X}) \Phi^T(\mathbf{X}) \mathbf{a}. \quad (6)$$

To avoid defining nonlinear mappings Φ , the kernel matrix is introduced as $\mathbf{K} = \Phi(\mathbf{X}) \Phi^T(\mathbf{X})$. With the kernel trick, the kernel matrix corresponding to the kernel function is shown in formula (7):

$$\mathbf{K}(i, j) = \Phi^T(\mathbf{x}_i) \Phi(\mathbf{x}_j) = \text{ker}(\mathbf{x}_i, \mathbf{x}_j). \quad (7)$$

Further, formula (6) can be written as formula (8):

$$\begin{aligned} \Phi(\mathbf{X}) \mathbf{M} \Phi^T(\mathbf{X}) \mathbf{a} &= \lambda \Phi(\mathbf{X}) \Phi^T(\mathbf{X}) \mathbf{a} \\ \Phi(\mathbf{X}) \mathbf{M} \Phi^T(\mathbf{X}) \Phi(\mathbf{X}) \mathbf{a} &= \lambda \Phi(\mathbf{X}) \mathbf{a} \\ \Phi^T(\mathbf{X}) \Phi(\mathbf{X}) \mathbf{M} \Phi^T(\mathbf{X}) \Phi(\mathbf{X}) \mathbf{a} &= \lambda \Phi^T(\mathbf{X}) \Phi(\mathbf{X}) \mathbf{a} \\ \mathbf{K} \mathbf{M} \mathbf{K} \mathbf{a} &= \lambda \mathbf{K} \mathbf{a}. \end{aligned} \quad (8)$$

The eigenvalue decomposition of the matrix \mathbf{MK} is performed. Then, the mapping matrix \mathbf{A} from high dimension space into dimension space can be achieved by reconstructing eigenvector referring to the d smallest eigenvalue. That is: $\mathbf{A}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$.

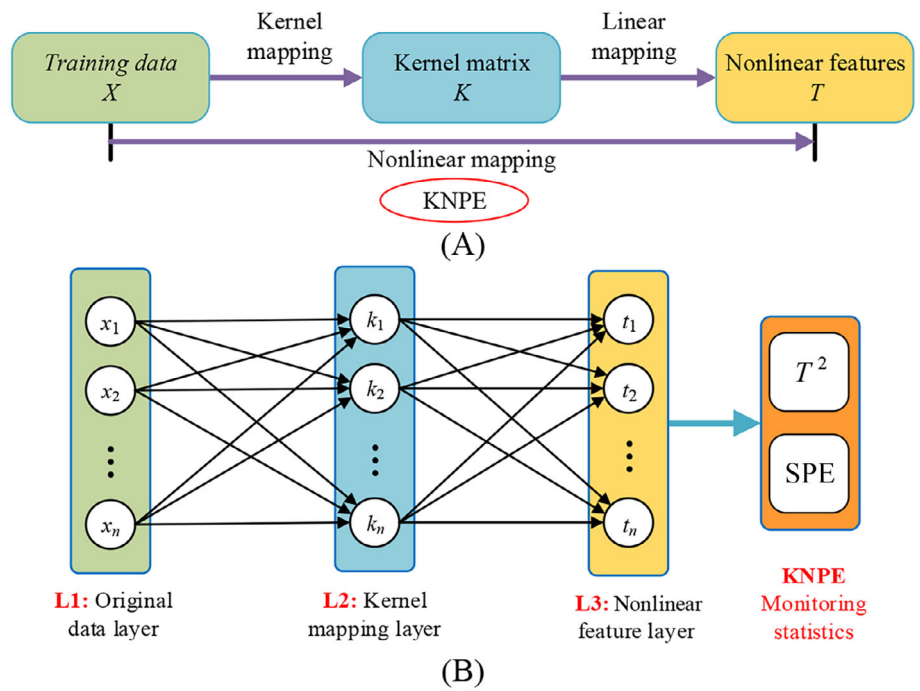
For the new sample data \mathbf{x}_{new} , statistics T^2 and SPE can be established for fault detection as shown in formulas (9) and (10):

$$T^2 = \mathbf{A}^T \Phi(\mathbf{x}_{\text{new}})^T \mathbf{\Lambda}^{-1} \Phi(\mathbf{x}_{\text{new}}) \mathbf{A}, \quad (9)$$

$$\text{SPE} = \left\| \Phi(\mathbf{x}_{\text{new}}) - \hat{\Phi}(\mathbf{x}_{\text{new}}) \right\|^2, \quad (10)$$

where $\mathbf{\Lambda} = \Phi(\mathbf{X}) \Phi^T(\mathbf{X}) / n - 1$ is the covariance matrix of the samples and $\hat{\Phi}(\mathbf{x}_{\text{new}})$ is the low-dimensional reconstruction of the test data. Figure 2 is the schematic for KNPE based fault detection. KNPE maps the original nonlinear data into the high dimensional feature space, which obtains the nonlinear features of the original data through kernel matrix mapping. The fault detection based on KNPE is shown in Figure 2B, which has a three-layer structure involving the original data layer L1, the kernel mapping layer L2, and the nonlinear feature layer L3. Similar to NPE, KNPE only monitors one layer of features. However, the difference is that KNPE introduces a kernel mapping layer to obtain nonlinear features.

FIGURE 2 Schematic for kernel neighbourhood preserving embedding (KNPE) based fault detection. (A) KNPE model training. (B) KNPE fault detection. SPE, squared prediction error.



2.3 | SNPE method

A Siamese network is a type of network that contains two identical subnetworks. The subnetworks are identical to each other in structure and parameters. Therefore, each subnetwork provides the same mapping of inputs to potential features. The output of the whole network is a measure of the difference between two potential features. Since the two networks are identical, if two very similar inputs are fed to the Siamese network, the resulting features should also be very close to each other in the feature space. The schematic for SNPE based fault detection is shown in Figure 3. The two NPE subnetworks are identical to each other and provide the same mapping parameters. The feature of SNPE network is a measure of the difference between the extracted features T_1 and T_2 , corresponding to input samples X_1 and X_2 , respectively. The objective function of the SNPE network is given as shown in formula (11):

$$T = d\{T_1, T_2\} = \|\Phi(X_1) - \Phi(X_2)\|_2^2, \quad (11)$$

where $d\{T_1, T_2\}$ is a measure of the distance between the mapping features T_1 and T_2 . If the Euclidean distance is taken as a measure of distance, it can be written as $\|\Phi(X_1) - \Phi(X_2)\|_2^2$.

For the test data x_{new} , statistics T^2 and SPE can be established for fault detection, shown in formulas (12) and (13):

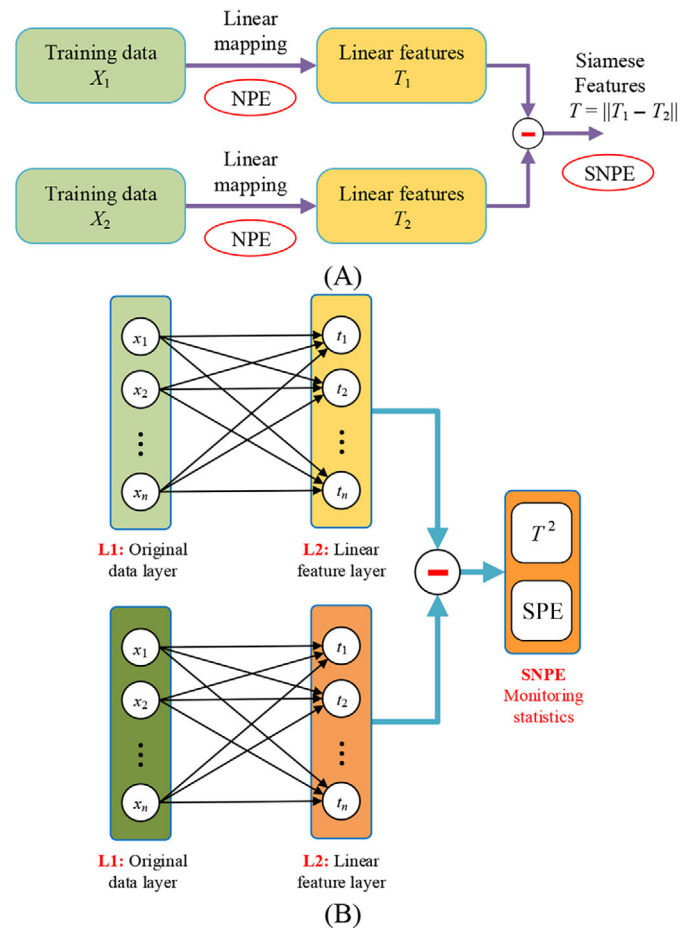


FIGURE 3 Schematic for Siamese neighbourhood preserving embedding (SNPE) based fault detection. (A) SNPE model training. (B) SNPE fault detection. NPE, neighbourhood preserving embedding; SPE, squared prediction error.

$$T^2 = T^T \Lambda^{-1} T, \quad (12)$$

$$\text{SPE} = \|\mathbf{x}_{\text{new}} - \hat{\mathbf{x}}_{\text{new}}\|^2, \quad (13)$$

where T is the Siamese feature output of the SNPE network, Λ^{-1} is the covariance matrix of the samples, and $\hat{\mathbf{x}}_{\text{new}}$ is the low-dimensional reconstruction of the test data. The Siamese structure can measure the difference between features, so it is sensitive to minor faults, which is described in further detail in Section 3 in this paper. In order to make the fault detection with nonlinear and incipient features better, based on SNPE, Siamese network structure has a better fault detection effect for complex processes.

3 | THE PROPOSED SIAMESE DeNPE METHOD (SDeNPE)

Compared with NPE, KNPE adds a nonlinear mapping layer to effectively solve the nonlinear characteristic of the input data. SNPE is based on the NPE Siamese structure, which is more sensitive to incipient faults in fault detection. At the same time, deep learning networks consist of multiple nonlinear feature layers. More deeper features are obtained layer by layer. Therefore, deep learning networks are beneficial for improving the monitoring performance of nonlinear processes. We propose a SDeNPE network with

deep feature extraction capability to improve the fault detection performance of batch process. This section mainly introduces the proposed SDeNPE method.

3.1 | The construction of SDeNPE network

The proposed SDeNPE network extracts linear and nonlinear features by integrating NPE and KNPE in an L-layer structure. Meanwhile, the difference characteristics are obtained by utilizing the ideology of twin networks. The structure is shown in Figure 4.

In the SDeNPE deep feature extraction structure, for the training data $X = [x_1, x_2, \dots, x_n] \in R^{m \times n}$, n is the sample size obtained by sampling, and m is the dimension of measure variable, which is first mapped into the high-dimensional kernel space to obtain the first layer of features $K = \{k_{11}, k_{12}, \dots, k_{1n}\} \in R^{\tilde{n} \times n}$, where $\tilde{n} = M$. Then, the features in the kernel space are linearly mapped to the next layer by NPE to obtain the nonlinear features. In this order, KNPE is applied to the $LN \in \{2, \dots, l, \dots, N\}$ layer to obtain deeper nonlinear features. Finally, the SDeNPE network outputs the difference degree $S(x)$ between the deeper nonlinear features of the two sub-networks.

More specifically, at the first feature layer, we use a nonlinear mapping $\Phi(\cdot)$ to map the input data

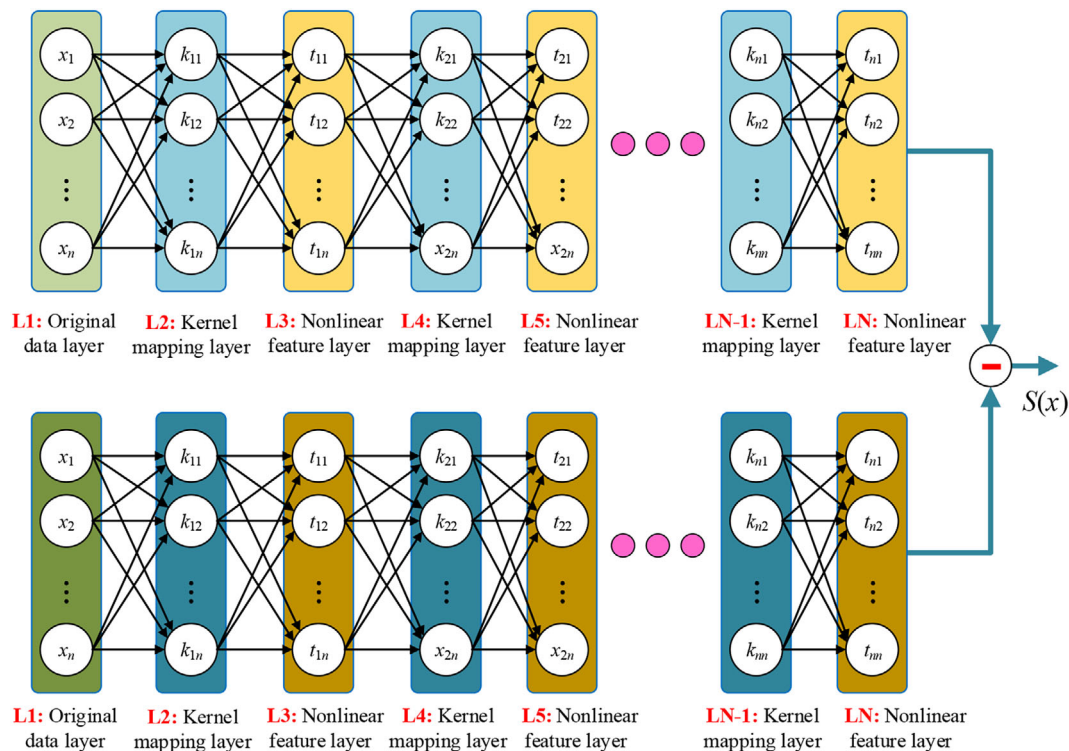


FIGURE 4 Schematic of Siamese deep neighbourhood preserving embedding network (SDeNPE) framework.

into the high dimensional feature space. Kernel function operation converts into original data through calculation of inner product: $\mathbf{K} = \Phi(\mathbf{X}_i)\Phi^T(\mathbf{X}_j)$, where $\Phi(\mathbf{X}) = [\Phi(x_1), \Phi(x_2), \dots, \Phi(x_n)] \in R^v$ is the data of high dimensional feature space and v is the dimension of the high dimensional feature space. The Euclidean distance between two points in a high-dimensional feature space is shown in formula (14):

$$\begin{aligned} & \|\Phi(x_i) - \Phi(x_j)\| \\ &= \sqrt{(\Phi(x_i) - \Phi(x_j))^T (\Phi(x_i) - \Phi(x_j))} \\ &= \sqrt{\Phi^T(x_i)\Phi(x_i) + \Phi^T(x_j)\Phi(x_j) - 2\Phi^T(x_i)\Phi(x_j)} \\ &= \sqrt{\mathbf{K}(x_i, x_i) + \mathbf{K}(x_j, x_j) - 2\mathbf{K}(x_i, x_j)}. \end{aligned} \quad (14)$$

Then, a linear mapping is performed at the next layer by using NPE algorithm based on preserving the local nearest neighbour information, which is calculated according to formula (15):

$$\begin{aligned} \Phi(\mathbf{W}) &= \min \sum_{i=1}^n \left\| \Phi(x_i) - \sum_{j \in Q(i)} \mathbf{w}_{ij} \Phi(x_j) \right\|^2 \\ &= \min \sum_{i=1}^k \left(\mathbf{K}(x_i, x_i) - 2 \sum_{j=1}^k \mathbf{W}_{ij} \mathbf{K}(x_i, x_j) \right. \\ &\quad \left. + \sum_{j=1}^k \sum_{j=1}^k \mathbf{W}_{ij} \mathbf{W}_{ij} \mathbf{K}(x_j, x_j) \right), \end{aligned} \quad (15)$$

where $Q(i)$ is the set of nearest neighbour samples; \mathbf{W} should satisfy the normalization constraint.

At the l th feature layer for $2 \leq l \leq L$, the features are further extracted by KNPE, where the features in the $(l-1)$ th layer are the training data for the l th layer. For the last layer of features, the solution is solved by using the Lagrange multiplier method, as shown in formula (16):

$$\begin{aligned} \Phi(T_n) \mathbf{D} \Phi^T(T_n) \mathbf{a} &= \lambda \Phi(T_n) \Phi^T(T_n) \mathbf{a} \\ \mathbf{K}^n \mathbf{D} \mathbf{K}^n \mathbf{a} &= \lambda \mathbf{K}^n \mathbf{a}, \end{aligned} \quad (16)$$

where $\Phi(T_n)$ is the mapping feature of the $LN = N - 2$ layer, which is also the input data of the $LN = N - 3$ layer, and \mathbf{K}^n is the kernel matrix of the $LN = N - 1$ layer. The eigenvalue decomposition of the matrix $\mathbf{D} \mathbf{K}^n$ is performed. Then the mapping matrix \mathbf{D} from high dimension space into dimension space can be attained by reconstructing eigenvector referring to the d smallest eigenvalue, that is: $\mathbf{D}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$.

The output features of the two sub-networks are passed through the Siamese network to obtain the difference features, as shown in formula (17):

$$\mathbf{S}(x) = d\{T_{n1}, T_{n2}\} = \|\Phi(T_{n1-1}) - \Phi(T_{n2-1})\|_2^2, \quad (17)$$

where $\mathbf{S}(x)$ is the differential feature of the two sub-networks; $d\{T_{n1}, T_{n2}\}$ is a measure of the distance between the mapping features; T_{n1} and T_{n2} are the output feature of each sub-network; and $\|\cdot\|_2^2$ is the square of Euclidean distance. It is worth emphasizing that the SDeNPE network achieves layer-by-layer feature extraction, with optimization at each layer of the network.

As the SDeNPE network contains multiple KNPEs, the choice of kernel parameters is an important issue. In practice, it is necessary to select the kernels with good interpolation and extrapolation capabilities, that is, good generalization capabilities. A mixed kernel function is one that uses a linear combination of multiple kernel functions, and this combination requires the optimization of the mixture weight increasing the complexity of the network training. Currently, the selection of mixture weight is mainly empirical.

In this paper, SDeNPE is a multilayer network. To ensure the diversity of the extracted non-linear features, we choose different kernel functions at each kernel mapping layer, that is, global kernel function and local kernel function are crossed to extract features, and Gaussian radial basis function is used for odd kernel mapping layers and polynomial kernel function is used for even kernel mapping layers.

However, there are different kernel functions that satisfy the conditions. Jordan noted two main types of kernel functions including global kernel and local kernel functions,^[37] as shown in Figure 5, in which the illustrations of kernel functions are shown for different parameters: Figure 5A shows a typical global kernel function, a polynomial kernel function, with the expression shown in formula (18); Figure 5B shows a typical local kernel function, a Gaussian radial basis function (RBF), with the expression shown in formula (19).

$$Kp(x, x') = (\langle x, x' \rangle + 1)^d, \quad (18)$$

$$Kg(x, x') = \exp\left(-\frac{\|x - x'\|^2}{t}\right), \quad (19)$$

where d is the kernel parameter that denotes the degree of the polynomial, and satisfies the Mercer condition for $d \in \mathbb{N}$; and t is the kernel width, which satisfies the Mercer condition for $c > 0$.

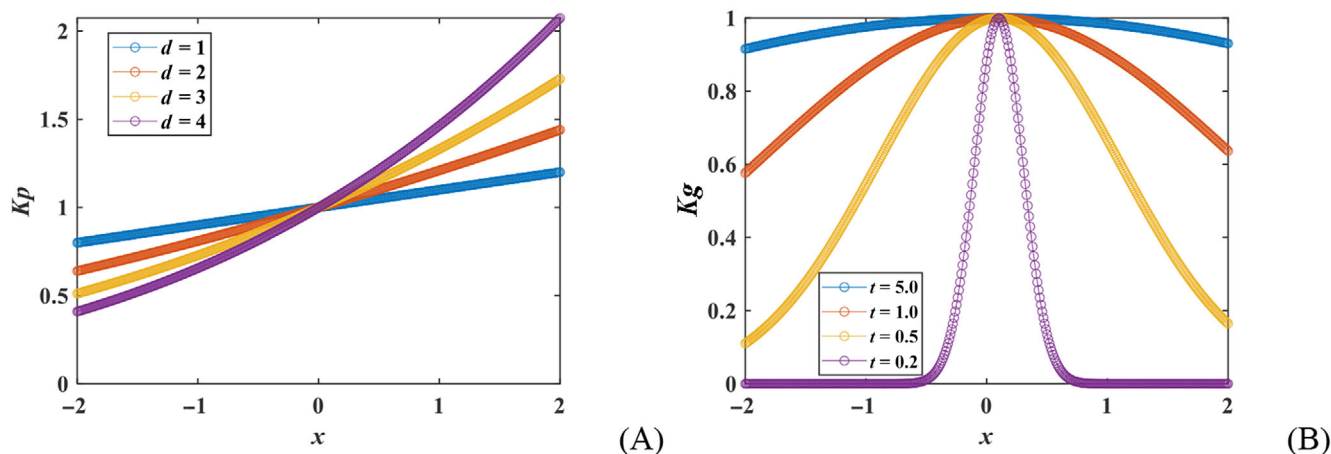


FIGURE 5 Sample plots of kernel values, where $x' = 0.1$, for: (A) Polynomial kernel; (B) radial basis function (RBF) kernel.

It can be seen from Figure 5 that different parameter values affect the interpolation and extrapolation capabilities of the kernel function. Therefore, it is necessary to choose suitable kernel parameters to ensure the feature extraction capability. In this paper, we choose the polynomial kernel with $d = 1$ and the RBF kernel function with $t = 2$.^[38] Furthermore, to illustrate different kernel functions at each kernel mapping layer to guarantee the diverse capabilities of feature extraction, meanwhile, to demonstrate the effectiveness of the proposed multilayer feature extraction for nonlinear processes, we illustrate it with a numerical example,^[39] as shown in formula (20):

$$\begin{aligned} z_{2i-1} &= \begin{bmatrix} z_{2i-1}^{(1)} \\ z_{2i-1}^{(2)} \\ z_{2i-1}^{(3)} \end{bmatrix} = \begin{bmatrix} 1.5a_i \\ 1.8a_i \\ 1.3a_i \end{bmatrix} + \begin{bmatrix} 0.28\beta_i \\ 0.385\beta_i \\ 0.42\beta_i \end{bmatrix}, \\ z_{2i} &= \begin{bmatrix} z_{2i}^{(1)} \\ z_{2i}^{(2)} \\ z_{2i}^{(3)} \end{bmatrix} = \begin{bmatrix} 1.57 & 2.37 & 1.8 \\ 2.73 & 1.05 & 1.4 \\ 1.22 & 1.60 & 2.4 \end{bmatrix} z_{2i-1} + \begin{bmatrix} 0.35\beta_i \\ 0.385\beta_i \\ 0.42\beta_i \end{bmatrix}, \end{aligned} \quad (20)$$

where α_i and β_i are the random variables from the standard normal distribution, $i = 1, 2, \dots, N$. According to formula (20), normal process data with 1000 samples, $Z = [z_1, z_2, \dots, z_{1000}]$, are generated. Fault process data are simulated with a step fault: a step change of 2.5 is added to the variable $z_{2i-1}^{(3)}$ from the 401st sample, and a step change of 1.22 is added to the variable $z_{2i}^{(1)}$ from the 402nd sample. The simulated fault data also contain 1000 samples. Figure 6 shows the visualization of the normal samples and the fault samples on the first two dimensions of different kernel mapping layer methods.

It can be seen from Figure 6 that the multilayer nonlinear feature extraction method has better results for

nonlinear processes. In Figure 6A, the fault samples and the normal samples are largely mixed up; also, Figure 6B has a partial mixture. Therefore, there is a limitation of feature extraction from a single layer of kernel function. In Figure 6C,D, two-layer mapping for RBF kernel and polynomial kernel has an enhanced kernel function extraction capability compared to a single layer, but its extraction capability is still limited. Thus, a multi-layer single kernel function does not extract the features well. In Figure 6E, two-layer mapping for mixed kernel is much more capable of distinguishing normal samples from fault samples, with good feature extraction. In Figure 6F, an increasing number of layers in the kernel function does not enhance the feature extraction capability anymore, so the most appropriate number of layers for this example is 2. The above example illustrates the choice of kernel mapping layer in this paper. However, the choice of the number of layers for the network is influenced by several factors. For the sample data of the process, the amount of data determines the number of layers chosen for the network. This is further discussed in Section 4 on the selection of the number of network layers. In practice, it may not always result in the best performance in all tasks to use a constant number of layers. It is really a challenging task to determine the optimal number of network layers for different processes, and there are no general definitive rules that can directly determine the optimal number of layers for a network. In this study, the number of network layers for a specific process is determined by using cross-validation.^[40] In order to better determine the number of network layers, the standard five-fold cross-validation method is used in this paper. The training samples are divided into five equal and mutually exclusive subsets, four of which are used as the training set and one as the test set. The test set is selected by letting the five subsets ergodic and averaging the results of the five validation tests.

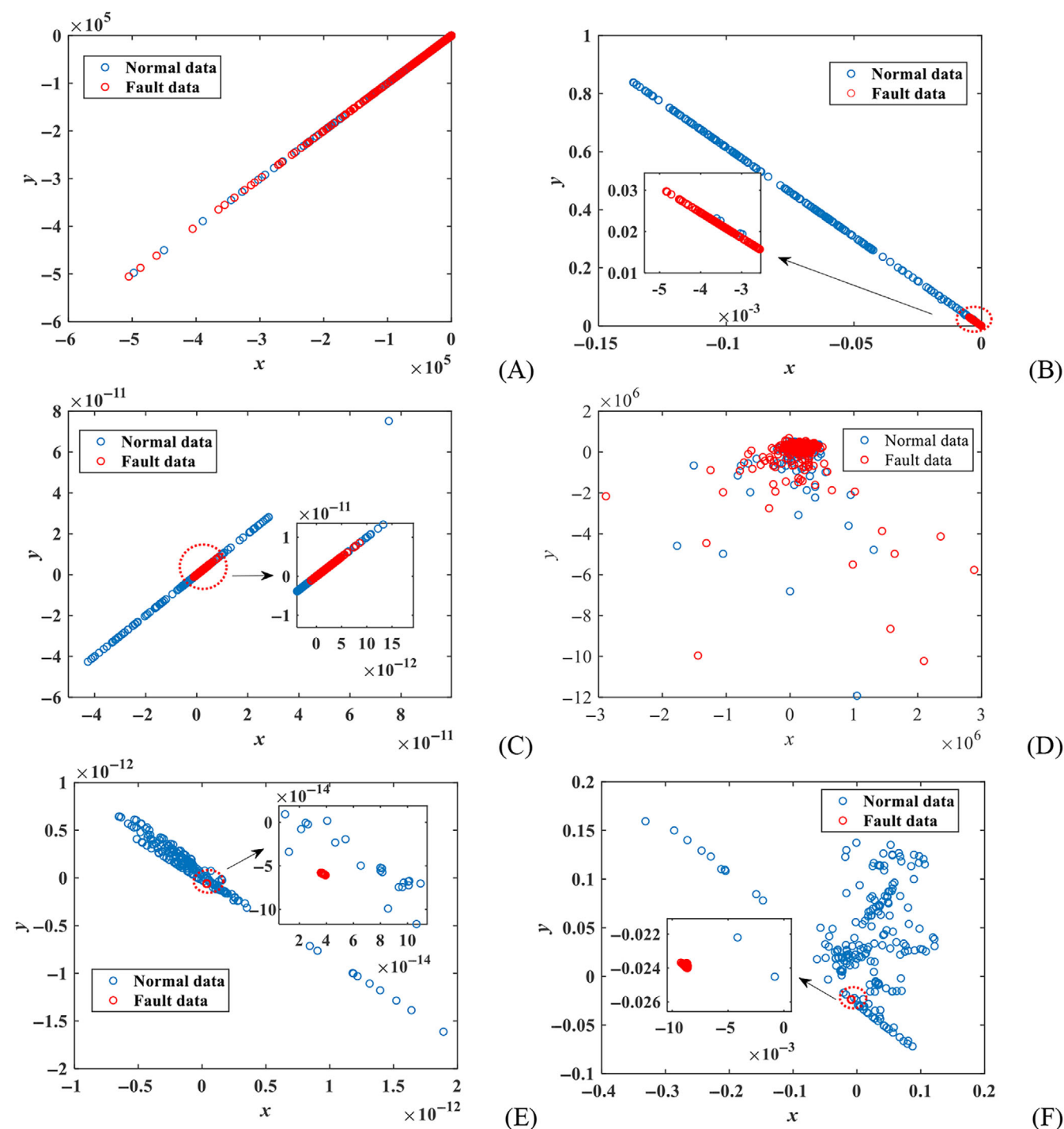


FIGURE 6 Visualization of the normal samples and the fault samples on the first two dimensions of different kernel mapping layer methods: (A) Polynomial kernel; (B) Radial basis function (RBF) kernel; (C) Two-layer mapping for RBF kernel; (D) Two-layer mapping for polynomial kernel; (E) Two-layer mapping for mixed kernel; (F) Three-layer mapping for mixed kernel.

3.2 | Fault detection based on statistical indicators for SDeNPE method

A batch process is a repetitive production process which has three-dimensional data compared to a continuous production process. For fault diagnosis of batch processes, three-dimensional data need to be unfolded as two-

dimensional data to build a fault diagnosis model. Therefore, the batch-variable 3D data expansion method is used to process the 3D data.^[7] First, the data matrix of each sampling moment is sequentially arranged along the time axis and unfolded to obtain two-dimensional data, and then standardized by columns, and the standardized two-dimensional matrix is arranged in the variable direction to

form a two-dimensional matrix. This expansion method can weaken the dynamic correlations which exist in the process to a certain extent and result in more effective monitoring of process changes.

After the data features are extracted by the offline trained model, statistical indicators are established to complete the fault detection. In this paper, we use the T^2 statistic and the SPE statistic for fault detection, and the control thresholds for these two statistics are calculated based on the assumptions of multivariate normal distribution and temporal independence of the observations. Since the distribution of the extracted features has no prior information, this paper approximates the control thresholds of the T^2 and SPE statistics by kernel density estimation (KDE).^[41]

The T^2 statistic is calculated as formula (21):

$$T^2 = \mathbf{S}(x)^T \Delta^{-1} \mathbf{S}(x), \quad (21)$$

where $\mathbf{S}(x)$ is the differential characteristic and Δ^{-1} is the covariance matrix of the samples.

The SPE statistic is calculated as formula (22):

$$\text{SPE} = \|\tilde{x}\|^2 = \|x - \mathbf{G}x\|^2, \quad (22)$$

where \mathbf{G} is the mapping matrix from input to output for the SDeNPE network. The method of kernel density estimation gives a univariate kernel function, which is shown in the following formula (23):

$$f(j) = \frac{1}{n} \sum_{i=1}^n K_h(j - j_i) = \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{j - j_i}{\sigma}\right), \quad (23)$$

where j is the sample data, j_i is the observation value, σ is the window width, n is the number of observation values, and K is the kernel function. In this paper, the Gaussian kernel function is selected, and the test level $\alpha = 0.95$ can be obtained by formulas (24) and (25) to seek the control limits T_α^2 and SPE_α .

$$\begin{aligned} \int_{-\infty}^{T_\alpha^2} f(T^2) d(T^2) &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{T^2 - T_i^2}{\sigma}\right) d(T^2) \\ &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(T^2 - T_i^2)^2}{2\sigma}\right) \right\} d(T^2) = \alpha. \end{aligned} \quad (24)$$

$$\begin{aligned} \int_{-\infty}^{\text{SPE}_\alpha} f(\text{SPE}) d(\text{SPE}) &= \int_{-\infty}^{\text{SPE}_\alpha} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{\text{SPE} - \text{SPE}_i}{\sigma}\right) d(\text{SPE}) \\ &= \int_{-\infty}^{\text{SPE}_\alpha} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(\text{SPE} - \text{SPE}_i)^2}{2\sigma}\right) \right\} d(\text{SPE}) = \alpha. \end{aligned} \quad (25)$$

If $T^2 > T_\alpha^2$ or $\text{SPE} > \text{SPE}_\alpha$, a fault occurs; otherwise, it is normal.

3.3 | Fault detection procedure based on SDeNPE network

The proposed SDeNPE network for batch process fault detection comprises offline modelling and online monitoring procedures. The fault detection steps of the proposed SDeNPE network are shown in Figure 7, which has two parts: offline modelling and online monitoring.

In the offline training stage, normal running data are collected, the SDeNPE network is constructed, and control confidence thresholds are calculated.

3.4 | Offline modelling

Step 1: Collect normal operational data, divide them into two datasets of the same number of samples, X_1 and X_2 , and normalize them with the mean and variance of the normal data.

Step 2: Build the SDeNPE network with training data.

Step 3: Obtain differentiation feature between the original data by the SDeNPE network and calculate the statistics T^2 and SPE for the network.

Step 4: Obtain the confidence thresholds T_α^2 and SPE_α by using the KDE method.

In the online monitoring stage, the new samples are projected into the SDeNPE model to extract the features. The statistics T^2 and SPE are also obtained to judge the process condition.

3.5 | Online monitoring

Step 1: Collect online samples x_{new} and then normalize with the mean and variance of the normal training data set.

Step 2: Project the x_{new} onto the SDeNPE model and calculate the output feature.

Step 3: Calculate the monitoring statistics T^2 and SPE with formulas (21) and (22).

Step 4: If $T^2 \geq T_\alpha^2$ or $\text{SPE} \geq \text{SPE}_\alpha$, the fault occurs; otherwise, return to step 1.

4 | CASE STUDIES

We conduct two batch process cases involving the penicillin fermentation process^[42] and the semiconductor etching process^[43] to verify the effectiveness and suitability of our

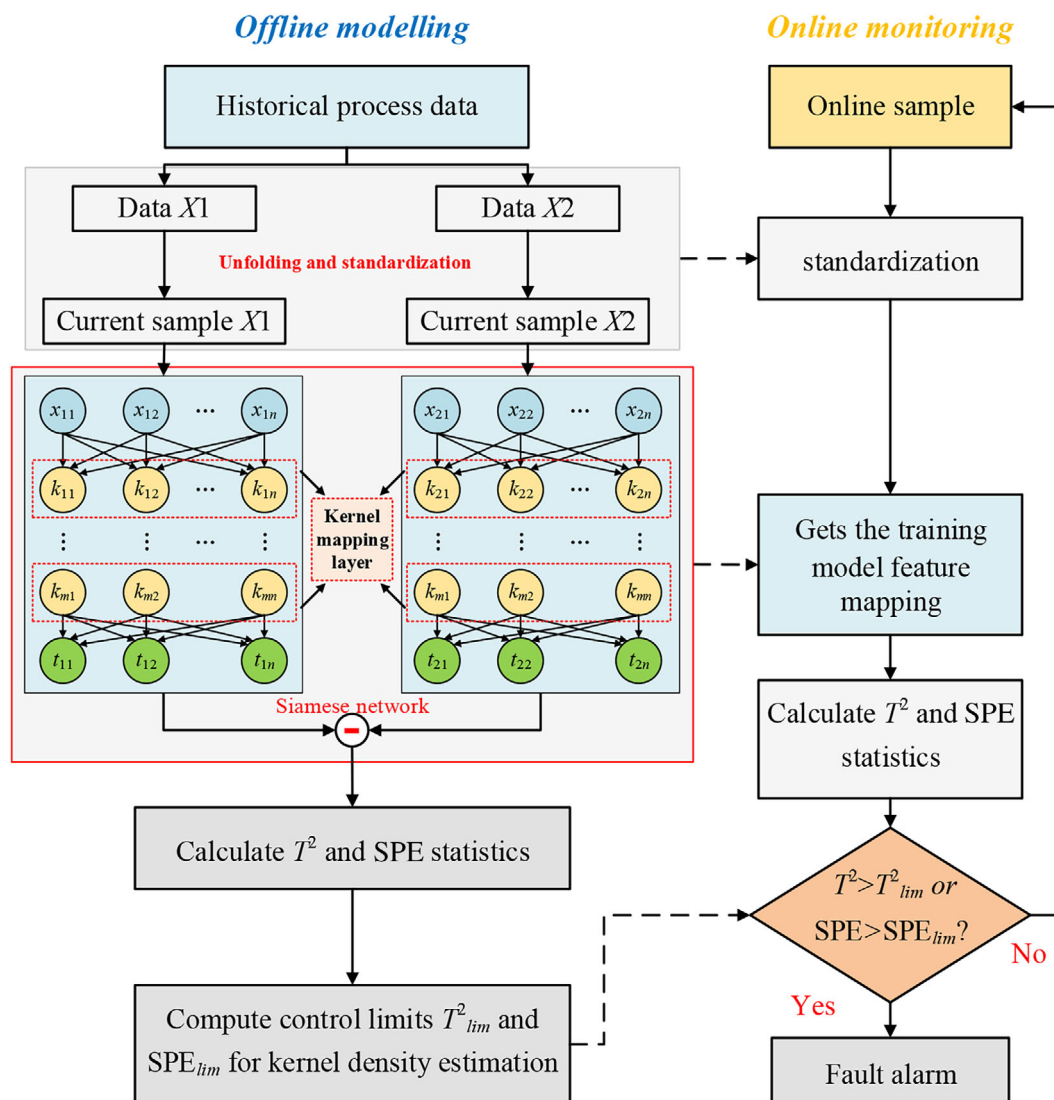


FIGURE 7 Fault detection framework of Siamese deep neighbourhood preserving embedding network (SDeNPE). SPE, squared prediction error.

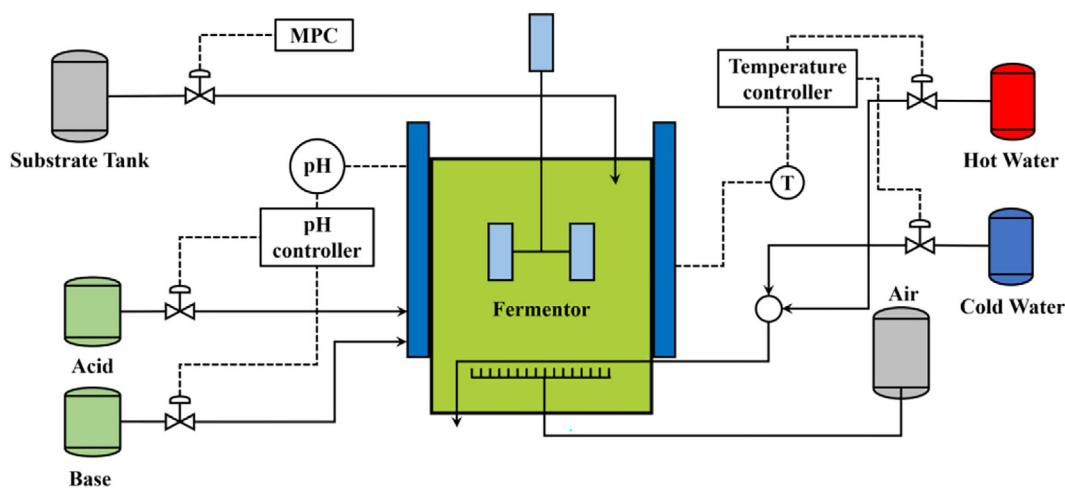


FIGURE 8 Schematic of the penicillin fermentation process. MPC, model predictive control.

TABLE 1 Considered variables in the penicillin fermentation process.

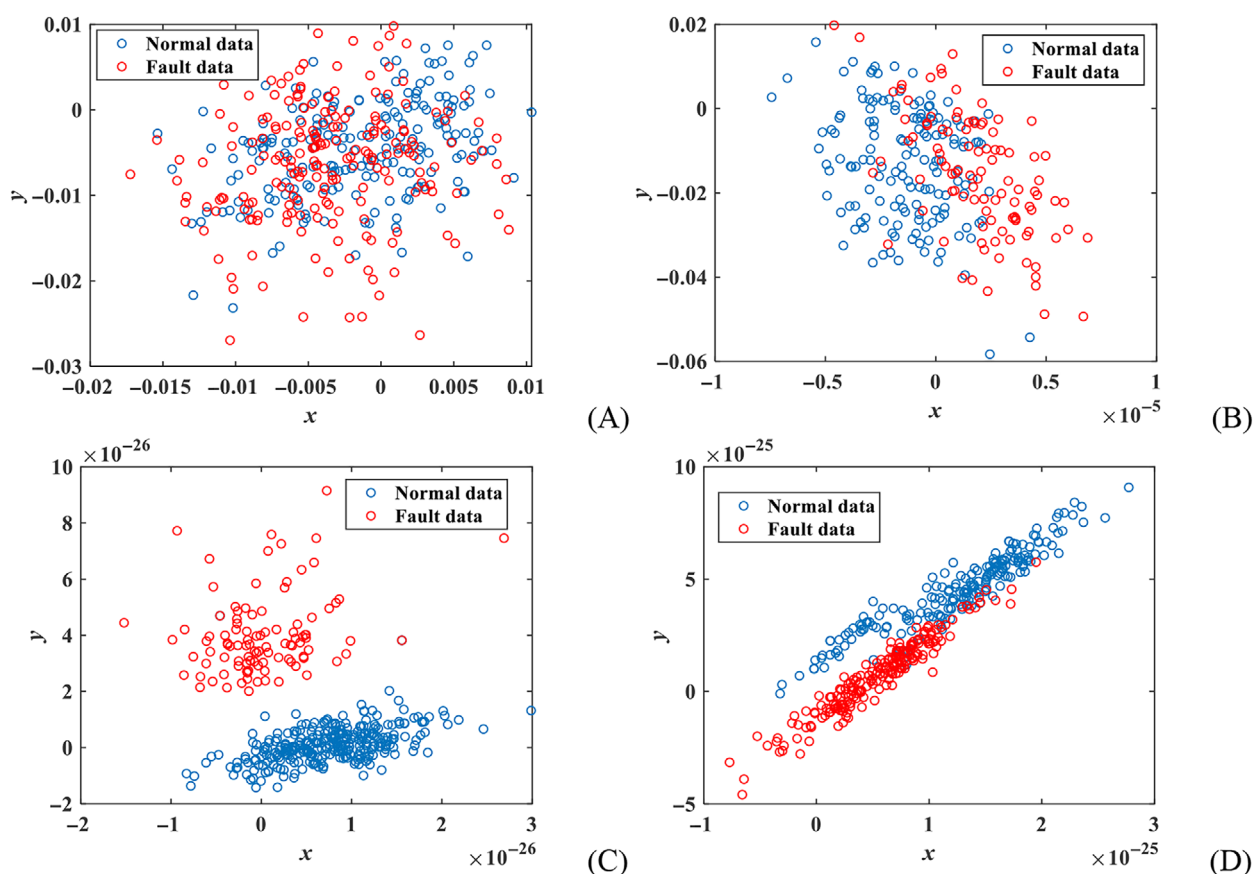
No.	Process variables
1	Aeration rate (L/h)
2	Agitator power (W)
3	Substrate feed flow rate (L/h)
4	Substrate feed flow temperature (K)
5	Substrate concentration (L/h)
6	Dissolved oxygen concentration (%)
7	Reactor volume (L)
8	Fermentation temperature (K)
9	pH
10	CO ₂ (%)

proposed SDeNPE network in fault detection, specifically, SDeNPE compared with multiway principal component analysis (MPCA), NPE, KNPE, SNPE, and DeNPE methods. For all these methods of building statistical models, KDE method calculates 95% confidence limits for each statistic. The fault detection rate (FDR) is defined as the percentage of samples that exceed the threshold among all fault samples, while FAR is defined as the percentage of samples that exceed the threshold among all normal samples.

4.1 | Penicillin fermentation process

Penicillin is a widely used antibiotic in clinical practice. Its production process is typically intermittent and the

Fault no.	Fault variable name	Disturbance	Amplitude	Fault period
F1	Aeration rate	Step	+0.8%	200–400 h
F2	Aeration rate	Ramp	+0.1	300–400 h
F3	Agitator rate	Step	+1%	200–300 h
F4	Agitator rate	Ramp	+1	250–400 h
F5	Substrate feeding rate	Step	+1.5%	200–300 h
F6	Substrate feeding rate	Ramp	+0.005	200–400 h

TABLE 2 Penicillin fermentation process fault modes.

FIGURE 9 Visualization of the normal and fault samples on the first two dimensions with the different number layers: (A) LN = 2; (B) LN = 3; (C) LN = 5; (D) LN = 7.

process is characterized by strong nonlinearity, dynamics, and mass production. The Illinois Institute of Technology designed the Pensim2.0 platform based on the penicillin fermentation process and it is widely used in areas such as the modelling and monitoring of batch processes. The initial state of the batch process should be different to reflect the difference between different batches. Multiple batches of data are generated in the normal range of setting different initial conditions and process parameters. A schematic of the penicillin fermentation process is shown in Figure 8.

In our experiments, we selected 10 variables in the penicillin fermentation process, as listed in Table 1.

We used the Pensim2.0 platform to generate 10 batches of normal data and 6 batches of fault data with different fault types. The initial parameter settings were different for each batch of data, but were within normal ranges. The simulation time was set to 400 h, and the sampling time was set to 1 h. The data under normal operating conditions were used for training and model building, where five batches of each sub network were used in training for the Siamese network. The data collected from abnormal conditions were used for fault detection. In order to simulate the actual penicillin fermentation process and increase the robustness of the model, Gaussian white noise $e \sim \mathcal{N}(0, 0.01)$ was introduced into the data. The fault modes are shown in Table 2.

The number of layers in a deep network is influenced by many complex factors and is certainly relevant to the problem. Therefore, for the penicillin fermentation process data, we determined the number of layers of the SDeNPE network as $LN = 5$ by cross-validation. Since the main nonlinear features of SDeNPE network are extracted at the kernel mapping layer, we verified the optimal choice of layers by $LN = 2, 3, 5$, and 7 , respectively. The results of the first two-dimensional visualization of different layers for normal and fault samples are shown in Figure 9, where the blue circle is normal samples and the red circle is fault samples.

It can be seen from Figure 9 that the feature extraction results are more obvious when the number of layers increases. However, when $LN = 5$, SDeNPE has a good extraction effect. When the number of layers is increased to $LN = 7$, normal and fault samples overlap and the extraction effect decreases. Therefore, for fault detection of the penicillin fermentation process, it is reasonable for the number of layers to be $LN = 5$ obtained by cross validation in this paper.

Table 3 shows the FDRs of different fault detection algorithms; the highest FDR for each fault has been marked in bold.

Table 4 shows the FARs of different fault detection algorithms. The best fault detection performance is when FAR is lower and in bold.

TABLE 3 Fault detection rates (FDRs) of the different methods for faults/%.

Fault no.	MPCA		NPE		KNPE		SNPE		DeNPE		SDeNPE	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
F1	12.00	3.50	7.01	2.48	15.42	67.16	74.12	90.10	99.50	85.07	89.55	99.50
F2	11.88	12.87	14.85	15.84	39.60	69.30	77.22	83.16	90.09	92.07	98.01	98.50
F3	13.86	9.09	16.83	6.93	34.65	38.61	68.31	67.32	93.06	95.04	97.02	99.00
F4	12.58	13.90	16.55	17.21	33.77	39.73	73.50	76.82	92.05	94.03	94.70	95.36
F5	2.97	5.94	4.95	9.09	27.72	30.69	66.33	65.34	91.08	92.07	92.07	96.03
F6	1.98	2.97	2.48	3.98	24.87	28.35	74.62	78.60	92.53	94.02	95.52	97.01
Mean	9.21	8.05	10.45	9.26	29.34	45.64	72.35	76.89	93.05	92.05	94.48	97.57

Abbreviations: DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

TABLE 4 FARs of the different methods for faults/%.

Fault no.	MPCA		NPE		KNPE		SNPE		DeNPE		SDeNPE	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
Mean	4.50	22.50	10.05	20.10	16.08	17.58	29.24	23.61	19.09	21.10	8.05	3.51

Abbreviations: DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

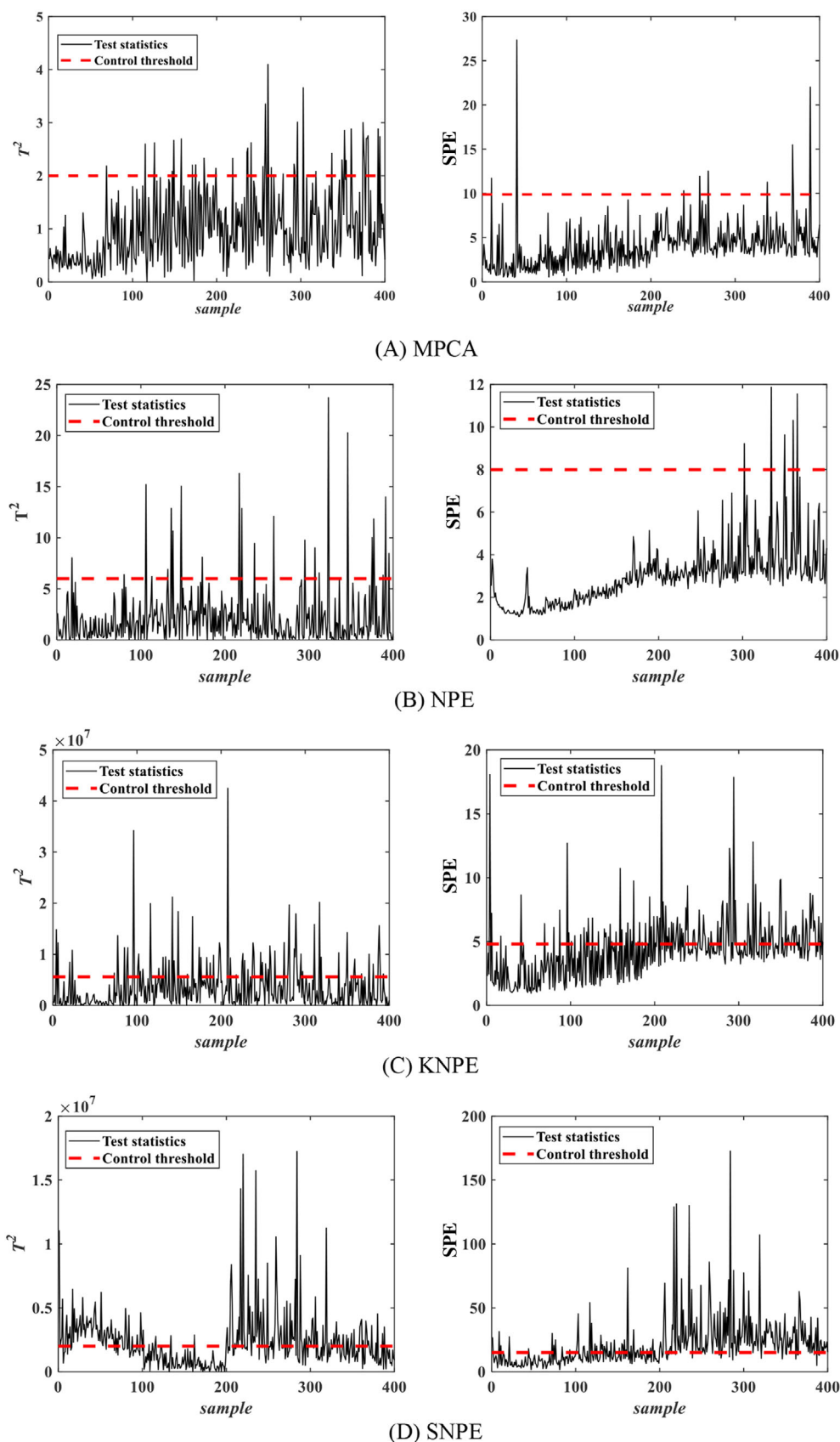


FIGURE 10 Monitoring charts for the penicillin fermentation process fault 1 according to fault detection method. DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

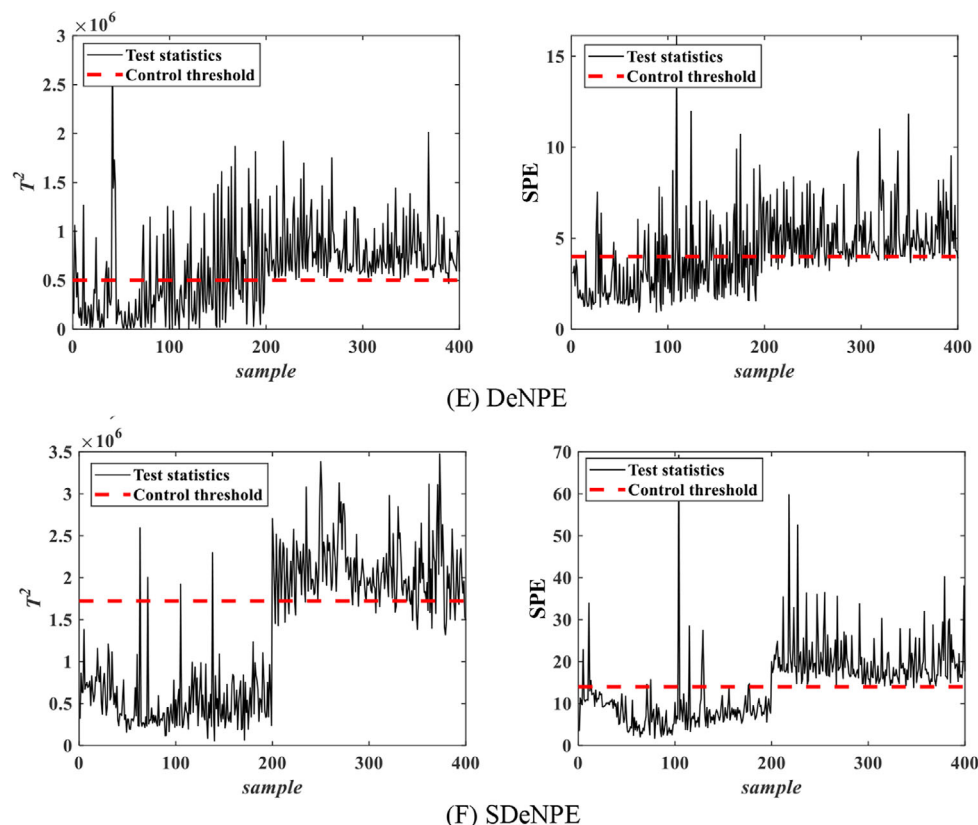


FIGURE 10 (Continued)

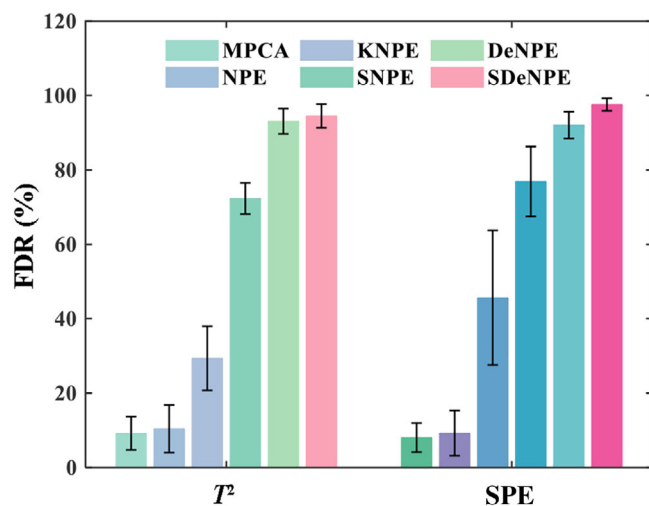


FIGURE 11 Histogram for mean fault detection rate (FDR) of the penicillin fermentation process. DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

In Table 3 and Table 4, the MPCA method has a low FDR for multiple faults. Compared with MPCA, the effect of NPE is improved, but the effect is not significant

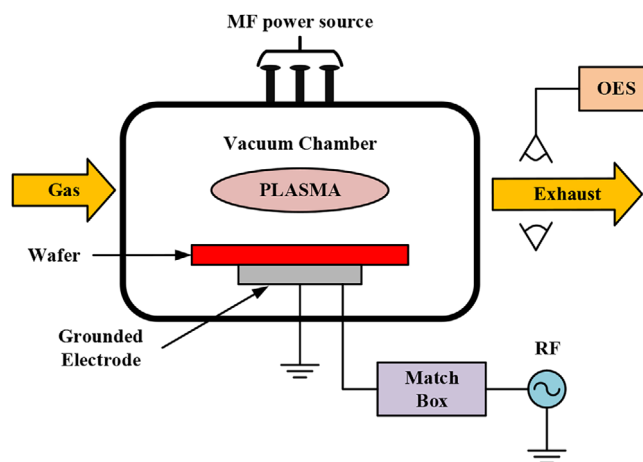


FIGURE 12 Schematic of the semiconductor etching process. MF, mass flow; OES, optical emission spectroscopy; RF, radio frequency.

enough, which is due to the limited number of batches of training data and the inadequate feature extraction ability of the data. Based on NPE, KNPE considers the non-linear characteristic of the data, resulting in better feature extraction, and its FDR is higher. However, feature extraction of shallow layer makes the FDR not so high overall. As SNPE is a Siamese structure, which is valid for the modelling of limited batches, the differential

features between the data are fully considered, resulting in a high FDR. However, the shallow feature extraction makes the extracted features inadequate. DeNPE is a deep nonlinear NPE extraction method with more adequate extraction for feature and higher FDR. The

TABLE 5 Considered variables in the semiconductor etching process.

No.	Measured variables	No.	Measured variables
1	BCl ₃ flow	10	RF power
2	Cl ₂ flow	11	RF impedance
3	RF bottom power	12	TCP tuner
4	Endpoint A detector	13	TCP phase error
5	Helium pressure	14	TCP impedance
6	Chamber pressure	15	TCP top power
7	RF tuner	16	TCP load
8	RF load	17	Vat valve position
9	Phase error	-	-

Abbreviations: RF, radio frequency; TCP, transformer coupled plasma.

SDeNPE method, based on DeNPE, uses Siamese structures to obtain difference features while extracting the features in depth, and has a higher FDR and a lower false alarm rate.

To verify the superiority of the SDeNPE method in fault detection performance in this paper, we further illustrate the superiority of the SDeNPE method by using the statistical information from the fault F1 monitoring graph in Figure 10.

In Figure 10A, only a small number of faults are detected by the MPCA method both in the T^2 and SPE statistics, owing to the low feature extraction capability of MPCA method, so it has low FDRs along with low fault false alarm rates. As NPE only considers the local manifold structure information, it does not show a high improvement in the detection effect compared to MPCA due to the limited number of batches for training the model and the poor feature extraction ability. KNPE is a nonlinear deduction of NPE that extracts the nonlinear features of the data; in Figure 10C, the improvement in the FDR of the SPE statistic of KNPE is obvious, however the T^2 statistic is not visible. For the SNPE method, in

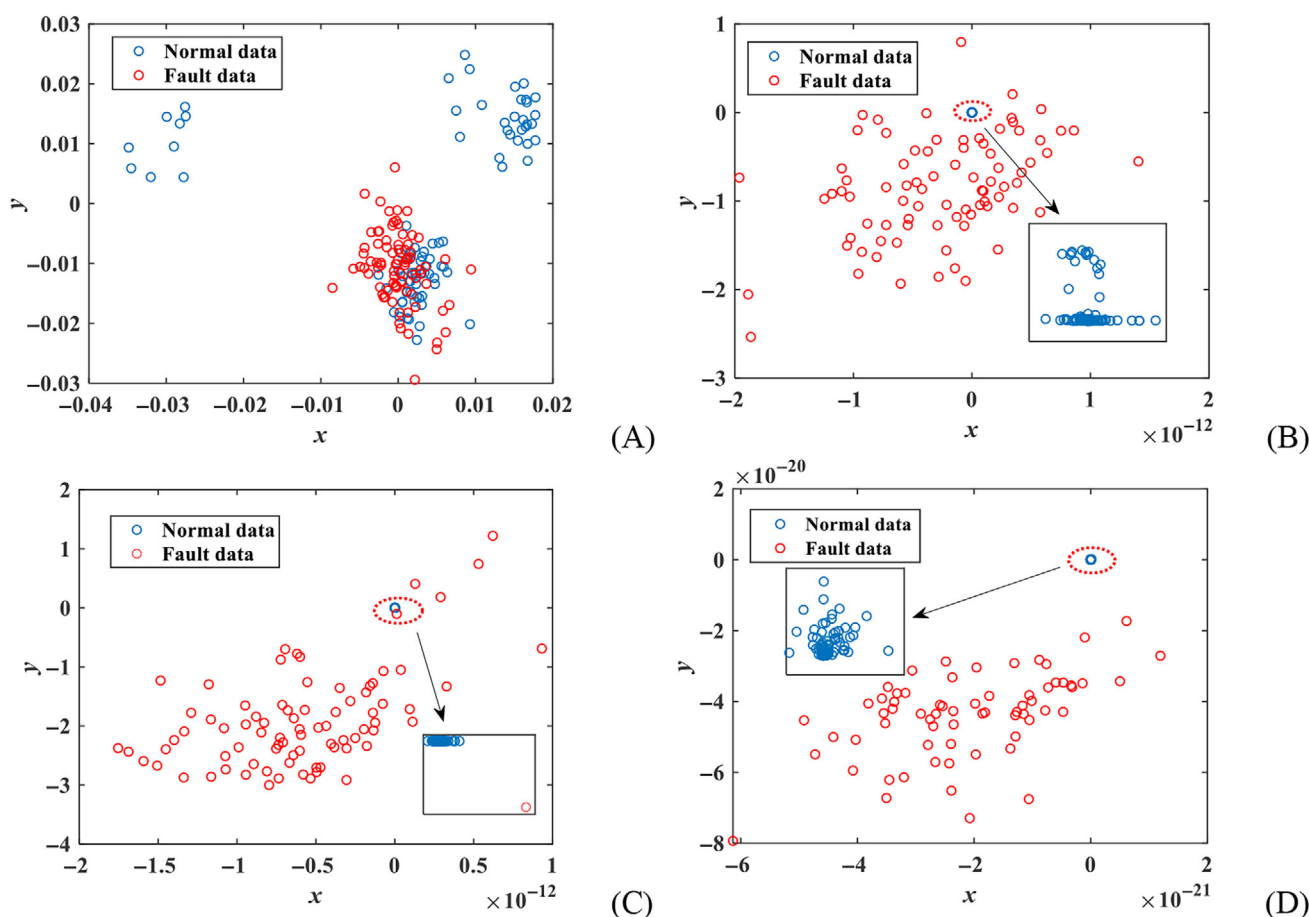


FIGURE 13 Visualization of the normal and fault samples on the first two dimensions with the different number layers: (A) LN = 2; (B) LN = 3; (C) LN = 5; (D) LN = 7.

Figure 10D, the T^2 and SPE statistics are effective for detection, but faulty false alarm is evident; the cause is that the fault cannot differentiate between incipient features and faults. As DeNPE is a deep network based on nonlinear NPE, the nonlinear feature extraction is much more capable; it can be seen from Figure 10E that both the T^2 and SPE statistics are detected by a more correct number of fault samples, but the faults are heavily misrepresented. Therefore, it is an important issue for reducing the fault false rate in deep networks. Based on DeNPE, SDeNPE proposed with Siamese structure effectively solves this problem; from Figure 10F, we can see that the SDeNPE network performs better in fault detection.

Based on the FDR results, we plot the histograms in Figure 11, where the error bars denote the variance. It can be seen from Figure 11 that the mean FDR of the proposed method in this paper is higher and more stable for all fault batches of the penicillin fermentation process. In other words, if the error bar is smaller, the FDR fluctuates less.

4.2 | Semiconductor etching process

The semiconductor etching process is a very important part of the semiconductor manufacturing process which needs to operate under different conditions and is a

typical nonlinear batch process. A schematic of the semiconductor etching process is shown in Figure 12.

Our experiment was performed on a Lam9600 plasma etching tool for stack etching with an inductively coupled Bl_3/Cl_2 plasma etch stack. The metal etcher used in this experiment is equipped with three sensor systems: machine state, radiofrequency monitors, and optical emission spectroscopy (OES). The device status sensor collects the device data during wafer processing, including 40 process setpoints that are sampled at 1 s interval during the etching process, such as gas flow, chamber pressure, RF power, and so forth. In this process, each of the 17 non-setpoint process variables with normal variation is monitored, as shown in Table 5.

In this experiment, we selected 10 normal batches for modelling, and each batch is 85 h in length. For the Siamese network structure, each sub-network used for 5 batches of data to train the network model.

For the semiconductor etching process data, we determined the number of layers of the SDeNPE network to be $\text{LN} = 7$ by cross-validation. To visualize the choice of the number of layers, we verified the optimal choice of layers by $\text{LN} = 2, 3, 5$, and 7, respectively. The results of the first two-dimensional visualization of different layers for normal and fault data are shown in Figure 13, where the blue circle is normal data and the red circle is fault data.

TABLE 6 Fault detection rates (FDRs) of the different methods for faults/%.

Fault no.	MPCA		NPE		KNPE		SNPE		DeNPE		SDeNPE	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
TCP + 10	3.52	24.71	3.52	17.64	15.29	45.88	78.82	82.35	91.76	89.41	96.47	98.82
RF-12	5.89	12.94	7.05	18.82	29.41	47.05	82.35	88.23	92.94	94.11	96.47	97.64
BCl_3 + 5	14.11	15.29	18.82	23.52	35.29	25.88	81.17	81.17	87.05	90.58	92.94	95.29
Pr-2	7.05	16.47	9.41	16.47	25.88	35.29	87.05	88.23	90.58	94.11	94.11	95.29
Cl_2 -5	18.82	20.00	20.00	21.17	29.41	25.88	80.00	81.17	85.88	88.23	92.94	96.47
He Chuck	21.17	23.52	23.52	30.58	31.76	32.94	78.82	80.00	82.35	84.70	91.76	94.11
Mean	11.76	18.82	13.72	21.37	27.84	35.49	81.36	83.53	88.43	90.19	94.11	96.27

Abbreviations: DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

TABLE 7 FARs of the different methods for faults/%.

Fault no.	MPCA		NPE		KNPE		SNPE		DeNPE		SDeNPE	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
Mean	12.94	9.41	8.23	7.05	5.88	4.70	4.70	3.52	5.82	2.35	1.17	0.00

Abbreviations: DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

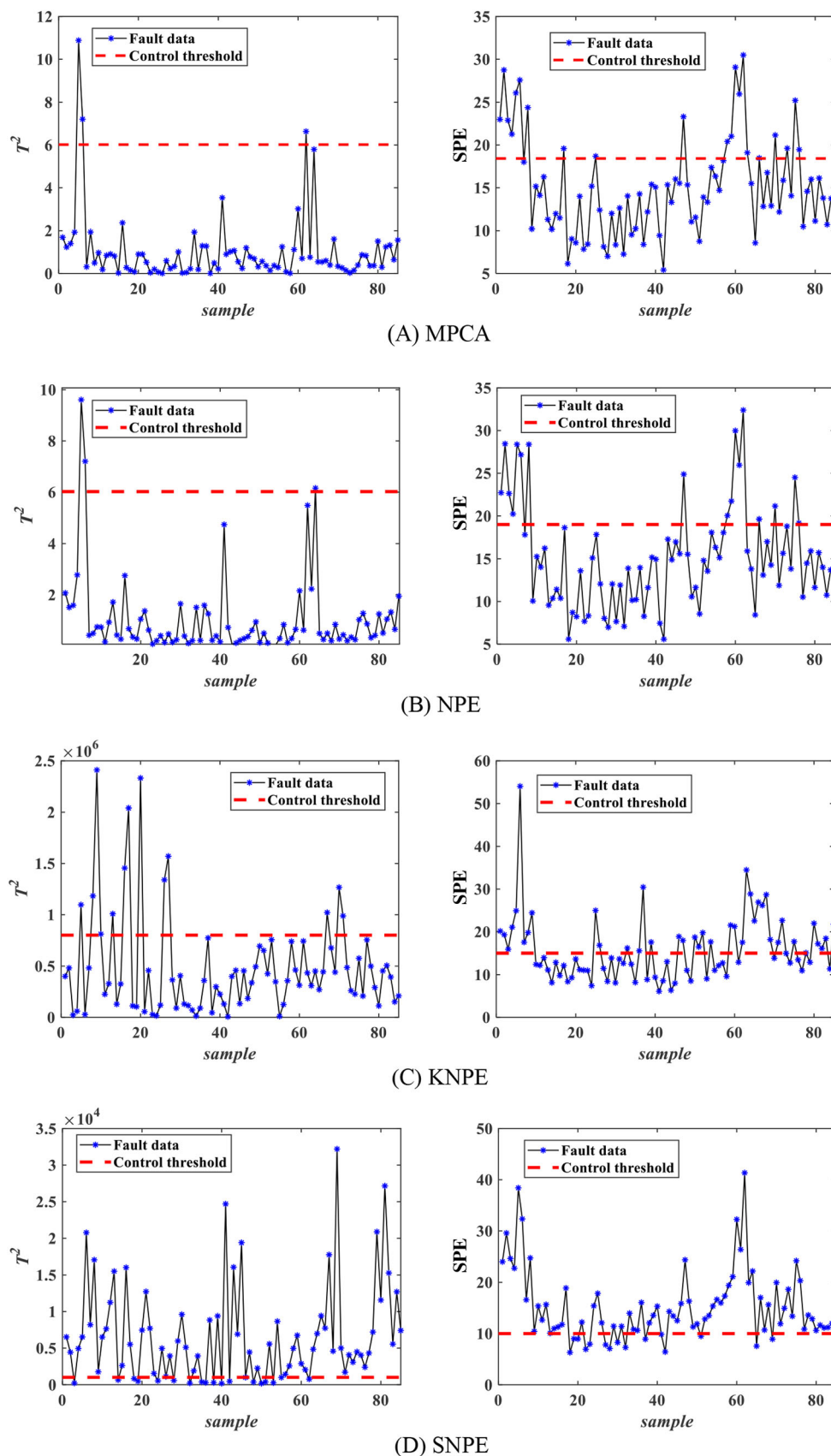


FIGURE 14 Monitoring charts for the semiconductor etching process faults according to fault detection method. DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

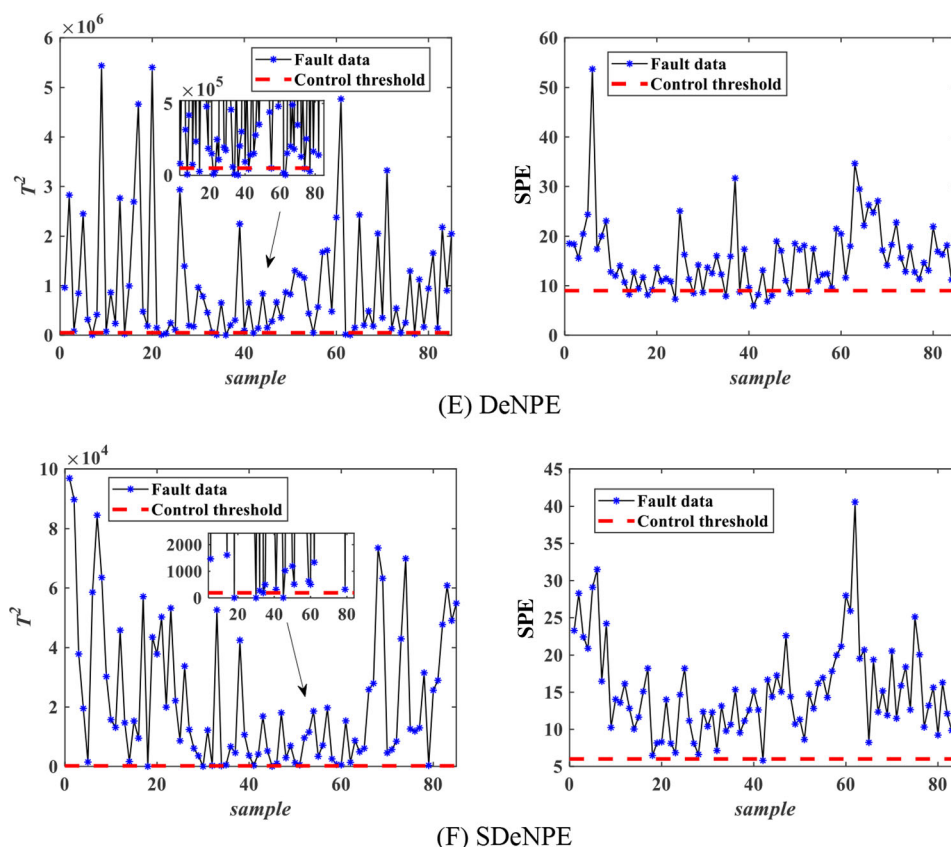


FIGURE 14 (Continued)

It can be seen from Figure 13A that normal and fault samples overlap, which are linearly nonseparable. In Figure 13B, the normal and fault samples are linearly separated, but some of the samples are not correctly classified. Compared with Figure 13B, Figure 13C shows better linear separation, that is, when layer $LN = 3$ is increased to $LN = 5$, the feature extraction capability is better. In Figure 13D, the fault and normal samples can be well linearly separable, and when the number of layers increases, the feature extraction results are more obvious. For the semiconductor etching process, when $LN = 7$ layers, the fault and normal samples are fully linearly separable. Therefore, it is reasonable for the SDeNPE network that $LN = 7$ layers is chosen in this paper.

We select $TCP + 10$, $RF-12$, $BCl_3 + 5$, $Pr-2$, Cl_2-5 , and He chuck pressure fault batches to introduce faults in all periods of each batch. Table 6 shows the FDRs of different fault detection algorithms; the FDR with the best performance for each fault has been marked in bold.

Table 7 shows the FARs of different fault detection algorithms for normal batches of the semiconductor etching process. Better fault detection performance occurs when FAR is lower and in bold.

In Table 6, the mean FDRs for MPCA, NPE, and KNPE methods are lower, which are attributed to the limited number of batches of training data. Although

KNPE is more efficient than NPE, shallow layer feature extraction makes the FDR not very high in general. It was mainly due to the poor accuracy of models trained in limited batches. FDRs based on the SNPE method are higher, which is because of the extraction of difference features between data, making it more effective for limited batches of data. As a deep non-linear NPE extraction method, DeNPE can more fully extract the features and therefore the FDR is high. Our proposed SDeNPE network has a higher FDR by extracting depth features based on DeNPE and by using the Siamese structure to obtain the difference features.

To further analyze the fault detection performance of the SDeNPE method in this paper, we illustrate the advantages of the SDeNPE method in this paper by means of the fault monitoring charts in Figure 14.

In Figure 14A, the fault detection performance of the MPCA is poor and only a few points are detected. In Figure 14B and Figure 14C, only a small number of faults are detected in the T^2 and SPE statistics; compared to NPE, KNPE is a nonlinear deduction of NPE that extracts the nonlinear features of the data and the FDR increases. For SNPE method, in Figure 14D, the T^2 and SPE statistics are effective for fault detection, with a high number of missed fault samples, the reason being the inadequate

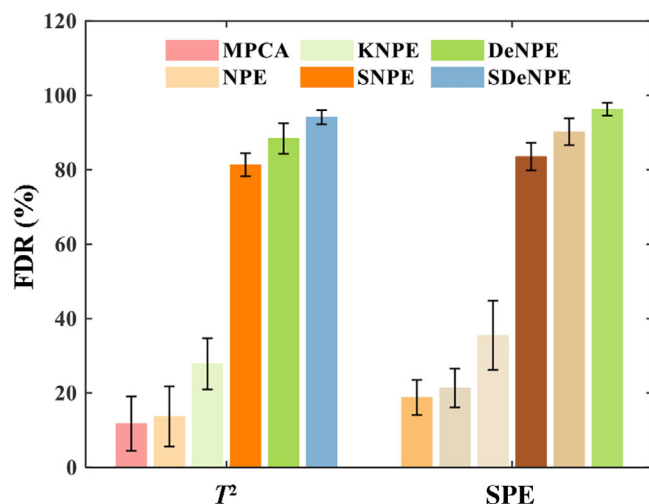


FIGURE 15 Histogram for mean fault detection rate (FDR) of the semiconductor etching process. DeNPE, deep neighbourhood preserving embedding network; KNPE, kernel NPE; MPCA, multiway principal component analysis; NPE, neighbourhood preserving embedding; SDeNPE, Siamese deep neighbourhood preserving embedding network; SNPE, Siamese neighbourhood preserving embedding; SPE, squared prediction error.

extraction of incipient fault features. DeNPE is a deep nonlinear feature extraction method with better feature extraction ability, and in Figure 14E, most of the fault samples are detected in T^2 and SPE statistics. In Figure 14F, the SDeNPE method proposed in this paper performs better for fault detection, that is, it has a higher FDR and lower fault false alarm rate than other methods.

Based on the FDR results, we plot the histograms for mean FDR in Figure 15, where the error bar denotes the variance. It can be seen from Figure 15 that the mean FDR of the method proposed in this paper is higher and is more stable for all fault batches of the semiconductor etching process. In brief, the fault detection performance of SDeNPE is better than other methods and the proposed method can better monitor the process.

5 | CONCLUSIONS

In this paper, a new statistical modelling method based on deep feature extraction, called SDeNPE, is proposed and applied to batch process fault detection. In the SDeNPE network, the alternating use of different kernel functions within the kernel mapping layer allows for a diversity of extracted nonlinear features; the Siamese deep model makes it more sensitive to incipient faults by extracting differential features between data; in particular, when this twinned deep model is used to extract

features, the number of batches of training data required is reduced while ensuring fault detection effectiveness. Two industrial examples are used, including a penicillin fermentation process and a semiconductor etching process. The results show that the SDeNPE network is superior in FDR and FAR when it applied to batch process monitoring and is more effective for process fault detection. SDeNPE is more prompt in detecting early faults and avoids further expansion of early micro faults. An efficient fault detection method can improve the accuracy and reliability of fault diagnosis. How to accomplish fault identification in an SDeNPE-based process monitoring will be a focus of our further research.

AUTHOR CONTRIBUTIONS

Kai Liu: Methodology; software; writing – original draft. **Xiaoqiang Zhao:** Conceptualization; funding acquisition; supervision; project administration; writing – review and editing. **Miao Mou:** Validation; methodology; software. **Yongyong Hui:** Investigation; validation; formal analysis.

ACKNOWLEDGEMENTS

This research work has been supported by the National Natural Science Foundation of China (Grant no. 62263021), the Science and Technology Project of Gansu Province (Grant no. 21JR7RA206, Grant no. 21YF5GA072), the Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (Grant no. 2022KX07), and the Gansu Provincial Education Department Postgraduate Innovation Star Project (Grant no. 2023CXZX-467, Grant no. 2023CXZX-474).

PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/cjce.25102>.

DATA AVAILABILITY STATEMENT

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

ORCID

Kai Liu  <https://orcid.org/0000-0002-8624-2208>

Miao Mou  <https://orcid.org/0000-0002-3846-4121>

Yongyong Hui  <https://orcid.org/0000-0001-5014-887X>

REFERENCES

- [1] P. Agarwal, M. Aghaee, M. Tamer, H. Budman, *Comput. Chem. Eng.* **2022**, 159, 107694.
- [2] C. Peng, Z. RuiYang, D. ChunHao, *Expert Systems with Applications* **2022**, 202, 117390.

- [3] S. Zhang, X. Bao, *J. Process Control* **2022**, 116, 185.
- [4] X. Zhao, K. Liu, Y. Hui, *Trans. Inst. Meas. Control* **2023**, 45, 89.
- [5] X. Ma, Y. Si, Y. Qin, Y. Wang, *IEEE Transactions on Automation Science and Engineering* **2022**, 20, 310.
- [6] Q. Jiang, F. Gao, H. Yi, X. Yan, *IEEE Transactions on Control Systems Technology* **2018**, 27, 1368.
- [7] Y. Wang, Y. Si, B. Huang, Z. Lou, *Can. J. Chem. Eng.* **2018**, 96, 2073.
- [8] Z. Lou, Y. Wang, Y. Si, S. Lu, *Automatica* **2022**, 138, 110148.
- [9] T. Kurita, *Computer Vision: A Reference Guide*, Springer, New York, NY **2019**.
- [10] Z. Ge, Z. Song, *Ind. Eng. Chem. Res.* **2007**, 46, 2054.
- [11] S. Yin, X. Zhu, O. Kaynak, *IEEE Transactions on Industrial Electronics* **2014**, 62, 1651.
- [12] Y. Hui, X. Zhao, *J. Process Control* **2018**, 69, 44.
- [13] A. Miao, Z. Ge, Z. Song, L. Zhou, *Ind. Eng. Chem. Res.* **2013**, 52, 13717.
- [14] X. Q. Zhao, T. Wang, *Chemom. Intell. Lab. Syst.* **2017**, 162, 94.
- [15] B. Song, S. Tan, H. Shi, *Control Engineering Practice* **2016**, 50, 48.
- [16] Y.-N. Sun, W. Qin, H.-W. Xu, R.-Z. Tan, Z.-L. Zhang, W.-T. Shi, *Inf. Sci.* **2022**, 608, 81.
- [17] H. Chen, J. Wu, B. Jiang, W. Chen, *ISA Trans.* **2020**, 104, 175.
- [18] L. Tan, P. Li, A. Miao, Y. Chen, *Meas. Control* **2019**, 52, 387.
- [19] Q. Zhu, N. Zhang, Y. Xu, Y. He, presented at 2021 IEEE 10th Data Driven Control and Learning Systems Conf. (DDCLS), Suzhou, China, May 2021.
- [20] Y. Fan, W. Du, Y. Zhang, X. Wang, *Math. Probl. Eng.* **2015**, 2015.
- [21] A. Miao, Z. Song, Q. Wen, Z. Ge, *IFAC Proceedings Volumes* **2012**, 45, 148.
- [22] A. M. Miao, P. Li, L. J. Ye, *Chemom. Intell. Lab. Syst.* **2015**, 147, 86.
- [23] H. Liu, E. Li, X. Liu, K. Su, S. Zhang, *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2021**, 15, 1.
- [24] M. Mou, X. Zhao, *J. Taiwan Inst. Chem. Eng.* **2022**, 132, 104115.
- [25] S. Tang, Y. Zhu, S. Yuan, *ISA Trans.* **2022**, 129, 555.
- [26] I. Lomov, M. Lyubimov, I. Makarov, L. E. Zhukov, *Journal of Industrial Information Integration* **2021**, 23, 100216.
- [27] L. Zhang, Z. Song, Q. Zhang, Z. Peng, *Neural Computing and Applications* **2022**, 34, 8575.
- [28] H. Zhao, Z. Lai, *Neural Networks* **2019**, 109, 6.
- [29] C. Liu, K. Wang, L. Ye, Y. Wang, X. Yuan, *Inf. Sci.* **2021**, 567, 42.
- [30] S. Jadon, A. A. Srinivasan, *Data Management, Analytics and Innovation*, Springer, Singapore **2021**, p. 353.
- [31] J. Duque Domingo, R. Medina Aparicio, L. M. González Rodrigo, *Appl. Sci.* **2021**, 11, 7839.
- [32] H. Takimoto, J. Seki, S. F. Situju, A. Kanagawa, *Applied Artificial Intelligence* **2022**, 36, 2094885.
- [33] F. Qin, D. H. Wu, *Nonlinear Dyn.* **2021**, 704, 2497.
- [34] Y. Yang, H. Wang, Z. Liu, Z. Yang, presented at 2020 11th International Conf. on Prognostics and System Health Management (PHM-2020 Jinan), Jinan, China, October 2020.
- [35] X. He, D. Cai, S. Yan, H.-J. Zhang, presented at Tenth IEEE International Conf. on Computer Vision (ICCV'05) Volume 1, Beijing, China, October 2005.
- [36] R. Chiplunkar, B. Huang, *IEEE Transactions on Industrial Electronics* **2020**, 68, 8953.
- [37] E. Jordaan, V. Van den Berg, F. Van Rooyen, C. Walsh, *South African Journal of Clinical Nutrition* **2020**, 33, 72.
- [38] K. E. S. Pilario, Y. Cao, M. Shafiee, *Comput. Chem. Eng.* **2019**, 123, 123.
- [39] Z. Hu, J. Peng, H. Zhao, *Int. J. Mach. Learn. Cybern.* **2021**, 12, 1501.
- [40] T.-T. Wong, P.-Y. Yeh, *IEEE Transactions on Knowledge and Data Engineering* **2019**, 32, 1586.
- [41] A. Bakdi, W. Bounoua, A. Guichi, S. Mekhilef, *International Journal of Electrical Power & Energy Systems* **2021**, 125, 106457.
- [42] G. Birol, C. Undey, S. J. Parulekar, A. Cinar, *Biotechnol. Bioeng.* **2002**, 77, 77.
- [43] N. B. Gallagher, B. M. Wise, S. W. Butler, D. D. White Jr., G. G. Barna, *IFAC Proceedings Volumes* **1997**, 30, 79.

How to cite this article: K. Liu, X. Zhao, M. Mou, Y. Hui, *Can. J. Chem. Eng.* **2024**, 102(3), 1167. <https://doi.org/10.1002/cjce.25102>

Kai Liu^{1,2}Xiaoqiang Zhao^{1,2,3,*}Yongyong Hui^{1,2,3}Hongmei Jiang^{1,2,3}

Nonlinear Quality-Related Fault Detection Using Neighborhood Embedding Neural Orthogonal Mapping Algorithm for Batch Process

Quality-related fault detection has become a hot research topic in recent years. It is not reliable to measure quality-related relationships only by mutual information among process variables and single-quality variables. Frequent alarms for quality-unrelated faults seriously affect the normal operation of industrial production. At the same time, the strong nonlinearity of the process data leads to the difficulty of feature extraction. In this paper, we propose a fault detection algorithm based on nonlinear quality-related neighborhood embedding neural orthogonal mapping (QR-NENOM). First, quality-related and quality-unrelated variables are selected by Bayesian fusion mutual information, and the weighted method of mutual information is used to enhance the quality-related information and suppress the quality-unrelated information. Second, local manifold information is obtained by reconstructing nearest neighbors of process data, and key features are extracted by the nonlinear method composed of neural network and orthogonal mapping. Then, statistical indicators are established to complete fault detection. Finally, the nonlinear feature extraction ability of NENOM is verified by numerical examples, and the QR-NENOM algorithm proposed in this paper is applied to the penicillin fermentation process. Comparative experiments show that QR-NENOM has better detection performance for quality-related faults and fewer alarms for quality-unrelated faults.

Keywords: Batch process, Fault detection, Neighborhood preserving embedding, Neural network, Quality related

Received: December 04, 2022; *revised:* January 27, 2024; *accepted:* January 30, 2024

DOI: 10.1002/ceat.202200577

1 Introduction

In the context of Industry 4.0, the manufacturing industry has achieved rapid development, and product quality is the most important production indicator in the manufacturing process [1–4]. According to whether the faults affect the quality of the products, the faults are divided into quality-related faults and quality-unrelated faults. In recent years, quality-related fault have gained a lot of attention in the process monitoring field. By reducing quality-unrelated fault alarm rate, it can reduce unnecessary factory downtime, resulting in greater economic benefits for the enterprise [5, 6].

As an important data-driven branch, multivariate statistical process monitoring (MSPM) methods [7, 8] analyze the entire process through the correlation between process variables, without accurate analytical models and specific structural basis. Moreover, such methods are easy to implement. Therefore, such methods have gradually become a research hotspot for process monitoring. Traditional MSPM methods include principal component analysis (PCA) [9], independent component analysis (ICA) [10], linear discriminant analysis [11], and so on. Considering the importance of product quality for the production process, and making the whole process of monitoring be more

accurate, scholars have proposed some quality-related process monitoring methods, such as partial least squares (PLS) [12], canonical correlation analysis [13], principal component regression [14], etc. However, these methods are all linear processing methods, which have poor ability to extract nonlinear features. As a result, the effectiveness and accuracy of the monitoring results are greatly reduced.

¹Kai Liu  <https://orcid.org/0000-0002-8624-2208>, Xiaoqiang Zhao  <https://orcid.org/0000-0001-5687-942X> (xqzhao@lut.edu.cn), Yongyong Hui  <https://orcid.org/0000-0001-5014-887X>, Hongmei Jiang

College of Electrical and Information Engineering, Lanzhou University of Technology, Qilihe Street, Lanzhou 730050, China.

²Kai Liu, Xiaoqiang Zhao, Yongyong Hui, Hongmei Jiang
Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou University of Technology, Qilihe Street, Lanzhou 730050, China.

³Xiaoqiang Zhao, Yongyong Hui, Hongmei Jiang
National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Qilihe Street, Lanzhou 730050, China.

To deal with nonlinear problems, most methods adopt kernel functions to projection nonlinear process data into a high-dimensional kernel space. Fan et al. [15] proposed a nonlinear process monitoring method of kernel independent component analysis (KICA)-PCA, which uses genetic algorithm to determine the kernel parameters by minimizing the false alarm rate and maximizing the detection rate to improve the monitoring performance. He et al. [16] proposed a new method of kernel locality preserving projection, the kernel trick is applied to the construction of nonlinear kernel models for batch monitoring. Liu et al. [17] proposed a kernel ICA for processing nonlinearities in metallurgical processes. Further, consider quality-related fault detection. Jiao et al. [18] proposed a nonlinear quality-related fault detection method based on the kernel partial least squares (KPLS) model. The method uses KPLS to realize the linear relationship between the kernel matrix and the output matrix and solves the nonlinear problem between the process variables. Wei et al. [19] considered the relationship between process variables and quality variables of using kernel functions to deal with nonlinearity and proposed a supervised self-optimizing kernel model for industrial processes. Ma et al. [20] proposed a new dynamic mixed kernel principal component analysis model based on the mixed kernel function, which revealed and quantified the linear and nonlinear dynamic interdependencies between process variables and quality variables. However, the kernel function method cannot maintain the characteristics of the original data. In addition, in the training stage of the model, the kernel matrix constructed based on a large amount of data is used for feature decomposition, which leads to an increase in the amount of calculation and affects the detection efficiency.

Recently, artificial neural networks (ANN) have more advantages in nonlinear data feature extraction. Yu et al. [21] proposed a broad convolutional neural network with incremental learning capability, which combined multiple consecutive samples into a data matrix, and then extracted fault trends and nonlinear structures from the obtained data matrix through a convolution operation. Wu et al. [22] proposed a fault diagnosis method based on deep convolutional neural network for nonlinear chemical process. Zhang et al. [23] proposed a bidirectional recurrent neural network (RNN) (BiRNN) fault diagnosis method, which proved its effectiveness in chemical processes. Xu et al. [24] proposed an online fault diagnosis method based on the deep transfer convolutional neural network framework, which was effective for solving data nonlinearity. The above process monitoring methods based on ANN do not consider the influence of monitoring on product quality. In addition, most of the neural network-based methods are global end-to-end models, which do not capture the local data structure. In the process of feature compression, important local neighbor information is lost, which is bad for monitoring.

On the other hand, neighborhood preserving embedding (NPE) [25] algorithm not only has good performance in extracting the manifold structure of the data, but also the ability to preserve the original local neighborhood information of the data. So, it has attracted much attention. Liu et al. [26] proposed a stacked neighborhood preserving autoencoder to extract neighborhood preserving features. Deep learning extracts features while preserving local neighbor structure. Yu et al. [27] proposed a deep neural network manifold regularization stacking

autoencoder for fault detection in complex industrial processes. Rezaei et al. [28] proposed a regularized multilabel learning method based on two-manifold learning (RMLDM) for industrial processes. Such methods integrate the NPE into the neural network to monitor the changes of the local structure of the process data and realize the fault detection by monitoring the local information changes. However, quality-related issues are not considered, so the relationship between process variables and quality variables should be further considered.

As meaningful supplements, strongly inspired by the above observations, we propose a fault detection method based on nonlinear quality-related neighborhood embedding neural orthogonal mapping (QR-NENOM). This method fully considers the nonlinear characteristic of batch process and establishes the supervision relationship between process variables and quality variables through the weighting idea of mutual information (MI) and Bayesian inference. Simultaneously, it preserves local information. In addition, the fault detection indicators are rationally designed to improve the performance of fault detection. The method is used for fault detection of penicillin fermentation process, and its validity is verified.

In this paper, as a new nonlinear quality-related fault detection framework, QR-NENOM can meet the practical needs of batch process monitoring. The main contributions lie in the following aspects.

- (1) Our proposed fault detection method pays more attention to process variables that affect product quality, which can reduce production line shutdowns and equipment life losses due to quality-unrelated variable faults.
- (2) Under the joint action of neighborhood embedding and neural network, the extracted features not only deal with the strong nonlinearity of the data, but also better extract the local structural information.
- (3) Neighborhood embedding neural network under orthogonal constraints can eliminate feature redundancy and reduce computational complexity.

The remainder of this paper is structured as follows. Sect. 2 explains the fundamentals of MI and NPE. Sect. 3 offers the principles and optimization problems of NENOM algorithm. Sect. 4 is focused on the NENOM fault detection framework for quality-related and quality-unrelated faults. Simulation example as well as penicillin fermentation process are explained in Section 5 to check the performance of the proposed approach. This work ends with conclusions in the last part.

2 Preliminaries

In this section, the basic ideas of MI and NPE are reviewed briefly. Then, the problem formulation is given.

2.1 Mutual Information (MI)

MI [29] is an information measure in the field of information theory, which measures the amount of information that one random variable contains in another random variable. It can not only

represent the linear relationship between two random variables, but also be used to evaluate the nonlinear relationship. If the two variables are unrelated of each other, the MI value is 0. The MI can be used to measure the common information contained between two random variables, so the degree of correlation between the two random variables can be expressed by MI. The calculation of MI is as formula (1):

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (1)$$

where, $(X, Y) \sim p(x, y)$, $I(X, Y)$ is the MI between X and Y .

For a single random variable x , the Shannon entropy is expressed as formula (2).

$$H(x) = - \int_x p(x) \log p(x) dx \quad (2)$$

where, $p(x)$ is probability density of x .

The joint entropy is shown in formula (3).

$$H(x_1, x_2) = - \int_{x_1} \int_{x_2} p(x_1, x_2) \log p(x_1, x_2) dx_1 dx_2 \quad (3)$$

Therefore, the MI for two variables is obtained by formula (4).

$$I(x_1, x_2) = \int_{x_1} \int_{x_2} p(x_1, x_2) \log \left(\frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right) dx_1 dx_2 \quad (4)$$

It can also be expressed as formula (5).

$$I(x_1, x_2) = H(x_1) + H(x_2) - H(x_1, x_2) \quad (5)$$

For the same process variable x_1 , the MI obtained from different quality variables y_j is usually different. For the process monitoring model, it is unreasonable to determine the overall quality information contained according to the MI between process variables and a single-quality variable. Therefore, we adopt the idea of weighting in Bayesian inference and integrate the MI obtained by the same x_i and each y_j to obtain a weighted fusion MI.

2.2 Neighborhood Preserving Embedding (NPE)

NPE [25] is a manifold learning algorithm that can obtain the local manifold structure of the data. The main idea is to achieve dimensionality reduction by extracting local linear information of the manifold. Specifically, NPE uses a local linear reconstruction to represent the local linear structure of the manifold, which is in the form of mean square error. The steps are as follows.

For the original data $\mathbf{X}(x_1, x_2, \dots, x_n) \in R^D$, set the number of samples to m , the dimension to n , the dimension to d after dimensionality reduction, and $Q(i)$ to the set of k nearest-neighbor samples of sample i . Then the objective function representing the reconstruction error in the high-dimensional space is shown in formula (6).

$$\Phi(\mathbf{W}) = \min \sum_{i=1}^m \left\| \mathbf{x}_i - \sum_{j \in Q(i)} \mathbf{W}_{ij} \mathbf{x}_j \right\|^2 \quad (6)$$

where \mathbf{W} should satisfy the normalization constraint.

The matrix \mathbf{W} obtained by optimizing the objective function contains the local information of the manifold. In the process of

dimensionality reduction, NPE retains the same local linear reconstruction as the original space in the low-dimensional space (that is, uses the same weight matrix \mathbf{W} in the objective function) to achieve linear dimensionality reduction. The objective function in low-dimensional space is shown in formula (7).

$$\Phi(\mathbf{y}) = \sum_{i=1}^m \left(\mathbf{y}_i - \sum_{j=1}^n \mathbf{W}_{ij} \mathbf{y}_j \right)^2 \quad (7)$$

where $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ is the linear reconstruction of \mathbf{X} in low-dimensional space. \mathbf{X} and \mathbf{Y} satisfy $\mathbf{Y} = \mathbf{A}^T \mathbf{X}$, $\mathbf{A}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$ is the reduced projection matrix.

The dimensionality-reduced linear map is computed by solving for the generalized eigenvectors, as in formula (8).

$$\mathbf{X} \mathbf{M} \mathbf{X}^T \mathbf{a} = \lambda \mathbf{X} \mathbf{X}^T \mathbf{a} \quad (8)$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$, $\mathbf{I} = \text{diag}(1, \dots, 1)$. Arrange the eigenvectors \mathbf{a} in ascending order of eigenvalues to obtain the matrix \mathbf{A} . The NPE algorithm has been widely used for process monitoring due to the local manifold structure that can obtain the data. However, NPE is a linear feature extraction algorithm, and it is difficult to extract nonlinear features. Therefore, we combine neural network and NPE to extract nonlinear features while retaining local neighbor information.

3 Neighborhood Embedding Neural Orthogonal Mapping (NENOM)

This section focuses on the derivations of the proposed NENOM. It has been proved that the optimization procedures of NENOM will converge to a local optimum.

3.1 The Theoretical Derivations of Neighborhood Embedding Neural Orthogonal Mapping

This subsection proposes a nonlinear feature extraction algorithm called neighborhood NENOM, as shown in Fig. 1. First, the original data preserves the local manifold structure information through nearest-neighbor selection and weight reconstruction; second, a nonlinear neural network is used as a nonlinear feature extractor; finally, the features are mapped back to the original data space using a linear orthogonal transformation.

Due to the objective function formula (6) of NPE algorithm, this paper designs the optimization problem of NENOM as shown in formula (9).

$$[\mathbf{W}^*, \mathbf{b}^*, \mathbf{V}^*] = \arg \min_{\mathbf{W}, \mathbf{b}, \mathbf{V}} \left\| \mathbf{X} - \mathbf{W} f(\mathbf{X}; \mathbf{W}, \mathbf{b}) \mathbf{V}^T \right\|^2 \quad (9)$$

where subject to $\mathbf{V}^T \mathbf{V} = \mathbf{I}$, \mathbf{W} is neighborhood reconstruction weight matrix, $f(\mathbf{X}; \mathbf{W}, \mathbf{b})$ is a feedforward neural network which consists of one input layer with m inputs, one hidden layer with m units, and one output layer with p nodes. In this paper, $f(\mathbf{X}; \mathbf{W}, \mathbf{b}) = S(\mathbf{X} \mathbf{W} + \mathbf{b})$, where S is sigmoid function. A nonlinear feature map of the data is learnt by adaptively adjusting the parameters \mathbf{W} and \mathbf{b} .

Unlike kernel methods, the dimension of NENOM input does not increase with the increase of training samples. Furthermore,

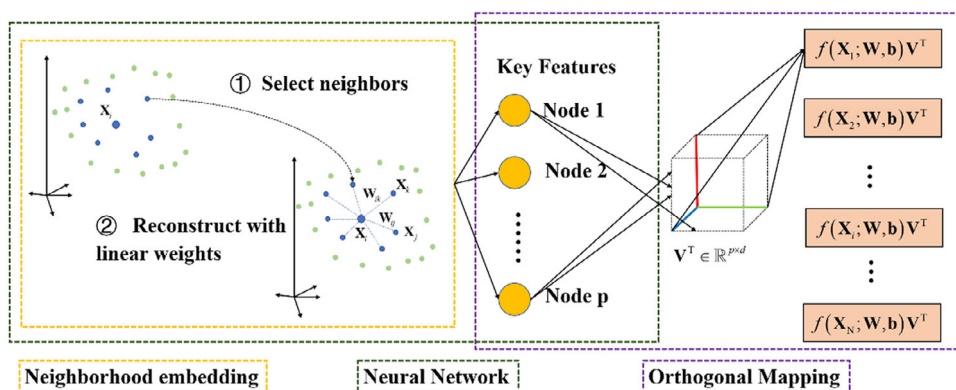


Figure 1. Illustration of neighborhood embedding neural orthogonal mapping (NENOM).

there are no prefixed kernels and related parameters in NENOM, the parameters W and b can be learnt adaptively during the training phase.

We represent the neural network in the form of a matrix; formula (9) is further expressed as formula (10).

$$[W^*, b^*, V^*] = \arg \min_{W, b, V} \|X - WFV^T\|^2 \quad (10)$$

where $F = f(X; W, b)$, matrix F is the key feature that needs to be extracted for further analysis to perform fault detection of the process.

3.2 Optimization Problem of NENOM

For the objective function of NENOM to be a nonconvex function, it is difficult to iteratively calculate the parameters at the same time. Therefore, we first iteratively calculate to determine the parameters W and b and then calculate V . The calculation of V can be optimized by formula (11).

$$V = \arg \min_V \|X - WFV^T\|^2 \quad (11)$$

After obtaining V , the parameters W and b are updated by formula (12).

$$W, b = \arg \min_{W, b} \|X - WFV^T\|^2 \quad (12)$$

The parameters W and b in formula (12) can be adaptively optimized by the backpropagation algorithm that is often used in training the forward neural network. The matrix V in formula (11) is an orthogonal matrix that rotates F to fit the original data X . Orthogonal transformation can be seen as an approximate rotation or reflection that tries to find the optimal transformation of one matrix relative to another.

The original objective function of NENOM in each iteration process can be written as formula (13).

$$J(W^{(j)}, b^{(j)}, V^{(j)}) = \sum_{i=1}^N \|X_i - Wf(X_i; W^{(j)}, b^{(j)}) (V^{(j)})^T\|^2 \quad (13)$$

where $[W^{(j+1)}, b^{(j+1)}, V^{(j+1)}]$ is the optimal solutions of objective function in the j -th iteration. The neural network is trained by

backpropagation and the objective function is further reduced; there are $[W^{(j+1)}, b^{(j+1)}, V^{(j+1)}] \leq [W^{(j)}, b^{(j)}, V^{(j)}]$. Therefore, the optimization procedures of NENOM will converge to a local optimum. After training, the obtained matrix F is used for subsequent feature extraction and fault detection.

4 Fault Detection with Quality-Related NENOM

In this section, the quality-related and quality-unrelated variables are selected by Bayesian fusion MI. We establish statistical indicators and control limits to complete fault detection and give the monitoring procedure.

4.1 Quality-Related and Quality-Unrelated Variables Selections Based on Bayesian Fusion MI

We use MI evaluation for the correlation between process variables and quality variables. The solution of MI can be calculated in the way of Renyi entropy according to formula (5). Unfortunately, it is difficult to be widely used in complex multivariate processes by estimating entropy through accurate probability density function. It is feasible to evaluate a quantity of Renyi entropy using the normalized eigenspectral of the Hermitian matrix of the projected data in the regenerated kernel Hilbert space, thus estimating the entropy directly from data without probability density function estimation.

The MI is shown in formula (14).

$$\begin{aligned} I(x_1, x_2) &= H(x_1) + H(x_2) - H(x_1, x_2) \\ &= H_\alpha(x_1) + H_\alpha(x_2) - H_\alpha(x_1, x_2) \\ &= \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^n \lambda_i(A)^\alpha \right) \\ &\quad + \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^n \lambda_i(B)^\alpha \right) - H_\alpha \left(\frac{A \circ B}{\text{tr}(A \circ B)} \right) \end{aligned} \quad (14)$$

where $\mathbf{A}_{ij} = k(x_i, x_j)$, $\mathbf{B}_{ij} = k(y_i, y_j)$, and $\mathbf{A} \circ \mathbf{B}$ denotes the Hadamard product between the matrices \mathbf{A} and \mathbf{B} . We use the Gaussian kernel to obtain the Gram matrices.

For process variables $\mathbf{X} = [x_1, x_2, \dots, x_m] \in R^{n \times m}$ and quality variables $\mathbf{Y} = [y_1, \dots, y_c] \in R^{n \times c}$, MI is usually different for the same process variable and different quality variables. It is unreasonable to determine how much information process variables contain the overall quality variable only based on the MI of x_i and a certain y_j . Therefore, we use the Bayesian fusion method to weigh the MI of each y_j corresponding x_i to obtain new MI, as shown in formula (15).

$$I(x_i) = \sum_{j=1}^c \frac{p(x_i|Q_j) p(Q_j|x_i)}{\sum_{j=1}^c p(x_i|Q_j)} = \sum_{j=1}^c \frac{p(x_i|Q_j) \frac{p(Q_j)}{p(x_i)}}{\sum_{j=1}^c p(x_i|Q_j)} \quad (15)$$

Further, $p(x_i)$ can be obtained by formula (16).

$$p(x_i) = p(x_i|U_j) p(U_j) + p(x_i|Q_j) p(Q_j) \quad (16)$$

where $p(Q_j)$ and $p(U_j)$ represent the related and unrelated probabilities of x_i and y_j , respectively, and the sum of the probabilities is 1. $p(Q_j)$ is the ratio of the number of variables that the mutual of x_i and y_j is greater than the average MI to the total number of process variables. The calculation of the average value of MI \bar{I}_{xj} is shown in formula (17).

$$\bar{I}_{xj} = \frac{\sum_{i=1}^m I(x_i, y_j)}{m} \quad (17)$$

where m is the number of process variables.

The conditional probabilities $p(x_i|Q_j)$ and $p(x_i|U_j)$ are shown in formulae (18) and (19).

$$p(x_i|Q_j) = \exp\left(-\frac{I_{ij}(x_i, y_j)}{\bar{I}_{xj}}\right) \quad (18)$$

$$p(x_i|U_j) = \exp\left(-\frac{\bar{I}_{xj}}{I_{ij}(x_i, y_j)}\right) \quad (19)$$

After the fusion MI $I(x_i)$ is calculated by the above method, it is sorted in descending order, and the variables corresponding to the top k largest MI $I(x_i)$ are selected to form quality-related variables. The remaining variables are quality-unrelated variables. The number of quality-related process variables can be obtained from formula (20).

$$\frac{\sum_{i=1}^k I(x_i)}{\sum_{i=1}^m I(x_i)} \geq \theta_{ax} \quad (20)$$

where θ_{ax} is set as 0.85.

4.2 Fault Detection Based on Statistical Indicators

The number of downtimes increases the production cost of the enterprise. In order to prevent the shutdown caused by fault false alarm that does not affect the product quality, the quality-unrelated fault alarm should be reduced. We use the

method of weighted data by MI to process the data, to ensure a higher fault detection rate of quality-related fault variables and a lower fault detection rate of quality-unrelated fault variables. The weight is determined according to the ratio of the variable mutual trust value to the maximum MI value, as shown in formula (21).

$$\mathbf{W}(x_i) = \frac{I(x_i)}{\max\{I(x_1), \dots, I(x_i), \dots, I(x_n)\}} \quad (21)$$

For offline training, the training data \mathbf{X} is trained through the of NENOM to obtain the feature F . In the fault detection stage, the detection data x_{new} amplifies the quality-related fault variables according to the weighting method of MI, the suppression weight of the quality-unrelated information is calculated as formula (21), and the weighted detection data is: $x^* = \mathbf{W}(x_i)x_{\text{new}}$.

The detection data feature is extracted by the offline trained model, and after the features are acquired, statistical indicators are established to complete the fault detection.

In this paper, we use the T^2 statistic and the SPE statistic for fault detection, and the control limits for these two statistics are calculated based on the assumption of multivariate normal distribution and temporal independence of the observations. Since the distribution of the extracted features has no prior information, this paper approximates the control limits of the T^2 and SPE statistics by kernel density estimation (KDE).

The T^2 statistic is calculated as formula (22).

$$T^2 = F \Sigma_F^{-1} F^T \quad (22)$$

where Σ_F^{-1} is the covariance matrix associated with the features F .

The SPE statistic is calculated as formula (23).

$$\text{SPE} = \|\mathbf{X} - FV^T\|^2 \quad (23)$$

The method of KDE: given a univariate kernel function is shown in the following formula (24):

$$f(j) = \frac{1}{n} \sum_{i=1}^n K_h(j - j_i) = \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{j - j_i}{\sigma}\right) \quad (24)$$

where j is the sample data, j_i is the observation value, σ is the window width, n is the number of observation values, and K is the kernel function. In this paper, the Gaussian kernel function is selected, and the test level $\alpha = 0.95$ can be obtained by formula (25) and (26) to find the control limits T_α^2 and SPE_α .

$$\begin{aligned} & \int_{-\infty}^{T_\alpha^2} f(T^2) d(T^2) \\ &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{T^2 - T_i^2}{\sigma}\right) d(T^2) \\ &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(T^2 - T_i^2)^2}{2\sigma^2}\right) \right\} d(T^2) \\ &= \alpha \end{aligned} \quad (25)$$

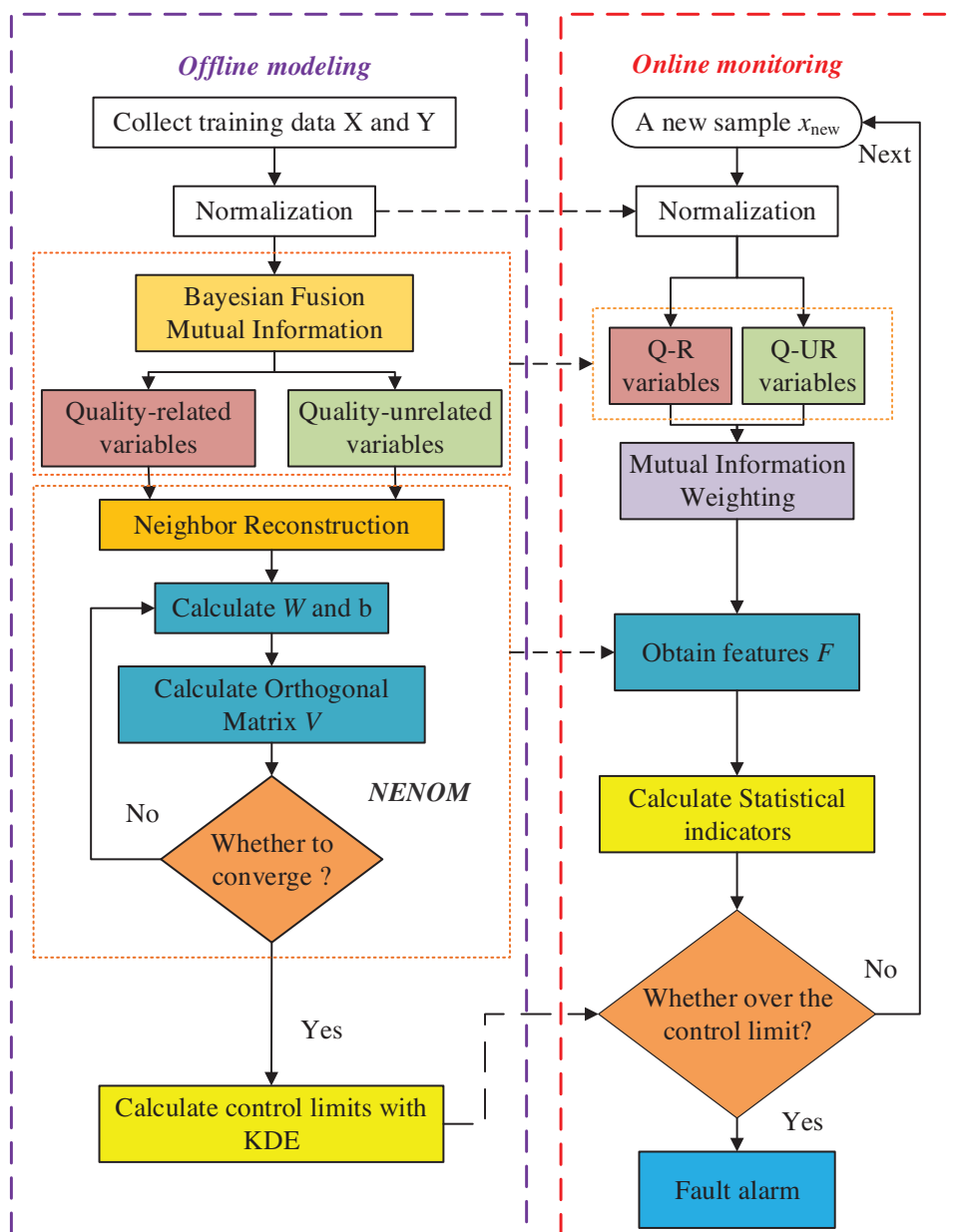


Figure 2. Flowchart of the proposed QR-NENOM algorithm for fault detection.

$$\begin{aligned}
 & \int_{-\infty}^{\text{SPE}_{\alpha}} f(\text{SPE}) d(\text{SPE}) \\
 &= \int_{-\infty}^{\text{SPE}_{\alpha}} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{\text{SPE} - \text{SPE}_i}{\sigma}\right) d(\text{SPE}) \\
 &= \int_{-\infty}^{\text{SPE}_{\alpha}} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(\text{SPE} - \text{SPE}_i)^2}{2\sigma^2}\right) \right\} d(\text{SPE}) \\
 &= \alpha
 \end{aligned} \tag{26}$$

If $T^2 > T_{\alpha}^2$ or $\text{SPE} > \text{SPE}_{\alpha}$, it is fault otherwise normal.

4.3 Monitoring Procedure

The nonlinear quality-related fault detection framework based on QR-NENOM is shown in Fig. 2.

1) Offline modeling

Step 1: Process data X and quality data Y are collected and normalized.

Step 2: The MI between each process variable x_i and the quality variable y_j is obtained by the Bayesian fusion method of formula (15), and the quality-related variables and quality-unrelated variables are selected according to the MI of each variable and the threshold value θ_{ax} by formula (20).

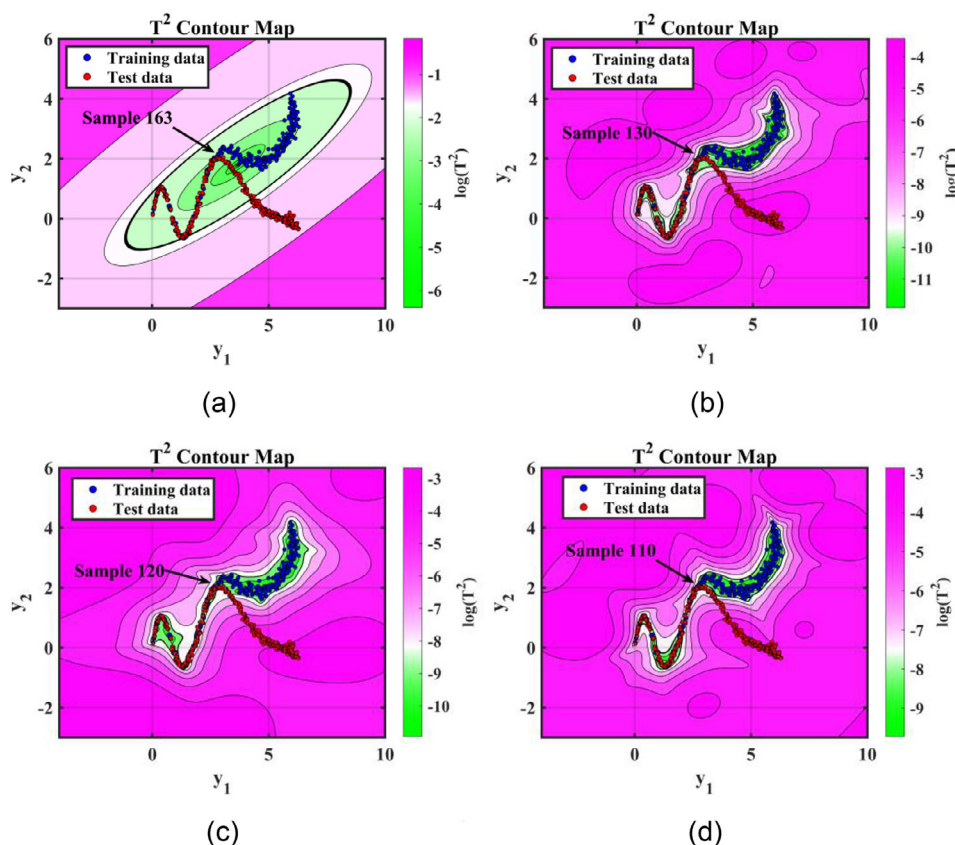


Figure 3. Contour maps of the T^2 statistical index (log scale). (a) NPE; (b) KNPE with RBF kernel; (c) NPE with polynomial kernel; (d) NENOM.

Step 3: According to the objective function of formula (9) to train the NENOM model, the feature F is obtained for online monitoring.

Step 4: The statistical indicators T^2 and SPE of the training data in the offline process are obtained by formula (22) and (23). The control limits T^2_α and SPE_α are obtained through KDE.

2) Online monitoring

Step 1: A new detection sample x_{new} is normalized using the mean and variance during the offline modeling stage.

Step 2: The quality-related variables and quality-unrelated variables are divided in the offline modeling stage; it is used to judge whether the variables that detect sample fault are related to quality.

Step 3: The detection samples are further processed using the MI of the samples to weight, then: $x^* = \mathbf{W}(x_i)x_{\text{new}}$.

Step 4: The statistical indicators of the test sample are calculated and compared with the control limits. If the statistical indicators are more than the control limits, that is, $T^2 > T^2_\alpha$ or $SPE > SPE_\alpha$ indicates that a fault occurs; otherwise, it is normal and next data are detected again.

5 Experiments

In this section, the performance of the proposed algorithm for quality-related faults will be illustrated with numerical example and penicillin fermentation process.

5.1 Numerical Example

A nonlinear example [30] is shown in formula (27). We use this example to verify the nonlinear processing capability of the NENOM algorithm.

$$\begin{aligned} x_1 &= t + e_1 \\ x_2 &= t^2 - 3t + e_2 \\ x_3 &= -t^3 + 3t^2 + e_3 \\ y_1 &= -x_2(1 + x_1) \\ y_2 &= x_3 - \sin(1.5\pi x_2) \end{aligned} \quad (27)$$

where y_1 and y_2 are the only observable variables. x_1, x_2 , and x_3 are the training data at time $t \in [0.01, 2]$, with $e_{1,2,3} \sim \mathcal{N}(0, 0.001)$. Meanwhile, a test dataset is generated during that time period. Among them, a slow linear drift fault occurs in y_2 starting from the 100th sample, and each group of data contains 300 samples. As faults occur, the test samples gradually deviate from the normal path. In this example, the purpose of the detection algorithm is to distinguish faulty data from normal data as early as possible.

As a manifold learning algorithm, NPE could extract local structural information of data and calculate T^2 statistic for each point in the data space $y_1 - y_2$, which represents it as a contour map. Fig. 3 shows the contour maps of the T^2 statistical indicator for the different algorithms, the blue points represent the training data, the red points represent the test data, the green

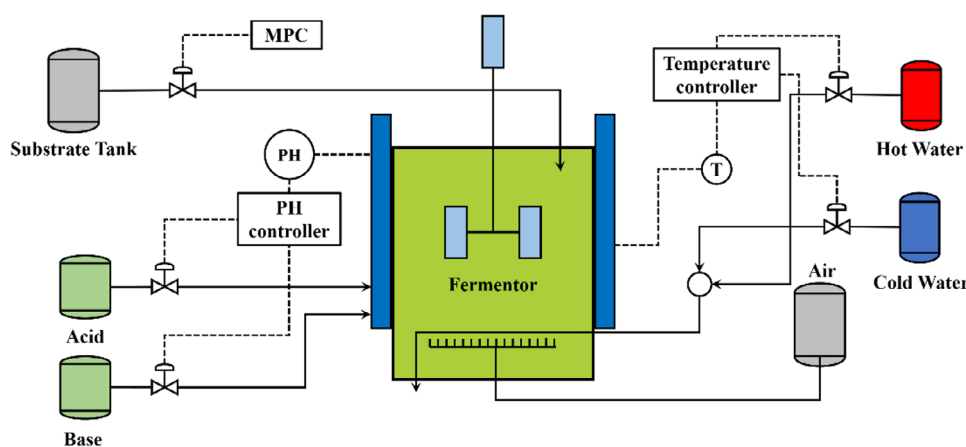


Figure 4. Schematic of the penicillin fermentation process.

part represents the process data in the normal range, the purple part represents the process data in the fault range, and the control limit is described as the envelope around the training data.

We can see from the T^2 contour map in Fig. 3(a) that it is difficult for NPE algorithm to fully express the nonlinear characteristic of the process data, and only at the 163rd sample point, the test data leave the normal area to detect anomalies. In process monitoring, the fault is detected after the delay of 63 samples, so it cannot achieve good results in nonlinear process monitoring. By finding nonlinear projections of the original data, kernel NPE (KNPE) can detect fault sample points faster with better detection performance. KNPE results in tighter bounds around the data under normal circumstances, as the nonlinear behavior of the process is captured more accurately. Fig. 3(b,c) shows the T^2 contour maps of the KNPE with radial basis function (RBF) kernel and polynomial kernel, respectively. It can be seen from Fig. 3(b,c) that KNPE with radial basis kernel function and polynomial kernel function can better describe the nonlinear characteristic of process data than NPE. The normal sample points are in the green area, which fit the path of the normal data. Compared with Fig. 3(b), Fig. 3(c) has better nonlinear feature extraction ability in the green area. In Fig. 3(b), KNPE with radial base kernel detects the fault at the 130th sample point, and in Fig. 3(c), KNPE with polynomial kernel detects the fault at the 120th sample point, which is lower than the NPE detection delay. Although Fig. 3(b,c) both use the kernel function to extract nonlinear features, the detection time is different. The reason is that when NPE effectively extracts the local structure of the data, as the global kernel function, the polynomial kernel function can effectively extract the global features of the data structure, which makes the feature extraction of the data more comprehensive.

Fig. 3(d) shows the T^2 contour map of NENOM algorithm in this paper. It can be seen from Fig. 3(d) that T^2 statistic can describe the process data path of almost the whole process and can effectively deal with the nonlinear characteristic of the process data. The fault is detected at 110 sample points, and the delay is the lowest. The detection performance of the proposed algorithm in this paper is improved significantly.

5.2 Penicillin Fermentation Process

5.2.1 Process Description

The penicillin fermentation process is a typical nonlinear batch process that involves complex biological and chemical changes from the raw materials to the final products. There are many control loops and variables in the fermentation process, and faults often occur, which seriously affect the production performance and product quality. Therefore, it is a hot topic to ensure the high-quality and high-efficiency operation of penicillin through reasonable quality-related fault detection technology.

In 2002, Birol et al. [31] developed the simulation production platform Pensim 2.0 based on the fermentation process of penicillin. It provides a benchmark platform for batch monitoring and fault diagnosis of the penicillin production process. It can simulate the effective fermentation of penicillin and obtain a series of parameters such as process microbial concentration, carbon dioxide, pH value, etc., which solves the problem involving real-time acquisition of variable data. Fig. 4 shows the schematic of the penicillin fermentation process.

In the experiment, we choose ten main variables to describe the penicillin fermentation process. The curve of the variable is shown in Fig. 5, the process variable has strong nonlinear characteristics. We use the pensim2.0 platform to generate 30 batches of normal data and 6 batches of fault data with different fault types. The simulation time was set to 400 h, and the sampling time was set to 1 h. Data under normal operating conditions is used for data training and model building, and data collected from abnormal conditions is used for fault detection. In order to simulate the actual penicillin fermentation process and increase the robustness of the model, Gaussian white noise $e \sim N(0, 0.01)$ is introduced into the data.

According to different fault types, six groups of faults are introduced for fault detection. The selected variables and information about the type of fault are shown in Tab. 1. Among them, the bold values represent abnormal data values of the fault variables.

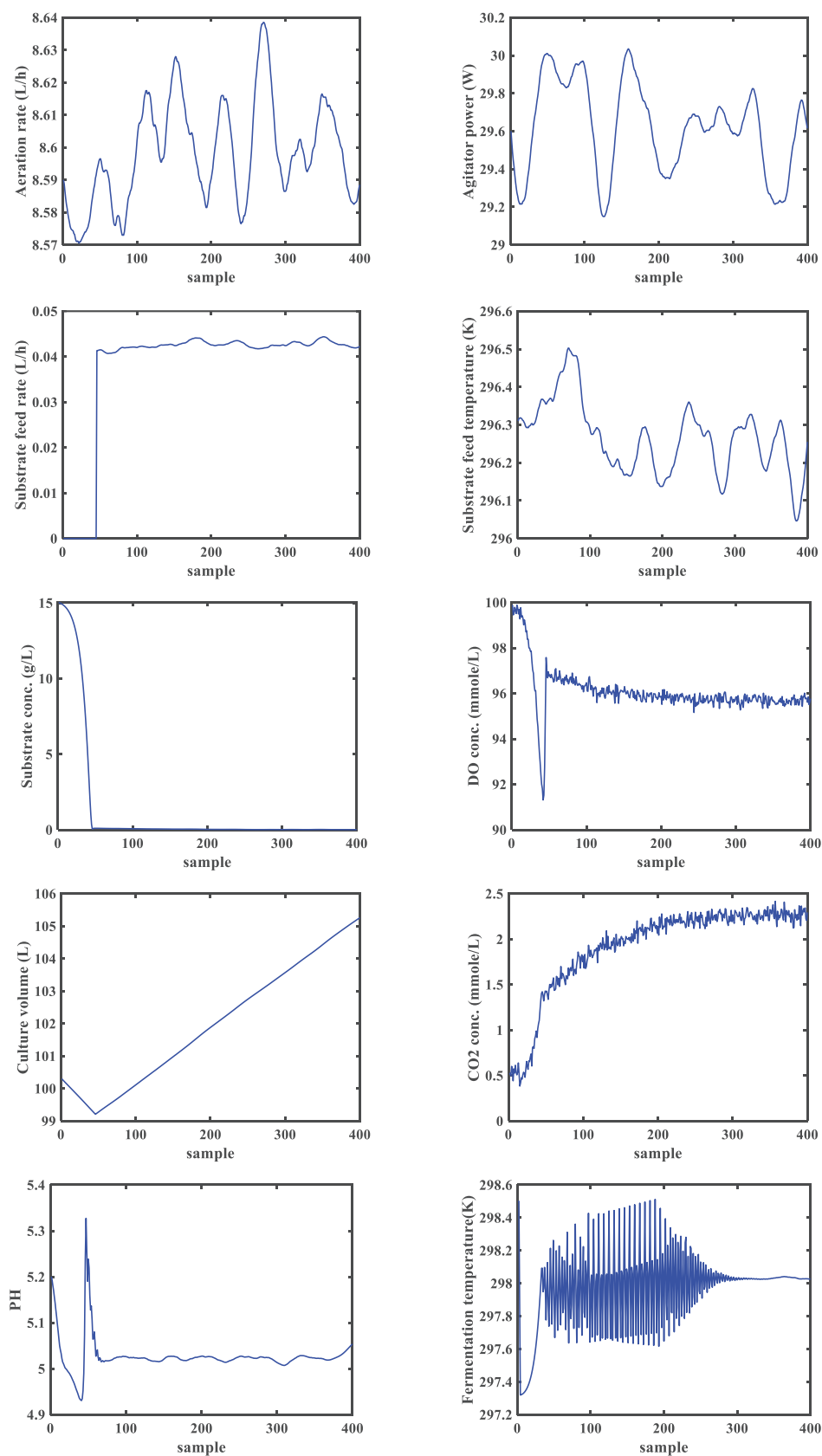
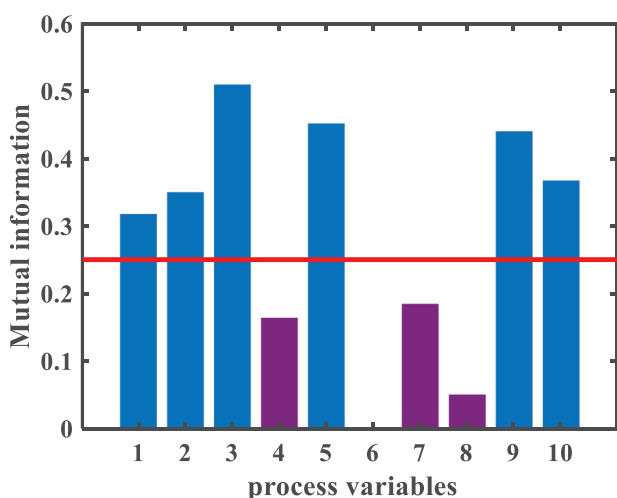


Figure 5. The curve of the process variable under normal condition.

Table 1. Condition settings for normal and fault samples.

Process variables	F1	F2	F3	F4	F5	F6	Normal
1. Aeration rate [L h^{-1}]	8.9	8.6	8.6	8.6	8.6	8.6	8.6
2. Agitator power [W]	30.0	30.8	30.0	30.0	30.0	30.0	30.0
3. Substrate feed rate [L h^{-1}]	0.042	0.042	0.048	0.042	0.042	0.042	0.042
4. Substrate feed temperature [K]	296.0	296.0	296.0	303.0	296.0	296.0	296.0
5. Substrate conc. [g L^{-1}]	16.0	15.0	14.9	15.0	15.0	14.8	15.0
6. DO conc. [mmole L^{-1}]	1.16	1.16	1.16	1.16	1.16	1.16	1.16
7. Culture volume [L]	100.0	102.0	100.0	101.0	100.0	103.0	100.0
8. CO_2 conc. [mmole L^{-1}]	0.6	0.5	0.5	0.6	0.3	1.1	0.5
9. pH	4.8	5.0	5.0	5.0	5.0	5.0	5.0
10. Fermentation temperature [K]	299.0	297.0	298.0	298.0	299.0	298.0	298.0

Values in italics indicate outliers.

**Figure 6.** The mutual information between the quality variables and all process variables.

5.2.2 Detection Results and Discussion

To judge whether the fault variables are quality related or quality unrelated, the MI between each process variable and the quality variable is obtained by Bayesian fusion. Fig. 6 shows the MI between process variables and quality variables.

It can be seen from Fig. 6 that the variables above the threshold are quality related, and the below are quality unrelated, where the red line is the threshold θ_{ax} obtained by formula (20). Among them, variables 1, 2, 3, 5, 9, and 10 are quality related (1. aeration rate, 2. agitator power, 3. substrate feed rate, 5. substrate conc., 9. pH, 10. fermentation temperature), and variables 4, 6, 7, and 8 are quality unrelated (4. substrate feed temperature, 6. DO conc., 7. culture volume, 8. CO_2 conc.).

For process monitoring, fault detection rate (FDR) is computed as the rate of effective alarmed fault data corresponding to the entire fault dataset. The higher the FDR, the stronger the discriminating performance. The calculation of FDR is shown in formula (28).

$$\text{FDR} = \frac{\text{Number of false alarmed data}}{\text{Number of fault data set}} \times 100\% \quad (28)$$

To ensure product quality and reduce the frequent alarms of quality-unrelated faults affecting normal production, there

Table 2. FDRs of the different algorithms for faults/%.

Fault no. ^{a)}	ANN		MNPE		MKNPE		MKPLS		QR-NENOM	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
F1	64.18	69.31	67.13	73.27	81.12	84.67	89.90	87.72	91.12	100.00
F2	72.27	74.25	68.31	74.25	82.17	80.19	89.10	88.12	92.07	97.02
F3	59.40	61.38	62.37	64.35	74.25	76.23	78.21	79.20	87.12	90.09
F4	73.25	89.75	90.75	92.75	94.00	95.25	100.00	100.00	1.50	2.00
F5	17.25	14.75	71.25	80.75	77.25	88.75	79.75	100.00	4.51	3.25
F6	25.17	30.18	69.25	76.18	78.12	89.25	80.12	93.28	4.75	6.00

Values in italics indicate the best value for the results. a) F1 ~ F3 are quality-related faults; F4 ~ F6 are quality-unrelated faults.

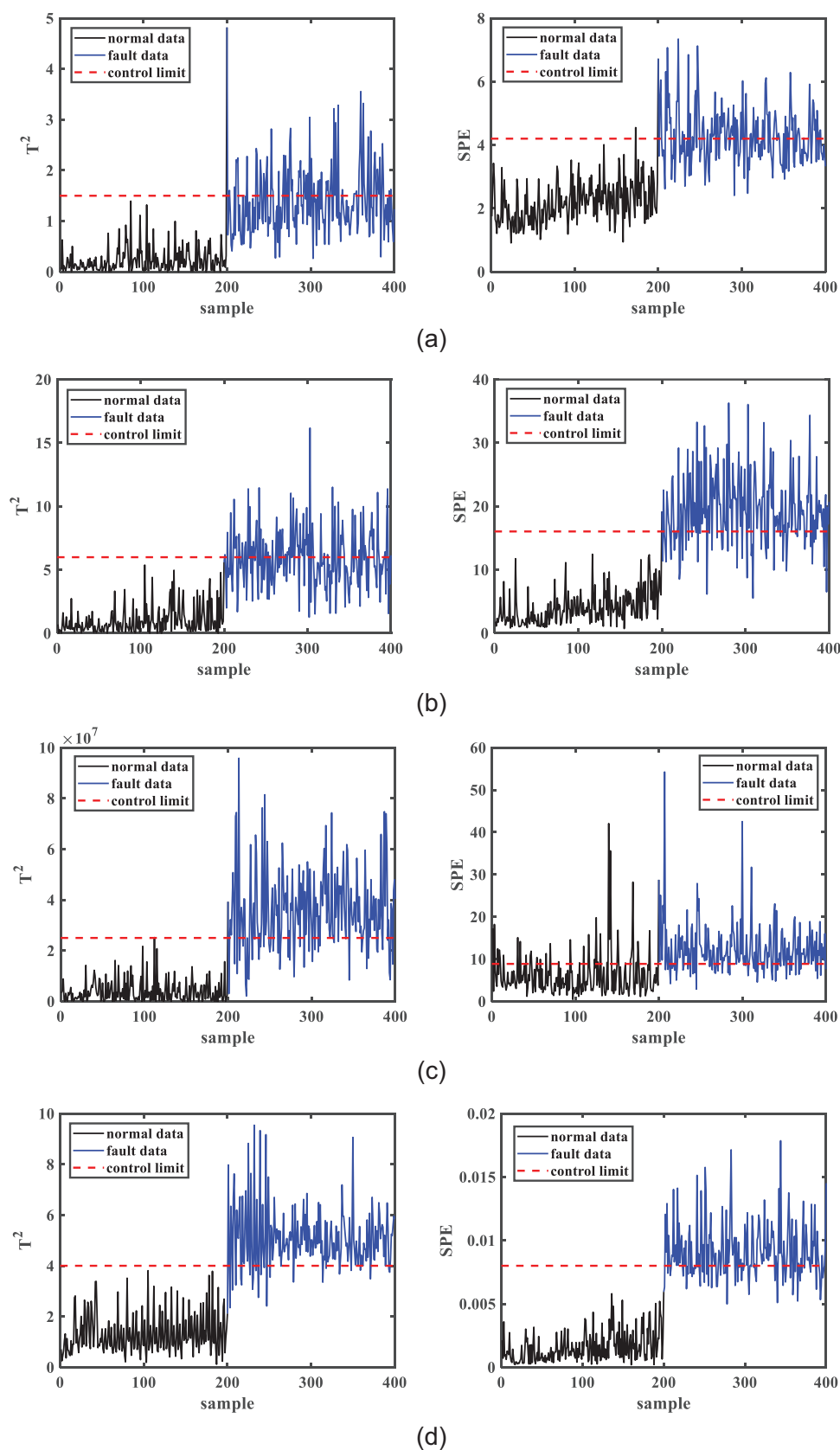


Figure 7. Fault detection performance for different algorithms under F1 condition. (a) ANN; (b) MNPE; (c) MKNPE; (d) MKPLS; (e) QR-NENOM.

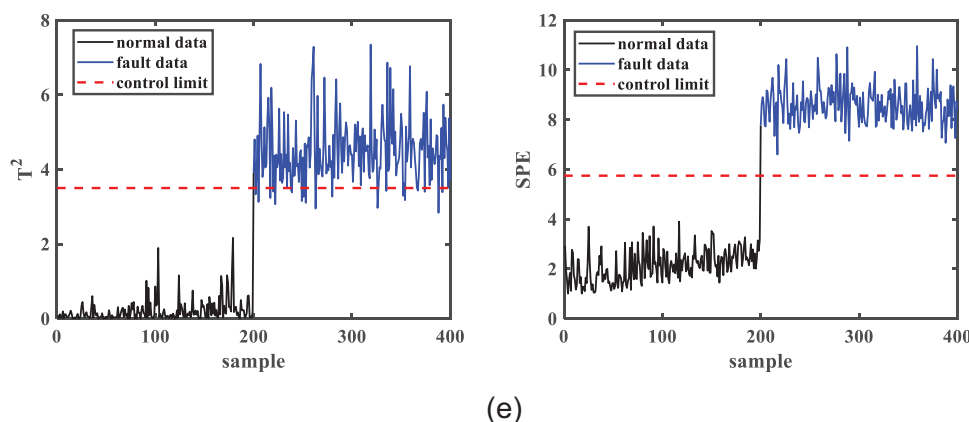


Figure 7. Continued

are higher FDRs for quality-related faults and lower FDRs for quality-unrelated faults. In other words, the higher the fault detection rate of quality-related faults and the lower the fault detection rate of quality-unrelated faults, the better the performance of the fault detection algorithm. Tab. 2 shows FDRs of different fault detection algorithms, and the fault detection rate with the best performance for each fault has been shown in bold.

In Tab. 2, F1 ~ F3 are quality-related faults, and F4 ~ F6 are quality-unrelated faults. For F1 ~ F3 fault detection rates, the higher the better. For quality-related faults, QR-NENOM has a higher fault detection rate than other algorithms. It shows that QR-NENOM can detect more accurately for quality-related fault detection, thereby realizing effective monitoring of quality-related faults during penicillin fermentation. The lower the fault detection rate for F4 ~ F6, the better the algorithm performance. From Tab. 2, we can see that QR-NENOM algorithm has a lower fault detection rate and better fault detection performance. After a quality-unrelated fault occurs, it is compensated by the control loop in the process and does not affect the quality of the final product. A low fault detection rate can greatly reduce the number of unnecessary downtimes, thereby increasing the productivity of the enterprise.

Fig. 7 shows the fault performance for the quality-related fault F1, the black curve is normal data, the blue curve is fault data, and the red dashed line is the fault control limit. Fig. 7(a) shows statistics T^2 and SPE of ANN algorithm, most of the fault data are below the control limit, resulting in serious fault missed alarm. In Fig. 7(b), the fault missed alarm of statistic SPE is less than ANN algorithm, but statistic T^2 is still the obvious fault missed alarm. In Fig. 7(c), compared with multiway neighborhood preserving embedding (MNPE), multiway kernel neighborhood preserving embedding (MKNPE) introduces the kernel method to deal with nonlinearity, and the fault missed alarms of statistics T^2 and SPE are lower, but the statistics SPE of MKNPE algorithm have lots of fault false alarms. In Fig. 7(d), compared with ANN, MNPE, and MKNPE algorithms, statistics T^2 and SPE of multiway kernel partial least square (MKPLS) algorithm significantly have reduced fault missed alarm, but the fault missed alarms are still higher than QR-NENOM algorithm in this paper. In Fig. 7(e), QR-NENOM algorithm in this paper

has the highest fault detection rate, with no fault missed alarm in statistics SPE.

Fig. 8 shows the fault performance for the quality-unrelated fault F5, the blue curve is fault data and the red dashed line is fault control limit. In Fig. 8(a), ANN algorithm has a low fault detection rate for quality-unrelated faults. Combined with Fig. 7(a), the fault detection rate of quality-related fault is also low, so it cannot detect all faults well in the process, so the overall fault detection performance is poor. As shown in Fig. 8(b,c,d), MNPE, MKNPE, and MKPLS algorithms have poor fault detection performance. Fig. 8(e), QR-NENOM algorithm in this paper hardly shows alarm for quality-unrelated faults, and there are only individual points of alarm. Combined with Fig. 7(e), there is a higher fault detection rate for quality-related faults. Therefore, the fault detection performance of QR-NENOM algorithm proposed in this paper is better.

Fig. 9 shows the difference of loading matrix between QR-NENOM with and without orthogonal mapping. Fig. 9(a) shows the result of the nonorthogonal mapping; obviously, with the gradual extraction of nonlinear features, the load matrix contains redundant features. Fig. 9(b) shows the result of the orthogonal mapping, and it is easy to see that the loading matrix contains orthogonal columns.

Fig. 10 shows the histograms of the average fault detection rates for quality-related faults and quality-unrelated faults, respectively. It can be seen intuitively that the fault detection rate is higher for quality-related faults and lower for quality-unrelated faults of QR-NENOM algorithm in this paper, with almost few alarms for quality-unrelated faults. QR-NENOM algorithm has a high fault detection rate of quality-related faults and reduces the shutdown due to quality-unrelated fault alarms. Therefore, it can effectively ensure the safety of industrial production and further reduce unnecessary maintenance.

6 Conclusion

This paper proposed a nonlinear quality-related NENOM fault detection algorithm for batch process monitoring, which can not only have a higher fault detection rate for quality-related faults,

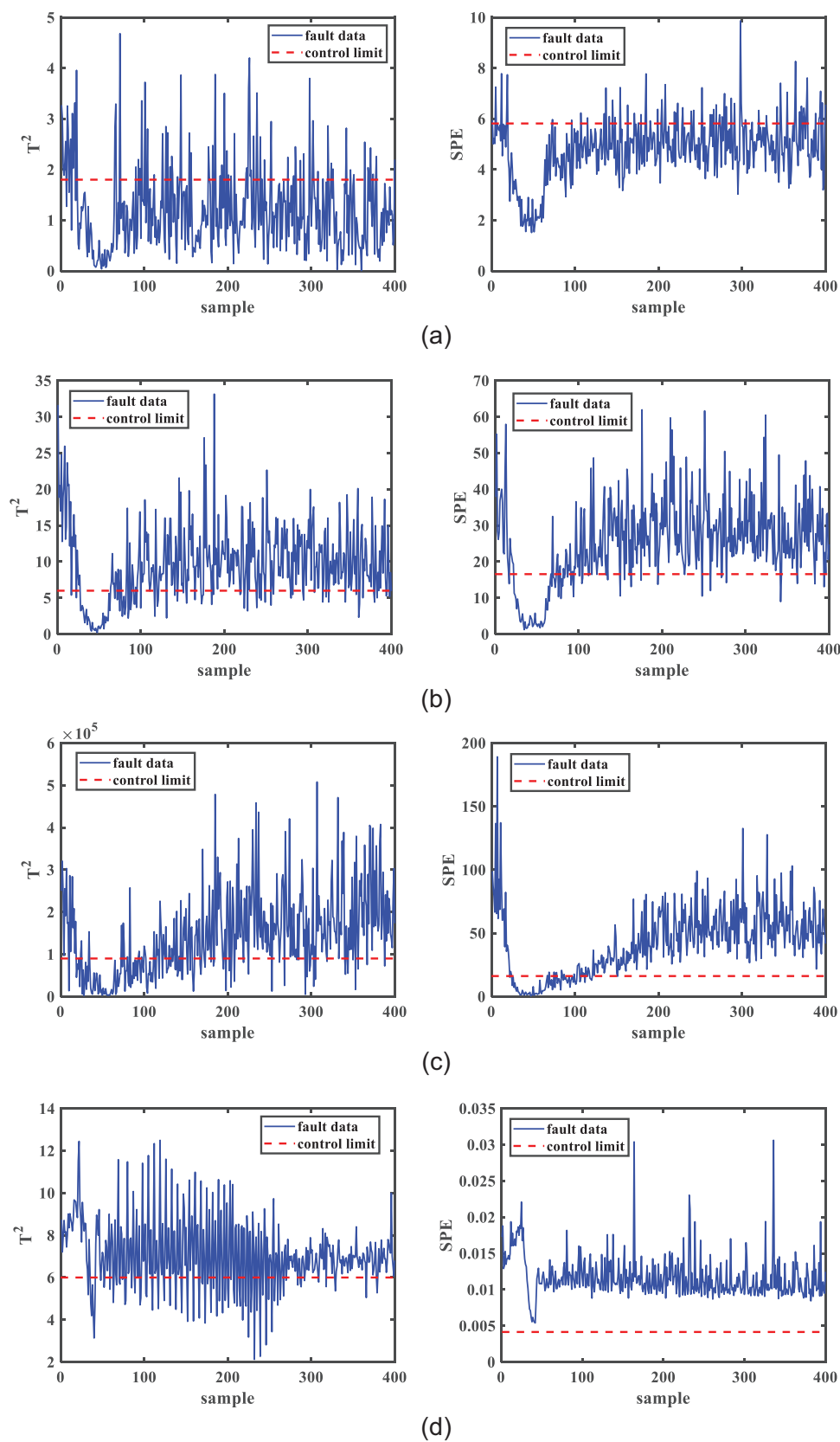


Figure 8. Fault detection performance for different algorithms under F5 condition. (a) ANN; (b) MNPE; (c) MKNPE; (d) MKPLS; (e) QR-NENOM.

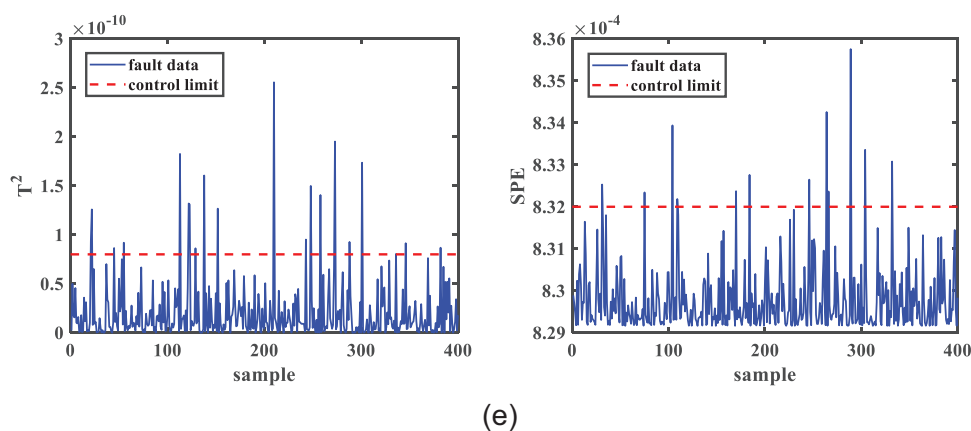


Figure 8. Continued

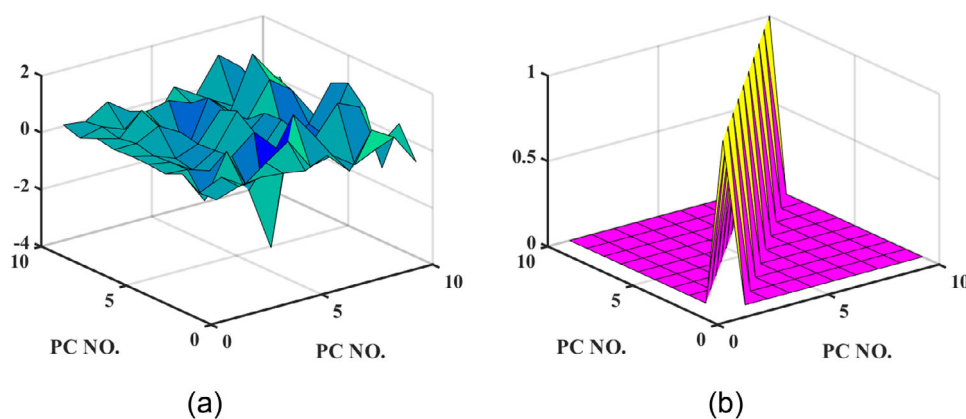


Figure 9. Illustration of orthogonal mapping. (a) nonorthogonal mapping; (b) after orthogonal mapping.

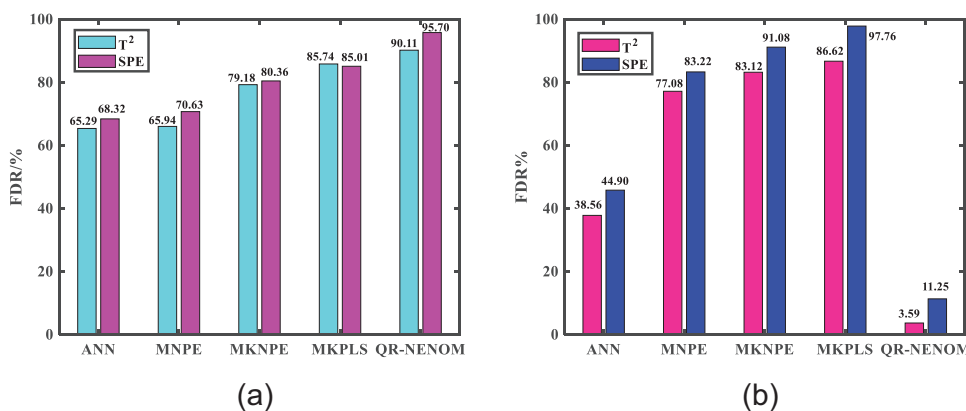


Figure 10. Histogram of the FDR for quality-related faults and quality-unrelated faults. (a) quality-related faults; (b) quality-unrelated faults.

but also have fewer alarms for quality-unrelated faults. At the same time, the local manifold structure information is preserved while extracting the key nonlinear features of the data. The process monitoring is completed by establishing corresponding fault detection indicators. To illustrate the effectiveness and superiority of QR-NENOM algorithm proposed in this paper, a typical batch process-penicillin fermentation process is used to conduct relevant experiments. The verification results show that, compared with several advanced fault detection algorithms of the batch process, ANN, MNPE, MKNPE, and MKPLS, QR-

NENOM achieves better detection results and richer detection information in quality-related fault detection. It is the primary interest in future research that the detected faults are effectively identified and diagnosed.

Acknowledgments

This research work has been awarded by the National Natural Science Foundation of China (62263021), the College Industrial

Support Project of Gansu Province (2023CYZC-24), and the Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (2022KX07).

Abbreviations

ANN	artificial neural network
CCA	canonical correlation analysis
ICA	independent component analysis
KNPE	kernel neighborhood preserving embedding
LDA	linear discriminant analysis
MI	mutual information
MKNPE	multiway kernel neighborhood preserving embedding
MKPLS	multiway kernel partial least squares
MNPE	multiway neighborhood preserving embedding
MSPM	multivariate statistical process monitoring
NENOM	neighborhood embedding neural orthogonal mapping
NPE	neighborhood preserving embedding
PCA	principal component analysis
PCR	principal component regression
PLS	partial least squares
QR-NENOM	quality-related neighborhood embedding neural orthogonal mapping
SPE	square prediction error

Data Availability Statement

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

References

- [1] P. Agarwal, M. Aghaee, M. Tamer, H. Budman, *Comput. Chem. Eng.* **2022**, 159, 107694. DOI: <https://doi.org/10.1016/j.compchemeng.2022.107694>
- [2] M. Mou, X. Zhao, *IEEE Trans. Instrum. Meas.* **2022**, 71, 1–11. DOI: <https://doi.org/10.1109/TIM.2022.3170967>
- [3] S. Zhang, X. Bao, *J. Process Control* **2022**, 116, 185–198. DOI: <https://doi.org/10.1016/j.jprocont.2022.06.007>
- [4] Z. Hu, J. Peng, H. Zhao, *Int. J. Mach. Learn. Cybern.* **2021**, 12 (5), 1501–1516. DOI: <https://doi.org/10.1007/s13042-020-01250-z>
- [5] C. Sun, Y. Yin, H. Kang, H. Ma, *Inf. Sci.* **2022**, 600, 301–322. DOI: <https://doi.org/10.1016/j.ins.2022.03.069>
- [6] C.-Y. Sun, Y.-Z. Yin, H.-B. Kang, H.-J. Ma, *IEEE Trans. Autom. Sci. Eng.* **2022**, 19, 3942–3952. DOI: <https://doi.org/10.1109/TASE.2021.3139766>
- [7] Y. Wang, Y. Si, B. Huang, Z. Lou, *Can. J. Chem. Eng.* **2018**, 96 (10), 2073–2085. DOI: <https://doi.org/10.1002/cjce.23249>
- [8] K. Zhang, H. Hao, Z. Chen, S. X. Ding, K. Peng, *J. Process Control*, **2015**, 33, 112–126. DOI: <https://doi.org/10.1016/j.jprocont.2015.06.007>
- [9] N. Lu, Y. Yao, F. Gao, F. Wang, *AIChE J.* **2005**, 51 (12), 3300–3304. DOI: <https://doi.org/10.1002/aic.10568>
- [10] J.-M. Lee, C. Yoo, I.-B. Lee, *J. Process Control* **2004**, 14 (5), 467–485. DOI: <https://doi.org/10.1016/j.jprocont.2003.09.004>
- [11] A. Tharwat, T. Gaber, A. Ibrahim, A. E. Hassanien, *AI Commun.* **2017**, 30 (2), 169–190. DOI: <https://doi.org/10.3233/AIC-170729>
- [12] B. Zhong, J. Wang, J. Zhou, H. Wu, Q. Jin, *Ind. Eng. Chem. Res.* **2016**, 55 (6), 1609–1622. DOI: <https://doi.org/10.1021/acs.iecr.5b02559>
- [13] B. Song, H. Shi, S. Tan, Y. Tao, *IEEE Trans. Ind. Inf.* **2020**, 17 (9), 6368–6378. DOI: <https://doi.org/10.1109/TII.2020.3015034>
- [14] J. Jiao, W. Zhen, W. Zhu, G. Wang, *IEEE Trans. Ind. Inf.* **2020**, 17 (9), 6347–6356. DOI: <https://doi.org/10.1109/TII.2020.2989810>
- [15] J. Fan, S. J. Qin, Y. Wang, *Control Eng. Pract.* **2014**, 22, 205–216. DOI: <https://doi.org/10.1016/j.conengprac.2013.06.017>
- [16] F. He, C. Wang, S.-K. S. Fan, *Chemom. Intell. Lab. Syst.* **2018**, 183, 79–89. DOI: <https://doi.org/10.1016/j.chemolab.2018.10.010>
- [17] Y. Liu, F. Wang, Y. Chang, F. Gao, D. He, *Chem. Eng. Sci.* **2019**, 209, 115167. DOI: <https://doi.org/10.1016/j.ces.2019.115167>
- [18] J. Jiao, N. Zhao, G. Wang, S. Yin, *ISA Trans.* **2017**, 66 (66), 275–283. DOI: <https://doi.org/10.1016/j.isatra.2016.10.015>
- [19] C. Wei, Z. Song, *IEEE Trans. Ind. Electron.* **2020**, 67 (12), 10876–10886. DOI: <https://doi.org/10.1109/TIE.2019.2962468>
- [20] L. Ma, J. Dong, C. Hu, K. Peng, *Neurocomputing* **2021**, 428, 30–41. DOI: <https://doi.org/10.1016/j.neucom.2020.11.045>
- [21] W. Yu, C. Zhao, *IEEE Trans. Ind. Electron.* **2019**, 67 (6), 5081–5091. DOI: <https://doi.org/10.1109/TIE.2019.2931255>
- [22] H. Wu, J. Zhao, *Comput. Chem. Eng.* **2018**, 115, 185–197. DOI: <https://doi.org/10.1016/j.compchemeng.2018.04.009>
- [23] S. Zhang, K. Bi, T. Qiu, *Ind. Eng. Chem. Res.* **2019**, 59 (2), 824–834. DOI: <https://doi.org/10.1021/acs.iecr.9b05885>
- [24] G. Xu, M. Liu, Z. Jiang, W. Shen, C. Huang, *IEEE Trans. Instrum. Meas.* **2019**, 69 (2), 509–520. DOI: <https://doi.org/10.1109/TIM.2019.2902003>
- [25] X. He, D. Cai, S. Yan, H.-J. Zhang, *Tenth IEEE Int. Conf. on Computer Vision (ICCV'05) Volume I*, IEEE, Beijing, Chinam, October **2005**. DOI: <https://doi.org/10.1109/ICCV.2005.167>
- [26] C. Liu, K. Wang, L. Ye, Y. Wang, X. Yuan, *Inf. Sci.* **2021**, 567, 42–57. DOI: <https://doi.org/10.1016/j.ins.2021.03.026>
- [27] J. Yu, C. Zhang, *J. Process Control* **2020**, 92, 119–136. DOI: <https://doi.org/10.1016/j.jprocont.2020.06.001>
- [28] M. Rezaei-Ravari, M. Eftekhari, F. Saberi-Movahed, *Eng. Appl. Artif. Intell.* **2021**, 97, 104062. DOI: <https://doi.org/10.1016/j.engappai.2020.104062>
- [29] Songjun Zhong, Jinglin Zhou, *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, Liuzhou, China, November **2020**. DOI: <https://doi.org/10.1109/DDCLS49620.2020.9275039>
- [30] K. E. S. Pilario, Y. Cao, M. Shafiee, *Comput. Chem. Eng.* **2019**, 123, 143–154. DOI: <https://doi.org/10.1016/j.compchemeng.2018.12.027>
- [31] G. Birol, C. Undey, S. J. Parulekar, A. Cinar, *Biotechnol. Bioeng.* **2002**, 77 (5), 538–52. DOI: <https://doi.org/10.1002/bit.10115>

RESEARCH ARTICLE

An adaptive strategy for time-varying batch process fault prediction based on stochastic configuration network

Kai Liu^{1,2}  | Xiaoqiang Zhao^{1,2,3}  | Yongyong Hui^{1,2,3} | Hongmei Jiang^{1,2,3}

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

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

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

Correspondence

Xiaoqiang Zhao, College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China.
Email: xqzhao@lut.edu.cn

Funding information

College Industrial Support Project of Gansu Province, Grant/Award Number: 2023CYZC-24; Lanzhou Youth Science and Technology Innovation Program, Grant/Award Number: 2023-QN-36; National Natural Science Foundation of China, Grant/Award Number: 62263021; Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process, Grant/Award Number: 2022KX07

Abstract

Fault prediction ensures safe and stable production, and cuts maintenance costs. Due to the changing operating conditions that lead to the changes in the characteristics of industrial processes, there is a need to monitor the fault state of batch processes in real-time and to accurately predict fault trends. An adaptive slow feature analysis-neighborhood preserving embedding-improved stochastic configuration network (SFA-NPE-ISCN) algorithm for batch process fault prediction is proposed. Firstly, SFA is used to extract the time-varying features of process data and establish the update index of the NPE model. Then, to extract local nearest-neighbor features and reconstruct them by the NPE model with adaptive update capability, square prediction error (SPE) statistics are constructed as fault state features based on the reconstructed error. Further, the hunter-prey optimization (HPO) algorithm optimizes the weights and biases in the stochastic configuration network, and the singular value decomposition (SVD) and QR decomposition of column rotation are introduced to solve the ill-posed problem of SCN and obtain the prediction model of ISCN. Finally, the obtained statistics SPE is formed into a time series, and the ISCN model is used to predict the process state trend. The effectiveness of the proposed algorithm is verified by case studies of industrial-scale penicillin fermentation processes and the Hot strip mill process.

KEYWORDS

adaptive strategy, batch process, fault prediction, stochastic configuration network, time-varying

1 | INTRODUCTION

In the Industry 4.0 era, industrial systems are developing in the direction of scale, intelligence, and complexity. As one of the important ways of industrial production, batch processes have the advantages of small batches, variety, and flexibility in operating batches.¹ With increased automation and increasingly complex processes, the safety and reliability of systems are becoming increasingly important.² Therefore, it is important to monitor and predict the process states, predict the occurrence of faults in advance, and take timely measures to mitigate or avoid losses.^{3–6}

With the development of the Industrial Internet of Things, we can collect nearly all production process data, which can be used to analyze the process status and predict faults. In simple production processes, detecting or predicting faults before production interruptions may be easy to achieve, but in complex industrial processes, especially in the complex batch process industry, the challenges are still serious.⁷ The challenges are mainly due to the large amount of

high-dimensional data and the constantly changing operating conditions, which make process monitoring even more difficult. Traditional algorithms based on gated recurrent units (GRUs) and long and short-term memory (LSTM) are used to predict faults. Zhai et al⁸ fused GRU and extreme Gradient Boosting (XGBoost) to propose a multivariate time series model for predicting the temperature of a heating furnace. Yao et al⁹ constructed a predictive model of deep dynamic regression networks with GRU units that was used for quality prediction in chemical processes. Bai et al¹⁰ proposed a fault alarm prediction model for critical variables in chemical processes based on dynamic inner-principal component analysis (DiPCA) and long- and short-term memory (LSTM). LSTM was used to learn the relationships between variables and predict critical variable fault alarms. Xu et al¹¹ proposed a novel neural network-based classification model using generative adversarial network (GAN) and LSTM to predict pipeline leaks. Xu et al¹² established the TCN-BiGRU-WD fault prediction model for the lifetime prediction of continuous casting rolls. These deep neural networks have many parameters and their training process is quite time-consuming.

In recent years, Wang et al¹³ proposed a stochastic learning framework, called stochastic configuration networks (SCN), which was an incremental learning method with universal approximation properties that used a supervised mechanism to randomly configure the parameters of hidden layer nodes under a set of inequality constraints and was faster for training compared to traditional iterative algorithms. To attenuate the effects caused by noisy data or outliers throughout the training process, Li et al¹⁴ proposed a robust random configuration network (RSCN) framework to deal with the problem of data modeling when given samples contain noise or outliers. Zhang et al¹⁵ proposed a parallel stochastic configuration network (PSCN) model that not only performed well in parameter optimization but also in bearing life prediction. To improve the feature learning capability and the effectiveness of SCN, inspired by the broad learning system (BLS) architecture, Zhang et al¹⁶ proposed a novel extensive stochastic configuration network (BSCN) in which the original features were transformed into mapped features at the feature layer, the mapped features were augmented at the augmentation layer, and then the input weights and biases of the augmented nodes were determined according to the supervision mechanism, and the output weight matrix was calculated by the standard least squares method, resulting in a network with higher regression accuracy and stability. To speed up the construction efficiency of SCN and reduce the redundant nodes of the constructed model, Li et al¹⁷ further compressed the singular value-based model decomposition algorithm and proposed an improved SCN network model for industrial processes. Fang et al¹⁸ proposed a collaborative stochastic configuration network (CSCN) based on a differential evolutionary sparrow search algorithm (DESSA), which improved the prediction accuracy.

Despite SCN and its variants achieved better performance in terms of learning accuracy and generalization ability. However, there may be some better weights and biases that could reduce the SCN residuals faster. In addition, the hidden output matrix of SCN may be rank deficient or multicollinear, thus making the stability of SCN significantly worse.

Furthermore, process data have complex features and directly perform fault prediction on raw high-dimensional complex, which can cause problems such as high computational effort and information redundancy. Effective information fusion is also a difficult task for multi-sensor data. Therefore, it is necessary to find a time series that reflects the process state information.^{19,20} Square prediction errors form the SPE time series, and the fault trend of the process is predicted by predicting the SPE time series. Here, the reconstruction of the input features is the foundation for obtaining the SPE time series. Traditional multivariate statistical process monitoring methods have been applied in many practical engineering areas.^{21,22} In practical industrial production, aging equipment, and changes in operating conditions exist, these factors make static models ineffective. Therefore, it is necessary to update a model for time-varying processes to improve the accuracy of the feature reconstruction according to the changing state of the industrial process. Shang et al²³ proposed a recursive canonical variational analysis based on first-order perturbation theory (RCVA-FOP) to detect faults in time-varying processes, this method was not only effectively adapted to the natural variability of time-varying processes but also detect and identify sensor accuracy degradation. Elshenawy et al²⁴ proposed modified reconstruction-based contributions for sensor fault diagnosis in continuous time-varying processes, this method could adapt to time-varying characteristics and still detect sensor faults. Zhang et al²⁵ proposed an incremental deep computing model for IoT wireless big data feature learning by designing two incremental learning algorithms, namely the parameter-based incremental learning algorithm (PI-TAE) and the structure-based incremental learning algorithm (SI-TAE). To avoid performance drop-in time-varying models, Wu et al²⁶ proposed an adaptive update framework for deep learning based on just-in-time fine-tuning of the stacked autoencoder (JIT-SAE). However, the above adaptive methods cannot well solve the problems of how to distinguish between normal varying working conditions and faults.

Motivated by the above discussion, to better predict the state of time-varying batch processes and effectively predict faults, we propose an adaptive slow feature analysis-neighborhood preserving embedding-improved stochastic

configuration network (SFA-NPE-ISCN) algorithm for batch process fault prediction. Firstly, the adaptive SFA-NPE model effectively monitors the time-varying process while calculating the SPE of the process data constituting the process state sequence by the adaptive SFA-NPE model. Secondly, through the HPO optimization of SCN based on the introduction of singular value decomposition (SVD) and QR decomposition of column rotation, which solve SCN the ill-posed problem to establish the prediction model of ISCN. Finally, the effectiveness of the SFA-SCN-NPE model is verified by two typical batch process cases.

In this paper, a proposed adaptive SFA-NPE-ISCN is used for time-varying batch process fault prediction. The main contributions lie in the following aspects:

1. To describe the dynamic characteristic of time-varying processes, we propose an adaptive model updating strategy. Specifically, we use SFA to extract the dynamic features of the process data, and the failure judgment of the NPE model is established. When the model needs to be updated, the updated data set is selected from the training data in a window near the online sample data.
2. We propose a prediction model of ISCN, which optimizes the parameters of the added hidden nodes using the HPO algorithm under the supervised mechanism of SCN. It can reduce the network residuals faster while combining SVD and QR with column rotation to solve the non-positive definitions problem of SCN.
3. The adaptive NPE model is used to extract data features and feature reconstruction, by which square prediction error (SPE) statistics are constructed as fault state features. The ISCN model is used to achieve state trend prediction.

The structure of the paper is as follows. In Section 2, a brief overview of Slow Feature Analysis (SFA), Stochastic Configuration Network (SCN), and Hunter-prey optimization (HPO) is given, while our proposed adaptive SFA-NPE-ISCN algorithm for time-varying batch process fault prediction is detailed in Section 3. In Section 4, two case studies are used to validate the proposed algorithm, and the conclusions are drawn in Section 5.

2 | PRELIMINARIES

In this section, the basic ideas of Slow Feature Analysis (SFA),²⁷ Stochastic Configuration Networks (SCN),¹³ and Hunter-prey optimization (HPO)²⁸ are reviewed.

2.1 | Slow feature analysis (SFA)

Slow feature analysis prefers to deal with the time-varying characteristic of the process data. The main principles are as follows: Given an m -dimensional input signal $x(t) = \{x_1(t), \dots, x_m(t)\}^T$ with $t \in [t_0, t_1]$, SFA seeks a transformation function $g(x) = [g_1(x), \dots, g_J(x)]^T$ to make the output $y(t) = [y_1(t), \dots, y_J(t)]^T$ of the function change as slowly as possible, where $y_j(t) = g_j(x(t))$. By minimizing the variance of the first derivative of the slow feature, the optimization of SFA is:

$$\begin{aligned} \min \langle \dot{y}_i^2 \rangle_t, i = 1, \dots, m \\ \text{s.t. } \langle y_i \rangle_t = 0 \\ \langle y_i^2 \rangle_t = 1 \\ \langle y_i y_j \rangle_t = 0, \forall i \neq j \end{aligned} \quad (1)$$

where, \dot{y}_i represents the first-order derivative of the slow feature and $\langle \cdot \rangle_t$ represents the average over time, which is defined as formula (2).

$$\langle f \rangle_t \approx \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t) dt \quad (2)$$

where, the angle brackets indicate temporal averaging.

2.2 | Stochastic configuration network (SCN)

SCN is a kind of randomized learning model with supervisory mechanisms. The basic idea of SCN is to start with a small network structure and then gradually add hidden nodes until an acceptable tolerance is achieved. The network structure of SCN is shown in Figure 1.

Under the SCN framework, three algorithmic implementations, are named SC-I, SC-II, and SC-III. In terms of both learning efficiency and generalization, SC-III outperforms the others. SCN is implemented using the SC-III algorithm in this paper.

Suppose that $\text{span}(\Gamma)$ is dense in L_2 space and $\forall g \in \Gamma, 0 < \|g\| < b_g$ for $b_g \in R^+$. Given $0 < r < 1$ and a non-negative real number sequence $\{\mu_L\}$ with $\lim_{L \rightarrow +\infty} \mu_L = 0, \mu_L \leq 1 - r$, for $L = 1, 2, \dots$, expressed by:

$$\delta_L = \sum_{q=1}^m \delta_{L,q}, \delta_{L,q} = (1 - r - \mu_L) \|e_{L-1,q}\|^2, q = 1, 2, \dots, m \quad (3)$$

If the random basis function g_L is generated to satisfy the following inequalities:

$$\langle e_{L-1,q}, g_L \rangle^2 \geq b_g^2 \delta_{L,q}, q = 1, 2, \dots, m \quad (4)$$

Based on inequality constraint, the output weights are evaluated by

$$[\beta_1, \beta_2, \dots, \beta_L] = \arg \min_{\beta} \left\| f - \sum_{j=1}^L \beta_j g_j \right\| \quad (5)$$

It holds that $\lim_{L \rightarrow +\infty} \|f - f_L\| = 0$, where $f_L = \sum_{j=1}^L \beta_j g_j, \beta_j = [\beta_{j,1}, \dots, \beta_{j,m}]^T$.

2.3 | Hunter-prey optimization (HPO)

Hunter prey algorithm is a population-based optimization algorithm inspired by the behavior of predatory animals and has a similar general structure and other optimization algorithms.

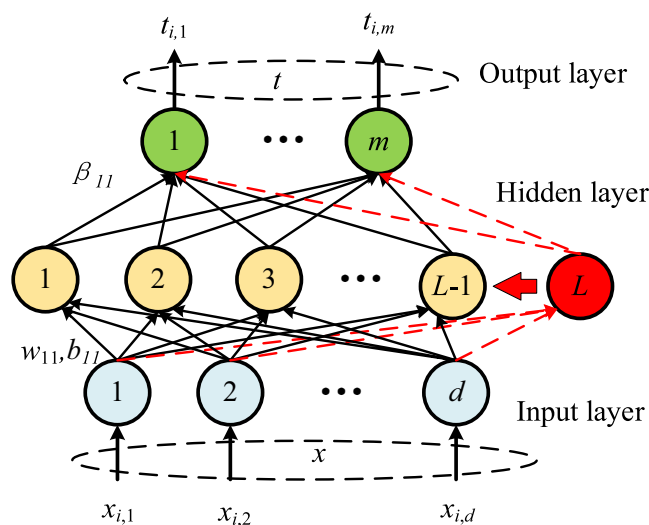


FIGURE 1 The network structure of SCN.

Firstly, the initial overall population is randomly set to $x = \{x_1, x_2, \dots, x_n\}$, the objective function for all members of the population overall is then calculated as $o = \{o_1, o_2, \dots, o_n\}$. The position of each member of the initial population is generated randomly in the search space by formula (6).

$$x_i = \text{rand}(1, d) \cdot (ub - lb) + lb \quad (6)$$

where, x_i is the position of the hunter or prey, lb is the minimum value of the problem variable (lower bound), ub is the maximum value of the problem variable (upper bound), and d is the number of problem variables (dimension). The lower and upper bounds of the search space are defined as shown in formulas (7) and (8):

$$lb = [lb_1, lb_2, \dots, lb_d] \quad (7)$$

$$ub = [ub_1, ub_2, \dots, ub_d] \quad (8)$$

After generating the initial overall and determining the position of each agent, the fitness value of each solution is calculated using the objective function $o_i = f(x_i)$.

The mathematical model of the hunter's search mechanism is given in formula (9):

$$x_{ij}(t+1) = x_{ij}(t) + 0.5 \left[(2CZP_{pos(j)} - x_{ij}(t)) + (2(1-C)Z\mu(j) - x_{ij}(t)) \right] \quad (9)$$

where, $x(t)$ is the current hunter position, $x(t+1)$ is the next iteration position of the hunter, P_{pos} is the location of the prey, μ is the average value of all positions, Z is the adaptive parameter calculated by formula (10), C is the balance parameter between exploration and exploitation calculated by formula (11).

$$\begin{aligned} P &= \vec{R}_1 < C; IDX = (P == 0); \\ Z &= R_2 \otimes IDX + \vec{R}_3 \otimes (\sim IDX) \end{aligned} \quad (10)$$

$$C = 1 - it \left(\frac{0.98}{MaxIt} \right) \quad (11)$$

where, \vec{R}_1 and \vec{R}_3 are random vectors in $[0, 1]$, P is the index value of $\vec{R}_1 < C$, R_2 is a random number within $[0, 1]$, IDX is the index value of the vector \vec{R}_1 that satisfies the condition $(P == 0)$, it is the current number of iterations, and $MaxIt$ is the maximum number of iterations.

To calculate the position of the prey (P_{pos}), we first calculate the average of all positions (μ) from formula (12) and then calculate the distance of each search agent from this average position from formula (13).

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (12)$$

$$D_{enc}(i) = \left(\sum_{j=1}^d (x_{ij} - \mu_j)^2 \right)^{\frac{1}{2}} \quad (13)$$

The search agent with the largest distance from the location mean is considered prey (P_{pos}) by formula (14):

$$\vec{P}_{pos} = \vec{x}_i \mid i \text{ is index of } \text{Max}(\text{end}) \text{ sort}(D_{enc}) \quad (14)$$

If the maximum distance between the search agent and the mean position (μ) is considered at each iteration, the algorithm would delay convergence. To solve this problem, we consider a decreasing mechanism, as shown in formula (15):

$$kbest = \text{round}(C \times N) \quad (15)$$

where, N is the number of search agents.

The location of the prey is calculated by converting formula (14) to formula (16):

$$\vec{P}_{pos} = \vec{x}_i \mid i \text{ is sorted } D_{enc}(kbest) \quad (16)$$

Update the prey position by formula (17):

$$x_{ij}(t+1) = T_{pos(j)} + CZ \cos(2\pi R_4) \times (T_{pos(j)} - x_{ij}(t)) \quad (17)$$

where, $x(t)$ is the current position of the prey, $x(t+1)$ is the next iteration position of the prey; T_{pos} is the global optimal position, R_4 is a random number within $[0, 1]$.

3 | FAULT PREDICTION BASED ON ADAPTIVE SFA-NPE-ISCN ALGORITHM

This section gives the problem formulation of the research and concretizes the implementation of the proposed fault prediction strategy based on the adaptive SFA-NPE-ISCN algorithm.

3.1 | Problem formulation

For batch processes, due to the changes in factors, such as working conditions and the environment, the industrial characteristics also change, which this change is reflected in the gradual change in the correlation of the variables over time. Given the time-varying characteristic, if we still use a static model to describe the process, that will result in a model mismatch and, furthermore, lead to inaccurate monitoring results. It is an important task for process monitoring to monitor the fault states of complex batch processes in real-time and to accurately predict fault trends.

To accurately characterize the variability of a process, it is necessary to develop a model that can learn the dynamic characteristics of a time-varying process while predicting fault trends. We propose a monitoring model based on the adaptive SFA-NPE-ISCN algorithm. Firstly, SFA is used to extract dynamic features and establish a model update index to achieve model mismatch judgment. When the update index exceeds the control limit, it indicates that the model is not mismatched at this time and the parameters of the existing model need to be updated. Otherwise, it means that the model is not mismatched at this time and the existing model will continue to be used for online monitoring. Secondly, NPE is used to extract the process data features, build the process monitoring time series *SPE*, and complete the prediction of the *SPE* trend using the optimized SCN. When the monitoring model mismatches and needs to be updated, an updated dataset is created using normal time-domain samples of the online data, and the model is updated using a neural network incremental strategy to accommodate the dynamic characteristic. The concrete implementation of the proposed monitoring strategy is described below.

3.2 | Adaptive SFA-NPE algorithm scheme

In this paper, SFA is used to extract the slow features of the process training data and construct the model update-index. The extracted slow features are divided into dominant slow features $s_d \in R^M$ and residual slow features $s_e \in R^{M_e}$, with dominant features changing slowly and residual features changing quickly, where M_e denotes the number of slow features that are faster than all inputs $\{x_i(t), 1 \leq i \leq m\}$, and the number of dominant slow features is $M = m - M_e$. To monitor the match properties of the model, the statistics are defined as follows.

$$\begin{aligned} T^2 &= s_s^T s_d \\ T_e^2 &= s_e^T s_e \end{aligned} \quad (18)$$

Meanwhile, the control limits T_{lim}^2 and T_{elim}^2 of the statistics are calculated using the kernel density estimation (KDE) method. When at least one of $T_{lim}^2 \geq T^2$ or $T_e^2 \geq T_{elim}^2$ satisfies the condition, the current model is considered to need to be updated, otherwise, the model does not need to be updated.

Neighborhood preserving embedding (NPE) is used to extract process features and construct a time series *SPE* for fault prediction. NPE is implemented as follows:

For data matrix X with n samples and m variables for each sample, NPE represents the local linear structure of the manifold by local linear reconstruction, which is in the form of mean square error. The objective function representing the reconstruction error in the high-dimensional space is shown in formula (19):

$$\phi(W) = \min \sum_{i=1}^n \left\| x_i - \sum_{j \in Q(i)} w_{ij} x_j \right\|^2 \quad (19)$$

where, $Q(i)$ is the set of neighboring samples, W should satisfy the normalization constraint.

During dimensionality reduction, NPE retains the same local linear reconstruction in the lower dimensional space as in the original space. The dimensionality reduction linear map is computed by solving the generalized eigenvectors, as shown in formula (20):

$$XMX^T a = \lambda XX^T a \quad (20)$$

where, $M = (I - W)^T(I - W)$, $I = \text{diag}(1, \dots, 1)$. The eigenvectors a is arranged in ascending order of eigenvalues to obtain the matrix A .

A relationship between the original data X and the reduced dimensional data Y exists as follows:

$$Y = A^T X = (B^T B)^{-1} B^T X \quad (21)$$

where, $(B^T B)^{-1} B^T$ denotes the transformation matrix. Time series *SPE* is obtained from formula (22):

$$SPE = \|x - BA^T x\|^2 \quad (22)$$

From a certain time, window near the online sampling data in the historical dataset, the updated dataset is selected. The updated dataset is then used to update the NPE model. The updated model learns the characteristics of the new normal data. The update strategy is shown in formula (23):

$$\begin{aligned} \phi(W) &= \min \sum_{i=1}^n \left\| \Delta x_i - \sum_{j \in Q(i)} w_{ij} \Delta x_j \right\|^2 \\ \Delta x &= \varphi(x, \theta + \Delta\theta) - x \end{aligned} \quad (23)$$

where, θ is the original parameter obtained by training the NPE model on the old training set, $\Delta\theta$ is the parameter increment obtained by training the NPE model on the incremental data set, $\varphi(x, \theta)$ denotes the reconstructed output of the old NPE model, and $\Delta x = \varphi(x, \theta + \Delta\theta) - x$ denotes the reconstructed error of the model after the parameter is updated from θ to $\Delta\theta$.

3.3 | Improved stochastic configuration network (ISCN)

The SCN construction process is equivalent to gradually reducing the network residuals by adding a new hidden node. Owing to the inherent randomness of the SCN algorithm, it may be that there are better-hidden parameters that can

reduce the network residuals faster. Compared to optimization algorithms such as PSO, TSA, LFD, HHO, and WOA, the HPO algorithm is used to obtain better network parameters.²⁸ Therefore, we propose an improved stochastic configuration network (ISCN).

In ISCN, the network residuals are defined as a fitness function of the HPO algorithm under the SCN supervision mechanism. The optimized algorithm can minimize the network residuals, while its general approximation is guaranteed by the supervisory mechanism of SCN.

For a given objective function f , the current residual error with $L-1$ hidden neurons is defined as:

$$e_{L-1} = f - f_{L-1} = [e_{L-1,1}, \dots, e_{L-1,q}, \dots, e_{L-1,m}]^T \quad (24)$$

Further, $\|e_L\|^2$ is shown by formula (25):

$$\begin{aligned} \|e_L\|^2 &= \|e_{L-1} - \beta_L g_L\|^2 \\ &= \sum_{q=1}^m \langle e_{L-1,q} - \beta_{L,q} g_L, e_{L-1,q} - \beta_{L,q} g_L \rangle \\ &= \sum_{q=1}^m \left(\langle e_{L-1,q}, e_{L-1,q} \rangle - 2 \langle e_{L-1,q}, \beta_{L,q} g_L \rangle + \langle \beta_{L,q} g_L, \beta_{L,q} g_L \rangle \right) \\ &= \sum_{q=1}^m \left(\langle e_{L-1,q}, e_{L-1,q} \rangle - \frac{\langle e_{L-1,q}, g_L \rangle^2}{\|g_L\|^2} \right) \end{aligned} \quad (25)$$

where, $\beta_{L,q}$ is defined as $\langle e_{L-1,q}, g_L \rangle / \|g_L\|^2$.

To improve the reduction rate of network residuals and the generalization ability of SCN, we choose the network residuals as the fitness function of the predation optimization algorithm. The fitness function is defined as:

$$\begin{aligned} \phi_{L,q} &= \begin{cases} +\infty, \xi_{L,q} \leq 0 \\ e_{L-1,q}^T(X) e_{L-1,q}(X) - \frac{(e_{L-1,q}^T(X) \cdot h_L(X))^2}{h_L^T(X) \cdot h_L(X)} \\ +\Psi(L)(\|w_L\|^2 + \|b_L\|^2), \xi_{L,q} > 0 \end{cases} \\ fit(L) &= \sum_{q=1}^m \phi_{L,q}, q = 1, 2, \dots, m \end{aligned} \quad (26)$$

where, $\Psi(L)$ is the regularization function and h_L is the output of the L th hidden node from formula (27):

$$h_L = [g_L(w_L^T x_1 + b_L), \dots, g_L(w_L^T x_N + b_L)]^T \quad (27)$$

In SCN, the output weight $\beta^* = [\beta_1, \beta_2, \dots, \beta_L]$ can be analytically determined by solving the least-squares solution of the linear system $H_L \beta = T$.

$$\beta^* = \arg \min \|H_L \beta - T\|_F^2 = H_L^\Delta T \quad (28)$$

where H_L is the hidden layer output matrix. The SVD decomposition of H_L^Δ is given by formula (29).

$$\begin{aligned} H_L^\Delta &= V \Sigma^- U^T = \sum_{i=1}^k \frac{1}{\sigma_i} v_i u_i^T \\ \Sigma^- &= \begin{bmatrix} \Sigma_k^- & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (29)$$

where U and V are two orthogonal column matrices, $\Sigma_k^- = \text{diag}(\sigma_1^{-1} \sigma_2^{-1} \dots \sigma_k^{-1})$ and $\sigma_1 \geq \sigma_2 \dots \geq \sigma_k \geq 0$. σ_i is the singular value of H_L . If the output matrix H_L is severely ill-posed, σ_i may be close to zero, and σ_i^{-1} would be large. When these singular values are close to zero in the calculation, it makes the solution unstable.

If the hidden layer output matrix is rank deficient or multicollinear, it may cause serious instability of the least-squares solution. We incorporate truncated SVD and QR with column pivoting to produce a well-conditioned submatrix of the hidden output. Firstly, the rank of the hidden output matrix HL is estimated using singular value decomposition; secondly, a submatrix of the permutation and orthogonal matrices is obtained using QR with column pivots that are as well-conditioned as possible; then, a more independent set of HL columns is selected based on the permutation matrix. Finally, we obtain a minimum norm least squares solution using a well-conditioned submatrix. The flowchart of ISCN is shown in Figure 2.

3.4 | Fault prediction procedure based on adaptive SFA-NPE-ISCN algorithm

A flowchart of the proposed adaptive SFA-NPE-ISCN algorithm for fault prediction is shown in Figure 3, the specific steps of which are as follows:

- Step 1: Collect offline data and normalize them to train the NPE model; meanwhile, the SFA algorithm is used to calculate updated indicators T^2 and T_e^2 by formula (19), and further calculate control limits T_{lim}^2 and T_{elim}^2 by kernel density estimation (KDE) method, which is described in detail later.
- Step 2: Collect and standardize online data and use SFA to calculate updated indicators T_{new}^2 and T_{new}^2 for online data; When at least one of $T_{new}^2 \geq T_{lim}^2$ or $T_{new}^2 \geq T_{elim}^2$ satisfies the condition, the current NPE model is considered to need to be updated, otherwise NPE model does not need to be updated.
- Step 3: When the NPE model needs to be updated, the updated dataset is selected from a time window near the online sample data in the historical dataset by formula (24). The updated NPE model learns the features of new normal data.
- Step 4: According to the features learned from the NPE model to extract the statistic SPE by formula (23). SPE is arranged to form a time series by the preceding and following moments, and ISCN is then used to predict SPE state trends.

The specific steps of the KDE method are as follows:

KDE gives a univariate kernel function, which is shown in formula (30):

$$f(j) = \frac{1}{n} \sum_{i=1}^n K_h(j - j_i) = \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{j - j_i}{\sigma}\right) \quad (30)$$

where, j is the sample data, j_i is the observation value, σ is the window width, n is the number of observation values, and K is the kernel function. In this paper, the Gaussian kernel function is selected, and the test level $\alpha = 0.95^{29}$ can be obtained by formula (31) and formula (32) to seek the control limits T_{lim}^2 and T_{elim}^2 .

$$\begin{aligned} \int_{-\infty}^{T_a^2} f(T^2) d(T^2) &= \int_{-\infty}^{T_a^2} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{T^2 - T_i^2}{\sigma}\right) d(T^2) \\ &= \int_{-\infty}^{T_a^2} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(T^2 - T_i^2)^2}{2\sigma}\right) \right\} d(T^2) \\ &= \alpha \end{aligned} \quad (31)$$

$$\begin{aligned} \int_{-\infty}^{T_{ea}^2} f(T_e^2) d(T_e^2) &= \int_{-\infty}^{T_{ea}^2} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{T_e^2 - T_{ei}^2}{\sigma}\right) d(T_e^2) \\ &= \int_{-\infty}^{T_{ea}^2} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(T_e^2 - T_{ei}^2)^2}{2\sigma}\right) \right\} d(T_e^2) \\ &= \alpha \end{aligned} \quad (32)$$

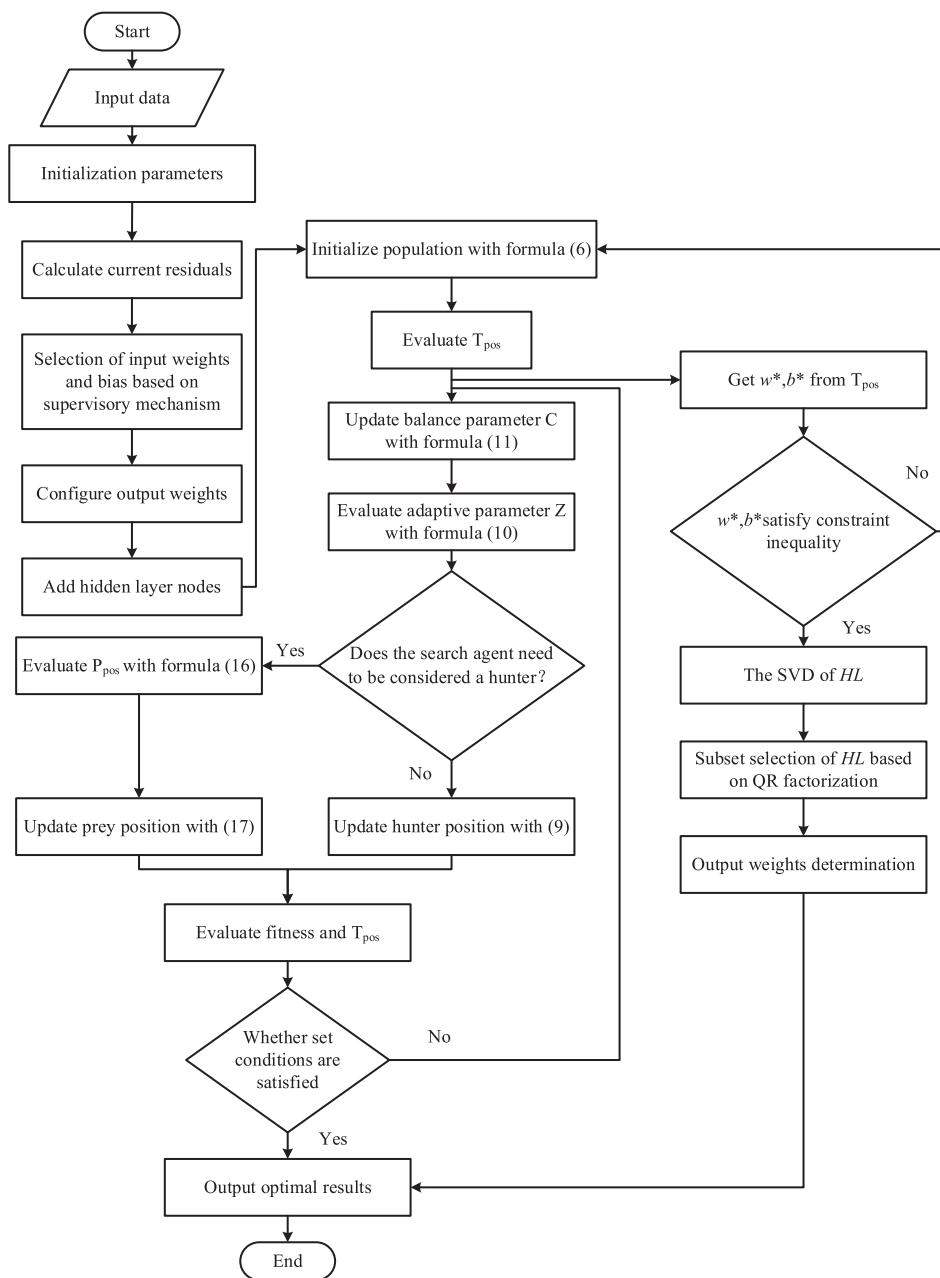


FIGURE 2 The flowchart of ISCN.

To verify the validity of the model, we use Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and coefficient of determination R^2 to evaluate the prediction effect, which is calculated as follows:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \quad (33)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (34)$$

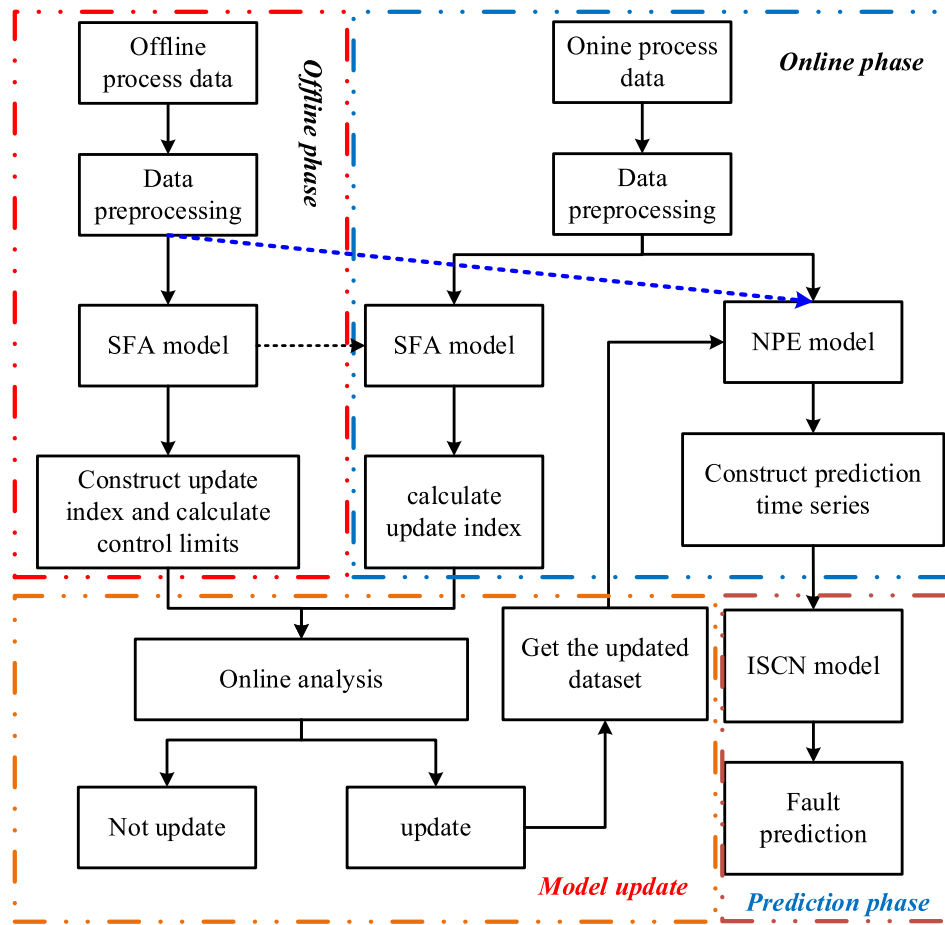


FIGURE 3 Fault prediction framework of adaptive SFA-NPE-ISCN algorithm.

$$MAPE = \frac{100\%}{m} \sum_{i=1}^m \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (35)$$

$$R^2 = 1 - \frac{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}{\frac{1}{m} \sum_{i=1}^m (y_i - \bar{y})^2} \quad (36)$$

where m is the number of prediction points, i is the ordinal number of prediction points, y_i is the actual value, \bar{y}_i is the average of y_i , \hat{y}_i is the predicted value. MAE and MAPE values are as small as possible and take values greater than 0. RMSE values are smaller, which indicates higher accuracy. R^2 describes the ability of the prediction model to fit the actual data curve, the larger the better, and the range of values is $(-\infty, 1)$.

4 | CASE STUDIES

We conduct two batch process cases involving the Industrial-scale fed-batch fermentation process and the Hot strip mill process to verify the effectiveness of our proposed adaptive SFA-NPE-ISCN algorithm in fault prediction, specifically, adaptive SFA-NPE-ISCN algorithm is compared with NPE-LSTM, TNPE-LSTM, NPE-SCN, and TNPE-SCN algorithms. In addition, we give two structures with different parameters, namely, adaptive SFA-NPE-ISCN-I and adaptive SFA-NPE-ISCN-II. For adaptive SFA-NPE-ISCN-I, it only applies the HPO algorithm to optimize the randomly hidden parameters. Based on adaptive SFA-NPE-ISCN-I, the singular value decomposition (SVD) and QR decomposition of

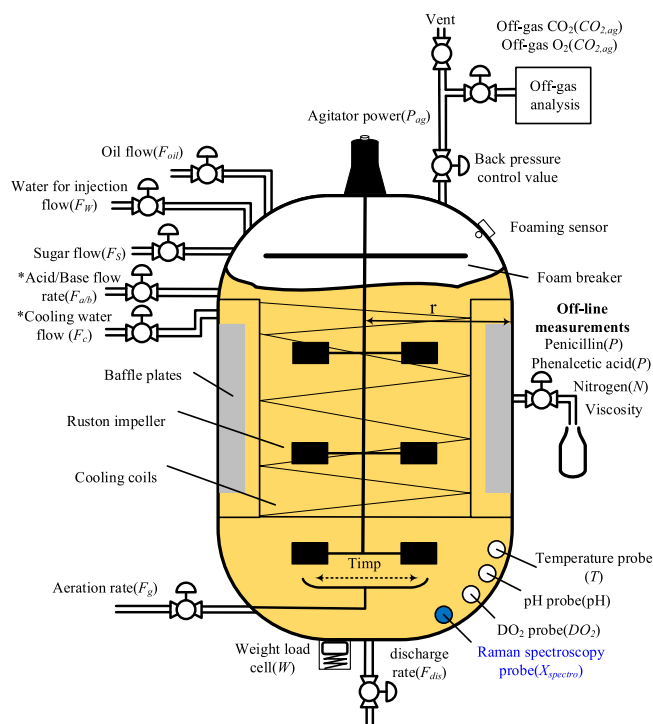


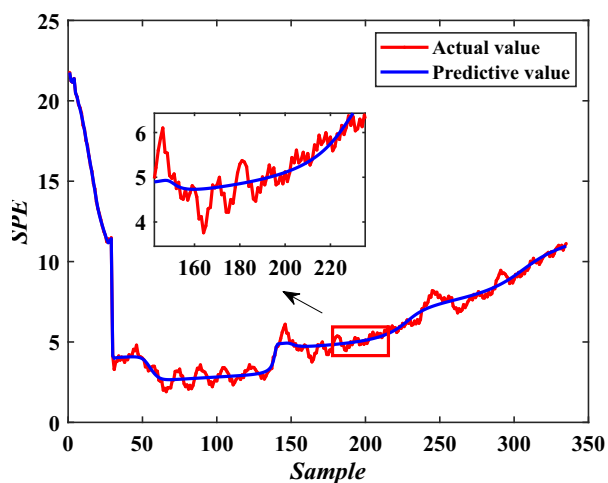
FIGURE 4 Schematic of the Industrial-scale fed-batch fermentation process.

TABLE 1 Description of process variable in the industrial-scale penicillin fermentation process.

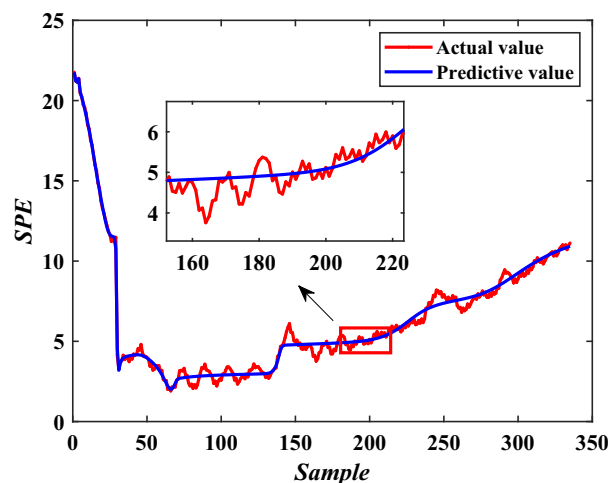
Variable no.	Variable description	Variable no.	Variable description
1	Aeration rate (L/h)	11	Substrate concentration (g/L)
2	Agitator RPM (RPM)	12	Dissolved oxygen conc. (mg/L)
3	Sugar feed rate (L/h)	13	Vessel Volume (L)
4	Acid flow rate (L/h)	14	Vessel Weight (Kg)
5	Base flow rate (L/h)	15	pH (pH)
6	Heating/cooling water flow rate (L/h)	16	Temperature (K)
7	Heating water flow rate (L/h)	17	Generated heat (kJ)
8	Water for injection/dilution (L/h)	18	CO ₂ percent in off-gas (%)
9	Airhead pressure (bar)	29	PAA flow (L/h))
10	Dumped broth flow (L/h)	20	Oil flow (L/hr)

TABLE 2 Comparison of process monitoring indexes.

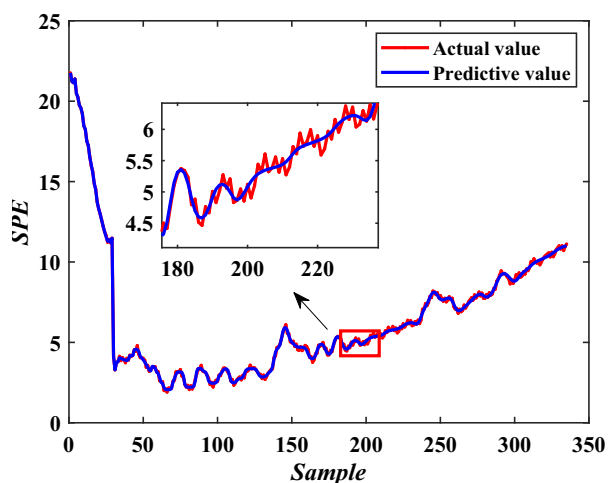
Algorithms	FDR	MAR	FAR
NPE	0.8767	0.1233	0.0371
TNPE	0.9034	0.0966	0.0485
SFA-NPE	0.9934	0.0066	0.0000



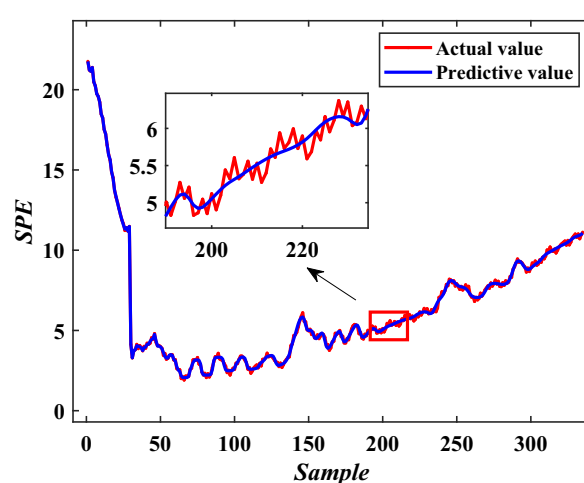
(A) NPE-LSTM



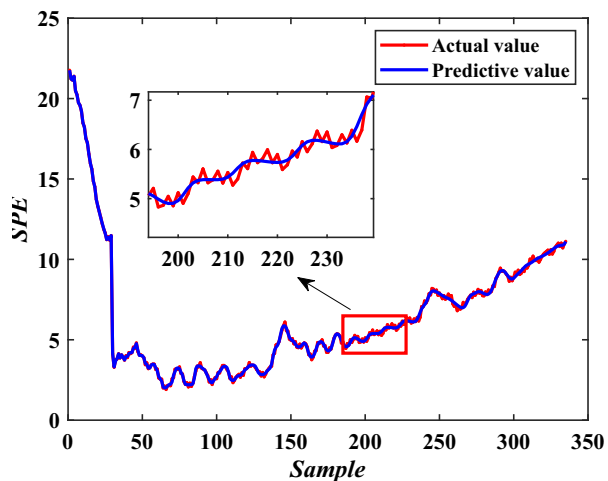
(B) TNPE-LSTM



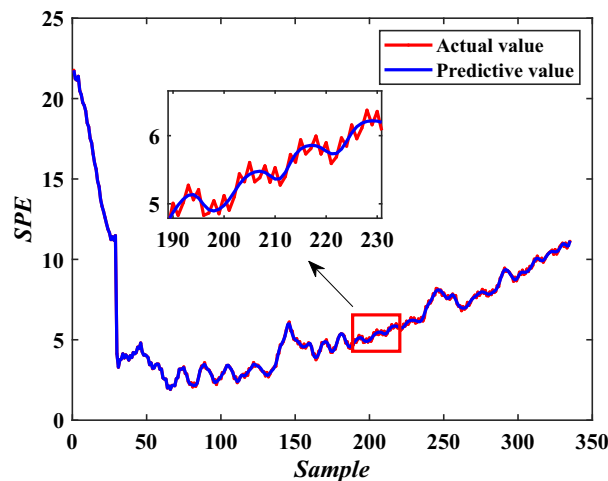
(C) NPE-SCN



(D) TNPE-SCN



(E) SFA-NPE-ISCN-I



(F) SFA-NPE-ISCN-II

FIGURE 5 Prediction result chart for different algorithms.

column rotation are introduced adaptive SFA-NPE-ISCN-II to extract a linearly independent subset of the hidden layer output matrix for ill-posed problems.

4.1 | Industrial-scale fed-batch fermentation process

IndPenSim is an industrial-scale penicillin fermentation simulation platform. Compared to traditional benchmark penicillin fermentation platforms,³⁰ it considers the growth, metabolism, and degradation of large-scale penicillin fermentation besides modeling the necessary online and offline variables. By specifying multiple control strategies to reduce system fluctuations, and maximize the yield of penicillin. The schematic of the Industrial-scale fed-batch fermentation process is shown in Figure 4. More specific descriptions can be referenced from the corresponding original paper.³¹

In this case study, we select normal 45 batches and 5 batches of data under different control strategies and fermentation reaction times of 230 h, in which the data are collected once every 12 min. we select 20 main process variables, as shown in Table 1.

In this case, the five fault batches are disturbance in aeration flow rate, disturbance in vessel back pressure, disturbance in substrate feed rate, disturbance in base flow rate, and disturbance in coolant flow rate. The above fault batches are monitored by NPE, TNPE, and SFA-NPE. The average monitoring indexes for all fault batches are given in Table 2. The Fault Detection Rate (FDR) represents the ratio between the number of faults correctly detected and the total number of faults present, and the Missed Alarm Rate (MAR) represents the ratio between the number of faults not correctly detected and the total number of faults present, and the False Alarm Rate (FAR) represents the ratio between the number of false alarms and the number of faults not present.

It can be noticed from Table 2 that SFA-NPE with adaptive capability has better process monitoring performance. Specifically, it has a higher FDR and lower MAR and FAR. Adaptive SFA-NPE with better monitoring performance can provide a more accurate time series for fault prediction. The fault state feature SPE is formed on a time series, which is then divided into two parts: the training set and the test set. We use the former 815 samples as the training data set and the latter 335 samples as the test data set.

This paper involves an LSTM network with the following parameters: the maximum epoch number is 1000, the hidden unit number is 20, and the mini-batch number is 10. The ISCN network parameters are: maximum hidden node number is 500, training tolerance is 0.00001, and maximum candidate node number is 100. After the model is trained, the network parameters are saved, and then the trained model is used to predict the fault state trends. The prediction results of disturbance in aeration flow rate are shown in Figure 5, where the blue curve is the predicted value and the red curve is the true value. The SCN-related method parameters are consistent with ISCN.

To visualize the effectiveness of the models, MAE, RMSE, MAPE, and R^2 are used to evaluate the prediction effect, and the results are shown in Table 3.

From Figure 5 and Table 3, it can be concluded that the proposed SFA-NPE-ISCN is much more accurate than NPE-LSTM, TNPE-LSTM, NPE-SCN, and TNPE-SCN in fault conditions of prediction. TNPE-SCN and TNPE-LSTM are more conducive to state prediction than NPE-SCN and NPE-LSTM because TNPE considers the time-series features of the process data. The SPE sequences obtained are more time-seriousness and more conducive to state prediction. TNPE-SCN and NPE-SCN have higher prediction accuracy than TNPE-LSTM and NPE-LSTM, due to the SCN having greater robustness for disturbances in the process data. In addition, compared to LSTM, SCN has fewer number of parameters and is faster for training. Our proposed SFA-NPE-ISCN algorithm further optimizes SCN by HPO algorithm to improve the

TABLE 3 Comparison of algorithms in evaluation index.

N0.	Algorithms	MAE	RMSE	MAPE	R^2
1	NPE-LSTM	0.2672	0.3482	6.0100	0.9922
2	TNPE-LSTM	0.2669	0.3495	5.6851	0.9921
3	NPE-SCN	0.1000	0.1235	2.0585	0.9990
4	TNPE-SCN	0.0998	0.1284	2.0957	0.9989
5	SFA-NPE-ISCN-I	0.0929	0.1226	1.8823	0.9991
6	SFA-NPE-ISCN-II	0.0747	0.0956	1.4693	0.9994

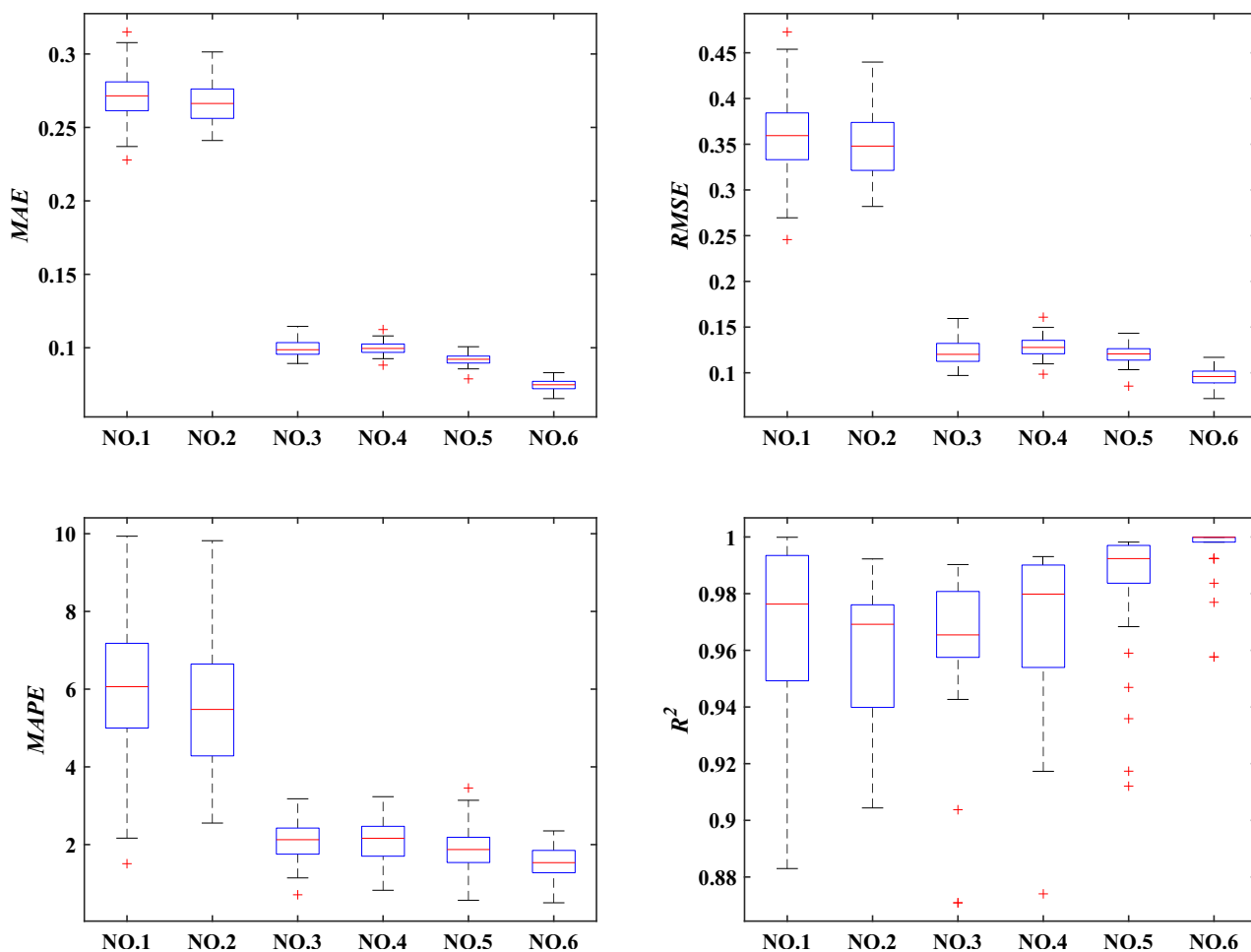


FIGURE 6 Statistics on the performance of different algorithms in 50 test batches.

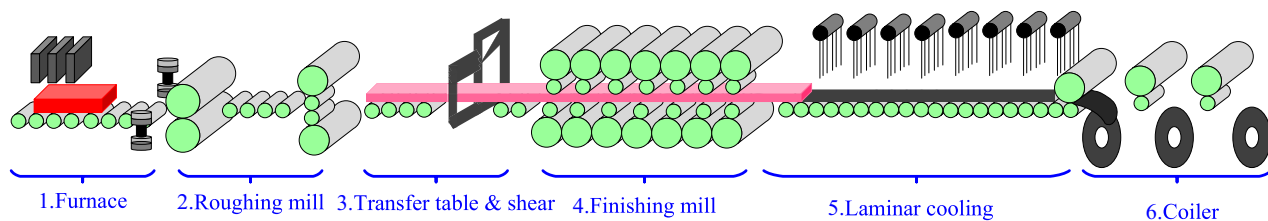


FIGURE 7 Schematic of the hot strip mill process.

TABLE 4 Process and quality variables in finishing mill.

Variable no.	Type	Description	Unit
1-7	Process variable	Average roll gap of 7 stands	mm
8-14	Process variable	Roll force of 7 stands	MN
15-20	Process variable	Bending roll force of 6 stands	MN
21	Quality variable	Exit thickness at the last finishing mill stand	Mm

prediction performance, and consider the data time-varying characteristic, so that the prediction accuracy is higher. SFA-NPE-ISCN-II further improves the prediction performance by considering the SCN the ill-posed problem compared to SFA-NPE-ISCN-I. The statistical box plots of the predictive indexes for the 50 test batches of data are shown in Figure 6.

TABLE 5 Comparison of process monitoring indexes.

Algorithms	FDR	MAR	FAR
NPE	0.9086	0.0914	0.0136
TNPE	0.9441	0.0599	0.0079
SFA-NPE	0.9895	0.0105	0.0000

TABLE 6 Comparison of algorithms in evaluation index.

No.	Algorithms	MAE	RMSE	MAPE	R ²
1	NPE-LSTM	0.0333	0.0418	2.8452	0.9952
2	TNPE-LSTM	0.0109	0.0170	1.1688	0.9991
3	NPE-SCN	0.0080	0.0118	0.9138	0.9996
4	TNPE-SCN	0.0057	0.0084	0.5192	0.9998
5	SFA-NPE-ISCN-I	0.0013	0.0023	0.1265	1.0000
6	SFA-NPE-ISCN-II	0.0011	0.0021	0.1079	1.0000

As can be seen from Figure 6, our proposed method has higher prediction accuracy according to MAE, RMSE, MAPE, and R². Specifically, the values of MAE, RMSE, and MAPE are smaller and more concentrated, while the value of R² is closer to 1 which indicates that the predicted value is closer to the actual value.

4.2 | Hot strip mill process

The hot strip rolling process is a high investment, high quality, and high productivity rolling process. A typical hot strip rolling line includes a heating furnace, roughing mill, transfer table & shear, finishing mill, laminar cooling equipment, and coiler. The Schematic of the Hot strip mill process is shown in Figure 7. The process is as follows: the slab is heated in a heating furnace and then rolled several times through the roughing mill to form an intermediate slab. The intermediate slab is then quickly transported to the finishing mill using flying shear and intermediate delay roller tables for more precise control to obtain a strip that meets the requirements. Finally, the strip is treated in a laminar flow cooling plant to improve the properties and coiled for storage. Strip thickness is one of the key factors in the production of hot-rolled strips. According to customer requirements, the hot rolled strip process can produce strip thicknesses from 1.5 mm to 12.7 mm to meet different requirements. More detailed descriptions of the hot-rolled strip process can be found in the literature.³²

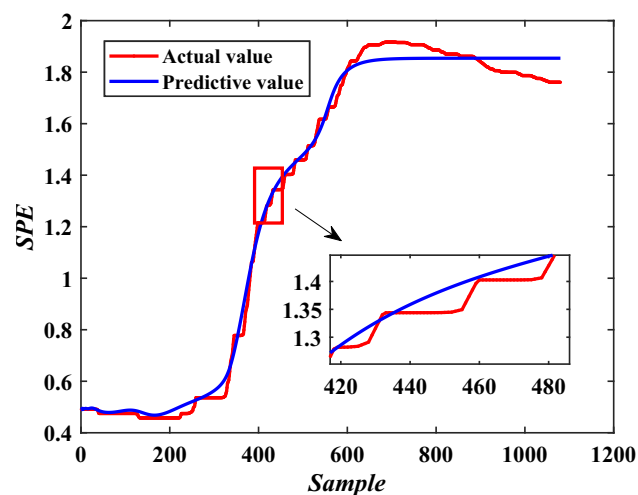
In this case study, a hot strip continuous rolling line (width 1700 mm) at a steel company is used for the study, and experimental verification is carried out by collecting field data. We focus on the process variables of roll gap, rolling force, and bending roll force for the seven stands of the finishing mill. The exit thickness of the finishing mill stand is a quality variable. The process and quality variables for the finishing mill are shown in Table 4.

In this case study, three typical types of faults in finishing rolling mills are considered: the faults in the bending roll system, the faults in the press-down system, and the faults in the cooling water between stands. The average indexes of fault detection for several batches are shown in Table 5.

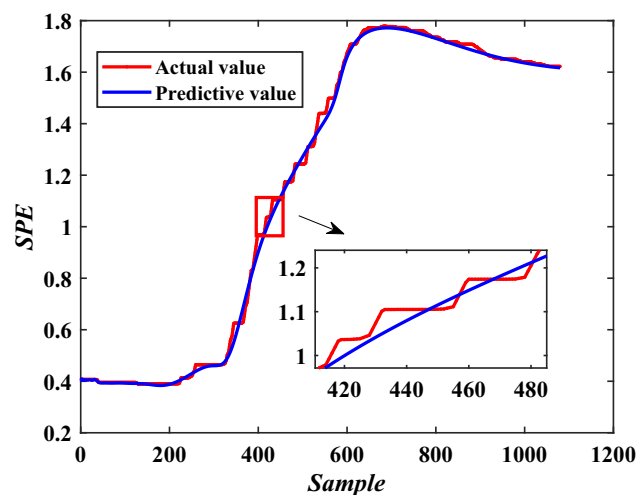
It can be concluded from Table 5 that adaptive SFA-NPE with better monitoring performance can provide a more accurate time series for fault prediction. The fault state feature SPE is formed into a time series, which is then divided into two parts: the training set and the test set. We use the former 2,520 samples as the training data set and the latter 1,080 samples as the test data set. The network parameters are the same as those of the case study 1. The prediction results of faults in the press-down system are shown in Figure 5, where the blue curve is the predicted value and the red curve is the true value.

To visualize the effectiveness of the models, MAE, RMSE, MAPE, and R² are used to evaluate the prediction effect, and the results are shown in Table 6.

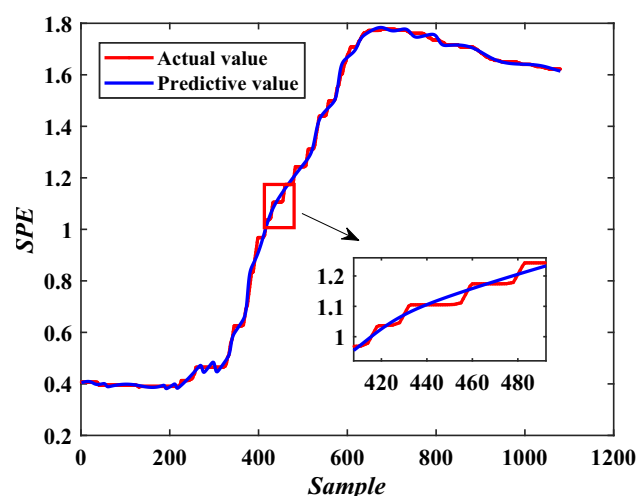
In Figure 8A, the NPE-LSTM prediction performance is poor and does not predict the actual state trend well after 700 sample points. In Figure 8B, TNPE-LSTM significantly improves the prediction performance compared with



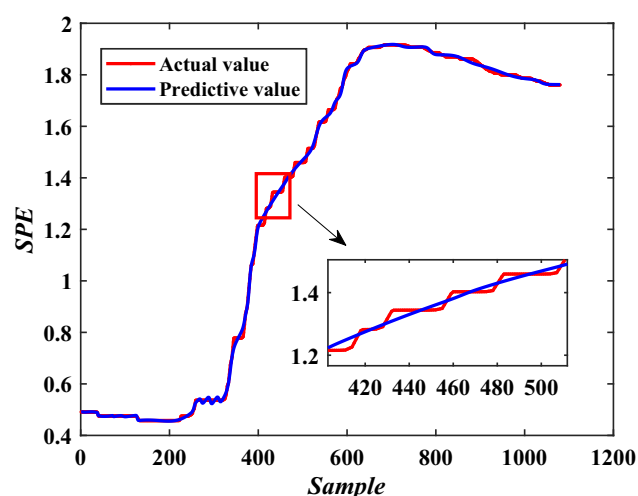
(A) NPE-LSTM



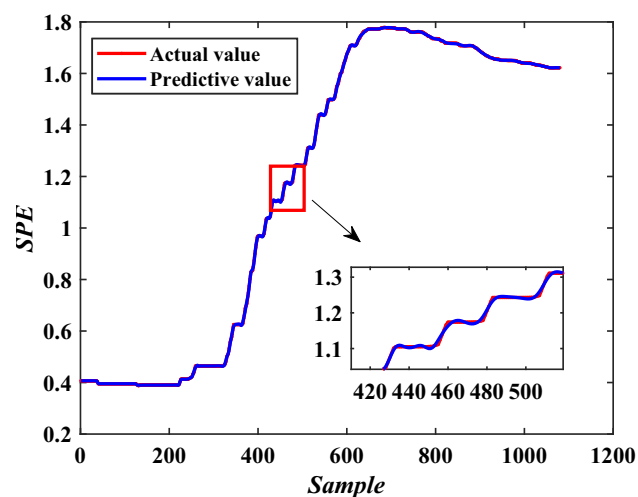
(B) TNPE-LSTM



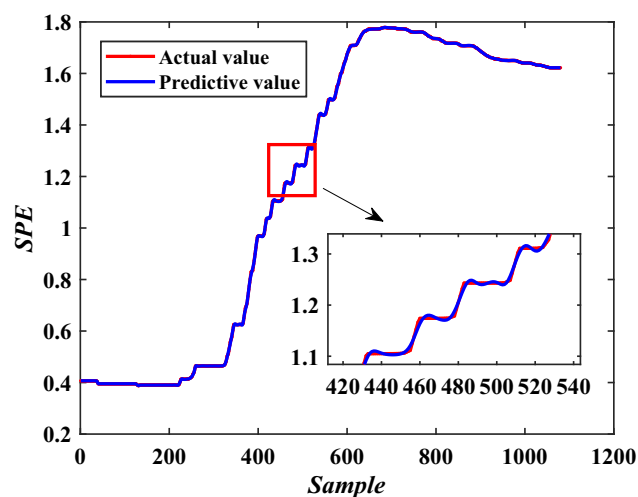
(C) NPE-SCN



(D) TNPE-SCN



(E) SFA-NPE-ISCN-I



(F) SFA-NPE-ISCN-II

FIGURE 8 Prediction result chart for different algorithms.

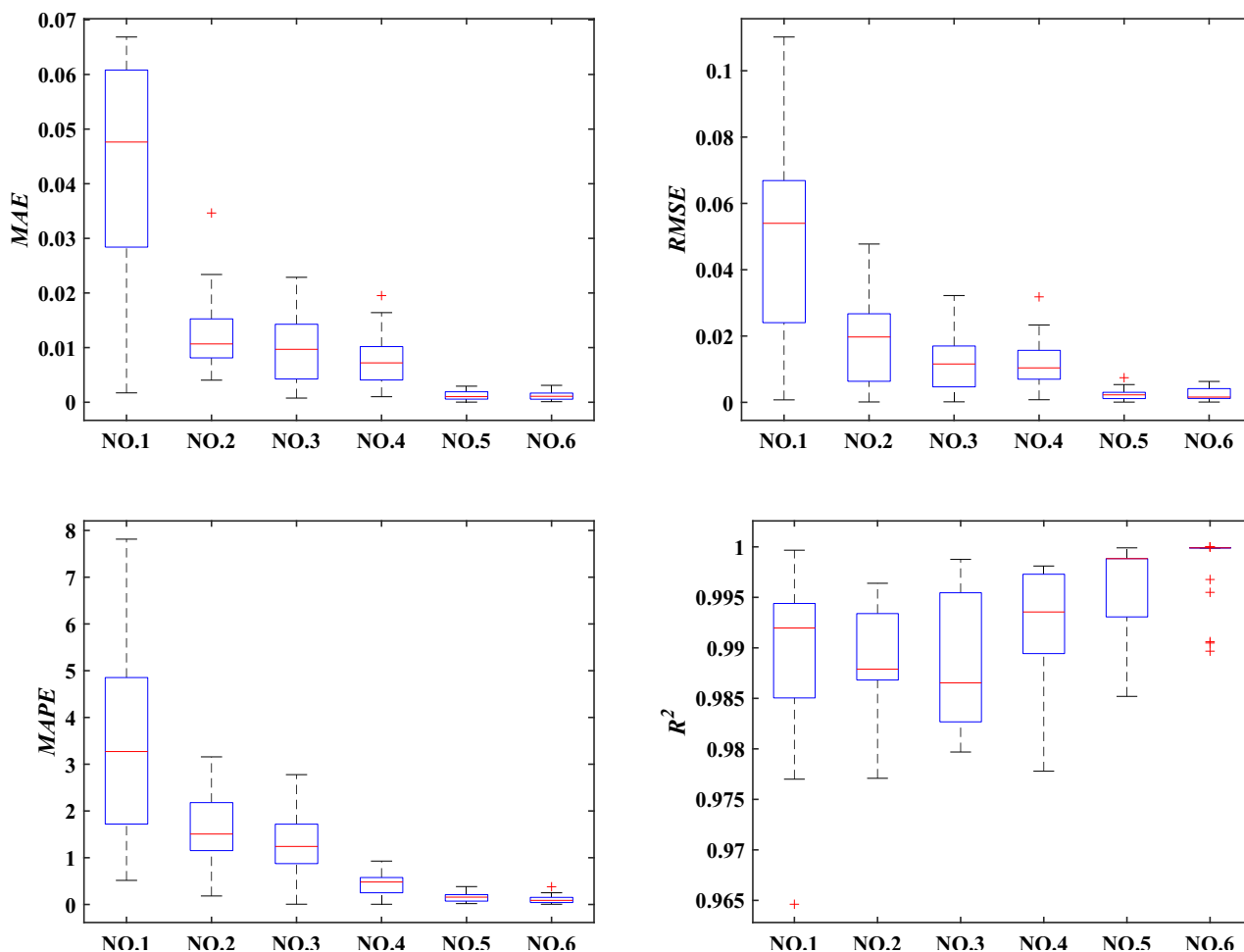


FIGURE 9 Statistics on the performance of different algorithms in 22 test batches.

Figure 8A. SFA-NPE-ISCN-II in Figure 8F and SFA-NPE-ISCN-I in Figure 8E show a significant improvement in prediction performance compared to TNPE-SCN and NPE-SCN. Combined with the analysis of the predictive indexes in Table 6, SFA-NPE-ISCN-II in Figure 8F predicts better compared to SFA-NPE-ISCN-I in Figure 8E, and TNPE-SCN in Figure 8D predicts better compared to NPE-SCN in Figure 8C. Overall, our proposed SFA-NPE-ISCN-II algorithm is the most effective for process state prediction. The statistical box plots of the predictive indexes for the 22 test batches of data are shown in Figure 9.

It can be noted from the box plots in Figure 9 that the proposed method in this paper has higher predictive accuracy and less statistical bias.

5 | CONCLUSION

In this study, to detect the fault states of complex batch processes as soon as possible and to accurately predict the trends of faults, an adaptive slow feature analysis -neighborhood preserving embedding-improved stochastic configuration network (SFA-NPE-ISCN) algorithm is proposed for fault prediction. Firstly, NPE models are adaptively updated based on the extraction of time-varying properties of the process using SFA, and the updated models reconstruct the process data to construct squared prediction error statistics as fault state features. Then, the Hunter-prey optimization (HPO) algorithm is used to optimize the weights and deviations of the stochastic configuration network, and the singular value decomposition (SVD) and QR decomposition of column rotation are introduced to solve the discomfort problem of the SCN to obtain the prediction model of ISCN. Finally, the obtained SPE are composed into time series and the ISCN prediction model is used to achieve process state trend prediction. The verification of the SFA-NPE-ISCN algorithm in the industrial-scale fed-batch fermentation process and the hot strip mill process demonstrated significant

improvements in prediction accuracy compared to other comparative algorithms. On data representation, deep neural networks are more effective than shallow neural networks. Therefore, in the future, we will extend our work to deep networks.

ACKNOWLEDGEMENTS

This research work has been awarded by the National Natural Science Foundation of China (62263021), College Industrial Support Project of Gansu Province (2023CYZC-24), Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (2022KX07), and Lanzhou Youth Science and Technology Innovation Program (2023-QN-36).

CONFLICT OF INTEREST STATEMENT

No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

DATA AVAILABILITY STATEMENT

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

Data Availability and Reproducibility Statement

Data are available on request from the authors.

ORCID

Kai Liu  <https://orcid.org/0000-0002-8624-2208>

Xiaoqiang Zhao  <https://orcid.org/0000-0001-5687-942X>

REFERENCES

1. Zhao X, Liu K, Hui Y. Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm. *Trans Inst Meas Contr*. 2023;45(1):89-103. doi:10.1177/01423312221108519
2. Tan S, Zhou X, Shi H, Song B. Adaptive slow feature analysis-sparse autoencoder based fault detection for time-varying processes. *J Taiwan Inst Chem Eng*. 2023;142:104599. doi:10.1016/j.jtice.2022.104599
3. Gao X, Ma D, Han H, Gao H. Fault prediction of complex industrial process based on DAE and TCN. *Chin J Sci Instrum*. 2021;42(6):140-151.
4. Arunthavanathan R, Khan F, Ahmed S, Imtiaz S. An analysis of process fault diagnosis methods from safety perspectives. *Comput Chem Eng*. 2021;145:107197. doi:10.1016/j.compchemeng.2020.107197
5. Khan F, Rathnayaka S, Ahmed S. Methods and models in process safety and risk management: past, present and future. *Process Saf Environ Prot*. 2015;98:116-147. doi:10.1016/j.psep.2015.07.005
6. Khan S, Yairi T. A review on the application of deep learning in system health management. *Mech Syst Signal Process*. 2018;107:241-265. doi:10.1016/j.ymssp.2017.11.024
7. Li Y. A fault prediction and cause identification approach in complex industrial processes based on deep learning. *Comput Intell Neurosci*. 2021;2021:1-13. doi:10.1155/2021/6612342
8. Zhai N, Yao P, Zhou X. Multivariate time series forecast in industrial process based on XGBoost and GRU. Paper presented at: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) 2020.
9. Yao L, Ge Z. Causal variable selection for industrial process quality prediction via attention-based GRU network. *Eng Appl Artif Intel*. 2023;118:105658. doi:10.1016/j.engappai.2022.105658
10. Bai Y, Xiang S, Cheng F, Zhao J. A dynamic-inner LSTM prediction method for key alarm variables forecasting in chemical process. *Chin J Chem Eng*. 2023;55:266-276. doi:10.1016/j.cjche.2022.08.024
11. Xu P, Du R, Zhang Z. Predicting pipeline leakage in petrochemical system through GAN and LSTM. *Knowl Based Syst*. 2019;175:50-61. doi:10.1016/j.knsys.2019.03.013
12. Xu E, Zou F, Shan P. A multi-stage fault prediction method of continuous casting machine based on Weibull distribution and deep learning. *Alex Eng J*. 2023;77:165-175. doi:10.1016/j.aej.2023.06.079
13. Wang D, Li M. Stochastic configuration networks: fundamentals and algorithms. *IEEE Trans Cybern*. 2017;47(10):3466-3479. doi:10.1109/TCYB.2017.2734043
14. Li M, Huang C, Wang D. Robust stochastic configuration networks with maximum correntropy criterion for uncertain data regression. *Inform Sci*. 2019;473:73-86. doi:10.1016/j.ins.2018.09.026

15. Zhang C, Ding S, Zhang J, Jia W. Parallel stochastic configuration networks for large-scale data regression. *Appl Soft Comput*. 2021;103:107143. doi:[10.1016/j.asoc.2021.107143](https://doi.org/10.1016/j.asoc.2021.107143)
16. Zhang C, Ding S, Du W. Broad stochastic configuration network for regression. *Knowl Based Syst*. 2022;243:108403. doi:[10.1016/j.knosys.2022.108403](https://doi.org/10.1016/j.knosys.2022.108403)
17. Li K, Yang C, Wang W, Qiao J. An improved stochastic configuration network for concentration prediction in wastewater treatment process. *Inform Sci*. 2023;622:148-160. doi:[10.1016/j.ins.2022.11.134](https://doi.org/10.1016/j.ins.2022.11.134)
18. Fang W, Shen B, Pan A, Zou L, Song B. A cooperative stochastic configuration network based on differential evolutionary sparrow search algorithm for prediction. *Syst Sci Control Eng*. 2024;12(1):2314481. doi:[10.1080/21642583.2024.2314481](https://doi.org/10.1080/21642583.2024.2314481)
19. Qiu S, Zhao H, Jiang N, et al. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: state-of-the-art and research challenges. *Inform Fusion*. 2022;80:241-265. doi:[10.1016/j.inffus.2021.11.006](https://doi.org/10.1016/j.inffus.2021.11.006)
20. Ye M, Yan X, Jiang D, Xiang L, Chen N. MIFDELN: a multi-sensor information fusion deep ensemble learning network for diagnosing bearing faults in noisy scenarios. *Knowl Based Syst*. 2024;284:111294. doi:[10.1016/j.knosys.2023.111294](https://doi.org/10.1016/j.knosys.2023.111294)
21. Wang J, He QP. Multivariate statistical process monitoring based on statistics pattern analysis. *Ind Eng Chem Res*. 2010;49(17):7858-7869. doi:[10.1021/ie901911p](https://doi.org/10.1021/ie901911p)
22. Wang Y, Si Y, Huang B, Lou Z. Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. *Can J Chem Eng*. 2018;96(10):2073-2085. doi:[10.1002/cjce.23249](https://doi.org/10.1002/cjce.23249)
23. Shang L, Liu J, Zhang Y. Recursive fault detection and identification for time-varying processes. *Ind Eng Chem Res*. 2016;55(46):12149-12160. doi:[10.1021/acs.iecr.6b02653](https://doi.org/10.1021/acs.iecr.6b02653)
24. Elshenawy LM, Mahmoud TA. Fault diagnosis of time-varying processes using modified reconstruction-based contributions. *J Process Control*. 2018;70:12-23. doi:[10.1016/j.jprocont.2018.07.017](https://doi.org/10.1016/j.jprocont.2018.07.017)
25. Zhang Q, Yang LT, Chen Z, Li P. Incremental deep computation model for wireless big data feature learning. *IEEE Trans Big Data*. 2019;6(2):248-257. doi:[10.1109/TBDDATA.2019.2903092](https://doi.org/10.1109/TBDDATA.2019.2903092)
26. Wu Y, Liu D, Yuan X, Wang Y. A just-in-time fine-tuning framework for deep learning of SAE in adaptive data-driven modeling of time-varying industrial processes. *IEEE Sens J*. 2020;21(3):3497-3505. doi:[10.1109/JSEN.2020.3025805](https://doi.org/10.1109/JSEN.2020.3025805)
27. Zhang S, Zhao C. Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly. *IEEE Trans Ind Electron*. 2018;66(5):3773-3783. doi:[10.1109/TIE.2018.2853603](https://doi.org/10.1109/TIE.2018.2853603)
28. Naruei I, Keynia F, Sabbagh Molahosseini A. Hunter–prey optimization: algorithm and applications. *Soft Comput*. 2022;26(3):1279-1314. doi:[10.1007/s00500-021-06401-0](https://doi.org/10.1007/s00500-021-06401-0)
29. Liu K, Zhao X, Mou M, Hui Y. Siamese DeNPE network framework for fault detection of batch process. *Can J Chem Eng*. 2024;102(3):1167-1187. doi:[10.1002/cjce.25102](https://doi.org/10.1002/cjce.25102)
30. Birol G, Ündey C, Cinar A. A modular simulation package for fed-batch fermentation: penicillin production. *Comput Chem Eng*. 2002;26(11):1553-1565. doi:[10.1016/S0098-1354\(02\)00127-8](https://doi.org/10.1016/S0098-1354(02)00127-8)
31. Goldrick S, Duran-Villalobos CA, Jankauskas K, Lovett D, Farid SS, Lennox B. Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process. *Comput Chem Eng*. 2019;130:106471. doi:[10.1016/j.compchemeng.2019.05.037](https://doi.org/10.1016/j.compchemeng.2019.05.037)
32. Ding SX, Yin S, Peng K, Hao H, Shen B. A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill. *IEEE Trans Industr Inform*. 2012;9(4):2239-2247. doi:[10.1109/TII.2012.2214394](https://doi.org/10.1109/TII.2012.2214394)

How to cite this article: Liu K, Zhao X, Hui Y, Jiang H. An adaptive strategy for time-varying batch process fault prediction based on stochastic configuration network. *Journal of Chemometrics*. 2024;e3555. doi:[10.1002/cem.3555](https://doi.org/10.1002/cem.3555)

PAPER

A generalized zero-shot semantic learning model for batch process fault diagnosis

To cite this article: Kai Liu *et al* 2025 *Meas. Sci. Technol.* **36** 016228

View the [article online](#) for updates and enhancements.

You may also like

- [Fine-grained fault diagnosis of photovoltaic systems based on DMAD-GAN and IFD-FGIF](#)
Juanjuan Qiao, Yongsheng Qi, Liqiang Liu et al.
- [Detection method for microleakage acoustic wave signals in natural gas pipelines based on chaotic oscillators](#)
Hongyu Chen, Kang Yi, Changxin Cai et al.
- [An intelligent compound fault diagnosis method using generalized zero-shot model of bearing](#)
Jian Cen, Bichuang Zhao, Xi Liu et al.

 The Electrochemical Society
Advancing solid state & electrochemical science & technology

UNITED THROUGH SCIENCE & TECHNOLOGY

248th ECS Meeting

Chicago, IL
October 12-16, 2025
Hilton Chicago



Science + Technology + YOU!

SUBMIT ABSTRACTS by March 28, 2025

[SUBMIT NOW](#)

A generalized zero-shot semantic learning model for batch process fault diagnosis

Kai Liu^{1,2}, Xiaoqiang Zhao^{1,2,3,*} , Miao Mou^{1,2} and Yongyong Hui^{1,2,3} 

¹ College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China

² Gansu Key Laboratory of Advanced Control for Industrial Processes, Lanzhou, People's Republic of China

³ National Experimental Teaching Centre of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China

E-mail: xqzhao@lut.edu.cn

Received 23 July 2024, revised 9 November 2024

Accepted for publication 3 December 2024

Published 13 December 2024



Abstract

In industrial monitoring, although zero-shot learning successfully solves the problem of diagnosing unseen faults, it is difficult to diagnose both unseen and seen faults. Motivated by this, we propose a generalized zero-shot semantic learning fault diagnosis model for batch processes called joint low-rank manifold distributional semantic embedding and multimodal variational autoencoder (mVAE). Firstly, joint low-rank representation and manifold learning makes the training samples map to the low-rank space, which obtains the global–local features of the samples while reducing the redundancy in the inputs for the training model; secondly, the bias of human-defined semantic attributes is corrected by predicting the attribute error rate; then, fault samples and corrected semantic vectors are embedded into the consistency space, in which the samples are reconstructed using the mVAE to fully integrate the cross-modal information, meanwhile, Barlow matrix is designed to measure the consistency between the fault samples and the attribute vectors, the higher the consistency, the higher the learning efficiency of attribute classifiers; finally, the generalized zero-shot fault diagnosis experiments are designed and conducted on the penicillin fermentation process and the semiconductor etching process to validate the effectiveness, the results show that the proposed model is indeed possible to diagnose target faults without their samples.

Keywords: fault diagnosis, batch process, generalized zero-shot, semantic correction, variational autoencoder, distributed semantic embedding

1. Introduction

As a significant method of industrial production, batch processing is widely employed in microbial fermentation, semiconductor etching, and the chemical industry due to its versatility and convenience in using the same equipment to produce various high-value products, without the need for dedicated

continuous processes. To satisfy the demands for high-quality and high added-value products, the modern batch process is more complex and flexible, once the system fault to produce incalculable losses, batch process safety and production quality monitoring are widely recognized by research scholars and engineers [1–3].

For fault diagnosis problems, traditional multivariate statistics are effectively applied [4–6]. However, it is greatly limited due to the assumption that the data satisfy the ideal distribution of effectively dealing with the nonstationary and

* Author to whom any correspondence should be addressed.

nonlinear characteristics of the process data. Such methods as Bayesian-based methods [7], support vector machines [8] and Gaussian mixture models [9] have been widely studied in process diagnostics. However, the limitation of such methods that they are based on the premise that valid features are assumed to be available. It is difficult to extract features directly from the process signals to accurately capture the effective features of the faults without being familiar with the feature selection techniques, as the effective features are usually predefined and manually selected for the specific fault diagnosis problem. In recent years, deep learning methods have achieved vigorous development in the field of industrial process fault diagnosis. Chen *et al* [10] proposed a one-dimensional convolutional autoencoder deep learning model which was used for fault detection and diagnosis of the Tennessee Eastman process and the refill batch fermentation penicillin process. Guo *et al* [11] proposed a joint approach that combined a coupled convolutional neural network (CNN) and dynamic time warping for fault detection and diagnosis for semi-batch crystallization process. Wang *et al* [12] proposed a fusion method of long short-term memory (LSTM) and CNN models for fault diagnosis, LSTM and CNN extracted the features and fusion the features extracted respectively, and utilized multilayer perceptron for feature compression and extraction and obtaining diagnosis results of the chemical process. Ji *et al* [13] constructed a novel differential recurrent neural network by embedding differential operations in LSTM neural network, and it monitored and controlled the industrial-scale penicillin fermentation process (PFP). Zhang *et al* [14] proposed a gated recurrent unit-enhanced deep CNN model for fault detection and diagnosis in chemical processes. However, these methods need to obtain enough fault samples through industrial sensors, and in real scenes and practical applications, they often suffer from sampling difficulties. These difficulties included long data collection time, large investment, and not allowing many faults in the actual production process. These problems limit the applications of supervised deep learning fault diagnosis methods. Therefore, many scholars have tried to use transfer learning method to overcome the sampling difficulties. Transfer learning learns empirical knowledge from historical data or other faults in the same process for difficulties in capturing faults and builds cross-domain models for fault diagnosis. Wang *et al* [15] proposed a linear discriminant analysis for chemical process fault classification based on weighted maximum mean difference designed for domain adaptation, the fault diagnosis effectiveness was improved in the Tennessee Eastman process and the actual hydrocracking process. Zabin *et al* [16] proposed a hybrid deep transfer learning architecture that consisted of deep CNNs and long and short-term memory layers for extracting spatiotemporal features of two-dimensional images of process data enhanced by the Hilbert transform for industrial machinery fault diagnosis. Chen *et al* [17] proposed a deep parameter less cosine network with PA Swish, which could help to adjust the network weights of domain specific features and domain invariant features by constructing an attention module based on cosine adjustment, further a reconstruction-based domain adaptive approach was used to

realize cross domain fault diagnosis. Ding *et al* [18] proposed a new deep imbalance domain adaptive fault diagnosis framework, and used it for bearing fault diagnosis. Liu and Ren [19] proposed a generalized transfer framework with evolving ability for solving the problem of lack of fault samples to improve fault diagnosis performance.

However, the transfer learning method solves the problem of domain transfer between source and target domains for the same type of faults. With further advancement of intelligent manufacturing, the complex advanced systems would be more diversified along with the types of faults, which systems are hard to obtain for some special fault samples, and there are many unknown faults. When the target faults have no samples for training, such transfer learning methods are no more applicable.

In fact, the diagnosis of fault patterns in those unlabeled samples is called zero-shot fault diagnosis [20, 21]. Zero-shot learning aims to solve the problem that there are no test class samples in the training phase, attempts to transfer knowledge from known class samples to unknown class samples through a shared semantic space. Lampert *et al* [22] proposed a classical zero-sample learning method called direct attribute prediction (DAP), which did not directly infer that test sample label, but tried to study the relationship between the input data and semantic properties. Lampert *et al* [23] also proposed indirect attribute prediction method for zero-shot learning. Inspired by DAP, Feng and Zhao [24] proposed an attribute transfer method based on fault description to solve the zero-shot fault diagnosis problem, which introduced the idea of zero-shot learning into the industrial field. Hu *et al* [25] proposed semantically consistent embedding (SCE) approach for industrial zero-shot fault diagnosis, which inserted fault samples and their human-specified attribute vectors into a semantically consistent space, and then reconstructed the fault samples from that space. It attempted to classify unseen classes of errors by using only seen classes of errors for training. Chen *et al* [26] proposed a zero-sample industrial process fault diagnosis model based on explicit and implicit joint attribute transfer, which utilized partially known fault samples to identify unknown faults. Chen *et al* [27] proposed a pyramidal ZSL (PZSL) model with multi-granularity hierarchical attributes to deal with the inability to explore hierarchical features between attributes, which utilized a hierarchical constraint network to predict attributes layer by layer. However, ZSL fault diagnosis method can only categorize unknown faults of the training process in the diagnosis stage. It cannot categorize cases where both seen and unseen faults are possible, which severely limits their practical application. To solve the above problems, Huang *et al* [28] proposed a generalized zero-sample learning (GZSL), which could classify seen faults and unseen faults during the online fault diagnosis stage. There has been relatively little research on GZSL in the field of fault diagnosis, and mainly in the field of rolling bearing fault diagnosis [29–31].

Regarding these issues, in this paper, we select the typical batch processes PFP and semiconductor etching process (SEP) as the research objects, and propose a generalized zero-shot fault diagnosis model for batch processes called joint low-rank

manifold distributional semantic embedding and multimodal variational autoencoder (JLMDSE-mVAE). The main contributions of this paper are as follows:

- (1) The proposed low-rank manifold structure (LRM) effectively extracts the global–local structural information of process data, thus reducing the input computation complexity of the model and improving the training efficiency of the model.
- (2) The human-defined semantic attribute information is affected by the labeler’s subjectivity with semantic bias, which is achieved by the attribute prediction error rate to correct the original semantic description information and improve the accuracy of model classification.
- (3) In generalized zero-shot classification, fault samples and attribute vectors are used as cross-modal data, and the mVAE structure is used to embed the cross-modal data into a semantically consistent space, which is fused and reconstructed in the output. The consistent space can fully reflect the cross-modal information, meanwhile, the more consistent the cross-modal embedding is between them, better the learning efficiency of the attribute classifiers.

The rest of this paper is organized as follows. The fault attribute descriptions and formulation of generalized zero-shot fault diagnosis are illustrated in section 2. The proposed JLMDSE and mVAE batch process generalized zero-shot fault diagnosis method is shown in section 3. The experiments of the proposed method based on the PFP and the SEP are validated in section 4. And section 5 summarizes this paper.

2. Problem formulation of generalized zero-shot fault diagnosis

In this section, we first introduce the attribute descriptions of industrial faults and then formulate the problem of generalized zero-shot fault diagnosis.

2.1. Fault attribute descriptions

Fault attributes are expert knowledge that plays an important role in zero-shot fault diagnosis. That is like semantic information in computer vision, which usually describes the fault classification with a word or short phrase. Every fault is described by several fault attributes, which include the location where the fault occurred, the reason for generating the fault, and the fault variables that are affected, etc.

The attributes of each fault form the attribute vector $a \in R^C$, where C is the number of attributes, each fault can be uniquely represented by the attribute vector a . For m kinds of faults, we get m attribute vectors to construct the fault description matrix $A \in R^{m \times C}$. The matrix A contains either 1 or 0 for each element, where 1 indicates that the fault has the attribute description and 0 indicates that the attribute is not satisfied. Through the fault description matrix A , all the faults can be accurately depicted.

2.2. Formulation of generalized zero-shot fault diagnosis

In traditional industrial process fault diagnosis, it is often assumed that all classes of fault samples are known when the model is trained. In other words, all the samples that need to be diagnosed in the test set are known in the training set. However, in industrial processes, few processes are allowed to generate all fault types, and some fault types exist only for specific systems, equipment, and variables. If these fault types do not occur, then there are no data samples of such faults, which makes it impossible to obtain labeled samples of these fault types during the fault diagnosis process. These fault types for which labeled samples are not available are named unseen faults, while other fault types for which labeled samples are available are named seen faults.

For clarity, we first describe the zero-shot fault diagnosis task. The implication of zero-shot fault diagnosis is that unseen faults are diagnosed only if seen faults are known. Our goal is to diagnose and recognize p faults, with a target fault set of $T = \{t_1, t_2, \dots, t_p\}$. There are no fault samples in T for training. The set of seen types of faults used for training is: $S = \{s_1, s_2, \dots, s_q\}$, where q is the number of seen faults. In zero-shot fault diagnosis, T and S are mutually disjoint, that is, $T \cap S = \emptyset$. Zero-shot fault diagnosis assumes that the target fault set contains only unseen faults. For industrial processes, the target set contains not only unseen faults but also seen faults. For generalized zero-shot fault diagnosis, our goal is to diagnose m unseen faults and n seen faults with the target fault set of $T_g = \{t_1, \dots, t_m, s_k, \dots, s_{k+n}\}$. Then we have $T_g \cap S = S_K \neq \emptyset$, where $S_K = \{s_k, \dots, s_{k+n}\}$, S_K is fault type of the target fault set involved in training of the model. The samples of the training set S are denoted: $\chi = \{X_S \in R^{N_S \times D}, Y_S \in R^{N_S}\}$, where N_S is the number of samples, and D is the feature dimension, X_S and Y_S denote the samples and labels of the training faults. The generalized zero-shot fault diagnosis learns a mapping f from S to T_g , the objective function is derived from formula (1) as follows,

$$\min \text{CLoss} \left(Y_{T_g}, \hat{Y}_{T_g} \right), \text{ and } \hat{Y}_{T_g} = f(X_S, Y_S, A | X_{T_g}) \quad (1)$$

where CLoss denotes the classification loss, X_{T_g} and Y_{T_g} denote the samples and labels of the training faults, respectively, $A = [A_S, A_{T_g}] \in R^{L \times C}$ is the attribute description matrix, and $L = m + n + q$. Noteworthy, the attribute description matrices A_S for S and A_{T_g} for T_g are both used for model training.

Since both seen and unseen faults are present in the need diagnosed faults. Therefore, this paper attempts to establish a more generalized zero-shot fault diagnosis method, which is possible to diagnose both seen and unseen faults in the presence of only seen fault training models. A diagram of generalized zero-shot fault diagnosis as shown in figure 1, in which ‘ASD#’ represents the attribute of the fault, ‘√’ means with the attribute, ‘×’ means without the attribute.

3. Methodology

In this section, we introduce the motivation of this paper, the proposed JLMDSE-VAE method is then explained in details.

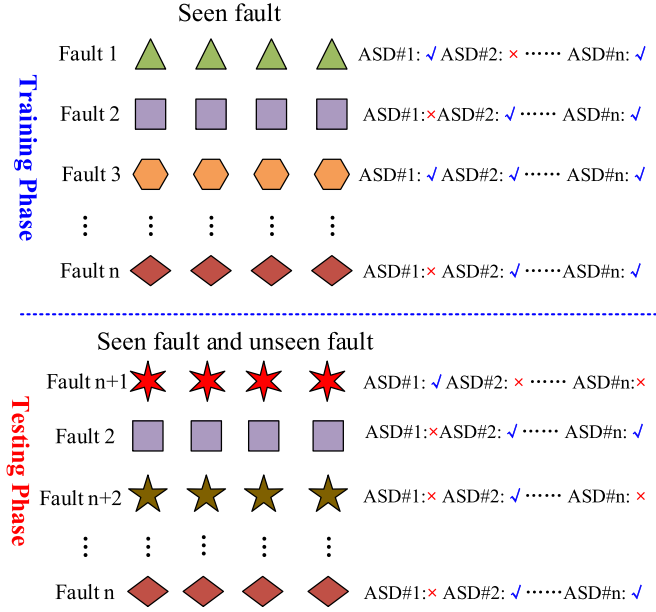


Figure 1. Illustration of generalized zero-shot fault diagnosis.

3.1. Motivation

As mentioned above, each fault can be described with specific attribute vector. Yet, the labeled of attributes is often affected by the subjectivity of the labeler, which is called semantic bias. Therefore, we accomplish semantic correction of the semantic description information of seen faults through attribute prediction error rate method, which reduces the influence of human subjectivity on the results and improves the model classification accuracy. Further, during sample training, more redundant features in the training samples lead to the complexity in the training inputs while reducing the training efficiency of the model. In fault classification with generalized zero-shot containing both unseen and seen faults, fault samples and attribute vectors can affect the accuracy of fault diagnosis as cross-modal data. The cross-modal data embedded into a semantically consistent space for fusion and reconstruction can fully reflect the cross-modal information. Meanwhile, the more consistent the cross-modal embedding between them, the higher the learning efficiency of the attribute classifier.

In the next subsection, the proposed JLMDSE-mVAE model is introduced to handle the above-mentioned considerations.

3.2. The proposed JLMDSE-mVAE model

The structure of the proposed JLMDSE-mVAE model in this paper is illustrated in figure 2. The model contains three main parts: (1) LRM (2) semantic revision structure; (3) mVAE with one encoder and two decoders. Specifically, original seen fault samples obtain global-local features through a LRM which reduces the complexity of training samples. Human-defined semantic attribute information is revised by the prediction error rate, which reduces the semantic bias influenced from

the labeler's subjectivity. The mVAE model consists of one encoder and two decoders, whose purpose is to align two different modes of sample features and semantic attributes in the latent space.

3.2.1. Construct low-rank manifold space. Low rank representation (LRR) [32] can accurately describe the original space of data by capturing the lowest rank of data. LRR can solve the problem of dimensionality catastrophe due to the superposition of repetitive information in the data. Also, it can well reveal the global structural information of the data in terms of spatial distribution. To further preserve the local structural information between the original feature space and the approximate feature space, it is regularized by means of local manifold learning. We use neighborhood preserving embedding (NPE) with regularized low-rank representation to construct a low-rank manifold space.

(1) LRR:

Given dataset $X = (x_1, x_2, \dots, x_n)$, $X \in R^{m \times n}$, where m denotes the sample dimension and n denotes the number of samples, so $X = D + E$, where $D, E \in R^{m \times n}$, D is the low-rank representation of the dataset, E stands for residuals. The low-rank representation D can be expressed as $D = BZ$, where $B \in R^{m \times n}$ is the base of the data and $Z \in R^{m \times n}$ is the coefficient of variation matrix. The low-rank solution is shown in formula (2):

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1} \text{ s.t. } X = BZ + E \quad (2)$$

where, $\|Z\|_*$ denotes the nuclear norm of Z , $\|Z\|_{2,1}$ denotes the norm of the residual, and λ is the regularization parameter. In practice, B is always unknown, and a better option is to set $A = X$. Since Z is the self-affinity matrix of the data X , we conduct the symmetric constraint $Z = Z^T$. Furthermore, formula (2) could be transformed into formula (3):

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1} \text{ s.t. } X = XZ + E, Z = Z^T \quad (3)$$

(2) NPE:

NPE [33] represents the local linear structure of manifold by local linear reconstruction, which is in the form of mean square error. The embedding cost function of NPE can be indicated as shown in formula (4):

$$\phi(Y) = \min \sum_{i=1}^n \|y_i - \sum_{j \in Q(i)} w_{ij} y_j\|^2 \quad (4)$$

where, $Q(i)$ is the set of neighboring samples; $y_j (y_j) \in R^{D \times P}$ is local manifold features extracted from X , D is the dimension of the feature vector, $W \in R^{P \times P}$ should satisfy the normalization constraint.

The modeling for formula (4) can be transformed into the form of matrix as shown in formula (5):

$$\Phi(Y) = \text{Tr}(Y(I - W)^T(I - W)Y^T) \quad (5)$$

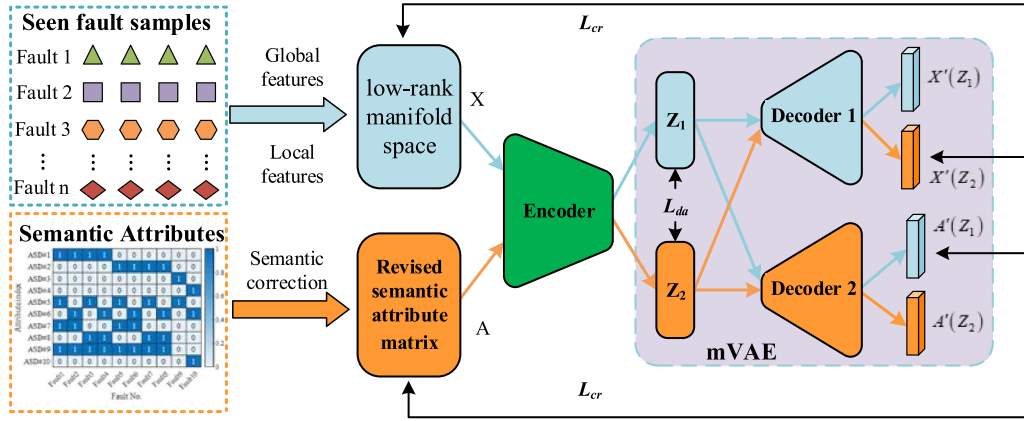


Figure 2. Structure diagram for the proposed JLMDSE-mVAE algorithm.

where, $Y \in R^{D \times P}$ is the matrix form of the local manifold features, $I = \text{diag}(1, \dots, 1)$.

(3) LRM:

LRM is obtained by NPE regularization of LRR, and data samples are obtained global–local features in low-rank manifold space. LRM can be modeled by formula (6):

$$\begin{aligned} \min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1} + \beta \text{Tr} \left(Z(I - W)^T (I - W) Z^T \right) \\ \text{s.t. } X = XZ + E, Z = Z^T \end{aligned} \quad (6)$$

where β is a balance parameter.

3.2.2. Semantic information correction. For the problem of excessive human-defined subjectivity of semantic description information, an attribute prediction error rate is utilized to realize semantic correction of semantic description information of seen class faults and to reduce the influence of human subjectivity on the results and improve the classification accuracy of the model. We calculate the attribute prediction error rate by inputting the training data into each trained attribute classifier and comparing the output with the labeled data of each faulty attribute. The attribute error rate matrix is shown in formula (7):

$$E = \begin{bmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,j-1} & e_{1,j} \\ e_{2,1} & e_{2,2} & \dots & e_{2,j-1} & e_{2,j} \\ \dots & \dots & \dots & \dots & \dots \\ e_{i-1,1} & e_{i-1,2} & \dots & e_{i-1,j-1} & e_{i-1,j} \\ e_{i,1} & e_{i,2} & \dots & e_{i,j-1} & e_{i,j} \end{bmatrix} \quad (7)$$

where, $e_{i,j}$ represents the attribute error rate corresponding to the j th attribute of the i th fault. If $e_{i,j} > \eta$, the corresponding attribute is changed, and if η is 0.9, it indicates that the attribute is changed when the prediction error rate of the attribute is more than 90%. The optimum value of η is further discussed in the following section.

3.2.3. mVAE. For the JLMDSE-mVAE model proposed in this paper, mVAE consists of one encoder and two decoders whose purpose is to align data from two different modalities, that is, sample data and semantic attributes, in the latent space. The encoder consists of three fully connected layers. Specifically, the first layer has 256 neurons with ReLU activation, the second layer has 128 neurons with ReLU activation, and the final layer outputs the latent vector with 64 neurons using a linear activation function. This structure effectively captures essential features for embedding into a lower-dimensional latent space. Both decoders use a similar three-layer structure, with 128 and 256 neurons in the first two layers (ReLU activation), followed by an output layer with sigmoid activation to maintain output consistency with the input data's range. This dual-decoder approach enables cross-modal data alignment within a unified semantic space. Once the original sample data are mapped into the low-rank manifold space, the encoder converts the low-rank manifold features and modified semantic attributes into low-dimensional potential vectors Z_1 and Z_2 , respectively. Z_1 and Z_2 are subsequently decoded into the feature space to be reconstructed with decoder 1 and decoder 2. Therefore, we give the functional expression of mVAE as shown in formula (8):

$$\begin{aligned} L_{\text{mVAE}} &= L_{\text{VAE}}^X + L_{\text{VAE}}^A \\ &= \sum_{i=1}^m E_{q\phi(z|x^{(i)})} \left[\log p_{\theta}(x^{(i)}|z) \right] \\ &\quad - D_{\text{KL}}(q\phi(z|x^{(i)}) \| p_{\theta}(z)) \end{aligned} \quad (8)$$

where, L_{VAE}^X and L_{VAE}^A denote the VAE loss for sample modality and semantic modality, respectively. $m = 2$ denotes two modalities. Specifically, $x^{(1)} \in X'$ and $x^{(2)} \in A'$. The first term of the loss function formula (8) attempts to search for the true conditional probability distribution $p_{\theta}(x^{(i)}|z)$ on the variable z . The second term measures the Kullback–Leibler divergence between $q\phi(z|x^{(i)})$ and $p_{\theta}(z)$.

To make the proposed model further capture similar representations between modalities, we attempt to decode the latent embeddings of samples which belong to another modality though from the same category. Thus, each modality specific decoder can be trained on potential vectors from different modalities. The cross-reconstruction loss between sample and semantics is shown in formula (9):

$$L_{cr} = |X_i' - D_1(E(A_i'))| + |A_i' - D_2(E(X_i'))| \quad (9)$$

where, E is a shared encoder of two different modalities, D_1 and D_2 are the decoders of visual features and semantics, respectively.

In general, we distribute aligned sample features and semantic attribute features by minimizing the Wasserstein distance between the latent multivariate Gaussian distributions of the two-modal data, and a loss function as shown in formula (10):

$$L_{wda} = \sum_i^M \sum_{j \neq i}^M \left(\|\mu_i - \mu_j\|_2^2 + \left\| \sum_i^{1/2} - \sum_j^{1/2} \right\|_F^2 \right)^{\frac{1}{2}} \quad (10)$$

where, μ and \sum are the distributional parameters of the model, which denote the mean and variance. That satisfies $Z \sim N\left(\mu, \sum\right)$, $\|\bullet\|_F$ is the F -norm. Due to the higher consistency between cross-modal embeddings, the better the learning performance of the attribute classifiers. To align the sample and attribute embeddings to measure the consistency of cross-modal embeddings simultaneously, the design of specific Barlow matrix as shown in formula (11):

$$B_{ij} = \frac{\sum_b z_{b,i}^X z_{b,j}^A}{\sqrt{\sum_b (z_{b,i}^X)^2 \sum_b (z_{b,j}^A)^2}} \quad (11)$$

where, b denotes the batch sample, B_{ij} denotes the consistency of the i th dimension of the sample embedding with the j th dimension of the attribute embedding, which ranges $(-1, 1)$, that is, ‘-1’ means perfect negative correlation and ‘1’ is perfect correlation. The consistency alignment loss function is defined as shown in formula (12):

$$L_{da} = \sum_{i=1}^n (1 - B_{ii})^2 + \chi \sum_{i=1}^n \sum_{j \neq i}^n B_{ij}^2 \quad (12)$$

where, χ is the tradeoff parameter. For the loss function formula (8), the first term denotes the semantic consistency of the cross-modal embedding and the second term denotes the redundancy information between the different elements.

The proposed JLMDSE-mVAE can be indicated by formula (13):

$$L_{JLMDSE-mVAE} = L_{mVAE} + \alpha L_{cr} + \beta L_{da} \quad (13)$$

where, α and β are the tradeoff parameters for the cross-reconstruction loss function and the consistency alignment embedding function, respectively.

3.2.4. The JLMDSE-mVAE model for fault diagnosis. The fault diagnosis procedures for the JLMDSE-mVAE model are shown in figure 3, which are divided into training and testing two parts.

In the training phase, the seen fault training samples are mapped into the low-rank space to obtain low-rank manifold features with global-local. The raw semantic information is corrected by the prediction error rate, which reduces the human factor influence. Furthermore, the fault sample features and the revised semantic matrix are mapped into the latent space utilizing one encoder. The semantic latent features and seen fault latent features are decoded and reconstructed by using two decoders to train the complete mVAE model. The seen fault samples are passed through the trained mVAE model to obtain fault embeddings, which are obtained to train the attribute classifier.

The trained classifier gives the estimated value of $p(a_i|z)$ and completes the representation of the fault to the attribute layer, as shown in formula (14):

$$p(a|z) = \prod_{i=1}^c p(a_i|z) = \prod_{i=1}^c a_i(z) \quad (14)$$

where, a_i represents the i th classifier, z represents the latent variable of the fault sample.

By using Bayes rule, label inference is performed. The reasoning from attribute to class is as shown in formula (15):

$$p(y|a) = \sum_{y=1}^m \frac{p(y)}{p(a^y)} p(a^y|y) \quad (15)$$

where, a^y represents the attribute vector of the y th class, and m represents the number of class of faults. $p(a|y) = [a = a^y]$, if $a = a^y$, $[a = a^y] = 1$ and otherwise $[a = a^y] = 0$.

In the testing phase, the test sample includes both seen faults and unseen faults. The test samples are mapped according to the LRM from the training phase to obtain fault features. The latent features of the test samples are further obtained through the encoder input to the attribute classifier. Since all the attributes of the unseen occur in the seen, we can get the classification of the faults by using the maximum posteriori probability.

A posteriori probability of the sample fault class is shown in formula (16):

$$p(y|z) = \sum_a p(y|a) p(a|z) = \frac{p(y)}{p(a^y)} \prod_{j=1}^m p(a_j^y|z) \quad (16)$$

Using the maximum a posterior-maximum estimation to give the best matching class from all test classes is shown in formula (17):

$$y = \operatorname{argmax} \prod_{j=1}^m \frac{p(a_j^y|z)}{p(a_j^y)} \quad (17)$$

where, y denotes the test result output. The fault with the highest posterior probability is taken as the prediction result.

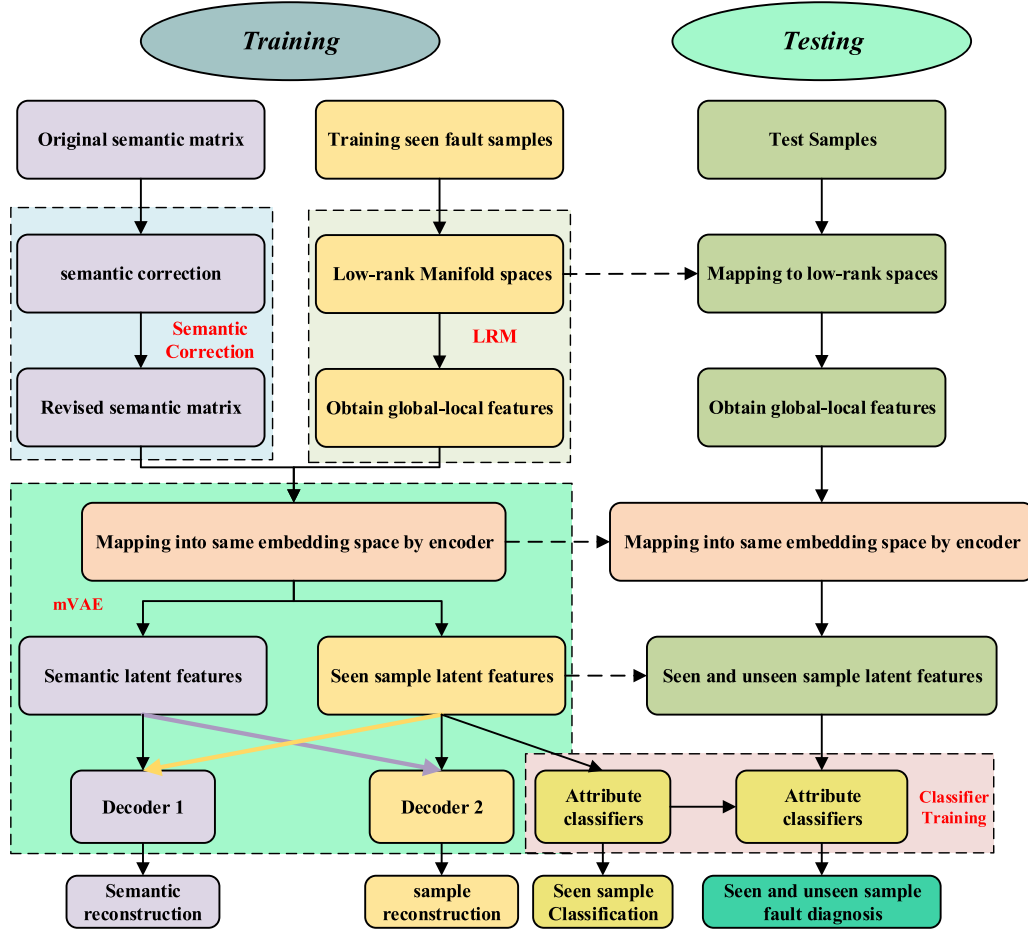


Figure 3. Procedures of JLMDSE-mVAE generalized zero-shot diagnosis.

In this paper, the performance of the proposed generalized zero-shot fault diagnosis model is evaluated by using the harmonic mean, which is widely used in classification tasks, as shown in the formula (18):

$$H = 2 \times \frac{\text{Acc}_s \times \text{Acc}_u}{\text{Acc}_s + \text{Acc}_u} \quad (18)$$

where, Acc_s and Acc_u denote the fault diagnosis accuracies for seen and unseen fault, respectively. For generalized zero-shot fault diagnosis whole accuracy is shown in formula (19):

$$A = \frac{m}{m+n} \text{Acc}_s + \frac{n}{m+n} \text{Acc}_u \quad (19)$$

where, m and n denote the numbers of samples with seen faults and unseen faults. Given the definition of fault diagnosis accuracy is shown in formula (20):

$$\text{Acc} = \frac{N_C}{N} \quad (20)$$

where, N_C denotes the number of samples classified accurately, N denotes the total number of test samples.

4. Case studies

In this section, we evaluate the performance of the proposed model for zero-sample fault diagnosis and generalized zero-sample fault diagnosis by two typical batch processes: PFP and SEP, and demonstrate the validity and superiority of the proposed model.

4.1. PFP

Penicillin is an extensively used antibiotic with high clinical medical value, its production process is a complex biochemical reaction and typical batch production process. For this case, the PFP data have been obtained from the pensim2.0 platform designed by the Illinois Institute of Technology [34]. A schematic diagram of the PFP is shown in figure 4. The fermentation system mainly contains fermentation tanks, agitation motors, aeration equipment and other major equipment, in addition to the addition of acid, base, hot water, cold water, substrate and other reactant's part, and is equipped with the corresponding temperature and PH controller. PFP utilizes specific production bacteria to grow and multiply under certain conditions, and after the concentration of the produced

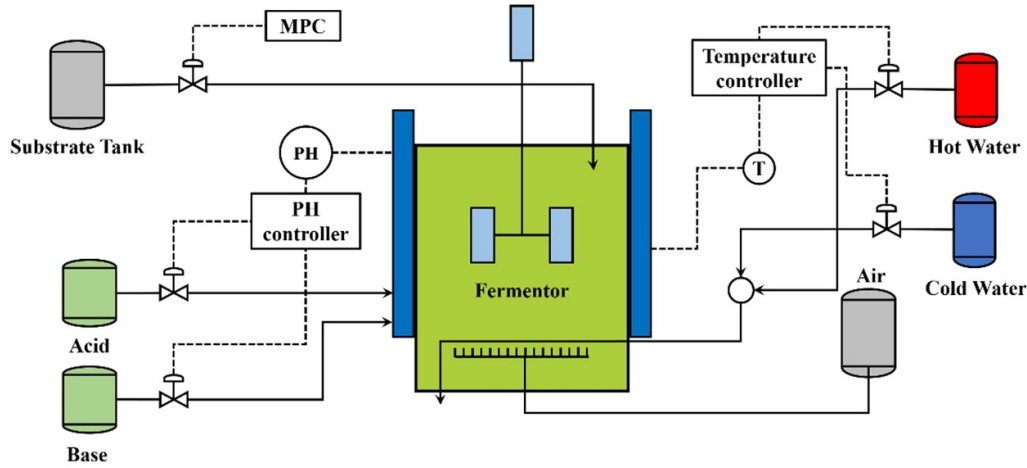


Figure 4. Schematic of the penicillin fermentation process.

Table 1. Penicillin fermentation process fault information.

No.	Fault state information
1	+3% step type, aeration rate fault
2	−2% step type, aeration rate fault
3	+5 l h ^{−1} ramp type, aeration rate fault
4	−3 l h ^{−1} ramp type, aeration rate fault
5	+5% step type, agitator power fault
6	−2% step type, agitator power fault
7	+3 W ramp type, agitator power fault
8	−5 W ramp type, agitator power fault
9	+6 K substrate feed temperature fault
10	−0.3 mmole L ^{−1} CO ₂ conc. Fault

Table 2. Fault attribute semantic description for PFP.

No.	Fault state information
ASD#1	Aeration rate is changed
ASD#2	Agitator power is changed
ASD#3	Substrate feed temperature is changed
ASD#4	CO ₂ conc. is changed
ASD#5	Fault change value is positive
ASD#6	Fault change value is negative
ASD#7	Fault change in step signal
ASD#8	Fault change in ramp signal
ASD#9	Strong related with final product quality
ASD#10	Weak related with final product quality

bacteria reaches a certain concentration, penicillin begins its production as a product of metabolism, and to ensure the production of penicillin, it is necessary to constantly replenish the nutrient elements such as nitrogen and sugar. In penicillin fermentation, the condition monitoring relies on a range of sensors including pH sensors, dissolved oxygen probes, temperature sensors and carbon dioxide sensors, etc.

To validate the proposed model, we use the PFP simulation platform Pensim2.0 to generate 10 different types of faults, which are described in detail in table 1. For each fault, the reaction time is 400 h, the sampling time is 1 h, and the fault is introduced from the reaction start moment to the reaction finished moment.

The attribute semantic descriptions of the faults contain the information about the faults types as shown in table 2. Each fault is described by 10 fine-grained attributes.

The semantic description matrix of the fault attributes of the PFP is shown in figure 5, the value ‘1’ means that the fault has the current attribute, and ‘0’ means that the fault does not have the current attribute.

The 10 types of faults are divided into 7:3 ratio, of which seven are training faults and three are target faults. According to different combinations, those faults are grouped into six groups, and the specific fault groups are shown in table 3.

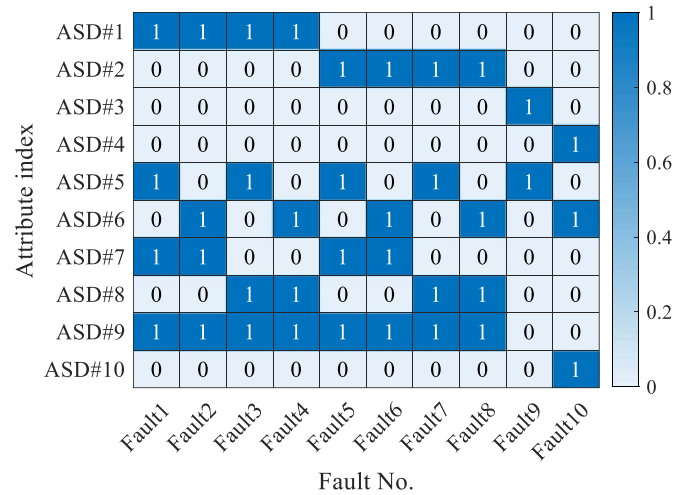
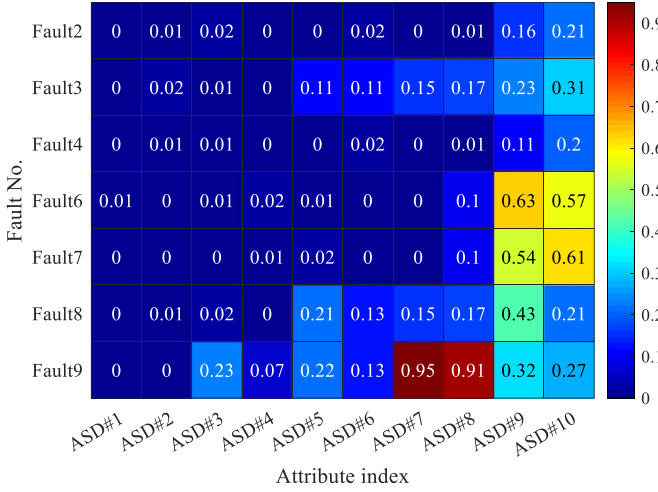


Figure 5. Attribute vectors of the PFP.

Significantly, for groups A, B and C, the training faults are all seen faults, and the target faults are unseen faults, which are used to verify the effectiveness of the zero-shot fault diagnosis. For groups D, E and F, the training faults are all seen faults and the target faults are both seen and unseen faults to verify the effectiveness of the generalized zero-shot fault diagnosis.

Table 3. Six different fault groups for PFP.

No.	Training faults	Target faults
A	2–4, 6–9	1, 5, 10
B	1–3, 5–7, 10	4, 8, 9
C	1–2, 5–7, 9–10	3, 4, 8
D	1–3, 5–7, 10	2, 6, 8
E	2, 4–7, 9–10	1, 3, 4
F	1–4, 7–8, 10	2, 5, 9

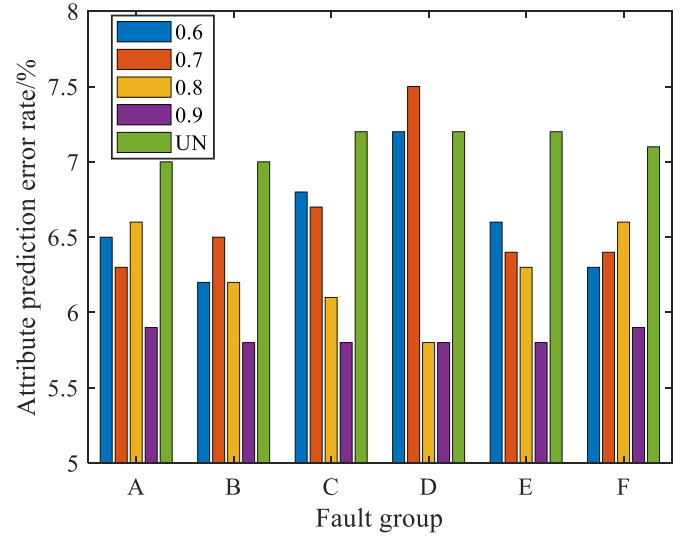
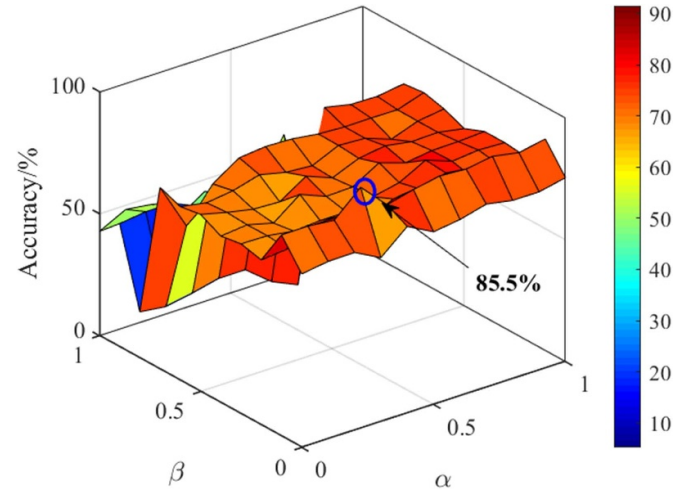
**Figure 6.** Attribute prediction error rate for group A.

The output data and the labeled data of each fault attribute are compared to calculate the attribute prediction error rate table. The attribute prediction error rate is shown in figure 6, where the column direction represents that the 7 types of seen faults are classified in group A, and the row direction represents the 10 attributes of the semantic description of the faults, respectively, where the deeper color means that the error rate of fault corresponding to the attribute is higher.

The attributes are changed when the attribute error rate is more than the threshold value. After making attribute corrections and retraining the model, the overall attribute prediction error rates resulting from attribute prediction with the training set are shown in figure 7, ‘UN’ indicates no semantic correction.

It can be seen from figure 7 that the attribute prediction error rate is minimized when $\eta = 0.9$, and the attribute prediction error rate is the same as η equals to 0.8 and 0.9 for group D data. For different groups of experiments with the change of η , its trend correlation is poor, so this paper chooses the threshold of the smallest attribute prediction error rate, i.e. $\eta = 0.9$.

The JLMDSE-mVAE loss function has two parameters that need to be determined, i.e. the balance parameters α and β . Taking the fault diagnosis accuracy as criterion, the balance parameters α and β are obtained by the grid search method, and the search range of α and β is between 0 and 1. In the case of group A, figure 8 shows the diagnostic accuracies when α

**Figure 7.** The influence of different correction thresholds η for fault diagnosis results.**Figure 8.** Fault diagnosis accuracy for different values of α and β (group A faults).

and β take different values. According to the search results, we choose $\alpha = 0.3$ and $\beta = 0.1$.

In table 4, the sample fault diagnosis accuracies of the proposed model for different fault groups are given and compared with two well advanced zero-shot fault diagnosis models, FDAT and SCE, with probabilistic naive Bayes (NB) and nonlinear random forest (RF) selected as the attribute learners. It can be seen from table 4 that the proposed model performs better than the two compared models overall, especially in fault group B. With the NB attribute learner, the zero-shot fault diagnosis accuracy of JLMDSE-mVAE is improved by 19.75% compared to FDAT and 12.25% compared to SCE.

The generalized zero-shot fault diagnosis accuracies of the three models which are given in table 5, where A denotes the overall diagnostic accuracy of seen class faults and unseen class faults, and H denotes the harmonic mean. H measures the

Table 4. Fault diagnosis results Acc for PFP (%).

Method	Classifier	A	B	C	D	E	F
FDAT	NB	75.25	65.75	68.50	66.84	66.25	67.75
	RF	72.75	64.25	67.25	65.75	65.00	67.00
SCE	NB	79.75	73.25	75.00	69.83	69.25	72.75
	RF	77.25	70.25	67.75	69.75	68.75	71.50
JLMDSE-mVAE	NB	85.50	85.50	83.25	77.17	71.50	77.25
	RF	83.75	84.25	81.50	72.50	70.25	75.00

Table 5. Generalized zero-shot fault diagnosis results *A* and *H* for PFP (%).

Method	Classifier	D		E		F	
		<i>A</i>	<i>H</i>	<i>A</i>	<i>H</i>	<i>A</i>	<i>H</i>
FDAT	NB	66.84	67.49	66.21	67.14	67.75	68.26
	RF	65.75	66.37	65.13	66.23	67.00	68.11
SCE	NB	69.66	70.47	69.13	70.43	72.75	73.61
	RF	69.75	70.63	67.66	70.33	71.50	73.01
JLMDSE-mVAE	NB	77.16	77.60	70.26	72.11	77.25	78.11
	RF	72.37	73.26	69.96	71.03	75.00	77.21

Table 6. Fault diagnosis results Acc with different loss functions (%).

Method	A	B	C	D	E	F	Mean
CA-mVAE (NB)	68.42	69.08	67.50	64.92	63.50	64.75	66.36
CR-mVAE (NB)	70.08	70.92	69.67	66.58	64.25	66.42	67.99
JLMDSE-mVAE-I (NB)	73.58	73.42	73.00	74.33	71.42	71.92	72.95
JLMDSE-mVAE-II (NB)	85.50	85.50	83.25	77.17	71.50	77.25	80.03

generalized zero-shot fault diagnosis effectiveness as it reacts to the aggregate mean of seen faults and unseen faults. It can be obtained from table 5 that the proposed model in this paper is better than FDAT and SCE for generalized zero-shot fault diagnosis.

To verify the effectiveness of the proposed loss function in zero-shot fault diagnosis in PFP, the results of the ablation experiments are given as shown in table 6. From both tables 4 and 5, the results of using NB for attribute learning are better than RF, so only NB is used for attribute learning in the ablation experiments. In table 6, CA-mVAE indicates that only the cross-reconstruction loss function is included, CR-mVAE indicates that only the consistent alignment loss function is included, and JLMDSE-mVAE-I indicates that the above two loss functions are included at the same time without the semantic information being corrected. JLMDSE-mVAE-II representation includes both above loss functions and the semantic information is corrected. As shown in table 6, both CA-mVAE and CR-mVAE models achieve high fault diagnosis accuracies when the model contains only a single loss function. When two loss functions are introduced, the fault diagnosis accuracy of the JLMDSE-mVAE-II gets significantly improved compared with CA-mVAE and CR-mVAE. JLMDSE-mVAE-II improves the mean fault diagnosis

accuracy 13.67 and 12.04 compared to CA-mVAE and CR-mVAE, respectively. This result not only proves the effectiveness of the added loss function, but also demonstrates its significant advantage in improving the fault diagnosis performance. The mean fault diagnosis accuracy of JLMDSE-mVAE-II is improved by 7.08% compared to JLMDSE-mVAE-I, which indicates that the semantic correction has a significant improvement on the generalized zero-shot fault diagnosis performance.

The Barlow matrixes of JLMDSE-mVAE on groups A and B are shown in figure 9, which reveal that the diagonal elements are close to 1 and the off-diagonal elements are close to 0. This indicates that the proposed method in this paper has high consistency of sample embedding and attribute embedding alignment, while reducing the redundant information.

Figure 10 plots the confusion matrixes of FDAT, SCE and the proposed JLMDSE-mVAE with NB as the attribute classifier.

From figure 10, the diagnosis results of FDAT, SCE and JLMDSE-mVAE for faults 2, 6 and 8 where significantly higher than random guessing (33.3%), which proves the feasibility of generalized zero-shot fault diagnosis. JLMDSE-mVAE-II (79% on fault 2, 77.5% on fault 6, and 75% on fault 8) has significantly higher diagnostic results than FDAT

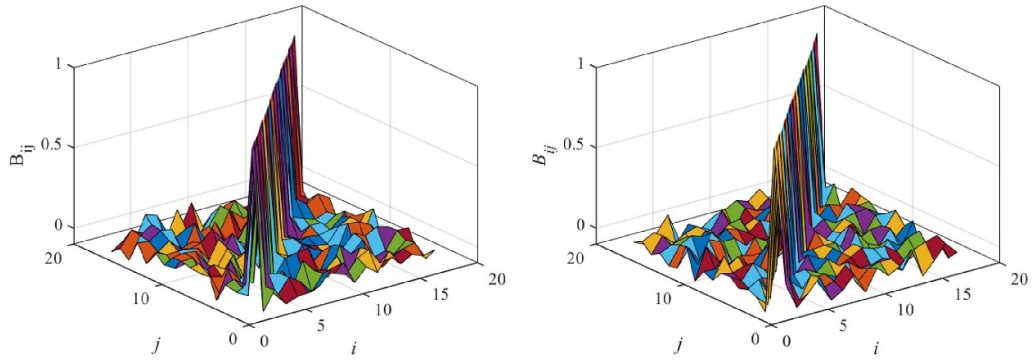


Figure 9. Visualization of the Barlow matrix on group A and B.

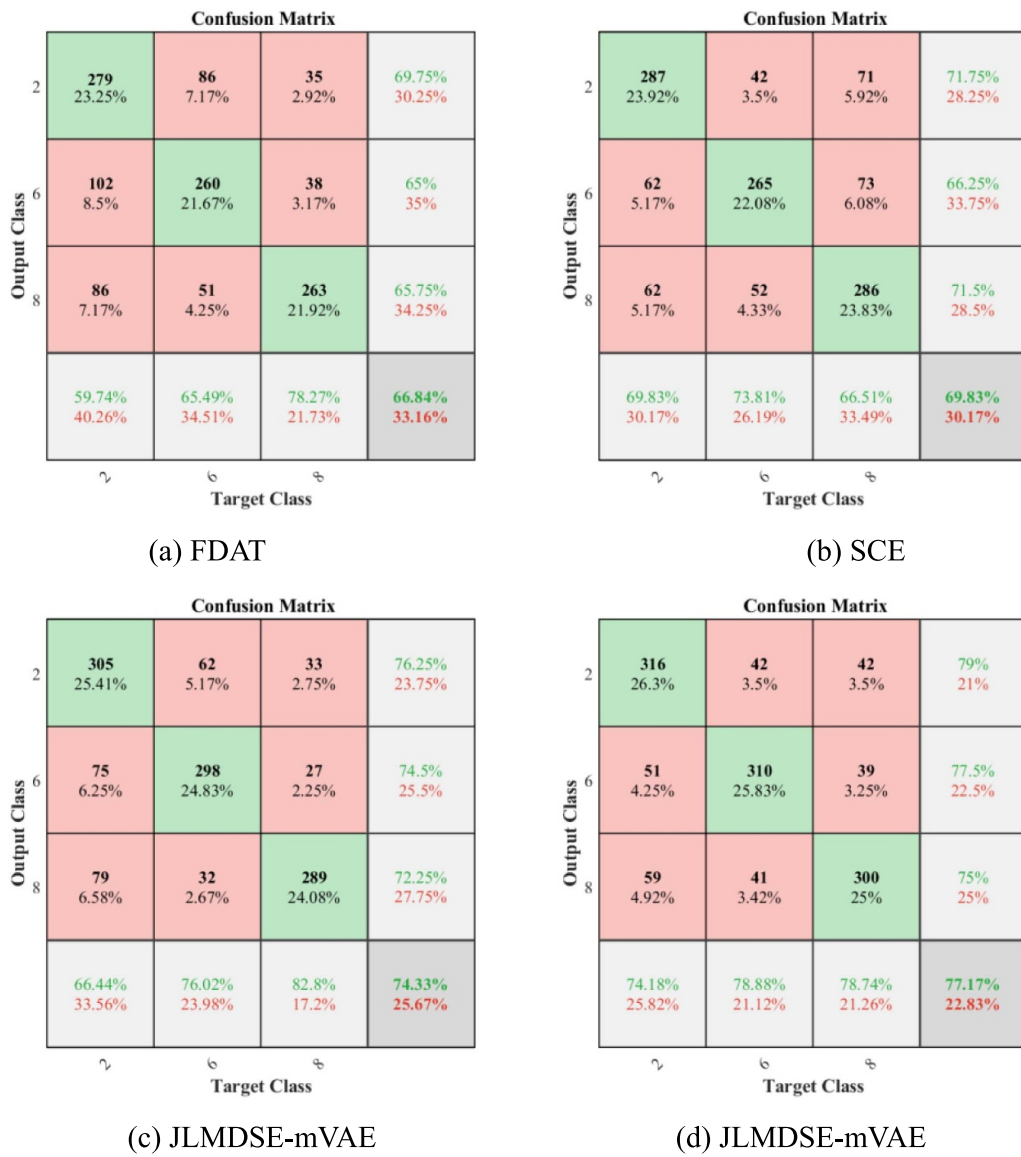


Figure 10. Confusion matrices of the groups D for different method.

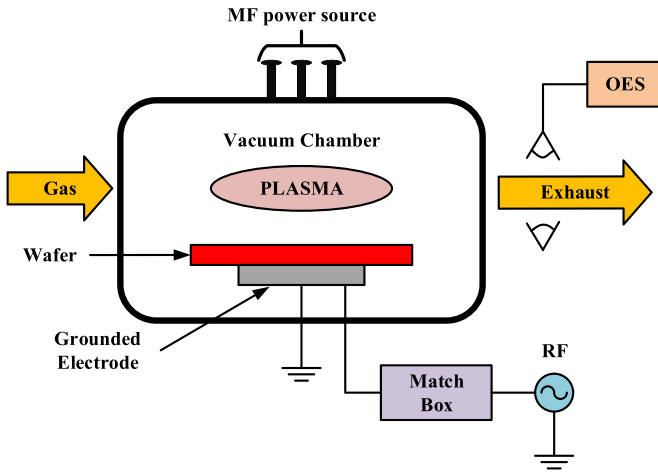


Figure 11. Schematic of the semiconductor etching process.

(69.75% on fault 2, 69.75% on fault 6 of 65%, and fault 8 of 65.75%) by 9.25%, 12.5%, and 9.25%, compared to SCE (fault 2 of 71.75%, fault 6 of 66.25%, and FAULT 8 of 71.5%) by 7.25%, 11.25%, and 3.5%, and compared to the uncorrected JLMDSSE-mVAE-I (fault 2 of 76.25%, fault 74.5% for fault 6 and 72.25% for fault 8) by 2.75%, 3% and 2.25%, respectively. It shows that the JLMDSSE-mVAE proposed in this paper can provide more critical features for fault diagnosis.

4.2. SEP

Semiconductor etching is one of the important process steps in the semiconductor manufacturing process, which is typically batch process [35]. The most popular etching method is plasma etching, in which gases are energized into plasma by means of coils or electrodes, and semiconductor etching is achieved by physical impact and chemical reactions between the plasma and wafer surface. The schematic of the SEP is shown in figure 11.

In this case, it has been performed on the Lam9600 plasma etching tool, which uses inductively coupled Bf_3/Cl_2 plasma etching of $\text{TiN}/\text{Al}-0.5\% \text{ Cu}/\text{TiN}/\text{oxide}$ for stacking. The key parameters are the linewidth of the etched Al lines, its uniformity on the wafer and the oxide loss. There are three sensor systems equipped with the metal etcher used in the experiments: device status, radio frequency monitor, and optical emission spectroscopy. The device status sensor collects the device data during wafer processing, including 40 process set points during etching sampled at 1 s intervals, such as gas flow, chamber pressure, radio frequency power, etc. To validate the proposed model, we use 10 different types of faults in the SEP, which are described in detail in table 7.

The attribute semantic descriptions of the faults contain the information about the faults class as shown in table 8, each fault is described by 9 fine-grained attributes.

The semantic description matrix of the fault attributes of the SEP is shown in figure 12, the value ‘1’ means that the fault

Table 7. Semiconductor etching process fault information.

No.	Fault state information
1	Radio frequency power – 12 W
2	Radio frequency power + 10 W
3	He chuck pressure + 3 Pa
4	Transformer coupled plasma power + 10 W
5	BCl_3 flow rate + 5 sccm
6	He chuck pressure – 2 Pa
7	Cl_2 flow rate – 5 sccm
8	Cl_2 flow rate + 5 sccm
9	BCl_3 flow rate – 5 sccm
10	Transformer coupled plasma power – 15 W

Table 8. Fault attribute semantic description for SEP.

No.	Fault state information
ASD#1	Radio frequency power is changed
ASD#2	He chucks pressure is changed
ASD#3	Transformer coupled plasma power is changed
ASD#4	BCl_3 flow rate is changed
ASD#5	Cl_2 flow rate is changed
ASD#6	Fault change value is positive
ASD#7	Fault change value is negative
ASD#8	Gas flow rate change
ASD#9	Power is changed

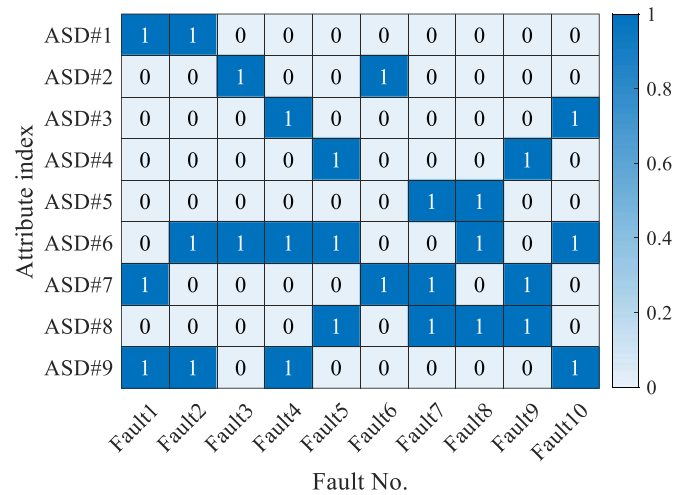


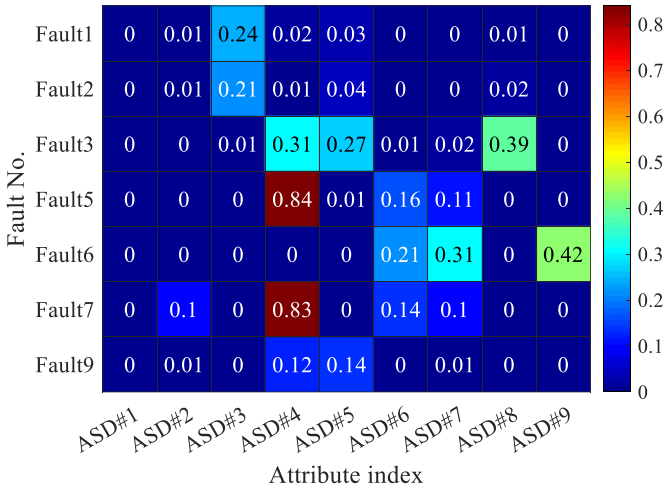
Figure 12. Attribute vectors of the SEP.

has the current attribute, and ‘0’ means that the fault does not have the current attribute.

The 10 faults are divided into 7 training faults and 3 target faults, and are categorized into 6 groups according to different situations, which are shown in table 9. The first three groups of training faults are all seen and the target faults is unseen, which is used to verify the effectiveness of the zero-shot fault diagnosis. The rest of the groups of training faults are seen faults and the target faults are both seen and unseen faults,

Table 9. Six different fault groups for SEP.

No.	Training faults	Target faults
A	1, 3–5, 7–9	2, 6, 10
B	1–3, 5–7, 9	4, 8, 10
C	2–3, 5–7, 9–10	1, 4, 8
D	2–3, 5–6, 8–10	1, 5, 8
E	1–4, 7–8, 10	2, 5, 9
F	1–2, 4–6, 8–9	3, 5, 7

**Figure 13.** Attribute prediction error rate.

which are used to verify the effectiveness of the generalized zero-shot fault diagnosis.

The output data and the labeled data of each fault attribute are compared to calculate the attribute prediction error rate. The attribute prediction error rate is shown in figure 13.

When the attribute error rate exceeds the threshold, the attribute would be changed. According to the way that the threshold is selected for the penicillin fermentation process, the threshold with the lowest attribute prediction error rate is selected in this case, i.e. the attribute prediction error rate is minimized in $\eta = 0.8$. The two equilibrium parameters α and β of the JLMDSE-mVAE loss function, which are selected in the same way as for the PFP using the grid search method, and the search range is between 0 and 1. In this case, we choose $\alpha = 0.2$ and $\beta = 0.1$.

In table 10, the sample fault diagnosis accuracies of the proposed model for different fault groups are given and compared with two well advanced zero-shot fault diagnosis models, FDAT and SCE, with probabilistic NB and nonlinear RF selected as the attribute learners. It can be seen from table 4 that the proposed model performs better than the two compared models overall. Especially for group A, with the NB classifier, the diagnostic accuracies of JLMDSE-mVAE

increase by 30.37% and 19.63% compared to FDAT and SCE respectively.

The generalized zero-shot fault diagnosis accuracies of the three models are given in table 11. It can be obtained from table 11 that the proposed model in this paper is better than FDAT and SCE for generalized zero-shot fault diagnosis. Overall, when NB is used as a classifier, the diagnostic accuracies of the three methods are higher than that of RF classifier. Specifically, when NB is utilized as a classifier, the H value of JLMDSE-mVAE is most significantly improved by 26.94% compared to FDAT on group D, and JLMDSE-mVAE is most significantly improved by 25.05% compared to SCE for group F.

To verify the effectiveness of the proposed loss function in improving the accuracy of zero-shot fault diagnosis in PFP, the results of the ablation experiments are given as shown in table 12. In ablation experiments, NB is used only for attribute learning, due to which classification is better than RF. In table 6, CA-mVAE only includes the cross-reconstruction loss function, CR-mVAE only includes the consistent alignment loss function, JLMDSE-mVAE-I includes both above loss functions and is not corrected for the semantic information, and JLMDSE-mVAE-II denotes the inclusion of the above loss functions with correction for semantic information. JLMDSE-mVAE-II has an increase for mean accuracy of 23.24% and 18.72 compared to CA-mVAE and CR-mVAE, respectively, which indicates that the additional loss function has a significant advantage in improving the fault diagnosis performance. The mean fault diagnosis accuracy of JLMDSE-mVAE-II is improved by 3.95% compared to JLMDSE-mVAE-I, which indicates that the semantic correction has a significant improvement on the generalized zero-shot fault diagnosis performance.

The Barlow matrixes of JLMDSE-mVAE on groups A and D are shown in figure 14, which reveal that the diagonal elements are close to 1 and the off-diagonal elements are close to 0. This indicates that the proposed method in this paper has high consistency of sample embedding and attribute embedding alignment, while reducing the redundant information.

The T-SNE visualization of the fault features extracted by FDAT, SCE and the JLMDSE-mVAE proposed in this paper in 2D feature space is shown in figure 15.

From figures 15(a) and 14(b), there are a large amount of overlap in the features extracted by FDAT and SCE, which lead to misclassification of unseen faults as seen faults. Figures 15(c) and 14(d) show that the JLMDSE-mVAE extracted features have clearer boundaries and the distinction between different fault features is more obvious, which makes JLMDSE-mVAE more effective for generalized zero-shot fault diagnosis. Compared with figure 15(c), figure 15(d) has clearer contours, which indicates that the semantically corrected semantic attribute matrix is more responsive to the correlation information among the samples.

Table 10. Fault diagnosis results Acc for SEP (%).

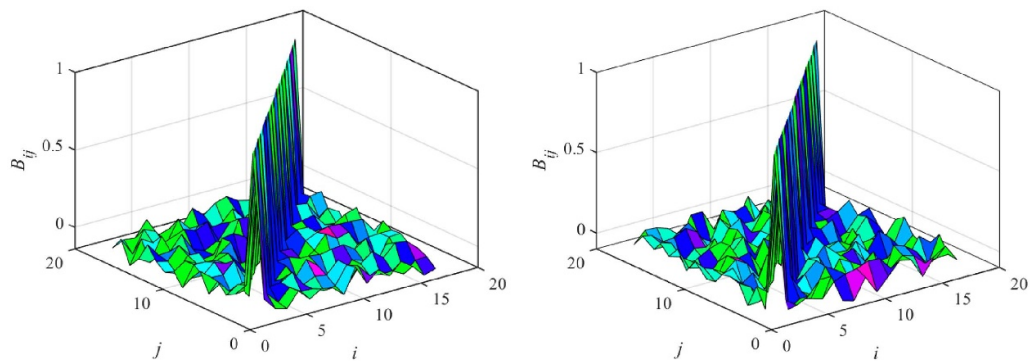
Method	Classifier	A	B	C	D	E	F
FDAT	NB	52.22	41.11	62.22	58.89	51.94	49.44
	RF	51.48	35.56	55.19	55.19	48.06	46.67
SCE	NB	62.96	65.56	78.06	66.30	74.17	48.33
	RF	59.26	59.63	70.74	62.59	73.06	47.78
JLMDSE-mVAE	NB	82.59	84.44	80.83	75.93	78.89	74.44
	RF	78.15	80.37	78.15	72.22	75.19	73.70

Table 11. Generalized zero-shot fault diagnosis results A and H for SEP (%).

Method	Classifier	D		E		F	
		A	H	A	H	A	H
FDAT	NB	57.83	58.93	51.74	52.18	49.47	50.26
	RF	56.23	57.11	48.21	50.11	46.68	47.72
SCE	NB	66.31	67.26	74.16	74.18	48.34	50.11
	RF	62.48	64.41	73.16	74.25	47.79	48.11
JLMDSE-mVAE	NB	75.92	76.12	78.89	79.12	74.49	75.16
	RF	72.21	72.26	75.21	75.69	74.70	75.12

Table 12. Fault diagnosis results Acc with different loss functions (%).

Method	A	B	C	D	E	F	Mean
CA-mVAE (NB)	64.81	65.56	63.70	35.56	58.89	48.33	56.14
CR-mVAE (NB)	67.04	66.30	66.67	49.44	62.59	51.94	60.66
JLMDSE-mVAE-I (NB)	79.26	81.11	77.04	70.37	74.07	70.74	75.43
JLMDSE-mVAE-II (NB)	82.59	84.44	80.00	75.93	78.89	74.44	79.38

**Figure 14.** Visualization of the Barlow matrix on group A and D.

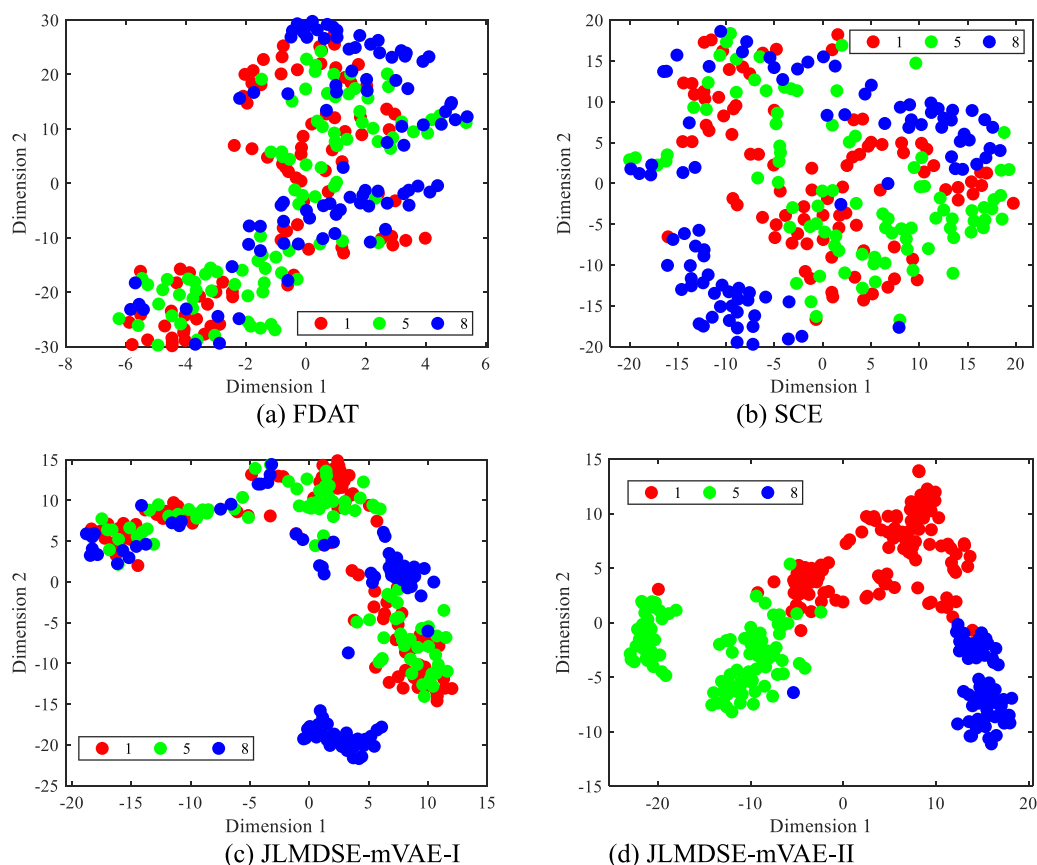


Figure 15. T-SNE visualization of feature extraction for different models on group D.

5. Conclusion

In this paper, we propose a generalized zero-shot fault diagnosis model for batch processes called JLMDSE-mVAE, which can also detect new seen and unseen fault samples by only using the sample training dataset of seen faults. In particular, (1) the training samples are mapped into a low-rank space by joint low-rank representation and manifold learning to obtain global local features of the samples while reducing the redundancy of the inputs for model. (2) The method in this paper corrects the bias of human-defined semantic attributes by predicting the attribute error rate. (3) The fault embeddings and attribute embeddings are aligned to learn better attribute classifiers by embedding the reconstructed structure and Barlow matrix. The applicability and effectiveness of the proposed JLMDSE-mVAE are verified by two PFPs and SEPs, and in both cases, the proposed method exhibits excellent performance, is much better than the compared methods. In the future, we will further optimize the model structure to improve the diagnostic performance while making the model lightweight.

Data availability statement

The data cannot be made publicly available upon publication because they are owned by a third party and the terms of use prevent public distribution. The data that support the findings

of this study are available upon reasonable request from the authors.

Acknowledgment

This research work has been awarded by the National Natural Science Foundation of China (62263021), Industrial Support Project of Education Department of Gansu Province (2023CYZC-24), Gansu Province University Young Doctor Support Project (2024QB-037), Lanzhou Youth Science and Technology Innovation Program (2023-QN-36), Science and Technology Project of Gansu Province (24JRRA172).

ORCID iDs

Xiaoqiang Zhao  <https://orcid.org/0000-0001-5687-942X>
Yongyong Hui  <https://orcid.org/0000-0002-3577-6752>

References

- [1] Zhou Y, Cao Z, Lu J, Zhao C, Li D and Gao F 2022 Objectives, challenges, and prospects of batch processes: arising from injection molding applications *Korean J. Chem. Eng.* **39** 3179–89
- [2] Peng C and ChunHao D 2022 Monitoring multi-domain batch process state based on fuzzy broad learning system *Expert Syst. Appl.* **187** 115851

- [3] Gao X, Li X, Li B, Han H, Gao H and Qi Y 2024 GELU-LSTM-encoder-decoder fault prediction for batch processes based on the global-local percentile method *Can. J. Chem. Eng.* **102** 2208–27
- [4] Yan Z, Huang B L and Yao Y 2015 Multivariate statistical process monitoring of batch-to-batch startups *AIChE J.* **61** 3719–27
- [5] Wang J and He Q P 2010 Multivariate statistical process monitoring based on statistics pattern analysis *Ind. Eng. Chem. Res.* **49** 7858–69
- [6] Ramos M, Ascencio J, Hinojosa M V, Vera F, Ruiz O, Jimenez-Feijoó M I and Galindo P 2021 Multivariate statistical process control methods for batch production: a review focused on applications *Prod. Manuf. Res.* **9** 33–55
- [7] Zhao Y, Xiao F and Wang S 2013 An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network *Energy Build.* **57** 278–88
- [8] Han T, Zhang L, Yin Z and Tan A C 2021 Rolling bearing fault diagnosis with combined convolutional neural networks and support vector machine *Measurement* **177** 109022
- [9] Jiang Q, Huang B and Yan X 2016 GMM and optimal principal components-based Bayesian method for multimode fault diagnosis *Comput. Chem. Eng.* **84** 338–49
- [10] Chen S, Yu J and Wang S 2020 One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes *J. Process Control* **87** 54–67
- [11] Guo P, Rao S, Hao L and Wang J 2022 Fault diagnosis of a semi-batch crystallization process through deep learning method *Comput. Chem. Eng.* **164** 107807
- [12] Wang N, Yang F, Zhang R and Gao F 2020 Intelligent fault diagnosis for chemical processes using deep learning multimodel fusion *IEEE Trans. Cybern.* **52** 7121–35
- [13] 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
- [14] Zhang J, Zhang M, Feng Z, Ruifang L, Lu C, Dai Y and Dong L 2023 Gated recurrent unit-enhanced deep convolutional neural network for real-time industrial process fault diagnosis *Process Saf. Environ. Prot.* **175** 129–49
- [15] Wang Y, Wu D and Yuan X 2020 LDA-based deep transfer learning for fault diagnosis in industrial chemical processes *Comput. Chem. Eng.* **140** 106964
- [16] Zabin M, Choi H-J and Uddin J 2023 Hybrid deep transfer learning architecture for industrial fault diagnosis using Hilbert transform and DCNN-LSTM *J. Supercomput.* **79** 5181–200
- [17] Chen Y, Zhang D, Zhu K and Yan R 2023 An adaptive activation transfer learning approach for fault diagnosis *IEEE ASME Trans. Mechatronics* **28** 2645–56
- [18] Ding Y, Jia M, Zhuang J, Cao Y, Zhao X and Lee C-G 2023 Deep imbalanced domain adaptation for transfer learning fault diagnosis of bearings under multiple working conditions *Reliab. Eng. Syst. Saf.* **230** 108890
- [19] Liu J and Ren Y 2020 A general transfer framework based on industrial process fault diagnosis under small samples *IEEE Trans. Ind. Inform.* **17** 6073–83
- [20] Zhang S, Wei H-L and Ding J 2023 An effective zero-shot learning approach for intelligent fault detection using 1D CNN *Appl. Intell.* **53** 16041–58
- [21] Fan L, Chen X, Chai Y and Lin W 2023 Attribute fusion transfer for zero-shot fault diagnosis *Adv. Eng. Inf.* **58** 102204
- [22] Lampert C H, Nickisch H and Harmeling S (eds) 2009 Learning to detect unseen object classes by between-class attribute transfer. *2009 IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE)
- [23] Lampert C H, Nickisch H and Harmeling S 2013 Attribute-based classification for zero-shot visual object categorization *IEEE Trans. Pattern Anal. Mach. Intell.* **36** 453–65
- [24] Feng L and Zhao C 2020 Fault description based attribute transfer for zero-sample industrial fault diagnosis *IEEE Trans. Ind. Inform.* **17** 1852–62
- [25] Hu Z, Zhao H, Yao L and Peng J 2022 Semantic-consistent embedding for zero-shot fault diagnosis *IEEE Trans. Ind. Inform.* **19** 7022–31
- [26] Chen E, Wang Z and Zheng J 2023 A zero-sample industrial process fault diagnosis model based on joint explicit and implicit attribute transfer *Measurement* **218** 113236
- [27] Chen X, Zhao C and Ding J 2023 Pyramid-type zero-shot learning model with multi-granularity hierarchical attributes for industrial fault diagnosis *Reliab. Eng. Syst. Saf.* **240** 109591
- [28] Huang J, Li Z and Zhou Z 2023 A simple framework to generalized zero-shot learning for fault diagnosis of industrial processes *IEEE CAA J. Autom. Sin.* **10** 1504–6
- [29] Xu J, Liang S, Ding X and Yan R 2023 A zero-shot fault semantics learning model for compound fault diagnosis *Expert Syst. Appl.* **221** 119642
- [30] Zhang Y, Qin N, Huang D, Yang A, Jia X and Du J 2024 Generalized zero-shot approach leveraging attribute space for high-speed train Bogie *IEEE Trans. Instrum. Meas.* **73** 3512412
- [31] Xu J, Zhang H, Chen W, Fan Y and Ding X 2024 CGASNet: a generalized zero-shot learning compound fault diagnosis approach for bearings *IEEE Trans. Instrum. Meas.* **73** 2513111
- [32] Xiao S, Tan M, Xu D and Dong Z Y 2015 Robust kernel low-rank representation *IEEE Trans. Neural Netw. Learn. Syst.* **27** 2268–81
- [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] Birol G, Ündey C and Cinar A 2002 A modular simulation package for fed-batch fermentation: penicillin production *Comput. Chem. Eng.* **26** 1553–65
- [35] Hirai T and Kano M 2015 Adaptive virtual metrology design for semiconductor dry etching process through locally weighted partial least squares *IEEE Trans. Semicond. Manuf.* **28** 137–44

基于S-RNPAE算法的间歇过程早期故障监测

刘 凯^{1,2}, 赵小强^{1,2,3†}, 牟 森^{1,2}, 张 妍^{1,2}

- (1. 兰州理工大学 电气工程与信息工程学院, 兰州 730050;
2. 甘肃省工业过程先进控制重点实验室, 兰州 730050;
3. 兰州理工大学 国家级电气与控制工程实验教学中心, 兰州 730050)

摘 要: 针对具有多变量、非线性和高维度特点的间歇过程数据使得早期故障信号易被噪声干扰且故障幅值低导致故障监测效果不佳的问题, 提出一种基于堆叠鲁棒邻域保持自编码 (stack-robust neighborhood preserving autoencoder, S-RNPAE) 的间歇过程早期故障监测方法. 首先, 通过 $L_{2,1}$ 范数重新设计自编码器的目标函数, 以提高模型对噪声和离群点的鲁棒性; 其次, 利用邻域保持嵌入来正则化鲁棒自编码器的方式构建鲁棒邻域保持自编码 (robust neighborhood preserving autoencoder, RNPAE) 模块, 解决自编码器作为一种全局模型而忽略包含早期故障特征的局部近邻信息的提取问题; 然后, 将多个 RNPAE 模块堆叠构造 S-RNPAE 网络, 从而获取深层全局-局部特征, 保证对早期微小故障信息提取更充分, 并建立检测统计量实现过程检测; 最后, 利用一种适用于非线性过程的贡献图方法完成故障诊断, 其诊断结果更准确. 通过 Swiss Roll 数据集和青霉素发酵过程的实验表明, 所提方法的特征提取能力更强, 对间歇过程的早期故障更敏感, 具有更好的早期故障监测效果.

关键词: 间歇过程; 早期故障; 过程监测; 自编码器; 邻域保持嵌入

中图分类号: TP277

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.1401

引用格式: 刘凯, 赵小强, 牟森, 等. 基于 S-RNPAE 算法的间歇过程早期故障监测[J]. 控制与决策, 2024, 39(5): 1577-1586.

Incipient fault monitoring of batch process based on S-RNPAE algorithm

LIU Kai^{1,2}, ZHAO Xiao-qiang^{1,2,3†}, MOU Miao^{1,2}, ZHANG Yan^{1,2}

- (1. College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China; 2. Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou 730050, China; 3. National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: For batch process data with multivariate, nonlinear, and high dimensional characteristics, the early fault signals are easy to be disturbed by noise and the fault amplitude is low, which lead to poor fault monitoring performance, we propose an early fault detection method for batch processes based on stack-robust neighbourhood preserving autoencoder (S-RNPAE) in this paper. Firstly, the objective function of the autoencoder is redesigned by the $L_{2,1}$ norm to improve the model robustness to noise and outliers. Secondly, the robust neighbourhood preserving autoencoder (RNPAE) module is constructed by using neighbourhood preserving embedding to regularize the robust autoencoder, which solves the problem of the autoencoder as a global model that ignores the local neighbour information containing early fault features. Then, we construct the S-RNPAE network by stacking multiple RNPAE modules, which can obtain deep global-local features to ensure the extraction of early minor fault information more fully, and we establish detection statistics to realize process monitoring. Finally, the nonlinear contribution diagram method is used to complete the fault diagnosis, which is more accurate for nonlinear variables. The examples of the Swiss Roll dataset and the simulation process of penicillin fermentation show that the proposed method has stronger feature extraction ability, which is more sensitive to the early fault of batch process and has a better early fault monitoring performance.

Keywords: batch process; incipient fault; process monitoring; autoencoder; neighborhood preserving embedding

收稿日期: 2022-08-03; 录用日期: 2022-12-20.

基金项目: 国家重点研发计划项目(2020YFB1713600); 国家自然科学基金项目(62263021, 62163023); 甘肃省科技计划项目(21JR7RA206, 21YF5GA072); 甘肃省工业过程先进控制重点实验室开放基金项目(2022KX07); 甘肃省教育厅产业支撑项目(2021CYZC-02).

责任编委: 赵春晖.

[†]通讯作者. E-mail: xqzhao@lut.edu.cn.

0 引言

现代工业生产中,间歇过程由于其产品种类多、规模小、个性化、附加值等优点备受关注,由于其生产的灵活和高效的特点,被广泛应用于轧钢、注塑、晶圆蚀刻、微生物发酵等工业领域.然而,间歇过程由于复杂的生产方式导致系统一旦发生故障,直接影响产品质量,甚至还会引发安全事故危及操作人员的生命安全^[1-2].由于间歇生产过程的间断性和生产环境的复杂性,早期故障的过程数据具有多变量、非线性、高维的特点.当早期故障发生时,由于其故障幅度小,容易被噪声所掩盖,从而极难有效地诊断故障.如何有效地检测和诊断间歇过程的早期故障对保证生产安全具有十分重要的意义^[3-4].

随着计算机和控制技术的发展,大量的过程数据被收集和存储,促进了数据驱动的多变量统计过程监测方法的快速发展.众多学者致力于多变量统计监测方法的研究,典型的多变量统计方法如主成分分析^[5]、独立成分分析^[6]、偏最小二乘^[7]以及其扩展形式被广泛应用于过程监控. Harrou等^[8]结合非线性投影潜变量结构建模的优点和Hellinger距离度量的优点,以识别高度相关的多元数据中的异常变化,提出多变量故障检测方法. Harmouche等^[9]基于主成分分析和Kullback-Leibler散度的概率分布度量提出一种基于概率分布度量的故障检测方法用于监测早期故障. Ji等^[10]通过引入两种具有代表性的平滑技术,提出一种通用故障检测指标的新型早期故障检测策略,该策略对早期故障表现出较高的敏感度. Shang等^[11]提出一种递归变换分量统计分析的方法用于早期故障检测,这类早期故障检测的方法未考虑过程数据的非线性特征提取,造成部分信息丢失.随着分布式控制系统的广泛应用,以及数据存储、传输和处理技术的不断发展,基于神经网络的智能故障检测方法具有较强的非线性处理能力而得到广泛的应用,如深度信念网络^[12]、自编码器(autoencoder, AE)^[13]、长短期记忆^[14]和卷积神经网络^[15]被成功应用于间歇过程故障监测.其中,以AE及其变体为主的算法由于其模型泛化性强、训练过程不需要数据标注的优点备受关注. Gao等^[16]提出一种多路拉普拉斯自编码器的方法,用于间歇过程监控; Agarwal等^[17]提出一种多路偏最小二乘自动编码器架构用于间歇过程故障检测; Wang等^[18]通过长短期记忆网络从过程每个阶段的时间序列中提取与质量相关的综合隐藏特征,提出一种基于堆叠自动编码器的复杂特征表示和质

量预测框架用于间歇过程.

尽管上述这些机器学习的智能故障监测在间歇过程中取得了成功,但仍然存在一些局限性.基于AE的故障检测方法易受噪声和离群点的影响,导致系统的鲁棒性较差,因此提高AE算法的鲁棒性对于有效监测过程是否发生故障十分重要.同时,过程数据的局部信息包含重要的早期微小故障信息,然而AE是一种全局端到端模型,而并未捕获局部近邻信息.此外,连续的特征压缩可能会丢失原始数据中重要的局部邻域信息,这对于过程监控是不利的,需要从原始输入数据中提取局部邻域信息.流形学习的方法能够将复杂的高维空间转化为低维空间,从而在低维空间中提取高维空间的数据信息.典型的流形学习方法包括等距特征映射^[19]、拉普拉斯特征映射^[20]、局部切空间对齐^[21]、邻域保持嵌入(neighborhood preserving embedding, NPE)^[22]等.其中,NPE不仅在提取数据的流形结构方面有很好的性能,而且能够保持数据原有的局部邻域信息. Liu等^[23]受流形学习NPE思想的启发,提出一种基于NPE正则化的深度学习方法来提取邻域保持特征,用于工业软测量; Li等^[24]提出一种基于流形学习的过程监控方案,该方案将统计局部方法融入NPE中,以监控过程数据局部结构的变化,不仅继承了NPE挖掘数据局部结构的能力,而且通过监测新观测的局部信息变化实现在线故障检测; Yu等^[25]提出一种深度神经网络流形正则化堆叠自动编码器,用于复杂工业过程的故障检测.以上方法在考虑数据的全局结构的同时更加关注局部近邻信息,但对于过程的早期故障并未考虑.

综上所述,针对间歇过程在故障发生早期幅值小、信号微弱导致监测难的问题,本文提出一种堆叠鲁棒邻域保持自编码(stack-robust neighborhood preserving autoencoder, S-RNPAE)的间歇过程故障监测方法.首先,在基于AE有效处理过程非线性的基础上,计算AE重构误差时采用 $L_{2,1}$ 范数增强算法对噪声和异常点的鲁棒性;其次,为了解决传统AE作为一种全局模型忽略过程数据局部近邻信息,难以提取过程数据全部信息的问题,通过NPE作为AE的正则项的方式构造鲁棒邻域保持自编码(robust neighborhood preserving autoencoder, RNPAE)结构,该结构能够有效提取过程数据的全局-局部特征;然后,采用多层堆叠的方式进一步提取过程数据深层微小特征,使得对于包含早期微小故障信息提取更充分;最后,通过Swiss Roll数据集验证S-RNPAE的特征提

取能力,同时将本文方法应用于青霉素发酵过程,验证所提方法在间歇过程早期故障监测的有效性和优越性。

1 基本方法

1.1 自编码器(AE)

AE^[26]是一种人工神经网络,广泛应用于高维数据降维和特征提取. 一个基本的自编码器网络由编码器和解码器两部分组成,编码器将原始数据传递到隐藏层获得数据低维表示形式,解码器将低维数据传递到输出层,以获得原始数据重构,使得重构误差足够小,重构数据可以保留原始数据的大部分信息. 在自编码器网络训练过程中,将输入数据 x 映射到隐藏层,得到编码后的 z ,然后通过反向解码得到重构数据 \hat{x} ,输入数据的维数为 m ,输入样本的个数为 n ,原始数据经过编码压缩后的隐藏层 z 的维度为 d .

编码器公式为

$$z = f(W_e x + b_e). \quad (1)$$

其中: f 是编码器的激活函数, W_e 是 $m \times d$ 维的加权矩阵, b_e 是维度为 d 的偏置向量。

解码器公式为

$$\hat{x} = g(W_d z + b_d). \quad (2)$$

其中: g 是解码器的激活函数, W_d 是 $d \times m$ 维的加权矩阵, b_d 是维度为 m 的偏置向量。

自编码器网络训练的损失函数如下式所示:

$$J_{AE} = \frac{1}{2n} \sum_{i=1}^n \|\hat{x}_i - x_i\|^2, \quad (3)$$

其中 n 为训练样本的个数. 自编码器网络训练的目标是使输入数据重构的误差最小,因此为了使重构数据尽可能与输入数据相等,将损失函数的值最小化。

1.2 邻域保持嵌入(NPE)

NPE^[27]是一种最优地保留了数据集的邻域结构特征的流形学习算法,它是通过构造邻接图来揭示数据的邻域关系,对数据点最近邻域进行重构,将数据投影到低维空间,并在低维空间保留其邻域结构. 假设原始数据集 X 包含 n 个训练样本,通过重构每个数据点 k 个最近邻的权重系数来反映NPE的邻域结构, w_{ij} 是用于重构 x_i 的数据点 x_j 的权重系数. 权重系数的计算可以通过最小化重构误差获得,如下所示:

$$\min_w \sum_{i=1}^n \left(x_i - \sum_{j=1}^n w_{ij} x_j \right)^2, \quad (4)$$

其中 w_{ij} 是 x_i 和 x_j 的连接权重,若 x_j 不是 x_i 的近邻,则 $w_{ij} = 0$, w_{ij} 满足约束 $\sum_{j=1}^n w_{ij} = 1$.

原始数据通过映射矩阵 A 降维到低维空间中,映射矩阵可通过求解如下的最小化代价函数获得:

$$J_{NPE} = \min_w \sum_{i=1}^n \left(y_i - \sum_{j=1}^n w_{ij} y_j \right)^2, \quad (5)$$

其中 $Y = \sum_{j=1}^n y_j$ 为原始数据 $X = \sum_{i=1}^n x_i$ 通过映射矩阵 A 在低维空间的数据表示. 式(5)的矩阵表示如下所示:

$$J_{NPE} = \min Y^T (I - W)^T (I - W) Y = \min A^T X (I - W)^T (I - W) X^T A, \quad (6)$$

约束条件为 $Y^T Y = A^T X X^T A = I$. 引入拉格朗日乘子法,将式(6)的求解问题转化为如下求广义特征值的问题:

$$X M X^T a = \lambda X X^T a, \quad (7)$$

其中 $M = (I - W)^T (I - W)$,求解 M 所对应的前 d 个最小特征值($\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_d$)的特征向量构成映射矩阵 $A(\alpha_1, \alpha_2, \dots, \alpha_d)$,即满足 $Y = A^T X$.

2 基于S-RNPAE算法的早期故障监测

2.1 鲁棒自编码(robust autoencoder, RAE)

自编码器网络在训练过程中,编码器将原始数据嵌入到一个潜在的低维空间中,解码器重构尽可能接近原始输入. 最小化重构误差函数来优化网络参数 $\theta = \{w_e, w_d, b_e, b_d\}$,如下所示:

$$\min_{\theta} \frac{1}{2n} \sum_{i=1}^n \|\hat{x}_i - x_i\|_2^2 = \min_{\theta} \frac{1}{2n} \|\hat{X} - X\|_F^2. \quad (8)$$

对式(8)求偏导,得

$$\frac{\partial \frac{1}{2n} \sum_{i=1}^n \|\hat{X} - X\|_F^2}{\partial \hat{x}_i} = \frac{1}{n} (\hat{x}_i - x_i). \quad (9)$$

式(9)表明,具有较大重构误差的噪声和异常点在AE网络的目标函数起重要作用,影响过程监测的效果,然而使用Frobenius范数训练网络对噪声敏感. 为了克服这一问题,采用 $L_{2,1}$ 范数作为目标函数,能够在网络训练过程中更好地处理噪声和异常点.

$$\min_{\theta} \frac{1}{2n} \|\hat{X} - X\|_{2,1}. \quad (10)$$

对式(10)求偏导得

$$\frac{\partial \frac{1}{2n} \|\hat{X} - X\|_{2,1}}{\partial \hat{x}_i} = \frac{1}{n} \frac{\hat{x}_i - x_i}{\|\hat{x}_i - x_i\|_2 + \varepsilon}. \quad (11)$$

其中: ε 是一个非常小的常数, $\frac{\hat{x}_i - x_i}{\|\hat{x}_i - x_i\|_2 + \varepsilon}$ 近似于 $\frac{\hat{x}_i - x_i}{\|\hat{x}_i - x_i\|_2}$.

通过对比式(9)和(11)可以发现, $L_{2,1}$ 范数对每个残差进行归一化, 异常点只占过程数据的小部分, 归一化可以抑制残差较大的异常点对AE参数调节的影响. 此外, $L_{2,1}$ 范数更多关注大部分数据集. 具有 $L_{2,1}$ 范数的AE网络可以提取到更多有用信息, 对噪声和异常点具有更强的鲁棒性.

2.2 鲁棒邻域保持自编码(RNPAE)

经过改进的鲁棒自编码器是一个全局网络, 进行特征提取时忽略了数据近邻结构, 通过这种网络提取到的特征只能捕获单个数据的分布特征, 没有考虑样本关系. 为了提高学习网络的泛化性能, 通过NPE正则化的方式保持输入层空间中每个数据的局部邻域结构特征. 将具有NPE正则化的RAE称为RNPAE, RNPAE的目标不仅是使重建误差尽可能小, 而且要保持原始数据的邻域结构. 采用如下所示的目标函数来训练RNPAE模型:

$$J(\theta) = J_{\text{RAE}}(\theta) + \eta J_{\text{NPE}}(\theta) = \min_{\theta} \frac{1}{2n} \|\hat{X} - X\|_{2,1} + \eta \min_w \sum_{i=1}^n \left(y_i - \sum_{j=1}^n w_{ij} y_j \right)^2, \quad (12)$$

其中 η 是一个平衡重构误差和局部保留误差权重的正则化参数.

RNPAE的目标函数由两部分组成, 前者是数据在输入层和输出层之间的重构误差, 以确保重构的数

据与原始数据尽可能一致; 后者是数据在输出层的局部邻接误差, 通过局部邻接约束确保重构数据与原始数据保持相同的局部邻近结构, 在RNPAE中保留局部近邻信息. 为了解决式(12)的优化问题, 采用基于反向传播(BP)的梯度下降算法进行迭代更新, 对于参数 $\theta = \{W^m, b^{(m)}\}_{m=1}^M$ 第 t 次更新迭代如下所示:

$$\begin{cases} W_t^{(m)} = W_{t-1}^{(m)} - \mu \frac{\partial J_t(\theta)}{\partial W_t^{(m)}}, \\ b_t^{(m)} = b_{t-1}^{(m)} - \mu \frac{\partial J_t(\theta)}{\partial b_t^{(m)}}. \end{cases} \quad (13)$$

$J(\theta)$ 相对于 $W^{(m)}$ 和 $b^{(m)}$ 的子梯度可通过下式所得:

$$\begin{cases} \frac{\partial J(\theta)}{\partial W^{(m)}} = \frac{\partial J_{\text{RAE}}(\theta)}{\partial W^{(m)}} + \eta \frac{\partial J_{\text{NPE}}(\theta)}{\partial W^{(m)}} = \sum_{i=1}^N (\nabla_{1i}^{(m)})^T h_i^{(m-1)} + \eta (\nabla_{2i}^{(m)})^T h_i^{(m-1)}, \\ \frac{\partial J(\theta)}{\partial b^{(m)}} = \frac{\partial J_{\text{RAE}}(\theta)}{\partial b^{(m)}} + \eta \frac{\partial J_{\text{NPE}}(\theta)}{\partial b^{(m)}} = \sum_{i=1}^N (\nabla_{1i}^{(m)})^T + \eta (\nabla_{2i}^{(m)})^T. \end{cases} \quad (14)$$

其中: $\nabla_{1i}^{(m)}$ 和 $\nabla_{2i}^{(m)}$ 表示梯度, $h_i^{(m-1)}$ 表示网络的潜在表征.

2.3 堆叠鲁棒邻域保持自编码(S-RNPAE)的故障监测

S-RNPAE是由多个RNPAE依次连接而成的深层网络结构, 图1所示为S-RNPAE网络结构.

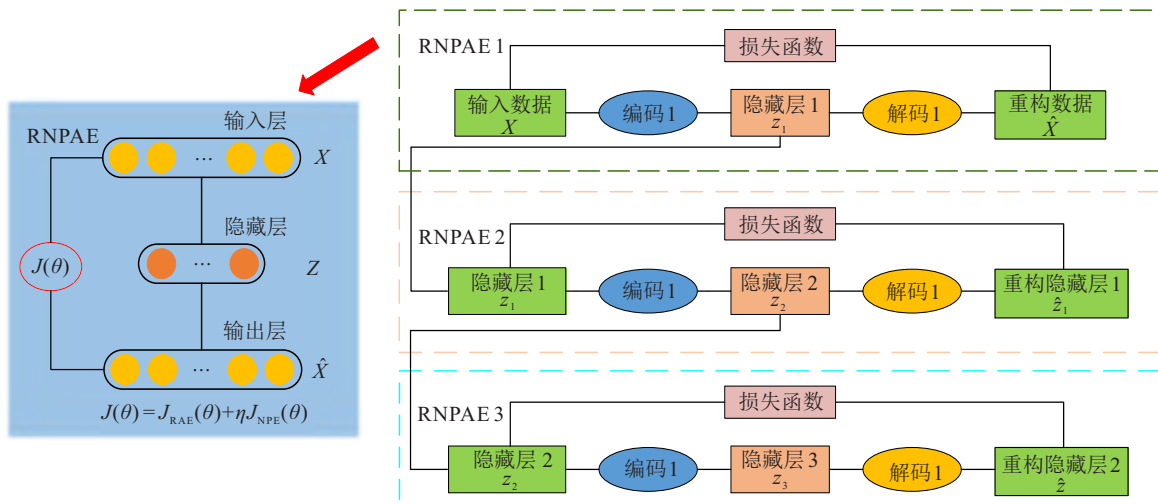


图1 S-RNPAE网络结构

本文S-RNPAE是由3个顺序连接的RNPAE组成的深度网络结构, 在S-RNPAE算法中, NPE正则化作

为附加项加入到逐层预训练的损失函数中, 对于每个RNPAE, 首先通过寻找近邻的方法为其输入数据

构建邻接图,然后通过计算每个输入数据节点与其相邻节点之间的权重系数来获得相似性矩阵,基于新的损失函数,利用输入数据和得到的相似度矩阵对RNPAE进行预训练,同样从第一个RNPAE到最后一个RNPAE,邻域结构特征被逐层保留.利用S-RNPAE对深度网络的训练参数进行初始化,可以提高模型的监测性能.本文提出S-RNPAE网络的主要目标是恢复一个完整的潜在空间,该空间能够很好地揭示输入数据的局部近邻结构.

通过将正常数据训练获得特征空间和残差空间,在特征空间建立 T^2 统计量,用于度量投影在特征空间的样本的变化. T^2 的计算公式为

$$T^2 = z^T \Phi^{-1} z. \quad (15)$$

其中: z 是一个向量,表示通过S-RNPAE从原始高维数据中提取的低维特征; Φ 是协方差矩阵.

在残差空间建立SPE统计量,用于度量样本在残差空间的投影,定义为算法模型所生成的残差的平方和,SPE的计算公式为

$$\text{SPE} = \|x - \hat{x}\|^2. \quad (16)$$

利用核密度估计的方法确定监测统计量的控制限.

当检测到故障发生时,通过确定哪些变量与异常行为的发生直接相关,从而诊断故障发生的根本原因.本文通过统计量 T^2 和SPE的贡献图来实现故障诊断,贡献图通过计算潜在故障变量对监测统计的贡献来识别潜在故障变量,对于识别故障变量表现出较好的效果.在贡献图中,对监控统计具有最大贡献值的变量通常是故障源.由于过程具有的非线性特性,传统线性贡献图通常假设过程线性,诊断效果并不理想,本文所采用的非线性贡献图方法灵敏度更高,对小扰动引起的故障更加灵敏^[28].

对于统计量 T^2 和SPE的贡献图的计算公式如下:

$$\text{CT}_i^2 = x_i \frac{\partial T^2}{\partial x_i}, \quad (17)$$

$$\text{CSPE} = x_i \frac{\partial \text{SPE}}{\partial x_i}. \quad (18)$$

对于贡献度在整个训练数据集上执行的均值和标准差运算,相对贡献表达式如下:

$$\text{RCT}_i^2 = \frac{\text{CT}_i^2 - \text{mean}(\text{CT}_{i,\text{normal}}^2)}{\text{std}(\text{CT}_{i,\text{normal}}^2)}, \quad (19)$$

$$\text{RSPE}_i = \frac{\text{CSPE}_i - \text{mean}(\text{CSPE}_{i,\text{normal}})}{\text{std}(\text{CSPE}_{i,\text{normal}})}. \quad (20)$$

其中: $\text{CT}_{i,\text{normal}}^2$ 和 $\text{CSPE}_{i,\text{normal}}$ 表示正常操作条件下第*i*个过程变量的贡献值, $\text{mean}(\cdot)$ 和 $\text{std}(\cdot)$ 表示均

值和标准差.对于贡献值的计算可正可负,由于反应过程变量的主要影响因素为数值的大小,因此在绘制贡献图时采用绝对值的方式.

2.4 堆叠鲁棒邻域保持自编码(S-RNPAE)的故障监测流程

S-RNPAE可以从原始数据中学习深层的邻域特征,这些特征能够保留原始数据的邻域结构信息,更适合早期故障的监测.图2所示为基于S-RNPAE的间歇过程故障监测的基本过程,该过程主要包含离线建模和在线监测两部分,同时进一步地诊断出发生故障的变量.

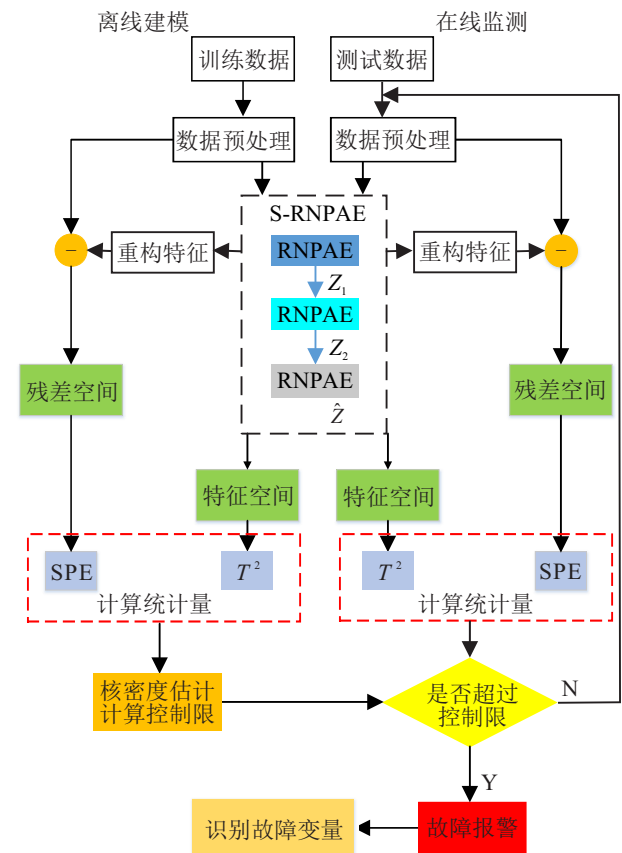


图2 基于S-RNPAE的故障监测流程

2.4.1 离线建模

step 1: 正常生产条件下,采集间歇过程的运行数据组成三维训练数据集 X ,将训练数据集 $X(I \times J \times K)$ 按照先沿批次方向展开成二维矩阵 $X(I \times JK)$,再沿变量方向排列为 $X(IK \times J)$ 的批次-变量混合展开的方式展开并标准化(I 表示批次, J 表示变量, K 表示采样时间);

step 2: 通过式(12)构造RNPAE网络,在有效提取数据特征的同时充分挖掘数据的邻域结构,利用堆叠的方式建立3层的S-RNPAE网络模型,对S-RNPAE网络进行训练获得网络参数;

step 3: 获得特征空间和残差空间,分别在特征空

间和残差空间建立 T^2 和SPE统计量,利用核密度估计的方法求取统计量的监测控制限.

2.4.2 在线监测

step 1: 在线采集监测数据按照批次-变量展开的方式展开且标准化数据;

step 2: 将标准化后的数据输入到训练好的S-RNPAE网络中,并将其投影到特征空间和残差空间;

step 3: 通过式(15)和(16)计算 T^2 和SPE统计量,一旦两个统计值中的任何一个超过其对应的阈值,就触发警报;

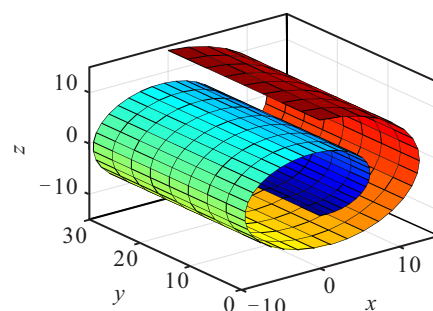
step 4: 在使用S-RNPAE有效地检测到故障之后,通过式(19)和(20)计算故障变量贡献值并绘制贡献图,通过贡献图完成故障诊断.

3 实验结果与分析

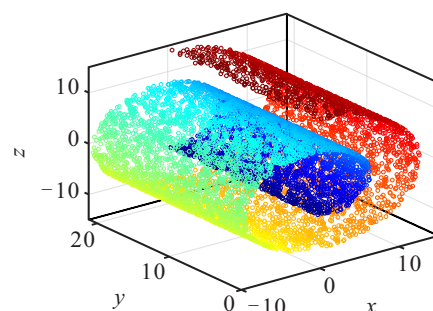
3.1 Swiss Roll数据集

Swiss Roll数据集是常见的验证算法嵌入结果的数据集,通过在三维欧氏空间中的Swiss Roll曲面上随机采样,并利用不同颜色区分样本的分布,近邻由相同颜色标识. 本文利用Swiss Roll数据集,本征维数为2,样本点数为8000. 图3所示为Swiss Roll曲面和采样数据点分布.

通过降维将原始数据复杂的非线性流形展开,观察降维过程中数据结构的保持性. 将S-RNPAE算法



(a) 原始Swiss Roll



(b) 样本数据集

图3 Swiss Roll数据集

与AE、NPE、堆栈自编码器(stacked autoencoder, SAE)^[29]、动态图嵌入(dynamic graph embedding, DGE)^[30]和领域保持自编码(neighborhood preserving autoencoder, NPAE)^[31]进行对比,以验证所提出算法的特征提取能力. 图4所示为特征空间的Swiss Roll数据投影.

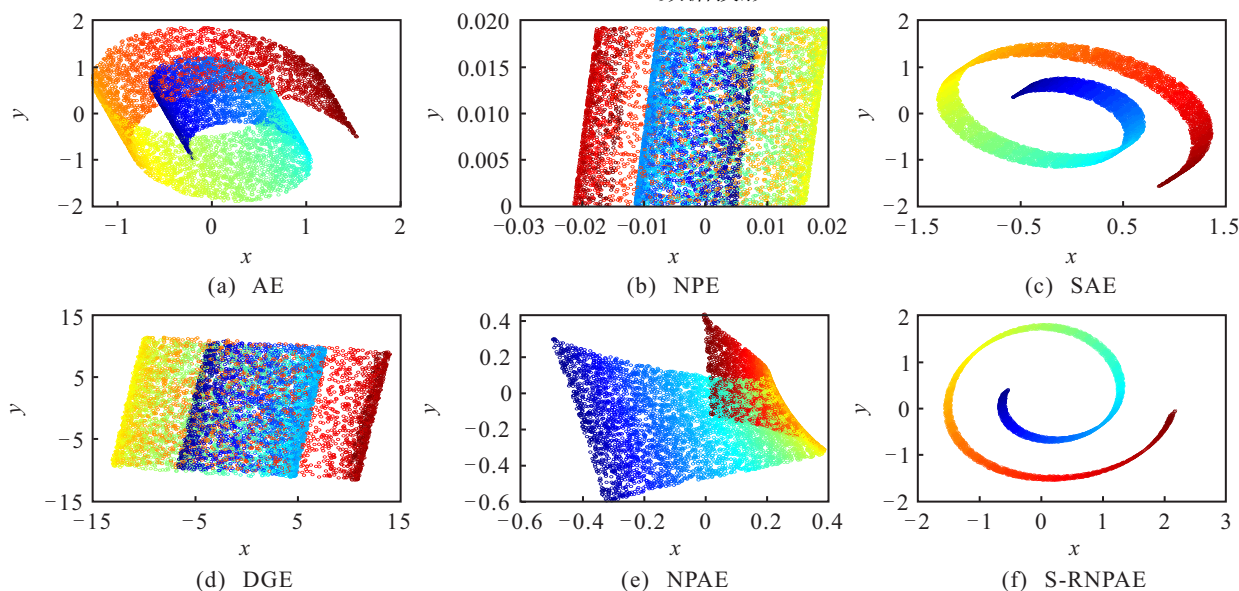


图4 特征空间的Swiss Roll数据投影

图4(a)中,AE在一定程度上能够保持数据的原有结构,但存在大量的数据点重合,重合处未能反映数据的近邻信息;图4(b)中,NPE保留了数据的局部近邻结构信息,但存在较多的重合点,提取效果较差;图4(c)中,SAE相较于AE提取更多的数据近邻特征,

但结构变形严重,不能够很好地保持原有结构;图4(d)中,DGE对局部近邻信息保留较好,但存在大量的重合点,特征提取效果一般,未能提取原始数据的完整结构;图4(e)中,NPAE保持近邻信息较好,但在一定区域内存在较多的重合部分,不能完全提取数据

特征;图4(f)中,本文所提S-RNPAE能够很好地保持局部近邻信息,降维结果清晰可靠,特征提取能力强.

3.2 青霉素发酵过程

青霉素发酵作为典型的间歇过程,其过程数据的非线性和高维度等特性使得对生产过程状态的监控更加复杂.美国伊利诺州立理工学院研究小组研发的青霉素发酵平台是一个模拟实际发酵过程的仿真平台^[32],该平台通过设定初始变量的参数和运行条件产生过程数据,完成青霉素发酵过程的建模、监测和控制.通过该平台可以设定通风速率、搅拌功率和底物加速率3个故障变量.当扰动信号为斜坡信号时,由于故障变化缓慢,在故障发生的早期幅值较小、信号微弱,很难被检测.为了训练所提网络模型,本文通过仿真平台产生30批次的正常数据作为训练

样本,每个批次的反应时间为400 h,采样时间为1 h,从18个变量中选取10个过程变量(按照变量编号从1~10依次为通风速率、搅拌功率、底物流速率、底物流温度、底物浓度、溶解氧浓度、反应器体积、发酵罐温度、PH和CO₂),所有过程数据引入高斯白噪声模拟实际工况,表1所示为故障样本.

表1 故障样本

故障	变量名	扰动类型	幅值	时段/h
F_1	通风速率/(L/h)	阶跃	+2	250~400
F_2	搅拌速率/(r/min)	斜坡	+6	300~400
F_3	底物流加速率/(L/h)	阶跃	+1	100~400

对比本文所提算法与NPE、AE、SAE、DGE和NPAE.在不同样本的故障检测率,结果如表2所示.由表2可以看出本文所提算法的检测率更高.

表2 不同方法每批次故障样本的故障检测率

故障	NPE		AE		SAE		DGE		NPAE		S-RNPAE	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
F_1	0.74	0.89	0.88	0.90	0.87	0.90	0.89	0.95	0.92	0.94	0.96	0.99
F_2	0.51	0.72	0.77	0.81	0.83	0.85	0.87	0.90	0.89	0.93	0.95	0.98
F_3	0.61	0.82	0.79	0.86	0.84	0.87	0.91	0.92	0.91	0.94	0.92	0.93

图5和图6所示为故障 F_2 分别采用NPE、AE、SAE、DGE、NPAE和本文所提S-RNPAE算法的故障检测图.由图5(a)和图6(a)可以看出NPE算法对早期故障敏感性差, T^2 统计量在350 h时才能检测到故障,SPE统计量较 T^2 统计量更早,在320 h检测到故障,不能及时地检测到故障,且存在大量的漏报和误报.从图5(b)和图6(b)所示AE算法对故障的统计量检测结果来看,相较于NPE算法漏报较少,但早期故障发生时不能及时地检测到, T^2 和SPE统计量

分别在324 h和320 h检测到故障.图5(c)和图5(b)中,SAE算法相较于AE算法,网络层数增加提取故障特征的能力更强,检测到故障的时间较早, T^2 和SPE统计量均在310 h检测到故障,相较于AE和具有局部邻域信息提取的NPE算法能够更早地检测到故障,但仍然有部分早期故障未被检测出来.图5(d)和图6(d)中,DGE算法由于考虑过程数据的动态特性和近邻信息,故障检测时间更加提前, T^2 和SPE统计量分别在305 h和304 h检测到故障,但从统计图

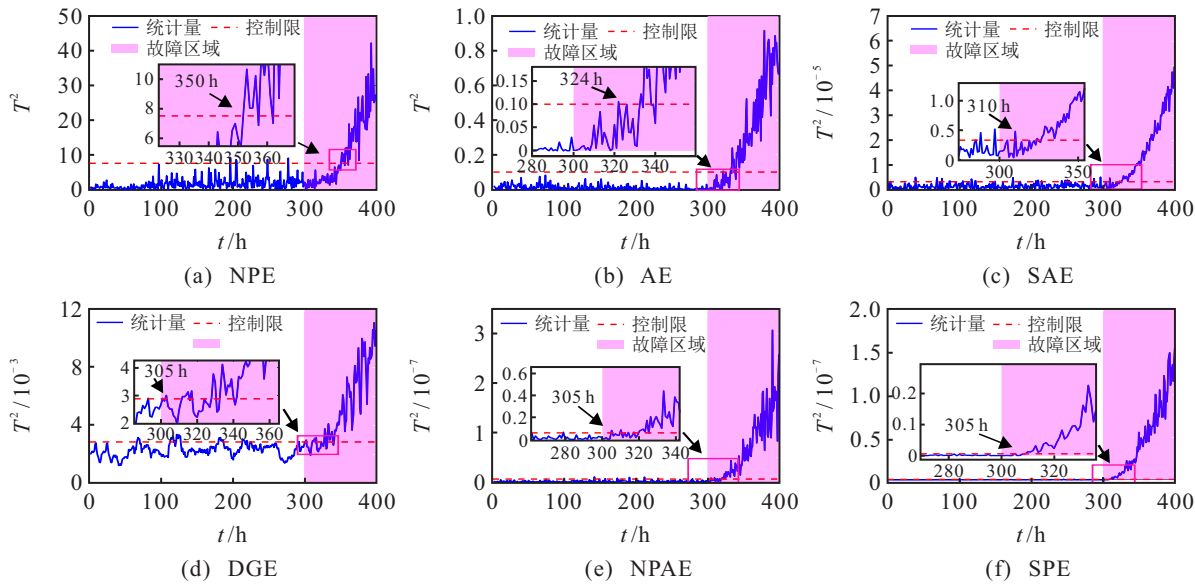


图5 T^2 统计量检测图

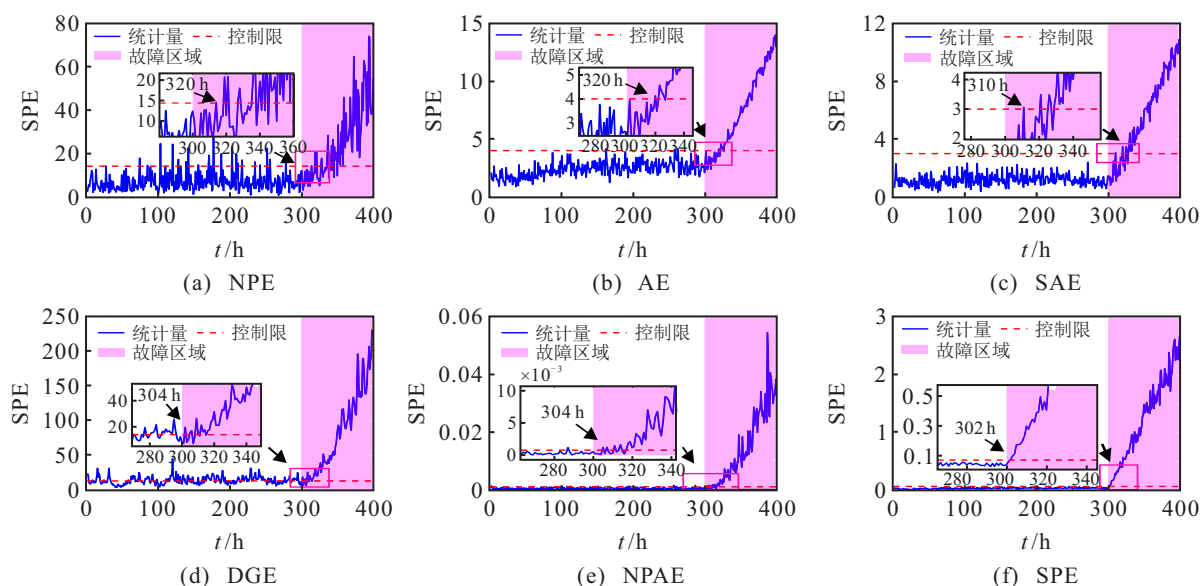


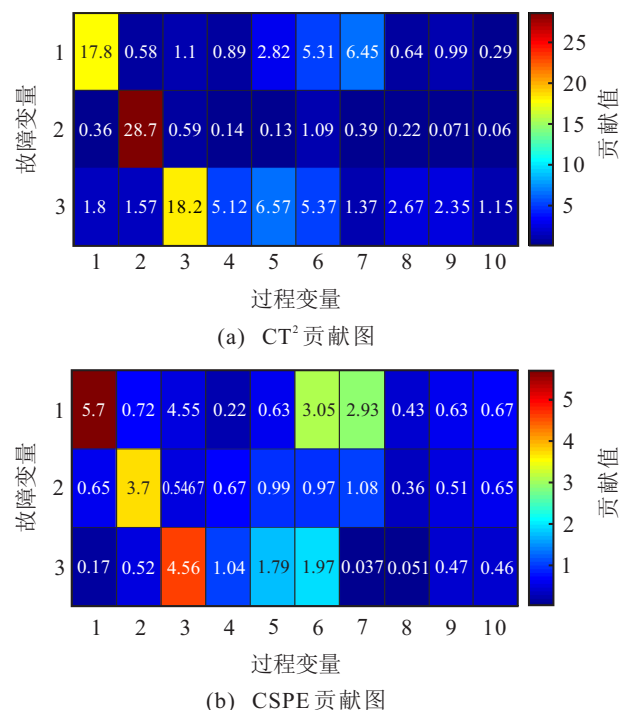
图6 SPE统计量检测图

来看统计量 T^2 和SPE均存在较多的误报,且当故障发生后有少量的漏报.图5(e)和图6(e)的NPAE算法在AE算法的基础上保留了原始数据的局部信息,检测到故障的时间较早, T^2 和SPE统计量分别在305h和304h检测到故障,由于算法鲁棒性不强仍然存在较多的漏报,相较于DGE算法漏报较少.图5(f)和图6(f)为本文所提S-RNPAE算法的检测图, T^2 和SPE统计量分别在305h和302h检测到故障,检测时间较其他算法更早.有效检测到早期故障的原因在于本文算法在保留数据原始局部结构信息的基础上,通过堆叠网络的方式提取深层特征,对早期故障更敏感,对青霉素发酵过程早期故障的检测更及时.同时,通过在目标函数的计算中采用 $L_{2,1}$ 范数增强算法对噪声和离群点的鲁棒性,使得检测的漏报和误报更少.

使用S-RNPAE有效检测到故障之后,对发生故障的变量通过 T^2 和SPE统计信息绘制贡献图.故障诊断采用贡献图的方式,图7所示为非线性贡献图故障诊断结果.从图7诊断结果来看,非线性贡献图不仅能够诊断出发生故障的变量,而且可以观察到过程变量的变化.

通过贡献图的冷暖色调可以观察到造成过程故障的根本原因和影响,故障 F_1 主要是通风速率所带来的影响,同时还导致发酵过程溶解氧的变化,当溶解氧浓度发生变化时,青霉素产率发生变化,反应容器体积发生改变;故障 F_2 主要是搅拌功率改变所引起的故障,其他过程变量整体变化较小;故障 F_3 是底物流速率故障,当底物流速发生变化时,菌丝生长所需的溶解氧浓度也发生变化.在故障发生后, CT^2 和

CSPE都可以根据所建立的基于S-RNPAE算法得到的非线性分量来定位对每个故障有贡献的变量.

图7 故障 $F_1 \sim F_3$ 的贡献图诊断结果

4 结论

本文提出一种堆叠鲁棒邻域自编码的间歇过程早期故障监测方法.该方法利用 $L_{2,1}$ 范数作为AE的目标函数,降低了监测过程中噪声和异常点的干扰,NPE作为AE的正则项能够保持数据的局部近邻结构,采用堆叠的方式获取数据的深层特征.对于检测到的故障采用一种新的贡献图的方法进行故障诊断,相较于传统贡献图而言,在诊断发生故障变量的

同时,可以观察到每个故障发生时过程变量的变化情况。本文方法在Swiss Roll数据集中有更好的特征提取能力,同时所提方法在青霉素发酵过程的监测中具有良好的表现,能够更好地监测到早期故障。

参考文献(References)

- [1] 孔祥玉, 杨治艳, 刘佑民, 等. 基于独立成分分析及其扩展模型的工业过程监测方法综述[J]. 控制与决策, 2022, 37(4): 799-814.
(Kong X Y, Yang Z Y, Liu Y M, et al. Overview of industrial process monitoring methods based on independent component analysis and its extended model[J]. Control and Decision, 2022, 37(4): 799-814.)
- [2] 刘强, 卓洁, 郎自强, 等. 数据驱动的工业过程运行监控与自优化研究展望[J]. 自动化学报, 2018, 44(11): 1944-1956.
(Liu Q, Zhuo J, Lang Z Q, et al. Perspectives on data-driven operation monitoring and self-optimization of industrial processes[J]. Acta Automatica Sinica, 2018, 44(11): 1944-1956.)
- [3] Zhang M, Li X J, Wang R Q. Incipient fault diagnosis of batch process based on deep time series feature extraction[J]. Arabian Journal for Science and Engineering, 2021, 46(10): 10125-10136.
- [4] Chen B, Luo X L. Incipient fault detection benefited from voting fusion strategy on analysis of process variation[J]. Chemometrics and Intelligent Laboratory Systems, 2021, 215: 104347.
- [5] Chen J, Liao C M, Lin F R J, et al. Principle component analysis based control charts with memory effect for process monitoring[J]. Industrial & Engineering Chemistry Research, 2001, 40(6): 1516-1527.
- [6] Stefatos G, Ben H A. Dynamic independent component analysis approach for fault detection and diagnosis[J]. Expert Systems with Applications, 2010, 37(12): 8606-8617.
- [7] Yin S, Ding S X, Zhang P, et al. Study on modifications of PLS approach for process monitoring[J]. IFAC Proceedings Volumes, 2011, 44(1): 12389-12394.
- [8] Harrou F, Madakyaru M, Sun Y. Improved nonlinear fault detection strategy based on the Hellinger distance metric: Plug flow reactor monitoring[J]. Energy and Buildings, 2017, 143: 149-161.
- [9] Harmouche J, Delpha C, Diallo D. Incipient fault detection and diagnosis based on Kullback-Leibler divergence using principal component analysis: Part II[J]. Signal Processing, 2015, 109: 334-344.
- [10] Ji H Q, He X, Shang J, et al. Incipient fault detection with smoothing techniques in statistical process monitoring[J]. Control Engineering Practice, 2017, 62: 11-21.
- [11] Shang J, Chen M Y, Ji H Q, et al. Recursive transformed component statistical analysis for incipient fault detection[J]. Automatica, 2017, 80: 313-327.
- [12] Liu F, Wang P L, Cai Z D, et al. Batch process fault diagnosis based on the combination of deep belief network and long short-term memory network[C]. 2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes. Xiamen, 2020: 208-214.
- [13] Yu J B, Zheng X Y, Liu J T. Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map[J]. Computers in Industry, 2019, 109: 121-133.
- [14] 高学金, 刘腾飞, 徐子东, 等. 基于循环自动编码器的间歇过程故障监测[J]. 化工学报, 2020, 71(7): 3172-3179.
(Gao X J, Liu T F, Xu Z D, et al. Intermittent process fault monitoring based on recurrent autoencoder[J]. CIESC Journal, 2020, 71(7): 3172-3179.)
- [15] 张海利, 王普, 高学金, 等. 基于批次图像化的卷积自编码故障监测方法[J]. 控制与决策, 2021, 36(6): 1361-1367.
(Zhang H L, Wang P, Gao X J, et al. Fault detection of batch image-based convolutional autoencoder[J]. Control and Decision, 2021, 36(6): 1361-1367.)
- [16] Gao X J, Xu Z D, Li Z, et al. Batch process monitoring using multiway Laplacian autoencoders[J]. The Canadian Journal of Chemical Engineering, 2020, 98(6): 1269-1279.
- [17] Agarwal P, Aghaee M, Tamer M, et al. A novel unsupervised approach for batch process monitoring using deep learning[J]. Computers & Chemical Engineering, 2022, 159: 107694.
- [18] Wang K, Gopaluni R B, Chen J, et al. Deep learning of complex batch process data and its application on quality prediction[J]. IEEE Transactions on Industrial Informatics, 2020, 16(12): 7233-7242.
- [19] Han Y K, Cheng Q S, Hou Y D. Fault detection method based on improved isomap and SVM in noise-containing nonlinear process[C]. 2018 International Conference on Control, Automation and Information Sciences. Hangzhou, 2018: 461-466.
- [20] 张鑫, 郭顺生, 李益兵, 等. 基于拉普拉斯特征映射和深度置信网络的半监督故障识别[J]. 机械工程学报, 2020, 56(1): 69-81.
(Zhang X, Guo S S, Li Y B, et al. Semi-supervised fault identification based on Laplacian eigenmap and deep belief networks[J]. Journal of Mechanical Engineering, 2020, 56(1): 69-81.)
- [21] Ma P, Zhang H L, Fan W H, et al. Early fault detection of bearings based on adaptive variational mode

- decomposition and local tangent space alignment[J]. Engineering Computations, 2019, 36(2): 509-532.
- [22] 姚红娟, 赵小强, 李炜, 等. 基于双权重多邻域保持嵌入的间歇过程故障检测[J]. 控制与决策, 2021, 36(12): 3023-3030.
(Yao H J, Zhao X Q, Li W, et al. Fault detection of batch process based on double weight and multiple neighborhoods preserving embedding[J]. Control and Decision, 2021, 36(12): 3023-3030.)
- [23] Liu C L, Wang K, Ye L J, et al. Deep learning with neighborhood preserving embedding regularization and its application for soft sensor in an industrial hydrocracking process[J]. Information Sciences, 2021, 567: 42-57.
- [24] Li N, Yan W W, Yang Y P. Spatial-statistical local approach for improved manifold-based process monitoring[J]. Industrial & Engineering Chemistry Research, 2015, 54(34): 8509-8519.
- [25] Yu J B, Zhang C Y. Manifold regularized stacked autoencoders-based feature learning for fault detection in industrial processes[J]. Journal of Process Control, 2020, 92: 119-136.
- [26] Zhang C Q, Liu Y Q, Fu H Z. AE2-nets: Autoencoder in autoencoder networks[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2020: 2572-2580.
- [27] He X F, Cai D, Yan S C, et al. Neighborhood preserving embedding[C]. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Beijing, 2005: 1208-1213.
- [28] Bounoua W, Bakdi A. Fault detection and diagnosis of nonlinear dynamical processes through correlation dimension and fractal analysis based dynamic kernel PCA[J]. Chemical Engineering Science, 2021, 229: 116099.
- [29] Cui J, Shi G, Zhang Z R. Fault detection of aircraft generator rotating rectifier based on SAE and SVDD method[C]. 2017 Prognostics and System Health Management Conference. Harbin, 2017: 1-5.
- [30] Zhao H T. Dynamic graph embedding for fault detection[J]. Computers & Chemical Engineering, 2018, 117: 359-371.
- [31] Lee Y, Kwon H, Park F. Neighborhood reconstructing autoencoders[J]. Advances in Neural Information Processing Systems, 2021, 34: 536-546.
- [32] Birol G, Ündey C, Parulekar S J, et al. A morphologically structured model for penicillin production[J]. Biotechnology and Bioengineering, 2002, 77(5): 538-552.

作者简介

刘凯 (1996—), 男, 博士生, 从事间歇过程故障检测与诊断等研究, E-mail: lutlk196@163.com;

赵小强 (1969—), 男, 教授, 博士生导师, 从事故障诊断、图像处理、数据挖掘等研究, E-mail: xqzhao@lut.edu.cn;

牟淼 (1997—), 男, 博士生, 从事复杂工业过程故障诊断与质量预测等研究, E-mail: m_miao7@163.com;

张妍 (1991—), 女, 博士生, 从事间歇过程故障检测与诊断等研究, E-mail: zhylut@163.com.

Fault monitoring of batch process based on multi-stage optimization regularized neighborhood preserving embedding algorithm

Transactions of the Institute of

Measurement and Control

2023, Vol. 45(1) 89–103

© The Author(s) 2022

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/01423312221108519

journals.sagepub.com/home/tim



Xiaoqiang Zhao^{1,2,3} , Kai Liu¹ and Yonyong Hui^{1,2,3}

Abstract

Batch process is an important type of industrial production process, and the process mechanism is complex. It is difficult to accurately describe the dynamic changes of the production process of multi-stage time-varying batch process. In addition, the data of batch process contain not only global information but also local information. The traditional neighborhood preserving embedded algorithm is used to maintain the local geometric structure of data while ignoring the global information, and the extracted latent variables cannot fully characterize batch process. Therefore, we propose a multi-stage optimization regularized neighborhood preserving embedding (ORNPE) algorithm. First, the multiple process stages are separated by affinity propagation (AP) algorithm. Second, based on maintaining local information of neighborhood preserving embedding algorithm, slow feature analysis algorithm is used to extract dynamic time-varying global information. Then, cross-entropy is used to optimize the global information, and the extraction ability of the global information is improved. Finally, a monitoring index based on support vector data description is constructed to eliminate adverse effects of non-Gaussian data for monitoring performance. The effectiveness and advantages of the proposed algorithm based on monitoring strategy are illustrated by the penicillin fermentation process and a semiconductor industry process.

Keywords

Batch process, multi-stage, fault detection, affinity propagation clustering, neighborhood preserving embedding

Introduction

Batch process is widely used in semiconductor, pharmaceutical, injection molding, and other production processes due to its high added value, small production batches, and meeting individual needs. It is a crucial production mode in modern manufacturing (Luo and Bao, 2018; Qin, 2012). At present, as the complexity of the production process increases and the scale continues to expand, the production process also contains vast safety risks and the probability of faults continues to grow (Lavanya et al., 2021; Prasanth, 2021). Therefore, real-time monitoring of process ensures that faults can be detected timely, and accurate fault detection has essential economic value and practical significance (Fu and Zhang, 2017; Jiang et al., 2020; Zhang et al., 2018, 2019, 2020).

With the development of advanced control systems, a large amount of process data is collected and stored, which provides a development basis for data-driven modeling and monitoring technology. Multivariate statistics methods (e.g. Principal Component Analysis (PCA; Abdi and Williams, 2010; Cotrufo and Zmeureanu, 2016)) and Partial Least Squares (PLS; Helland, 2014; Li, 2010) have been widely used in process monitoring. These methods have good performances for dimension reduction and process monitoring. A strategy of multi-directional expansion data processing is used to obtain Multiway Principal Component Analysis

(MPCA; Jeffy et al., 2018; Majid et al., 2011) and Multiway Partial Least Squares (MPLS; Camarrone and Van Hulle, 2018; Wang et al., 2016), which are applied in fault monitoring of the process. Besides, some experts and scholars have done a lot of researches (Jiang and Yin, 2018; Peng et al., 2020; Zhang et al., 2021a, 2021b). Although these researches have achieved good results in fault monitoring, they usually consider the global structure of data and ignore the local feature information.

However, He et al. (2005) proposed Neighborhood Preserving Embedding (NPE) algorithm based on Locally Linear Embedding (LLE; Roweis and Saul, 2000). Compared with LLE algorithm, NPE algorithm can more accurately

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

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

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

Corresponding author:

Xiaoqiang Zhao, College of Electrical and Information Engineering, Lanzhou University of Technology, 287, Langongping Road, Qilihe District, Lanzhou 730050, China.

Email: xqzhao@lut.edu.cn

obtain the mapping matrix and has better practicability. In process monitoring, unlike PCA, PLS, and other global structure preserving algorithms, NPE maintains the neighborhood structure by projecting neighboring points in high-dimensional space to low-dimensional space, and pays more attention to the local structure information.

Generally speaking, the occurrence of faults often leads to changes of the global and local structure information. In this sense, it is essential to consider both the global structure information and the local structure information. Therefore, only focusing on the global structure or the local structure can hardly reflect the actual working conditions of batch process. To extract the global and local information of batch process simultaneously, Zhang et al. (2011) proposed a Global-Local Structure Analysis (GLSA) algorithm that considered the extraction of global structure and local structure features. On this basis, Yu (2016) proposed a monitoring model of local and global principal component analysis (LGPCA) algorithm, which could better extract the global and local structure. Zhao et al. (2016) proposed a Global Neighborhood Preserving Embedding (GNPE) algorithm in fault detection of batch process, which combined PCA and NPE algorithm. Xu and Ding (2021) proposed a Manifold Regularized Slow Feature Analysis (MRSFA) algorithm, which gave sufficient consideration to the global time change and local structure information of original data: so that the extracted latent variables can more truly represent the process data. These algorithms preserve the global and local information of the data through different strategies and also achieve better fault monitoring results. However, the lack of process optimization for preserving global information mediately loses part of the data information. As a result, some faults are not detected, and the effect of fault monitoring is directly affected. Because each stage of batch process has its own process mechanism, different stages show different process characteristics. Therefore, batch process has a multi-stage characteristic. However, traditional algorithms such as MPCA, MPLS, Multiway Neighborhood Preserving Embedding (MNPE), and their improved algorithms regard batch process as a single stage in the monitoring process. They do not consider the multi-stage information of batch process. In order to improve the monitoring performance of batch process under the influence of multi-stage characteristic, some researchers have proposed different modeling methods to strengthen process monitoring. Ningyun et al. (2010) used K-Means (KM) algorithm to divide the process into multiple stages through clustering. Zhao and Sun (2013) proposed a step-wise sequential phase division method. Liu et al. (2016) proposed a step-wise sequential phase division method based on windows. Gao et al. (2014) proposed a multi-stage modeling method by using fuzzy clustering, and process data could be divided into multiple categories simultaneously to complete the stage division. Guo et al. (2017) divided the stages according to the characteristics of different stages of batch process. Ge et al. (2012) proposed a new staged method based on a defined repeatability factor. However, the above clustering methods need to set the clustering model parameters in advance during the stage division process. The

manual set parameters affect the accuracy of stage segmentation and cannot accurately reflect the stage characteristic of the process.

To solve the accuracy problem of stage division caused by the manual setting of parameters, affinity propagation (AP) clustering algorithm is used to classify data without knowing the number of classifications in advance, which can divide batch process into different operating stages. We utilize Dynamic Time Warping (DTW) algorithm to process the data of different batches at the same stage with equal length. The divided sub-stages have similar data structures and characteristics, which are convenient for statistical monitoring. In each sub-stage, we use ORNPE algorithm for dimension reduction and feature extraction. Fault detection is performed by constructing statistics of T^2 and R^2 based on the multi-stage ORNPE algorithm. Finally, the Penicillin fermentation process and semiconductor etching process are used to verify the effect of the proposed algorithm on batch process monitoring.

Batch process is a typical multi-stage process. The data at the same stage have similar characteristics and correlations which can establish a unified model to ensure that the established monitoring model has the smallest error. In process monitoring, the traditional NPE algorithm only considers the local manifold structure of the data, but ignores the global information of the data, resulting in incomplete information. Therefore, it is necessary to improve NPE algorithm to extract important global information while extracting local information.

According to the aforementioned problems, we propose a multi-stage ORNPE algorithm for batch process monitoring. The contributions of this paper are given as follows:

1. AP clustering algorithm is used to divide batch process into stages. Due to the difference in the stage division of each batch, DTW algorithm processes the same stage of different batches with equal length.
2. A new dimensionality reduction algorithm named ORNPE is proposed to extract latent variables in each stage, which considers the global dynamic information and the local geometric structure information of the original data, consequently providing a more faithful low-dimensional representation of the original data.
3. At each stage, since the original data is non-Gaussian, based on the ORNPE algorithm, SVDD algorithm is used to construct statistical indicator to avoid the adverse effects of the non-Gaussian distribution of process data.

The rest of this paper is organized as follows. The section "Preliminaries" introduces the principle of NPE algorithm, AP clustering algorithm, and slow feature analysis algorithm. The section "Multi-stage monitoring process based on ORNPE algorithm" proposes a multi-stage process monitoring method based on ORNPE algorithm. The proposed algorithm is demonstrated through the penicillin fermentation process and the semiconductor industry process in the section "Case studies." The section "Conclusion" concludes the work.

Preliminaries

NPE

NPE (He et al., 2005) is a manifold learning algorithm that approximates LLE algorithm. It can mine local topological structures and high-dimensional structure relationships. The basic idea is to linearize the original training data set $X(x_1, x_2, \dots, x_n) \in R^D$ in high-dimensional space through the nearest neighbors. The dimensionality is reduced through mapping matrix from high-dimension to low-dimensional space. The obtained data set is as follows: $Y = (y_1, y_2, \dots, y_d)$, where $A(a_1, a_2, \dots, a_d)(d \leq D)$ is a mapping from high-dimension space to low-dimensional space and satisfies $Y = A^T X$. Specific steps are as follows:

Step 1: Construct a neighborhood graph

The Euclidean distance between samples in the training data set $X(x_1, x_2, \dots, x_n) \in R^D$ is arranged by the K Nearest Neighbor (KNN) method in ascending order. The first k points of each sample are selected as neighbor points, and a neighborhood graph is formed by comparing and connecting.

Step 2: Calculate the weight matrix

If the sample point x_j is the neighbor point corresponding to the selected connection, its weight coefficient is w_{ij} . And if there is no connection, then $w_{ij} = 0$. All the weight coefficients form the weight matrix W . The weight matrix W can be obtained by minimizing the reconstruction error

$$\Phi(W) = \min \sum_{i=1}^n \left\| x_i - \sum_{j=1}^n w_{ij} x_j \right\|^2 \quad (1)$$

The normalization constraint is $\sum_{j=1}^n w_{ij} = 1$ that the sum of the weights meets the normalization condition, and the result is 1.

Step 3: Calculate the mapping matrix

The mapping matrix A is obtained by solving equation (2) to minimize the cost function

$$\begin{aligned} f(A) &= \sum_{i=1}^n \left(y_i - \sum_{j=1}^k w_{ij} y_j \right)^2 \\ &= y^T (I - W)^T (I - W) y \\ &= a^T X (I - W)^T (I - W) X^T a \end{aligned} \quad (2)$$

That is $A = \arg \min_a^T X(I - W)^T (I - W) X^T a$, where the constraint is $y^T y = a^T X X^T a = 1$.

We introduce the Lagrange multiplier method to transform the solution of equation (2) into the generalized eigenvalues of equation (3)

$$X M X^T a = \lambda X X^T a \quad (3)$$

where $M = (I - W)^T (I - W)$. We can find the eigenvectors corresponding to the first d smallest eigenvalues

$(\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_d)$ from the mapping matrix $A(\alpha_1, \alpha_2, \dots, \alpha_d)$, which satisfies $Y = A^T X$.

AP clustering

AP clustering (Frey and Dueck, 2007) was a clustering method proposed by Frey and Dueck in 2007. This algorithm does not need to set the number of clusters in advance, but automatically obtains clusters based on the cluster samples in the iterative process. Clustering centers are clustered according to the similarity between data objects. Before clustering, all sample points are regarded as potential cluster centers, and the similarity calculation performs in a loop. Each sample point converges through iteration to obtain the final representative point set of the best category.

Given a data set $X = \{x_1, x_2, \dots, x_i\}$, where x_i represents a sample point and d represents the dimension of the data point. There is $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$. The calculation steps of AP clustering algorithm are as follows:

Step 1: Calculate similarity matrix

The similarity between samples selects different measurement criteria according to different scenarios, such as Euclidean distance. Depending on the similarity criteria, the similarity between samples may be symmetrical or asymmetrical. These similarities form a similarity matrix with $N \times N$ dimension similarity matrix S (N is the number of data samples). Here, we choose the square of the negative Euclidean distance to calculate the similarity and construct the similarity matrix, that is, $s(i, k) = -\|x_i - x_k\|^2$, similarity matrix S is shown as equation (4)

$$S = \begin{bmatrix} s(x_1, x_1) & s(x_1, x_2) & \dots & s(x_1, x_N) \\ s(x_2, x_1) & \ddots & \dots & s(x_2, x_N) \\ \vdots & \vdots & s(x_k, x_k) & \vdots \\ s(x_N, x_1) & s(x_N, x_2) & \dots & s(x_N, x_N) \end{bmatrix} \quad (4)$$

Step 2: Divide clusters based on cluster centers

To select the appropriate cluster centers, the degree of attribution $a(i, k)$ and the degree of attraction $r(i, k)$ are defined. The degree of attribution $a(i, k)$ indicates that the data point x_i belongs to one of the categories represented by the data point x_k . The stronger the numerical information expressed by $a(i, k)$ and $r(i, k)$, the greater the probability that the point is the center of the cluster. The degree of attribution $a(i, k)$ and the degree of attraction $r(i, k)$ are shown in equations (5) and (6), and the initial value is 0

$$a(i, k) = \begin{cases} \min \left\{ 0, r(k, k) + \sum_{i' \notin [i, k]} \max \{ 0, r(i', k) \} \right\}, & i \neq k \\ \sum_{i' \neq k} \max \{ 0, r(i', k) \}, & i = k \end{cases} \quad (5)$$

$$r(i, k) = s(i, k) - \max_{k' \neq k} \left\{ a(i, k') + s(i, k') \right\} \quad (6)$$

The damping coefficient is added to adjust the convergence speed during each iteration update, which prevents oscillations during the iteration process. By repeating the above steps, equations (7) and (8) are updated

$$r_{t+1}(i, k) = \lambda \times r_t(i, k) + (1 - \lambda) \times r_{t+1}(i, k) \quad (7)$$

$$a_{t+1}(i, k) = \lambda \times a_t(i, k) + (1 - \lambda) \times a_{t+1}(i, k) \quad (8)$$

Cluster center point H is obtained. The original data set is divided into H clusters through cluster center point H .

Slow feature analysis

Slow feature analysis (SFA; Shang et al., 2016) is a dimensionality reduction algorithm, which can find a projection matrix to convert time series data into latent variables that change as slowly as possible, which latent variables contain the main information of the original data. The specific steps are as follows:

Given an m -dimensional input signal $x(t) = [x_1(t), x_2(t), \dots, x_i(t), \dots, x_m(t)]^T$, where $t \in [t_0, t_1]$ represents time, SFA seeks a transformation function $g(x) = [g_1(x), \dots, g_J(x)]^T$ to change the output $y(t) = [y_1(t), \dots, y_J(t)]^T$ of the function as slowly as possible, where $y_j(t) = g_j(x(t))$. By minimizing the variance of the first derivative of the slow feature, the optimization of SFA is derived

$$\min \langle \dot{y}_i^2 \rangle_t \quad (9)$$

And, three constraints are derived

$$\langle y_i \rangle_t = 0 \quad (10)$$

$$\langle y_i^2 \rangle_t = 1 \quad (11)$$

$$\forall i \neq j, \langle y_i y_j \rangle_t = 0 \quad (12)$$

where \dot{y}_i represents the first-order derivative of the slow feature and $\langle \rangle_t$ represents the average over time, which is defined as equation (13)

$$\langle f \rangle_t = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t) dt \quad (13)$$

The constraint condition of equation (10) ensures that the mean value of the extracted slow feature information is zero. The purpose of equation (11) is to exclude the trivial solution of the output signals. The constraint condition of equation (12) guarantees that the output signals are multiple. The components are uncorrelated so that they carry different aspects of information.

The slow feature of x can be expressed as y . The projection matrix $A = [\alpha_1, \alpha_2, \dots, \alpha_I]$ is obtained by two-step singular value decomposition.

To whiten the slow feature, singular value decomposition of the covariance matrix of the original data is performed, and shown as equation (14)

$$\langle xx^T \rangle_t = U \sum U^T \quad (14)$$

where U represents the eigenvector and \sum represents the eigenvalue. The data after whitening is c and meets $\text{cov}(c) = \langle cc^T \rangle_t = I$, and the data after whitening eliminates the cross-correlation between variables. The covariance of the first derivative of c is found, and the second step of singular value decomposition is performed, as equation (15)

$$\langle \dot{c} \dot{c}^T \rangle_t = \Xi^T \Omega \Xi \quad (15)$$

The slow feature of x is obtained by equation (16)

$$y = \Xi c = \Xi^{-1} \sum U^T x \quad (16)$$

where $\Omega = \text{diag}\{w_1, w_2, \dots, w_m\}$ is composed of eigenvalues and arranged on its diagonal in ascending order to satisfy $w_i = \langle \dot{y}_i^2 \rangle_t$, thus it ensures that the slowest feature has the lowest exponent. The matrix Ξ is composed of eigenvectors corresponding to eigenvalues.

Multi-stage monitoring process based on ORNPE algorithm

This section introduces the process of multi-stage fault monitoring based on ORNPE algorithm: including data preprocessing, stage division, algorithm model and establishment of statistical indicators.

Three-dimensional expansion of batch process data

Compared to continuous process, batch process has three-dimensional data. When fault diagnosis of batch process is performed, it is necessary to unfold the three-dimensional data into the two-dimensional data to establish a fault diagnosis model. Therefore, the three-dimensional data expansion method is used to process the three-dimensional data. First, the three-dimensional data are unfolded as two-dimensional data along the batch direction, as shown in equation (17), then, it is standardized by column, and the standardized two-dimensional matrix is arranged according to the variable direction to form a two-dimensional matrix, as shown in equation (18). The expansion method can eliminate data pre-estimation problems during online monitoring and can monitor process changes more effectively

$$X(I \times KJ) = \begin{bmatrix} x_{1,1}^1, x_{2,1}^1 \cdots x_{J,1}^1 \cdots x_{1,K}^1, x_{2,K}^1 \cdots x_{J,K}^1 \\ x_{1,1}^2, x_{2,1}^2 \cdots x_{J,1}^2 \cdots x_{1,K}^2, x_{2,K}^2 \cdots x_{J,K}^2 \\ \vdots \quad \quad \quad \ddots \quad \quad \quad \vdots \\ x_{1,1}^I, x_{2,1}^I \cdots x_{J,1}^I \cdots x_{1,K}^I, x_{2,K}^I \cdots x_{J,K}^I \end{bmatrix} \quad (17)$$

where $x_{j,k}^i$ represents the variable x_j of the i th batch at the k th sampling time

$$X(KI \times J) = \begin{bmatrix} [(x_{1,1}^1, x_{1,1}^2, \dots, x_{1,1}^J)]^T & \dots & [(x_{1,1}^1, x_{1,1}^2, \dots, x_{1,1}^J)]^T \\ [(x_{2,1}^1, x_{2,1}^2, \dots, x_{2,1}^J)]^T & \dots & [(x_{2,1}^1, x_{2,1}^2, \dots, x_{2,1}^J)]^T \\ \vdots & \ddots & \vdots \\ [(x_{K,1}^1, x_{K,1}^2, \dots, x_{K,1}^J)]^T & \dots & [(x_{K,1}^1, x_{K,1}^2, \dots, x_{K,1}^J)]^T \end{bmatrix} \quad (18)$$

where $x_{k,i}^j$ represents the variable x_j of the i th batch at the k th sampling time.

The process of data obtained by the expansion method is shown in Figure 1 (I represents the batch of data, J represents the variable of data, K represents the sample time of data).

Stage division based on AP clustering

After three-dimensional data are processed, AP clustering algorithm is used to divide the process data into different stages. The steps of AP clustering are as follows:

- Step 1: Divide data into the matrix $X = (x^1, \dots, x^i, \dots, x^J)^T$, where x^i represents the i th batch of data.
- Step 2: Slice X along time to get K time slices and calculate the similarity matrix S according to equation (4).
- Step 3: Take $i(i = 1, 2, \dots, I)$ batch data as the AP clustering input and replace $s(k, k)$ in the similarity matrix $s(i, k)$ with the optimal bias parameter p . We take the average of all similarities as the setting value of p , and the number of clusters obtained at this time is medium. If the value of the minimum similarity is taken as the value of p , the number of clusters obtained is small. The AP clustering bias parameter divides the batch process into stages. The process is initially divided into n stages.

The stage division of each batch is shown in Figure 2.

Sub-stage unequal length processing based on DTW algorithm

DTW is an algorithm of speech recognition technology. The algorithm can find the optimal curve path, and match the coordinate points in one time series with the coordinate points with the most significant feature similarity in another time series.

First, for two unequal length series, the distance between two series as matrix d and the distance between series g_i and h_i are defined as equation (19)

$$d(g_i, h_j) = \|g_i - h_j\|_p \quad (19)$$

where $\|\cdot\|_p$ represents the norm and p usually takes the value 2.

DTW algorithm synchronizes two unequal length time series by searching for the shortest distance and the optimal path, and optimization performance index of the shortest distance is shown as equations (20) and (21)

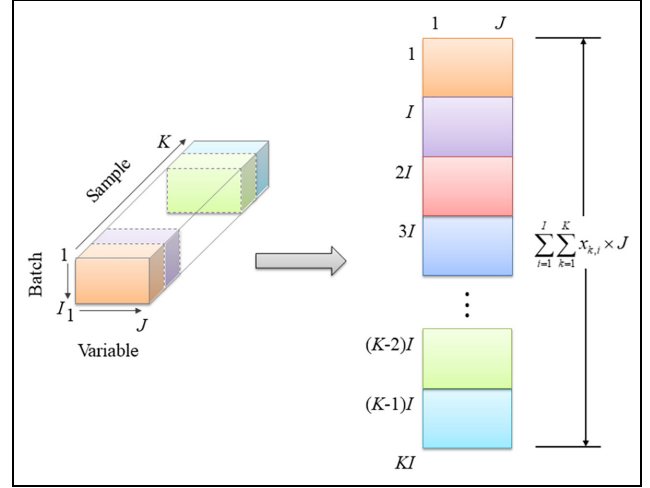


Figure 1. The process of three-dimensional variable expansion.

$$D(m, n) = \min_F \left[\sum_{l=1}^L d(i(l), j(l)) \right] \quad (20)$$

$$W = \arg \min_F [D(m, n)] \quad (21)$$

where $\max(m, n) \leq L \leq m + n$, $D(m, n)$ is the sum of the two series of local shortest distances along the optimal path, the optimal path $W = \{w_1, w_2, \dots, w_L\}$ $w_l = (i(l), j(l))$ is a series searched in the $m \times n$ grid based on the shortest distance $D(m, n)$.

To avoid excessive data distortion and skipping, the W setting of each step is continuous. There are three options for the points before the grid point: point $(i-1, k)$, point $(i-1, k-1)$, point $(i-1, k-2)$; therefore, equation (20) can be obtained by equation (22)

$$D(i, k) = \begin{cases} D(i, k) + d(i, k) \\ D(i-1, k-1) + d(i, k) \\ D(i-1, k-2) + d(i, k) \end{cases} \quad (22)$$

After the cumulative distance matrix $D(m, n)$ is determined, the last element is used to find the point where the optimal path passes gradually. Finally, we find the corresponding data to be synchronized through the optimal path value. The newly obtained series is a synchronized series. DTW algorithm is used to regularize the sub-stages of different batches, and the data of each batch after equal length is shown in Figure 3.

Monitoring scheme based on ORNPE

Cross-entropy. Cross-entropy represents the similarity of two probability density functions, and it has high efficiency for multi-objective global optimization. It can update the rules

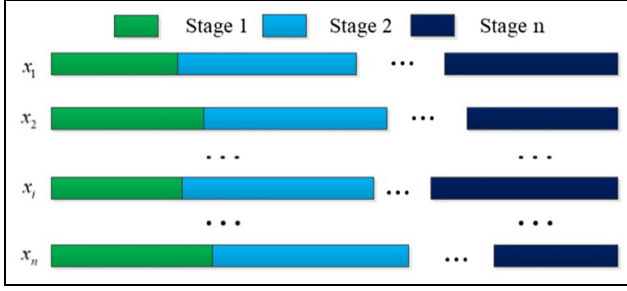


Figure 2. Unequal length of batch data.

through analysis and calculation to make the optimization path more efficient and faster.

Cross-entropy is used to globally optimize the two-dimensional data matrix and feature reduction through multiple iterations. Defining the probability density functions as f and g , the definition of cross-entropy is given in equation (23)

$$\begin{aligned} D(f(x) \parallel g(x)) &= \sum_{x \in X} f(x) \ln \frac{f(x)}{g(x)} \\ &= E_f(x) \ln \frac{f(x)}{g(x)} \end{aligned} \quad (23)$$

where $E()$ is the mathematical expectation.

From equation (23), we know that $D(f(x) \parallel g(x)) \geq 0$, because cross-entropy is a downward convex function, there is only $f(x) = g(x)$, which meets $D(f(x) \parallel g(x)) = 0$. Therefore, cross-entropy characterizes the similarity of the two distributions. The more similar the two distributions, the smaller the cross-entropy between them. For matrix $A(M \times N) f(i, j) (\geq 0)$, the definition of two-dimensional entropy of the matrix is given by equation (24)

$$H = - \sum_{i=1}^M \sum_{j=1}^N p_{ij} \log p_{ij} \quad (24)$$

$$p_{ij} = \frac{f(i, j)}{\sum_{i=1}^M \sum_{j=1}^N f(i, j)} \quad (25)$$

The iterative optimization of cross-entropy can significantly improve the global structure of the matrix after dimensionality reduction, thereby improving the fault detection rate.

Regularized NPE. When a fault occurs, it usually causes the global and local structure of normal data to change. Only considering local or global information would inevitably lead to the loss of some information. Therefore, in the process of fault diagnosis, global structure information and local structure information have the same importance.

SFA algorithm only reveals the time change of the global information without considering the local geometric structure. NPE algorithm only considers the local structure. Therefore, to consider the global and local information of the process data simultaneously, a global-local feature extraction algorithm is constructed through SFA and NPE, which

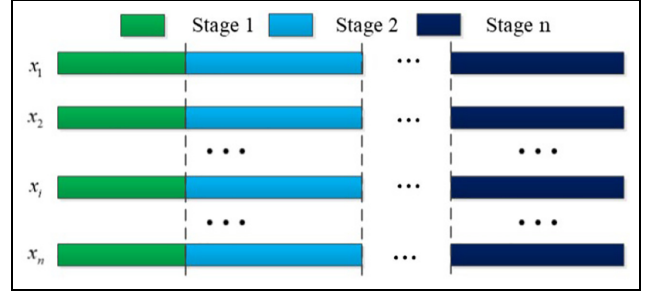


Figure 3. Each batch of data after the equal length.

comprehensively considers the global time change and local structure information of original data.

Equation (9) can be rewritten as equation (26) according to the trace

$$J_{SFA} = \sum_{i=1}^n \|\dot{y}_i\|^2 = \text{tr}(\dot{Y}^T \dot{Y}) = \text{tr}(K^T (\dot{Z}^T \dot{Z}) K) \quad (26)$$

where K is the projection matrix, $\text{tr}()$ denotes the trace of a matrix, and $\dot{Z} = [\dot{z}_1, \dot{z}_2, \dots, \dot{z}_{n-1}]^T$, evidently, $\dot{Y}^T \dot{Y} = K^T (\dot{Z}^T \dot{Z}) K = K^T K = I$.

We introduce cross-entropy algorithm in the training set and calculate the value of the objective function corresponding to each sample. We sort from small to large and update the probability distribution function by using the sample features with the larger function value. In this way, the sample quality is continuously improved globally, and finally, the optimal value is obtained

$$J_{SFA} = \sum_{i=1}^n \|\dot{y}_i\|^2 S_{ij}^p \quad (27)$$

where S_{ij}^p is the corresponding cross-entropy probability matrix.

Because the data are sampled at discrete intervals, therefore, the first-order derivative of the collected data can be approximate as the corresponding first-order difference

$$\dot{z}_j(t) \approx \frac{z_j(t) - z_j(t - \Delta t)}{\Delta t} \quad (28)$$

Therefore, the objective function of ORNPE algorithm is given by equation (29)

$$\begin{aligned} & \argmin \left(\sum_{i=1}^{n-1} \|\dot{y}_i\|^2 S_{ij}^p + \eta \sum_{i=1}^{n-1} \left\| y_i - \sum_{j=1}^{n-1} w_{ij} y_j \right\|^2 \right) \\ &= \text{tr}(\dot{Y}_s^T \dot{Y}_s) + \eta \text{tr}(Y^T (I - W)^T (I - W) Y) \\ &= \text{tr}(K^T (\dot{Z}^T \dot{Z}) K) + \eta \text{tr}(K^T (Z^T M Z) K) \\ &= \text{tr}(K^T H K) \end{aligned} \quad (29)$$

where η is the tuning parameter, $H = \dot{Z}^T \dot{Z} + \eta Z^T M Z$, and $M = (I - W)^T (I - W)$. Equation (29) can be obtained by seeking the generalized characteristic solution of equation (30)

$$H K = K \Psi \quad (30)$$

According to the reference (Xu and Ding, 2021), the tuning parameters η can be determined by the spectral radius, shown as equation (31)

$$\eta = \frac{\rho(\Phi)}{\rho(G) + \rho(\Phi)} \quad (31)$$

where $G = \dot{Z}^T \dot{Z}$, $\Phi = Z^T M Z$, and $\rho()$ denotes the spectral radius of a matrix.

Monitoring statistics based on Support Vector Data Description algorithm

The goal of Support Vector Data Description (SVDD) algorithm is to find a hypersphere with the smallest volume that contains all or most of the training data. A relaxation factor ξ_i and a penalty factor C are introduced to successfully solve outliers' influence caused by measurement errors and noise interference. Here, we apply SVDD algorithm to monitor the data distribution in the projection matrix $Y = [y_1, y_2, \dots, y_i, \dots, y_n]$, and monitoring statistics are constructed. For the projection matrix Y of the normal working condition data, the SVDD optimization problem is given in equation (32)

$$\begin{aligned} \min_{R, a, \xi_i} & \left(R^2 + C \sum_i \xi_i \right) \\ \text{s.t.} & \|\phi(x_i) - \varepsilon\| \leq R^2 + \xi_i \end{aligned} \quad (32)$$

where ε is the center of the sphere and R is the radius of the hypersphere $\xi_i \geq 0, i = 1, 2, 3, \dots, n$.

The optimization problem of equation (33) is transformed into a dual form, shown as equation (34)

$$\begin{aligned} \min_{\alpha_i} & \sum_{i=1}^n \alpha_i \langle \phi(x_i) \times \phi(x_i) \rangle - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \langle \phi(x_i) \times \phi(x_j) \rangle \\ \text{s.t.} & \sum_i \alpha_i = 1, 0 \leq \alpha_i \leq C \end{aligned} \quad (33)$$

where α is the Lagrangian factor. The kernel function $K(x_i \times x_j)$ is used to replace the inner product $\langle \phi(x_i) \times \phi(x_j) \rangle$ to realize the linear conversion of the nonlinear problem in the high-dimensional kernel space and obtain the dual equation (34)

$$\begin{aligned} \min_{\alpha_i} & \sum_i \alpha_i K(x_i \times x_i) - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(x_i \times x_j) \\ \text{s.t.} & \sum_i \alpha_i = 1, 0 \leq \alpha_i \leq C \end{aligned} \quad (34)$$

The radius R and the ε core of the hypersphere are shown as equations (35) and (36)

$$\varepsilon = \sum_{i=1}^n \alpha_i \phi(x_i) \quad (35)$$

$$R^2 = K(x_k \times x_k) - 2 \sum_{i=1}^n \alpha_i K(x_k \times x_i) + \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(x_i \times x_j) \quad (36)$$

where x_k is the support vector of SVDD.

The distance between the new sample and the center of the hypersphere can be expressed by equation (37)

$$\begin{aligned} R_{\text{new}}^2 &= \|\phi(x_{\text{new}}) - \varepsilon\|^2 \\ &= K(x_{\text{new}} \times x_{\text{new}}) - 2 \sum_{i=1}^n \alpha_i K(x_{\text{new}} \times x_i) \\ &\quad + \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(x_i \times x_j) \leq R^2 \end{aligned} \quad (37)$$

If $R_{\text{new}}^2 \leq R^2$, the sample is normal, otherwise, it is abnormal.

Monitoring process

Offline training

Step 1: Under normal working conditions, obtain training sample set $X \in R^{m \times n}$, and the three-dimensional data are unfolded according to batch-variable method and standardized. The method for training sample $X(I \times J \times K)$, first, along the batch direction, according to equation (17), a two-dimensional matrix $X(I \times JK)$ is obtained, and then it is arranged as $X(IK \times J)$ along the variable path according to equation (18).

Step 2: According to section "Stage division based on AP clustering," AP clustering algorithm is used to divide multiple batch processes into stages, and DTW algorithm is used to perform equal length processing on the divided sub-stages.

Step 3: According to section "Monitoring scheme based on ORNPE," ORNPE algorithm is used for feature extraction for each division stage to obtain the low-dimensional structure $Y = [y_1, y_2, \dots, y_d]$.

Step 4: The kernel density estimation method is used to obtain the control limits of offline training samples.

Online monitoring

Step 1: Process data are collected online, unfolded, and standardized by using the batch-variable method. The pre-processed test data are obtained.

Step 2: The preprocessed data are divided into stages according to the stage division in the offline modeling process.

Step 3: For each stage, ORNPE algorithm is used to obtain the low-dimensional structure y_{new} , in which the R^2 and T^2 statistics of the online data are obtained.

Step 4: The online statistics are compared with the control limits to determine whether the limits are exceeded. If the control limit is exceeded, the fault occurs, otherwise, go back to Step 1.

The fault detection of batch process based on multi-stage ORNPE algorithm includes offline modeling and online detection process. The flow chart is shown in Figure 4.

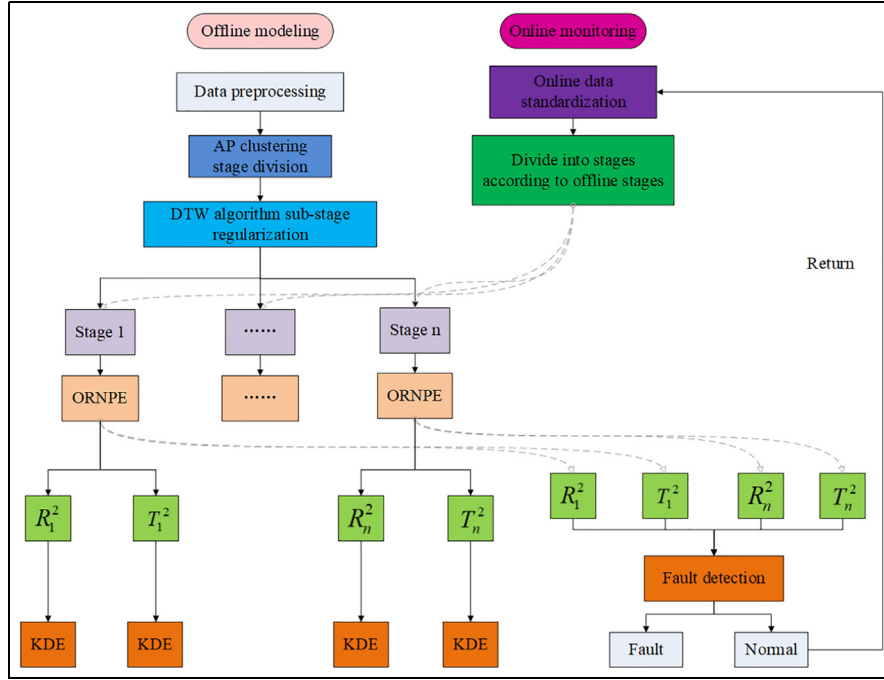


Figure 4. Flow chart of multi-stage ORNPE algorithm.

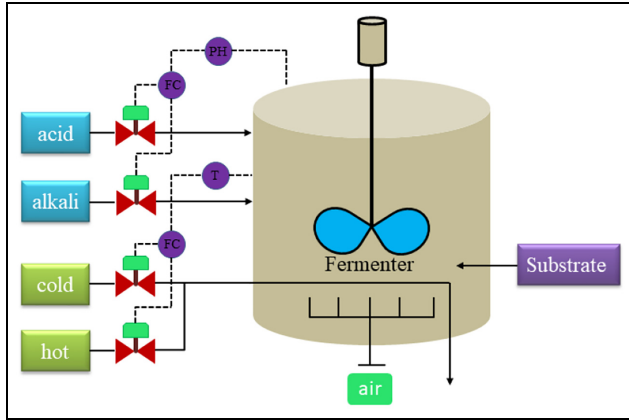


Figure 5. Schematic diagram of the penicillin fermentation process.

Case studies

In this section, the effectiveness of the multi-stage ORNPE algorithm is verified through two examples of batch processes. The two examples are the penicillin fermentation simulation process and the actual semiconductor industry process—the AI reactor corrosion process. They are typical batch processes. A modified multi-stage GNPE method (Yao et al., 2021), Multiway Slow Feature Analysis (MSFA; Shumei Zhang and Zhao, 2018), MNPE (Sun et al., 2018), and MPCA (Zhaomin et al., 2014) are used for comparison.

Penicillin fermentation simulation process

Penicillin is an antibiotic that has been widely used in clinical medicine, and its fermentation process is a typical batch process—the schematic diagram of the penicillin fermentation process is shown in Figure 5. To facilitate data monitoring analysis, the Pensim2.0 simulation platform designed by Birol et al. (2002), Chicago Illinois Institute of Technology, USA, is selected to simulate batch process. The simulation platform generates batch data of different conditions by setting the initial conditions. The Pensim2.0 simulation platform can set three types of faults: Aeration rate fault, agitator power fault, and substrate feeding rate fault. The introduced fault signal types include step signal and ramp signal, which can effectively simulate various variables in many different types.

We select 10 process variables of 18 variables in the platform as experimental test variables, as shown in Table 1.

We collect 30 batches of data under normal operating conditions on this platform as training samples. The reaction time of each batch is set to 400h, and the sampling interval is set to 1h. To get the closer actual fermentation process, we add Gaussian white noise to the process data. A training data set $X(30 \times 400 \times 10)$ is obtained. The 30 batches of obtained training data are unfolded into a two-dimensional matrix and divided into stages. The results of the stage division are shown in Figure 6. The penicillin fermentation process is divided into three stages: the first stage is 1~43h, the second stage is 44~169h, and the third stage is 170~400h.

The corresponding fault batches are formed by adding step and ramp disturbances into the three variables of aeration

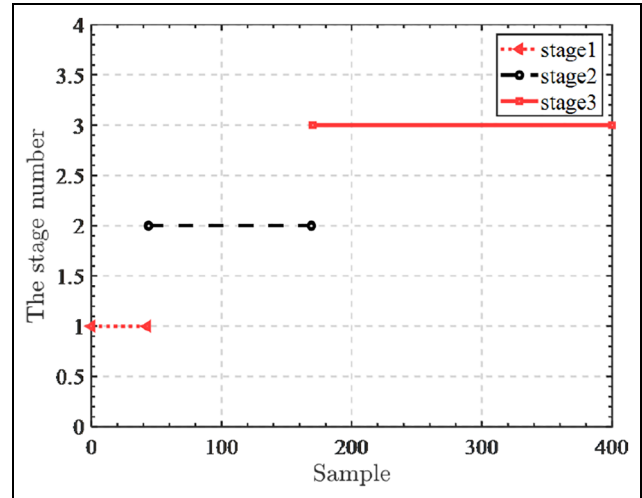
Table 1. Detection variables.

No.	Process variables
1	Aeration rate (L/h)
2	Agitator power (r/min)
3	Substrate feed flow rate (L/h)
4	Substrate feed flow temperature (K)
5	Substrate concentration (L/h)
6	Dissolved oxygen concentration (%)
7	Reactor volume (L)
8	Fermentation temperature (K)
9	PH
10	CO ₂ (%)

rate, agitator power, and substrate feeding rate, as shown in Table 2. To make the process data closer to the actual fermentation process, all data variables are added Gaussian white noise.

The detection rates of MPCA, MSFA, MNPE, multi-stage GNPE, and multi-stage ORNPE algorithms for the faults F1 and F6 in Table 2 are shown in Table 3. We can see that fault F1 and fault F2 are the easiest to be detected, and all algorithms can fully detect them. The proposed algorithm in this paper has the highest detection rates for faults F3–F6. Ramp fault disturbances are more challenging to be detected due to the slow changes of the variables. The fault caused by the substrate feeding rate variable is more difficult to be detected, this is because the propagation speed of the reaction substrate is slow.

Figure 7 shows the fault monitoring results of fault F4 by using MPCA, MNPE, MSFA, multi-stage GNPE, and the multi-stage ORNPE algorithm. Fault F4 is agitation rate, and it added a ramp disturbance with a fault amplitude of 4%

**Figure 6.** Results of stage division of penicillin fermentation process.

between 300~400h, Figure 7(a) is the statistics T^2 and SPE monitoring diagram of MPCA under fault F4, and it can be seen that the fault is detected at the 322h and 309h, respectively, and the fault false alarm is serious under normal operating conditions. Figure 7(b) is the statistics T^2 and SPE monitoring diagram of MNPE algorithm under fault F4, and it can be seen that the fault is detected at the 345h and 323h, respectively, and compared with MPCA algorithm, the fault false alarm is more miniature. But when the fault is detected, the time is delayed, which means that the fault cannot be detected in time. Figure 7(c) is the statistics S^2 and SPE monitoring diagram of MSFA algorithm under fault F4, and it can be seen that the fault is detected at the 336h and 322h respectively, which the fault false alarms are fewer than MPCA

Table 2. Fault batch samples.

Fault no.	Fault variable name	Disturbance type	Amplitude (%)	Fault introduction period
F1	Aeration rate	Step	4	100–300 h
F2	Aeration rate	Ramp	2	300–400 h
F3	Agitator rate	Step	2	250–350 h
F4	Agitator rate	Ramp	4	300–00 h
F5	Substrate feeding rate	Step	3	200–300 h
F6	Substrate feeding rate	Ramp	0.8	250–400 h

Table 3. Fault detection rate of each fault batch by different methods.

Fault no.	MPCA		MNPE		MSFA		multi-stage GNPE		multi-stage ORNPE	
	T^2	SPE	T^2	SPE	S^2	SPE	T^2	SPE	T^2	R^2
F1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F3	0.556	0.621	0.596	0.684	0.699	0.732	0.745	0.809	0.911	0.932
F4	0.780	0.910	0.550	0.770	0.640	0.780	0.920	0.940	0.980	0.990
F5	0.357	0.383	0.543	0.497	0.552	0.649	0.675	0.773	0.717	0.803
F6	0.903	0.908	0.931	0.944	0.933	0.940	0.951	0.955	0.971	0.972

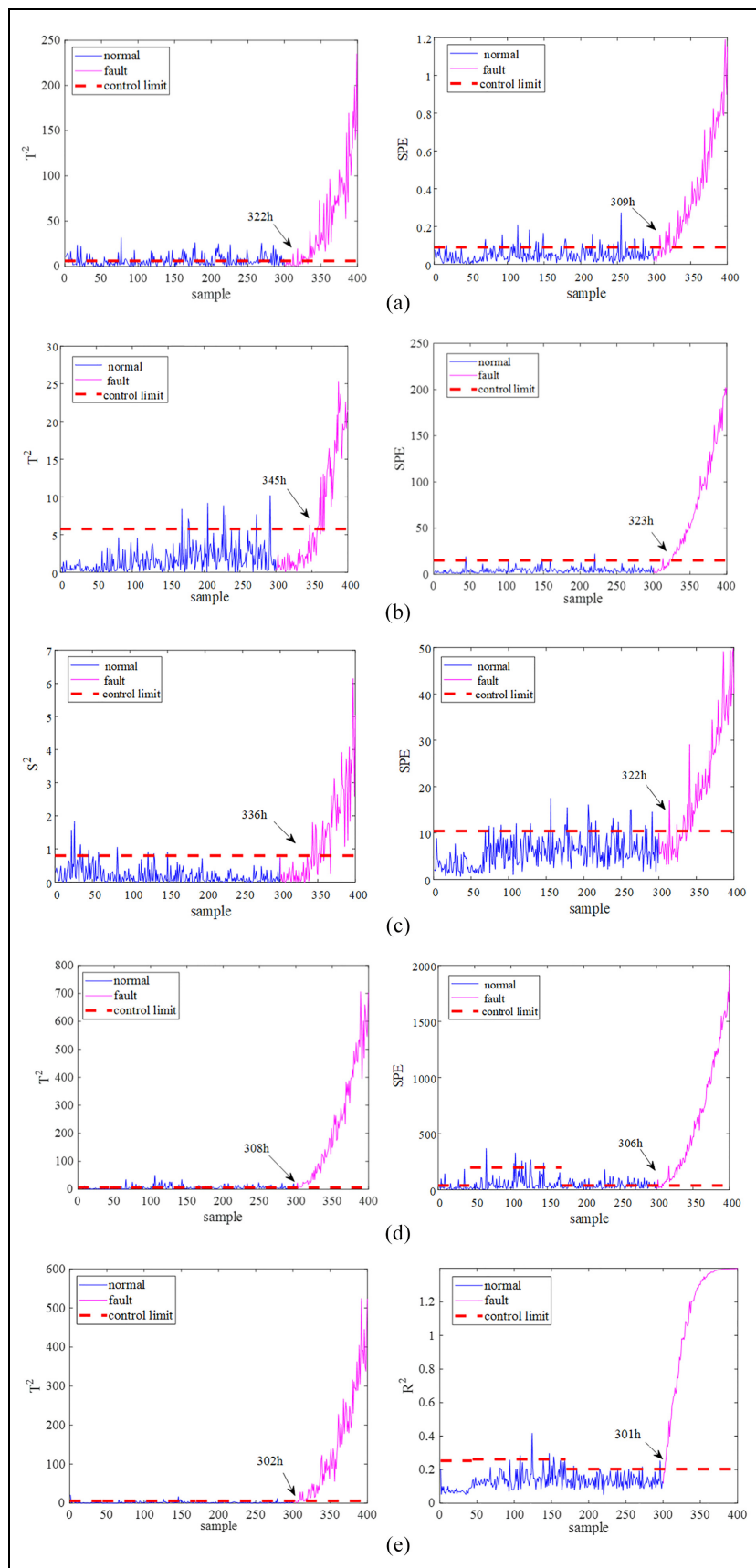
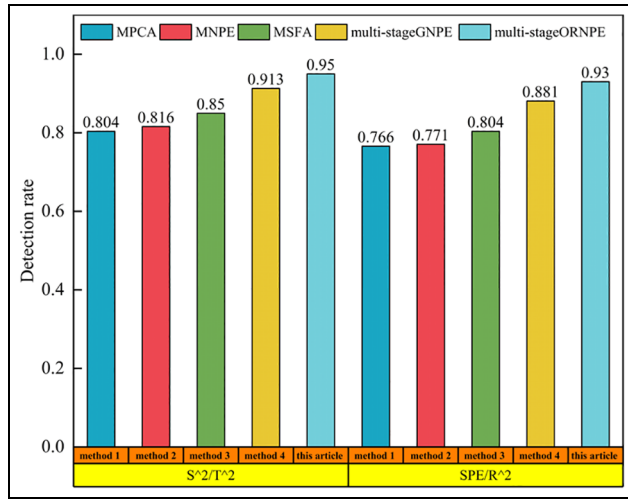


Figure 7. Monitoring results for the penicillin fermentation process of the fault F4: (a) MPCA, (b) MNPE, (c) MSFA, (d) multi-stage GNPE, and (e) multi-stage ORNPE.

Table 4. Test variables.

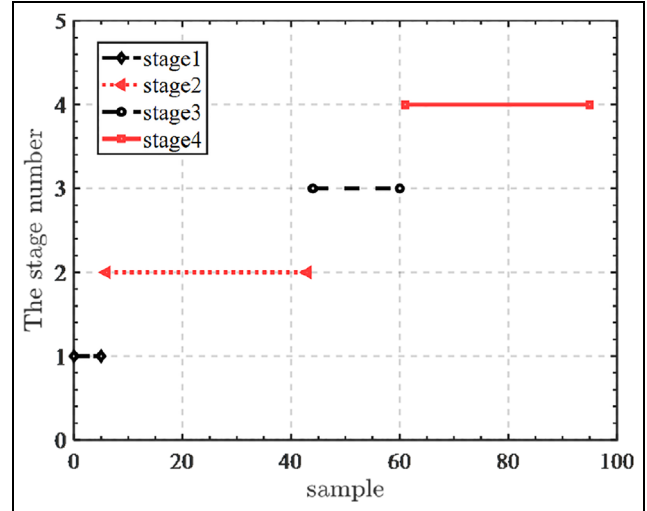
No.	Measured variables	No.	Measured variables
1	BCl ₃ Flow	10	RF power
2	Cl ₂ Flow	11	RF impedance
3	RF bottom power	12	TCP tuner
4	Endpoint A Detector	13	TCP phase error
5	Helium pressure	14	TCP impedance
6	Chamber pressure	15	TCP top power
7	RF tuner	16	TCP load
8	RF load	17	Vat valve position
9	Phase error	—	—

**Figure 8.** Graph of average fault detection rate of fault batches during penicillin fermentation.

algorithm, and the fault is detected earlier than MNPE algorithm. Figure 7(d) is the statistics T^2 and SPE monitoring diagram of multi-stage GNPE algorithm under fault F4, and it can be seen that the fault is detected at the 308h and 306h, respectively, and due to the multi-stage characteristic of the penicillin fermentation process, multi-stage GNPE algorithm performs modeling and monitoring in stages to reduce fault false alarms under normal operating conditions. Multi-stage algorithm can detect the occurrence of faults in a timely manner than single-stage algorithms. Figure 7(e) is the statistics

Table 5. Fault detection rates of each fault type by different algorithms.

Fault type	MPCA		MNPE		MSFA		Multi-stage GNPE		Multi-stage ORNPE	
	T^2	SPE	T^2	SPE	S^2	SPE	T^2	SPE	T^2	R^2
TCP + 50	0.323	0.351	0.379	0.382	0.401	0.442	0.698	0.833	0.794	0.820
RF-12	0.310	0.368	0.336	0.402	0.399	0.437	0.599	0.779	0.801	0.812
BCl ₃ + 5	0.322	0.372	0.411	0.454	0.405	0.532	0.671	0.831	0.779	0.802
Pr-2	0.347	0.389	0.421	0.436	0.368	0.452	0.757	0.884	0.852	0.915
Cl ₂ -5	0.331	0.441	0.401	0.427	0.399	0.411	0.597	0.644	0.799	0.852
He Chuck	0.348	0.339	0.397	0.422	0.402	0.541	0.661	0.689	0.833	0.901

**Figure 9.** Results of stage division of semiconductor industry process.

T^2 and R^2 monitoring diagram of the proposed algorithm in this paper under fault F4, and it can be seen that the fault is detected at the 302h and 301h, respectively. There are few fault false alarms under normal operating conditions, and compared to other algorithms, the proposed algorithm can detect the fault earlier and have a better monitoring effect.

For process data, fault detection is delayed due to the dynamic characteristic of the data. The proposed algorithm uses SFA to extract dynamic characteristic and effectively deal with the influence of dynamic characteristic on detection delay. In addition, the proposed algorithm also considers the stage characteristic of the data. For the stage of fault occurrence, the calculation of statistics is more in line with the actual situation. Therefore, the real-time performance of fault detection is better.

Figure 8 is a bar graph that compares the average fault detection rates of selected fault batches. From Figure 8, we can see the superiority of the proposed algorithm for the fault detection of the penicillin fermentation process.

Semiconductor industry process

The semiconductor industry process—AI reactor corrosion process is applied to compare the performances of different fault detection algorithms. The data are derived from the

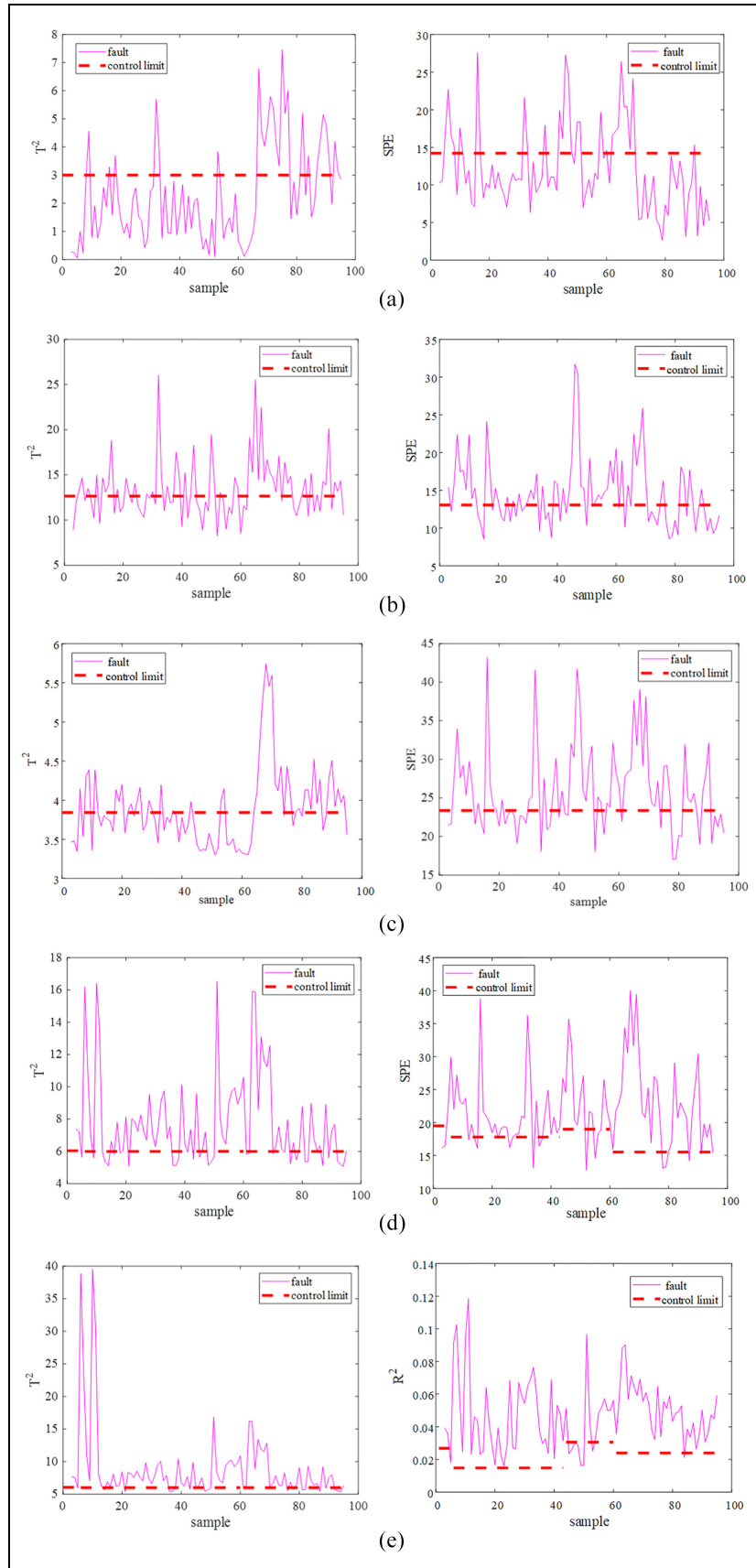


Figure 10. Monitoring results for semiconductor industry process of the Pr-2: (a) MPCA, (b) MNPE, (c) MSFA, (d) multi-stage GNPE, and (e) multi-stage ORNPE.

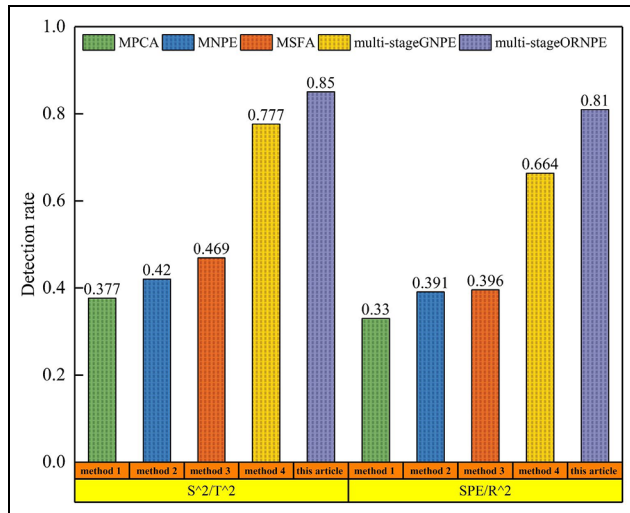


Figure 11. Graph of average fault detection rate of fault batches during semiconductor etching.

actual data of the semiconductor production process of Texas Instruments (Azamfar et al., 2020). It is a typical complex multi-stage batch process. The data set is composed of 107 normal batches and 20 fault batches, and the fault batch contains Tape Carrier Package (TCP) power fault, Radio Frequency (RF) power fault, pressure fault, Cl_2 fault or BCl_3 flow fault. We select 17 variables from 40 measured variables as test variables, as shown in Table 4.

In this experiment, we select 50 normal batches for modeling, and each batch is 95 hours in length. We select 17 variables to monitor process status. The three-dimensional normal sample modeling data matrix is $X(50 \times 17 \times 95)$. The new batch of test data is $X_{test}(95 \times 17)$. The semiconductor industry process is divided into four stages: 0~5h, 6~43h, 44~60h, and 61~95h. The results of the stage division are shown in Figure 9.

This experiment selects TCP + 50, RF-12, BCl_3 + 5, Pr-2, Cl_2 -5, and He chuck pressure fault batches to introduce faults in all periods of each batch. Table 5 shows the fault detection rates of six types of faults under different algorithms. It can be seen from Table 5 that although the fault detection rate from MPCA, MNPE to MSFA has increased in turn. Because they are single-stage detection algorithms, the overall detection effects of various faults are poor. Multi-stage GNPE algorithm has a higher detection rate than the single-stage algorithms, multi-stage ORNPE algorithm proposed in this paper has a higher detection rate than multi-stage GNPE algorithm.

Figure 10 shows the fault monitoring results of the fault batch Pr-2 by using MPCA, MNPE, MSFA, multi-stage GNPE, and multi-stage ORNPE algorithm. It can be seen from Figure 10(a)–(c), three algorithms have seriously missed fault alarms during the fault detection process, and more faults are not detected. It can be seen from Figure 10(a)–(c) that the fault detection effect is gradually getting better. The detection effect of statistic SPE is better than the statistics T^2/S^2 . Still, MPCA, MNPE, and MSFA belong to single-stage

detection algorithms, so their detection effects for batch process are poor. The detection algorithms used in Figure 10(d) and (e) are multi-stage detection algorithms. Figure 10(d) is the statistical monitoring chart of multi-stage GNPE algorithm. Figure 10(e) is the statistical monitoring chart of multi-stage ORNPE algorithm proposed in this paper, which has better detection results than the multi-stage GNPE algorithm of Figure 10(d). The main reason is that the proposed algorithm preserves the dynamic global details of the data while considering the local information of the process data.

Figure 11 is a bar graph that compares the average fault detection rates of selected semiconductor industrial process for detection algorithms under multiple fault batches. The fault detection effect of the multi-stage ORNPE algorithm is better than other four algorithms. The proposed algorithm can better monitor the process.

Conclusion

In this paper, we propose a fault monitoring strategy based on multi-stage ORNPE for batch process. First, after the initial division of each batch of data into stages, the same stages of different batches are processed in equal length. Second, on the basis of NPE which can reveal the local structure information and lose global information of data, a global objective function is established by cross-entropy optimized SFA to extract both essential features and dynamic global information of the process data. Finally, the statistics are built to monitor batch process. The penicillin fermentation process and a real semiconductor process are adopted to verify the effectiveness and superiority of the proposed algorithm. For the penicillin fermentation process, the average fault detection rates of statistics T^2 and R^2 of the proposed algorithm are 0.95 and 0.93, the average detection rates of statistics T^2 and R^2 of the semiconductor process are 0.85 and 0.81, respectively. The detection rates are higher than other comparison algorithms. Compared with the comparison algorithms, the proposed algorithm in this paper increases the calculation amount, but the computational complexity still is $O(n^2)$ since the proposed algorithm only considers fault detection and does not consider the identification of fault variables. In the future, based on improving the effect of fault detection, the further research on fault diagnosis and fault prediction will be carried out.

Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding


The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by funding received from the National Nature Science Foundation of China (No. 61763029), the Science and Technology Project of Gansu Province (21YF5GA072, 21JR7RA206), Open Fund project

of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (2022KX07), and the National Key Research and Development Plan (2020YFB1713600).

ORCID iDs

Xiaoqiang Zhao  <https://orcid.org/0000-0001-5687-942X>

Kai Liu  <https://orcid.org/0000-0002-8624-2208>

Yonyong Hui  <https://orcid.org/0000-0001-5014-887X>

References

- Abdi H and Williams LJ (2010) Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2(4): 433–459.
- Azamfar M, Li X and Lee J (2020) Deep learning-based domain adaptation method for fault diagnosis in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 33(3): 445–453.
- Birol G, Undey C, Parulekar SJ, et al. (2002) A morphologically structured model for penicillin production. *Biotechnology and Bioengineering* 77(5): 538–552.
- Camarrone F and Van Hulle MM (2018) Fast multiway partial least squares regression. *IEEE Transactions on Biomedical Engineering* 66(2): 433–443.
- Cotrufo N and Zmeureanu R (2016) PCA-based method of soft fault detection and identification for the ongoing commissioning of chillers. *Energy and Buildings* 130: 443–452.
- Frey BJ and Dueck D (2007) Clustering by passing messages between data points. *Science (New York, N.Y.)* 315(5814): 972–976.
- Fu Y and Zhang Y (2017) Fault detection and diagnosis of batch process using kernel local FDA. In: *2017 Chinese Automation Congress (CAC)*, Jinan, China, 20–22 October, pp. 3997–4001. New York: IEEE.
- Gao X, Cui N, Qi Y, et al. (2014) Fault detection of multi-phase batch process based on adaptive FCM. In: *Proceedings of the 33rd Chinese control conference*, Nanjing, China, 28–30 July, pp. 3088–3093. New York: IEEE.
- Ge Z, Zhao L, Yao Y, et al. (2012) Utilizing transition information in online quality prediction of multiphase batch processes. *Journal of Process Control* 22(3): 599–611.
- Guo R, Guo K and Dong J (2017) Phase partition and online monitoring for batch process based on multiway BEAM. *IEEE Transactions on Automation Science and Engineering* 14(4): 1582–1589.
- He X, Cai D, Yan S, et al. (2005) Neighborhood preserving embedding. In: *Tenth IEEE international conference on computer vision (ICCV'05)*, Beijing, China, 17–21 October, vol. 1, pp. 1208–1213. New York: IEEE.
- Helland I (2014) Partial least squares regression. *Wiley StatsRef: Statistics Reference Online*. Epub ahead of print 29 September. DOI: 10.1002/9781118445112.stat03287.
- Jeffy F, Gugaliya JK and Kariwala V (2018) Application of multiway principal component analysis on batch data. In: *2018 UKACC 12th international conference on control (CONTROL)*, Sheffield, 5–7 September, pp. 414–419. New York: IEEE.
- Jiang Y and Yin S (2018) Recent advances in key-performance-indicator oriented prognosis and diagnosis with a MATLAB toolbox: DB-KIT. *IEEE Transactions on Industrial Informatics* 15(5): 2849–2858.
- Jiang Y, Yin S and Kaynak O (2020) Performance supervised plant-wide process monitoring in industry 4.0: A roadmap. *IEEE Open Journal of the Industrial Electronics Society* 2: 21–35.
- Lavanya S, Prasanth A, Jayachitra S, et al. (2021) A tuned classification approach for efficient heterogeneous fault diagnosis in IoT-enabled WSN applications. *Measurement* 183: 109771.
- Li G, Qin SJ and Zhou D (2010) Geometric properties of partial least squares for process monitoring. *Automatica* 46(1): 204–210.
- Liu J, Liu T and Zhang J (2016) Phase partition for nonlinear batch process monitoring. *IFAC-PapersOnLine* 49(7): 1181–1186.
- Luo L and Bao S (2018) Knowledge-data-integrated sparse modeling for batch process monitoring. *Chemical Engineering Science* 189: 221–232.
- Majid NAA, Taylor MP, Chen JJJ, et al. (2011) Aluminium process fault detection by multiway principal component analysis. *Control Engineering Practice* 19(4): 367–379.
- Ningyun LU, Gao F and Wang F (2010) Sub-PCA modeling and on-line monitoring strategy for batch processes. *Aiche Journal* 50(1): 255–259.
- Peng C, RuiWei L, Kang O, et al. (2020) Batch process fault detection for multi-stage broad learning system. *Neural Networks* 129: 298–312.
- Prasanth A (2021) Certain investigations on energy-efficient fault detection and recovery management in underwater wireless sensor networks. *Journal of Circuits, Systems and Computers* 30(8): 2150137.
- Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis (Review). *Annual Reviews in Control* 36(2): 220–234.
- Roweis ST and Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science (New York, N.Y.)* 290(5500): 2323–2326.
- Shang C, Huang B, Yang F, et al. (2016) Slow feature analysis for monitoring and diagnosis of control performance. *Journal of Process Control* 39: 21–34.
- Sun Y, Xie G, Cao Y, et al. (2018) A fault diagnosis method for train plug doors based on MNPE and IPSO-MSVM. In: *2018 international conference on control, automation and information sciences (ICCAIS)*, Hangzhou, China, 24–27 October, pp. 467–471. New York: IEEE.
- Wang X, Wang P, Gao X, et al. (2016) On-line quality prediction of batch processes using a new kernel multiway partial least squares method. *Chemometrics and Intelligent Laboratory Systems* 158: 138–145.
- Xu X and Ding J (2021) Decentralized dynamic process monitoring based on manifold regularized slow feature analysis. *Journal of Process Control* 98: 79–91.
- Yao H, Zhao X, Li W, et al. (2021) Batch process monitoring based on global enhanced multiple neighborhoods preserving embedding. *Transactions of the Institute of Measurement and Control* 44: 620–633.
- Yu JB (2016) Process monitoring through manifold regularization-based GMM with global/local information. *Journal of Process Control* 45: 84–99.
- Zhang H, Deng X, Zhang Y, et al. (2021a) Dynamic nonlinear batch process fault detection and identification based on two-directional dynamic kernel slow feature analysis. *Canadian Journal of Chemical Engineering* 99(1): 306–333.
- Zhang H, Tian X, Deng X, et al. (2018) Batch process fault detection and identification based on discriminant global preserving kernel slow feature analysis. *ISA Transactions* 79: 108–126.
- Zhang M, Ge Z, Song Z, et al. (2011) Global-local structure analysis model and its application for fault detection and identification. *Industrial & Engineering Chemistry Research* 50(11): 6837–6848.
- Zhang M, Wang R, Cai Z, et al. (2020) Phase partition and identification based on kernel entropy component analysis and multi-class support vector machines-fireworks algorithm for multi-phase batch process fault diagnosis. *Transactions of the Institute of Measurement and Control* 42(12): 2324–2337.
- Zhang M, Yi Y and Cheng W (2021b) Multistage condition monitoring of batch process based on multi-boundary hypersphere SVDD with modified bat algorithm. *Arabian Journal for Science and Engineering* 46(2): 1647–1661.

- Zhang S and Zhao C (2018) Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly. *IEEE Transactions on Industrial Electronics* 66(5): 3773–3783.
- Zhang S, Zhao C and Gao F (2019) Incipient fault detection for multiphase batch processes with limited batches. *IEEE Transactions on Control Systems Technology* 27(1): 103–117.
- Zhao CH and Sun YX (2013) Step-wise sequential phase partition (SSPP) algorithm based statistical modeling and online process monitoring. *Chemometrics and Intelligent Laboratory Systems* 125: 109–120.
- Zhao X, Wang T and Hui Y (2016) MGNPE-LICA algorithm for fault diagnosis of batch process. *Canadian Journal of Chemical Engineering* 94(10): 1947–1954.
- Zhaomin L, Qingchao J and Xuefeng Y (2014) Batch process monitoring based on multisubspace multiway principal component analysis and time-series Bayesian inference. *Industrial & Engineering Chemistry Research* 53(15): 6457–6466.



WOPCA-EGNPE algorithm based on variable division for fault monitoring of batch process

Xiaoqiang Zhao^{1,2,3} · Kai Liu^{1,2} · Yonyong Hui^{1,2,3} · Hongmei Jiang^{1,2,3}

Received: 13 September 2021 / Revised: 10 August 2022 / Accepted: 7 September 2022 / Published online: 15 September 2022
© The Author(s) under exclusive licence to Associação Brasileira de Engenharia Química 2022

Abstract

The traditional fault monitoring methods assume that batch process is stationary, but the actual batch process variables are stationary and nonstationary mixed distribution. The fault information is easily submerged in the normal nonstationary signals, causing difficulty in fault monitoring. In this paper, we propose a joint stationary-nonstationary Bayesian statistical indicator monitoring algorithm based on weighted orthogonal principal component analysis-exponential global neighborhood preserving embedding (WOPCA-EGNPE) with variable division. Firstly, the augmented dickey-fuller (ADF) test divides the stationary space and the nonstationary space through the stationarity of variables; secondly, nonstationary variables obtain a stationary residual series by the cointegration analysis method, and for the stationary residual series, exponential global neighborhood preserving embedding (EGNPE) algorithm constructs the model, which considers both global and local information of the data, and highlights the important global and local data information through an exponential transformation; then, weighted orthogonal principal component analysis (WOPCA) algorithm construct the model in the stationary space; finally, a joint monitoring index is established by using Bayesian inference method to realize the process monitoring. The penicillin fermentation process is applied to verify the effectiveness of the proposed algorithm.

Keywords Batch process · Fault monitoring · Nonstationarity · Variable division · Joint statistics

Introduction

For the urgent market requirements of multi-species, multi-standard, and high-quality products, batch production with its small production volume and high added value is widely used in plastic products, integrated circuit manufacturing, pharmaceutical, food processing, chemical production, and other fields (Wang et al. 2020; Zhang et al. 2020; Zhao and Mou 2021). At the same time, with the rapid development of batch processes, there is a more urgent need for the efficient operation of batch processes. To ensure the safe operation of batch process, process monitoring and timely fault detection

are of great importance (Li et al. 2021; Nawaz et al. 2021; Peng and Ruiwei 2021; Hanyuan Zhang et al. 2021a).

In recent years, multivariate statistical process monitoring (MSPM) methods are widely used in process monitoring (Yunus et al. 2020; Yu et al. 2019; Muñoz et al. 2018; He and Wang 2018; Wang et al. 2018). In the MSPM methods, for process monitoring, PCA and PLS have significant results (Chen and Liu 2001; Gunther et al. 2009), MPCA and MPLS with multi-directional data expansion strategies are applied in industrial processes (Monroy et al. 2011; Stubbs et al. 2013). These methods explore the basic characteristics of the process by projecting the observed data into a low-dimensional space, build monitoring statistics based on low-dimensional space and use them to implement anomaly detection. The traditional MSPM methods are easy to conceal useful information when normal data are used for training. The fluctuation of the process variable data is submerged, and the ability of detection fault is poor. Huang et al. (2017) proposed a weighted principal component analysis method for sensor fault detection and isolation based on the fault sensitivity of each main direction in which the same fault occurred on a certain sensor. However,

✉ Xiaoqiang Zhao
xqzhao@lut.edu.cn

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

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

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

the non-orthogonal weight vectors easily led to local spatial distortion structure. Neighborhood preserving embedding algorithm is a linear dimensionality reduction algorithm in manifold learning (He et al. 2005). Its advantage is that it can maintain the local neighborhood structure of the data on the manifold, but it ignores the global information of the data in the feature extraction process. Zhao et al. (2016) proposed a Global Neighborhood Preserving Embedding (GNPE) algorithm, which fully considered global and local information, and applied it to fault monitoring in batch process. When it obtained global and local information, some important features of the data were concealed, and the data could not be used effectively.

A lot of research works on fault monitoring assume that the production process is stationary (Aggoun and Chetouani, 2021; Gao et al. 2020; Jiang et al. 2019; Li et al. 2017; Zhang et al. 2017, 2021b). Due to equipment wear, unknown disturbances, human factors, and changes in working conditions, industrial processes often have an obvious nonstationary characteristic, which makes traditional multivariate statistical methods unfavorable for fault monitoring. For the nonstationary characteristic of the process, Chen et al. (2009) introduced cointegration analysis into the field of process monitoring as an effective method to study the long-term balance relationship between nonstationary variables. Li et al. (2014) extracted multiple stationary residual series to establish a monitoring model through a cointegration analysis and establish the monitoring indicators based on the nonstationary multivariate latent structure to effectively detect the occurrence of abnormality. Sun et al. (2017) distinguished and separated nonstationary variables from stationary variables so that a cointegration analysis model was constructed to describe the long-term equilibrium relationship between nonstationary variables. In the actual process, some of the variables have nonstationary characteristic, and some of the variables have stationary characteristic, therefore, the process has the characteristic of the stationary and nonstationary mixing distribution, for such characteristic, it is difficult to extract high-dimensional characteristic of process monitoring variables. And the overall modeling is not conducive to effective process monitoring. Zhao and Huang (2018) proposed a new full-state monitoring strategy based on cointegration analysis (CA) and slow feature analysis (SFA) to build multiple monitoring statistics in different spaces to monitor the overall status of the process. However, Simultaneous detection of multiple statistics increases the burden of process monitoring to a certain extent.

Aiming at the above problems, we propose a WOPCA-EGNPE fault monitoring algorithm of batch process based on variable division. The algorithm divides the original process data into two subspaces by division variables method: stationary subspace and nonstationary subspace. In the nonstationary subspace, a stationary residual series containing

the nonstationary characteristic of the variables is obtained through the theory of cointegration, and the obtained stationary residual series considers both global and local information, we adopt the exponential transformation method to highlight the global and local important information, so the EGNPE model is established. We use the EGNPE model to extract the features of nonstationary subspace data, and build monitoring statistics; in the stationary subspace, we use the inverse variance to weight PCA, the WOPCA model is obtained by orthogonalizing the weight matrix. The optimal dimensionality reduction mapping matrix is obtained by the WOPCA model, further the optimal low-dimensional features are obtained and the monitoring statistics are constructed. We establish a joint statistic by Bayesian inference method to achieve joint monitoring. The effectiveness and superiority of the algorithm are verified by the penicillin simulation experiment. The novelties of this paper are given as follows:

- (1) The algorithm divides the original process data into two subspaces by division variables method: stationary subspace and nonstationary subspace.
- (2) WOPCA and EGNPE models are established in stationary and non-stationary spaces respectively, and monitoring statistics are established.
- (3) A joint statistic is established by Bayesian inference method to realize joint monitoring and monitor batch process.

Preliminaries

Cointegration analysis (CA)

As an effective method to describe the relationship between nonstationary variables, cointegration analysis (Granger 2004) can combine the advantages of short-run and long-run models in time series analysis, and provide a better solution for nonstationary time series in the modeling process.

For a nonstationary time series $\mathbf{Z}(m \times n) = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t, \dots, \mathbf{z}_n]$, where $\mathbf{z}_t = (z_1, z_2, \dots, z_m)$, n is the number of nonstationary time series, and m is the number of sampling points, the nonstationary time series has a long-run equilibrium relationship, then there exists a vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)^T$ such that the linear combination of the nonstationary time series has the following relationship:

$$\begin{aligned}\xi_t &= \beta_1 z_1 + \beta_2 z_2 + \dots + \beta_n z_n \\ &= \boldsymbol{\beta}^T \mathbf{z}_t \\ (t &= 1, \dots, m)\end{aligned}\quad (1)$$

where ξ_t denotes the smooth residual series, $\boldsymbol{\beta}$ is the cointegration matrix, and \mathbf{z}_t is the cointegration variable.

\mathbf{z}_t of the vector autoregressive (VAR) model is built as:

$$\mathbf{z}_t = \mathbf{a}_1 \mathbf{z}_{t-1} + \cdots + \mathbf{a}_p \mathbf{z}_{t-p} + \mathbf{c} + \boldsymbol{\mu}_t \quad (2)$$

where $\mathbf{a}_i (n \times n)$ is the coefficient matrix, $\boldsymbol{\mu}_t (n \times 1)$ is the white noise vector with Gaussian distribution, $\mathbf{c} (n \times 1)$ is the constant vector, and p is the order of the VAR model.

The error correction model of formula (3) is obtained by subtracting \mathbf{z}_{t-1} from the left and right ends of formula (2).

$$\Delta \mathbf{z}_t = \sum_{i=1}^{p-1} \boldsymbol{\Omega}_i \Delta \mathbf{z}_{t-i} + \boldsymbol{\Gamma} \mathbf{z}_{t-1} + \boldsymbol{\mu}_t \quad (3)$$

where $\boldsymbol{\Gamma} = -\mathbf{I}_N + \sum_{i=1}^p \mathbf{a}_i$, $\boldsymbol{\Omega}_i = -\sum_{j=i+1}^p \mathbf{a}_j$, $i = 1, 2, \dots, p-1$.

$\boldsymbol{\Gamma}$ can be decomposed into two matrices which are full-rank columns $\boldsymbol{\Gamma} = \mathbf{A}\mathbf{B}^T$, where $\mathbf{A} (n \times r)$, $\mathbf{B} (n \times r)$. Then formula (3) can be transformed as:

$$\Delta \mathbf{z}_t = \sum_{i=1}^{p-1} \boldsymbol{\Omega}_i \Delta \mathbf{z}_{t-i} + \mathbf{A}\mathbf{B}^T \mathbf{z}_{t-1} + \boldsymbol{\mu}_t \quad (4)$$

The residual series $\boldsymbol{\gamma}_{t-1}$ can be obtained from formula (5):

$$\boldsymbol{\gamma}_{t-1} = \mathbf{B}^T \mathbf{z}_{t-1} \quad (5)$$

Formula (4) is taken into formula (5), the calculation formula of the residual sequence $\boldsymbol{\gamma}_{t-1}$ can be further obtained as shown formula (6):

$$\boldsymbol{\gamma}_{t-1} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \left(\Delta \mathbf{z}_t - \sum_{i=1}^{p-1} \boldsymbol{\Omega}_i \Delta \mathbf{z}_{t-i} - \boldsymbol{\mu}_t \right) \quad (6)$$

Since \mathbf{z}_t is a first-order single integer, $\Delta \mathbf{z}_t$ and $\Delta \mathbf{z}_{t-i}$ are stationary, and then the elements on the right-hand side of formula (5) are all stationary. $\mathbf{B}^T \mathbf{z}_{t-1}$ denotes a nonstationary linear combination, then \mathbf{B} is the covariance matrix.

Neighborhood preserving embedding (NPE)

Neighborhood preserving embedding (NPE) (Xiaofei He et al. 2005) is a local manifold learning algorithm. This algorithm obtains a low-dimensional space dataset $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ by linearly reconstructing the high-dimensional original data $\mathbf{X}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}^D$ to reduce the dimensionality, the relationship between \mathbf{X} and \mathbf{Y} satisfies: $\mathbf{Y} = \mathbf{A}^T \mathbf{X}$, where $\mathbf{A}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$, $(d \leq D)$ is the reduced projection matrix. NPE uses the k -nearest neighbor algorithm to find the nearest neighbors in the original data \mathbf{X} . The samples are connected to the corresponding nearest neighbor points to form the neighborhood map. The reconstruction is carried out through the linear combination of the sample points and their nearest neighbors. The weight matrix \mathbf{W} is calculated by minimizing the reconstruction error through formula (7).

$$\Phi(\mathbf{W}) = \min \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{x}_j \right\|^2 \quad (7)$$

$$\sum_{j=1}^n \mathbf{w}_{ij} = 1$$

The projection matrix \mathbf{A} can be obtained by minimizing the cost function of formula (8) as follows:

$$f(\mathbf{A}) = \min \sum_{i=1}^n \left(\mathbf{y}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{y}_j \right)^2 \quad (8)$$

$$= \min \mathbf{y}^T (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{y}$$

$$= \min \mathbf{a}^T \mathbf{X} (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{X}^T \mathbf{a}$$

where the constraint is: $\mathbf{y}^T \mathbf{y} = \mathbf{a}^T \mathbf{X} \mathbf{X}^T \mathbf{a} = 1$, the solution of formula (8) can be transformed into the problem of finding the generalized eigenvalue of formula (9).

$$\mathbf{X} \mathbf{M} \mathbf{X}^T \mathbf{a} = \lambda \mathbf{X} \mathbf{X}^T \mathbf{a} \quad (9)$$

where, $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$, the eigenvectors corresponding to the first d minimum eigenvalues of the solution form the mapping matrix \mathbf{A} .

WOPCA-EGNPE algorithm of fault monitoring with stationary-nonstationary variables

In this paper, a WOPCA-EGNPE fault monitoring algorithm is proposed. We divide the stationary and nonstationary subspaces into two subspaces by judging the stationarity of the variables by ADF test and establish the EGNPE and WOPCA models based on improved NPE and PCA, which are applied to the two subspaces respectively. The Bayesian method is used to establish joint statistics to achieve fault monitoring.

Division between stationary-nonstationary variables

We utilize augmented dickey-fuller test (ADF) (Ajewole et al. 2020) to determine whether the variables are stationary or not, and divide the variables into different subspaces. The regression model for the time series $\mathbf{y}_t = (t = 1, 2, \dots, T)$ is developed as formula (10):

$$\mathbf{y}_t = \beta \mathbf{y}_{t-1} + \sum_{k=1}^l b_k \Delta \mathbf{y}_{t-k} + \mathbf{c} + \mathbf{e}_t \quad (10)$$

where \mathbf{c} is the constant, \mathbf{e}_t is the random error, It and the process variables satisfy independent identical distribution.

Δ is the difference operation, then, $\Delta \mathbf{y} = \mathbf{y}_t - \mathbf{y}_{t-1}$, and l is the time lag term.

The estimation of the regression coefficient β by least squares is as follows:

$$[\hat{\mathbf{c}}, \hat{\beta}, \hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \dots, \hat{\mathbf{b}}_l] = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{y} \quad (11)$$

where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]^T$, T is the number of sampling points and the matrix Ψ is shown as formula (12):

$$\Psi = \begin{bmatrix} 1 & \mathbf{y}_0 & \Delta \mathbf{y}_0 & \cdots & \Delta \mathbf{y}_{1-t} \\ 1 & \mathbf{y}_1 & \Delta \mathbf{y}_1 & \cdots & \Delta \mathbf{y}_{2-t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{y}_{T-1} & \Delta \mathbf{y}_{T-1} & \cdots & \Delta \mathbf{y}_{T-t} \end{bmatrix} \quad (12)$$

The null hypothesis test $\mathbf{H}_0 : \beta = 1$ is used to determine whether the series is nonstationary. If the test statistic is greater than the critical threshold, the original hypothesis is accepted, and the time series \mathbf{y} has nonstationary characteristic at this time; conversely, the original hypothesis is rejected, and the series is stationary within a certain confidence level.

The process monitoring variables are divided into stationary and nonstationary subspaces by the judgment of stationarity of variables, that is, stationary space Ω_S and nonstationary space Ω_{NS} .

Fault monitoring with stationary-nonstationary variables based on WOPCA-EGNPE algorithm

The variables are divided into two subspaces, Ω_S and Ω_{NS} by ADF test. We adopt weighted orthogonal principal component analysis to calculate monitoring statistics T_S^2 and SPE_S in subspace Ω_S , and adopt exponential global neighborhood preserving embedding to calculate monitoring statistics T_{NS}^2 and SPE_{NS} in subspace Ω_{NS} . To consider the influence of stationary space and nonstationary space on process monitoring at the same time, we use Bayesian inference method to create a joint monitoring metric.

Weighted orthogonal Principal component analysis (WOPCA)

Principal component analysis uses linear combinations of variables in the high-dimensional data to form principal components for most of the high-dimensional data, and such principal components provide information about the complexities of the simplified raw data. When PCA is used for dimensionality reduction in space Ω_S , the inverse variance weighting is applied to highlight key information, and the WOPCA model is built. For the original data $\mathbf{X} = \sum_{i=1}^n \mathbf{x}_i$, we

use the WOPCA model to reduce the dimensionality process as follows.

The original data \mathbf{X} can be decomposed as:

$$\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \cdots + \mathbf{t}_n \mathbf{p}_n^T = \mathbf{T} \mathbf{P}^T + \mathbf{E} \quad (13)$$

where, $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n]$ is the feature matrix, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ is the load matrix, and \mathbf{E} is the residual vector.

The features and load matrices are obtained by constructing the covariance matrix \mathbf{S} , shown as formula (14):

$$\mathbf{S} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X} \quad (14)$$

where, \mathbf{T} is the diagonal matrix composed of the eigenvalues of the covariance matrix \mathbf{S} , and the eigenvectors corresponding to the eigenvalues form the load matrix \mathbf{P} .

The covariance matrix \mathbf{S} uses the inverse variance weighting manner to highlight key information, and a new weighted covariance matrix \mathbf{H} is obtained, shown as formula (15).

$$\mathbf{H} = \sum_{i=1}^n \frac{1}{\sigma_i^2} \mathbf{S} \quad (15)$$

where, σ_i^2 is a variance of the column direction of \mathbf{S} .

Since the new weighted covariance matrix \mathbf{H} is not orthogonal, we orthogonalize the covariance matrix as shown in formula (16), $\mathbf{W}\mathbf{O}$ is the weighted covariance matrix after orthogonalization.

$$\mathbf{W}\mathbf{O} = \left[\text{diag} \left(\sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^2} \right) \right]^{-1} \quad (16)$$

where, $\bar{\mathbf{x}}$ is the sample mean.

By taking the eigenvalues and eigenvectors of the weighted orthogonal matrix $\mathbf{W}\mathbf{O}$, we obtain the eigenmatrix \mathbf{T} and the load matrix of the WOPCA model. The process data vectors are projected into two orthogonal subspaces (principal and residual spaces) by the WOPCA model, and the statistics T_S^2 and SPE_S are constructed in each subspace, shown as formula (17) and formula (18).

$$T_S^2 = \mathbf{X}^T \mathbf{P} \mathbf{\Lambda}^{-1} \mathbf{P}^T \mathbf{X} \quad (17)$$

$$SPE_S = \mathbf{X} (\mathbf{I} - \mathbf{P} \mathbf{P}^T) \mathbf{X}^T \quad (18)$$

where, $\mathbf{\Lambda}$ is the covariance matrix of the original data after dimensionality reduction data \mathbf{Y} , and \mathbf{I} is the unit matrix.

The control limits $T_{S\text{lim}}^2$ and $SPE_{S\text{lim}}$ of the statistics T_S^2 and SPE_S are obtained by using kernel density estimation.

Exponential global neighborhood preserving embedding (EGNPE)

NPE algorithm only considers the local manifold structure in the modeling process and ignores the global information of the data. Therefore, we consider the global and local information of the data at the same time, and highlight the important global and local data information through exponential transformation, which important information of the data becomes larger and non-important information of the data becomes smaller, the EGNPE model is established in space Ω_{NS} . For sample $\mathbf{X} = \sum_{i=1}^n \mathbf{x}_i$, the dimensionality reduction projection is performed by EGNPE algorithm to obtain $\mathbf{Y} = \sum_{i=1}^n \mathbf{y}_i$ and $\mathbf{Y} = \mathbf{A}^T \mathbf{X}$, where \mathbf{A} is the mapping matrix from high-dimensional space to low-dimensional space.

The modeling process is as follows:

The objective function of the global structure is shown as formula (19):

$$\begin{aligned} J_G &= \max \sum_{i=1}^n \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2 \\ &= \max \sum_{i=1}^n \mathbf{A}^T (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{A} \\ &= \max \mathbf{A}^T \mathbf{C} \mathbf{A} \end{aligned} \quad (19)$$

where, $\mathbf{C} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T$, $\bar{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i$, $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$.

Then the objective function of the exponential global structure is:

$$J_{EG} = \max \mathbf{A}^T \mathbf{D} \mathbf{A} \quad (20)$$

where, $\mathbf{D} = \exp(\lambda_i^C)$, and λ_i^C denotes the i th eigenvalue of matrix \mathbf{C} .

The objective function of the local structure is shown as formula (21).

$$\begin{aligned} J_{NPE} &= \min \sum_{i=1}^n \left(\mathbf{y}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{y}_j \right)^2 \\ &= \min \mathbf{A}^T \mathbf{X} (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{X}^T \mathbf{A} \\ &= \min \mathbf{A}^T \mathbf{Q} \mathbf{A} \end{aligned} \quad (21)$$

where, $\mathbf{Q} = \mathbf{X} (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{X}^T$.

Then the objective function of the exponential local structure is:

$$J_{ENPE} = \min \mathbf{A}^T \mathbf{P} \mathbf{A} \quad (22)$$

where, $\mathbf{P} = \exp(\lambda_i^Q)$, and λ_i^Q denotes the i th eigenvalue of matrix \mathbf{Q} .

The objective function of the EGNPE model can be obtained by combining the global and local structures, shown as formula (23):

$$J_{EGNPE} = \max \frac{\mathbf{A}^T \mathbf{D} \mathbf{A}}{\mathbf{A}^T \mathbf{P} \mathbf{A}} \quad (23)$$

Formula (16) is solved through Laplace transform as shown in formula (24).

$$\mathbf{D} \mathbf{A} = \lambda \mathbf{P} \mathbf{A} \quad (24)$$

After dimensionality reduction, the space Ω_{NS} is divided into the feature space and the residual space, and the statistics T_{NS}^2 and SPE_{NS} are constructed in the two divided spaces. For the online data \mathbf{x}_{new} , which dimensionality reduction is represented as $\mathbf{y}_{new} = \mathbf{A}^T \mathbf{x}_{new}$, the statistics are constructed as shown in formulas (25) and (26).

$$T_{NS}^2 = \mathbf{y}_{new} \mathbf{\Lambda}^{-1} \mathbf{y}_{new}^T \quad (25)$$

where, $\mathbf{\Lambda}^{-1} = \frac{1}{n-1} \sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^T$, and $\mathbf{\Lambda}$ is the covariance matrix of data \mathbf{Y} after dimensionality reduction of the original data.

$$SPE_{NS} = \|\mathbf{x}_{new} - \mathbf{A} \mathbf{y}_{new}\|^2 \quad (26)$$

The control limits T_{NSlim}^2 and SPE_{NSlim} of the statistics T_{NS}^2 and SPE_{NS} are obtained by using kernel density estimation.

The method of kernel density estimation: given a univariate kernel function is shown in the following formula (27):

$$f(j) = \frac{1}{n} \sum_{i=1}^n K_h(j - j_i) = \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{j - j_i}{\sigma}\right) \quad (27)$$

where, j is the sample data, j_i is the observation value, σ is the window width, n is the number of observation values, and K is the kernel function. In this paper, the Gaussian kernel function is selected, and the test level $\alpha = 0.95$ can be obtained by formula (28) and (29) to find the control limits.

$$\begin{aligned} \int_{-\infty}^{SPE_\alpha} f(SPE) d(SPE) &= \int_{-\infty}^{SPE_\alpha} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{SPE - SPE_i}{\sigma}\right) d(SPE) \\ &= \int_{-\infty}^{SPE_\alpha} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(SPE - SPE_i)^2}{2\sigma^2}\right) \right\} d(SPE) \\ &= \alpha \end{aligned} \quad (28)$$

$$\begin{aligned} \int_{-\infty}^{T_\alpha^2} f(T^2) d(T^2) &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma} \sum_{i=1}^n K\left(\frac{T^2 - T_i^2}{\sigma}\right) d(T^2) \\ &= \int_{-\infty}^{T_\alpha^2} \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^n \left\{ \exp\left(-\frac{(T^2 - T_i^2)^2}{2\sigma^2}\right) \right\} d(T^2) \\ &= \alpha \end{aligned} \quad (29)$$

Bayesian joint monitoring

The statistics T_S^2 , SPE_S , T_{NS}^2 and SPE_{NS} constructed in the two divided subspaces Ω_S and Ω_{NS} are implemented to monitor each subspace, and a joint statistic is established by Bayesian inference to consider the overall monitoring of the process.

In space Ω_S , the fault probability of T_S^2 is shown in formula (30):

$$P_{T_S^2(F|X_S)} = \frac{P_{T_S^2(\Omega_S|F)}P_{T_S^2F}}{P_{T_S^2(\Omega_S|N)}P_{T_S^2N} + P_{T_S^2(\Omega_S|F)}P_{T_S^2F}} \quad (30)$$

where, N and F represent normal and fault conditions respectively, the conditional probabilities $P_{T_S^2N} = \alpha$ and $P_{T_S^2F} = 1 - \alpha$, α is confidence probability, $P_{T_S^2(\Omega_S|N)}$ and $P_{T_S^2(\Omega_S|F)}$ are conditional probabilities calculated as shown in formula (31) and formula (32):

$$P_{T_S^2(\Omega_S|N)} = \exp\left(-\frac{T_S^2}{T_{S\lim}^2}\right) \quad (31)$$

$$P_{T_S^2(\Omega_S|F)} = \exp\left(-\frac{T_{S\lim}^2}{T_S^2}\right) \quad (32)$$

where, $T_{S\lim}^2$ is the control limit of the statistic T_S^2 by using kernel density estimation.

According to the same principle, we can obtain the follows:

- (1) In space Ω_S , the fault probability of SPE_S is $P_{SPE_S(F|X_S)}$;
- (2) In space Ω_{NS} , the fault probability of T_{NS}^2 is $P_{T_{NS}^2(F|X_{NS})}$;
- (3) In space Ω_{NS} , the fault probability of SPE_{NS} is $P_{SPE_{NS}(F|X_{NS})}$;

The Bayesian statistic is obtained from formula (33):

$$BIC = \frac{P_{T_S^2(F|X_S)}^2 + P_{SPE_S(F|X_S)}^2 + P_{T_{NS}^2(F|X_{NS})}^2 + P_{SPE_{NS}(F|X_{NS})}^2}{P_{T_S^2(F|X_S)} + P_{SPE_S(F|X_S)} + P_{T_{NS}^2(F|X_{NS})} + P_{SPE_{NS}(F|X_{NS})}} \quad (33)$$

where, the control limit of the statistic BIC is $1 - \alpha$, and α takes the value of 0.95.

Monitoring steps of batch process based on WOPCA-EGNPE algorithm

Batch process monitoring based on WOPCA-EGNPE algorithm includes two parts: offline modeling and online monitoring. The monitoring flowchart is shown in Fig. 1.

Offline modeling

Step1: Under normal working conditions, sample data are collected to form a training sample set $\mathbf{X} \in R^{m \times n}$, and \mathbf{X} is unfolded into two-dimensional data in a batch-variable manner. That is, for $\mathbf{X}(I \times J \times K)$, we unfold it along the batch direction to get a two-dimensional matrix $\mathbf{X}(I \times JK)$, and then it is arranged along the variable direction as $\mathbf{X}(IK \times J)$ (I represents the batch of process data, J represents the number of variables, K represents the number of sampling points).

Step 2: ADF test is used to determine whether the variable is nonstationary, and the preprocessed sample data are divided into the nonstationary space Ω_{NS} and the stationary space Ω_S .

Step 3: In the nonstationary space Ω_{NS} , the method of cointegration analysis is used to obtain the cointegration variables \mathbf{B} and a stationary residual series $\boldsymbol{\gamma}$. For the obtained stationary residual series $\boldsymbol{\gamma}$, we perform EGNPE algorithm to obtain the feature mapping matrix \mathbf{A}_{NS} , thereby obtaining the low-dimensional representation of the high-dimensional series: $\mathbf{Y}_{NS} = \mathbf{A}_{NS}^T \boldsymbol{\gamma}$.

Step 4: In the stationary space Ω_S , WOPCA algorithm is performed to get the feature mapping matrix \mathbf{A}_S and obtain the low-dimensional representation: $\mathbf{Y}_S = \mathbf{A}_S \mathbf{X}_S$.

Step 5: Kernel density estimation method is used to calculate the control limits $T_{NS\lim}^2$ and $SPE_{NS\lim}$ of the statistics T_{NS}^2 and SPE_{NS} in the nonstationary space; similarly, we find the control limits $T_{S\lim}^2$ and $SPE_{S\lim}$ of the statistics T_S^2 and SPE_S in the stationary space Ω_S .

Step 6: According to Bayesian inference, the Bayesian statistic BIC and control limit BIC_{\lim} can be obtained by formula (33).

Online monitoring

Step 1: The collected online data \mathbf{X}_{new} are preprocessed.

Step 2: According to the variable division of the offline process, the online data are divided into two subspaces: nonstationary space Ω_{NSnew} and stationary space Ω_{Snew} .

Step 3: In the nonstationary space Ω_{NSnew} , we use the cointegration variables \mathbf{B} obtained offline to calculate the online stationary residual series $\boldsymbol{\gamma}_{new}$, and use the feature mapping matrix \mathbf{A}_{NS} obtained offline to obtain the low-dimensional representation of the online stationary residual series: $\mathbf{Y}_{NSnew} = \mathbf{A}_{NS}^T \boldsymbol{\gamma}_{new}$.

Step 4: In the stationary space Ω_{Snew} , we obtain the low-dimensional representation from the offline mapping matrix \mathbf{A}_S : $\mathbf{Y}_{Snew} = \mathbf{A}_S^T \mathbf{X}_{Snew}$.

Step 5: The statistics T_{NSnew}^2 and SPE_{NSnew} are calculated in the space Ω_{NSnew} , and the statistics T_{Snew}^2 and SPE_{Snew} are calculated in the space Ω_{Snew} . According to Bayesian inference, we calculate the Bayesian statistic by formula (33).

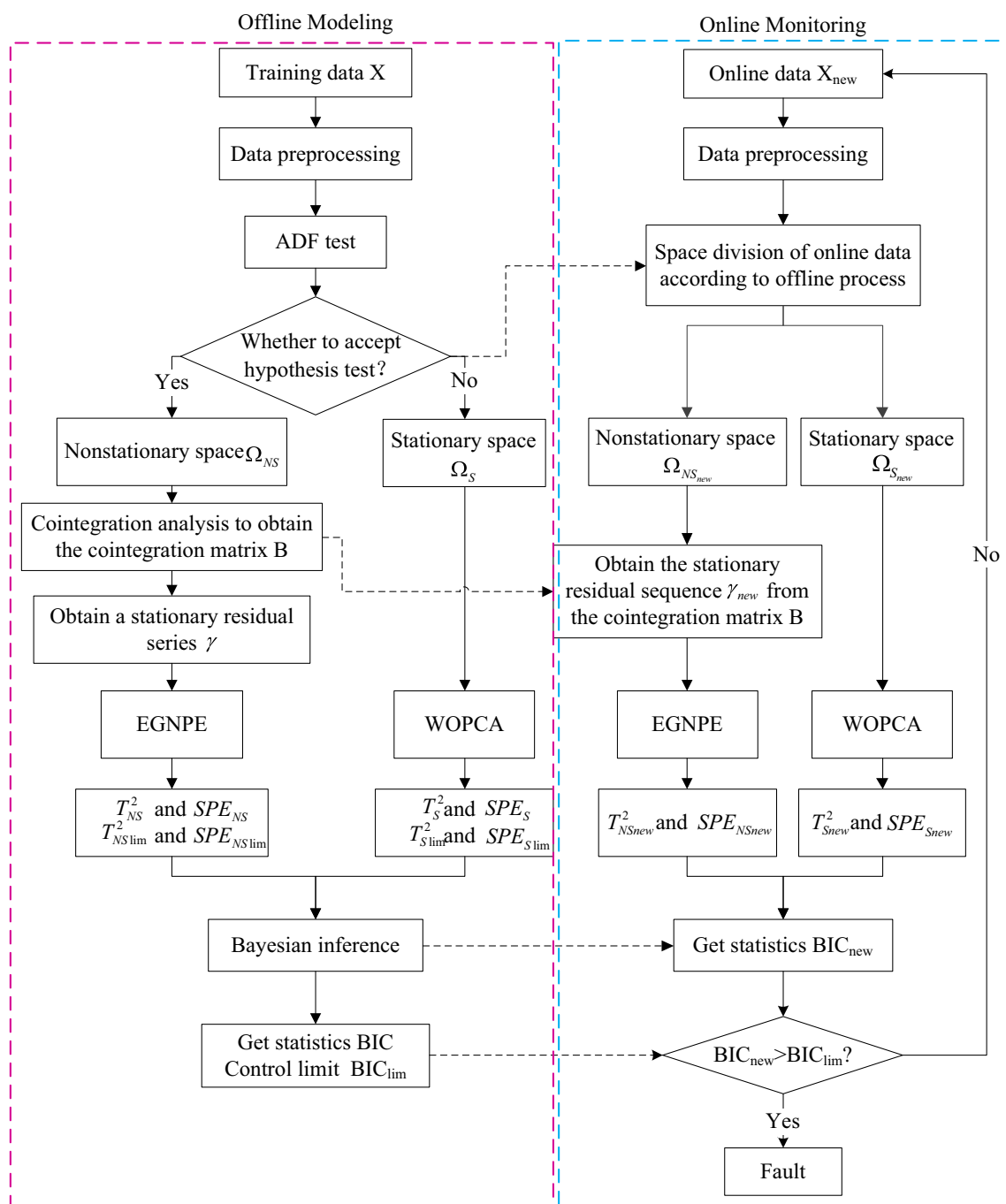


Fig. 1 Flow chart of WOPCA-EGNPE algorithm

Step 6: If BIC_{new} is larger than BIC_{lim} , it indicates that the fault has occurred; otherwise, it is normal.

The results and analysis of the penicillin fermentation process simulation experiment

As an antibiotic, penicillin has a wide range of clinical medical value, and its production process is a typical batch process. In this paper, we use the Pensim2.0 simulation platform to generate penicillin fermentation process data. The platform is a simulation software for modeling, monitoring,

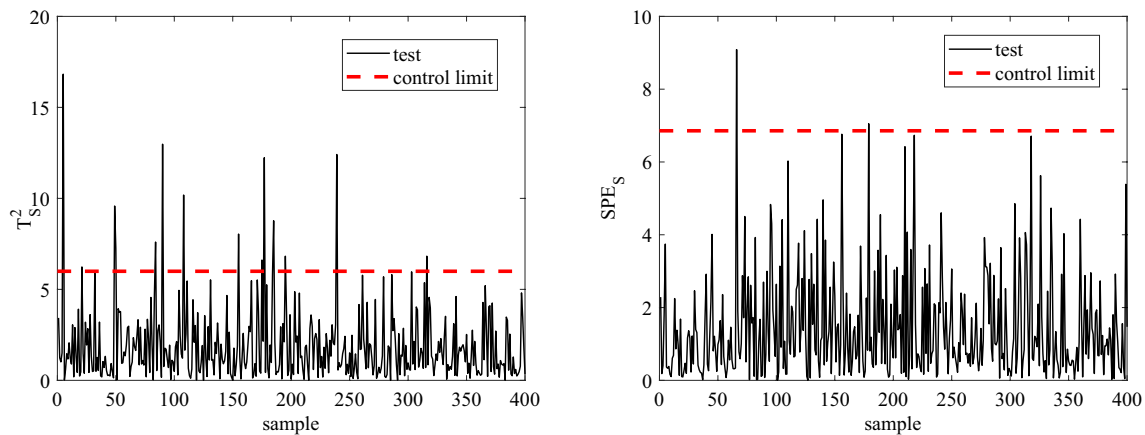


Fig. 2 Monitoring diagram of statistics in stationary space under fault F1

Table 1 Parameter setting of initial conditions in penicillin fermentation process

No	Process variables	Set value	Variable unit
1	Substrate concentration	15	g/L
2	Dissolved oxygen concentration	1.16	g/L
3	Bacteria initial concentration	0.1	g/L
4	Penicillin concentration	0	g/L
5	Culture volume	100	L
6	CO ₂ concentration	0.5	mmol/L
7	pH	5	/
8	Temperature	298	K
9	Aeration rate	8.6	L/h
10	Agitator power	30	W
11	Substrate feed rate	0.042	L/h
12	Substrate feed temperature	296	K
13	Temperature set point	298	K

and controlling the penicillin fermentation process independently developed by the research team of Professor Cinar of Illinois State Institute of Technology in the United States in 2002 (Birol et al. 2002). We apply the platform to generate 30 batches data under normal operating condition, and the reaction time of each batch is set to 400 h, the sampling time is set to 1h, and 16 process variables are selected as monitoring variables. Parameter setting of initial conditions of penicillin fermentation process is set, as shown in Table 1. To simulate the noise disturbance of actual process, Gaussian white noise are introduced to all variables.

For the process data, in the case of a 5% confidence threshold, the threshold is -1.9414 , ADF test is used to determine the stationarity of the variables, as shown in Table 2.

According to Table 2, variables 5, 14, 15, and 16 are stationary, and other variables are nonstationary, that is,

Table 2 ADF test of process variables

No	Process variables	ADF test value	Stationarity
1	Aeration rate (L/h)	-0.0504	Nonstationary
2	Agitator power (W)	0.0306	Nonstationary
3	Substrate feed flow rate (L/h)	0.0293	Nonstationary
4	Substrate feed flow temperature (K)	-0.3460	Nonstationary
5	Substrate concentration (L/h)	-11.1379	stationary
6	Dissolved oxygen concentration (%)	-0.7465	Nonstationary
7	Biomass concentration (g/L)	5.3686	Nonstationary
8	Penicillin concentration (g/L)	12.1669	Nonstationary
9	Reactor volume (L)	20.2279	Nonstationary
10	CO ₂ (mmol/L)	0.3956	Nonstationary
11	PH	0.0689	Nonstationary
12	Fermentation temperature (K)	0.0068	Nonstationary
13	Generated heat (kcal/h)	5.3694	Nonstationary
14	Acid flow rate (ml/h)	-10.5589	Stationary
15	Base flow rate (ml/h)	-4.2308	Stationary
16	Cold water flow rate (L/h)	-12.0209	Stationary

stationary space $\Omega_S = (x_5, x_{14}, x_{15}, x_{16})$ and nonstationary space $\Omega_{NS} = (x_1, \dots, x_4, x_6, \dots, x_{13})$.

In this paper, we generate 4 batches of fault data by setting different fault types and amplitudes, as shown in Table 3, the fault data also introduce noise disturbances to simulate the real process.

Table 4 is the detection rates of MPCA, MNPE, and the proposed WOPCA-EGNPE for different faults.

Table 5 is the false alarm rates of MPCA, MNPE, and WOPCA-EGNPE for different faults.

From Table 4, for MPCA and MNPE algorithms, we can see that the detection rates of statistic T^2 are low, which

Table 3 Fault batches in the penicillin fermentation process

Fault No	Fault variable name	Disturbance type	Amplitude (%)	Fault introduction period
F1	Agitator power	Step	2	200–300 h
F2	Agitator power	Ramp	4	300–400 h
F3	Substrate feeding rate	Step	6	100–300 h
F4	Substrate feeding rate	Ramp	0.8	250–400 h

Table 4 Fault detection rate of each fault by different methods

Fault no	MPCA		MNPE			WOPCA-EGNPE			
	T^2	SPE	T^2	SPE	T_S^2	SPE_S	T_{NS}^2	SPE_{NS}	BIC
F1	0.10	0.98	0.17	0.21	0.02	0.00	0.31	0.85	0.98
F2	0.78	0.91	0.55	0.76	0.06	0.00	0.64	0.89	0.94
F3	0.85	0.97	0.53	0.61	0.83	0.91	0.67	0.72	1.00
F4	0.88	0.90	0.92	0.94	0.90	0.92	0.92	0.94	0.96

Table 5 False alarm rate of each fault by different methods

Fault no	MPCA		MNPE		WOPCA-EGNPE
	T^2	SPE	T^2	SPE	BIC
F1	0.02	0.49	0.11	0.00	0.10
F2	0.23	0.28	0.13	0.12	0.02
F3	0.25	0.43	0.17	0.01	0.23
F4	0.09	0.29	0.13	0.36	0.08

indicate that the variable relationship under normal working condition represented by these two statistical models is destroyed and the working condition changes are not obvious; for the proposed WOPCA-EGNPE algorithm, the statistics T_S^2 and SPE_S of fault F1 and fault F2 in the stationary space Ω_S are almost 0, which indicate that the fault change of the agitator rate would not cause the changes of working conditions in the stationary space, that is, no fault occurs in the stationary space. The monitoring diagram of statistics in the stationary space under fault F1 is shown in Fig. 2. It can be seen from Fig. 2 that a few points exceed the control limit and cause fault false alarms, the main reason is that we simulate the actual fermentation process and introduce Gaussian white noise to each variable, which leads to fault alarms in the monitoring process.

For the proposed WOPCA-EGNPE algorithm in this paper, the statistics T_S^2 and SPE_S of fault F3 and fault F4 in the stationary space Ω_S have higher detection rates. Since fault F3 and fault F4 are substrate feeding rate faults of different signal types, the substrate feeding rate faults would cause the variable relationship changes in the stationary space Ω_S . From Fig. 3, for stationary space variables, we can see that the occurrence of substrate feeding rate fault

would cause the acid flow rate, base flow rate, and cold water flow rate to change. The acid flow rate changes significantly after about 325h, which results in a large number of false alarms after this time. Figure 4 is the monitoring diagram of statistics in the stationary space under fault F3, which shows that the fault can be better detected when the fault occurs. A large number of fault alarms occur during the 300 ~ 400h period, and the main reason is that the substrate feeding rate fault causes a large change of the acid flow rate.

According to Tables 3 and 4, the statistics BIC of the proposed WOPCA-EGNPE algorithm has a higher fault detection rate, a lower fault false alarm rate, and a better monitoring effect.

Figure 5 is the statistical monitoring diagram of MPCA, MNPE, and the proposed WOPCA-EGNPE algorithm under fault F1. Fault F1 is a step disturbance with an amplitude of 2% added to the agitator rate between 200–300 h. Figure 5a is the monitoring diagram of statistics T^2 and SPE of MPCA algorithm, it can be seen that statistic T^2 has very few fault false alarms and the detection rate is also very low, while statistic SPE has a high detection rate and the fault false alarms are serious and the monitoring effect is not good; Fig. 5b is the monitoring diagram of statistics T^2 and SPE of MNPE algorithm, it can be seen that the detection of the two statistics have very few faults false alarms, but the fault miss alarms are serious; Fig. 5c is the statistical monitoring diagram of the proposed WOPCA-EGNPE algorithm, which has a better monitoring effect than other algorithms.

Figure 6 is the statistical monitoring diagram of MPCA, MNPE, and the proposed WOPCA-EGNPE algorithm under fault F2. Fault F2 is a ramp disturbance with an amplitude of 4% added to the agitator rate between 300–400 h. Figure 6a is the monitoring diagram of statistics T^2 and SPE of MPCA algorithm, it can be seen that the fault is detected at 342 h

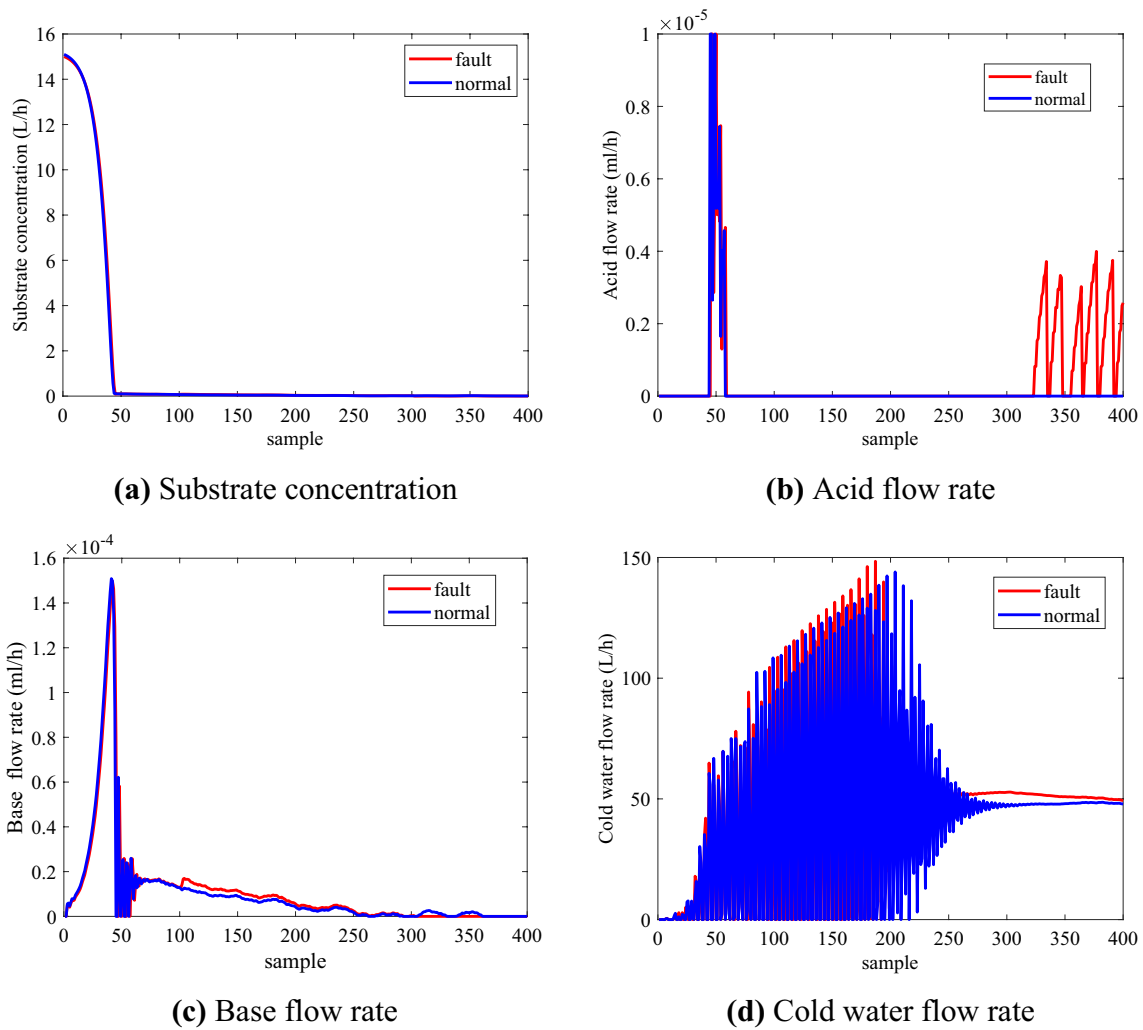


Fig. 3 Stationary variable curve under fault F3 (a substrate concentration, b Acid flow rate, c Base flow rate, and d Cold water flow rate)

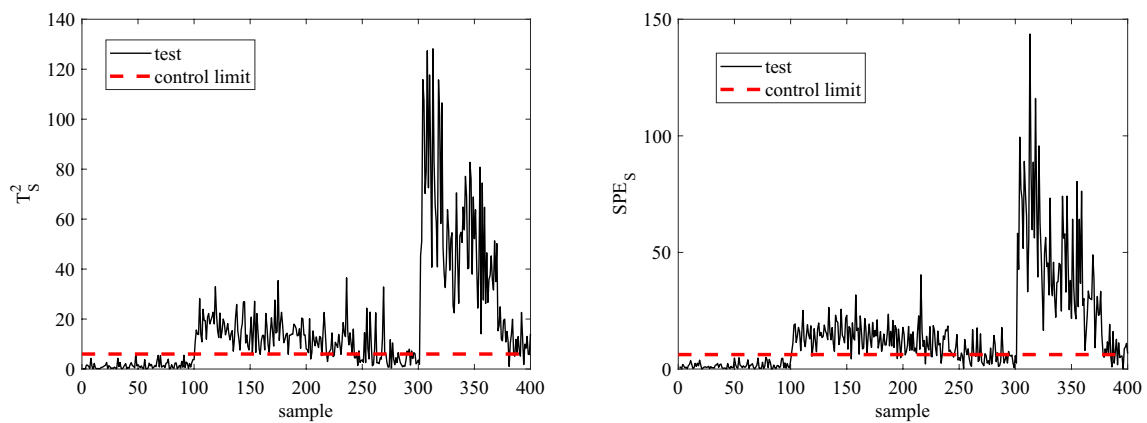


Fig. 4 Monitoring diagram of stationary space statistics under fault F3

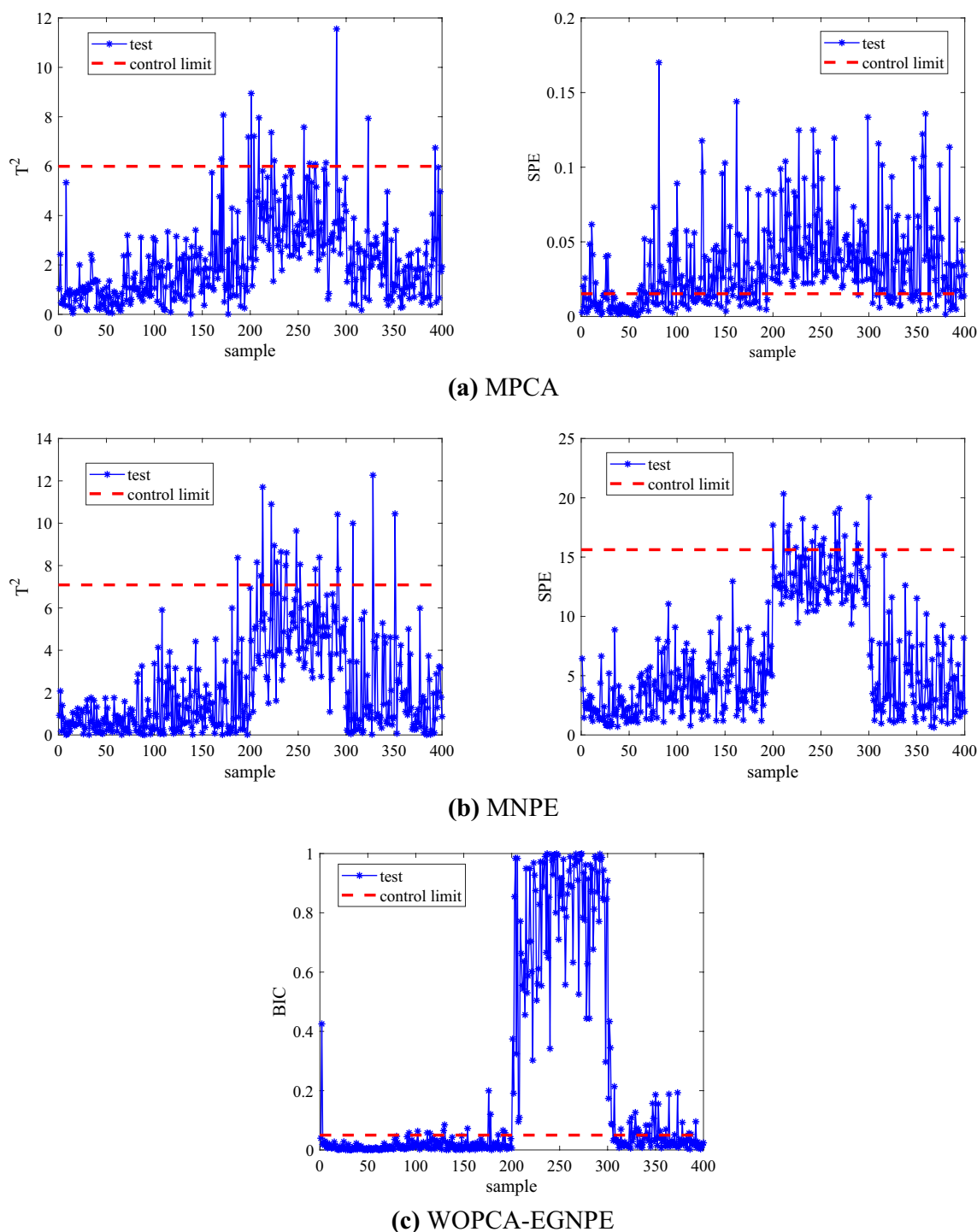


Fig. 5 Statistics monitoring diagram under fault F1 (**a** MPCA, **b** MNPE, and **c** WOPCA-EGNPE algorithm)

and 309 h, respectively, and there is a large delay in fault monitoring; Fig. 5b is the monitoring diagram of statistics T^2 and SPE of MNPE algorithm, which has a lower fault false alarm rate than MPCA algorithm, and the fault detects at 335h and 324h respectively; Fig. 6c shows the statistical BIC monitoring diagram of the proposed WOPCA-EGNPE

algorithm, which detects the fault at 304h, and has more timely fault detection, higher detection rate and smaller false alarm rate.

Figure 7 is the statistical monitoring diagram of MPCA, MNPE, and the proposed WOPCA-EGNPE algorithm under fault F3. Fault F3 is the substrate feeding rate added a step

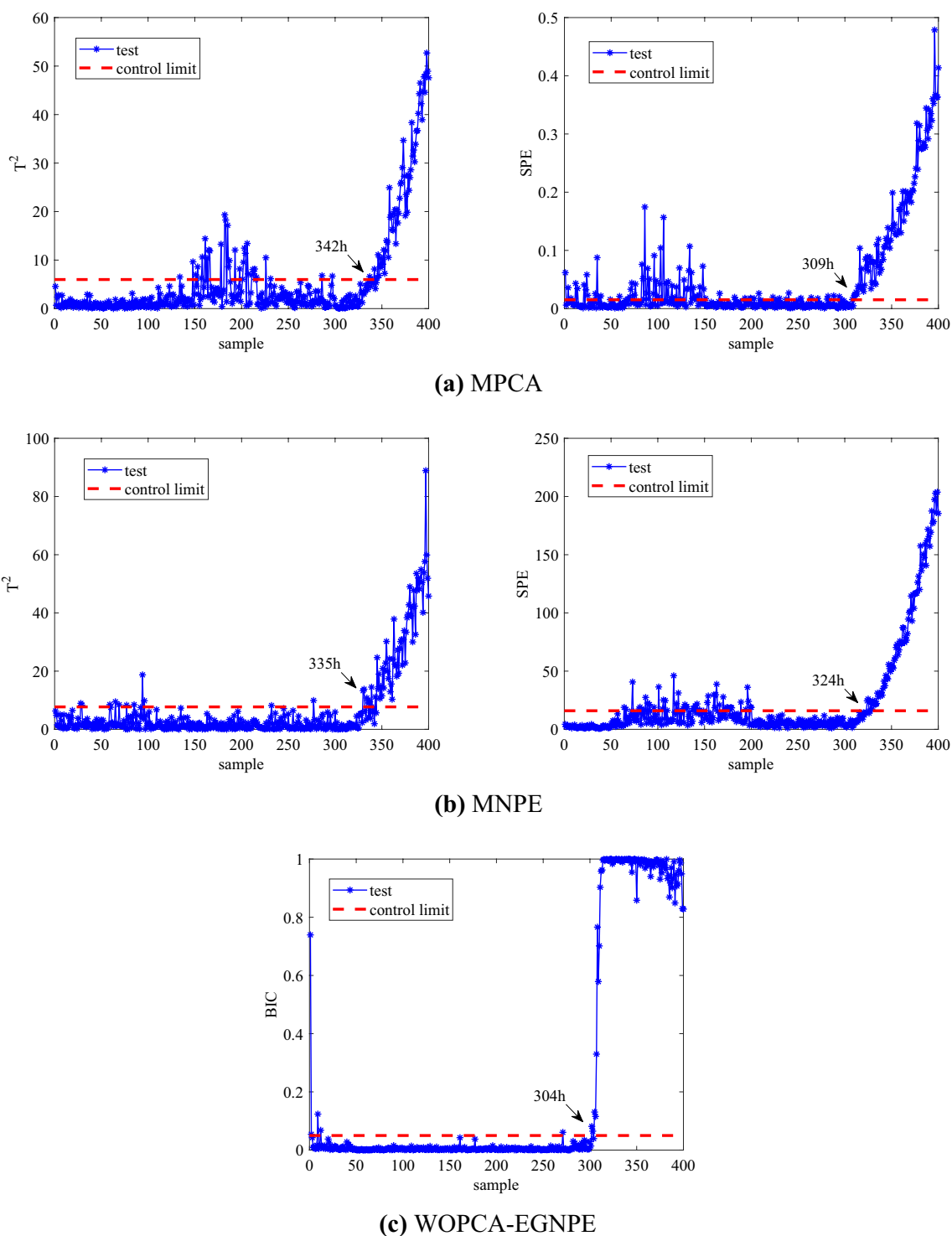


Fig. 6 Statistics monitoring diagram under fault F2 (**a** MPCA, **b** MNPE, and **c** WOPCA-EGNPE algorithm)

disturbance with an amplitude of 6% between 100 ~ 300h. We can see that the proposed WOPCA-EGNPE algorithm has the better monitoring effect. However, the proposed algorithm also has higher false alarm rate after 300h, the reason is that the acid flow rate significantly changes after

300h under fault F3. In general, the monitoring effect of our proposed algorithm is better than other algorithms.

Figure 8 is the stacked histogram of the average fault detection rate and the average false alarm rate under the four faults. The horizontal coordinates 1, 2, 3, 4, and 5

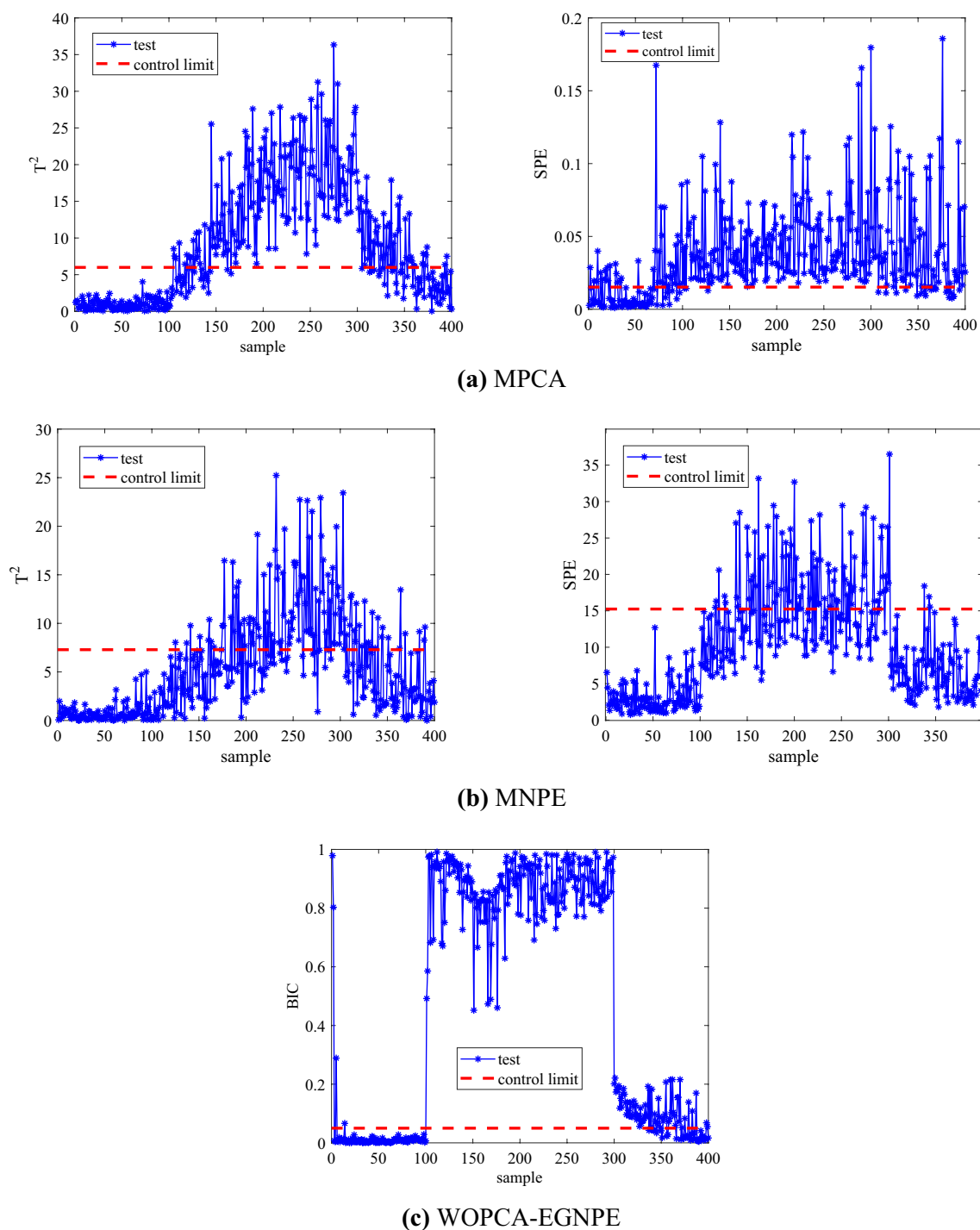


Fig. 7 Statistics monitoring diagram under fault F3 (**a** MPCA, **b** MNPE, and **c** WOPCA-EGNPE algorithm)

indicate the statistics of MPCA, MPCA, MNPE, MNPE, and WOPCA-EGNPE, respectively, while the blue color shows the average detection rate of different methods and the red color shows the average false alarm rate of different methods. It can be visualized that the proposed WOPCA-EGNPE algorithm has a better fault detection

rate and a lower fault false alarm rate in the penicillin fermentation process, and has a better fault monitoring effect.

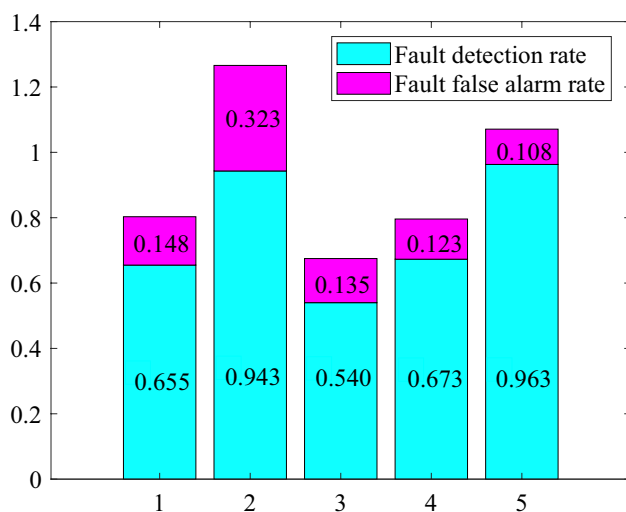


Fig. 8 Stacked histogram of fault detection rate and false alarm rate

Conclusion

We propose a WOPCA-EGNPE algorithm based on variable division for fault monitoring in batch process. Since the variables of batch process have the characteristic of stationary and nonstationary mixture distribution, the variables are divided into the stationary space and nonstationary space by judging their stationarity. WOPCA algorithm and EGNPE algorithm are used to model in the stationary space and the nonstationary space respectively, the fault detection statistics in two spaces are obtained respectively, and Bayesian method is utilized to construct a joint statistical indicator to realize process monitoring. Compared with MPCA algorithm and MNPE algorithm, the monitoring results verify that the fault detection effectiveness of the proposed algorithm in the penicillin fermentation process is better. Because a variable fault often affects the other related variable fault, resulting in multiple faults, in the future research, corresponding monitoring and diagnosis technologies will be studied to identify multiple faults.

Acknowledgements This research work has been awarded by the National Key Research and Development Plan (2020YFB1713600), The Science and Technology Project of Gansu Province (21JR7RA206, 21YF5GA072), the Open Fund project of Gansu Provincial Key Laboratory of Advanced Control for Industrial Process (2022KX07), the Industrial Support Project of Education Department of Gansu Province (2021CYZC-02).

Declarations

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

References

- Aggoun L, Chetouani Y (2021) Fault detection strategy combining NARMAX model and Bhattacharyya distance for process monitoring. *J Franklin Inst* 358(3):2212–2228
- Ajewole K, Adejuwon S, Jemilohun VJJM (2020) Test for stationarity on inflation rates in nigeria using augmented dickey Fuller test and Phillips-Persons test. *J Math* 16:11–14
- Birol G, Undey C, Parulekar SJ, Cinar A (2002) A morphologically structured model for penicillin production. *Biotechnol Bioeng* 77(5):538–552
- Chen J, Liu J (2001) Derivation of function space analysis based PCA control charts for batch process monitoring. *Chem Eng Sci* 56(10):3289–3304
- Chen Q, Kruger U, Leung AY (2009) Cointegration testing method for monitoring nonstationary processes. *Ind Eng Chem Res* 48(7):3533–3543
- Gao X, Xu Z, Li Z, Wang P (2020) Batch process monitoring using multiway Laplacian autoencoders. *Can J Chem Eng* 98(6):1269–1279
- Granger CW (2004) Time series analysis, cointegration, and applications. *Am Econ Rev* 94(3):421–425
- Gunther JC, Conner JS, Seborg DE (2009) Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture. *J Process Control* 19(5):914–921
- He QP, Wang J (2018) Statistical process monitoring as a big data analytics tool for smart manufacturing. *J Process Control* 67:35–43
- He X, Cai D, Yan S, Zhang H-J (2005) Neighborhood preserving embedding. *Tenth IEEE Int Conf Comput vis* 1:1208–1213
- Huang H-B, Yi T-H, Li H-N (2017) Bayesian combination of weighted principal-component analysis for diagnosing sensor faults in structural monitoring systems. *J Eng Mech* 143(9):04017088
- Jiang Q, Yan S, Yan X, Yi H, Gao F (2019) Data-driven two-dimensional deep correlated representation learning for nonlinear batch process monitoring. *IEEE Trans Ind Inf* 16(4):2839–2848
- Li G, Qin SJ, Yuan T (2014) Nonstationarity and cointegration tests for fault detection of dynamic processes. *IFAC Proc Vol* 47(3):10616–10621
- Li T, Yang D, Wei Q, Zhang H (2017) Batch process fault monitoring based on LPGD-kNN and its applications in semiconductor industry. *International conference on neural information processing*. Springer, Berlin, pp 534–544
- Li Z, Chang P, Wang K, Wang P (2021) The batch process fault monitoring using adversarial auto-encoder and k-nearest neighbor rule. In: *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, pp 41–48
- Monroy I, Villez K, Graells M, Venkatasubramanian V (2011) Dynamic process monitoring and fault detection in a batch fermentation process: comparative performance assessment between MPCA and BDPCA. *Computer aided chemical engineering*. Elsevier, Amsterdam, pp 1371–1375
- Muñoz CA, Telen D, Nimmegeers P, Van Impe J (2018) Feature extraction for batch process monitoring and fault detection via simultaneous data scaling and training of tensor based models. *IFAC-PapersOnLine* 51(24):433–440
- Nawaz M, Maulud AS, Zabiri H, Taqvi SAA, Idris A (2021) Improved process monitoring using the CUSUM and EWMA-based multiscale PCA fault detection framework. *Chin J Chem Eng* 29:253–265
- Peng C, Ruiwei L (2021) Process monitoring of batch process based on overcomplete broad learning network. *Eng Appl Artif Intell* 99:104139
- Stubbs S, Zhang J, Morris J (2013) Multiway interval partial least squares for batch process performance monitoring. *Ind Eng Chem Res* 52(35):12399–12407

- Sun H, Zhang S, Zhao C, Gao F (2017) A sparse reconstruction strategy for online fault diagnosis in nonstationary processes with no a priori fault information. *Ind Eng Chem Res* 56(24):6993–7008
- Wang Y, Si Y, Huang B, Lou Z (2018) Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. *Can J Chem Eng* 96(10):2073–2085
- Wang Y, Pan Z, Yuan X, Yang C, Gui W (2020) A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans* 96:457–467
- Yu J, Wang K, Ye L, Song Z (2019) Accelerated kernel canonical correlation analysis with fault relevance for nonlinear process fault isolation. *Ind Eng Chem Res* 58(39):18280–18291
- Yunus MYM, Zhang J, Al-Amshawee SK (2020) Utilization of fault detection performance in process monitoring. *J Chem Eng Ind Biotechnol* 6(1):1–11
- Zhang H, Tian X, Deng X (2017) Batch process monitoring based on multiway global preserving kernel slow feature analysis. *IEEE Access* 5:2696–2710
- Zhang C, Xu T, Li Y (2020) A novel fault detection scheme based on difference in independent component for reliable process monitoring: Application on the semiconductor manufacturing processes. *J Chem Eng Jpn* 53(7):313–320
- Zhang H, Deng X, Zhang Y, Hou C, Li C (2021a) Dynamic non-linear batch process fault detection and identification based on two-directional dynamic kernel slow feature analysis. *Can J Chem Eng* 99(1):306–333
- Zhang M, Yi Y, Cheng W (2021b) Multistage condition monitoring of batch process based on multi-boundary hypersphere SVDD with modified bat algorithm. *Arab J Sci Eng* 46(2):1647–1661
- Zhao C, Huang B (2018) A full-condition monitoring method for non-stationary dynamic chemical processes with cointegration and slow feature analysis. *AIChE J* 64(5):1662–1681
- Zhao X, Mou M (2021) Markov chain neighborhood sparse preserving graph embedding based on tensor factorization for batch process monitoring. *IEEE Access* 9:16211–16224
- Zhao X, Wang T, Hui Y (2016) MGNPE-LICA algorithm for fault diagnosis of batch process. *Can J Chem Eng* 94(10):1947–1954

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

证书号第7768514号



专利公告信息

发明专利证书

发明名称：一种基于变分自编码器的无线通信网络中继选择方法

专利权人：兰州理工大学

地址：730050 甘肃省兰州市七里河区兰工坪路287号

发明人：刘凯;赵小强;惠永永;宋昭漾;牟森;张妍;强睿儒

专利号：ZL 2023 1 0971760.6

授权公告号：CN 116916410 B

专利申请日：2023年08月03日

授权公告日：2025年02月28日

申请日时申请人：兰州理工大学

申请日时发明人：刘凯;赵小强;惠永永;宋昭漾;牟森;张妍;强睿儒

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。
专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

申长雨



证书号第6682099号



发明专利证书

发明名称：基于孪生深度邻域保持嵌入网络的间歇过程故障检测方法

发明人：刘凯;赵小强;宋昭漾;惠永永;强睿儒;张妍

专利号：ZL 2023 1 0559773.2

专利申请日：2023年05月18日

专利权人：兰州理工大学

地址：730050 甘肃省兰州市七里河区兰工坪路287号

授权公告日：2024年02月02日

授权公告号：CN 116502168 B

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，颁发发明专利证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。专利权期限为二十年，自申请日起算。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。



局长
申长雨

申长雨



证书号第7565884号



专利公告信息

发明专利证书

发明名称：基于自适应随机配置网络的间歇过程故障预测方法

专利权人：兰州理工大学

地址：730050 甘肃省兰州市兰工坪287号

发明人：刘凯;赵小强;宋昭漾;牟淼;强睿儒;张妍;张亚洲;柴靖轩

专利号：ZL 2024 1 0220859.7

授权公告号：CN 118151627 B

专利申请日：2024年02月28日

授权公告日：2024年11月29日

申请日时申请人：兰州理工大学

申请日时发明人：刘凯;赵小强;宋昭漾;牟淼;强睿儒;张妍;张亚洲;柴靖轩

国家知识产权局依照中华人民共和国专利法进行审查，决定授予专利权，并予以公告。
专利权自授权公告之日起生效。专利权有效性及专利权人变更等法律信息以专利登记簿记载为准。

局长
申长雨

申长雨

