Local–Global Modeling and Distributed Computing Framework for Nonlinear Plant-Wide Process Monitoring With Industrial Big Data

Qingchao Jiang, *Senior Member, IEEE*, Shifu Yan, Hui Cheng, and Xuefeng Yan, *Member, IEEE*

***Abstract*— Industrial big data and complex process nonlinearity have introduced new challenges in plant-wide process monitoring. This article proposes a local–global modeling and distributed computing framework to achieve efficient fault detection and isolation for nonlinear plant-wide processes. First, a stacked autoencoder is used to extract dominant representations of each local process unit and establish the local inner monitor. Second, mutual information (MI) is used to determine the neighborhood variables of a local unit. Afterward, a joint representation learning is then performed between the local unit and the neighborhood variables to extract the outer-related representations and establish the outer-related monitor for the local unit. Finally, the outer-related representations from all process units are used to establish global monitoring systems. Given that the modeling of each unit can be performed individually, the computation process can be efficiently completed with different CPUs. The proposed modeling and monitoring method is applied to the Tennessee Eastman (TE) and laboratory-scale glycerol distillation processes to demonstrate the feasibility of the method.**

***Index Terms*— Deep neural network (DNN), distributed computing, industrial big data, local–global modeling, nonlinear plant-wide processes.**

I. Introduction

PROCESS monitoring, which is crucial in maintaining favorable operating conditions and ensuring plant safety, has attracted considerable attention from researchers in academic and industrial communities [1]–[4]. At present, industrial processes are usually constructed using several subprocess units. A large number of variables, complex relations between units, and process nonlinearity contribute to the difficulty in developing first-principle models of processes and prevent the use of model-based monitoring methods [5]. These complexities introduce new challenges in the field of industrial process monitoring [6]–[8]. Moreover, the abundance of process data due to the wide implementations of distributed control systems contributes to the attractiveness of data-driven monitoring [9]–[11]. A large volume of process data is collected during the long operation process history. Dealing with industrial big data, extracting meaningful features, and establishing efficient monitoring models remain challenging.

Various multivariate statistical process monitoring (MSPM) methods have been developed in the past several decades [12], [13]. MSPM methods seek to extract certain representations from normal operating data and generate the corresponding residuals. Moreover, these methods examine the status of representations and residuals by establishing monitoring statistics and thresholds. For example, principal component analysis (PCA) extracts representations that capture the largest variance in the process data and generates the data reconstruction residual. The partial least squares (PLS) method extracts the maximally related representations with the quality variables and generates the quality prediction residual. Canonical correlation analysis (CCA) extracts the most correlated representations and produces the uncorrelated residuals. These methods form the basis of the MSPM methods, and their corresponding improvements were proposed to deal with the complex process characteristics [14].

Kernel and artificial neural network (ANN) approaches have been developed to deal with process nonlinearity [15]–[17]. Kernel approaches map data into a high-dimensional space where representations are extracted. For example, kernel PCA (KPCA), kernel PLS, and kernel CCA methods have been developed [13], [18]. For plant-wide processes, variables are divided into subblocks according to process knowledge or data characteristic analysis to perform kernel learning. However, kernel methods have the following problems in dealing with industrial big data. First, kernel computing is performed between each query and reference samples. A large amount of training data increases the computational complexity. Second, the representation capability is limited for complex nonlinearity due to the fixed form of kernel functions. Third, determining appropriate parameters in the kernel function is generally difficult. ANN-based monitoring approaches have been proposed [19], [20], but conventional ANNs may fail in handling complex nonlinear processes.

In recent years, deep neural networks (DNNs) have been developed and shown effective performance in various fields, such as image, audio, and video processing [21], [22]. DNN has been recently applied to the analysis of industrial big

Manuscript received June 30, 2019; revised December 21, 2019 and March 3, 2020; accepted March 28, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61973119, Grant 61603138, and Grant 21878081, in part by the Natural Science Foundation of Shanghai under Grant 16ZR1407300, and in part by the Program of Introducing Talents of Discipline to Universities through the 111 Project under Grant B17017. *(Corresponding author: Qingchao Jiang.)*

The authors are with the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, China, and also with the Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 200092, China (e-mail: qchjiang@ecust.edu.cn; yanshifu@mail.ecust.edu.cn; huihyva@ecust.edu.cn; xfyan@ecust.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2985223

data [23]–[25]. For example, Shang *et al.* [26] analyzed the feasibility of DNN for process modeling and developed a deep belief neural network-based soft sensor for predicting the cut point of a crude distillation unit. Yuan *et al.* [27] developed a modeling method of variable weighted stacked autoencoder (SAE)-based soft sensor, which has been tested on a debutanizer column. Sun and Ge [28] proposed a probabilistic sequential method of network-based soft sensing and applied it to a CO_2 absorption column. Jiang *et al.* [29] proposed a denoising SAE-based fault detection approach for process fault detection and diagnosis. This approach considers time-series information [29] while ignoring the relationship between two sets of variables in this denoising SAE. Jiang and Yan [30] recently proposed a deep relevant representation learning-based fault detection method that generates representations of process input and output and then establishes the linear model between representations for fault detection. However, the obtained representations may be uncorrelated because the SAE learns in an unsupervised manner. This feature may lead to a poorly characterized relationship between different sets of variables. Hence, monitoring performance can be further improved.

In this article, a data-driven distributed monitoring framework for nonlinear plant-wide processes incorporating deep CCA (DCCA) and the distributed computing technique is proposed. The key contributions of this work can be summarized as follows. First, a local–global modeling framework is proposed to achieve efficient monitoring of industrial plant-wide processes. Second, a distributed modeling and computing strategy is proposed to reduce the computation cost in modeling. Third, the characteristics of the proposed approach are theoretically analyzed. The effectiveness of the proposed approach is illustrated through experimental studies on the Tennessee Eastman (TE) benchmark and laboratory-scale distillation processes.

II. Existing Techniques and Problem Formulation

A. CCA-Based Fault Detection

CCA is a classical multivariate analysis technique for extracting correlated representations [31], [32]. This technique explores and characterizes the linear correlation among variables. Let $\boldsymbol{u} \in \mathbb{R}^p$ and $\boldsymbol{y} \in \mathbb{R}^q$ be two sets of random variables. CCA finds canonical vectors (CVs) Ψ_u and Ψ_y to extract the maximally correlated representations as

$$\begin{aligned}\left(\Psi_u^*, \Psi_y^*\right) &= \arg\max_{\Psi_u, \Psi_y} \operatorname{corr}\left(\Psi_u^T u, \Psi_y^T y\right) \\ &= \arg\max_{\Psi_u, \Psi_y} \frac{\Psi_u^T \boldsymbol{\Sigma}_{uy} \Psi_y}{\left(\Psi_u^T \boldsymbol{\Sigma}_u \Psi_u\right)^{1/2} \left(\Psi_y^T \boldsymbol{\Sigma}_y \Psi_y\right)^{1/2}} \end{aligned} \tag{1}$$

where $\boldsymbol{\Sigma}_{uy}$, $\boldsymbol{\Sigma}_u$, and $\boldsymbol{\Sigma}_y$ are the covariance matrices. The matrix Θ is constructed to obtain the solutions of (1). Singular value decomposition is performed as

$$\Theta = \boldsymbol{\Sigma}_u^{-1/2} \boldsymbol{\Sigma}_{uy} \boldsymbol{\Sigma}_y^{-1/2} = \boldsymbol{R} \Sigma \boldsymbol{V}^T \tag{2}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times q}$ is a diagonal matrix. Then, CVs Ψ_u and Ψ_y are obtained as

$$\Psi_u = \boldsymbol{\Sigma}_u^{-1/2} \boldsymbol{R} \in \mathbb{R}^{p \times p} \tag{3}$$

$$\Psi_y = \boldsymbol{\Sigma}_y^{-1/2} \boldsymbol{V} \in \mathbb{R}^{q \times q}. \tag{4}$$

Two fault detection residuals for process monitoring are constructed as

$$\boldsymbol{r}_u = \Psi_u^T \boldsymbol{u} - \boldsymbol{\Sigma} \Psi_y^T \boldsymbol{y} \tag{5}$$

$$\boldsymbol{r}_y = \Psi_y^T \boldsymbol{y} - \boldsymbol{\Sigma}^T \Psi_u^T \boldsymbol{u}. \tag{6}$$

T^2 statistics are then established to examine the status of residuals as

$$T_u^2 = \boldsymbol{r}_u^T \boldsymbol{\Sigma}_{ru}^{-1} \boldsymbol{r}_u \tag{7}$$

$$T_y^2 = \boldsymbol{r}_y^T \boldsymbol{\Sigma}_{ry}^{-1} \boldsymbol{r}_y \tag{8}$$

where $\boldsymbol{\Sigma}_{ru}$ and $\boldsymbol{\Sigma}_{ry}$ denote the covariance matrices. T_u^2 examines the status of $\boldsymbol{u}$ by considering correlation information from $\boldsymbol{y}$, whereas T_y^2 examines the status of $\boldsymbol{y}$ by considering correlation information from $\boldsymbol{u}$. Hence, the T^2 statistics are proven optimal for detecting faults that affect only $\boldsymbol{u}$ or $\boldsymbol{y}$ [32], [33].

B. Problem Formulation and Motivation

Jiang *et al.* [32] extended the CCA-based fault detection to the distributed monitoring of plant-wide processes. A monitor is established for each local unit with correlation information from the considered neighboring units. The genetic algorithm (GA) is used to determine the communication variables to reduce communication costs and remove the influence of unrelated variables. The distributed monitoring scheme is more efficient in detecting a local fault (a fault that affects only a local part of a process) than the centralized monitoring scheme. However, the CCA and distributed CCA methods are only suitable for linear processes and may fail in monitoring nonlinear processes.

As an extension of CCA to the nonlinear form, DCCA has attracted considerable attention from researchers in various fields [34]. DCCA is constructed using two SAEs and a top layer CCA neural network. Let the inputs of SAEs be $\boldsymbol{u}$ and $\boldsymbol{y}$, and the outputs of SAEs be $f_u(\boldsymbol{u})$ and $f_y(\boldsymbol{y})$. DCCA seeks to optimize two SAEs such that $f_u(\boldsymbol{u})$ and $f_y(\boldsymbol{y})$ are correlated and maximized, that is

$$\left(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*\right) = \arg\max_{(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \operatorname{corr}\left(f_u(\boldsymbol{u}; \boldsymbol{\theta}_1),\, f_y(\boldsymbol{y}; \boldsymbol{\theta}_2)\right) \tag{9}$$

where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ represent the parameters, such as the weights and biases, which are optimized during the training process. An illustration of the DCCA neural network is provided in Fig. 1. The training process of DCCA includes two procedures, namely, the self-training of each SAE and the fine-tuning of the two SAEs by adding a CCA top layer. Additional details on the DCCA basics are provided in [34]. The DCCA offers a new approach in dealing with nonlinear process data [35]. However, the means of using DCCA for distributed process monitoring have not been discussed. The following issues in monitoring a nonlinear plant-wide process must be addressed: 1) characterization of the complex nonlinear relationship within a local unit and the relationship between

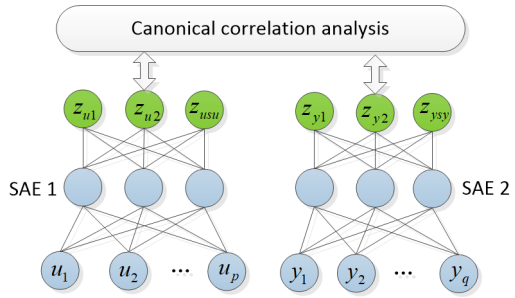


Fig. 1. DCCA neural network.

different units, 2) realization of an efficient global monitoring of the entire process, 3) identification of the fault location once a fault is detected, and 4) reduction of computation cost to satisfy the practical application requirement.

III. Distributed Modeling and Computing Framework

The proposed local–global modeling and distributed computing framework are comprehensively discussed in this section. Characteristics of the framework are also summarized.

A. Local Unit Inner Monitor

Sensors that measure the physical quantities are found in a local unit. Assume that the measured variables in the bth unit ($b = 1, 2, \ldots, B$, where B is the number of operation units) are denoted as $\boldsymbol{u}_b = [u_{b,1}, u_{b,2}, \ldots, u_{b,pb}]^T$. An SAE-based DNN is established to characterize the complex variable relationships in the local unit. This article focuses more on establishing the inner monitor for the local unit rather than elaborating on the SAE details. Let the SAE-extracted representations be $\boldsymbol{h}_b = [h_{b,1}, h_{b,2}, \ldots, h_{b,sb}]^T$. These representations are expected to contain the most important information for reconstructing the measured variable $\boldsymbol{u}_b$. A full-layer neural network is then added to the SAE to reconstruct the original measured variable $\boldsymbol{u}_b$. Let the reconstructed variables be $\hat{\boldsymbol{u}}_b = [\hat{u}_{b,1}, \hat{u}_{b,2}, \ldots, \hat{u}_{b,pb}]^T$. The following statistics are constructed to examine the status of the local unit

$$T_{hb}^2 = \boldsymbol{h}_b^T \boldsymbol{\Sigma}_{hb}^{-1} \boldsymbol{h}_b \tag{10}$$

$$Q_b = \boldsymbol{r}_b^T \boldsymbol{r}_b = (\boldsymbol{u}_b - \hat{\boldsymbol{u}}_b)^T (\boldsymbol{u}_b - \hat{\boldsymbol{u}}_b). \tag{11}$$

The thresholds for the statistics should be determined after establishing the monitoring statistics. Kernel density estimation (KDE) is used to determine the thresholds because no prior information on the data distribution is available and the nonlinear process data generally do not follow a multivariate Gaussian distribution. This article does not elaborate on the determination of the KDE threshold, but the details can be found in [7].

The Q_b statistic indicates the deviations between measured and model-reconstructed variables. The contribution of the ith variable in the bth unit is calculated as

$$\text{CONT}_{b,i} = |u_{b,i} - \hat{u}_{b,i}|. \tag{12}$$

A variable with a large deviation increasingly contributes to the fault.

Remark 1 (Characteristic of the Inner Monitor for the Local Unit): Considering that SAE is regarded as the generalization of PCA to the nonlinear case, the SAE-based local unit monitoring can be considered the generalization of PCA-based monitoring. The T_{hb}^2 statistic examines the dominant part of the local unit, whereas Q_b evaluates the residual part of the local unit. Both statistics are similar to those of PCA monitoring. These statistics monitor the status of local units with information from only the considered local units while ignoring the relationship with other units. Considering the correlation with variables outside the units is important. Thus, exploring the relationships between units and establishing the outer-related monitors of local units are important.

B. Outer-Related Monitor for the Local Unit

The relationships between variables are complex in the nonlinear process. Determining which variables should be involved in modeling must be performed first to characterize the relationship between a unit and the coupled units. The selection of inappropriate variables will either discard the relevant variable relationships or disturb the information of unrelated variables. The correlation coefficient is a conventional method for evaluating relationships between variables. However, this coefficient is limited in the linear correlation case and generally fails in the nonlinear situation. The neighborhood variable determined using mutual information (MI), which considers the linear and nonlinear relationships between variables, is performed to eliminate the unrelated ones.

MI examines the dependence of two variables from the entropy aspect. The MI between x and y is calculated as

$$I(x, y) = \iint_{x,y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) dx dy \tag{13}$$

where $p(x, y)$ is the joint density, and $p(x)$ and $p(y)$ are the marginal densities. If a variable from other units is independent of variables of the local unit, then including the variables in the model is unnecessary. The MI-based neighborhood variable determination for a local unit is summarized in **Algorithm 1**.

Algorithm 1 Determination of MI-Based Neighborhood Variable

Step 1: Construct the reference variable x_r, which is independent of all considered variables.

Step 2: For the ith variable in the bth local unit, MI is calculated between $u_{b,i}$ and the reference variable x_r to obtain the value $MI_{b,i}$.

Step 3: MI between $u_{b,i}$ and all variables from other units are calculated.

Step 4: Variables with MI less than $\delta \times MI_{b,i}$ (δ is used to control the number of eliminated variables and is suggested to be between 1.1 and 1.5) are regarded as independent of $u_{b,i}$, and these variables are eliminated.

Repeat Steps 2–4 to determine the related variables for the bth local unit.

Let the selected neighborhood variables for the bth local unit be $\boldsymbol{y}_b = [y_{b,1}, y_{b,2}, \ldots, y_{b,qb}]^T$. DCCA is

performed between $\boldsymbol{u}_b$ and $\boldsymbol{y}_b$ to find the transformations of the original variables using $f_u(\boldsymbol{u}_b)$ and $f_y(\boldsymbol{y}_b)$. The correlated representations for the local units are extracted as $\boldsymbol{z}_{ub} = [f_{ub,1}(\boldsymbol{u}_b), f_{ub,2}(\boldsymbol{u}_b), \ldots, f_{ub,sb}(\boldsymbol{u}_b)]^T = [z_{ub,1}, z_{ub,2}, \ldots, z_{ub,sb}]^T$. The extracted correlated representations for variables from the neighboring units are denoted as $\boldsymbol{z}_{yb} = [f_{yb,1}(\boldsymbol{y}_b), f_{yb,2}(\boldsymbol{y}_b), \ldots, f_{yb,sb}(\boldsymbol{y}_b)]^T = [z_{yb,1}, z_{yb,2}, \ldots, z_{yb,sb}]^T$. Then, CCA is performed between $\boldsymbol{z}_{ub}$ and $\boldsymbol{z}_{yb}$. Let the $\boldsymbol{\Sigma}_{zub}$, $\boldsymbol{\Sigma}_{zyb}$ and $\boldsymbol{\Sigma}_{zuyb}$ be the covariance matrices of $\boldsymbol{z}_{ub}$ and $\boldsymbol{z}_{yb}$. The canonical correlation vectors Ψ_{ub} and Ψ_{yb}, intermediate matrix $\boldsymbol{V}_b$ and $\boldsymbol{R}_b$, and the correlation matrix $\boldsymbol{\Sigma}_{uyb}$ are obtained. The fault detection residuals are generated for the bth local unit as

$$\boldsymbol{r}_{ub} = \Psi_{ub}^T \boldsymbol{z}_{ub} - \boldsymbol{\Sigma}_{uyb} \Psi_{yb}^T \boldsymbol{z}_{yb}. \tag{14}$$

The T^2 statistic for examining the residual is then established as

$$T_{ub}^2 = \boldsymbol{r}_{ub}^T \boldsymbol{\Sigma}_{rub}^{-1} \boldsymbol{r}_{ub} \tag{15}$$

where $\boldsymbol{\Sigma}_{rub} = \boldsymbol{I}_{ub} - \boldsymbol{\Sigma}_{uyb} \boldsymbol{\Sigma}_{uyb}^T$, and $\boldsymbol{I}_{ub}$ denotes the identity matrix with rank s_b. T_{ub}^2 detects the fault occurrence in the bth local unit by considering correlation information from the neighboring units.

Remark 2 (Characteristic of the Local Unit Outer-Related Monitor): CCA is used for local unit monitoring with correlation information from the considered neighborhood variables. Once the relation between $\boldsymbol{u}_b$ and $\boldsymbol{y}_b$ can be expressed as $\boldsymbol{A}(\boldsymbol{u}_b + \boldsymbol{\varepsilon}) = \boldsymbol{B}\boldsymbol{y}_b$, the CCA-based local unit monitoring obtains the optimal performance. Given that DCCA is a generalization of CCA in the nonlinear form, the feasibility of using the T_{ub}^2 statistic for fault detection can be obtained when the relationship between $\boldsymbol{u}_b$ and $\boldsymbol{y}_b$ is generalized as $\boldsymbol{A}(f_u(\boldsymbol{u}_b) + \boldsymbol{\varepsilon}) = \boldsymbol{B} f_y(\boldsymbol{y}_b)$.

Proof: According to the CCA transformation, $\Psi_{yb} = \boldsymbol{\Sigma}_{zyb}^{-1} \boldsymbol{V}_b$ and $\boldsymbol{\Sigma}_{zub}^{-1/2} \boldsymbol{\Sigma}_{zuyb} \boldsymbol{\Sigma}_{zyb}^{-1/2} = \boldsymbol{R}_b \boldsymbol{\Sigma}_{uyb} \boldsymbol{V}_b^T$; thus, $\sum_{uyb} \boldsymbol{V}_b^T = \boldsymbol{R}_b^T \sum_{zub}^{-1/2} \boldsymbol{\Sigma}_{zuyb} \sum_{zyb}^{-1/2}$ and $\sum_{uyb} \boldsymbol{\Psi}_{zyb}^T \boldsymbol{z}_{yb} = \sum_{uyb} \boldsymbol{V}_b^T \sum_{zyb}^{-1/2} \boldsymbol{z}_{yb} = \boldsymbol{R}_b^T \sum_{zub}^{-1/2} \boldsymbol{\Sigma}_{zuyb} \sum_{zyb}^{-1} \boldsymbol{z}_{yb}$, respectively. Hence

$$\begin{aligned} \boldsymbol{r}_{ub} &= \Psi_{ub}^T \boldsymbol{z}_{ub} - \boldsymbol{\Sigma}_{uyb} \Psi_{yb}^T \boldsymbol{z}_{yb} \\ &= \boldsymbol{R}_b^T \sum\nolimits_{zub}^{-1/2} \left(\boldsymbol{z}_{ub} - \boldsymbol{\Sigma}_{zuyb} \boldsymbol{\Sigma}_{zyb}^{-1} \boldsymbol{z}_{yb} \right) \end{aligned} \tag{16}$$

where $\hat{\boldsymbol{z}}_{ub} = \boldsymbol{\Sigma}_{zuyb} \boldsymbol{\Sigma}_{zyb}^{-1} \boldsymbol{z}_{yb}$ is the least square estimation of $\hat{\boldsymbol{z}}_{ub}$ using $\boldsymbol{z}_{yb}$. The residual obtains the minimal covariance, and T_{ub}^2 is an optimal statistic for monitoring the local fault affecting the variables in the bth unit.

Together with T_{hb}^2 and Q_b, the three statistics effectively monitor the local unit. The inner and outer-related monitors for local unit construct the monitoring matrix, which helps identify the process status and fault location.

C. Global and Comprehensive Local Unit Monitor

After the relationships between variables within the local unit and those between different units are well characterized, global monitoring of all units is important in the plant-wide process. The outer-related representations for each local unit are obtained through DCCA modeling. According to the objective of DCCA, these outer-related representations contain all the relationships between a local unit and the other units. Let the outer-related representations for the bth unit be $\boldsymbol{z}_{ub} (b = 1, 2, \ldots, B)$. A joint representation vector $\boldsymbol{z} = [\boldsymbol{z}_{u1}^T, \boldsymbol{z}_{u2}^T, \ldots, \boldsymbol{z}_{uB}^T]^T$ can be constructed. This joint representation characterizes the relation between all local units, and then the T^2 statistic for the joint representation vector $\boldsymbol{z}$ is constructed as

$$T_z^2 = \boldsymbol{z}^T \boldsymbol{\Sigma}_z^{-1} \boldsymbol{z} \tag{17}$$

where $\boldsymbol{\Sigma}_z$ denotes the covariance matrix. T_z^2 examines the status of the outer-related representations from all local units and monitors the relationships between units. Therefore, T_z^2 is a global monitoring statistic that governs the relationships throughout the entire process. Although the distributions of the representations are generally unavailable, the threshold of T_z^2 can also be determined through KDE.

A Bayesian inference-based statistic fusion strategy is used to evaluate the status and the results from all local unit monitors comprehensively. Let J and J_{th} be the monitoring statistic and the threshold of the statistic, respectively. Then, J can denote the T_{hb}^2, Q_b, and T_{ub}^2 values. Assume that the S $(S = 3B)$ statistics are available and denoted as $J_1, \ldots, J_S$. Then, the comprehensive index J_{comp} can be calculated using Bayesian inference as

$$J_{\text{comp}} = \sum_{s=1}^{S} \left\{ \frac{P(J_s|F) P(F|J_s)}{\sum_{j=1}^{S} P\left(J_j|F\right)} \right\} \tag{18}$$

where

$$P(F|J_s) = \frac{P(J_s|F) P(F)}{P(J_s)} \tag{19}$$

$$P(J_s) = P(J_s|N) P(N) + P(J_s|F) P(F) \tag{20}$$

$$P(J_s|N) = e^{-J_{s,\text{new}}/J_{s,th}},\ P(J_s|F) = e^{-J_{s,th}/J_{s,\text{new}}} \tag{21}$$

where "N" and "F" represent normal and faulty conditions, respectively. Given a confidence level α, the threshold of the J_{comp} index is determined as $1 - \alpha$. Additional details on the index are presented in [36].

Remark 3 (Characteristic of the Global and Comprehensive Local Unit Monitor): T_z^2 is constructed by gathering outer-related representations from all units. Given that these representations are extracted using DCCA, a correlation generally exists among the representations, in which the T^2 test is optimal for fault detection. T_z^2 is a global monitoring statistic governing the relationship among units throughout the entire process. Once the control limit of the statistic is exceeded, the correlation between units is destroyed, and the fault will possibly affect several operation units. The J_{comp} is a statistic that provides a comprehensive evaluation of results from all the local units. At least one local unit is affected by the fault once the control limit of the statistic is exceeded.

D. Distributed Computing

The computational complexity should be considered when dealing with industrial big data. The modeling process in this article is performed in a distributed computing manner to reduce the computational time and meet the practical

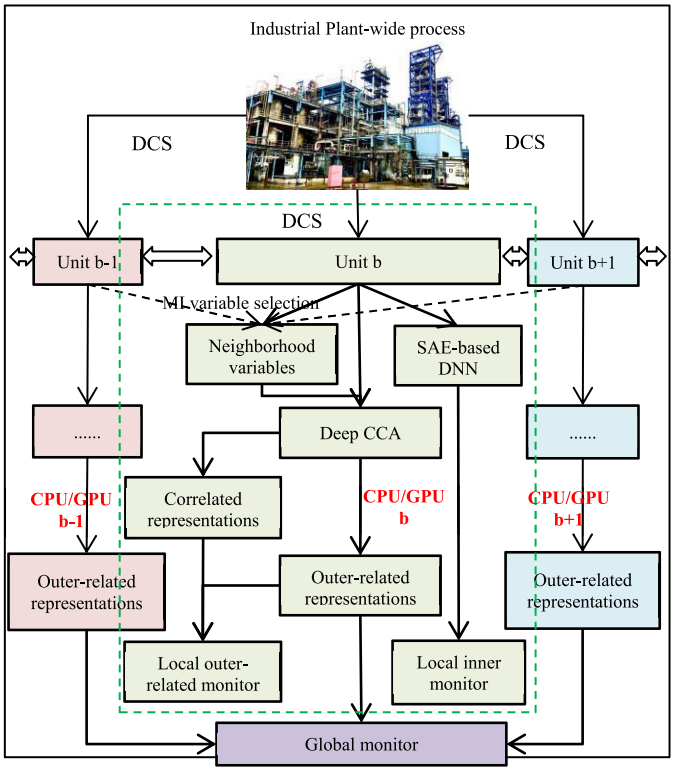


Fig. 2. Distributed modeling and computing framework.

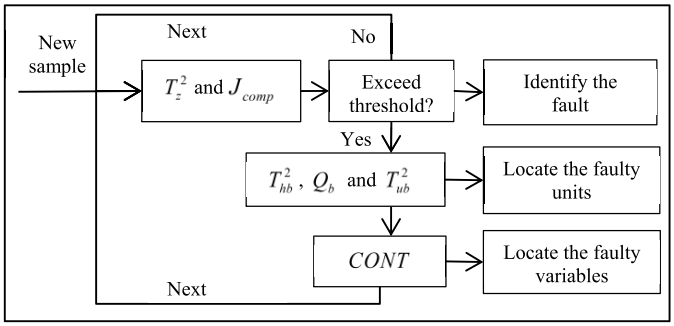


Fig. 3. Flowchart for the process status identification.

application requirements. Given that the proposed distributed modeling framework for the plant-wide process with B number of local units incorporates only the local process information for the modeling of each unit, the key computing task can be divided and assigned to the B number of CPUs or GPUs. Each CPU or GPU trains the DNN for a local unit to reduce the computation time significantly.

The local–global modeling and distributed computing framework is summarized as **Algorithm 2**, and the framework is illustrated in Fig. 2.

Remark 4 (Structure Optimization of DNNs): Notably, the structure optimization of a DNN remains unclear. In the proposed local–global modeling framework, two DNNs, namely, SAE and DCCA, are involved. The structure of an SAE can be determined via the cross-validation method, which minimizes the data reconstruction error [21]. The structure of DCCA can be determined to maximize the correlation between the two sets of correlated representations [34]. An appropriate structure for a specific practical application problem is generally obtained to meet the practical application requirement.

Algorithm 2 Distributed Modeling and Computing Framework for Nonlinear Process Monitoring

Offline modeling

Step 1: Determine the inner unit variables of each unit and the neighborhood variables from other units using MI-based variable selection.

Step 2: Establish the local unit monitors.

2.1 Train an SAE for the bth unit using the bth CPU;

2.2 Add a full-layer neural network to the SAE to reconstruct the inner unit variables of the bth unit;

2.2 Calculate the T_{hb}^2 and Q_b statistics;

2.3 Determine the threshold $T_{hb,th}^2$ and $Q_{b,th}$;

2.4 Perform DCCA between the bth unit and neighborhood variables and obtain the correlated representations using the bth CPU;

2.5 Calculate the T_{ub}^2 statistics;

2.6 Calculate the threshold $T_{ub,th}^2$ for T_{ub}^2;

Step 3: Establish the global monitoring statistic.

3.1 Collect the obtained outer-related representations from all local units as z;

3.2 Calculate the T_z^2 statistic;

3.3 Determine the threshold $T_{z,th}^2$ for the T_z^2.

Step 4: Construct the comprehensive monitoring statistic of local unit J_{comp} and determine the threshold.

Online monitoring

Once a new sample is obtained, the following three-step procedure is performed to identify the process status and fault location.

Step 1: Examine the global and comprehensive local unit monitoring statistics.

If T_z^2 exceeds the control limit, then the fault is a global fault that generally affects several operation units;

If J_{comp} exceeds the control limit, then a fault that affects some local units exists.

Step 2: Examine the monitoring matrix of local unit. Once T_z^2 or J_{comp} exceeds the control limit, analyze the monitoring statistics of local unit to localize the fault unit.

If an outer-related monitor for the local unit with statistic T_{ub}^2 exceeds the control limit, then a fault in the local unit affecting the neighboring units exists;

If only an inner monitor for the local unit indicates a fault, then the fault is a local fault affecting the local unit only.

Step 3: Examine the variable contribution plot to locate the faulty variables. Once a fault is detected, calculate the contribution of variables to the fault.

The three-step process status identification approach is shown in Fig. 3.

IV. Experimental Studies

A general workstation with eight CPUs is used in this article. The computations are performed through the Python software with the hardware Intel (R) Core I7-5500U at 2.4 GHz 16.00 G RAM.

TABLE I
OPERATION UNITS AND VARIABLES IN TEP

Sub-unit	Variable name	Marker
Input unit	A	1
	D	2
	E	3
	A and C	4
	D flow	42
	A flow	43
	E flow	44
	A and C flow	45
Reactor	Feed rate	6
	Pressure	7
	Level	8
	Temp.	9
	Water temp.	21
	Cooling water flow	51
	Condenser cooling water flow	52
Separator	Temp.	11
	Level	12
	Pressure	13
	Underflow	14
	Water temp.	22
	Pot liquid flow	48
Stripper	Level	15
	Pressure	16
	Underflow	17
	Temp.	18
	Steam flow	19
	Liquid product flow	49
	Steam valve	50
Compressor	Recycle flow	5
	Purge rate	10
	Work	20
	Recycle valve	46
	Purge valve	47

A. TE Benchmark Process

The TE process (TEP) is created by simulating a real industrial process [37]. The process comprises five typical operation units as the input feed, reactor, separator, stripper, and compressor, as shown in Fig. 4. The corresponding variables are presented in Table I. The details of the TEP are not provided in this article [36], [38]. First, a set of 10 000 samples is generated through the simulator to test the computation efficiency. A workstation with eight CPUs is used, and the modeling is completed within 150 s. The computation speed meets the practical application requirement. However, if a KPCA model is established using the training data then the modeling time is more than 5600 s while the online computing time is approximately 52 s for one sample. The computing time is unsuitable for process monitoring with fast sampling time. The benchmark data found in http://web.mit.edu/braatzgroup/links.html are used to test the monitoring performance. The neighborhood variables from the other units are selected using MI-based variable selection. Then, the monitoring model is established on the basis of the inner and neighborhood variables for each operation unit. Five CPUs are used in this article to establish the monitoring model. In the training of SAEs, the model activations are "exponential linear unit (ELU)" in the hidden layers and "linear" in the output layer. A total of 2000 epochs are set with the early stopping strategy during the training phase. The batch size is 50 samples, and the optimizer is "Adam." In DCCA neural networks, the model activations are "ELU" in the hidden layers and "linear" in the output layer. A total of 2000 epochs are set with the early stopping strategy during the training phase. The batch size is 50 samples, and the optimizer is "Adam." The entire modeling is completed within 30 s. The threshold of each statistic is determined through KDE, which controls the false alarm rate under 2.5%.

The fault detection rates (FDRs) of the proposed distributed monitoring approach, global SAE, and global KPCA monitoring method are listed in Table II. Distributed monitoring competitively provides the best monitoring results for most of the faults (13 out of 21 faults). The monitoring performance of SAE for the fault affecting several operation units is satisfactory because the global SAE establishes a centralized model. However, the model of SAE-based global monitoring does not provide information on the operation status of the local unit. Thus, when a local fault occurs, the global SAE model may function poorly. Under this condition, the proposed distributed monitoring method will obtain an improved monitoring performance. The monitoring results for faults 4 and 5 are comprehensively analyzed in this article.

Fault 4 is a local fault affecting the inlet temperature of reactor cooling water, which causes a step change in the flow rate of reactor cooling water (variable 51). The other variables are slightly affected. The global and local monitoring statistics for the fault are, respectively, presented in Fig. 5(a) and (b). Fig. 5 shows that the global monitoring statistics are rarely affected, whereas the local comprehensive monitoring statistics are influenced. This finding indicates that the fault is possibly a local fault. Then, the matrix of local monitoring statistics is investigated as shown in Fig. 6. The Q_2 statistic is mostly influenced and continues to indicate the faults. The results show that the second local unit, that is, the reactor unit, is affected by the fault. The other units are seldom influenced because their corresponding statistics are mostly within the control limits. These results indicate that the fault is a local fault in only the reactor unit. The fault is unrelated to the other units. Finally, the contribution plot at the 161st point (the beginning of the fault) is examined, as shown in Fig. 7. The results indicate that the variables in the reactor provide the most contribution. The reactor temperature and cooling water flow obtain the largest contributions, and these findings are consistent with the real situation.

Fault 5 affects the inlet temperature of condenser cooling water. Several variables are affected at the beginning of the fault, and the fault effect is significant. However, the fault influence is removed in most variables after the 500th point, except the condenser cooling water flow, due to the existence

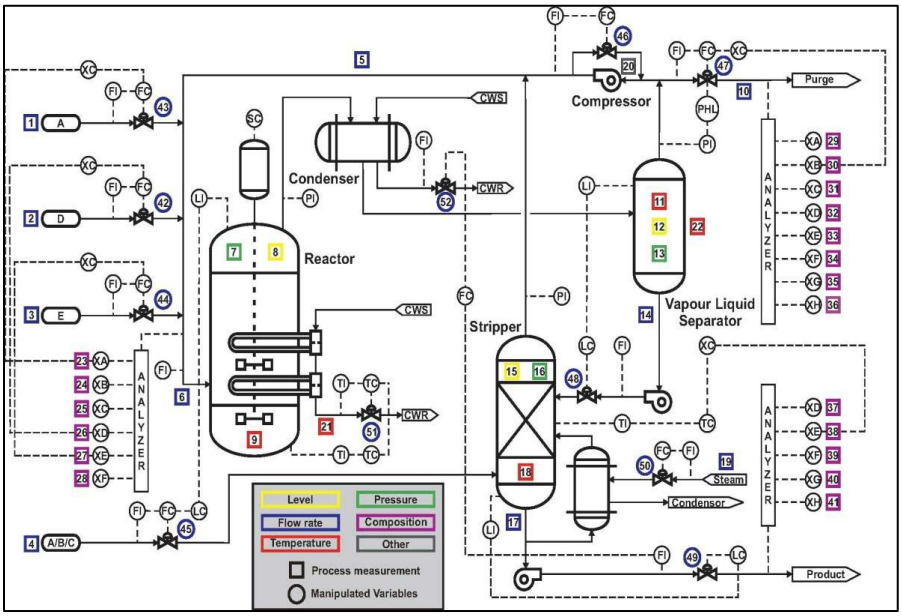


Fig. 4. Schematic of TEP [39].

TABLE II
MONITORING RESULTS FOR THE TEP (FDR)

Unit	Input feed			Reactor			Separator			Stripper			Compressor			Global	Compre hensive	Global SAE		Global KPCA	
Fault \Stat.	T^2_{h1}	Q_1	T^2_{u1}	T^2_{h2}	Q_2	T^2_{u2}	T^2_{h3}	Q_3	T^2_{u3}	T^2_{h4}	Q_4	T^2_{u4}	T^2_{h5}	Q_5	T^2_{u5}	T^2_z	J_{comp}	T^2	Q	T^2	Q
1	0.98	0.99	0.99	0.32	0.35	0.96	0.39	0.32	0.78	1.00	0.99	1.00	0.50	0.40	0.56	1.00	1	1	1	1	1
2	0.07	0.95	0.95	0.15	0.69	0.99	0.95	0.94	0.99	0.96	0.96	1.00	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	1
3	0.06	0.05	0.06	0.10	0.07	0.06	0.08	0.06	0.08	0.19	0.12	0.21	0.10	0.06	0.09	0.26	0.15	0.12	0.28	0.09	0.12
4	0.06	0.04	0.14	0.44	1.00	0.04	0.08	0.08	0.05	0.13	0.10	0.29	0.04	0.07	0.20	0.33	1	1	1	0.79	1
5	0.25	0.17	0.22	0.40	0.23	0.49	0.22	0.20	0.27	0.33	0.26	0.45	0.30	0.25	0.26	0.95	0.47	0.7	1	0.3	0.51
6	0.99	1.00	1.00	0.99	0.99	1.00	0.99	0.98	0.99	0.99	0.98	0.99	0.22	0.99	1.00	1.00	1	1	1	0.13	1
7	1.00	1.00	1.00	0.29	0.34	0.51	0.35	0.32	0.39	0.53	0.43	0.55	0.37	0.42	0.43	1.00	1	1	1	1	1
8	0.84	0.66	0.90	0.74	0.85	0.89	0.92	0.85	0.94	0.97	0.94	0.93	0.92	0.91	0.95	0.99	0.99	0.99	0.99	0.98	0.99
9	0.08	0.03	0.05	0.08	0.07	0.06	0.09	0.05	0.07	0.12	0.07	0.15	0.11	0.07	0.08	0.26	0.13	0.11	0.24	0.08	0.1
10	0.28	0.12	0.18	0.22	0.25	0.27	0.27	0.21	0.26	0.89	0.81	0.59	0.30	0.31	0.33	0.71	0.87	0.50	0.92	0.52	0.82
11	0.06	0.04	0.21	0.55	0.85	0.39	0.11	0.07	0.08	0.24	0.14	0.36	0.05	0.08	0.18	0.58	0.82	0.86	0.84	0.65	0.85
12	0.85	0.69	0.85	0.79	0.90	0.91	0.98	0.95	0.93	0.98	0.97	0.97	0.78	0.82	0.84	0.99	1.00	0.99	1	0.99	0.99
13	0.76	0.68	0.87	0.75	0.86	0.95	0.88	0.81	0.94	0.95	0.94	0.95	0.81	0.80	0.94	0.97	0.95	0.96	0.96	0.95	0.96
14	0.04	0.03	0.73	1.00	1.00	0.84	0.03	0.01	0.65	0.04	0.02	0.90	0.04	0.04	0.36	0.90	1	1	1	1	1
15	0.07	0.04	0.05	0.10	0.10	0.09	0.13	0.09	0.12	0.25	0.21	0.22	0.07	0.08	0.11	0.31	0.25	0.21	0.30	0.11	0.21
16	0.18	0.07	0.09	0.15	0.14	0.23	0.17	0.14	0.24	0.65	0.71	0.51	0.20	0.18	0.25	0.67	0.81	0.36	0.96	0.37	0.78
17	0.08	0.05	0.68	0.88	0.97	0.84	0.25	0.17	0.71	0.41	0.22	0.79	0.14	0.11	0.64	0.93	0.96	0.97	0.96	0.84	0.98
18	0.89	0.88	0.89	0.86	0.89	0.87	0.91	0.90	0.91	0.91	0.89	0.90	0.85	0.88	0.88	0.93	0.91	0.92	0.92	0.19	0.91
19	0.04	0.05	0.06	0.12	0.08	0.11	0.04	0.02	0.22	0.10	0.11	0.18	0.20	0.75	0.38	0.61	0.70	0.75	0.92	0.24	0.65
20	0.12	0.06	0.12	0.16	0.16	0.45	0.56	0.36	0.43	0.48	0.36	0.53	0.60	0.66	0.51	0.77	0.73	0.75	0.86	0.58	0.75
21	0.01	0.00	0.01	0.13	0.50	0.56	0.42	0.22	0.67	0.69	0.43	0.61	0.17	0.10	0.28	0.67	0.61	0.63	0.58	0.49	0.66

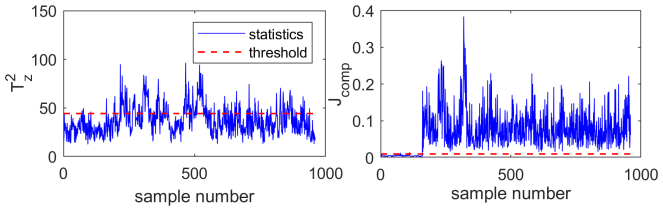


Fig. 5. Global and comprehensive local monitoring statistics for the TEP fault 4.

of a control loop. The global and local monitoring statistics for fault 5 are, respectively, provided in Fig. 8(a) and (b). The global monitoring statistics continue to indicate the fault, whereas the local monitoring statistics detect the fault at the beginning and return to normal after the 500th point. This phenomenon indicates that the status of the local units returns to normal because of the compensation of the control loop. However, the correlation between different units has changed, and the change cannot be compensated.

The matrices of local unit monitoring are presented in Fig. 9. The fault is detected at the beginning of almost all the units. Furthermore, the statuses of almost all units, except in the reactor unit after the 500th point, are normal. This normal status is attributed to the contained variable of the condenser cooling water flow in the reactor unit. The outer-related monitor of the

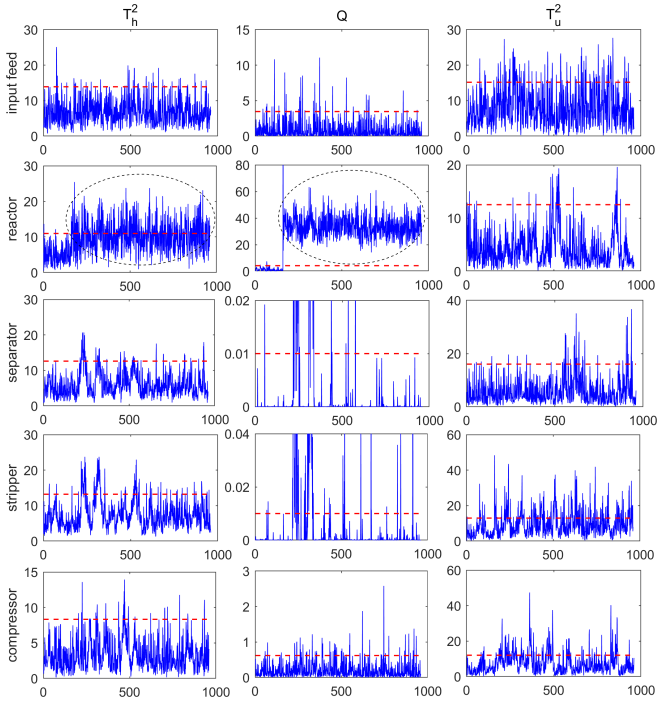


Fig. 6. Monitoring of the local units for the TEP fault 4.

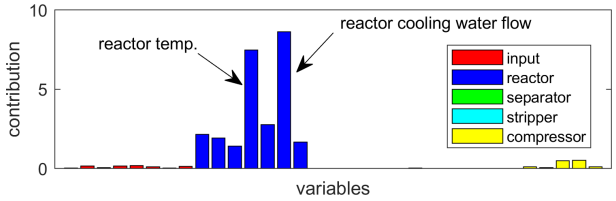


Fig. 7. Variable contribution plots for the TEP fault 4.

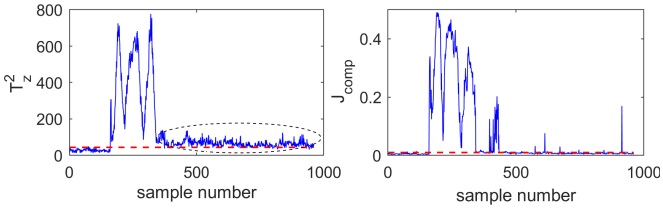


Fig. 8. Global and comprehensive local monitoring statistics for the TEP fault 5.

local unit detects the fault. Finally, the contribution plots in Figs. 10 and 11 are analyzed. Fig. 10 demonstrates the contribution plots at the 161st point (the beginning of the fault). The variables in the reactor and separator units provide the most contribution to the fault. Fig. 11 illustrates the contribution plots at the 600th point (after the fault is compensated). The contributions of all variables are low, and the variables in the reactor provide relatively higher contributions than those of the other units. This result indicates that the reactor unit is slightly affected by the fault, which is consistent with the real condition. From the results presented in Table II and the analysis above, the superiority of the proposed distributed monitoring scheme is demonstrated.

B. Glycerol Distillation Process

Glycerol is a widely used chemical material, and its distillation unit is important in glycerol production. The experimental equipment of the glycerol distillation process is built.

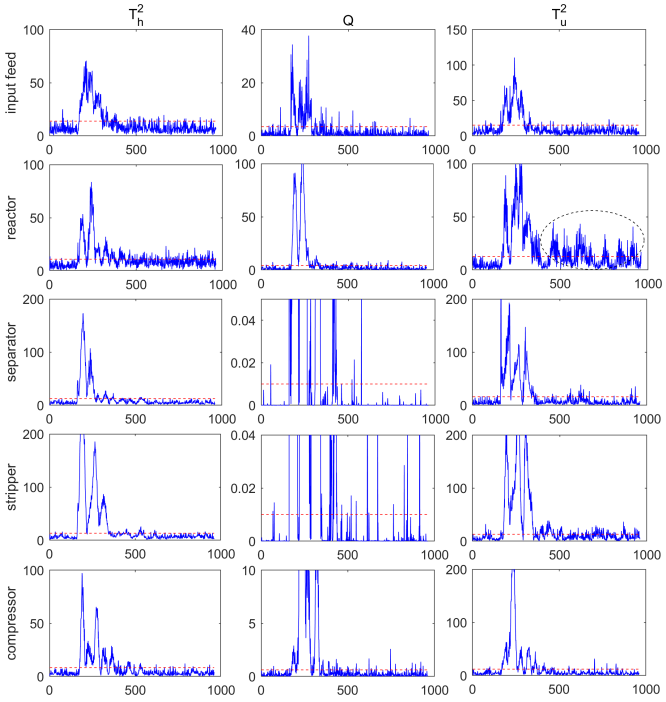


Fig. 9. Monitoring of the local units for the TEP fault 5.

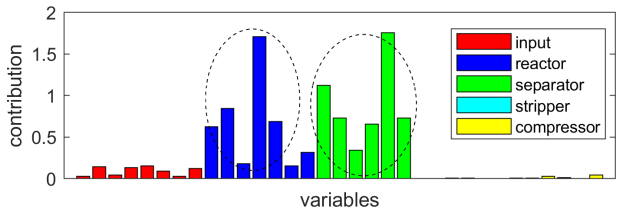


Fig. 10. Variable contribution plots for the TEP fault 5 at the beginning of the fault.

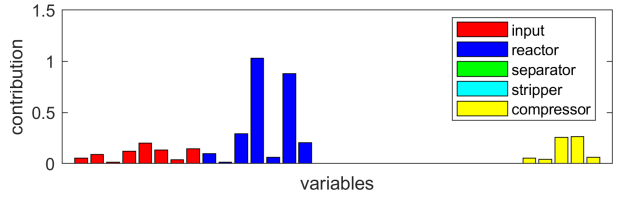
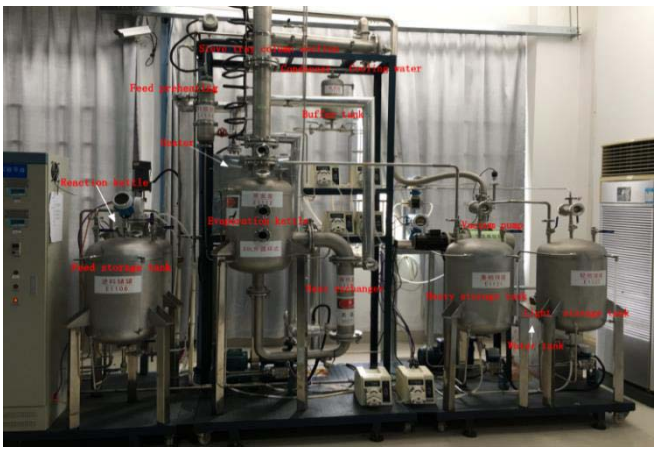
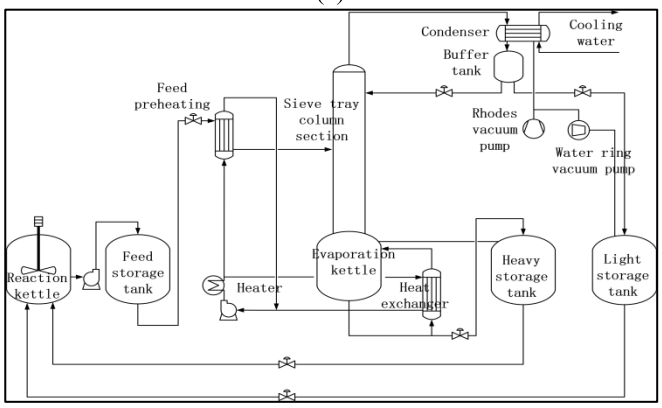


Fig. 11. Variable contribution plots for the TEP fault 5 after compensation.

This equipment comprises three typical units: the feed, evaporation, and product units. The physical figure and the simplified schematic of the distillation process are illustrated in Fig. 12(a) and (b). The feed liquid is a mixture of water and glycerol, which enters the preheating tank from the top of the tower. The lower end of the tower equipment is connected to the evaporator. The oil, which is used as the heat transfer medium to meet the heat demand of the process, is heated in a heater and then sent to the distillation column and the preheating tank through the circulating pump. The working pressure of the oil heating system of liquid-phase heat transfer is determined using the circulating pump. Therefore, the quality and type of circulating pump directly affect the heat transfer system. The evaporator heats the liquid to produce steam, which rises along the tray and comes into contact with the countercurrent of falling liquid. Heavy-phase components



(a)



(b)

Fig. 12. Laboratory-scale glycerol distillation process. (a) Physical figure. (b) Simplified schematic.

TABLE III
CONSIDERED VARIABLES IN THE DISTILLATION PROCESS

Input unit	Name	Output unit	Name
1	Feed flow rate	1	Feed storage tank level
2	Sensitive plate temp.	2-13	Tower plate temp. 1-12
3	Tower bottom liquid level	14	Condenser cooling water flow rate
4	Tower top reflux	15	Heave storage tank level
5	Overhead product flow	16	Light storage tank level

flow out of the reactor. The condenser at the top of the tower is connected to the tower body through a pipeline. The top distillate passes using the condenser and enters the buffer tank. Some distillates flow into the light component tank as light components, and the rest flows back to the distillation column for distillation and purification. The measured variables in each unit are listed in Table III.

A total of 2000 samples are collected as the training data during normal operation. In the training of SAEs, the activations of the models are "rectified linear unit (ReLU)" and "linear" in the hidden and output layers, respectively. A total of 2000 epochs are set with the early stopping strategy during the training phase. The batch size is 100 samples, and the optimizer is "Adam." In the DCCA neural networks, the activations of the models are "ReLU" and "linear" in the hidden and output layers, respectively. A total of 2000 epochs are set with the early stopping strategy during the training phase. The batch size is 100 samples, and the optimizer is "Adam." Two CPUs are involved in the computation, and the entire modeling process is completed within 30 s. However, if a KPCA model is established by using the training data, then the modeling time is more than 60 s. The following two faults are introduced after the 130th sample to test the monitoring performance of the distributed monitoring approach: fault 1 (a step change in the feed flow rate) and fault 2 (a slow drift in the temperature sensor 4).

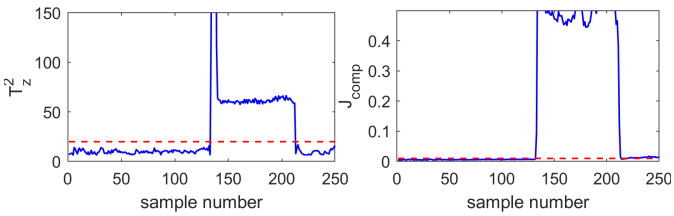


Fig. 13. Global and comprehensive local monitoring statistics for the distillation process fault 1.

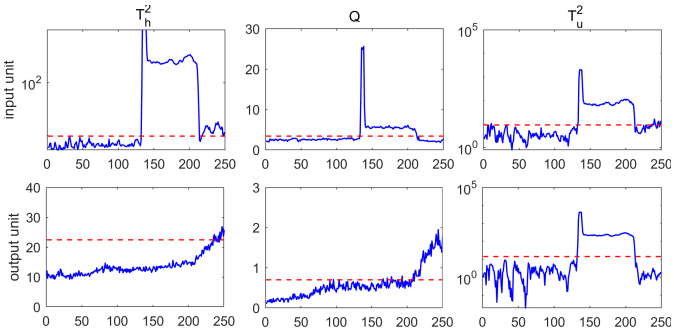


Fig. 14. Monitoring of the local units for the distillation process fault 1.

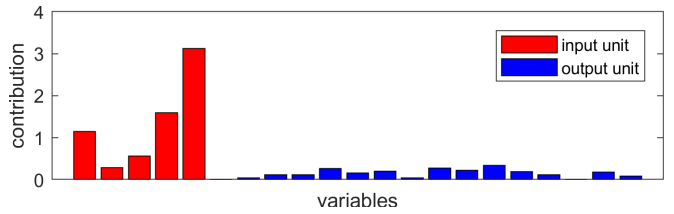


Fig. 15. Variable contribution plots of the distillation process fault 1.

The global and local monitoring statistics for fault 1 are presented in Fig. 13. The fault detection of global and local monitoring statistics indicates that the fault affects the global and local parts of the process. The local unit monitoring matrix shown in Fig. 14 illustrates that the inner part of the input unit and outer-related part of the output unit are significantly affected. The inner part of the output unit is unaffected. Then, the variable contribution plot shown in Fig. 15 is investigated. The variables in the input unit, especially the overhead product flow, tower top reflux, and feed flow rate, provide the most contribution. The fault detection and isolation results are in accordance with the designed real process status.

The global and local monitoring statistics of fault 2 are presented in Fig. 16. The local monitoring statistic is affected first, followed by the global monitoring statistic. The monitoring

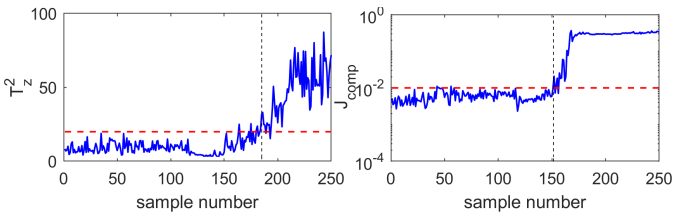


Fig. 16. Global and comprehensive local monitoring statistics for the distillation process fault 2.

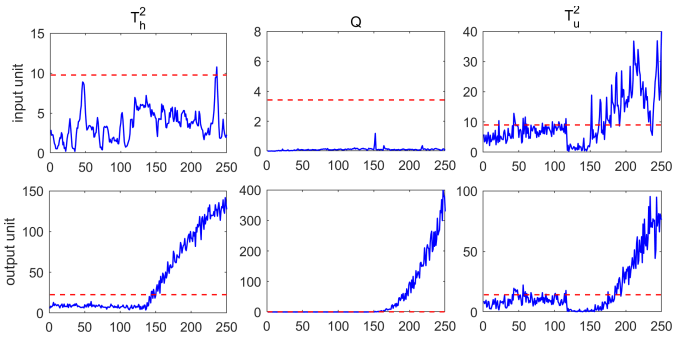


Fig. 17. Monitoring of the local units for the distillation process fault 2.

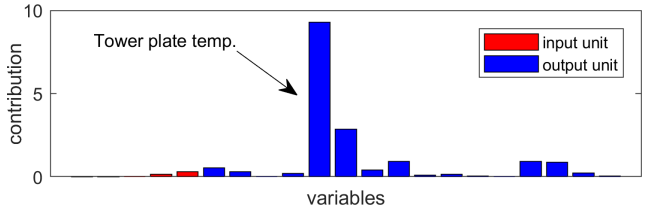


Fig. 18. Variable contribution plots of the distillation process fault 2.

statistic matrix of the local unit shown in Fig. 17 demonstrates that the inner part of the input unit is unaffected. The outer-related parts of the input and output are unaffected at the beginning of the fault. However, the outer-related parts are gradually affected as the fault influence increases. The variable contribution plots at the 200th point are analyzed, as shown in Fig. 18. The tower plate temperature 4 provides significant contributions to the fault. The results are consistent with the designed real process condition. The effectiveness of the proposed local–global modeling and distributed monitoring scheme is demonstrated.

V. Conclusion and Discussions

A local–global modeling and distributed computing framework for efficient monitoring of nonlinear plant-wide processes with industrial big data are proposed in this article. The distributed monitoring framework incorporates SAE-based DNN to characterize the complex variable relationship within a local unit and uses the selection of MI-based relevant variables and DCCA neural network to characterize the relationship between a local unit and its neighboring units. Outer-related representations for each unit are generated to establish a global monitoring model. Given that the distributed monitoring framework involves information from only local and neighboring units, the modeling task can be divided and completed by using different CPUs to make the process computationally efficient. The proposed distributed monitoring framework is tested on the TE benchmark and real glycol distillation processes, and the monitoring results verify its effectiveness.

Notably, the current work is a preliminary study for the monitoring of nonlinear plant-wide process. The following issues should be discussed further.

1) This work focuses only on the fault detection and isolation of large-scale processes. Fault diagnosis and fault path recognition can be performed in future studies.
2) The current work does not deal with the time-varying correlation. Once the variable correlations change with respect to time, additional discussions on the time-varying situation are required.
3) This study uses only the measurement obtained from conventionally used sensors. Multisource heterogeneous signals, such as photo, video, and audio, have become available in industrial data due to the prevalent use of advanced sensing techniques. The fusion of multisource signals and the extraction of correlated representations from different data sources are under investigation.

References

[1] S. J. Qin, "Process data analytics in the era of big data," *AIChE J.*, vol. 60, no. 9, pp. 3092–3100, Sep. 2014.
[2] J. Zhu, Z. Ge, and Z. Song, "Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1877–1885, Aug. 2017.
[3] S. Yin and O. Kaynak, "Big data for modern industry: Challenges and trends [point of view]," *Proc. IEEE*, vol. 103, no. 2, pp. 143–146, Feb. 2015.
[4] X. Peng, Y. Tang, W. Du, and F. Qian, "Multimode process monitoring and fault detection: A sparse modeling and dictionary learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4866–4875, Jun. 2017.
[5] S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. London, U.K.: Springer, 2014.
[6] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Nonlinear process fault diagnosis based on serial principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 560–572, Mar. 2018.
[7] Q. Jiang and X. Yan, "Parallel PCA–KPCA for nonlinear process monitoring," *Control Eng. Pract.*, vol. 80, pp. 17–25, Nov. 2018.
[8] C. Zhao and H. Sun, "Dynamic distributed monitoring strategy for large-scale nonstationary processes subject to frequently varying conditions under closed-loop control," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4749–4758, Jun. 2019.
[9] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.
[10] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
[11] W. Zhong, C. Jiang, X. Peng, Z. Li, and F. Qian, "Online quality prediction of industrial terephthalic acid hydropurification process using modified regularized slow-feature analysis," *Ind. Eng. Chem. Res.*, vol. 57, no. 29, pp. 9604–9614, Jul. 2018.
[12] S. Joe Qin, "Statistical process monitoring: Basics and beyond," *J. Chemometrics*, vol. 17, nos. 8–9, pp. 480–502, Aug. 2003.
[13] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Ind. Eng. Chem. Res.*, vol. 58, no. 29, pp. 12899–12912, Jul. 2019.
[14] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometric Intell. Lab. Syst.*, vol. 171, pp. 16–25, Dec. 2017.
[15] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chem. Eng. Sci.*, vol. 59, no. 1, pp. 223–234, Jan. 2004.
[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[17] L. Cai, X. Tian, and S. Chen, "Monitoring nonlinear and non-Gaussian processes using Gaussian mixture model-based weighted kernel independent component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 122–135, Jan. 2017.
[18] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Deep principal component analysis based on layerwise feature extraction and its application to nonlinear process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2526–2540, Nov. 2019, doi: 10.1109/TCST.2018.2865413.
[19] J. Chen and C.-M. Liao, "Dynamic process fault monitoring based on neural network and PCA," *J. Process Control*, vol. 12, no. 2, pp. 277–289, Feb. 2002.
[20] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.
[21] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[22] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[23] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Comput. Chem. Eng.*, vol. 107, pp. 395–407, Dec. 2017.
[24] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semiconductor Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017.
[25] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
[26] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *J. Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014.
[27] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3235–3243, Jul. 2018.
[28] Q. Sun and Z. Ge, "Probabilistic sequential network for deep learning of complex process data and soft sensor application," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2700–2709, May 2019.
[29] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.
[30] Q. Jiang and X. Yan, "Learning deep correlated representations for nonlinear process monitoring," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6200–6209, Dec. 2019.
[31] Z. Chen, S. X. Ding, K. Zhang, Z. Li, and Z. Hu, "Canonical correlation analysis-based fault detection methods with application to alumina evaporation process," *Control Eng. Pract.*, vol. 46, pp. 51–58, Jan. 2016.
[32] Q. Jiang, S. X. Ding, Y. Wang, and X. Yan, "Data-driven distributed local fault detection for large-scale processes based on the GA-regularized canonical correlation analysis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8148–8157, Oct. 2017.
[33] Z. Chen, S. X. Ding, T. Peng, C. Yang, and W. Gui, "Fault detection for non-Gaussian processes using generalized canonical correlation analysis and randomized algorithms," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1559–1567, Feb. 2018.
[34] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
[35] Q. Jiang, S. Yan, X. Yan, S. Chen, and J. Sun, "Data-driven individual–joint learning framework for nonlinear process monitoring," *Control Eng. Pract.*, vol. 95, Feb. 2020, Art. no. 104235.
[36] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.
[37] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.
[38] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London, U.K.: Springer, 2000.
[39] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May 2014.

Qingchao Jiang (Senior Member, IEEE) received the B.E. degree in measurement and control technology and instrument and the Ph.D. degree in control theory and engineering from the East China University of Science and Technology, Shanghai, China, in 2010 and 2015.

He has been a Post-Doctoral Fellow with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, CA, USA, a Humboldt Research Fellow with the Institute for Automatic Control and Complex Systems (AKS), University of Duisburg–Essen, Duisburg, Germany, and a Visiting Research Fellow in with the Department of Chemical and Biomolecular Engineering, The Hong Kong University of Science and Technology, Hong Kong. He is currently an Associate Professor with the Department of Automation, East China University of Science and Technology. His research interests include data mining and analysis, data-driven soft sensing, multivariate statistical process monitoring, and Bayesian fault diagnosis.

Shifu Yan received the B.E. degree in automation from the East China University of Science and Technology, Shanghai, China, in 2017, where he is currently pursuing the Ph.D. degree in control science and engineering.

His research interests include multivariate statistical process monitoring and deep learning-based process modeling.

Hui Cheng received the B.E. degree from the Automation Department, Shanghai Jiao Tong University, Shanghai, China, in 2001, and the Ph.D. degree from the Department of Chemical Engineering, Helsinki University of Technology, Espoo, Finland, in 2009.

From March 2009 to September 2011, he was a Post-Doctoral Fellow with the Department of Automation, East China University of Science and Technology (ECUST), Shanghai, where he has been an Associate Professor since September 2011. His research interests include data-driven soft sensing, multivariate statistical process monitoring with application in the field of chemical process.

Xuefeng Yan (Member, IEEE) received the B.S. degree in biochemical engineering and the Ph.D. degree in control theory engineering from Zhejiang University, Hangzhou, China, in 1995 and 2002, respectively.

He is currently a Professor with the Department of Automation, East China University of Science and Technology, Shanghai, China. His current research interests include complex chemical process modeling, optimizing and controlling, process monitoring, fault diagnosis, and intelligent information processing.