IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 22, 2021, accepted May 7, 2021, date of publication May 11, 2021, date of current version May 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3079184

Improved Mahalanobis Distance Based JITL-LSTM Soft Sensor for Multiphase Batch Processes

JIAQI ZHENG[1], FEIFAN SHEN[2], AND LINGJIAN YE[3], (Member, IEEE)
[1]College of Science and Technology, Ningbo University, Ningbo 315300, China
[2]School of Information Science and Engineering, NingboTech University, Ningbo 315100, China
[3]School of Engineering, Huzhou University, Huzhou 313000, China

Corresponding author: Feifan Shen (ffshen@nit.net.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673349, in part by the Natural Science Foundation of Zhejiang Province under Grant LQ19F030004, in part by the K. C. Wong Magna Fund in Ningbo University, in part by the Scientific Research Fund of Zhejiang Provincial Education Department under Grant Y201839765, in part by the Foundation of Key Laboratory of Advanced Process Control for Light Industry, Jiangnan University, under Grant APCLI1802 and Grant APCLI1804, in part by the Educational Science Planning Project of Zhejiang Province under Grant 2021SCG184, and in part by the Teaching Reform Research Project of Zhejiang Province under Grant jg20190607.

ABSTRACT To predict key variables of complicated batch processes, the long short-term memory (LSTM) soft sensor is developed to deal with both data nonlinearity and dynamics. To extract proper historical samples and implement the real-time modeling scheme with model updating strategy, the just-in-time learning (JITL) algorithm is widely used at the data selection stage of LSTM soft sensor. However, the multiphase issue of batch processes are not considered for the conventional JITL-LSTM soft sensor. In this paper, a multiphase Mahalanobis distance based JITL framework is developed to integrate the phase recognition strategy into the similarity measurement and data selection scheme, by which an extra step of phase identification can be avoided and the accuracy of JITL can be significantly improved. Thus, batch samples from different operating phases can be recognized without an additional phase identification step. By the use of the Mahalanobis Distance based JITL-LSTM Soft Sensor, the probability of data mismatch can be significantly reduced so that the accuracy of quality prediction can be promoted. Two simulation cases are provided to verify the effectiveness of the proposed method consisting of a fed-batch reactor process and the penicillin fermentation process.

INDEX TERMS Soft sensor, batch production systems, multiphase Mahalanobis distance, long short-term memory, just-in-time learning.

I. INTRODUCTION

Both batch and continuous processes are important modes of production in modern industry. Recently, with the rapid development of process industry, the proportion of batch processes is growing due to the increasing demand of high-value products such as pharmaceuticals, polymers and semi-conductors [1]–[6]. In terms of the situation, researchers focus on the quality prediction and monitoring problem of batch processes to ensure the product quality and process safety. However, the related research starts late so that it is common to apply the soft sensors designing for continuous processes to batch processes. Different from the continuous processes with steady operating condition, the batch process is always involved with frequently changed conditions. In fact, the statistical characteristics of the data collected from the batch process is more complicated than the continuous process [7]–[9].

To make full use of the batch process data to achieve the quality prediction scheme, several methods based on the multivariate statistical process control (MSPC) are developed in the past decades [10]–[12]. Several data-based soft sensors, which construct prediction models by extracting the latent data relationship between easy-to-measure process variables and quality variables are designed by researchers for batch processes. For instance, the partial least squares regression (PLSR) [13]–[15], the support vector regression (SVR) [16], [17] are conventional soft sensors for batch processes, while the artificial neural network (ANN) [18], [19] based methods are investigated to in recent years. By the use of these methods

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

and their improved forms, some complex data characteristics caused by the within-batch variations can be handled with. Kernel functions are adopted in PLSR and SVR based methods to deal with the nonlinear feature of process data, while a few time-series modeling methods are developed to solve the data dynamics. Among these methods, ANN is one of the most widely used model in recent years due to its advantage on dealing with process data with large scale, high dimension and nonlinear feature. However, the training procedure of ANN becomes difficult to accomplish since the gradient exploding problem will take place when the network structure is complicated.

To overcome the deficiencies of ANN, the deep learning algorithms are developed by improving the network structure and the training strategy, which has been applied to many areas such as the natural language processing (NLP) [20], [21], image processing [22], [23], pattern recognition [24], etc.. In the past decade, deep learning algorithms are also exploited to predict key variables of the industrial process [25]. Yuan and his research group develop several quality-relevant feature representation methods for the soft sensor modeling based on the stacked autoencoder (SAE) [26]–[28]. As an advanced structure of the conventional ANN, SAE is able to provide a more efficient modeling strategy to deal with the large-scale complex network and data nonlinearity. However, the SAE soft sensor is under the assumption that the process samples are independent. In practical industrial process, the temporal correlations of the time-series data should be taken into consideration. To extract the time-varying characteristic, the recurrent neural network (RNN) is introduced to construct nonlinear dynamic soft sensor models for the batch processes [29], [30]. However, RNN is also suffered from the issue of the gradient exploding because of its network structure. To this end, the LSTM neural network is designed with the concept of memory cells and gates. Although the LSTM based modeling issue has been studied for a long time, the history of its applications on the data-driven industry process modeling and quality prediction is only a few years. The LSTM based method has been proved as an efficient soft sensor model for nonlinear dynamic industrial processes, which can provide a better performance than the traditional models such as PLS, SAE and RNN [31].

As is known to all that the quality prediction performance of the LSTM based soft sensors is significant. However, the problem of the selection of modeling samples has not been studied intensively. Most existing LSTM based soft sensors used in the batch process are carried out by the global modeling strategy. However, the operating condition may vary from one batch to another gradually due to the raw material residual, the equipment aging, the external environment and other process uncertainties. As a result, it will lead to diverse batch trajectories because of the existence of batch-to-batch variations. Under such a situation, the performance of a static model constructed by the global learning strategy is not satisfying so that model updating is necessary. To implement the online local modeling scheme, the JITL algorithm is developed for model updating. Similarity measurements are conducted to extract the most relevant historical samples to construct the real-time model. Combined with soft sensor models, several JITL based quality prediction methods have been proposed for industrial processes [32], [33]. It is noted that most of the JITL based soft sensors study the data characteristics in the modeling stage, where the similarity measurement stage contributes little to the problem. Besides, the multiphase feature of the batch process has not been focused on in recent works of the soft sensor scheme. As a result, the mismatch of samples belonging to different operating phases may take place and an inaccurate online local soft sensor model will be constructed. Thus, the prediction result will definitely deviate from the true value although the current input trajectory is similar to the historical modeling trajectory.

Due to the aforementioned deficiency of the conventional JITL based soft sensor, the main purpose of this work is to develop an improved metric learning strategy for the JITL algorithm, where historical samples of different phases will no longer be extracted as the modeling samples. In order to address the problem of the multiple operating phases of the batch process and make full use of the JITL strategy, a multiphase Mahalanobis distance based JITL (MMJITL) method is designed to improve the performance of the relevant data collection step. A multiphase metric learning strategy is introduced with triplet constraints to determine the Mahalanobis matrix before the implementation of JITL. The samples belonging to different operating phases are expected to be discriminated without a specific phase identification step. Combined with the nonlinear dynamic LSTM soft sensor model, both the within-batch data characteristics and the batch-to-batch variations of the multiphase batch process can be handled with. Therefore, the prediction accuracy of the online local LSTM model can be significantly improved due to the novel modeling sample selection strategy based on JITL.

The rest of this paper is presented as follows. The next section provides a brief introduction of the fundamental approaches. Section 3 described the methodology and the procedure of the proposed MMJITL-LSTM soft sensor in detail. Then, 2 simulation cases are carried out as the testification of the proposed method. Finally, conclusions are made and future works are prospected.

II. PRELIMINARIES

A. EUCLIDEAN DISTANCE BASED JITL

For a stationary process, it is feasible to construct a conventional offline global model since all process samples can be involved with similar data characteristics. However, for batch processes, especially for those complicated processes with multiple operating phases, process dynamics and nonlinearity should be taken into consideration. As a result, the global learning and modeling strategy may fail to capture complex process features within a single model.

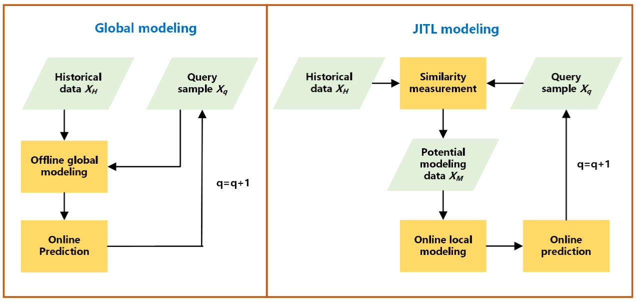

FIGURE 1. The comparison between the global modeling and the JITL modeling.

Recent years, the just-in-time learning method is developed to solve the aforementioned issues during the modeling stage of complex processes. Different from the traditional offline global learning, JITL is a local modeling technique by extracting relevant historical samples according to a predefined sample similarity measurement. Besides, the modeling stage of the JITL is executed online and the model can be continuously updated while the query samples are usually varies during different phases. The procedure of the JITL can mainly concluded to three steps. Firstly, historical samples with the largest similarity index are selected as the online modeling samples. Secondly, an online model for prediction, monitoring or other applications is constructed with the selected samples. Finally, the model output can be obtained with the query samples and the trained online model. Taking the quality prediction for example, the comparison between the global modeling and the JITL strategy is presented in Fig.1 [34].

The crucial point of the JITL method is to establish a reasonable sample selection scheme. As aforementioned, computations of the similarity between query samples and historical samples are necessary to evaluate the sample relevance. The most widely used similarity measurement is the Euclidean distance based approach, which is described in (1).

$$\begin{aligned} d_i &= \left\| x_q - x_i \right\| \\ s_i &= \exp(-d_i) \end{aligned} \tag{1}$$

where d_i is the Euclidean distance between a query sample x_q and the historical sample x_i; s_i is the final Euclidean distance based similarity factor. Therefore, samples with smaller distance and larger similarity factor will be selected as the potential online modeling samples.

In order to obtain a more precise similarity measurement, the angle between samples are also considered as a complementary factor to the Euclidean distance. Hence, the improved similarity measurement is described in (2).

$$\begin{aligned} d_i &= \left\| x_q - x_i \right\| \\ \cos(\theta_i) &= \langle x_q, x_i \rangle / \left(\left\| x_q \right\|_2, \|x_i\|_2 \right) \\ s_i &= \lambda \exp(-d_i) + (1-\lambda)\cos(\theta_i) \end{aligned} \tag{2}$$

In addition, $\cos(\theta_i)$ describes the angle between the query sample and the x_q and the historical sample x_i; λ is the weighting parameter as a tradeoff between the distance and angle. When $\lambda = 1$, only the distance measurement is adopted as Equation (1). It is noted that if $\cos(\theta_i) < 0$, the similarity factor s_i should be discarded since the direction of the vectors for computation are absolutely opposite.

B. MAHALANOBIS DISTANCE BASED JITL

The Euclidean distance measures the absolute position of samples, which may lose the information of the correlation between different variables. To overcome the deficiency, the Mahalanobis distance is proposed to extract data correlations by developing a weighting matrix [35]. The Mahalanobis distance based JITL method is shown in (3).

$$\begin{aligned} d_i &= \left(x_q - x_i\right)^T M \left(x_q - x_i\right) \\ s_i &= \exp(-d_i) \end{aligned} \tag{3}$$

It can be inferred from (3) that an additional matrix M is designed as the weighting matrix, which is a symmetric PSD matrix named as Mahalanobis matrix. The most widely used Mahalanobis matrix is the inverse matrix of the covariance matrix S, where $M = S^{-1}$. The definition of S is presented in (4).

$$S = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}_i\right)\left(Y_i - \bar{Y}_i\right)}{n-1} \tag{4}$$

where X_i, Y_i are two datasets and $\bar{X}_i$; $\bar{Y}_i$ are their mean matrix, respectively; n represents the total sample number in each dataset. Thus, the information of the covariance is adopted as the weighting factor during the calculation of Mahalanobis distance between two samples. When $S^{-1} = I$, which means it is an identity matrix, the Mahalanobis distance reduces to the Euclidean distance.

C. LONG SHORT-TERM MEMORY NEURAL NETWORK

To address the vanishing error problem of the conventional recurrent neural network (RNN), LSTM is developed as an improved gradient-based framework. LSTM keeps the advantage of RNN, where the time-varying and nonlinear features of process data can be extracted properly. Besides, by the development of cells and memory blocks, the exploding gradient problem is solved and long-term information is captured. Due to the aforementioned advantages, LSTM has been successfully applied to the soft sensing problem for complex processes with time-varying and nonlinear data characteristics [36]. The detailed structure of the LSTM soft sensor is illustrated in Fig.2.

According to Fig.2, the LSTM soft sensor model can be summarized as (5).

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ \tilde{c}_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ h_t &= o_t \odot \tanh(c_t) \\ \hat{y}_t &= \sigma\left(W_y h_t + b_y\right) \end{aligned} \tag{5}$$

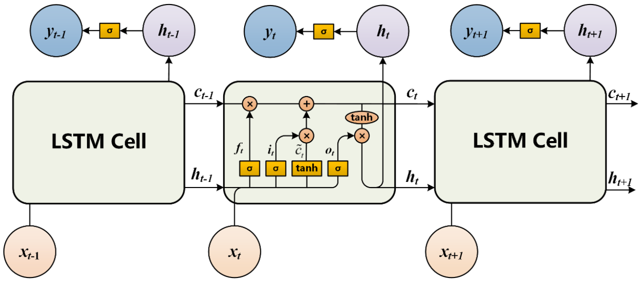

FIGURE 2. The structure of the LSTM soft sensor.

where i_t, f_t, o_t are the input gate, forget gate and output gate, respectively; $\sigma(\cdot)$ is the sigmoid activation function as

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

x_t is the model input at time instant t; h_t is the hidden state of the LSTM cell at time instant t; W_{**} are the weighting parameters in different gate structures and b_* are the biases; $\tilde{c}_t$ determines the relevant part of the cell input at time instant t by a tanh function. Therefore, the cell state at time instant t c_t can be defined as the sum of two pointwise multiplications involved with the information of the forget gate, the cell state of the previous time instant, the input gate and $\tilde{c}_t$. Hence, the hidden state h_t can be calculated by the pointwise multiplication of the output gate and the cell state. Finally, the predicted output of the LSTM soft sensor $\hat{y}_t$ can be figured out based on h_t.

To train a LSTM soft sensor with historical dataset, the back propagation through time (BPTT) method can be used to determine the model parameters as (7).

$$\begin{aligned}
L &= \frac{\sum_{t=1}^{T}\left(\hat{y}_t - y_t\right)^2}{T} \\
\delta h_t &= \frac{\partial L}{\partial h_t} + \delta i_{t+1} W_{hi}^T + \delta i_{f+1} W_{hf}^T + \delta o_{t+1} W_{ho}^T \\
&\quad + \delta \tilde{c}_{t+1} W_{h\tilde{c}}^T \\
\delta i_t &= (\delta h_t)^T \odot o_t \odot \left(1 - \tanh^2(c_t)\right) \odot c_{t-1} \\
&\quad \odot (i_t \cdot (1 - i_t)) \\
\delta f_t &= (\delta h_t)^T \odot o_t \odot \left(1 - \tanh^2(c_t)\right) \odot \tilde{c}_t \odot (f_t \cdot (1 - f_t)) \\
\delta o_t &= (\delta h_t)^T \odot \tanh(c_t) \odot (o_t \cdot (1 - o_t)) \\
\delta \tilde{c}_t &= (\delta h_t)^T \odot o_t \odot \left(1 - \tanh^2(c_t)\right) \odot i_t \\
&\quad \tilde{\odot} \left(1 - \tanh^2(\tilde{c}_t)\right) \\
\delta W_{x*} &= \sum\nolimits_{t=1}^{T} \langle \delta *_t, x_t \rangle \\
\delta W_{h*} &= \sum\nolimits_{t=1}^{T} \langle \delta *_t, h_{t-1} \rangle \\
\delta b_* &= \sum\nolimits_{t=1}^{T} \delta *_t
\end{aligned} \tag{7}$$

where L is the loss function; T is sample number for training; $\delta *_t$ are the gradients of different model parameters at time instant t; $\odot$ denotes the pointwise multiplication operation for two vectors. Therefore, the prediction error between the predicted output $\hat{y}_t$ and the real output y_t can be propagated back through the time series. By the use of gradient descent algorithms, the values of model parameters W_{h*}, W_{x*}, b_* can be determined after several iterations.

III. IMPROVED MAHALANOBIS DISTANCE BASED JITL-LSTM SOFT SENSOR

A. MULTIPHASE MAHALANOBIS DISTANCE and METRIC LEARNING

When handling with the soft sensor problem of multiphase batch processes, it is feasible to implement the JITL based online local modeling and prediction. Different from the multimode problem caused by batch-to-batch variations, the within-batch multiphase feature indicates that several operating stages exist during a single batch run due to the mechanism of reactions. For complex batch processes, data correlations always exist among process samples due to the coupling of different process units. Therefore, the Mahalanobis distance tends to provide a better metric of similarity comparison than the Euclidean distance based JITL. As mentioned in section 2, the key point of the Mahalanobis distance is the determination of the Mahalanobis matrix M. $M = S^{-1}$, which is the inverse form of the covariance matrix, is simple to carry out. However, it is a global metric learning strategy which regards all data samples as an entirety. The sample diversity of different batch phases is expected to be exploited during the Mahalanobis distance computation.

To this end, a multiphase LogDet Divergence based metric learning algorithm is introduced to determine the Mahalanobis matrix M for the data of multiphase batch processes [37], [38]. Assume n operating phases exist during a batch run and given the process dataset $X^p, p = 1, 2, \ldots, n$, it turns out as an optimization problem to figure out the best M to minimize the Mahalanobis distance between samples belonging to the same phase and maximize the distance between samples of different phases at the same time. In this work, the k-means clustering algorithm is adopted to divide historical batches into n phases. Define the Mahalanobis distance between two samples as

$$d_M\left(x_i, x_j\right) = \left(x_i - x_j\right)^T M \left(x_i - x_j\right) \tag{8}$$

The metric learning of the multiphase Mahalanobis matrix is shown in Fig.3, where the constraints of the optimization problem can be illustrated as

$$\begin{aligned}
D_M\left(X_i^p, X_j^p\right) &= d_M\left(x_i^p, x_j^p\right) \leq \varepsilon, \\
&\qquad x_i^p \in X^p, \quad x_j^p \in X^p, \ i \neq j \\
D_M\left(X_i^p, X_k^q\right) &= d_M\left(x_i^p, x_k^q\right) > \varepsilon, \\
&\qquad x_i^p \in X^p, \quad x_k^q \in X^q, \ p \neq q
\end{aligned} \tag{9}$$

where ε is the threshold which can distinguish samples of one phase category from another; l denotes the total sample number of phase p and m is the sample number of phase q. To satisfy the optimization constraints, integrated constraints named as triplet constraints are designed to simplify the optimization problem. The triplet constraints $\left\{X_i^p, X_j^p, X_i^q\right\}$ is defined as

$$D_M\left(X_i^p, X_j^p\right) - D_M\left(X_i^p, X_k^q\right) < -\rho \tag{10}$$

where ρ is the target margin with a positive value. Assume the current Mahalanobis matrix is $M(t)$. If $M(t)$ satisfies the triplet constraints $\left\{X_i^p, X_j^p, X_i^q\right\}$, $M = M(t)$ can be determined as the proper Mahalanobis matrix. Otherwise, More iterations should be conducted to reduce the following loss based on Equation (10).

$$L(M) = \rho + D_M\left(X_i^p, X_j^p\right) - D_M\left(X_i^p, X_k^q\right) \tag{11}$$

To avoid dramatic changes and ensure steady changes of M, the LogDet divergence algorithm is used to regularize the optimization process as

$$LD(M, M(t)) = tr\left(MM(t)^{-1}\right) - \log\left(\det\left(MM(t)^{-1}\right)\right) - n \tag{12}$$

where $tr(\cdot)$ denotes the trace of the matrix; $\det(\cdot)$ is the determinant; n represents the dimension of M. Therefore, the updated value of $M(t+1)$ can be calculated by a minimization problem as illustrated in (13).

$$M(t+1) = \underset{M>0}{\arg\min}\ LD(M, M(t)) + \alpha L(M) \tag{13}$$

where α is a weighting parameter between the LogDet function and the loss function. To solve the optimization problem, the gradient of the function should be zero to reach the global minimum since the second order derivative of the function is positive. Thus, the solution of the optimization problem can be demonstrated as

$$M(t+1) = \left(M(t)^{-1} + \alpha\left(YY^T - ZZ^T\right)\right)^{-1} \tag{14}$$

where $Y_{m\times p} = \left(X_i^p\right)_{m\times p} - \left(X_j^p\right)_{m\times p}$ and $Z_{m\times q} = \left(X_i^p\right)_{m\times q} - \left(X_k^q\right)_{m\times q}$ denotes the historical sample diversities used for calculation which belong to the same category and different phases, respectively. To reduce the computational load, not all the samples are used to build the triplet constraints. It is noted that it is an offline procedure based on historical dataset, where the computational complexity is not necessarily to be worried about. Meanwhile, the selection of α follows the rule as

$$\begin{cases} \alpha\left(YY^T - ZZ^T\right) + M(t)^{-1} \geq 0 \\ \alpha \geq 0 \end{cases} \tag{15}$$

Thus, a constant value of α can be determined to make sure that $M(t+1)$ is a PSD matrix.

B. MULTIPHASE MAHALANOBIS DISTANCE BASED SOFT SENSOR

Based on the normalized historical batch dataset with phase labels, the Mahalanobis matrix M is determined as demonstrated in subsection 3.2. Hence, the multiphase Mahalanobis distance based JITL (MMJITL) soft sensor model can be constructed. Assume that n consecutive online query samples are collected as the input trajectory, the similarity factors should firstly be figured out to obtain the most relevant historical input trajectory. Define the normalized query samples as $X_q = \left\{x_{q-n+1}, x_{q-n+2}, \ldots, x_q\right\}$, the similarity factor between the query trajectory X_q and the historical trajectory $H_k = \{h_{k-n+1}, h_{k-n+2}, \ldots, h_k\}$ can be expressed as

$$\begin{aligned} d_{i,j} &= \sqrt{\left(x_i - h_j\right)^T M \left(x_i - h_j\right)} \\ s_k &= \frac{\sum \exp\left(-d_{i,j}\right)}{n} \quad i = q-n+1,\ j = k-n+1; \\ &\qquad i = q-n+2, \quad j = k-n+2; \ldots; \\ &\qquad i = q-1, \quad j = k-1;\ i = q,\ j = k \end{aligned} \tag{16}$$

According to the result of similarity measurements, the historical process trajectory H_k with the largest value of the similarity factor s_k will be extracted as the training samples as well as the corresponding quality variables $Y_k = \{y_{k-n+1}, y_{k-n+2}, \ldots, y_k\}$. Thus, the online local LSTM soft sensor model $\hat{Y}_q = f\left(X_q\right)$ can be constructed with the collected model input H_k and the output Y_k referring to Fig.2 and (5).

To train the LSTM soft sensor model, several gradient descent algorithms can be adopted to estimate the model parameters where the gradients are listed in (7). It has been proved that the Adam algorithm has its advantages over other conventional algorithms such as the momentum gradient descent approach. However, there are some weaknesses during the procedure of the Adam algorithm. First of all, the learning rate may change rapidly at the later training period due to the use of the second-order moment estimation, which makes it difficult to converge. Meanwhile, as an algorithm with the adaptive learning rate, the over-fitting problem will occur at the early stage of the training procedure and miss the global optimal solution.

In order to fix the problem, the AMSGrad algorithm is applied to regularize the learning rate as expressed in Equation (16) [39].

$$\begin{aligned} m_w(t) &= \beta_1 m_w(t-1) + (1-\beta_1)\,\delta W(t) \\ m_b(t) &= \beta_1 m_b(t-1) + (1-\beta_1)\,\delta b(t) \\ v_w(t) &= \beta_2 v_w(t-1) + (1-\beta_2)\,\delta W(t) \odot \delta W(t) \\ v_b(t) &= \beta_2 v_b(t-1) + (1-\beta_2)\,\delta b(t) \odot \delta b(t) \\ \hat{v}_w(t) &= \max\left(v_w(t-1), v_w(t)\right) \\ \hat{v}_b(t) &= \max\left(v_b(t-1), v_b(t)\right) \\ W(t+1) &= W(t) - \alpha_t m_w(t) \Big/ \sqrt{\hat{v}_w(t) + \varepsilon} \\ b(t+1) &= b(t) - \alpha_t m_b(t) \Big/ \sqrt{\hat{v}_b(t) + \varepsilon} \end{aligned} \tag{17}$$

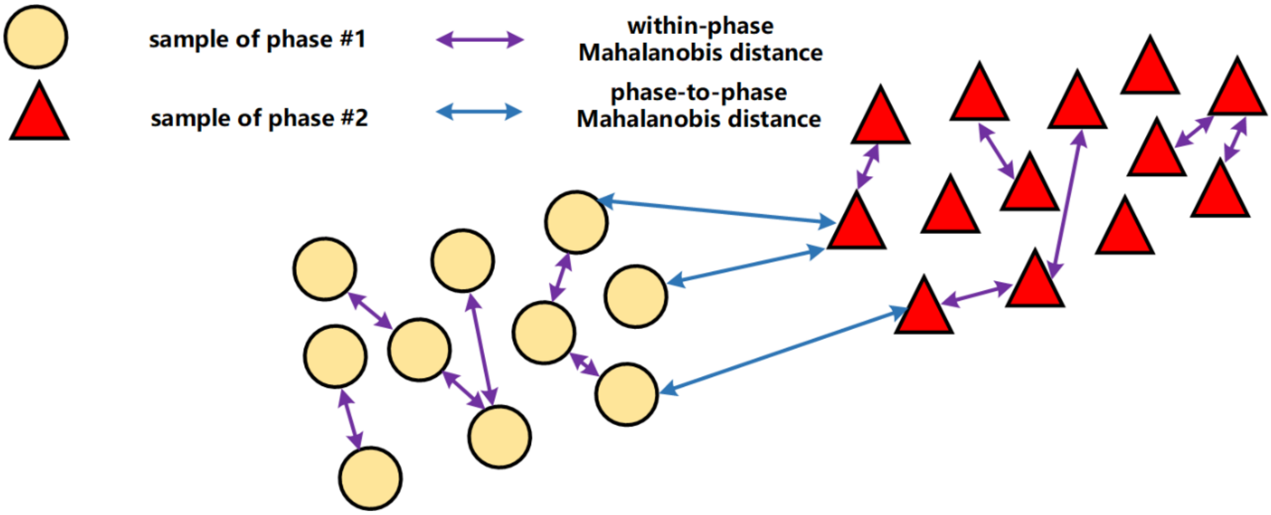
sample of phase #1
sample of phase #2
within-phase Mahalanobis distance
phase-to-phase Mahalanobis distance

FIGURE 3. The metric learning of the multiphase Mahalanobis matrix.

where $W(t) = W_{**}(t)$ and $b(t) = b_{*}(t)$ indicate the value of a certain gradient or bias at step t; $m_w(t)$ and $m_b(t)$ are the first-order moment estimations of the parameters; $\delta W(t)$ or $\delta b(t)$ denote the gradients of the parameters at step t; $v_w(t)$ and $v_b(t)$ are the second-order moment estimations; β_1 and β_2 are the decay rates; $\hat{v}_w(t)$ and $\hat{v}_b(t)$ record the maximum value of the second-order moment estimations to adjust the learning rate; ε is a small constant for computational stability; α_t is the step size of the gradient descending procedure which can be adjusted if the bias is incorrect. Therefore, after the computation is converged, all the model parameters are estimated online and the predicted output can be figured out as $\hat{Y}_q$.

To evaluate the effectiveness of the MMJITL-LSTM soft sensor model, two major indications are adopted. Firstly, the root mean squared error (RMSE) is calculated as

$$\mathrm{RMSE}_q = \sqrt{\frac{1}{n}\sum_{i=q-n+1}^{q}\left(\hat{y}_i - y_i\right)^2} \tag{18}$$

Obviously, a smaller value of the RMSE indicates a better performance of the soft sensor model. Besides, the mean absolute percentage error (MAPE) is also adopted and expressed as

$$\mathrm{MAPE}_q = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{19}$$

Similarly, the soft sensor model performs better when its MAPE is closer to 0.

In summary, the complete procedure of the proposed MMJITL-LSTM based soft sensor for multiphase batch processes is illustrated in Fig.4.

The main steps of the procedure are listed as follows:

(i) Unfold the three-way historical raw data $H_r \in \mathbb{R}^{I\times J\times K}$ into the two-way form $H \in \mathbb{R}^{J\times IK}$ and implement the data normalization.

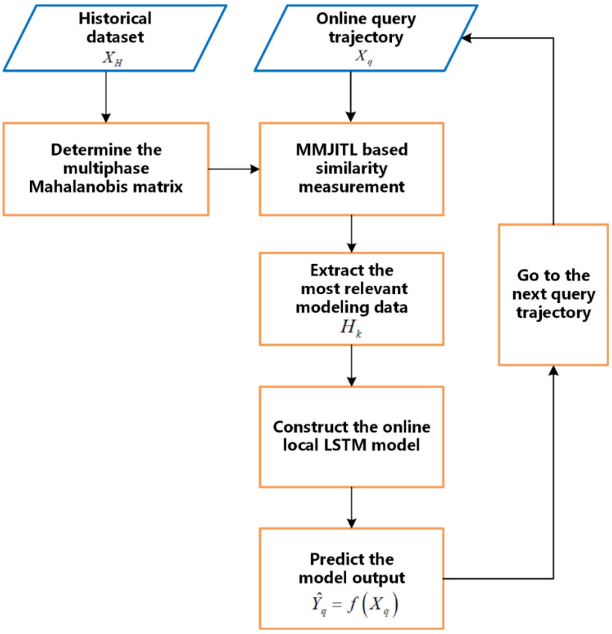
Historical dataset X_H
Online query trajectory X_q
Determine the multiphase Mahalanobis matrix
MMJITL based similarity measurement
Extract the most relevant modeling data H_k
Go to the next query trajectory
Construct the online local LSTM model
Predict the model output $\hat{Y}_q = f(X_q)$

FIGURE 4. The flow diagram of the MMJITL-LSTM soft sensor.

(ii) Learn the Mahalanobis matrix M of the multiphase batch process with labeled historical samples using the LogDet based metric learning algorithm;

(iii) Obtain the real-time input trajectory $X_q = \{x_{q-\mathrm{n}+1}, x_{q-\mathrm{n}+2}, \ldots, x_q\}$ and normalize the samples referring to historical data;

(iv) Calculate the similarity factors s_k between X_q and consecutive historical trajectories H_k based on the MMJITL algorithm;

(v) Extract the historical input trajectory and its output with the largest value of the similarity factor as the potential modeling data.

(vi) Construct the online local models using the LSTM soft sensor based on the extracted data;

(vii) Estimate the model parameters using the BPTT method based on the AMSGrad gradient descend algorithm;

(viii) Calculate the predicted output of the input trajectory X_q according to $\hat{Y}_q = f\left(X_q\right)$, where $\hat{Y}_q = \left\{\hat{y}_{q-n+1}, \hat{y}_{q-n+2}, \ldots, \hat{y}_q\right\}$;

(ix) Go to the next query trajectory for quality prediction.

C. DISCUSSIONS

There are two main contributions in the proposed framework. First of all, the multiway Mahalanobis distance based metric learning is introduced to determine the Mahalanobis matrix for the similarity measurement purpose. Compared to the traditional Mahalanobis distance using the covariance matrix, the proposed metric learning method of the Mahalanobis matrix takes the multiphase feature of batch process data. Combined with the JITL based online local modeling strategy, the chances of the mismatch between the online query trajectory and the historical data which belong to different phases can be significantly reduced. Meanwhile, the AMSGrad gradient descend algorithm is adopted to train the real-time LSTM soft sensor, which is expected to make the training process of the soft sensor model more stable and reliable.

With the real-time query batch trajectory, the most relevant historical samples can be extracted and an online local soft sensor model can be further established to predict the current output. Taking advantages of the aforementioned scheme, the MMJITL-LSTM soft sensor is of great significance to deal with the quality prediction problem of the multiphase batch processes. Both the within-batch data dynamics and nonlinearity can be dealt with owing to the LSTM model, while the multiphase property and the batch-to-batch variations are considered due to the improved Mahalanobis distance based JITL. To prove the effectiveness of the proposed framework, two simulation cases are presented in the following section.

IV. SIMULATION RESULTS

A. A FED-BATCH REACTOR PROCESS

In the first simulation case, a fed-batch reactor process is used to testify the effectiveness of the proposed method. As shown in Figure 5, two reactions take place as $A + B \rightarrow C$ and $2B \rightarrow D$, where A and B are reactants, C is the product and D is the byproduct. For the two reactants, A is fed at the beginning of the batch and B is continuously fed during a batch run [40].

The detailed mathematical model of the process is presented as

$$\begin{aligned}
\dot{c}_A &= -k_1 c_A c_B - c_A - \frac{c_A u}{V}, \\
c_A(0) &= c_{A0} \\
\dot{c}_B &= -k_1 c_A c_B - 2k_2 c_B^2 - \frac{\left(c_A - c_B^{in}\right) u}{V}, \\
c_B(0) &= c_{B0} \\
\dot{V} &= u, \quad V(0) = V_0
\end{aligned}$$

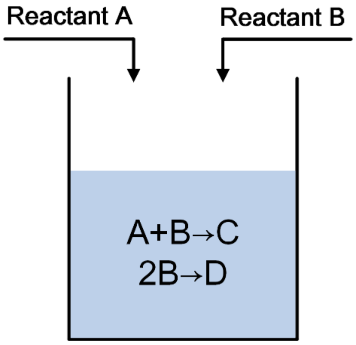

FIGURE 5. The fed-batch reactor process.

TABLE 1. Variables of the fed-batch reactor process.

Variable index	Variable description
1	Concentration of reactant A (mol/L)
2	Concentration of reactant B (mol/L)
3	reactor holdup (L)
4	feed rate of reactant B (L/min)
5	Concentration of product C (mol/L)

$$\begin{aligned}
\dot{c}_C &= k_1 c_A c_B - \frac{c_C u}{V}, \quad c_C(0) = c_{C0} \\
\dot{c}_D &= k_2 c_B^2 - \frac{c_D u}{V}, \quad c_D(0) = c_{D0}
\end{aligned} \tag{20}$$

where c_A, c_B, c_C, c_D are the concentrations of A, B, C, D, respectively; V denotes the volume; k_1, k_2 are two major kinetic coefficients; c_B^{in} is the inlet concentration of B; u is the feed rate of B as the manipulated variable.

The relevant variables for quality prediction are listed in Table. 1, where c_C is the quality variable to be predicted and the others are process variables.

Totally 20 historical batches are generated as historical batches, where 220 consecutive samples exist in each batch. To produce batch-to-batch variations in the simulation, the operating condition varies from batch to batch. In the early stage of the reaction, more B is fed into the reactor to produce the desired product C as phase #1. The influence of the reactant B feed rate becomes less significant at the rest stage of the reaction. Hence, two phases are recognized for the fed-batch reactor process.

For the determination of the multiphase Mahalanobis matrix, the quantity of triplets in each cycle is set as 20 and the total iteration number is 15. Given an online query trajectory of 5 consecutive samples, the historical trajectory of the same length with the largest value of the similarity factor is selected as the potential modeling data and the corresponding LSTM soft sensor model is constructed.

During the modeling stage, the size of the LSTM input is set as 4 since it is the number of the quality-relevant process variables as listed in Table 1. Meanwhile, the scale of the cell output is 1 representing the concentration of C.

The simulation is run in MATLAB 2020b, and the code of the batch reactor case can be downloaded at

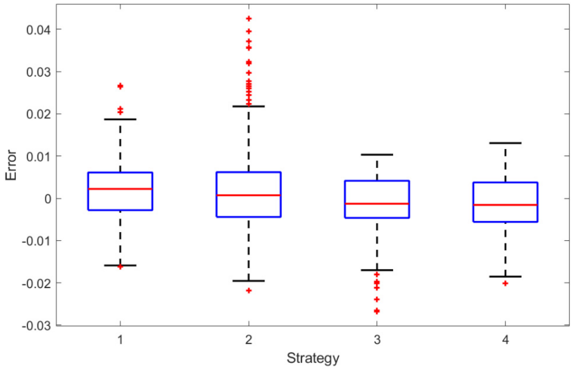

FIGURE 6. The error distributions of predictions for the fed-batch reactor process.

https://github.com/shenfeifan/MMJITL-LSTM-soft-sensor. After the prediction of the online quality $\hat{Y}_q$, the RMSE and MAPE are calculated to evaluate the performance of the MMJITL-LSTM soft sensor. To make comparisons, simulations using the Euclidean distance based JITL (EJITL), the Euclidean distance and angle based JITL (EAJITL), and the Mahalanobis distance based JITL (MJITL) are also carried out and combined with the LSTM soft sensor for quality prediction. For all the aforementioned methods, the model parameters of the LSTM soft sensor are the same to ensure fair comparisons are conducted. The number of the neurons in the LSTM hidden layer is set as 25. The decay rates of the AMSGrad algorithm are 0.9, 0.999 for β_1, β_2, respectively. The initial value of the step size α is 0.001. The value of the constant ε is 10^{-8}. The prediction errors of all the methods are listed in Table. 2.

It can be inferred from Table. 2 that the MMJITL-LSTM method offers the less RMSE and MAPE compared to other methods, which indicates that the performance of the proposed soft sensor is obviously promoted. Besides, the box plots of the prediction errors are shown in Fig. 6.

TABLE 2. Quality prediction results of the fed-batch reactor process.

	EJITL -LSTM	EAJITL -LSTM	MJITL -LSTM	MMJITL -LSTM
RMSE	0.0071	0.0075	0.0121	**0.0067**
MAPE	0.0337	0.0211	0.0313	**0.0206**

The boxes indicate the ranges between the upper and lower quartiles, which consist of 50% results of the prediction error for each method. It also can be observed that the number outliers "+" of these methods are 5, 20, 7 and 1, respectively. It demonstrates that the chance of mismatch between trajectories of different phases can be reduced significantly by the use of the MMJITL based metric learning and local modeling strategy. Meanwhile, the prediction results of these methods compared to the real values are integrated and shown in Fig.7.

B. THE PENICILLIN FERMENTATION PROCESS

In this part, the proposed method is further simulated on the fed-batch penicillin fermentation process to verify the effectiveness. The flow chart of the penicillin process is presented in Fig.8.

The fed-batch penicillin fermentation process can be divided into 3 operating phases. The first phase is the preculture stage in which the biomass reactants grow for preparation of the reaction. Next, the penicillin product occurs at the second phase and grows fast due to the strong reaction. Finally, the accumulation of the penicillin becomes steady and slower until the end of a batch. For the implementation of simulation, The PenSim v2.0 benchmark software is widely in most literatures for case studies [41]. To make the simulation more accurate and reliable, the penicillin simulation

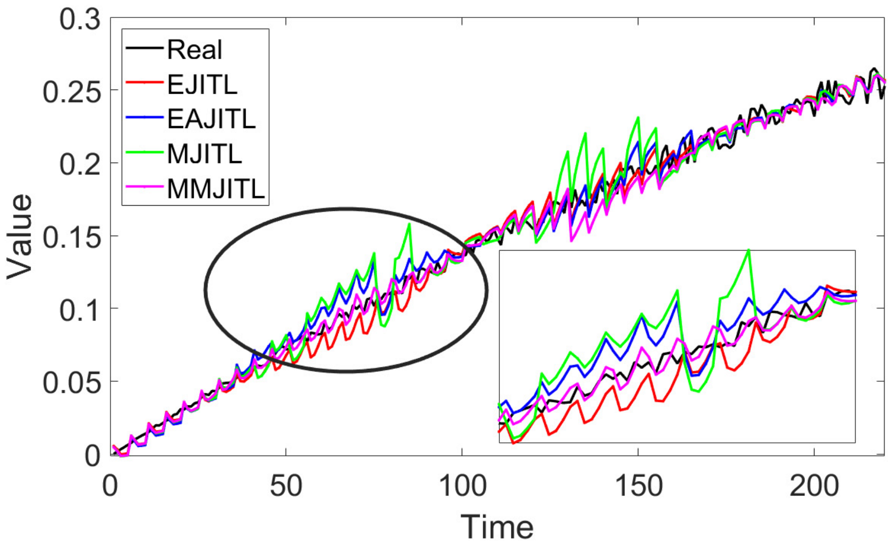

FIGURE 7. The predicted results compared to real values for the batch reactor process.

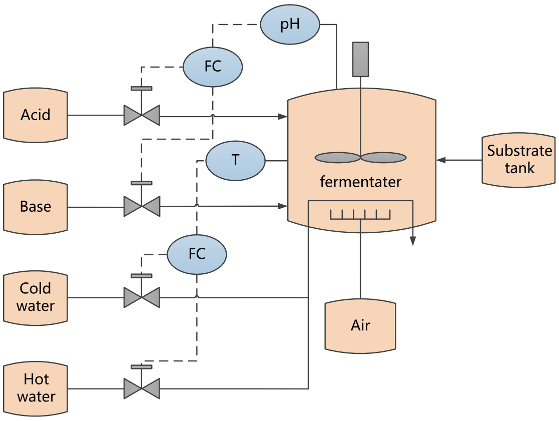

FIGURE 8. The flow chart of the penicillin fermentation process.

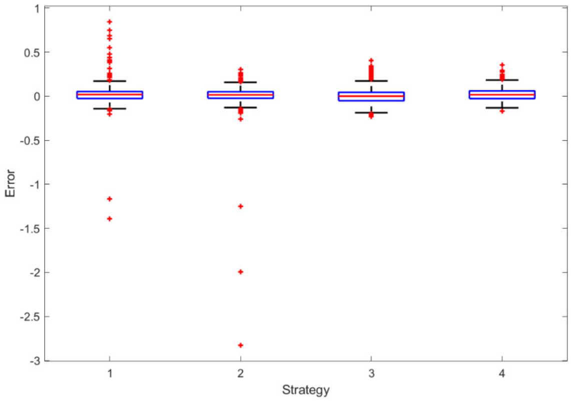

FIGURE 9. The error distributions of predictions for the penicillin fermentation process.

platform is improved by us on MATLAB to generate the process data [42]. Customized process trajectories with optional manipulated variables can be defined in our platform, which has been proved more flexible than the PenSim software. Thus, batch-to-batch variations can be generated easily in the penicillin simulation.

TABLE 3. Variables of the penicillin fermentation process.

Variable index	Variable description
1	aeration rate (L/h)
2	agitator power (W)
3	substrate feed rate (L/h)
4	substrate temperature (K)
5	substrate concentration (g/L)
6	dissolved oxygen concentration (g/L)
7	biomass concentration (g/L)
8	culture volume (L)
9	CO_2 concentration (g/L)
10	pH
11	generated heat (KCal)
12	cooling water flow rate (mL/h)
13	penicillin concentration (g/L)

The variables used for soft sensor modeling are listed in Table. 3, where the penicillin concentration is regarded as the quality variable to be predicted, while other 12 variables are defined as the process variables.

The running time of a single batch of the fed-batch penicillin fermentation process is 400 hours and the sampling interval is 1 hour. Hence, there are 400 samples in a single batch run. 5 batches are generated as the historical batches. As mentioned in the process description, 3 diverse operating phases are included in each batch of the penicillin fermentation process.

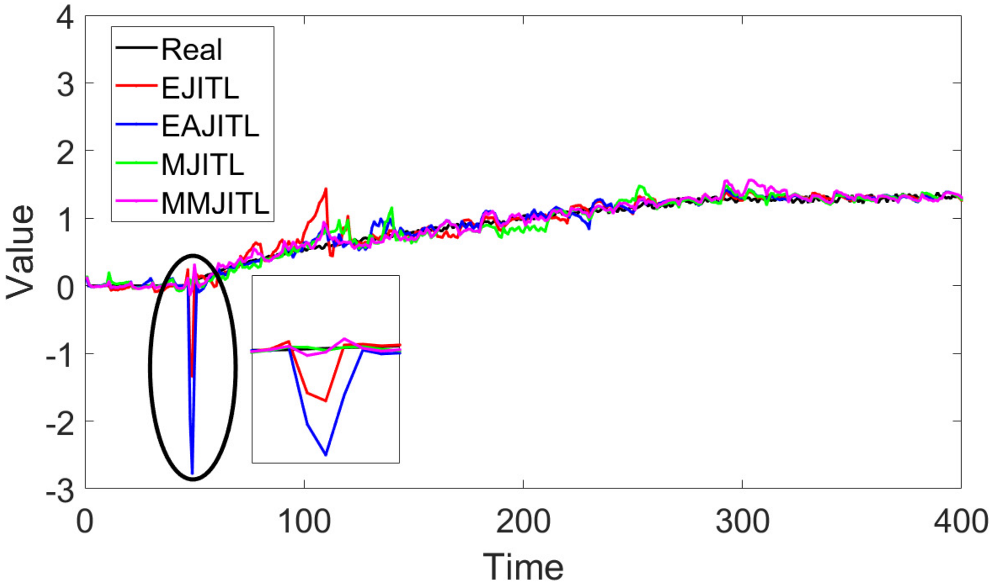

FIGURE 10. The predicted results compared to real values for the penicillin process.

TABLE 4. Quality prediction results of the penicillin fermentation process.

	EJITL -LSTM	EAJITL -LSTM	MJITL -LSTM	MMJITL- LSTM
RMSE	0.1476	0.1979	0.0930	**0.0820**
MAPE	0.1851	0.1159	0.1722	**0.1133**

The quantity of triplets in each cycle of the learning procedure of the Mahalanobis matrix is set as 25 and the total cycle number is 15. The length of the online query trajectory is set as 10. The size of the LSTM input is set as 12 and the size of the output is 1 according to Table 3. The number of the neurons in the LSTM hidden layer is set as 70. The parameters of the AMSGrad gradient descend algorithm are the same with the first simulation case.

Similarly, the RMSE and MAPE are calculated to prove the effectiveness of the MMJITL-LSTM soft sensor after $\hat{Y}_q$ is predicted. The comparisons to the EJITL-LSTM, the EAJITL-LSTM and the MJITL-LSTM are also conducted, where the prediction errors are listed in Table. 4. Model parameters of the LSTM soft sensor for these methods are the same to the first case except for the neuron numbers.

In this simulation case, the MMJITL-LSTM soft sensor provides a better quality prediction performance than other methods as well. The corresponding box plots of the prediction errors are also shown in Fig.9. The number of the outliers are 32, 27, 27 and 19, respectively. The prediction results of these methods for the penicillin simulation case compared to the real values are presented in Fig.10. It is noted that when the Euclidean distance based JITL is applied, serious prediction errors occur at the between-phase transition stage. The results definitely demonstrate the advantages of the proposed method in dealing with the quality prediction problem of multiphase batch processes.

V. CONCLUSION

In this work, a MMJITL-LSTM soft sensor framework is developed for quality prediction of multiphase batch processes. By the metric learning of the multiphase Mahalanobis matrix, samples of different operating phases can be distinguished without a specific phase identification step during the JITL implementation. Together with the LSTM soft sensor, the complicated characteristics of the unstationary batch process can be handled with thoroughly including the multiphase property, the data dynamics and the nonlinearity. Two simulation cases are carried out including the fed-batch reactor process and the penicillin fermentation process. The prediction results of the cases compared with the conventional JITL-LSTM soft sensors prove the effectiveness of the proposed method.

However, there are some limitations of the proposed method. Firstly, the proposed method is under the assumption that both the batch duration and the sampling interval are equal for different batches, where the acquisition of phase indexes is not difficult. Due to the complexity of practical batch processes, run-to-run batch variations will lead to the diversity of batch trajectory. Hence, practical batch processes with the uneven length will be focused on to study the local modeling and quality prediction problem with diverse sampling intervals in future research. Meanwhile, the performance of the original soft sensor model has not been focused on in the current work, where the case studies are mainly carried out to testify the performance of the improved modeling data extraction method. In future work, more state-of-the-art algorithms such as CNN-LSTM and attention-based LSTM will be introduced and combined to improve the performance of the soft sensor model.

REFERENCES

[1] Y. Wang, Q. Jiang, B. Li, and L. Cui, "Joint-individual monitoring of parallel-running batch processes based on MCCA," *IEEE Access*, vol. 6, pp. 13005–13014, 2018.

[2] J. Zhu, Y. Yao, and F. Gao, "Multiphase two-dimensional time-slice dynamic system for batch process monitoring," *J. Process Control*, vol. 85, pp. 184–198, Jan. 2020.

[3] F. Shen, J. Zheng, L. Ye, and D. Gu, "Quality-relevant monitoring of batch processes based on stochastic programming with multiple output modes," *Processes*, vol. 8, no. 2, p. 164, Feb. 2020.

[4] R. Guo and N. Zhang, "A process monitoring scheme for uneven-duration batch process based on sequential moving principal component analysis," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 2, pp. 583–592, Mar. 2020.

[5] R. Wang, T. F. Edgar, M. Baldea, M. Nixon, W. Wojsznis, and R. Dunia, "A geometric method for batch data visualization, process monitoring and fault detection," *J. Process Control*, vol. 67, pp. 197–205, Jul. 2018.

[6] K. Peng, Q. Li, K. Zhang, and J. Dong, "Quality-related process monitoring for dynamic non-Gaussian batch process with multi-phase using a new data-driven method," *Neurocomputing*, vol. 214, pp. 317–328, Nov. 2016.

[7] 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.

[8] 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.

[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. Gajjar and A. Palazoglu, "A data-driven multidimensional visualization technique for process fault detection and diagnosis," *Chemometric Intell. Lab. Syst.*, vol. 154, pp. 122–136, May 2016.

[11] Y. A. Siddiqui, A.-W.-A. Saif, L. Cheded, M. Elshafei, and A. Rahim, "Integration of multivariate statistical process control and engineering process control: A novel framework," *Int. J. Adv. Manuf. Technol.*, vol. 78, nos. 1–4, pp. 259–268, Apr. 2015.

[12] J. Huang and X. Yan, "Quality-driven principal component analysis combined with kernel least squares for multivariate statistical process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2688–2695, Nov. 2019.

[13] Q. Jiang, X. Yan, H. Yi, and F. Gao, "Data-driven batch-end quality modeling and monitoring based on optimized sparse partial least squares," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4098–4107, May 2020.

[14] F. Chu, X. Cheng, R. Jia, F. Wang, and M. Lei, "Final quality prediction method for new batch processes based on improved JYKPLS process transfer model," *Chemometric Intell. Lab. Syst.*, vol. 183, pp. 1–10, Dec. 2018.

[15] L. Luo, S. Bao, J. Mao, and D. Tang, "Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes," *Chemometric Intell. Lab. Syst.*, vol. 150, pp. 9–22, Jan. 2016.

[16] V. Gopakumar, S. Tiwari, and I. Rahman, "A deep learning based data driven soft sensor for bioprocesses," *Biochem. Eng. J.*, vol. 136, pp. 28–39, Aug. 2018.

[17] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2018.

[18] K. Wang, B. Gopaluni, J. Chen, and Z. Song, "Deep learning of complex batch process data and its application on quality prediction," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7233–7242, 2018.

[19] W. Yan, D. Tang, and Y. Lin, "A data-driven soft sensor modeling method based on deep learning and its application," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4237–4245, May 2017.

[20] Y. Goldberg, "Neural network methods for natural language processing," *Synth. Lectures Human Lang. Technol.*, vol. 10, no. 1, pp. 1–309, Apr. 2017.

[21] T. H. Wen, M. Gasic, N. Mrksic, P. H. Su, D. Vandyke, and S. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," Aug. 2015, *arXiv:1508.01745*. [Online]. Available: https://arxiv.org/abs/1508.01745

[22] K. Espinoza, D. L. Valera, J. A. Torres, A. López, and F. D. Molina-Aiz, "Combination of image processing and artificial neural networks as a novel approach for the identification of bemisia tabaci and frankliniella occidentalis on sticky traps in greenhouse agriculture," *Comput. Electron. Agricult.*, vol. 127, pp. 495–505, Sep. 2016.

[23] M. Han, W. Chen, and A. D. Moges, "Fast image captioning using LSTM," *Cluster Comput.*, vol. 4, pp. 1–13, Aug. 2018.

[24] Y. Liu, Q. Wu, L. Tang, and H. Shi, "Gaze-assisted multi-stream deep neural network for action recognition," *IEEE Access*, vol. 5, pp. 19432–19441, 2017.

[25] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, "Hierarchical quality-relevant feature representation for soft sensor modeling: A novel deep learning strategy," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3721–3730, Jun. 2020.

[26] 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.

[27] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "A layer-wise data augmentation strategy for deep learning networks and its soft sensor application in an industrial hydrocracking process," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 13, 2020, doi: 10.1109/TNNLS.2019.2951708.

[28] X. Yuan, C. Ou, and Y. Wang, "Development of nvw-saes with nonlinear correlation metrics for quality-relevant feature learning in process data modeling," *Meas. Sci. Technol.*, vol. 32, no. 1, 2020, Art. no. 015006.

[29] G. Kataria and K. Singh, "Recurrent neural network based soft sensor for monitoring and controlling a reactive distillation column," *Chem. Product Process Model.*, vol. 13, no. 3, Sep. 2018, Art. no. 20170044.

[30] O. Obst, "Distributed fault detection in sensor networks using a recurrent neural network," *Neural Process. Lett.*, vol. 40, no. 3, pp. 261–273, Dec. 2014.

[31] X. Yuan, L. Li, Y. Wang, C. Yang, and W. Gui, "Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network," *Can. J. Chem. Eng.*, vol. 98, no. 6, pp. 1377–1389, Jun. 2020.

[32] F. Shen, J. Zheng, L. Ye, and X. Ma, "LSTM soft sensor development of batch processes with multivariate trajectory-based ensemble just-in-time learning," *IEEE Access*, vol. 8, pp. 73855–73864, 2020.

[33] X. Yuan, Z. Ge, B. Huang, Z. Song, and Y. Wang, "Semisupervised JITL framework for nonlinear industrial soft sensing based on locally semisupervised weighted PCR," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 532–541, Apr. 2017.

[34] F. Shen, Z. Ge, and Z. Song, "Multivariate trajectory-based local monitoring method for multiphase batch processes," *Ind. Eng. Chem. Res.*, vol. 54, no. 4, pp. 1313–1325, Jan. 2015.

[35] N. A. H. Haldar, F. A. Khan, A. Ali, and H. Abbas, "Arrhythmia classification using mahalanobis distance based improved fuzzy C-Means clustering for mobile health monitoring systems," *Neurocomputing*, vol. 220, pp. 221–235, Jan. 2017.

[36] X. Yuan, L. Li, and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3168–3176, May 2020.

[37] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1363–1374, Jun. 2016.

[38] Y. Si, Z. Chen, J. Sun, D. Zhang, and P. Qian, "A data-driven fault detection framework using mahalanobis distance based dynamic time warping," *IEEE Access*, vol. 8, pp. 108359–108370, 2020.

[39] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–23.

[40] L. Ye, H. Guan, X. Yuan, and X. Ma, "Run-to-run optimization of batch processes with self-optimizing control strategy," *Can. J. Chem. Eng.*, vol. 95, no. 4, pp. 724–736, Apr. 2017.

[41] G. Birol, C. Ündey, and A. Çinar, "A modular simulation package for fed-batch fermentation: Penicillin production," *Comput. Chem. Eng.*, vol. 26, no. 11, pp. 1553–1565, Nov. 2002.

[42] L. Ye and J. Cheng, "Simulator of penicillin fermentation process in MATLAB/simulink environment," *J. Syst. Simul.*, vol. 27, no. 3, pp. 515–520, Mar. 2015.

JIAQI ZHENG received the B.Eng. degree in automation from the Beijing University of Chemical Technology, Beijing, China, in 2015, and the M.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2018.

Since 2019, she has been with the School of Mechanical Engineering and Automation, College of Science and Technology, Ningbo University, Ningbo, China, where she is currently an Assistant Researcher. Her research interests include statistical process control, dynamic process monitoring, and their applications in process systems.

FEIFAN SHEN received the B.Eng. degree in electronic information technology and instruments and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2011 and 2016, respectively.

Since 2017, he has been with the School of Information Science and Engineering, NingboTech University, Ningbo, China, where he is currently a Lecturer. His research interests include data-driven modeling, batch process control, process monitoring, soft sensor, and their applications in process systems.

LINGJIAN YE (Member, IEEE) received the B.Eng. degree in biochemical engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively.

Since 2021, he has been with the School of Engineering, Huzhou University, Huzhou, China, where he is currently a Professor. His research interests include control structure design, plant-wide control, real-time optimization, and their applications in process systems.