This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

Deep Learning of Complex Batch Process Data and Its Application on Quality Prediction

Kai Wang, Bhushan Gopaluni, Junghui Chen and Zhihuan Song

***Abstract*—Batch process quality prediction is an important application in manufacturing and chemical industries. The complexity of batch processes is characterized by multiphase, nonlinearity, dynamics and uneven durations so that modeling of these batch processes is rather difficult. Moreover, there are other challenges in face of quality prediction. Specifically, the process trajectories over the whole running duration potentially make specific contributions to the final targets so that the prediction issue embraces tremendously high-dimensional inputs but very low-dimensional outputs. This means that the prediction suffers from a severe dimensional imbalance between inputs and outputs. Motivated by these difficulties, this paper proposes a new deep learning-based framework for complex feature representative and quality prediction. Long short-term memory is used to extract comprehensive quality-relevant hidden features from a long-time sequence in each phase, significantly reducing the predictor dimensions. And these features from different phases are further integrated and compressed by a stacked auto-encoder. A practical industrial example testifies to the efficacy of the proposed framework.**

***Index Terms* — batch process, quality prediction, long short-term memory, stacked auto-encoder.**

I. Introduction

Batch processes play an important role in producing low volume and high value-added products, such as polymers, semiconductors, chemicals, and pharmaceuticals. Because of the complexity of the processes, unforeseen disturbances, equipment faults, and changed raw material components, the product quality at the termination of one batch can be far different from the specifications [1, 2]. Moreover, batch processes lack online measurements of quality variables. To obtain the quality measurements takes a long time after one batch is ended, so a well-constructed prediction model is invaluable for tasks such as quality control [2, 3] and quality-relevant batch monitoring [4].

Several research papers have been done to develop quality prediction models for batch processes. Multiway partial least squares (MPLS) [5] and its different extensions [1, 6] are the most widely used batch quality prediction techniques. As MPLS-based methods consider the overall batch duration and build a global model for quality prediction, they ignore the multistage or multiphase feature of batches with staged operations [7]. A separate model of each phase is able to adapt itself to the multiphase batch process [6, 8-10]. Most of the methods have assumed the data are independent and identically distributed (i.i.d.) within a phase so that a static linear statistical model can be used to delineate local-phase characteristics. From this perspective, a phase refers to samples with the identical statistical properties in a continuous time interval. But this rationale is not necessarily beneficial to modeling a quality prediction model when it comes to significant nonlinear dynamic correlations. Typically, the particle filter is used to estimate the parameters for nonlinear state space models in batch processes [11]. But the nonlinear model structure is required in advance. In addition, Kernel-based multiway non-Gaussian latent subspace projection approach was developed for process monitoring [12], but it did not consider multiphase batch processes. For the purpose of improving the prediction performance, nonlinear descriptions for phase behaviors and nonlinear regression modeling have received lots of attention, represented by support vector regression [13, 14], relevant vector machine [15, 16] and Gaussian process regression [17]. However, they are limited in their shallow feature known as single hidden layer features. They are effective in dealing with simple nonlinearities but they may be insufficient when dealing with complex processes. Deep learning recently received a lot of attention owing to its deep structure and deep feature extraction. Especially, since Hinton et al. proposed the greedy layer-wise unsupervised pre-training and supervised fine-tuning technique [18], deep learning has succeeded in many applications because the vanishing and exploding gradient problems of deep network backpropagation have been effectively solved.

So far many techniques associated with deep learning which are known as soft sensors in industrial systems have been applied to important indices prediction [19-22]. Typically, Shang et al. first exploited deep belief network to build soft sensors for a crude distillation unit [19]. Later, an integration of a de-noising auto-encoder with a neural network was utilized to improve the prediction performance and the robustness of soft sensors [20]. Yao et al. used a hierarchical extreme learning machine for semi-supervised soft sensors [21]. Recently, to enhance the function of deep learning in soft sensors, the variable-wise weighted stacked auto-encoder was proposed. It selected the input variables which are the most relevant to the output variables [22]. However, the weight coefficient is simply determined by Pearson correlation, which cannot effectively delineate nonlinear correlation. Besides, these soft sensors were developed for continuous processes only. Because many specific characteristics of batch processes are quite different from those of continuous processes, these existing deep approaches of soft sensors in continuous processes cannot be directly applied to the quality prediction of batch

This work is supported by National Nature Science Foundation of China under Grant 61573308, National Science and Technology Infrastructure Program of the Ministry of Science and Technology of China under Grant 2013BAC03B08, Ministry of Science and Technology, Taiwan, ROC under Grant MOST 106-2221-E-033-060-MY3, and China Scholarship Council (*Corresponding authors: Junghui Chen and Zhihuan Song.*)

K. Wang and Z. Song are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: 11432023@zju.edu.cn (K. Wang), songzhihuan@zju.edu.cn (Z. Song)).

R. B. Gopaluni is with the Department of Chemical and Biological Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: bhushan.gopaluni@ubc.ca)

J. Chen is with Department of Chemical Engineering, Chung-Yuan Christian University, Chungli, Taoyuan 32023, Taiwan, ROC (e-mail: jason@wavenet.cycu.edu.tw).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

processes. The main challenges of the quality prediction of batch processes are as follows:

(1) A single model is not applicable to multiphase features of batch processes mentioned above. That means multiple nonlinear deep networks are required for batch processes.

(2) There is an extreme imbalance between the high dimensional input space and the low dimensional output space because the overall durations in process variables are potential predictors for the terminal qualities. Hence, implementing an effective dimension reduction and simultaneously guaranteeing desired prediction performance is crucial.

(3) Unlike continuous processes, in which long-term steady state and short-term transient state appear alternately, batch processes are often dominated by continuously varying dynamics in each phase because of continuous physical operations and complex chemical reaction. Therefore, complex dynamic behavior should be paid lots of attention to.

(4) The uneven-length duration between batches should be aligned into the same length before these existing methods are applied. Even though many methods, such as indicator variable based alignment [23], correlation optimization warping (COW) [24] and dynamic time warping (DTW) [25], were proposed to synchronize the uneven-length batches, the drawbacks and application restrictions are also apparent [26]. That implies using a prediction model to directly deal with the uneven-length issue is promising.

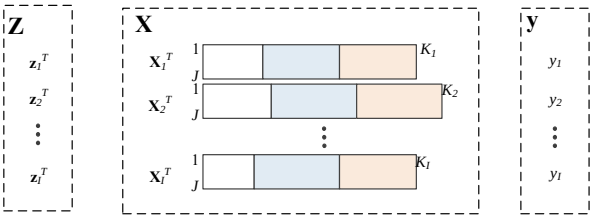

Fig. 1 Schematic of the data structure of multiphase batch processes

Because of the limitations of the existing methods, a new framework based on deep learning feature extraction is proposed to solve these important challenges in batch process quality prediction. In this paper, a special recurrent neural network (RNN), known as long short-term memory (LSTM), [27, 28] is utilized to extract phase features relevant to quality variables. LSTM inherits the recursive structure of RNN for time sequences so that the data of the uneven-length batches can be directly inputted into the network. Simultaneously, LSTM uses special units in hidden layers to remember some inputs at important time instances which are strongly relevant to the outputs. Moreover, the dimension would be sharply reduced by compressing the overall sequences in each phase into the hidden variables at the end of the phase. Then, these features represented by LSTM in each phase are concatenated and instilled into a stacked auto-encoder (SAE), as each phase is indispensable in predicting final targets. By layer-wise pre-training, the dimension-reduced, deep features can be obtained with SAE. And the supervised refining is able to adjust these features toward the direction of the quality prediction. In our proposed strategy, LSTM together with SAE produces an accurate prediction model. The "deep" means the original inputs are mapped into hidden features through the hierarchical structure defined by LSTM cascaded with SAE. The remaining parts are organized as follows: Section II gives a description of batch process data. Section III illustrates how to use LSTM to extract phase features. Then in Section IV, deep extraction of these phase features is introduced using SAE. An industrial example is tested to evaluate and compare the proposed method with other methods in Section V. And the final section draws conclusions.

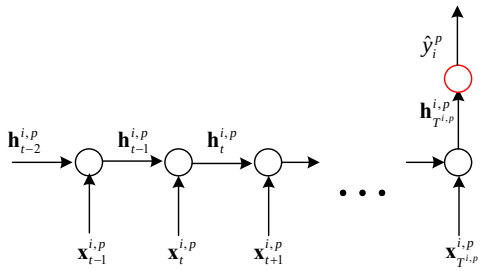

Fig. 2 Endpoint output structure of RNN

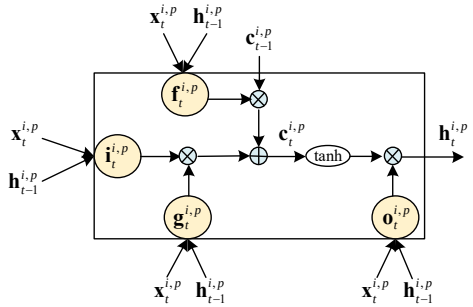

Fig. 3 Schematic of LSTM cell

II. Multiphase Batch Data Description

Data in batch processes are generally categorized into three types: initial conditions, online measured process trajectories, and the endpoint qualities. Initial conditions generally refer to the properties of raw materials and feedstocks that can be measured before a batch starts up. Let the matrix $\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_I \end{bmatrix}^T \in R^{I \times M}$ denote M initial conditions of I batches. Online measured process trajectories are the process measured variables which are measured at the regular time interval varying from a second to several minutes and the variables are provided for control systems to adjust manipulated variables and maintain the desired operation trajectories. The dataset for the measured process trajectories $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_I\}$ records I trajectories of J process variables and each batch, denoted as $\mathbf{X}_i \in R^{K_i \times J}$, has an unfixed batch length K_i. Regarding endpoint qualities, the one-dimensional quality variable is considered without loss of generality in this paper. If there are more than one quality variables, a separate prediction model can be built for each kind of quality. The corresponding endpoint quality dataset is represented by a vector $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_I \end{bmatrix}^T$. In multiphase batch processes, batch trajectory comprises several sequentially divisible phases with different characteristics. Suppose the phase in each batch has been partitioned in advance [7-9]; Fig. 1 shows the batch data structure with one kind of color

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

representing a specific phase in each batch. Hence, process variable trajectories in the *i*-th batch can be further divided into phase data blocks, i.e., $\mathbf{X}_i = \left\{\mathbf{X}^{i,1}, \mathbf{X}^{i,2}, \cdots, \mathbf{X}^{i,P}\right\}$ where *P* represents the number of phases. And each sample in the *p*-th phase of the *i*-th batch is denoted as $\mathbf{x}_t^{i,p}, t = 1,2,...,T^{i,p}$, where $T^{i,p}$ is the corresponding phase length and $\sum_{p=1}^{P} T^{i,p} = K_i$. In Fig. 1, the length of a phase in one batch is not necessarily identical with its counterparts in other batches because the operation duration in each stage can vary from batch to batch. One wishes to understand how the quality variable is influenced by the batch evolution and how different phases contribute to the final target.

III. LSTM Based Phase-wise Feature Representation

The terminal target is jointly influenced by initial conditions and process trajectories in different phases in process trajectories. A complete predictor consists of both different phases and initial conditions. There would be a very high-dimensional predictor in each phase and complex nonlinear dynamic correlations generally exist in the phase data. The final product quality depends on different initial conditions and a long time sequence of different phases. Hence, phase feature extraction would summarize the most quality-relevant and dimension-reduced features by modeling a nonlinear dynamic behavior contained in each phase.

In order to solve these challenges, an endpoint prediction structure of RNN in each phase (*p*) is adopted (Fig. 2). As one can see, there is a time-unfolded RNN driven by external inputs $\mathbf{x}_t^{i,p}, t = 1,2,...,T^{i,p}$. Each black node in Fig. 2 denotes an RNN cell, describing how the hidden states $\mathbf{h}_t^{i,p} \in R^{n_p}$, where n_p is the number of hidden units, are updated based on the previous hidden states $\mathbf{h}_{t-1}^{i,p}$ and the current inputs $\mathbf{x}_t^{i,p}$; i.e.,

$$\mathbf{h}_t^{i,p} = f\left(\mathbf{h}_{t-1}^{i,p}, \mathbf{x}_t^{i,p}\right) \tag{1}$$

where f is a nonlinear function to approximate the nonlinear dynamics. Because each phase is indispensable for a successful batch, they make different contributions to the final target. To learn the quality-relevant features, it is important to use the quality variables to train the RNN in a supervised way, i.e., there is an output layer in the RNN terminal, marked in a red node in Fig. 2, to output the predict $\hat{y}_i^p$ with respect to this phase (*p*) given by

$$\hat{y}_i^p = g\left(\mathbf{h}_{T^{i,p}}^{i,p}\right) \tag{2}$$

where g is an affine function in the prediction issue. From Eq.(2), the terminal hidden states $\mathbf{h}_{T^{i,p}}^{i,p}$ summarize the phase features relevant to the quality because $\mathbf{h}_{T^{i,p}}^{i,p}$ contains the information of the whole phase sequence and it is also a link from the phase sequence to the final target through the RNN structure. It can be seen that $\mathbf{h}_{T^{i,p}}^{i,p}$ is the learned quality-relevant feature in the corresponding phase. To optimize the network, the loss function for training this type of RNN is given by

$$L_p = \frac{1}{2I}\sum_{i=1}^{I}\left(y_i - \hat{y}_i^p\right)^2, \quad p = 1,\cdots,P \tag{3}$$

An ordinary RNN often uses a squashing function on the affine transformation of inputs and hidden variables to approximate the representation in Eq.(1). The optimization of an ordinary RNN using back-propagation through the time algorithm will cause gradient vanishing or exploding when the time sequence is very long. That is because one iteration is finished only if the gradient has been propagated back from the endpoint to the start point in each phase. Thus, the weights and biases close to the end point of each phase can be adjusted. It is the so-called gradient vanishing or exploding occurrence. It causes the loss of a lot of previous useful information. To improve the ability to learn long-term memory, LSTM reforms the ordinary RNN cell in a feasible memorizing and forgetting way, alleviating the vanishing and exploding gradient problems. The schematic diagram of an LSTM cell is shown in Fig. 3. The key idea behind LSTM is that the new defined cell states $\mathbf{c}_t^{i,p}$ adaptively memorize important information and forget the redundant information. Specifically, LSTM uses an input gate $\mathbf{i}_t^{i,p}$, a forget gate $\mathbf{f}_t^{i,p}$ and an output gate $\mathbf{o}_t^{i,p}$ to control the flow of information. At each time instance, LSTM determines what old information should be discarded through the following forget gate leaking some information in the previous cell states $\mathbf{c}_{t-1}^{i,p}$

$$\mathbf{f}_t^{i,p} = \sigma\left(\mathbf{W}_f^p \mathbf{x}_t^{i,p} + \mathbf{U}_f^p \mathbf{h}_{t-1}^{i,p} + \mathbf{b}_f^p\right) \tag{4}$$

where σ is an element-wise sigmoid function, $\mathbf{W}_f^p$ and $\mathbf{U}_f^p$ are weight matrices related to forget gates and $\mathbf{b}_f^p$ are the corresponding biases in this phase. Since the range of σ is from zero to one, a lot of previous information will be removed when most of the elements in $\mathbf{f}_t^{i,p}$ are close to zero.

Simultaneously, it needs to be determined what new information should be stored. The new information is represented by $\mathbf{g}_t^{i,p}$, which is an activation of the current value $\mathbf{x}_t^{i,p}$ and the hidden variable $\mathbf{h}_{t-1}^{i,p}$ given by,

$$\mathbf{g}_t^{i,p} = tanh\left(\mathbf{W}_g^p \mathbf{x}_t^{i,p} + \mathbf{U}_g^p \mathbf{h}_{t-1}^{i,p} + \mathbf{b}_g^p\right) \tag{5}$$

where *tanh* denotes the hyperbolic tangent function. Then, an input gate $\mathbf{i}_t^{i,p}$ is designed as

$$\mathbf{i}_t^{i,p} = \sigma\left(\mathbf{W}_i^p \mathbf{x}_t^{i,p} + \mathbf{U}_i^p \mathbf{h}_{t-1}^{i,p} + \mathbf{b}_i^p\right) \tag{6}$$

to keep the output range within (0,1) using a sigmoid function for filtering .. (Fig. 3). In Eqs.(5) and (6), there are corresponding weights and biases to be estimated in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

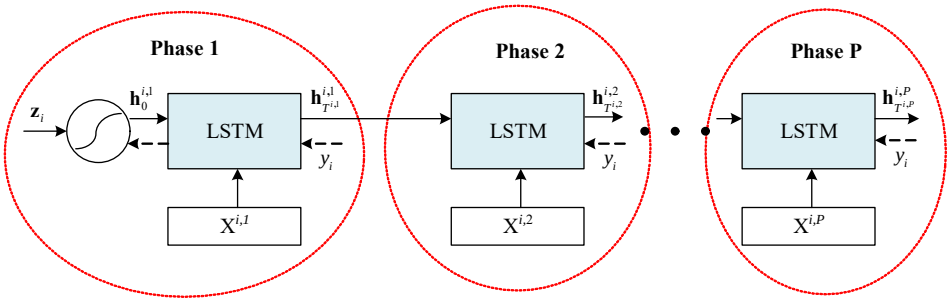

Fig. 4 The training structure of multiphase processes by LSTM with initial conditions.

this gate. Then the new cell states $\mathbf{c}_t^{i,p}$ are updated by the partial forgetting and partial storing information.

$$\mathbf{c}_t^{i,p} = \mathbf{f}_t^{i,p} \otimes \mathbf{c}_{t-1}^{i,p} + \mathbf{g}_t^{i,p} \otimes \mathbf{i}_t^{i,p} \tag{7}$$

where $\otimes$ denotes the element-wise product. Lastly, the new hidden states $\mathbf{h}_t^{i,p}$ are obtained by the output gate $\mathbf{o}_t^{i,p}$ and the new cell states $\mathbf{c}_t^{i,p}$

$$\mathbf{o}_t^{i,p} = \sigma\left(\mathbf{W}_o^p \mathbf{x}_t^{i,p} + \mathbf{U}_o^p \mathbf{h}_{t-1}^{i,p} + \mathbf{b}_o^p\right) \tag{8}$$

$$\mathbf{h}_t^{i,p} = \mathbf{o}_t^{i,p} \otimes tanh\left(\mathbf{c}_t^{i,p}\right) \tag{9}$$

Unlike the general variable selection, which can just extract information of a superficial or shallow layer, LSTM effectively filters information of the feature layer based on the forgetting and storing mechanism. By this means, there will be more chance to discover nonlinear quality-relevant features. Moreover, the recurrent network structure is able to compress a long time sequence into a comprehensive feature $\mathbf{h}_{T^{i,p}}^{i,p}$, implementing a sharp predictor dimension reduction. Besides, LSTM just builds a structured dynamic model rather than any statistical models so that the phase length is not required to be identical in different batches. One potential introduced by LSTM is the simplification of the phase division. As it has been mentioned, sequential statistical models along the time direction generally need to be constructed to analyze the variation of batch trajectories. The samples with close statistical properties in a continuous time interval will be grouped into a phase. This kind of phase division procedure has the risk of causing an undue partition and produces overloaded sub-models than the true physical operation stages for obtaining a sufficient approximation to a nonlinear batch running. In contrast, the universal approximation theorem [29] theoretically implies that LSTM has a high capacity to approximate any complex nonlinear continuous function governing a nonlinear dynamic sequence. From this perspective, the phase division is reduced to find out a few switched process operations which possibly produce discontinuous batch trajectories. These operations are easy to obtain by finding out some indicator variables with piecewise linear characteristics. For the case that the indicator variables are unavailable, one way with LSTM for locating phase switching points is to extract quality-relevant hidden features for the whole length and measure the difference of the hidden features between two adjacent times. The significant changes of the hidden features can be the indicators of phase switching.

Taking LSTM as the core, a complete phase-wise feature representation framework is drawn in Fig. 4. To achieve the optimal generalized performance, each phase is governed by a specific LSTM with the best match to the phase characteristics. Each LSTM in a specific phase is separately optimized using the corresponding phase dataset. Before the optimization procedure starts, each LSTM needs to set up initial hidden states to configure the network. If the initial conditions are unavailable, the initial hidden states can be tentatively assumed to be zero. Otherwise, the information in the initial conditions will be directly related to the initial hidden states $\mathbf{h}_0^{i,1}$ in the first phase. It can be obtained by activating the initial conditions $\mathbf{z}_i$ shown in Fig. 4, i.e.,

$$\mathbf{h}_0^{i,1} = tanh\left(\mathbf{W}_0 \mathbf{z}_i + \mathbf{b}_0\right) \tag{10}$$

where the parameters $\mathbf{W}_0$ and $\mathbf{b}_0$ as well as the parameters in LSTM are trained for maximizing the predictability of the quality by gradient backpropagation supervised by the quality variable shown in Fig. 4. For the other phases, the endpoint hidden states in the last phase can be fed into the next phase as the initial hidden states to represent the time link between two adjacent phases. When LSTM in the *p*-th phase is trained, the endpoint hidden states in the last phase are regarded as a known condition so that the cause-and-effect relationship is a one-way path, not a two-way path. The downstream info is always caused by info of the upper stream in the whole batch operation.

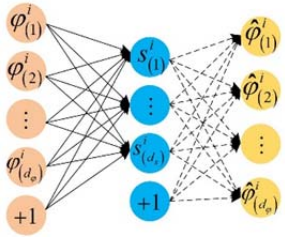

Fig. 5 The structure of the single layer auto-encoder.

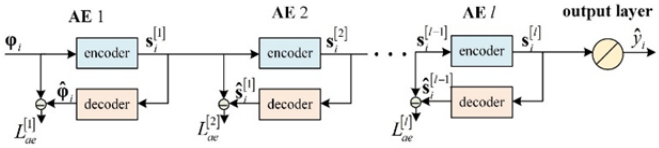

Fig. 6 The structure of SAE with an output.

IV. Deep Feature Representation And Quality Prediction

A. *Preliminaries of SAE*

The phase-wise training with LSTMs helps to extract complex and abstract quality-relevant features from different phases. These features are extracted from different phases separately while these features would concurrently influence the final target in a complex fashion. On the other hand, the dimension of accumulated features over all the phases,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

which is $\sum_{p=1}^{P} n_p$, is still sizable for the quality prediction application. These concerns motivate the phase features to be further deeply and interactively integrated and compressed.

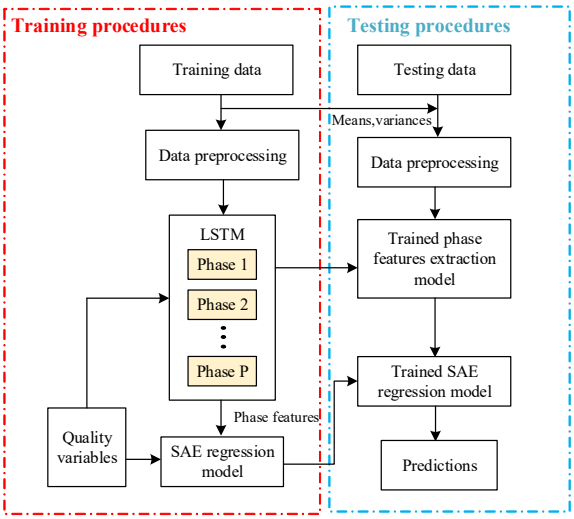

Fig. 7 The flow chart of quality prediction for a batch process.

SAE consists of a multi-layer auto-encoder. The network depth allows SAE to extract deep features and express complex input-output relationships hierarchically. A basic auto-encoder (AE) is a single hidden layer network composed of an encoder and a decoder illustrated in Fig. 5. Suppose the input vector of an AE is the concatenated phase features as $\boldsymbol{\varphi}_i = \left[\left(\mathbf{h}_{T^{i,1}}^{i,1}\right)^T \quad \left(\mathbf{h}_{T^{i,2}}^{i,2}\right)^T \quad \cdots \quad \left(\mathbf{h}_{T^{i,P}}^{i,P}\right)^T \right]^T \in R^{d_\varphi}$ and the hidden layer is denoted as $\mathbf{s}_i = \left[s_{(1)}^i \quad s_{(2)}^i \quad \cdots \quad s_{(d_s)}^i \right]^T$. Generally, the dimension of the hidden layer d_s is less than that of the input layer d_φ for the effective feature extraction and dimension reduction because a larger hidden space dimension will possibly learn an identity function for an autoencoder, causing overfitting. The encoder maps the input vector into the hidden features by a nonlinear activation function. Thus, there is

$$\mathbf{s}_i = f_{ae}\left(\mathbf{W}_{ae}\boldsymbol{\varphi}_i + \mathbf{b}_{ae}\right) \tag{11}$$

where f_{ae} is the selected activation function in the encoder layer while $\mathbf{W}_{ae}$ and $\mathbf{b}_{ae}$ are the weights and the biases for this encoder. In this paper, the activation function is set to be *tanh* function in the encoder. And the decoder reconstructs the hidden variables $\mathbf{s}$ into the original input space as follows.

$$\hat{\boldsymbol{\varphi}}_i = \tilde{f}_{ae}\left(\tilde{\mathbf{W}}_{ae}\mathbf{s}_i + \tilde{\mathbf{b}}_{ae}\right) \tag{12}$$

where $\tilde{f}_{ae}$ is the selected activation function in the decoder layer while $\tilde{\mathbf{W}}_{ae}$ and $\tilde{\mathbf{b}}_{ae}$ are the corresponding weights and biases. Since the objective is to reconstruct $\boldsymbol{\varphi}_i$, $\tilde{f}_{ae}$ is generally an element-wise linear unit. For *I* batches, the loss function can minimize the reconstruction error, given by

$$L_{ae} = \frac{1}{2I}\sum_{i=1}^{I}\left\|\boldsymbol{\varphi}_i - \hat{\boldsymbol{\varphi}}_i\right\|_2 \tag{13}$$

The gradient backpropagation is used to optimize the parameters $\left\{\mathbf{W}_{ae}, \mathbf{b}_{ae}, \tilde{\mathbf{W}}_{ae}, \tilde{\mathbf{b}}_{ae}\right\}$. The structure of SAE for predictions is illustrated in Fig. 6. There are *l* cascaded AEs stacking hierarchically. The first AE accepts the original input variables $\boldsymbol{\varphi}_i$ and other AEs take the hidden representation in the previous AE as the encoder inputs. Each AE is firstly pre-training separately in an unsupervised layer-wise fashion. Then, the output layer is added to the top of SAE to fine-tune the weights and biases. In other words, the parameters in each AE obtained by pre-training are set up as the initial values in the fine-tuning step. The purpose of prediction is to guarantee the output scope to be the whole real number field, so the activation function in the output layer is chosen to be an affine function, given by $\hat{y}_i = \mathbf{w}_y^T \mathbf{s}_i^{[l]} + b_y$ where $\mathbf{w}_y$ and b_y are weights vector and bias in the output layer. The loss function $L = \frac{1}{2I}\sum_{i=1}^{I}\left(y_i - \hat{y}_i\right)^2$ can minimize the prediction error in the training data.

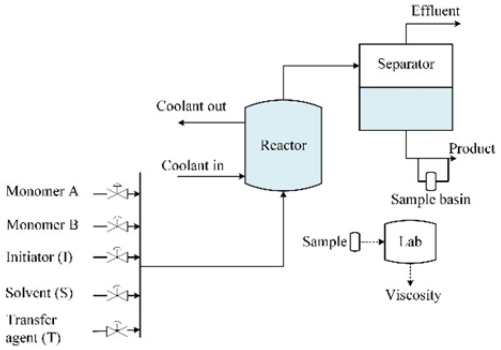

Fig. 8 The illustration of the MMA-VA process.

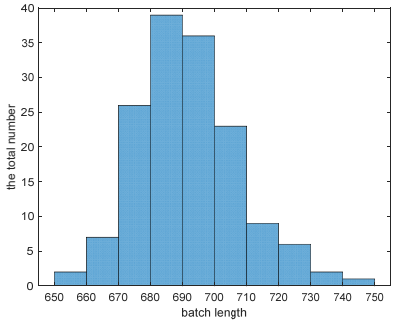

Fig. 9 The histogram of batch durations

B. SAE based quality prediction

Because SAE has a high capacity to learn abstract complex features, it is able to deeply integrate the phase features. Fig. 7 illustrates the whole framework of the quality prediction of multiphase batch processes. In the training stage, a feature representation model based on deep learning is designed. The detail in each step shown in Fig. 7 is as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

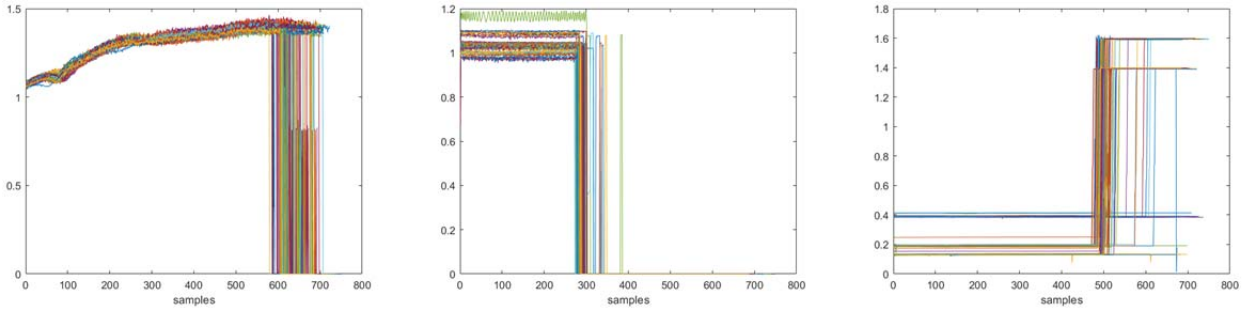

Fig. 10 The indicator variables in this example.

Step 1. Collect the training data from a specific batch process. Initial conditions, process variables and the quality in each batch are normalized using corresponding variable means and variable standard deviations. Divide each batch into several phases.

Step 2. Train the LSTM models for phase feature extraction shown in Fig. 4 in a phase-wise way supervised by the quality variable. The initial conditions are fed to the first phase through a single hidden layer with a compatible dimension. When the training is completed, collect the hidden endpoint states in each phase $\mathbf{h}_{T^{i,p}}^{i,p}$, $p = 1, 2, ..., P$.

Step 3. Let $\boldsymbol{\varphi}_i$ be the concatenated hidden endpoint states in each phase obtained in Step 2. Then the SAE regression model is trained with the unsupervised layer-wise pre-training and the supervised fine-tuning given by the quality variable.

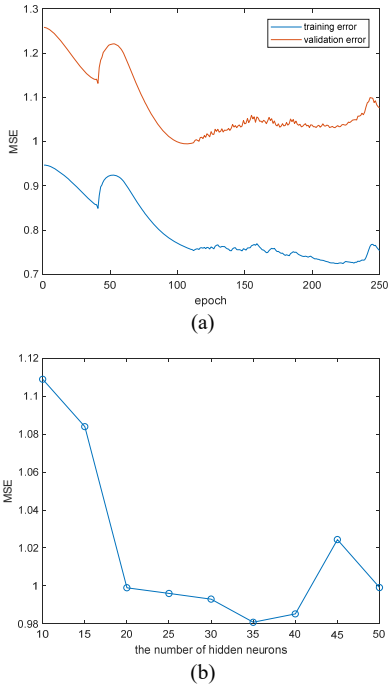

Fig. 11 The MSE of the first phase

The mean squared error (MSE) index is used to validate the prediction performance. It is defined as

$$MSE = \frac{1}{I-1}\sum_{i=1}^{I}\left(y_i - \hat{y}_i\right)^2 \qquad (14)$$

A smaller MSE indicates a better prediction result. Besides MSE, another commonly used index is R^2, defined as

$$R^2 = 1 - \sum_{i=1}^{I}\left(y_i - \hat{y}_i\right)^2 / \sum_{i=1}^{I}\left(y_i - \bar{y}\right)^2 \qquad (15)$$

where $\bar{y}$ is the mean of the true quality. R^2 is an index evaluating the reliability of the model because it can give information about how much of the total variance in the output variable data can be explained by the model. The model is more reliable when R^2 is closer to 1.

V. Industrial Applications

A practical chain growth copolymerization batch process of methyl methacrylate and vinyl acetate (MMA-VA) is tested in this paper. The MMA-VA copolymerization process consists of a well-mixed reactor followed by a product separator, both shown in Fig. 8. The feed to the reactor consists of the monomers (MMA and VA), the initiator (AIBN), the transfer agent (acetaldehyde) and the inhibitor (m-dinitrobenzene) dissolved in a solvent (benzene). A coolant is employed for the removal of heat released because of polymerization. The polymer product is separated from the unreacted hydrocarbons in a downstream separator. There are 11 process variables measured from different operating units. And one of the crucial indices tested in the lab is the intrinsic viscosity of the copolymer product. It varies for each grade of copolymer depending upon market requirements.

TABLE I. The optimal selection for hidden neuron numbers and epochs of the four phases

Phase No.	Minimum MSE	Optimal epoch	Optimal number of hidden neurons
1	0.9808	165	35
2	1.1487	23	15
3	1.0788	40	40
4	0.8909	141	10

In this industrial example, there are a total of 139 batches. About 90 batches of them are randomly chosen as the training dataset and the remaining are detracted into a validation dataset with 20 batches and a testing dataset with 29 batches. Among them, the validation dataset is used to determine suitable hyperparameters for network training. The 139 batches have different durations and the corresponding histogram with respect to batch duration distribution is shown in Fig. 9. The variable in each batch is firstly normalized by the sample mean and the sample variance calculated from the first batch, avoiding the inputs stuck in the saturation area of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

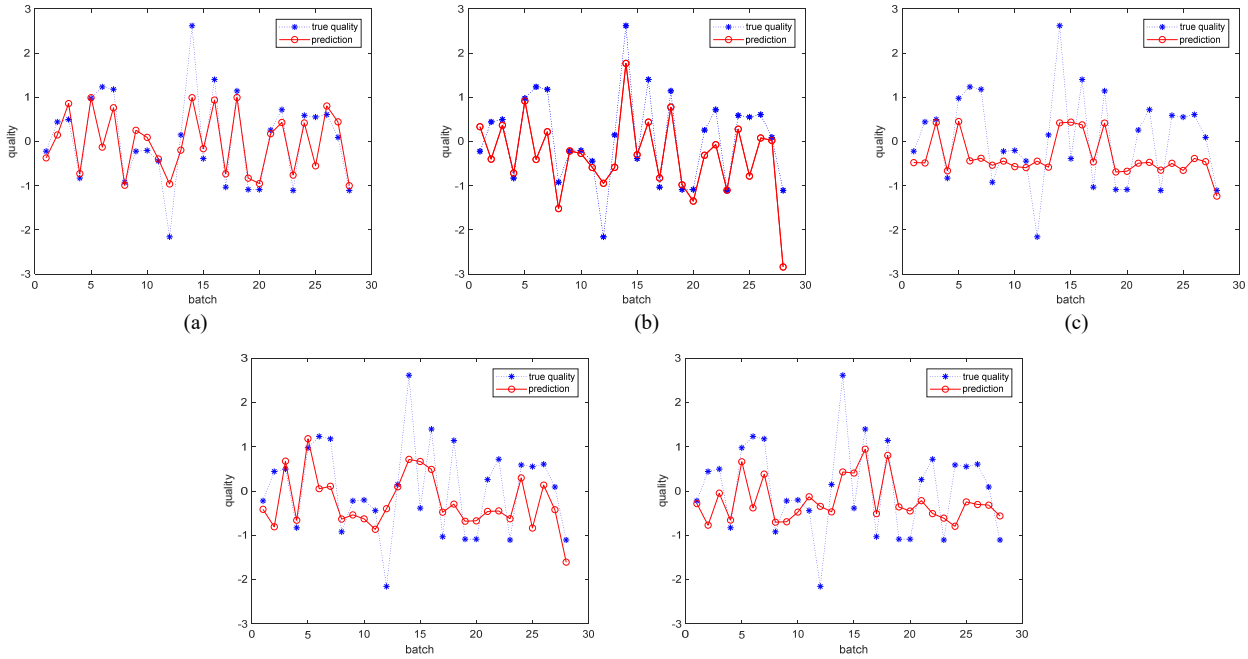

Fig. 12 Prediction results of the testing batch data. (a) LSTM-SAE; (b) LSTM-NN (c) RNN; (d) LSTM; (e) MPLS-RVM

the activation function. And this kind of preprocessing procedure is also well-suited to online learning by normalizing the newest batch with the first batch. Next, each batch should be divided into several phases. As the phase is divided because of different operations over a batch, the phase division is performed by locating the operation switching points in this example. Note that for a specific batch process, the operation mode and operating procedure are usually fixed for producing stable and identical products, so the number of phases among the batch data would be the same. In this application, there are several indicator variables which are piecewise stationarity or piecewise monotony so that they can be used to locate the phase switching points. The profiles of three selected variables are shown in Fig. 10. It can be clearly seen that the phase division points
for these three variables are separately located around k = 300, 500 and 500. Fig. 10 shows the true industrial data used in our case study. For enterprise's confidentiality, the variables shown in Fig. 10 are normalized and the variable names are hidden. For a specific batch, the first-order difference of these indicator variables is used to detect the concrete change points. Then, all the batches are divided into 4 phases along the time direction. For each phase, an LSTM-based supervised learning with the quality label is carried out. The stochastic mini-batch gradient descent algorithm with the momentum is used to search for the optimal parameters. There are 20 mini-batches, and the learning rate and the momentum are 0.03 and 0.9, respectively. There are two crucial hyperparameters to be determined for suppressing overfitting: the hidden neuron number, also known as the cell number in LSTM, and the number of epochs defined as updating the network parameters using all the training samples in the deep learning community. Both of the two hyperparameters are determined by the early stopping strategy based on MSE (the mean squared error) of the validation dataset. Specifically, the network is iteratively optimized untilthe error of the validation dataset has not been improved for a period of time. And the number of the optimal hidden neurons is determined by comparing MSE at the final epoch in the validation dataset.

Take the first phase with 25 hidden neurons, for example. Fig. 11(a) shows the progress of the training and validation performance indices during training. The training error continues to go down through the training process, but the minimum validation error occurs at the point, which corresponds to the training epoch 110. This means the iteration can stop early at this epoch given 25 hidden neurons. By incrementing or decreasing the hidden neuron number and conducting the early stopping strategy, the optimal number of hidden neurons is 35 at the first phase (Fig. 11(b)) because it has the smallest MSE on the validation dataset.

TABLE I further summarizes the selected optimal hidden neuron numbers in the four phases and the corresponding MSE on the validation dataset. Obtaining the suitable stopping epoch and the number of hidden neurons, the validation dataset will be merged into the training dataset to retrain the final network parameters for making full use of data. After condensing the data with respect to each phase into the corresponding terminal hidden states, All features extracted by LSTM in all phases are concatenated together and then are transferred into SAE. The SAE is used to reduce the dimension and extract features. So the level of SAE is uncorrelated with the number of phases. The structure of SAE is determined by the trial and error technique. Thus, the constructed SAE in this example consists of two AEs with 20 and 15 neurons, respectively. After these AEs are pre-trained in a layer-wise manner, they are further stacked into SAE and the output layer is added to the top of SAE for weight and bias refining and quality prediction. MSE and R^2 of the testing dataset are 0.32 and 0.69, respectively. The predicted quality and the true quality in each testing batch are shown in Fig. 12(a). In most batches, the predicted values fit the real values well. Some

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2880968, IEEE Transactions on Industrial Informatics

batches also have large prediction errors, probably because a new data pattern which is not similar to the training dataset is included in the testing dataset or the corresponding quality in this batch is close to an outlier.

TABLE II. MSE and R^2 of different methods

Approach	MSE	R^2
LSTM-SAE	0.32	0.69
LSTM-NN	0.54	0.47
RNN	0.86	0.15
LSTM	0.75	0.26
MPLS-RVM	0.79	0.28

To further testify the effectiveness of the new strategy, it is compared to other methods. Firstly, a phase-wise LSTM followed by a single hidden layer neural network (LSTM-NN) is trained. The number of hidden units is 20, which is the same as the first autoencoder in LSTM-SAE. Then, an ordinary RNN shown in Fig. 2 is trained by full lengths without any phase division to predict qualities. Likewise, without any phase division, an LSTM cell in place of the ordinary RNN cell is applied to carry out quality prediction. Note that the state transition structure in Fig. 2 does not require the same length in each batch. The original data can be directly used to train model parameters. For the LSTM network and the ordinary RNN network with full-length training, the parameters are initialized randomly. The learning rate of the learning algorithm is set as the same as that of the proposed scheme. In addition, a shallow model for nonlinearity quality prediction is also applied using MPLS for feature extraction in each phase and RVM for regressing these features into the ending quality[15]. The phase division points of this shallow model are the same as those of the proposed method. Note that the shallow model requires the same length for all the training batches in each phase. To satisfy such a requirement, DTW aligns those uneven batches for obtaining a compatible batch data. Fig.12 and Table II show the comparative results of the three methods with the proposed strategy. LSTM-NN produces a bit better prediction than the other comparative methods. That implies the importance of SAE in terms of extracting abstract representations. LSTM is better than RNN because LSTM has a stronger ability to learn a long-term memory. However, all of the three methods cannot be put into practice because the predictions in many batches contain significant prediction errors. In comparison, the prediction efficacy of LSTM-SAE has been significantly improved. The reason behind the improved performance is that the parameters governing the state transition are shared within a phase instead of the whole batch and LSTM-SAE has a higher model capacity to adapt to very complex batch processes. In this way, a more precise quality-relevant feature extraction is included for the challenging industrial example.

VI. Conclusions

This paper proposes a new strategy for batch process quality prediction based on deep learning, which means the deep hierarchical structure with LSTM cascaded by SAE is used to extract quality-relevant phase features. And then these phase features are concatenated together and are finally fed into an SAE network for learning a deep representation of batch trajectories. They are the comprehensive features for quality prediction. In this way, the new method has the following advantages.

- The endpoint prediction structure of LSTM and gates in LSTM help summarize a long phase sequence into several endpoint hidden states and bring about a large-scale reduction of predictor dimension.
- As a strong nonlinear dynamic model, LSTM makes phase division easier than linear statistical methods. In the framework of LSTM, the phase is determined by simply locating discontinuous points.
- Because of the recurrent structure, LSTM can directly deal with uneven-length sequences instead of using any sequence alignment methods to avoid the risk of distorting the original sequence structure.
- With the high capacity of SAE, LSTM-SAE is able to integrate phase features and further learn deep nonlinear features rather than simple or shallow features.

Based on these merits, LSTM-SAE can yield prediction results after effective modeling training; it can also be directly applied to real plants with little compromise on data preprocessing. The practical MMA-VA process has validated the effectiveness of the proposed framework. Since the proposed method makes a significant improvement in predicting the quality of complex batch processes, the quality-relevant process optimization, process monitoring and control are worthy of further study under the proposed framework.

References

[1] D. Wang, "Robust Data-Driven Modeling Approach for Real-Time Final Product Quality Prediction in Batch Process Operation," *IEEE Transactions on Industrial Informatics,* vol. 7, no. 2, pp. 371-377, 2011.

[2] B. Corbett, and P. Mhaskar, "Subspace identification for data-driven modeling and quality control of batch processes," *AIChE Journal*, vol. 62, no. 5, pp. 1581-1601, 2016.

[3] A. Mesbah, Z. K. Nagy, A. E. Huesman *et al.*, "Nonlinear model-based control of a semi-industrial batch crystallizer using a population balance modeling framework," *IEEE Transactions on Control Systems Technology,* vol. 20, no. 5, pp. 1188-1201, 2012.

[4] L. Luo, S. Bao, J. Mao *et al.*, "Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes," *Chemometrics and Intelligent Laboratory Systems,* vol. 150, pp. 9-22, 2016.

[5] P. Nomikos, and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and intelligent laboratory systems,* vol. 30, no. 1, pp. 97-108, 1995.

[6] S. Stubbs, J. Zhang, and J. Morris, "BioProcess performance monitoring using multiway interval partial least squares," *Computer Aided Chemical Engineering,* vol. 41, pp. 243-259, 2018.

[7] C. Undey, and A. Cinar, "Statistical monitoring of multistage, multiphase batch processes," *IEEE control systems,* vol. 22, no. 5, pp. 40-52, 2002.

[8] C. Zhao, F. Wang, Z. Mao *et al.*, "Quality prediction based on phase-specific average trajectory for batch processes," *AIChE Journal,* vol. 54, no. 3, pp. 693-705, 2008.

[9] Y. Qin, C. Zhao, X. Wang *et al.*, "Subspace decomposition and critical phase selection based cumulative quality analysis for multiphase batch processes," *Chemical Engineering Science,* vol. 166, pp. 130-143, 2017.

[10] W. Sun, Y. Meng, A. Palazoglu *et al.*, "A method for multiphase batch process monitoring based on auto phase identification," *Journal of Process Control,* vol. 21, no. 4, pp. 627-638, 2011.

[11] Z. Zhao, B. Huang, and F. Liu, "Parameter estimation in batch process using EM algorithm with particle filter," *Computers & Chemical Engineering,* vol. 57, pp. 159-172, 2013.

[12] J. Mori, and J. Yu, "Quality relevant nonlinear batch process performance monitoring using a kernel based multiway non-Gaussian latent subspace projection approach," *Journal of Process Control,* vol. 24, no. 1, pp. 57-71, 2014.

[13] J. Yu, "A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch

bioprocesses," *Computers & Chemical Engineering,* vol. 41, pp. 134-144, 2012.
[14] H. Jin, X. Chen, J. Yang *et al.*, "Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes," *Chemical Engineering Science,* vol. 131, pp. 282-303, 2015.
[15] Z. Ge, Z. Song, and F. Gao, "Nonlinear quality prediction for multiphase batch processes," *AIChE journal,* vol. 58, no. 6, pp. 1778-1787, 2012.
[16] J. Zhu, and F. Gao, "Improved Nonlinear Quality Estimation for Multiphase Batch Processes Based on Relevance Vector Machine with Neighborhood Component Variable Selection," *Industrial & Engineering Chemistry Research,* vol. 57, no. 2, pp. 666-676, 2018.
[17] L. Zhou, J. Chen, and Z. Song, "Recursive Gaussian process regression model for adaptive quality monitoring in batch processes," *Mathematical Problems in Engineering,* vol. 2015.
[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation,* vol. 18, no. 7, pp. 1527-1554, 2006.
[19] C. Shang, F. Yang, D. Huang *et al.*, "Data-driven soft sensor development based on deep learning technique," *Journal of Process Control,* vol. 24, no. 3, pp. 223-233, 2014.
[20] W. Yan, D. Tang, and Y. Lin, "A Data-Driven Soft Sensor Modeling Method Based on Deep Learning and its Application," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 5, pp. 4237-4245, 2017.
[21] L. Yao, and Z. Ge, "Deep Learning of Semisupervised Process Data With Hierarchical Extreme Learning Machine and Soft Sensor Application," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 2, pp. 1490-1498, 2018.
[22] X. Yuan, B. Huang, Y. Wang *et al.*, "Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Transactions on Industrial Informatics,* vol. 14, no. 7, pp. 3235-3243, 2018.
[23] D. Neogi, and C. E. Schlags, "Multivariate statistical analysis of an emulsion batch process," *Industrial & engineering chemistry research,* vol. 37, no. 10, pp. 3971-3979, 1998.
[24] M. Fransson, and S. Folestad, "Real-time alignment of batch process data using COW for on-line process monitoring," *Chemometrics and Intelligent Laboratory Systems,* vol. 84, no. 1, pp. 56-61, 2006.
[25] J. González-Martínez, A. Ferrer, and J. Westerhuis, "Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping," *Chemometrics and Intelligent Laboratory Systems,* vol. 105, no. 2, pp. 195-206, 2011.
[26] C. Zhao, S. Mo, F. Gao *et al.*, "Statistical analysis and online monitoring for handling multiphase batch processes with varying durations," *Journal of Process Control,* vol. 21, no. 6, pp. 817-829, 2011.
[27] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." In INTERSPEECH-2014, 338-342.
[28] X. Li, F. Duan, P. Loukopoulos *et al.*, "Canonical variable analysis and long short-term memory for fault diagnosis and performance estimation of a centrifugal compressor," *Control Engineering Practice,* vol. 72, pp. 177-191, 2018.
[29] T. Chen, and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks,* vol. 6, no. 4, pp. 911-917, 1995.

Kai Wang received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 2014. He is currently working toward the Ph.D. degree in the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.
He was a visiting doctoral student with the Department of Chemical and Biological Engineering at the University of British Columbia. His research interests include industrial data analytics, process monitoring and fault diagnosis.

Ratna Bhushan Gopaluni received the B.Tech degree in chemical engineering from the Indian Institute of Technology, Chennai, India, in 1997,and the Ph.D. degree in chemical engineering from the University of Alberta, Edmonton, AB, Canada, in 2003. He was an Engineering Consultant with Honeywell Process Solutions, North Vancouver, BC, Canada, from 2003 to 2005, during which he had designed and commissioned multivariable controllers in British Columbia's pulp and paper industry, and implemented numerous controller performance monitoring projects with Oil & Gas, BC, Canada, and other chemical industries. He is currently a Professor of the Department of Chemical and Biological Engineering with The University of British Columbia, Vancouver, BC, Canada. He specializes in data analytics for process industry and has authored over 90 refereed articles in reputed international journals and conferences.

Junghui Chen received his B.S. degree from the Department of Chemical Engineering, Chung-Yuan Christian University in 1982 and the M.S. degree from the Department of Chemical Engineering, National Taiwan University in 1984. Then he received Ph.D. from the Department of Chemical Engineering, University of Tennessee, Knoxville, U.S.A. in 1995.
Now he is a full Professor at the Chung-Yuan Christian University, Taiwan. His research interest is process system engineering, including process design for operability, nonlinear control, process monitoring and diagnosis, control loop performance assessment, batch control, model predictive control, data mining and analytics, and iterative learning design.

Zhihuan Song received the B.Eng. and M.Eng. degrees in Industrial Automation from Hefei University of Technology, Anhui, China, in 1983 and 1986, respectively, and the Ph.D. degree in Industrial Automation from Zhejiang University, Hangzhou, China, in 1997.
Since then, he has been with the Department of Control Science and Engineering, Zhejiang University, where he was at first a Postdoctoral Research Fellow, then an Associate Professor, and currently a Professor. His research interests include modeling and fault diagnosis of industrial processes, embedded control systems, and advanced process control technologies. He has published more than 200 papers in journals and conference proceedings.