Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

A mutual information-based Variational Autoencoder for robust JIT soft sensing with abnormal observations

Fan Guo, Biao Huang [*]

Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB T6G2G6, Canada

ARTICLE INFO

Keywords:
Variational autoencoder
Mutual information
Just-in-time learning
Outliers
Missing data

ABSTRACT

Considering industrial process with high-dimensional, intrinsic nonlinearities and possibly abnormal observations, a robust deep learning soft sensor model is developed under the just-in-time learning framework. As an unsupervised deep learning approach, Variational Autoencoder (VAE) has been successfully applied to soft sensing problems owing to its ability to describe the latent representations by probability distributions. In this work, to construct high performance soft sensor model, mutual information (MI) is first introduced for input variable selection. By further incorporating MI as weights on variable of the traditional VAE model, a MI-based output-relevant VAE is developed. For each new sample that arrives, by utilizing Symmetric Kullback-Leibler (SKL) divergence, its relevance with historical samples is determined. Based on the SKL divergence, the input samples that are most relevant to the query sample can be collected. The selected historical input samples and corresponding output samples are employed to build a Gaussian process regression (GPR) local model. Expectation maximation (EM) algorithm is utilized to deal with the nonlinearity and abnormal output observations in GPR local model simultaneously for robustness of the soft sensors. Numerical simulations and a benchmark process are employed to validate the effectiveness of the proposed soft sensor, which demonstrates its superior performance over traditional approaches.

1. Introduction

Data-driven modeling for industrial processes has received considerable attention. Compared with the traditional first principle modeling method, which requires certain rigorous physical assumptions, data-driven approach is more convenient and flexible, especially when dealing with a complex process [1–5]. Commonly, the development of soft sensor composes of different phases, aiming at constructing a high-quality predictive model with good prediction accuracy.

Generally speaking, when the practical industrial plant undergoes different operating modes or experiences nonlinearity, which may be caused by unstable transport conditions, catalyst deactivation, and instrument failure etc., a global model would be difficult to build if not impossible. By incorporating adaptive mechanism in soft sensor, several adaptive soft sensors have been developed to mitigate these problems [6]. As a local modeling approach, just-in-time learning (JITL) has proven to be more effective to mitigate abrupt-changes, complex nonlinearity, and time-varying behavior of processes [7–9]. The key step of JITL is relevant data sample determination or similarity measurement, along with construction of a local model.

Relevance measuring is a crucial step to ensure data samples used for modeling are most relevant to the query sample. Various relevance calculation methods have been presented in literatures [10–13]. However, most of these methods except few are deterministic, which cannot handle data with uncertainty. A probability similarity calculation method was developed by Yuan et al. [14], which can effectively mitigate uncertainty when handling noisy data. After completing relevance calculation, the relevant data samples in historical database can be selected for subsequent modeling. For establishing a predictive model, the entire historical database may need to be searched, which has a large quantity and is usually high dimensional along with redundancy and correlation. Therefore, extracting representative and often compressed features from massive historical data is an essential step. Generally, the model of extracted feature is expressed through a latent space model. Traditional celebrated latent models like principal component analysis (PCA), kernel PCA etc. [15–17] belong to the class of unsupervised learning. Recently, deep learning methods like convolutional neural network, autoencoder, and stacked autoencoder, etc. [18–20], have also been developed to extract features especially for image preprocessing and pattern recognition. Since Kingma and Welling [21] proposed Variational Autoencoder

[*] Corresponding author.
E-mail address: biao.huang@ualberta.ca (B. Huang).

https://doi.org/10.1016/j.chemolab.2020.104118
Received 9 May 2020; Received in revised form 12 July 2020; Accepted 16 July 2020
Available online 23 July 2020

(VAE), as a deep generative model, it has attracted increasing attention, and has been successfully applied in natural language processing, static images forecast and automatic speech recognition, etc. [22–25]. Despite this, VAE is an unsupervised deep learning method, in which extracted features only describe data information from inputs. A solution to improve prediction accuracy is by extracting output-relevant features from historical database. From this perspective, an output-relevant VAE is developed by first leveraging mutual information (MI) between inputs and outputs to select input variables. Subsequently, MI as variable weight is injected into the traditional VAE model, namely each individual input variable is assigned with a corresponding weight based on the calculated MI.

Since industrial process data commonly possess nonlinearities along with possibly abnormal observations, like outliers and missing data, which is caused by inevitable process upsets, sensor malfunctions, or mechanical breakdown etc. Current popular nonlinear regression modeling approaches in the literature include artificial neural networks, support vector regression, and Gaussian process regression (GPR) etc. [26,27]. As a machine learning method, GPR proceeds by introducing kernel technique and Gaussian prior. As a non-parametric modeling approach, GPR does not require many assumptions other than those related to the noise characteristics. In addition, since GPR can obtain prediction along with its uncertainty, it has been often employed to construct regression models [28,29]. Considering that industry data are usually contaminated with outliers and missing values, any modeling approach should be able to deal with these practical problems simultaneously. To mitigate outliers, various selections of noise distribution have been used by many researchers that can accommodate the outliers, like Laplace's distribution, t-distribution, and mixture Gaussian distributions etc. [30–33]. For handling missing data, commonly used methods include case-wise deletion, imputation, and expectation maximization (EM) algorithm etc. [34–36]. Among them, EM algorithm as an optimization algorithm is proceeded by iteration, which has been verified being more effective to address the problem of missing observations [37,38]. As aforementioned, GPR as a nonlinear regression modeling approach, is suitable to cope with small size of data samples and nonlinear relationship between input-output samples. These properties are especially suitable for JITL, since only a small amount of relevant historical samples are selected for establishing a local model under the JITL framework. Fig. 1 provides a general framework of JITL. Compared to a global modeling method, JITL, as a local modeling method, is performed based on query. When a query data arrives, a similarity measure is first calculated. Based on the similarity, a weight is assigned to each historical data point or a portion of historical data, which is the most relevant data to the query data and selected to build a local model. In addition, considering that the real industrial data commonly suffer from disturbances, like outliers along with missing data, this work will address these two practical problems simultaneously by employing the EM algorithm when building the GPR local model.

Motivated by the traditional VAE that is an unsupervised learning method, we propose a MI-based output-relevant VAE by introducing MI for selecting input variables that can be nonlinearly related to the output. Based on the calculated MI, weights are applied on the input variables of the VAE network. Use of MI in soft sensor modeling has also been considered in Refs. [39–41], to enhance the relationship between latent variables and quality variable, however under the stacked autoencoder framework that has a deterministic structure. In Ref. [42], correlation coefficients are utilized to describe the relationship between the input variables and the output variable under the VAE framework. However, unlike correction coefficients that can only express linear relationship between input variables and the output variable, MI is adopted in this work since it not only can describe linear correction but also nonlinear correlation. In addition, when the data are contaminated with outliers or there exist missing data, or both, a practical modeling method should be capable of simultaneously handling these problems, effective solution of which is another objective of this paper. In this work, outliers are modeled through the noise term of the GPR model by employing a two-component mixture Gaussian distribution with zero mean and two different variances, one small and the other large. The small variance represents normal noises while the large one represents the outliers. By utilizing the EM algorithm, the simultaneous solution of robustness to outliers and handling of missing data in GPR local model is obtained.

The rest of this paper proceeds as follows. Section II starts from the traditional VAE, then it develops a MI-based output-relevant VAE model. In Section III, the GPR model is revisited, along with a derivation of the robust GPR local modelling with capability of handling missing data. Section IV demonstrates the results through a numerical simulation and penicillin fermentation benchmark process. Section V concludes this paper.

2. Relevant calculation through proposed Variational Autoencoder

2.1. Overview of VAE

Built upon variational Bayes and deep learning methodologies, VAE reconstruct data by considering probability distribution as the latent variables. Let $\mathbf{X}$ represent the original dataset. As a generative model, given a prior distribution $p(\mathbf{Z})$, latent variable $\mathbf{Z}$ is first generated. Then, a reconstructed dataset $\mathbf{X}^{'}$ is generated through the conditional generative distribution $p_{\phi}(\mathbf{X}|\mathbf{Z})$, where ϕ denotes parameters. According to Refs. [21], traditionally, latent variables are output variables of a recognition network $q(\mathbf{Z}|\mathbf{X})$ with parameters θ, namely, $q_{\theta}(\mathbf{Z}|\mathbf{X})$, which are discrete variables. By utilizing the reparameterization technique, the discrete latent variable are transferred to the continuous. The parameters in the VAE generative model can then be determined through stochastic gradient variational Bayes. Fig. 2 gives the traditional VAE model structure. For more systematic introduction to the inferencing of the traditional VAE, the readers can refer to Ref. [21,22].

2.2. An MI-based output-relevant VAE

Considering that the traditional VAE is an unsupervised learning

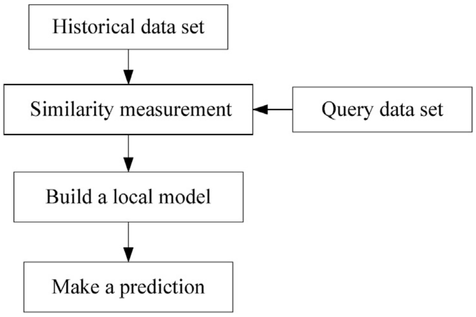

Fig. 1. JITL framework.

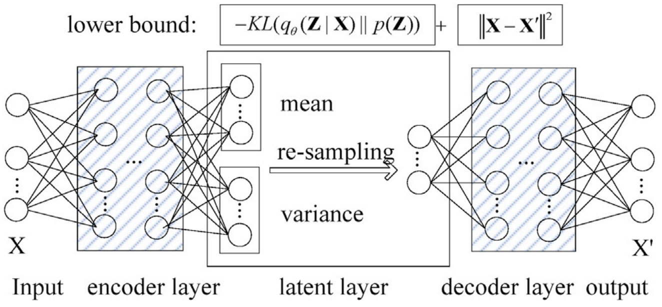

Fig. 2. VAE structure.

approach, this section will discuss an output-relevant VAE aiming at describing the relationship between latent representations and output variable by leveraging MI. MI can be utilized for evaluating both linear and nonlinear relationship between input-output variables. It is defined as follows [39],

$$\mathrm{MI}(x,x^{'}) = \iint_{x,x^{'}} p(x,x^{'})\log(p(x,x^{'})\,/\,p(x)p(x^{'}))dxdx^{'}. \tag{1}$$

where $p(x,x^{'})$ represents the joint probability density between variable x and variable $x^{'}$. $p(x)$ and $p(x^{'})$ represent their marginal probability, respectively. Further, introduce Shannon entropy $H(x)$ of random variable x as

$$H(x) = -\int p(x)\log p(x)dx. \tag{2}$$

The corresponding joint entropy between variable x and variable $x^{'}$, denoted as $H(x,x^{'})$, can be computed as below,

$$H(x,x^{'}) = -\int p(x,x^{'})\log p(x,x^{'})dxdx^{'}. \tag{3}$$

Substituting (2) and (3) into (1), the MI can be rewritten as,

$$\mathrm{MI}(x,x^{'}) = H(x) + H(x^{'}) - H(x,x^{'}). \tag{4}$$

With the MI values, nonlinear correction between input variables and output variable can be obtained. Compared with the traditional VAE, the proposed VAE extracts features from output data in the latent layer, which plays an important role in establishing the soft sensor model. The proposed VAE retains the same structure in the recognition network as that of the traditional VAE, however, it introduces an additional weighted variable-wise reconstruction within the generative network. As a result, the lower bound of the proposed VAE is evaluated as below,

$$V(\phi,\theta,\mathbf{X}) = -\mathrm{KL}(q_\theta(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})) + \sum_{n=1}^{N}\sum_{d=1}^{D}\lambda_d(x_{nd}-\widehat{x}_{nd})^2. \tag{5}$$

where $\mathrm{KL}(q_\theta(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z}))$ represents the KL divergence between the encoder $q_\theta(\mathbf{Z}|\mathbf{X})$ and prior $p(\mathbf{Z})$. $\sum_{n=1}^{N}\sum_{d=1}^{D}\lambda_d(x_{nd}-\widehat{x}_{nd})^2$ denotes the weighted reconstruction error between the d-th variable of the n-th input sample x_n and that of the n-th reconstructed sample $\widehat{x}_n$. Here, λ_d denotes the weight on the d-th MI-based input variable, which is computed by

$$\lambda_d = |\mathrm{MI}_d| \Bigg/ \sum_{d=1}^{D}|\mathrm{MI}_d|. \tag{6}$$

As aforementioned, the input variables will be sorted by using the calculated MI values, that is, input variables with the smaller MI values will be eliminated. Hence, determination of an appropriate threshold is important for selecting correct input variables for subsequent modeling. An efficient strategy for selecting appropriate threshold has been considered on the stacked autoencoder models [39]. This paper adopts the same method to determine threshold as follows: Generate a random variable x_{rand}, which is independent of the output variable. Then, calculate the MI values through (4), and use the calculated MI as the threshold. Correspondingly, for the given data set, if the MI value between an input and the output is smaller than the threshold, the input variables will be removed. After this procedure, a set of input variables is determined, which is most relevant to the output variable. This set of input variables will be used in the subsequent steps.

2.3. Selected new modeling samples

The proposed VAE generates a network that produces the latent variables which represent the distributions of the data. After the network is trained, given a data sample, by inputting the data to the network, its distribution of the individual latent variable can be determined. Subsequently the Symmetric Kullback-Leibler (SKL) can be calculated between any two Gaussian distributions, which is defined by Refs. [43,44].

$$\begin{aligned}&\mathrm{SKL}(\mathcal{N}(\overline{z}_n,\tilde{z}_n),\mathcal{N}(\overline{z}_s,\tilde{z}_s))\\ &=\sum_{t=1}^{T}\mathrm{SKL}(\mathcal{N}(\overline{z}_{nt},\tilde{z}_{nt}),\mathcal{N}(\overline{z}_{st},\tilde{z}_{st}))\\ &=\sum_{t=1}^{T}0.5\times trace\left[(\tilde{z}_{st}-\tilde{z}_{nt})\left(\tilde{z}_{nt}^{-1}-\tilde{z}_{st}^{-1}\right)\right]\\ &+0.5\times(\overline{z}_{st}-\overline{z}_{nt})^T\left(\tilde{z}_{nt}^{-1}+\tilde{z}_{st}^{-1}\right)(\overline{z}_{st}-\overline{z}_{nt}).\end{aligned} \tag{7}$$

where $\mathcal{N}(\overline{\mathbf{z}}_n,\tilde{\mathbf{z}}_n)$ and $\mathcal{N}(\overline{\mathbf{z}}_s,\tilde{\mathbf{z}}_s)$ represent the distribution of the n-th latent sample and s-th latent sample, respectively. Meanwhile, $\mathcal{N}(\overline{\mathbf{z}}_{nt},\tilde{\mathbf{z}}_{nt})$ and $\mathcal{N}(\overline{\mathbf{z}}_{st},\tilde{\mathbf{z}}_{st})$ are the distribution of their t-th variable, respectively. Here, $\mathcal{N}(\overline{\mathbf{z}}_{nt},\tilde{\mathbf{z}}_{nt})$ denotes a Gaussian distribution with mean $\overline{\mathbf{z}}_{nt}$ and variance $\tilde{\mathbf{z}}_{nt}$, $t=1,\ldots,T$, and T denotes the latent variable dimension.

Under the JITL framework, when a query sample comes, SKL divergence between the query sample and each historical data sample is calculated. Then, order the historical data according to SKL divergence values. A smaller SKL value means higher relevance of the corresponding data. Based on the threshold discuss earlier, select relevant historical data that have smaller SKL value along with their corresponding outputs to establish a GPR local model subsequently.

3. Proposed JIT-based soft sensor modeling with outliers and missing outputs

After selecting the relevant training data as outlined in section II, a GPR model is employed to construct a local model in JITL framework. Additionally, considering that the common existence of outliers and missing observations in output dataset, we utilize EM algorithm to cope with outliers and missing data for parameter estimation, simultaneously. This section will first overview GPR model, and then provides the detailed derivation for dealing with outliers and missing data through the EM algorithm.

3.1. GPR model

Consider input $\mathbf{X}$ and scalar output $\mathbf{y}$, a GPR model is expressed as follows [28],

$$y_n = f(\mathbf{x}_n) + \varepsilon. \tag{8}$$

where $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, $\mathbf{x}_n \in \mathbb{R}^D$, $n = 1,\ldots,N$, N denotes the total data sample number, D is the input variable dimension, and $\mathbf{y} = \{y_1,\ldots,y_N\}$ $y_n \in \mathbb{R}$ is the output. Noise term ε follows normal distribution, that is, $\varepsilon\tilde{\ }\mathcal{N}(0,\sigma_y^2)$. The function $f(\cdot)$ denotes the unknown function also known as latent function in the GPR model, which is assumed by a Gaussian prior with $m(\cdot)$ as mean function and $k(\cdot)$ as covariance function. Commonly, this mean function $m(\cdot)$ is taken to be zero, that is,

$$f(\mathbf{x})\tilde{\ }\mathrm{GP}(0,k(\mathbf{x},\mathbf{x}^{'})). \tag{9}$$

As for covariance function, $k(\mathbf{x},\mathbf{x}^{'}) = \mathrm{cov}(f(\mathbf{x}),f(\mathbf{x}^{'}))$ is generally described through the following square exponential kernel function with kernel parameters σ_f^2 and l_d^2

$$k(\mathbf{x},\mathbf{x}^{'}) = \sigma_f^2\exp\left(-\sum_{d=1}^{D}\left(x_d-x_d^{'}\right)^2/2l_d^2\right). \tag{10}$$

Therefore, the distribution of the output $\mathbf{y}$ in (8) can be inferred as below,

$$\mathbf{y}\tilde{\ }\mathcal{N}\left(0,\mathbf{K}+\sigma_y^2\mathbf{I}_N\right) \tag{11}$$

where $\mathbf{K}$ represents a symmetric kernel matrix, in which covariance function $k(\cdot)$ denotes each element in $\mathbf{K}$, which can be computed through (10). $\mathbf{I}_N$ represents an $N \times N$ unit matrix.

All hyperparameters in GPR model are expressed as $\varPhi_{GP} = \{\sigma_f^2, l_1^2, ..., l_D^2, \sigma_y^2\}$, which can be determined by maximizing following $\log p(\mathbf{y}|\mathbf{X}, \varPhi_{GP})$ through the conjugate gradient (CG) method

$$\log p(\mathbf{y}|\mathbf{X}, \varPhi_{GP}) = N/2\log(2\pi) + 1/2\log\left|\mathbf{K} + \sigma_y^2\mathbf{I}_N\right| \\ +1/2\mathbf{y}^{\mathrm{T}}\left(\mathbf{K} + \sigma_y^2\mathbf{I}_N\right)^{-1}\mathbf{y}. \tag{12}$$

Further, given a query (or test) input data set $\mathbf{X}_{test}$, the joint distribution of the corresponding noise-free output $\mathbf{f}_{test}$ and the training output $\mathbf{y}$ can be expressed by

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_{test} \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_y^2\mathbf{I}_N & \mathbf{K}(\mathbf{X},\mathbf{X}_{test}) \\ \mathbf{K}(\mathbf{X}_{test},\mathbf{X}) & \mathbf{K}(\mathbf{X}_{test},\mathbf{X}_{test}) \end{bmatrix}\right). \tag{13}$$

Additionally, the normal distribution of $\mathbf{f}_{test}$ can be written as

$$\mathbf{f}_{test}|\mathbf{X}_{test}, \mathbf{X}, \mathbf{y} \sim \mathscr{N}(\mathbf{m}_{test}, \mathbf{K}_{test}). \tag{14}$$

where mean $\mathbf{m}_{test}$ and covariance $\mathbf{K}_{test}$, which are the predicted outputs along with their covariance, can be respective computed by

$$\mathbf{m}_{test} = \mathbf{K}(\mathbf{X}_{test}, \mathbf{X})\left(\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_y^2\mathbf{I}_N\right)^{-1}\mathbf{y}. \tag{15}$$

$$\mathbf{K}_{test} = \mathbf{K}(\mathbf{X}_{test}, \mathbf{X}_{test}) - \mathbf{K}(\mathbf{X}_{test}, \mathbf{X})\left(\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_y^2\mathbf{I}_N\right)^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_{test}). \tag{16}$$

3.2. Local GPR modeling with outliers and missing data based on the EM algorithm

Under the maximum likelihood estimation (MLE) framework, the EM algorithm is an effective algorithm to estimate model parameters with hidden variables or missing data [45]. It is achieved by iterating between E step and M step. The E step aims at calculating the Q function through an measured dataset C_{obs}, a hidden dataset C_{mis}, and a unknown parameter set $\varPhi$ as shown below,

$$Q\left(\varPhi\middle|\varPhi^{(h)}\right) = E_{C_{mis}|C_{obs},\varPhi^{(h)}}\{\log p(C_{obs}, C_{mis}|\varPhi)\}. \tag{17}$$

where $\varPhi^{(h)}$ represents the parameter set of the h-th iteration step. Then, the M step is to maximize (17) with respect to parameter set $\varPhi$, and obtain the new solution for parameter set $\varPhi^{(h+1)}$ by

$$\varPhi^{(h+1)} = \mathrm{argmax}Q\left(\varPhi\middle|\varPhi^{(h)}\right). \tag{18}$$

The convergence of this algorithm has been shown in Refs. [46].

In this work, the data outliers are described by heteroscedastic Gaussian noise ε_n , which is combination of two modes of probability distributions, respectively $\mathscr{N}(0, \sigma_1^2)$ and $\mathscr{N}(0, \sigma_J^2)$. Define a hidden variable $\mathbf{I} = \{I_1, ...I_N\}$, where each $I_n \in \{1, ..., J\}$, $j = 1, ..., J$, and J is the number of the noise modes (equals 2 for the two-mode case). Output data set is $\mathbf{y} = \{\mathbf{y}_s, \mathbf{y}_u\}$, in which $\mathbf{y}_s$ represents observed output data, that is, $\mathbf{y}_s = \{\mathbf{y}_{s,1}, ..., \mathbf{y}_{s,N_s}\}$, and $\mathbf{y}_u$ denotes missing output data, that is, $\mathbf{y}_u = \{\mathbf{y}_{u,1}, ..., \mathbf{y}_{u,N_u}\}$, where $N_s + N_u = N$. With observed dataset$C_{obs} = \{\mathbf{X}, \mathbf{y}_s\}$ and hidden dataset $C_{mis} = \{\mathbf{f}, \mathbf{I}, \mathbf{y}_u\}$, the Q function can be written as

$$\begin{aligned} &Q\left(\varPhi\middle|\varPhi^{(h)}\right) \\ &= E_{C_{mis}|C_{obs},\varPhi^{(h)}}\{\log p(\mathbf{X}, \mathbf{y}, \mathbf{f}, \mathbf{I}|\varPhi)\} \\ &= E_{C_{mis}|C_{obs},\varPhi^{(h)}}\left\{\begin{array}{l} \log p(\mathbf{y}|\mathbf{f}, \mathbf{I}, \mathbf{X}, \varPhi) + \log p(\mathbf{f}|\mathbf{I}, \mathbf{X}, \varPhi) \\ +\log p(\mathbf{I}|\mathbf{X}, \varPhi) + \log p(\mathbf{X}|\varPhi) \end{array}\right\}. \end{aligned} \tag{19}$$

Considering that $\mathbf{X}$ is not related with parameter set $\varPhi$, we can treat the third term in above bracket as a constant C_0. The first two terms can be expressed as follows, respectively,

$$\log p(\mathbf{y}|\mathbf{f}, \mathbf{I}, \mathbf{X}, \varPhi) = -1/2\begin{bmatrix} N\log 2\pi + \log|\mathbf{W}| \\ +(\mathbf{y} - \mathbf{f})^{\mathrm{T}}\mathbf{W}^{-1}(\mathbf{y} - \mathbf{f}) \end{bmatrix}. \tag{20}$$

where $\mathbf{W} = diag[\sigma_{I_1}^2, ..., \sigma_{I_N}^2]$.

$$\log p(\mathbf{f}|\mathbf{I}, \mathbf{X}, \varPhi) = -1/2\left[\log|2\pi\mathbf{K}| + \mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}\right]. \tag{21}$$

Consequently, the posterior distribution of $p\left(\mathbf{f}\middle|\mathbf{y}, \mathbf{I}, \mathbf{X}, \varPhi^{(h)}\right)$ can be derived as follows,

$$p\left(\mathbf{f}\middle|\mathbf{y}, \mathbf{I}, \mathbf{X}, \varPhi^{(h)}\right) \propto p\left(\mathbf{y}\middle|\mathbf{f}, \mathbf{I}, \mathbf{X}, \varPhi^{(h)}\right)p\left(\mathbf{f}\middle|\mathbf{X}, \varPhi^{(h)}\right). \tag{22}$$

where the first probability term can be derived by

$$\begin{aligned} p\left(\mathbf{f}\middle|\mathbf{y}, \mathbf{I}, \mathbf{X}, \varPhi^{(h)}\right) &\sim \exp\left\{-1/2\left(\mathbf{f} - \mathbf{S}^{-1}\mathbf{L}\right)^{\mathrm{T}}\mathbf{W}\left(\mathbf{f} - \mathbf{S}^{-1}\mathbf{L}\right)\right\} \\ &\sim \mathscr{N}\left(\mathbf{m}^{(h)}, \mathbf{B}^{(h)}\right) \quad . \end{aligned} \tag{23}$$

where $\mathbf{S} = \left(\mathbf{K}^{(h)}\right)^{-1} + \left(\mathbf{W}^{(h)}\right)^{-1}$, $\mathbf{L} = \left(\mathbf{W}^{(h)}\right)^{-1}\mathbf{y}$,

$\mathbf{m}^{(h)} = \mathbf{S}^{-1}\mathbf{L}$, and $\mathbf{B}^{(h)} = \mathbf{S}^{-1}$.

Further, substituting (20)–(23) into (19), the Q function can be further derived as

$$\begin{aligned} &Q\left(\varPhi\middle|\varPhi^{(h)}\right) \\ &= E_{\mathbf{f},\mathbf{I},\mathbf{y}_u|C_{obs},\varPhi^{(h)}}\left\{\begin{array}{l} -1/2\left[N\log 2\pi + \log|\mathbf{W}| + (\mathbf{y} - \mathbf{f})^{\mathrm{T}}\mathbf{W}^{-1}(\mathbf{y} - \mathbf{f})\right] \\ -1/2\left[\log|2\pi\mathbf{K}| + \mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}\right] + \log p\left(\mathbf{I}\middle|\mathbf{X}, \varPhi\right) + C_0 \end{array}\right\} \\ &= E_{\mathbf{I}|C_{obs},\varPhi^{(h)}}\left\{\begin{array}{l} -1/2\left(N\log 2\pi + \log|\mathbf{W}| + tr\left(\mathbf{W}^{-1}\mathbf{B}^{(h)}\right)\right) \\ +\log p(\mathbf{I}|\mathbf{X}, \varPhi) \end{array}\right\} \\ &\quad +E_{\mathbf{I},\mathbf{y}_u|C_{obs},\varPhi^{(h)}}\left\{-1/2\left(\mathbf{y} - \mathbf{m}^{(h)}\right)^{\mathrm{T}}\mathbf{W}^{-1}\left(\mathbf{y} - \mathbf{m}^{(h)}\right)\right\} \\ &\quad -1/2\left[\log|2\pi\mathbf{K}| + \left(\mathbf{m}^{(h)}\right)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{m}^{(h)} + tr\left(\mathbf{K}^{-1}\mathbf{B}^{(h)}\right)\right] + C_0 \\ &= \sum_{j=1}^{J}\sum_{n=1}^{N}p\left(I_n = j\middle|C_{obs}, \varPhi^{(h)}\right)\left\{\begin{array}{l} -1/2\left[N\log 2\pi + \log\sigma_j^2 + B_{nn}^{(h)}\middle/\sigma_j^2\right] \\ -1/2\sum_{i=1}^{N_s}\left(y_{s,i} - m_{s,i}\right)^2\middle/\sigma_{I_{s,i}}^2 + \log\gamma_j \end{array}\right\} \\ &\quad +E_{\mathbf{I},\mathbf{y}_u|C_{obs},\varPhi^{(h)}}\left\{-1/2\sum_{i=1}^{N_u}\left(y_{u,i} - m_{u,i}\right)^2\middle/\sigma_{I_{u,i}}^2\right\} \\ &\quad -1/2\left[\log|2\pi\mathbf{K}| + \left(\mathbf{m}^{(h)}\right)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{m}^{(h)} + tr\left(\mathbf{K}^{-1}\mathbf{B}^{(h)}\right)\right] + C_0\,. \end{aligned} \tag{24}$$

where

$$E_{\mathbf{I},\mathbf{y}_u|C_{obs},\varPhi^{(h)}}\left\{-1/2\sum_{i=1}^{N_u}\left(y_{u,i} - m_{u,i}\right)\middle/\sigma_{I_{u,i}}^2\right\} = N_u.$$

Finally, the Q function can be obtained as below,

$$\begin{aligned} Q\left(\varPhi\middle|\varPhi^{(h)}\right) &= \sum_{j=1}^{J}\sum_{n=1}^{N}p\left(I_n = j\middle|C_{obs}, \varPhi^{(h)}\right)\left\{\begin{array}{l} -1/2\left[N\log 2\pi + \log\sigma_j^2 + B_{nn}^{(h)}\middle/\sigma_j^2\right] \\ -1/2\sum_{i=1}^{N_s}\left(y_{s,i} - m_{s,i}\right)^2\middle/\sigma_{I_{s,i}}^2 + \log\gamma_j \end{array}\right\} \\ &\quad -1/2N_u - 1/2\left[\log|2\pi\mathbf{K}| + \left(\mathbf{m}^{(h)}\right)^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{m}^{(h)} + tr\left(\mathbf{K}^{-1}\mathbf{B}^{(h)}\right)\right] + C_0. \end{aligned} \tag{25}$$

By maximizing the Q function with respect to parameter set $\varPhi_{GP}$, the two noise variances can be obtained, respectively,

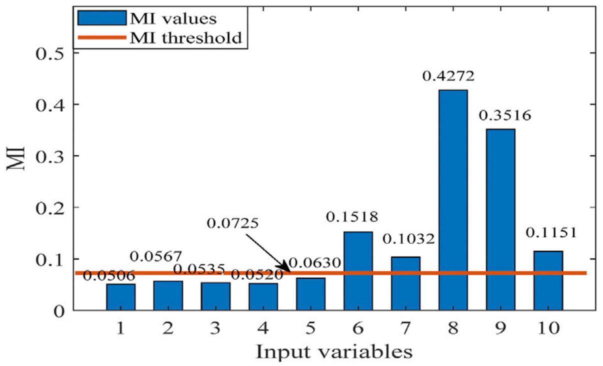

Fig. 3. MI values of input variables.

$$\frac{\partial Q}{\partial \Phi_{GP}} = -1/2\left[tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Phi_{GP}}\right) - \left(\mathbf{m}^{(h)}\right)^{\mathrm{T}}\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Phi_{GP}}\mathbf{m}^{(h)}\right]$$
$$+1/2tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Phi_{GP}}\mathbf{K}^{-1}\mathbf{B}^{(h)}\right). \tag{26}$$

$$\left(\sigma_j^2\right)^{(h+1)} = \frac{\sum_{i=1}^{N_s} p\left(I_{s,i}=j\middle|C_{obs},\Phi^{(h)}\right)\left[\left(y_{s,i}-m_{s,i}\right)^2\right]}{\sum_{n=1}^{N} p\left(I_n=j\middle|C_{obs},\Phi^{(h)}\right)}$$

$$+\frac{\sum_{n=1}^{N} p\left(I_n=j\middle|C_{obs},\Phi^{(h)}\right)B_{nn}^{(h)}}{\sum_{n=1}^{N} p\left(I_n=j\middle|C_{obs},\Phi^{(h)}\right)}. \tag{27}$$

In addition, solution for the coefficient parameters γ_j is a constraint optimization problem, which can be calculated by introducing the Lagrange multiplier and the solution is given by,

$$\gamma_j = \sum_{n=1}^{N} p\left(I_n=j\middle|C_{obs},\Phi^{(h)}\right) \Big/ N. \tag{28}$$

Up to now, a robust GPR local model has been built, which can deal with outliers and missing data.

3.3. *Making predictions*

Finally, given a query dataset $\mathbf{X}^q$, the distribution of corresponding noise-free output $\mathbf{f}^q$ is calculated by the following distribution with mean $\mathbf{m}^q$ mean and covariance $\mathbf{B}^q$:

$$\mathbf{f}^q|\mathbf{X},\mathbf{X}^q,\mathbf{y} \sim \mathcal{N}(\mathbf{m}^q,\mathbf{B}^q). \tag{29}$$

$$\text{where}\quad \mathbf{m}^q = \mathbf{K}^q(\mathbf{K}+\mathbf{W})^{-1}\mathbf{y}. \tag{30}$$

$$\mathbf{B}^q = \mathbf{K}^{q,q} - \mathbf{K}^q(\mathbf{K}+\mathbf{W})^{-1}(\mathbf{K}^q)^{\mathrm{T}}. \tag{31}$$

4. Simulation and applications

A numerical example and a practical industrial process are utilized to validate the proposed method. To evaluate the prediction performance, by denoting $\hat{y}_n$ as the n-th predicted output, the following three error criteria, root-mean-squared error (RMSE), mean-absolute error (MAE), and mean-relative error (MRE) are calculated,

$$\text{RMSE} = \sqrt{\sum_{n=1}^{N}\left(y_n-\hat{y}_n\right)^2/N}. \tag{32}$$

$$\text{MAE} = \sum_{n=1}^{N}|y_n-\hat{y}_n|\,/\,N. \tag{33}$$

$$\text{MRE} = \sum_{n=1}^{N}|y_n-\hat{y}_n|\,/\,y_n. \tag{34}$$

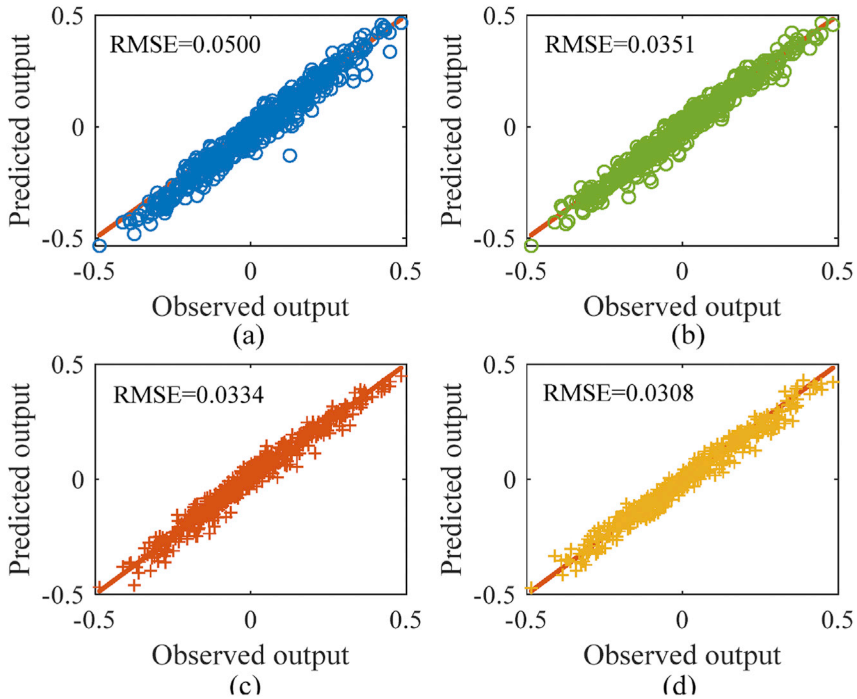

Fig. 4. Prediction results of JIT-based soft sensor model through different feature extraction methods ((a) traditional VAE; (b) output-relevant VAE [42]; (c) MI-VAE; (d) MI-based output-relevant VAE).

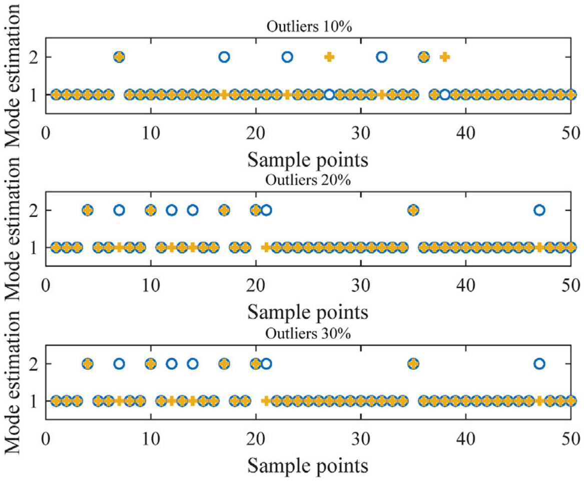

Fig. 5. Mode estimation with different outlier cases. (o: True mode; +: Estimation mode).

Table 1
Comparison results of different error criterions between CG and EM algorithm.

Missing percentage	MI-based output-relevant VAE JIT-based with CG			Proposed method		
	RMSE	MAE	MRE	RMSE	MAE	MRE
10	0.1536	0.1211	3.0764	0.0835	0.0620	1.0070
20	0.1567	0.1236	3.2830	0.1061	0.0832	1.0396
30	0.1621	0.1282	5.0840	0.1112	0.0908	1.0470

4.1. Numerical example

A nonlinear model consists of ten input variables and one output variable. Each input variable x_k is generated by a uniform distribution, which is bounded in $[0,1]$, $k=1,\ldots,10$ [39].

$$y = 4\exp(x_6) + 6\sqrt{x_7} + 12x_8^2 + 10x_9 + 5x_{10} + \varepsilon \tag{35}$$

Note that variables from x_1 to x_5 are not related to output y. Noise term ε is randomly generated from two-component mixture Gaussian distribution, $\mathcal{N}(0, 0.005)$ and $\mathcal{N}(0, 1.5)$. One thousand samples are generated in this simulation, where training set consists of the first five hundred samples, and the rest samples are employed as testing data set. The threshold is set as 0.0725, which is determined by using a random input sequence x_{rand} and output sequence y with no relation between them. Then the threshold is determined by computing $0.95 \times \mathrm{MI}(x_{rand}, y)$. Fig. 3 provides MI value for each input variable. We can clearly see that the values of MI are smaller than the given threshold from x_1 to x_5, while the MI values from x_6 to x_{10} are larger than the given threshold, which indeed is in accordance with (35).

A MI-based output-relevant VAE network is established with three hidden layers, and neuron in each layer is [3–5], respectively in the encoder. The decoder has a symmetric structure with the encoder. Without outliers and missing data, Fig. 4 plots the predicted output along with RMAE values of the JIT-based soft sensor with the same GPR local model and EM optimization algorithm by utilizing the various methods of feature extraction. These include traditional VAE, output-relevant VAE [42], MI-VAE (which first eliminates input variables irrelevant of the output variable according to the MI values, followed by the tradition VAE), and the proposed MI-based output-relevant VAE. As indicated by Fig. 4, the MI-based output-relevant VAE indeed extracts features that are more output-relevant than the others. Next, we simulate the scenarios with outliers and missing data and compare performance between the proposed method and others. With data missing ratio 20% and different outlier percentages 10%, 20%, 30%, the results of mode estimation by the proposed method are drawn in Fig. 5.

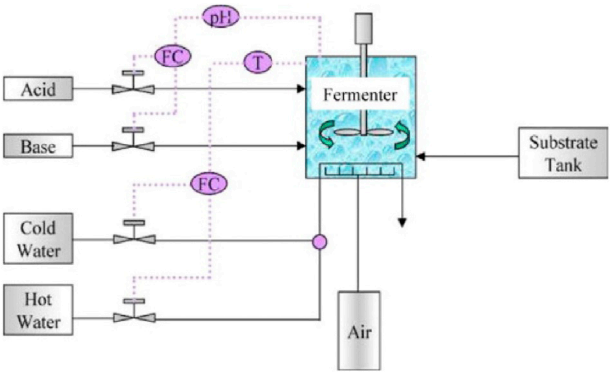

Fig. 6. Flowchart of penicillin process [48].

Table 2
Variables in penicillin process [48].

Variable	Variable description
Input x1	Agitator power
Input x2	Aeration rate
Input x3	Substrate feed temperature
Input x4	Substrate feed rate
Input x5	Cooling water flow rate
Input x6	Base flow rate
Input x7	Dissolved oxygen concentration
Input x8	Substrate concentration
Input x9	Culture volume
Input x10	Biomass concentration
Input x11	PH
Input x12	Carbon dioxide concentration
Input x13	Fermenter temperature
Output y	Penicillin concentration

Table 1 lists the prediction results of the MI-based output-relevant VAE as feature extraction method but with various optimization algorithms, namely, CG and EM algorithm, respectively. From Table 1, the EM algorithm as proposed in this paper shows the better estimation performance, further demonstrating that it is superior in coping with outliers and missing data.

4.2. Penicillin fermentation process

The flowchart of penicillin process is provided in Fig. 6. As a biochemical process, it is usually employed for monitoring, control, and soft sensing [47]. A website of this process can be found in http://simulator.itt.edu/web/pensim/index.html [47]. For soft sensing application, 13 variables are chosen as the inputs, and the penicillin concentration is selected as the output variable [48]. Corresponding description of input-output variables is provided in Table 2. The details of the

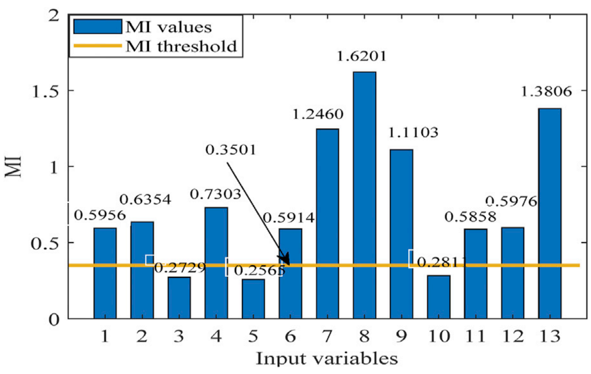

Fig. 7. MI values of input variables on penicillin process.

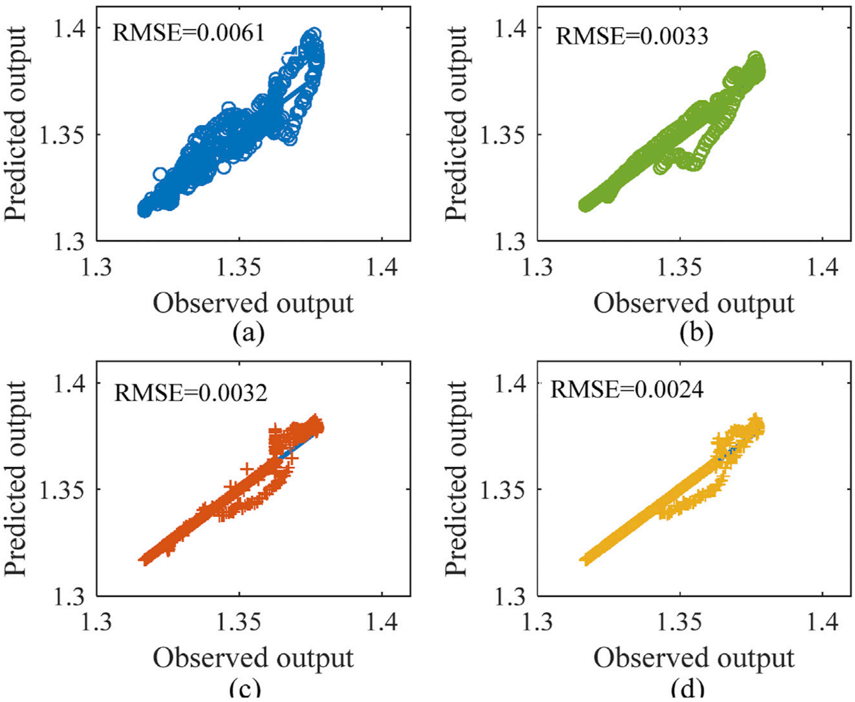

Fig. 8. Prediction results of JIT-based soft sensor model through different feature extraction methods ((a) traditional VAE; (b) output-relevant VAE [42]; (c) MI-VAE; (d) MI-based output-relevant VAE) for penicillin process.

Table 3
Comparison results of error criterions between CG and EM algorithm.

20% outliers and different missing ratio	MI-based output-relevant VAE JIT-based with CG			Proposed method		
	RMSE	MAE	MRE	RMSE	MAE	MRE
10%	0.1334	0.1248	0.0927	0.0067	0.0053	0.0044
20%	0.2603	0.2545	0.1893	0.0092	0.0079	0.0067
30%	0.3986	0.3944	0.2934	0.0113	0.0108	0.0095

penicillin process can be found in Ref. [47]. A total of one thousand six hundred data points are sampled from penicillin process through simulation. The first one thousand data points are used for training, the remaining for testing.

The selected network structure of the proposed VAE has three hidden layers, and the number of neurons in each layer is [5,7,10] in the encoder. Meanwhie, a symmetric structure is employed in the decoder. Fig. 7 gives the MI value of each input variable. Threshold is chosen as 0.3501 by using the random sampling method as discussed earlier. Without outliers and missing data, Fig. 8 provides the predictions along with RMSE values of the algorithms with four different feature extraction methods as aforementioned. Moreover, to demonstrate the prediction performance of the proposed soft sensor modeling for dealing with outliers and missing data, by randomly injecting 20% outliers into the training data artificially along with different data missing percentages 10%, 20%, 30%, both RMSE, MAE and MRE values for two methods are given in Table 3. These two methods include the CG algorithm and the proposed EM algorithm. From Table 3, the proposed method shows the superior prediction accuracy, due to its ability in effectively dealing with outliers and missing data.

5. Conclusion

A deep learning approach was presented for soft sensor development under the JITL framework by employing a MI- based output-relevant VAE. Input variables are first selected by calculating MI values between inputs and output, and then weights according to the calculated MI are assigned to each input variable. Subsequently, the weighted data are incorporated to train the VAE model, and the MI-based output-relevant VAE is developed for effective extraction of features. After that, a SKL divergence is used to measure similarity between the query sample and historical samples. Based on the SKL divergence, the relevant historical input data samples are selected. Those selected data samples and corresponding output samples are used for constructing a GPR local model. Considering the practical problems of outliers and missing data, the EM algorithm is utilized to cope with these problems. A numerical example and a benchmark simulation example demonstrated the effectiveness of the proposed soft sensor modeling approach in dealing with data with the redundancy, outliers and missing values.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Fan Guo: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Biao Huang:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Resources, Supervision, Writing - review & editing.

Acknowledgements

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC).

References

[1] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (2009) 795–814.
[2] S. Khatibisepehr, B. Huang, S. Khare, Design of inferential sensors in the process industry: a review of Bayesian methods, J. Process Contr. 23 (2013) 1575–1596.

[3] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, Soft Sensors for Monitoring and Control of Industrial Processes, Springer, Berlin, Germany, 2007.

[4] X.F. Yuan, J. Zhou, B. Huang, Y.L. Wang, C.H. Yang, W.H. Gui, Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy, IEEE Trans. Industr. Inform. 16 (2020) 3721–3730.

[5] X.F. Yuan, C. Ou, Y.L. Wang, C.H. Yang, W.H. Gui, A layer-wise data augmentation strategy for deep learning networks and its soft sensor application in an industrial hydrocracking process, IEEE Transactions on Neural Networks and Learning Systems (2019), https://doi.org/10.1109/TNNLS.2019.29, 51708.

[6] S. Joe Qin, Recursive PLS algorithms for adaptive data modeling, Comput. Chem. Eng. 22 (1998) 503–514.

[7] P. Kadlec, B. Gabrys, Local learning-based adaptive soft sensor for catalyst activation prediction, AIChE J. 57 (2011) 1288–1301.

[8] Z.Q. Ge, Z.H. Song, A comparative study of just-in-time-learning based methods for online soft sensor modeling, Chemometr. Intell. Lab. Syst. 104 (2010) 306–317.

[9] M. Chen, S. Khare, B. Huang, A unified recursive just-in-time approach with industrial near infrared spectroscopy application, Chemometr. Intell. Lab. Syst. 135 (2014) 133–140.

[10] C. Cheng, M.S. Chiu, A new data-based methodology for nonlinear process modeling, Chem. Eng. Sci. 59 (2004) 2801–2810.

[11] K. Fujiwara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (2009) 1754–1765.

[12] L.L.T. Chan, X.F. Wu, J.H. Chen, L. Xie, C.I. Chen, Just-in-time modeling with variable shrinkage based on Gaussian processes for semiconductor manufacturing, IEEE Trans. Semicond. Manuf. 31 (2018) 335–342.

[13] N. Magbool, B. Huang, E. Aris, Z. Luke, F. Xu, G. Lee, Just-in-time learning for the prediction of oilsands ore characteristics using GPS data in mining applications, Can. J. Chem. Eng. (2020) 1–12, https://doi.org/10.1002/cjce.23742.

[14] X.F. Yuan, Z.Q. Ge, B. Huang, Z.H. Song, A probabilistic just-in-time learning framework for soft sensor development with missing data, IEEE Trans. Contr. Syst. Technol. 25 (2017) 1124–1132.

[15] H.T. Chen, B. Jiang, N.Y. Lu, Z.H. Mao, Deep PCA based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains, IEEE Trans. Veh. Technol. 67 (2018) 4819–4830.

[16] N. Li, S.J. Guo, Y.Q. Wang, Weighted preliminary-summation-based principal component analysis for non-Gaussian processes, Contr. Eng. Pract. 87 (2019) 122–132.

[17] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks- ICANN'97, Springer, Berlin, Heidelberg, 1977, pp. 583–588, 1327.

[18] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[19] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: Proceedings of the 31st International Conference on Machine Learning (ICML), 2014 arXiv:1401.4082.

[20] J.G. Wang, Y. Wang, Y. Yao, B.H. Yang, S.W. Ma, Stacked autoencoder for operation prediction of coke dry quenching process, Contr. Eng. Pract. 88 (2019) 110–118.

[21] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: The 2nd International Conference on Learning Representations (ICLR), 2013 arXiv1312.6114.

[22] C. Doersch, Tutorial on Variational Autoencoders, 2016 arXiv:1606.05908.

[23] F. de-la-Calle-Silos, R.M. Stern, Synchrony-based feature extraction for robust automatic speech recognition, IEEE Signal Process. Lett. 24 (2017) 1158–1162.

[24] J. Walker, C. Doersch, A. Gupta, M. Hebert, An uncertain future: forecasting from static images using variational autoencoders, in: European Conference on Computer Vision (ECCV), 2016, pp. 835–851, arXiv:1606.07873.

[25] W.N. Hsu, Y. Zhang, J. Glass, Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation, in: Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop on, IEEE, 2017.

[26] H.P. Jin, X.G. Chen, J. Yang, H. Zhang, L. Wang, L. Wu, Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes, Chem. Eng. Sci. 131 (2015) 282–303.

[27] C.E. Rasmussen, Evaluation of Gaussian Processes and Other Methods for Non-linear Regression, University of Toronto, Canada, 1996. Ph.D. Thesis.

[28] M. Kuss, Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning, Technische Universität Darmstadt, Germany, 2006. Ph.D. Thesis.

[29] A. Daemia, Y. Alipouri, B. Huang, Identification of robust Gaussian process regression with noisy input using EM algorithm, Chemometr. Intell. Lab. Syst. 191 (2019) 1–11.

[30] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, vol. 589, John Wiley & sons, 2005.

[31] M.E. Tipping, N.D. Lawrence, Variational inference for student-t models: robust Bayesian interpolation and generalised component analysis, Neurocomputing 69 (2005) 123–141.

[32] P. Jylänki, J. Vanhatalo, A. Vehtari, Robust Gaussian process regression with a student-t likelihood, J. Mach. Learn. Res. 12 (2011) 3227–3257.

[33] W.M. Shao, Z.Q. Ge, Z.H. Song, K. Wang, Nonlinear industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines, Contr. Eng. Pract. 91 (2019) 104098.

[34] A. Daemi, H. Kodamana, B. Huang, Gaussian process modelling with Gaussian mixture likelihood, J. Process Contr. 81 (2019) 209–220.

[35] S. Khatibisepehr, B. Huang, Dealing with irregular data in soft Sensors: bayesian method and comparative study, Ind. Eng. Chem. Res. 22 (2008) 8713–8723.

[36] N. Sammaknejad, Y.J. Zhao, B. Huang, A review of the expectation maximization algorithm in data-driven process identification, J. Process Contr. 73 (2019) 123–136.

[37] R.B. Gopaluni, A particle filter approach to identification of nonlinear process under missing observations, Can. J. Chem. Eng. 86 (2018) 1 081–1092.

[38] J. Deng, B. Huang, Identification of nonlinear parameter varying systems with missing output data, AIChE J. 58 (2012) 3454–3467.

[39] X.F. Yan, J. Wang, Q.C. Jiang, Deep relevant representation learning for soft sensing, Inf. Sci. 514 (2020) 263–274.

[40] Q.C. Jiang, S.F. Yan, H. Cheng, X.F. Yan, Local-global modeling and distributed computing framework for nonlinear plant-wide process monitoring with industrial big data, IEEE Trans. Neur. Net. Lear. (2020), https://doi.org/10.1109/TNNLS.2020.2985223.

[41] Q.C. Jiang, X.F. Yan, B. Huang, Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes, Ind. Eng. Chem. Res. 58 (2019) 12899–12912.

[42] F. Guo, W.T. Bai, B. Huang, Output-relevant variational autoencoder for JIT soft sensor modeling with missing data, J. Process Contr. 92 (2020) 90–97.

[43] F. Guo, R.M. Xie, B. Huang, A deep learning just-in-time modeling approach for soft sensor based on variational autoencoder, Chemometr. Intell. Lab. Syst. 197 (2020) 103922.

[44] S. Kullback, Information theory and statistics, Am. Math. Mon. 504 (1968) 301.

[45] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. B. 39 (1977) 1–38.

[46] C.J. Wu, On the convergence properties of the EM algorithm, Ann. Stat. (1983) 95–103.

[47] G. Birol, C. Ündey, A. Çinar, A modular simulation package for fed-batch fermentation: penicillin production, Comput. Chem. Eng. 26 (2002) 1553–1565.

[48] X.F. Yuan, L. Li, Y.L. Wang, Nonlinear dynamic soft sensor modeling with supervised long short-term memory network, IEEE Trans. Industr. Inform. 16 (2019) 3168–3176.