Supervised Variational Autoencoders for Soft Sensor Modeling with Missing Data

Ruimin Xie, Nabil Magbool Jan, Kuangrong Hao, *Member, IEEE*, Lei Chen, *Member, IEEE,*
and Biao Huang, *Fellow, IEEE*

***Abstract*—Autoencoder (AE) is a deep neural network that has been widely utilized in process industry owing to its superior abilities of feature extraction and data reconstruction. Recently, assuming the latent variables to be random variables, a probabilistic variant of it called variational autoencoder (VAE), has achieved a major success in different applications. In this paper, we develop two novel sub-models based on deep VAEs (DVAE), which are furtherly utilized to establish a soft sensor framework. By the use of our first sub-model known as supervised DVAE (SDVAE), the distribution information of latent features can be obtained. This is used as a prior of the second sub-model known as modified unsupervised DVAE (MUDVAE). Then, a new soft sensor framework can be constructed by combing the encoder of SDVAE with the decoder of MUDVAE. Since our designed VAE has superior ability in data reconstruction, it also works well under the missing data situation which is common in process industries due to sensor failures. Thus, we extend the proposed soft sensor framework to handle the missing data situation. The effectiveness of our proposed soft sensor frameworks is finally demonstrated via an industrial polymerization dataset.**

***Index Terms*— Variational autoencoder (VAE), supervised deep VAE (SDVAE), modified unsupervised DVAE (MUDVAE), soft sensor frameworks, missing data, melt viscosity index (MVI).**

Manuscript received June 8, 2019; revised September 23, 2019; accepted October 24, 2019. This work was supported in part by the National Key Research and Development Plan from Ministry of Science and Technology under Grant 2016YFB0302701, in part by Natural Science Foundation of Shanghai under Grant 19ZR1402300, in part by National Natural Science Foundation of China under Grant 61603090, in part by the Graduate Innovation Fund in Donghua University under Grant CUSF-DH-D-2018099, and in part by Natural Science and Engineering Research Council of Canada under Grant RGPIN-2017-03833. (*Corresponding author: Kuangrong Hao, Biao Huang.)*

R. Xie, K. Hao and L. Chen are with the College of Information Sciences and Technology, and also with Engineering Research Center of Digitized Textile & Apparel Technology, Donghua University, Shanghai 201620, China (e-mail: xieruimin0309@foxmail.com; krhao@dhu.edu.cn; leichen@dhu.edu.cn).

N. Magbool Jan is with the Department of Chemical Engineering, Indian Institute of Technology Tirupati, Tirupati 517506, India (e-mail: nabil@iittp.ac.in; magboolj@ualberta.ca).

B. Huang is with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta, T6G2V4, Canada (e-mail: biao.huang@uablerta.ca).

I. Introduction

In modern industrial processes, automatic control and efficient monitoring strategies are highly desired for improving the overall process efficiency. Successful implementations of these strategies require reliable measuring devices [1] [2]. Sometimes physical sensing devices that can measure quality variables of interest are hard to implement due to extreme working environment or requirement of high maintenance cost to operate online. Hence, soft sensors, a kind of virtual sensing techniques, are used to estimate key quality variables by utilizing easy-to-measure process variables. They are widely applied to perform various industrial tasks such as estimation, process monitoring and fault diagnosis [3]-[6]. Depending on how they are modeled, soft sensor models can be broadly categorized into first-principles based models and data-driven models. Owing to the use of distributed control system in process industries, enormous amount of process data is collected and analyzed, which makes modeling based on data-driven approaches popular in practice. Among these data-driven soft sensor models, some traditional machine learning methods, such as principal component regression (PCR) [7][8] and partial least squares (PLS) [9] are commonly employed in the process industry. However, these algorithms are based on linear feature representations, which may be effective for simple processes but might yield limited prediction capability for complex systems. On the other hand, slow feature analysis methods are developed to address the soft sensor modeling for complex industrial systems. For example, by the sufficient use of temporal coherence, slow feature-based strategies are developed in [10]-[12]. In addition, owing to the superior nonlinear feature representations ability of neural network algorithms, some prominent neural network algorithms are also becoming the mainstream methods in complex soft sensor modeling. Some of the recent ones include hierarchical extreme learning machine (HELM) [13] and long-short term memory (LSTM) [14]. Especially the family of autoencoders, such as deep autoencoders (DAE) [13], denoising autoencoders (DN-AE) [15] [16], stacked autoencoders (SAE) [17] [18] and sparse autoencoders (SpAE) [19], have been widely applied. It should be noted that the above mentioned methods are based on deterministic autoencoder. In this work, we aim at developing soft sensor framework using its probabilistic counterpart and demonstrate its significance.

In general, the complete data points, which consist of the pairs of sample (process variable) and label (key quality variable), are required to build a soft sensor model. However, due to sensor failures (especially in harsh processing plants), missing data in multidimensional sample space is very common. To deal with it, there exists two main classes of repreparing methods – downsampling and imputation. Downsampling involves deletion of missing data records whereas imputation involves obtaining an estimate at the missing data instants. Though downsampling is simple to apply, it is useful only when there are a small number of missing data or when the number of training samples are high [20]. On the other hand, imputation techniques can be single imputation (such as mean replacement) or multiple imputation which is often an iterative model-based approach [21]. The most common multiple imputation approach is probabilistic principal component analysis (PPCA) [22]. Recently, a number of modern approaches such as autoencoders (AE) [23], extreme learning machine auto-encoder (ELM-AE) [24] and variational autoencoders (VAE) [25] have been used as multiple imputation techniques in various applications.

AE was first proposed in 1986 by Rumelhart to obtain learning representations [26]. Its representation ability is not strong due to single-hidden-layered structure until when Hinton improved its structure and used greedy layer-wise training approach to retrain hidden layers [27]. Since then, AEs have become prevalent and many variants have been developed. AE model maps observations into latent variables through a forward network (encoder) and reconstruct observations with latent variables through another forward network (decoder). Although the performance of AE is superior, its latent space is represented by data, which may lead to poor robustness. To this end, a probabilistic counterpart of AEs, known as variational autoencoders, was developed in 2014 [28]. VAE, also known as auto-encoding variational Bayes, combines the strengths of variational Bayesian and neural networks. On the one hand, it regards neural networks as powerful functional approximators through backpropagation to solve the intractable posterior problem using variational inference. On the other hand, the variational Bayes framework allows for a probabilistic interpretation and model the complex process nonlinearity in terms of distributions. It should be noted that unlike the conventional AE which learns a latent space represented by data, vanilla VAE learns feature extraction by assuming the latent distribution as standard normal distribution.

In this paper, we propose two novel sub-models based on deep variational autoencoders (DVAE). And then based on these two sub-models, two new soft sensor frameworks are developed to deal with missing data and no missing data scenarios. First, a supervised DVAE (SDVAE) is developed to learn the latent distribution which not only does the dimensionality reduction of sample space, but also can represent the relationship between samples and labels. Second, a modified unsupervised DVAE (MUDVAE) is presented whose benchmark latent distribution is modified to SDVAE's learnt latent distribution. The main motivation for this modification is that when the KL divergence between MUDVAE's learnt latent distribution and the modified benchmark latent distribution is small enough, sampling from MUDVAE's latent distribution is equivalent to sampling from SDVAE's. This enable us to obtain predictions for test samples. Then, combining the SDVAE's encoder and MUDVAE's decoder, the first new soft sensor framework is developed. Furthermore, to fully utilize the data reconstruction ability of the proposed DVAE, another soft sensor framework is developed to deal with missing data problem.

The remainder of this paper is organized as follows. Section II briefly outlines the fundamentals of VAE. Section III proposes a novel soft sensor framework by developing a supervised DVEA and modified unsupervised DVAE models. This section also discusses a suitable modification to the framework to deal with missing data. Then, Section IV exemplifies the proposed soft sensor design scheme to predict melt viscosity index of an industrial polymerization process. Finally, conclusions are presented in Section V.

II. Variational Autoencoder

Variational autoencoder is a popular unsupervised learning method in the realm of deep learning. It has found tremendous applications in image processing [28], [29]. It is a concoction of neural networks and probabilistic inference through variational Bayesian. The basic idea of VAE can be presented as follows [28].

VAE belongs to a class of generative model which maps the complicated observation space x onto a relatively simple latent space z. Then, the marginal likelihood is

$$p_\theta(x) = \int p_\theta(z)\, p_\theta(x|z)dz. \tag{1}$$

Unfortunately, the latent variable z and the generative model parameter θ are unknown, therefore, the integrand of the marginal likelihood is intractable and true posterior given by

$$p_\theta(z|x) = p_\theta(z)p_\theta(x|z)/p_\theta(x) \tag{2}$$

is also intractable. In order to solve the problems, a recognition model $q_\emptyset(z|x)$ is introduced using the idea of variational inference to approximate the true posterior $p_\theta(z|x)$. VAE jointly learns the parameters θ and $\emptyset$ by utilizing backpropagation of neural networks.

The marginal log-likelihood can be written as:

$$\mathrm{log}p_\theta(x) = \mathrm{KL}[q_\emptyset(z|x)\|p_\theta(z|x)] + \mathcal{L}(\theta,\emptyset;x), \tag{3}$$

where the first term in the right-hand side (RHS) is the KL divergence of the approximation from the true posterior, the second RHS term is the evidence lower bound on the marginal likelihood of data point x. The evidence lower bound can be expressed as:

$$\mathcal{L}(\theta,\emptyset;x) = -\mathrm{KL}[q_\emptyset(z|x)\|p_\theta(z)] + \mathbb{E}_{q_\emptyset(z|x)}[\mathrm{log}p_\theta(x|z)]. \tag{4}$$

To optimize the lower bound with respect to the variational parameters $\emptyset$ and generative distribution parameters θ, a practical estimator called stochastic gradient variational Bayes (SGVB) estimator $\tilde{\mathcal{L}}(\theta,\emptyset;x)$ is used which approximates the second term of evidence lower bound. Now, the approximated lower bound function is given by:

$$\tilde{\mathcal{L}}(\theta,\emptyset;x) = -\mathrm{KL}[q_\emptyset(z|x)\|p_\theta(z)] + \frac{1}{L}\sum_{l=1}^{L}\mathrm{log}p_\theta\left(x|z^{(l)}\right) \tag{5}$$

where

$$z^{(l)}=g_{\emptyset}\left(\epsilon^{(l)},x\right)\sim q_{\emptyset}(z|x),\text{ and }\epsilon^{(l)}\sim p(\epsilon).\quad(6)$$

Here ϵ is an auxiliary variable with independent marginal distribution $p(\epsilon)$, and $g_{\emptyset}$ is some continuous vector-valued function parameterized by $\emptyset$, l is the number of sampling. Eq. (6) is the well-known reparameterization trick in which sampling from the distribution $z\sim q_{\emptyset}(z|x)$ is parametrized as $z=g_{\emptyset}(\epsilon,x)$, $\epsilon\sim p(\epsilon)$[28]. In other words, a non-continuous sampling option (the gradient of $\emptyset$ is non-existent) is changed to a continuous option (the gradient about $\emptyset$ is existent). Owing to this, the backpropagation can be implemented to learn the parameters θ and $\emptyset$ efficiently.

Vanilla VAE assumes that prior distribution $p_{\theta}(z)$ is a standard multivariate Gaussian distribution $\mathcal{N}(z;0,I)$ (also called benchmark distribution), the true posterior $p_{\theta}(z|x)$ is also a multivariate Gaussian. In this case, the approximate posterior is assumed to be a multivariate Gaussian with an isotropic covariance:

$$\log q_{\emptyset}(z|x)=\log\mathcal{N}(z;\mu,\sigma^{2}I)\quad(7)$$

where μ and σ are the variational mean and standard deviation. Let μ_j and σ_j denote the j-th element of the mean and standard deviation vectors, then:

$$-\mathrm{KL}[q_{\emptyset}(z|x)\|p_{\theta}(z)]$$
$$=\frac{1}{2}\sum_{j=1}^{J}\left(1+log\sigma_j^2-\mu_j^2-\sigma_j^2\right)\quad(8)$$

The resulting estimator for this model at any data point x is:

$$\tilde{\mathcal{L}}(\theta,\emptyset;x)\approx\frac{1}{2}\sum_{j=1}^{J}\left(1+log\sigma_j^2-\mu_j^2-\sigma_j^2\right)$$
$$+\frac{1}{L}\sum_{l=1}^{L}\log p_{\theta}\left(x\middle|z^{(l)}\right)\quad(9)$$

where $z^{(l)}=\mu+\sigma\odot\epsilon^{(l)}$, $\epsilon^{(l)}\sim\mathcal{N}(0,I)$ [29], and $\odot$ denotes the element-wise multiplication.

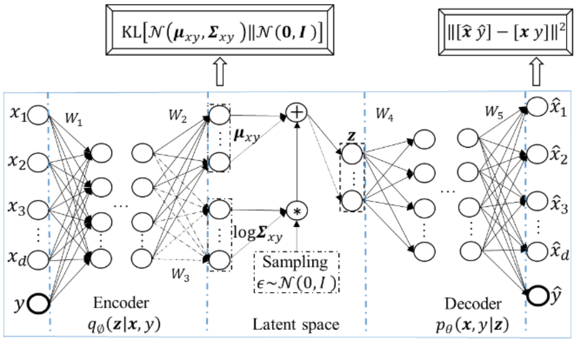
From the perspective of neural networks, the recognition model $q_{\emptyset}(z|x)$ is referred to as a probabilistic encoder which produces a distribution over latent code z given datapoint x. The variational mean μ and log-deviation $log\sigma^2$ are the outputs of the encoder neural network, and variational parameters $\emptyset$ are the encoder weights. Similarly, the generative model $p_{\theta}(x|z)$ can be regarded as a probabilistic decoder which produces a distribution over the possible corresponding values of x given code z. The generative parameters θ are the decoder weights.

III. Deep VAEs

In this section, we first propose a supervised deep VAE (SDVAE) model. Then a modified unsupervised deep VAE (MUDVAE) model is constructed by replacing the original fixed prior with the learned latent distribution in SDVAE. Based on these two novel sub-models, a new soft sensor framework is developed. Further, to fully utilize the data reconstruction ability of VAE, the proposed soft sensor framework is extended to solving the problem of missing data.

A. SDVAE

The traditional vanilla VAE, as discussed in the previous section, is an unsupervised method where only samples are considered. In this subsection, we modify the vanilla VAE by considering both samples and labels. The proposed SDVAE model is shown in Figure 1, label y has been added as an extra dimension of sample x. Hence, SDVAE utilizes its probabilistic encoder to abstract the features that constitute the latent distribution, and then sample from the latent distribution to reconstruct the sample-label pairs using the probabilistic decoder. Both the encoder and decoder are multi-hidden-layered fully connected neural networks.



$\mathrm{KL}[\mathcal{N}(\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})\|\mathcal{N}(\mathbf{0},\boldsymbol{I})]$
$\|[\hat{\boldsymbol{x}}\ \hat{y}]-[\boldsymbol{x}\ y]\|^2$
x_1
x_2
x_3
x_d
y
W_1
W_2
W_3
$\boldsymbol{\mu}_{xy}$
$\log\boldsymbol{\Sigma}_{xy}$
Sampling
$\epsilon\sim\mathcal{N}(0,I)$
$\boldsymbol{z}$
W_4
W_5
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_d$
$\hat{y}$
Encoder
$q_{\emptyset}(\boldsymbol{z}|\boldsymbol{x},y)$
Latent space
Decoder
$p_{\theta}(\boldsymbol{x},y|\boldsymbol{z})$

Fig. 1. Supervised deep variational autoencoder model.

In this model, the prior distribution $p_{\theta}(z)$ is assumed to be a standard multivariate Gaussian distribution $\mathcal{N}(z;\mathbf{0},\boldsymbol{I})$, and the approximate posterior $q_{\emptyset}(z|x,y)$ is assumed to be a multivariate Gaussian with a diagonal covariance $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$; then the KL divergence between the approximate posterior and prior is:

$$-\mathrm{KL}[q_{\emptyset}(z|x,y)\|p_{\theta}(z)]$$
$$=-\int\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})\log\frac{\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})}{\mathcal{N}(z;\mathbf{0},\boldsymbol{I})}dz$$
$$=\frac{1}{2}\sum_{j=1}^{J}\left(1+\log(\sigma_{xy}^{(j)})^2-(\mu_{xy}^{(j)})^2-(\sigma_{xy}^{(j)})^2\right)\quad(10)$$

where the vector $\boldsymbol{\mu}_{xy}$ and covariance matrix $\boldsymbol{\Sigma}_{xy}$ (or $\log\boldsymbol{\Sigma}_{xy}$) are learned from SDVAE encoder, that is to say, they are outputs of SDVAE encoder. $\mu_{xy}^{(j)}$ and $\sigma_{xy}^{(j)}$ denote the j-th element of mean and standard deviation vectors. Then the sampled z from $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$ is the input to SDVAE's decoder to reconstruct the samples and label.

Last, the reparameterization trick that $z\sim q_{\emptyset}(z|x,y)$ is reparameterized to $z=g_{\emptyset}(\epsilon,x,y)$, $\epsilon\sim p(\epsilon)$ is applied. Because z is assumed to be Gaussian distribution, $g_{\emptyset}$ can be defined as $z=\boldsymbol{\mu}_{xy}+\boldsymbol{\sigma}_{xy}\odot\epsilon$, and $\epsilon\sim\mathcal{N}(0,I)$. Then the SGVB lower bound estimator (the objective function of the neural network) is given by:

$$\tilde{\mathcal{L}}_1(\theta,\emptyset;x,y)$$
$$=-\mathrm{KL}[q_{\emptyset}(z|x,y)\|p_{\theta}(z)]+\mathbb{E}_{q_{\emptyset}(z|x,y)}[\log p_{\theta}(x,y|z)]$$
$$=\frac{1}{2}\sum_{j=1}^{J}\left(1+\log(\sigma_{xy}^{(j)})^2-(\mu_{xy}^{(j)})^2-(\sigma_{xy}^{(j)})^2\right)$$
$$+\frac{1}{L}\sum_{l=1}^{L}\log p_{\theta}\left(x,y\middle|z^{(l)}\right)\quad(11)$$

where $z^{(l)}=\boldsymbol{\mu}_{xy}+\boldsymbol{\sigma}_{xy}\odot\epsilon^{(l)}$, and $\epsilon^{(l)}\sim\mathcal{N}(0,I)$. L is the number of latent samples used in Monte-carlo approximation.

B. MUDVAE

In this subsection, we propose the MUDVAE model, whose structure is shown in Figure 2. Similar to the encoder and decoder of a vanilla VAE, the MUDVAE model is also multi-hidden-layered fully connected neural networks, but the latent distribution has been modified to capture the latent space of SDVAE model.

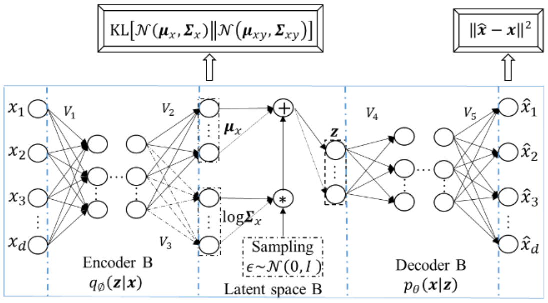

$\mathrm{KL}[\mathcal{N}(\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)\|\mathcal{N}(\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})]$
$\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|^2$
x_1
x_2
x_3
x_d
V_1
V_2
V_3
$\boldsymbol{\mu}_x$
$\log\boldsymbol{\Sigma}_x$
$\boldsymbol{z}$
V_4
V_5
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_d$
Sampling
$\epsilon\sim\mathcal{N}(0,I)$
Encoder B
$q_\emptyset(\boldsymbol{z}|\boldsymbol{x})$
Latent space B
Decoder B
$p_\theta(\boldsymbol{x}|\boldsymbol{z})$

Fig. 2. Modified unsupervised deep variational autoencoder model.

Hence, the prior distribution $p_\theta(z)$ in this case is assumed to be $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$ which is learned from the well-trained SDVAE model. That is to say, approximating posterior $q_\emptyset(z|x)$ follows $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy}\ )$. Thus, the modified KL divergence is expressed as:

$$-\mathrm{KL}[q_\emptyset(z|x)\|p_\theta(z)]$$
$$=-\int\mathcal{N}(z;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x\ )\log\frac{\mathcal{N}(z;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x\ )}{\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy}\ )}dz$$
$$=\frac{1}{2}\sum_{j=1}^{J}\left(1+\log\frac{(\sigma_x^{(j)})^2}{(\sigma_{xy}^{(j)})^2}-\frac{(\mu_x^{(j)}-\mu_{xy}^{(j)})^2}{(\sigma_{xy}^{(j)})^2}-\frac{(\sigma_x^{(j)})^2}{(\sigma_{xy}^{(j)})^2}\right)\qquad(12)$$

where the vector $\boldsymbol{\mu}_x$ and the covariance matrix $\boldsymbol{\Sigma}_x$ (or $\log\boldsymbol{\Sigma}_x$) are learned by MUDVAE's encoder, that is to say, they are the outputs of MUDVAE's encoder. It is important to note that $\boldsymbol{\mu}_{xy}$ and $\boldsymbol{\Sigma}_{xy}$ are obtained from well-trained SDVAE model. Then sampled z from $\mathcal{N}(z;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x\ )$ is the input to MUDVAE's decoder to reconstruct samples.

It is important to note that in vanilla VAE any distribution in d dimensions can be generated by taking a set of d variables that are normally distributed and map them through a sufficiently complicated function. Since the underlying distribution of process data is not often known, the vanilla VAE considers the standard normal distribution as the latent distribution to map it from the observation space. Hence, if powerful function approximators exist, we can simply learn a function which maps our independent, normally distributed z values to reconstruct samples. Since the vanilla VAE takes an unusual approach to learn representations, there is no simple interpretation of the dimensions of z, and in fact, the dimension of z is a hyperparameter in VAE which is often manually set.

It is important to emphasize that the prior distribution in MUDVAE model is $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy}\ )$. It can also be seen as an intermediate distribution. Further, the dimension of z distributed by $\mathcal{N}(z;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)$ in MUDVAE model should have the same dimension as in the latent space of SDVAE model, that is, $\mathcal{N}(z;\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$.

Then SGVB lower bound estimator (objective function) of MUDVAE model is:

$$\tilde{\mathcal{L}}_2(\theta,\emptyset;x)$$
$$=-\mathrm{KL}[q_\emptyset(z|x)\|p_\theta(z)]+\mathbb{E}_{q_\emptyset(z|x)}[\log p_\theta(x|z)]$$
$$=\frac{1}{2}\sum_{j=1}^{J}\left(1+\log\frac{\left(\sigma_x^{(j)}\right)^2}{\left(\sigma_{xy}^{(j)}\right)^2}-\frac{\left(\mu_x^{(j)}-\mu_{xy}^{(j)}\right)^2}{\left(\sigma_{xy}^{(j)}\right)^2}-\frac{\left(\sigma_x^{(j)}\right)^2}{\left(\sigma_{xy}^{(j)}\right)^2}\right)$$
$$+\frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(x|z^{(l)}\right)\qquad(13)$$

where $z^{(l)}=\boldsymbol{\mu}_x+\boldsymbol{\sigma}_x\odot\epsilon^{(l)}$, and $\epsilon^{(l)}\sim\mathcal{N}(0,I)$. L is the sampling number.

C. DVAEs-based Soft Sensor

In this subsection, we present our proposed DVAE-based soft sensor model. The basic idea of this framework is that first learn the latent representation of the sample-label pairs and then using the learned latent distribution as the benchmark distribution to train the encoder model of DVAE just utilizing samples, as is common done in vanilla VAE. The general soft sensor framework is presented in Figure 3. The main steps of the proposed approach are as follows:

Step1: Train the SDVAE model using complete data points as discussed in subsection III.A;

Step2: Construct the decoder of the soft sensor model from SDVAE's decoder parameters (i.e. weights and bias);

Step3: Train the MUDVAE model using all samples with the learned SDVAE's distribution, $(\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$ as the benchmark distribution following subsection III.B.

Step4: Construct the encoder of the soft sensor model from MUDVAE's encoder parameters (i.e. weights and bias);

Step5: Combine MUDVAE's encoder and SDVAE's decoder to form a general soft sensor model.

For any test samples, the MUDVAE's encoder projects onto the corresponding latent distribution $(\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)$ which is close to $(\boldsymbol{\mu}_{xy},\boldsymbol{\Sigma}_{xy})$. Then the SDVAE's decoder reconstructs the samples from the obtained latent space and outputs the predicted values of quality variable for the given test sample. This is accomplished by assuming the dimension of latent distribution in both the SDVAE model and MUDVAE to be the same.

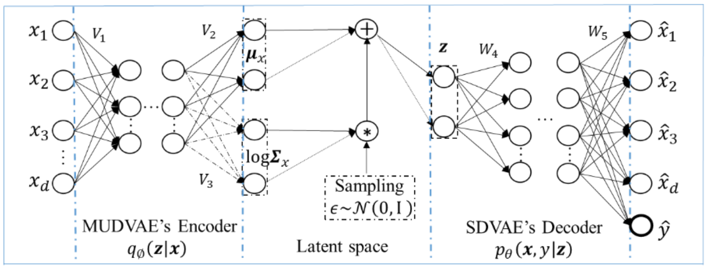


Fig. 3. DVAEs-based soft sensor framework.

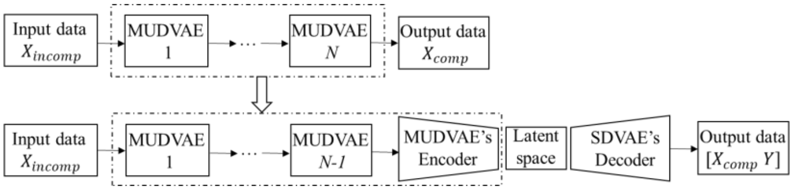


Fig. 4. DVAEs-based soft sensor framework with missing data.

D. DVAEs-based Soft Sensor with missing data

Now, we extend the proposed soft sensor scheme to dealing with missing input data. Owing to the strong ability of reconstruction, VAE has been successfully applied to missing data imputation [25]. The general philosophy for missing data imputation is that to first train the VAE model using complete samples to learn the latent distribution. Now, input missing data samples by randomly initializing and then output reconstructed complete samples. Then the reconstructed samples are used to reinitialize the missing data to produce better reconstruction by iteratively solving for it until the mean error between the successive imputed values is below threshold ε. Since missing data is very common in soft sensor applications, a general practice is to do preprocessing (such as deletion or imputation) and then input the imputed samples to a regression model. Unlike the iterative approach to impute the data, in this work, we propose a stacking VAE model to do input imputation as shown in Figure 4. The main steps of the soft sensor scheme to deal with missing data are detailed here.

Step1: Train and construct the MUDVAE's encoder and SDVAE's decoder to schematize the soft sensor model.

Step2: Design a stacking-based strategy in which N well-trained MUDVAE models are stacked in series such that only the output of the missing element is propagated between the stacked models. In other words, the non-missing element in the input remains unchanged when it is propagated through series of MUDVAE models. In this work, the number of stacking models is considered as hyper parameter and is set by trial and error.

Step3: Connect N-1 MUDVAEs in order, followed by an MUDVAE's encoder, and finally concatenating a SDVAE's decoder.

In the proposed soft sensor framework, incomplete samples X_{incomp} are input, and are processed through N-1 connected MUDVAEs. Since the goal is to predict the label. Hence, after N-1 connected MUDVAEs only one MUDVAE's encoder is added. From this encoder, we can obtain the most relevant latent features which are materialized to be $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. Due to the small KL divergence between $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ and $\mathcal{N}(\boldsymbol{\mu}_{xy}, \boldsymbol{\Sigma}_{xy})$, sampling from $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ can be equivalent to sampling from $\mathcal{N}(\boldsymbol{\mu}_{xy}, \boldsymbol{\Sigma}_{xy})$. Hence, z sampled from $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ can be fed to SDVAE's decoder to output $[X_{comp}\ Y]$. The proposed soft sensor scheme does not involve any iterative process in the imputation step. It can be simply regarded as a function; when given X_{incomp} , it can yield $[X_{comp}\ Y]$. Hence, this soft sensor framework can efficiently deal with missing data in samples.

IV. Case Study: Application to Polymerization Process

A. Process Description

A simplified process flow diagram of the polymerization process for the production of polyamide/polyester with all major process units is depicted in Figure 5 [30]. There are 15 process variables and 6 process units as described in Table I and Table II, respectively. The process mainly consists of three pivotal reaction kettles, i.e. esterification reactor, pre-polycondensation reactor and a final polycondensation reactor, plus three secondary units, i.e. a calorifier and two condensers. In the esterification reactor, raw materials, purified terephthalic acid (PTA) and ethylene glycol (EG), are chemically reacted, to produce bis-hydroxyethyl terephthalate (BHET). In order to promote the completion of reaction, the excess EG is usually added in the actual production. A mixture of excess EG and product BHET is then fed to pre-polycondensation reactor after being heated to a suitable temperature and removing impurities. Pre-polycondensation reactor is a tower-type upstream tank reactor, consisting of sixteen plates in which the Polycondensation is conducted in an ascending fashion. As reaction proceeds, the EG vapors are generated which plays the role of agitation and also accelerates the reaction rate. A condenser is fixed on the top of the pre-polycondensation reactor to recover the unutilized EG for further recycle. The prepolymer, metered by the metering pump, is transported to the inlet of the final polycondensation reactor which is also called a horizontal squirrel cage stirred tank reactor. Under the action of the agitator, the prepolymer moves from the inlet to the outlet, and the polycondensation occurs as it moves along the reactor, making the polymer more viscous. At the end of the whole polymerization process, a viscometer is employed to measure the quality of the product [30].

The melt viscosity index is a primary indicator of the quality of produced polymer product, and signifies the process efficiency and economics [31]. However, it is difficult to measure it precisely in real-time for two reasons. First, the high malfunction probability of hardware sensor of melt viscosity index indeed increases the maintenance cost. Second, the process variables contain a lot of missing data. Thus, the conventional soft sensor models tend to fail often due to unavailable or inaccurate inputs. Therefore, in this work, we have designed a VAE-based soft sensor that can predict the melt viscosity index even in the case of missing process variables.

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.2019.2951622, IEEE Transactions on Industrial Informatics

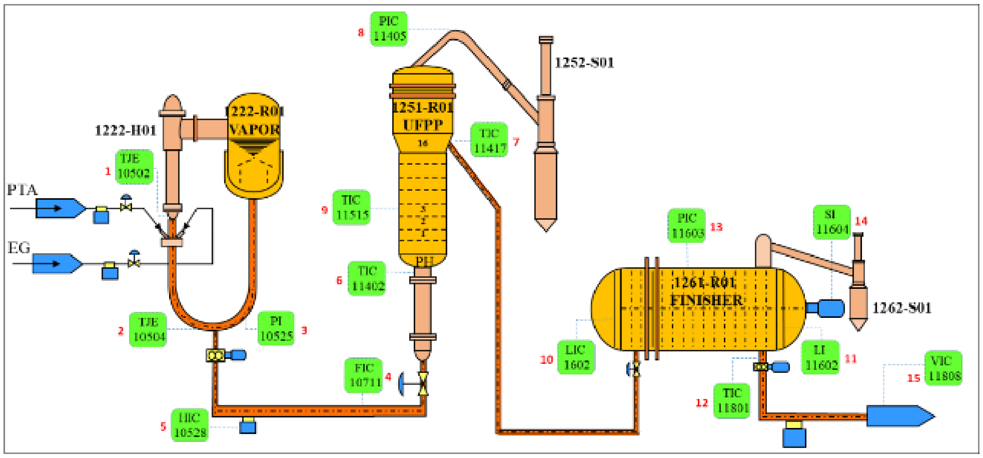


Fig. 5. Flowchart of the polymerization process.

TABLE I
PROCESS VARIABLES IN THE FLOWCHART.

No.	Tag Name	Description
1	TJE-10502	Front temperature of injection sizing agents
2	TJE-10504	Rear temperature of injection sizing agents
3	PI-10525	Pressure of siphon
4	FIC-10711	Flow of oligomer
5	HIC-10528	Pump speed of oligomer
6	TIC-11402	Outlet temperature of UFPP PH
7	TIC-11417	Outlet temperature of UFPP 16th
8	PIC-11405	Pressure of UFPP
9	TIC-11515	Spray temperature of UFPP EG
10	LIC-11602	Inlet liquid level of FINISHER
11	LI-11602	Outlet liquid level of FINISHER
12	TIC-11801	Outlet temperature of FINISHER
13	PIC-11603	Pressure of FINISHER
14	SI-11604	Agitator rotational velocity of FINISHER
15	VIC-11808	Melt viscosity

TABLE II
PROCESS UNITS IN THE FLOWCHART.

No.	Tag Name	Description
1	1222-H01	Calorifier
2	1222-R01	Esterification reactor
3	1251-R01	Pre-polycondensation reactor
4	1252-S01	Condenser
5	1261-R01	Final polycondensation reactor
6	1262-S01	Condenser

B. Dataset Description

Industrial data are collected from the data historian of distributed control system (DCS) of the discussed polyester plant in China. Sampling interval of the obtained data is one second. One key variable is the melt viscosity index. Fourteen secondary variables that include temperatures, pressures and material flows at different points in the plant are chosen as explanatory variables for soft sensor modeling based on the operator experience and process knowledge. 10000 datapoints have been collected in our dataset.

In order to verify the effectiveness of the proposed soft sensor models, we first divide the original dataset into three parts. The first 80% is used as training set, the next 10% consists of validation set and the last 10% is regarded as testing set. This industrial dataset is used to elucidate the proposed framework. In order to demonstrate the new soft sensor framework under missing data cases, three levels (i.e., light, medium and heavy) of missing data are considered in this application. The corruption degree/ratio of three situations are 10%, 30%, and 50%, respectively. The partition of training, validation and testing sets are same as before.

C. Experimental Setups

To utilize the proposed soft sensor frameworks, the first step is to determine two sub-models' structures. Obviously, there are 15 and 14 neurons for input layers of SDVAE and MUDVAE sub-models respectively since the sample has 14 dimensions and label is one dimensional, and the number of neurons for output layers of each sub-model are the same as their corresponding input dimensions. The hyperparameters are set according to trial and error technique. Thus, the SDVAE's encoder has 6 hidden layers and each of them is equipped with 13, 10, 8, 6, 5 and 4 neurons, the latent space is 3 dimensions, and the decoder's structure is set symmetric to its encoder. The MUDVAE's encoder consists of 4 hidden layers and each of them is equipped with 10, 8, 6 and 4 neurons respectively, the dimension of the latent space is the same as SDVAE's, and the decoder's structure is also symmetric to its encoder. The learning rates of these two models are both 0.01, and the optimizers are stochastic gradient descents. The activation functions of each hidden layer are either ReLU or Sigmoid. Once the structures of SDVAE and MUDVAE are determined, the soft sensor model is constructed using SDVAE's decoder and MUDVAE's encoder. To deal with missing data, MUDVAE is utilized to impute missing data, and the number of models stacked in series for imputation is nine. The raw datapoints are preprocessed by MaxMinScaler and shuffle techniques in deep learning.

D. Results and Analysis

To demonstrate the effectiveness and flexibility of the proposed soft sensor frameworks, two groups of comparative experiments are implemented. The first group includes multilayer NN model, AE-NN, VAE-NN and the proposed DVAEs-based model without considering missing data 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.2019.2951622, IEEE Transactions on Industrial Informatics

samples. Here, multilayer NN has neuron layer structure of [14 12 10 8 6 3 1], AE-NN and VAE-NN utilize AE/VAE to abstract latent features and then use NN to build a regression based on these features. The structure of AE/VAE part is [14 8 5 3 5 8 14] and the NN part is [3 2 1]. The prediction MSE, MAE and R^2 on the testing dataset are given in Table III.

TABLE III
PREDICTION RESULTS OF FOUR COMPETING APPROACHES.

Methods	MSE	MAE	R^2
multilayer NN	0.1262	0.1694	0.7213
AE-NN	0.0821	0.0871	0.8129
VAE-NN	0.0683	0.0752	0.8776
Proposed model	0.0319	0.0463	0.9195

As we can see, multilayer NN gives the worst prediction performance. The potential reason is that its ability to obtain nonlinear representation is not good enough for such high dimensional industrial data. By adopting the encoding and decoding structures to abstract latent features (equivalent to doing some dimensionality reduction) and then implementing regression operation, the latter three methods can learn the complicate data relationships better than simple NN. Moreover, AE-NN yields poor prediction result compared to VAE-NN. This can be attributed to the fact that the former model represents the latent space using data points whereas, in the latter model, latent space is represented by the distribution of features. However, no model can guarantee all latent features are learnt completely since the latent features learned are not output relevant. Hence, the performance of even VAE-NN, is no better than our proposed model which uses a supervised deep VAE to learn the regression relationship and latent distribution simultaneously.

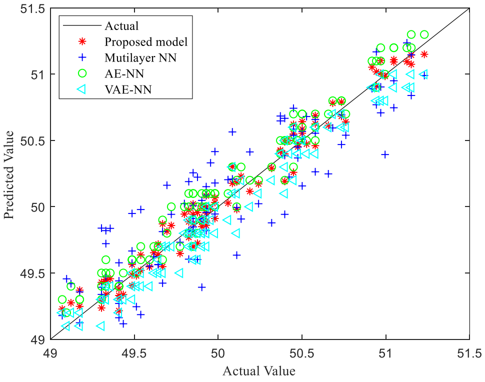


Fig. 6. Prediction performance of different models.

For a more intuitive comparison, the prediction results of the above models are shown in Figure 6. As shown, the x-axis and y-axis denote real and predicted values respectively. The black solid line reflects the ideal relation between real and predicted values. The closer the point is to the black line, the better the performance of the model. Hence, it can be inferred that the proposed model has the highest prediction accuracy and the multilayer NN poses the least prediction capability.

In second group of experiments, the goal is to demonstrate the superiority of the proposed model for the case of missing data. Following the traditional practice, three common solutions of missing data - deletion, mean imputation and PCA imputation - have been used to deal with missing data imputation. Then the VAE-NN, whose prediction performance is next only to our proposed general soft sensor model, is utilized to implement regression prediction. Above three methods are compared with our proposed soft sensor framework with missing data. The prediction MSE and R^2 for three corruption levels of testing datasets are shown in Figure 7. The red lines represent the MSE obtained using different approaches for various missing levels, and the blue lines denote the corresponding R^2 values. Based on the simulation results, the proposed model has shown the lowest MSE and highest R^2 values for all three missing levels considered. In particular, its superiority is more obvious when the corruption level is high. This can be attributed to VAE's strong abilities to feature representation and reconstruction. Moreover, it is clear that the simple deletion method shows poor performance due to inefficient learning of complex relationship hidden in data. The multiple imputation PCA is better than mean replacement and it is consistent with most literatures.

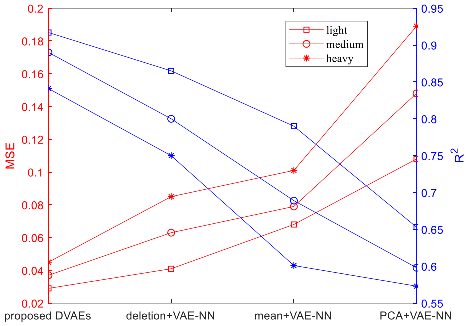


Fig. 7. Comparison of MSE and R^2 values.

As an example, the trends of actual and predicted values for light corruption level dataset are shown in Figure 8.

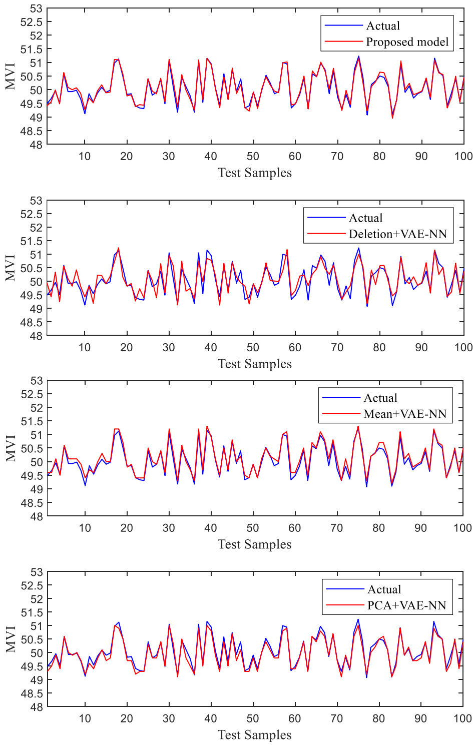


Fig. 8. Prediction results using different methods for medium corruption level

V. Conclusion

In this paper, taking the superior abilities of VAE in feature extraction and data reconstruction into consideration, two soft sensor frameworks based on novel deep VAEs were proposed for solving both complete and missing data problems respectively. The first DVAE-based soft sensor framework takes advantage of the unsupervised feature extraction ability of the proposed MUDVAE sub-model and the supervised regression ability of the proposed SDVAE sub-model to predict the key quality variables of the process. Further, to make full use of the data reconstruction ability of VAEs, the second DVAE-based soft sensor framework is designed to solving the problem of missing data. The effectiveness of the proposed soft sensor frameworks was shown through the case study on an actual polymerization process, and their superiority was demonstrated by comparison with other existing models.

References

[1]. B. Huang, Y. Qi, and A. M. Murshed, *Dynamic Modeling and Predictive Control in Solid Oxide Fuel Cells: First Principle and Data-based Approaches*, ISBN: 978-0-470-97391-2, John Wiley & Sons, 2013.

[2]. S. X. Ding, *Data-Driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. London, U.K.: Springer-Verlag, 2014.

[3]. H. Chen and B. Jiang, "A review of fault detection and diagnosis for the traction system in high-speed trains," *IEEE Trans. Intell. Transp. Syst.*, 2019, doi.10.1109/TITS.2019.2897583.

[4]. Y. Gao, F. Xiao, J. Liu and R. Wang, "Distributed soft fault detection for interval type-2 fuzzy-model-based Stochastic Systems with Wireless Sensor Networks," *IEEE Trans. Ind. Inform.*, vol. 15, no. 1, pp. 334-347, Jan. 2019.

[5]. X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song and W. Gui, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508-1517, Feb. 2018.

[6]. W. Shao, L. Yao, Z. Ge and Z. Song, "Parallel computing and SGD-based DPMM for soft sensor development with large-scale semisupervised data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6362-6373, Aug. 2019.

[7]. Z. Ge, "Mixture Bayesian regularization of PCR model and soft sensing application," *IEEE Trans. Ind. Electron*., vol. 62, no. 7, pp. 4336-4343, Jul. 2015.

[8]. H. Chen, B. Jiang, N. Lu, and Z. Mao, "Deep PCA based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains," *IEEE Trans. Veh. Technol*., vol. 67, no. 6, pp. 4819-4830, Jun. 2018.

[9]. R. Muradore, P. Fiorini, "A PLS-based statistical approach for fault detection and isolation of robotic manipulators," *IEEE Trans. Ind. Electron*., vol. 59, no. 8, pp. 3167-3175, Aug. 2012.

[10]. W. Yu and C. Zhao, "Recursive exponential slow feature analysis for fine-scale adaptive processes monitoring with comprehensive operation status identification," *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3311-3323, Jun. 2019.

[11]. C. Shang, B. Huang, F. Yang and D. Huang, " Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling," *AIChE J.*, vol. 61, no. 12, pp. 4126-4139, Dec. 2015.

[12]. Y. Ma, S. Zhao and B. Huang, "Feature extraction of constrained dynamic latent variables," *IEEE Trans. Ind. Inform.*, 2019, doi. 10.1109/TII.2019.2901934.

[13]. L. Yao and Z. Ge, "Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1490-1498, Feb. 2018.

[14]. X. Yuan, L. Li and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," *IEEE Trans. Ind. Inform*., 2019, doi: 10.1109/TII.2019.2902129.

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

[16]. X. Yuan, B. Huang, Y. Wang, C. Yang and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Trans. Ind. Inform*., vol. 14, no. 7, pp. 3235-3243, Jul. 2018.

[17]. W. Yu and C. Zhao, "Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net," *IEEE Trans on Control Syst. Technol.*, 2019, doi: 10.1109/TCST.2019.2897946.

[18]. J. Wang and X. Yan, "Mutual information-weighted principle components identified from the depth features of stacked autoencoders and original variables for oil dry point soft sensor," *IEEE Access*, vol. 7, pp. 1981-1990, Dec. 2018.

[19]. C. Li, W. Zhang, G. Peng and S. Liu, "Bearing fault diagnosis using fully-connected winner-take-all autoencoder," *IEEE Access*, vol. 6, pp. 6103-6115, 2018.

[20]. M. S. Osman, A. M. Abu-Mahfouz and P. R. Page, "A survey on data imputation techniques: water distribution system as a use case," *IEEE Access*, vol. 6, pp. 63279-63291, Jun. 2017.

[21]. M. P. Gómez-Carracedo, J. M. Andrade, P. López-Mahía, S. Muniategui and D. Prada, "A practical comparison of single and multiple imputation methods to handle complex missing data in air quality datasets," *Chemom. Intell. Lab. Syst*., vol. 134, pp. 23-33, 2014.

[22]. S. Dray, J. Josse, "Principal component analysis with missing values: a comparative survey of methods," *Plant Ecol*. vol. 216, pp. 657–667, May. 2015.

[23]. V. Miranda, J. Krstulovic, H. Keko, C. Moreira and J. Pereira, "Reconstructing missing data in state estimation with autoencoders," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 604-611, Dec. 2011.

[24]. C. Lu and Y. Mei, "An imputation method for missing data based on an extreme learning machine auto-encoder," *IEEE Access*, vol. 6, pp. 52930-52935, Sept. 2018.

[25]. J. T. McCoy, "Variational autoencoders for missing data imputation with application to a simulated milling circuit," *in IFAC PapersOnLine*, vol. 51, no. 21, pp. 141–146, Sept. 2018.

[26]. D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[27]. G. E. Hinton and R. R. Salakhutdinov, "Redecing the dimensionality of data with neural networks," *Sci.*, vol. 313, pp. 504-507, Jul. 2006.

[28]. D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv Prepr*. arXiv1312.6114, Dec. 2013.

[29]. K. Wang, M. G. Forbes, B. Gopaluni, J. Chen and Z. Song, "Systematic development of a new variational autoencoder model based on uncertain data for monitoring nonlinear processes," *IEEE Access*, vol. 7, pp. 22554-22565, Feb. 2019.

[30]. R. Xie, K. Hao, B. Huang, L. Chen, X. Cai, "Data-driven modeling based on two-stream λ gated recurrent unit network with soft sensor application," *IEEE Trans. Ind. Electron*., 2019, doi. 10.1109/TIE.2019.2927197.

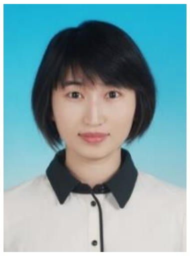
[31]. M. Zhang, B. B. Zhao, X.G. Liu, "Predicting industrial polymer melt index prediction via incorporating chaotic characters into Chou's general PseAAC," *Chemom. Intell. Lab. Syst.*, vol. 146. pp. 232-240, May. 2015.

Ruimin Xie is currently a Ph.D. candidate in Control Science and Engineering.at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She obtained the B.S. degree in Mathematics and Applied Mathematics from Jiangsu Normal University, Xuzhou, Jiangsu, China, in 2015. From September 2015 to March 2017, she was a master student at the College of Science, Donghua University. From September 2018 to September 2019, she is a visiting Ph.D. student in University of Alberta, Edmonton, AB, Canada. Her research interests are industrial process modeling and optimization, time-series analysis, variational Bayesian and deep learning.

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.2019.2951622, IEEE Transactions on Industrial Informatics

Dr. Nabil Magbool Jan received the Ph.D. degree from Indian Institute of Technology Madras, Chennai, India, in 2014. He worked for a year as Scientific staff at RWTH Aachen, Germany. From January 2016 to August 2019, he worked as Postdoctoral Fellow in the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada. Since August 2019, he has been working as Assistant Professor in the Dept. of Chemical Engineering, Indian Institute of Technology Tirupati, Tirupati, India. His research interests include soft sensors, state estimation, convex optimization, machine learning and deep learning for process systems

Dr. Kuangrong Hao (M’17) is currently a Professor at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She obtained her B.S. degree in Mechanical Engineering from Hebei University of Technology, Tianjin, China in 1984, her M.S. degree from Ecole Normale Supérieur de Cachan, Paris, France in 1991, and her Ph.D. degree in Mathematics and Computer Science from Ecole Nationale des Ponts et Chaussées, Paris, France in 1995. She has published more than 100 technical papers, and three research monographs. Her scientific interests include machine vision, image processing, intelligent robots, network intelligence, and brain like intelligence.



Dr. Lei Chen is currently at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She received the B.S. degree in electrical engineering and automation in 2006 and the Ph.D. degree in control theory and control engineering in 2014 from Jiangnan University, Wuxi, Jiangsu, China. From 2011 to 2013, she was a visiting student in University of Alberta, Edmonton, AB, Canada. Her research interests include process control, system identification, soft senor and state estimation.

Biao Huang received the B.Sc. and M.Sc. degrees in automatic control from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in process control from the University of Alberta, Edmonton, AB, Canada, in 1997.

In 1997, he joined the University of Alberta as an Assistant Professor with the Department of Chemical and Materials Engineering. He is currently a Professor, the Natural Sciences and Engineering Research Council Industrial Research Chair in Control of Oil Sands Processes, and the Alberta Innovates Technology Futures Industry Chair in Process Control. His research interests include process control, system identification, control performance assessment, Bayesian methods, and state estimation. He has applied his expertise extensively in industrial practice, particularly in the oil sands industry.

Dr. Huang is a Fellow of the Canadian Academy of Engineering and the Chemical Institute of Canada. He was a recipient of Germany's Alexander von Humboldt Research Fellowship, the Canadian Chemical Engineer Society's Syncrude Canada Innovation and D. G. Fisher Awards, the APEGA Summit Research Excellence Award, the University of Alberta McCalla and Killam Professorship Awards, the Petro-Canada Young Innovator Award, and a Best Paper Award from the Journal of Process Control.