



Xiaochu Tang ^{1,*}, Jiawei Yan ¹ and Yuan Li ²

- ¹ School of Automation, Shenyang Aerospace University, Shenyang 110136, China; yanjiawei@stu.sau.edu.cn
- ² College of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China; li-yuan@mail.tsinghua.edu.cn
- * Correspondence: tangxc@sau.edu.cn

Abstract: Variational auto-encoders (VAE) have been widely used in process modeling due to the ability of deep feature extraction and noise robustness. However, the construction of a supervised VAE model still faces huge challenges. The data generated by the existing supervised VAE models are unstable and uncontrollable due to random resampling in the latent subspace, meaning the performance of prediction is greatly weakened. In this paper, a new multi-layer conditional variational auto-encoder (M-CVAE) is constructed by injecting label information into the latent subspace to control the output data generated towards the direction of the actual value. Furthermore, the label information is also used as the input with process variables in order to strengthen the correlation between input and output. Finally, a neural network layer is embedded in the encoder of the model to achieve online quality prediction. The superiority and effectiveness of the proposed method are demonstrated by two real industrial process cases that are compared with other methods.

Keywords: soft sensor; deep learning; supervised model; variational auto-encoder

1. Introduction

It is important for effective quality control and process monitoring to obtain quality or key variables accurately and timely in industrial processes. However, these key variables are often difficult to measure directly, expensive to acquire, or delayed in analysis, which restricts the development of quality control. To make quality or key variables available online, the data-driven soft sensor has been widely applied to build prediction models between easy-to-measure process variables and hard-to-measure quality or key variables [1–3].

Traditional modeling methods, such as partial least squares (PLS) and principal component regression (PCR), have been widely studied and applied to industrial processes [4,5]. Although these methods can achieve online quality prediction, the prediction accuracy is difficult to meet the accuracy requirement for complex nonlinear processes as the result of nonlinearity between the process variables is not considered in the modeling process. In the past decades, some nonlinear modeling methods have been put forward to improve the performance of data-driven soft sensors. For example, some shallow learning methods such as support vector regression, Gaussian process regression, and neural networks have been introduced for nonlinear soft sensors [6–8]. In these methods, nonlinear information can be definitely explicitly considered. However, these shallow networks seem to have inadequate representation capabilities for complex nonlinear processes.

Recently, due to the superiority of nonlinear feature extraction, deep learning has been introduced for process monitoring and soft sensors [9–12]. Shang et al. proposed a deep belief network (DBN) model for sensors to fully extract nonlinear process characteristics [13]. Liu et al. designed a multilayer DBN to represent the nonlinear relationship between the flame images and the outlet oxygen content [14]. Yan et al. combined a de-noising autoencoder with a neural network to establish a soft sensor model [15]. Yuan et al. extracted the



Citation: Tang, X.; Yan, J.; Li, Y. Supervised Multi-Layer Conditional Variational Auto-Encoder for Process Modeling and Soft Sensor. *Sensors* 2023, 23, 9175. https://doi.org/ 10.3390/s23229175

Academic Editor: Roberto Teti

Received: 7 October 2023 Revised: 6 November 2023 Accepted: 10 November 2023 Published: 14 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). quality-relevant nonlinear feature by constructing a variable-wise weighted stacked autoencoder [16,17]. Yao and Ge developed a deep soft sensor model based on a hierarchical extreme learning machine [18]. Wang et al. integrated a long short-term memory with a stacked auto-encoder to achieve quality prediction in a batch process [19]. Feng et al. constructed a dual attention-based encoder-decoder based on the long short-term memory network [20]. In these methods, nonlinear features can be well described and represented. However, these methods lack the description abilities of data measurement noise and process uncertainty. To date, the variational auto-encoder (VAE) as a deep generative model has attracted increasing attention and has also been successfully applied to process modeling [21].

Jiang et al. proposed a variational deep embedding method to improve the data generative process of the VAE method [22]. Zhao et al. proposed a generative method by embedding truncated GMM in VAE to capture the multi-modal representation of data including outlier samples [23]. Dilokthanakul et al. used GMM as a prior distribution to substitute for traditional prior distribution in VAE [24]. Liu et al. obtained a new approximate posterior distribution of VAE to match the true posterior distribution [25]. However, VAE is an unsupervised deep generative model so it cannot be directly applied to soft sensor modeling. While existing process modeling methods based on VAE mainly focus on unsupervised modeling for fault detection and diagnosis, only a few supervised VAE models are developed for soft sensors in industrial processes. Shen et al. proposed a supervised nonlinear probabilistic latent variable regression model based on VAE for soft sensors [26,27]. In this method, nonlinear and dynamic features can be extracted simultaneously. However, VAE only is used to extract nonlinear characteristics between input variables, and the construction of the regression model is actually based on the additional neural networks. Guo et al. developed a Gaussian mixture variational auto-encoder to deal with the problem of multimode soft sensors [28,29]. Similarly, the regression model is built based on a mixture of the probabilistic principal component regression (MPPCR) model, rather than a supervised VAE framework. Xie et al. extended the unsupervised VAE to the supervised VAE by combining the encoder of unsupervised VAE with the decoder of supervised VAE [30]. It may construct a supervised VAE framework for soft sensors in a real sense. However, this method depends on the assumption that the distribution of two different latent variables' spaces are approximately the same. Whether the conditions are strictly met may affect the accuracy of the model.

It is a more critical problem, no matter what unsupervised VAE or supervised VAE, that the generated output is random and uncontrollable only according to the distribution. Therefore, it is critical for a variational auto-encoder to constrain and control data generation. Based on such an idea, a multi-layer conditional variational auto-encoder (M-CVAE) is proposed in order to improve stability and controllability without additional output networks. First, a CVAE model is constructed by inputting the label information to the input of the decoder with the latent variables [31,32]. In this way, the output data can be generated towards the target direction according to the label information instead of randomly. Furthermore, the key label information is introduced to the input of the CVAE model in order to strengthen the correlation between the input and output. Finally, a multi-label layers network is added to the CVAE framework to achieve online prediction.

The rest of this article is structured as follows. The traditional VAE model is introduced in Section 2. The proposed M-CVAE and two industrial cases that are given to demonstrate the effectiveness of the proposed method are illustrated in Section 3. Finally, the conclusions are made.

2. Variational Auto-Encoder

A variational auto-encoder (VAE) is an unsupervised deep generative model. In VAE, Bayesian probability is introduced to the neural network so that it can learn complex data distribution from a probabilistic latent variable space. The model structure of VAE is shown in Figure 1. The encoder and decoder are included in VAE. In the encoder, input data x are mapped to the latent space so that the latent variables **h** are obtained. The latent variables **h** are input to the decoder to obtain the output \hat{x} .



Figure 1. The model structure of VAE.

For VAE, instead of the joint distribution of input variables, the marginal likelihood can be calculated as:

$$p_{\theta}(\mathbf{x}) = \int_{\mathbf{h}} p_{\theta}(\mathbf{h}) p_{\theta}(\mathbf{x}|\mathbf{h}) d\mathbf{h}$$
(1)

where $p_{\theta}(\mathbf{h})$ is the prior of the latent variables, which follows the Gaussian distribution with parameter θ . Because \mathbf{h} and θ are unknown, the marginal likelihood is intractable. The latent variable \mathbf{h} is obtained from the input data \mathbf{x} and $p_{\theta}(\mathbf{h})$ can be substituted for $p_{\theta}(\mathbf{h}|\mathbf{x})$. The true posterior $p_{\theta}(\mathbf{h}|\mathbf{x})$ is given by

$$p_{\theta}(\mathbf{h}|\mathbf{x}) = p_{\theta}(\mathbf{h})p_{\theta}(\mathbf{x}|\mathbf{h})/p_{\theta}(\mathbf{x})$$
(2)

here, the method of variational inference is introduced. Here, $q_{\phi}(\mathbf{h}|\mathbf{x})$ follows Gaussian distribution with parameter ϕ , which is used to approximate true posterior distribution. Furthermore, the marginal log-likelihood can be written as:

$$\log p_{\theta}(\mathbf{x}) = \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{h}$$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log p_{\theta}(\mathbf{x},\mathbf{h}) / p_{\theta}(\mathbf{h}|\mathbf{x}) d\mathbf{h}$$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log \left(\frac{p_{\theta}(\mathbf{x},\mathbf{h})}{q_{\phi}(\mathbf{h}|\mathbf{x})} \cdot \frac{q_{\phi}(\mathbf{h}|\mathbf{x})}{p_{\theta}(\mathbf{h}|\mathbf{x})}\right) d\mathbf{h}$$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x},\mathbf{h})}{q_{\phi}(\mathbf{h}|\mathbf{x})} d\mathbf{h} + \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log \frac{q_{\phi}(\mathbf{h}|\mathbf{x})}{p_{\theta}(\mathbf{h}|\mathbf{x})} d\mathbf{h}$$

$$= \text{ELBO} + D_{\text{KL}} \left(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x}) \right)$$
(3)

where the marginal log-likelihood is divided into two parts: evidence lower bound (ELBO) and KL divergence between $q_{\phi}(\mathbf{h}|\mathbf{x})$ and $p_{\theta}(\mathbf{h}|\mathbf{x})$. The ELBO can be further written as:

$$\begin{aligned} \text{ELBO} &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log \left(p_{\theta}(\mathbf{x},\mathbf{h}) / q_{\phi}(\mathbf{h}|\mathbf{x}) \right) d\mathbf{h} \\ &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log p_{\theta}(\mathbf{x},\mathbf{h}) p(\mathbf{h}) / q_{\phi}(\mathbf{h}|\mathbf{x}) d\mathbf{h} \\ &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log p(\mathbf{h}) / q_{\phi}(\mathbf{h}|\mathbf{x}) d\mathbf{h} + \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}) \log p_{\theta}(\mathbf{x}|\mathbf{h}) d\mathbf{h} \\ &= - D_{\text{KL}} \left(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x}) \right) + \mathbb{E}_{q_{\phi}} [\log p_{\theta}(\mathbf{x}|\mathbf{h})] \end{aligned}$$
(4)

where $D_{KL}(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x}))$ is the divergence of $p_{\theta}(\mathbf{h}|\mathbf{x})$ and $q_{\phi}(\mathbf{h}|\mathbf{x})$. $\mathbb{E}_{q_{\phi}}[\log p_{\theta}(\mathbf{x}|\mathbf{h})]$ is the reconstructed error of $\log p_{\theta}(\mathbf{x}|\mathbf{h})$. Maximizing the marginal probability function can

be converted to maximizing the evidence lower bound. Thus, maximizing ELBO can be written as:

$$\max_{\theta,\phi} \text{ELBO} = \max_{\theta,\phi} \left\{ \mathbb{E}_{q_{\phi}}[\log p_{\theta}(\mathbf{x}|\mathbf{h})] - D_{\text{KL}}(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x})) \right\}$$
(5)

Prior $p_{\theta}(\mathbf{x})$ follows Gaussian distribution with zero mean and variance 1, and the approximate posterior $q_{\phi}(\mathbf{h}|\mathbf{x})$ is assumed to follow a multivariate Gaussian with mean $\mu_{\mathbf{x}}$ and variance $\Sigma_{\mathbf{x}}$. The KL divergence is:

$$D_{\mathrm{KL}}(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x})) = \int_{\mathbf{h}} N(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \log N(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) / N(0, \mathbf{I}) d\mathbf{h}$$

= $-\frac{1}{2} \sum_{t=1}^{T} \left(1 + \log(\Sigma_{\mathbf{x}}^{t})^{2} - (\mu_{\mathbf{x}}^{t})^{2} - (\Sigma_{\mathbf{x}}^{t})^{2} \right)$ (6)

where $\mu_{\mathbf{x}}^{t}$ and $\Sigma_{\mathbf{x}}^{t}$ denote the mean and variance of the sample at t-th time, $t = 1, 2, \dots, T$. The loss function of the VAE can be simplified as:

$$J(\theta, \phi; \mathbf{x}) = -\mathbf{D}_{\mathrm{KL}}(q_{\phi}(\mathbf{h}|\mathbf{x}), p_{\theta}(\mathbf{h}|\mathbf{x})) + \mathbb{E}_{q_{\phi}}[\log p_{\theta}(\mathbf{x}|\mathbf{h})]$$

= $\frac{1}{2}\sum_{t=1}^{T} \left(1 + \log(\sum_{\mathbf{x}}^{t})^{2} - (\mu_{\mathbf{x}}^{t})^{2} - (\sum_{\mathbf{x}}^{t})^{2}\right) + \frac{1}{T}\sum_{t=1}^{T} \log p_{\theta}(\mathbf{x}|\mathbf{h}^{(t)})$ (7)

where *T* is the sampling number and $\mathbf{h}^{(t)}$ denotes the latent variable of the t-th sample.

3. Methodology

In this section, the proposed multi-layer conditional variational auto-encoder (M-CVAE) is presented. Firstly, the structure of the M-CVAE model is described. The derivation of the M-CVAE algorithm is further given. Finally, the procedures of the soft sensor based on the M-CVAE model are presented.

3.1. Supervised Multi-Layer Conditional Variational Auto-Encoder

VAE provides an unsupervised modeling framework. However, it cannot be applied to soft sensor modeling between process variables and quality variables. A supervised M-CVAE structure is constructed in this section, as shown in Figure 2. In M-CVAE, a conditional model structure is constructed based on the basic VAE framework. The label x as a condition is added to the input of the encoder and decoder of VAE. The input of the encoder becomes a concatenation of the original data y and label information x, while the output remains unchanged. The input of the decoder becomes a concatenation of normal distribution sampling corresponding to latent variable h and label information, while the output remains unchanged. In this way, the label of the encoder can constrain the resampling range in specified label areas, rather than the entire normal distribution. The label of the decoder can control the generation of data according to the specified labels or conditions. Compared with basic VAE, in such a condition structure, data generation is still based on the probability distribution of latent variable space with noise. However, the data generation of the CVAE model is no longer entirely random, but becomes targeted by introducing conditions so that the prediction results become controllable rather than stochastic.



Figure 2. The model structure of M-CVAE.

In order to achieve online quality prediction, a multi-layer CVAE model is further constructed by adding a neural network layer to the CVAE model. The effect of the neural network is that it generates the initial predicted **y** that is used as the input of CVAE. Then, the initial prediction **y** with the corresponding label is injected into the CVAE model to make the final prediction closer to the actual value. The neural network layer has multiple hidden layers, which aim to match the input data **y** of CVAE and labels, to ensure the robustness and generalization ability of the CVAE model, and generate the initial prediction **y** as the input of CVAE, so that the final prediction value can be generated after passing through the conditional encoder and conditional decoder with **x** as the condition. Based on the neural network in front of CVAE, on the one hand, the input data **y** of CVAE and labels can be matched, and on the other hand, online prediction can be achieved.

Similar to the VAE model, to train the model, it is necessary to obtain the log-likelihood function, and the log-likelihood of M-CVAE can be written as follows:

$$\log p_{\theta}(\mathbf{y}|\mathbf{x}) = \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) d\mathbf{h} \cdot \log q_{\phi}(\mathbf{y}|\mathbf{x})$$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}) \log p_{\theta}(\mathbf{y} | \mathbf{x}) d\mathbf{h}$$

 $= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log(p_{\theta}(\mathbf{h}, \mathbf{x}, \mathbf{y}) / p_{\theta}(\mathbf{h}|\mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{x})) d\mathbf{h}$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log \left(\frac{p_{\theta}(\mathbf{h}, \mathbf{x}, \mathbf{y})}{q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{x})} \cdot \frac{q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y})}{p_{\theta}(\mathbf{h}|\mathbf{x}, \mathbf{y})} \right) d\mathbf{h}$$

$$= \int_{\mathbf{h}} q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}) \log \left(p_{\theta}(\mathbf{h}, \mathbf{x}, \mathbf{y}) / \left(q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{x}) \right) \right) d\mathbf{h} + \int_{\mathbf{h}} q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}) \log \left(q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}) / p_{\theta}(\mathbf{h} | \mathbf{x}, \mathbf{y}) \right) d\mathbf{h}$$

= ELBO + D_{KL}
$$(q_{\phi}(\mathbf{h}|\mathbf{x},\mathbf{y}), p_{\theta}(\mathbf{h}|\mathbf{x},\mathbf{y}))$$

where it can be estimated by the neural network of S-DCVAE, **x** is a process variable and also serves as a condition for the model, **h** is the latent variable, and $q_{\phi}(\mathbf{y}|\mathbf{x})$ is used to approximate $p_{\theta}(\mathbf{y}|\mathbf{x})$. ELBO was used as the objective function in variational inference. The log-likelihood function can be indirectly maximized by maximizing ELBO. ELBO can be further written as:

$$\begin{aligned} \text{ELBO} &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log(p_{\theta}(\mathbf{h}, \mathbf{x}, \mathbf{y}) / q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \cdot p_{\theta}(\mathbf{x})) d\mathbf{h} \\ &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log\frac{p_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{h}) p_{\theta}(\mathbf{h}|\mathbf{x}) p_{\theta}(\mathbf{x})}{q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) p_{\theta}(\mathbf{x})} d\mathbf{h} \\ &= \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log(p_{\theta}(\mathbf{h}|\mathbf{x}) / q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y})) d\mathbf{h} + \int_{\mathbf{h}} q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}) \log p_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{h}) d\mathbf{h} \\ &= -D_{\mathrm{KL}} (q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y}), p_{\theta}(\mathbf{h}|\mathbf{x})) + \mathbb{E}_{q_{\phi}} [\log p_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{h})] \end{aligned}$$
(9)

(8)

Furthermore, maximizing ELBO can be written as:

$$\max_{\theta,\phi} \text{ELBO} = \max_{\theta,\phi} \left\{ \mathbb{E}_{q_{\phi}}[\log p_{\theta}(\mathbf{y}|\mathbf{x},\mathbf{h})] - D_{\text{KL}}(q_{\phi}(\mathbf{h}|\mathbf{x},\mathbf{y}), p_{\theta}(\mathbf{h}|\mathbf{x})) \right\}$$
(10)

where the approximate posterior $q_{\phi}(\mathbf{h}|\mathbf{x}, \mathbf{y})$ is assumed to be a multivariate Gaussian distribution with mean $\mu_{\mathbf{xy}}$ and variance $\Sigma_{\mathbf{xy}}$. The KL divergence can be given as follows:

$$D_{\mathrm{KL}}(q_{\phi}(\mathbf{h}|\mathbf{x},\mathbf{y}), p_{\theta}(\mathbf{h}|\mathbf{x})) = \int_{\mathbf{h}} N(\mu_{\mathbf{x}\mathbf{y}}, \Sigma_{\mathbf{x}\mathbf{y}}) \log(N(\mu_{\mathbf{x}\mathbf{y}}, \Sigma_{\mathbf{x}\mathbf{y}})/N(0, \mathbf{I})) d\mathbf{h}$$

$$= -\frac{1}{2} \sum_{t=1}^{T} \left(1 + \log\left(\Sigma_{\mathbf{x}\mathbf{y}}^{t}\right)^{2} - \left(\mu_{\mathbf{x}\mathbf{y}}^{t}\right)^{2} - \left(\Sigma_{\mathbf{x}\mathbf{y}}^{t}\right)^{2} \right)$$
(11)

Therefore, the loss function of the supervised M-CVAE can be written as follows:

$$J(\theta, \phi; \mathbf{x}, \mathbf{y}) = -\mathbf{D}_{\mathrm{KL}} \left(q_{\phi}(\mathbf{h} | \mathbf{x}, \mathbf{y}), p_{\theta}(\mathbf{h} | \mathbf{x}) \right) + \mathbb{E}_{q_{\phi}} [\log p_{\theta}(\mathbf{y} | \mathbf{x}, \mathbf{h})]$$

$$= \frac{1}{2} \sum_{t=1}^{T} \left(1 + \log \left(\sum_{\mathbf{xy}}^{t} \right)^{2} - \left(\mu_{\mathbf{xy}}^{t} \right)^{2} - \left(\sum_{\mathbf{xy}}^{t} \right)^{2} \right) + \frac{1}{T} \sum_{t=1}^{T} \log p_{\theta} \left(\mathbf{y} | \mathbf{x}, \mathbf{h}^{(t)} \right)$$
(12)

where *T* is the number of samples and $\mathbf{h}^{(t)} = \boldsymbol{\mu}_{\mathbf{xy}}^{(t)} + \boldsymbol{\Sigma}_{\mathbf{xy}}^{(t)} \times \varepsilon, \varepsilon \sim N(0, \mathbf{I})$ is obtained by reparameterization.

3.2. Soft Sensor Based on M-CVAE

In this section, a detailed derivation of the M-CVAE model has been illustrated and M-CVAE is applied for an online soft sensor. A series of online samples, denoted as \mathbf{x}_{new}^t , have been obtained. The predicted variable \mathbf{y}_{new}^t can be given as follows:

$$\begin{aligned} \mathbf{y}_{new}^{t} &= f\left(\mathbf{x}_{new}^{t}\right) \\ \mathbf{h}_{new}^{t} &= \mu_{\mathbf{x}_{\mathbf{y}_{new}}}^{t} + \Sigma_{\mathbf{x}_{\mathbf{y}_{new}}}^{t} \times \varepsilon, \ \varepsilon \sim N(0, I) \\ \mathbf{\hat{y}}_{new}^{t} &= g\left(\mathbf{h}_{new}^{t}\right) \end{aligned} \tag{13}$$

where \mathbf{y}_{new}^t is the quality value obtained from the neural network, and f(*) and g(*) are the nonlinear functions corresponding to the neural network layer and CVAE, respectively. \mathbf{h}_{new}^t , μ_{xynew}^t , Σ_{xynew}^t are the latent variable, mean, and variance of the online sample \mathbf{x}_{new}^t . The main procedures for the M-CVAE are summarized as follows:

- 1. Collect input data and output data for the training set.
- 2. Determine the process variables for and standardize the training set.
- 3. Initialize M-CVAE network parameters θ , ϕ .
- 4. Train the M-CVAE model for output prediction.
- Choose a different number of the latent variables for the M-CVAE model. Repeat step 4 to determine the optimal number of latent variables.
- 6. Collect the test data and standardize the test data set.
- 7. Predict the quality variable $\hat{\mathbf{y}}_{new}^t$.

The flowchart of M-CVAE is provided in Figure 3.

3.3. Case Studies

In this section, the M-CVAE model is applied to two real industrial cases for soft sensors. To evaluate the performance of the proposed M-CVAE, three indexes, the mean absolute error (MAE), the root mean square error (RMSE), and the coefficient of determination index R², are calculated comparing the results with other regression models.



Figure 3. The flowchart of M-CVAE.

3.3.1. Debutanizer Column

The debutanizer column is an important part of the refinery process in petroleum production processes, which can separate propane and butane from the naphtha stream [33]. The process flowchart is shown in Figure 4. Due to the butane content at the bottom of the debutanizer column being very low, the measurement of butane concentration is difficult and there is usually a great delay. Therefore, it is valuable to introduce the soft senor for butane concentration.



Figure 4. The flowchart of the debutanizer column.

In this paper, a total of 2394 samples have been collected from the debutanizer column. The first 1900 samples are used for the training set and the last 400 samples are used for the test set. Seven process variables are selected as the input of the M-CVAE model to predict the butane concentration and a detailed description of the seven process variables and the output of the predicted variable is shown in Table 1. The trend of input variables and output variables is shown in Figure 5. It can be seen that the input of these variables has obvious fluctuations, which indicates that there is significant nonlinearity in such a process.

In order to demonstrate the nonlinearity between the input variables and output variables, the degree of repeatability (DR) and the differential degree of repeatability (DDR) are introduced [34]. The DR and DDR can reflect the similarity and difference of the correlation of the sample blocks.

Table 1. Description of process variables.



Figure 5. The trend distributions of input variables and output variables.

Firstly, the training set is divided into 100 blocks with 19 samples in each block. The DR of each block and the DDR of two adjacent blocks are calculated as shown in Figure 6. In Figure 6, the values of the DR and DDR have random change among the sample blocks, which indicates that the correlation between input and output is not consistent throughout the whole process. It is illustrated that the nonlinearity between the process variables and output variable is obvious.



Figure 6. The trend of correlation between input variables and output variable of the prosses.

Based on the training data set, the M-CVAE model is built. For comparison, the other soft sensor models including PLS, SVR, DVAE, and Supervised NDS models are also built. Here, DVAE is supervised VAE proposed in the paper [30]. Supervised NDS is a supervised nonlinear dynamic model composed of VAE, a neural network and dynamic system, which is proposed in the paper [27].

The encoder of M-CVAE consists of three convolutional layers and a fully connected layer, and the decoder consists of three deconvolution layers and a fully connected layer. The number of convolution kernels is 32, the size of the convolution kernel is 3×3 , and the activation function is the Relu function. The number of latent variables is a key parameter, which affects the performance of the model to a certain extent. Therefore, the average evaluation indices of MAE, RMSE, and R² for the M-CVAE model, which are based on the seven test experiments, are shown in Figure 7. It can show the fluctuation degree of MAE, RMSE, and R² with different numbers of latent variables. That is to say, the smaller the rectangular area, the more stable the performance of the model.



Figure 7. The evaluation indices versus different numbers of latent variables.

Therefore, it can be seen that the M-CVAE model has the most stable performance when the number of latent variables is 7. While the number of latent variables is 6, more optimal values of MAE, RMSE, and R² can be obtained. Also, the stable performance is quite excellent. Furthermore, the optimal values of MAE, RMSE, and R² corresponding to different numbers of latent variables are given in Table 2. From Table 2, it can be seen that the optimal values of MAE, RMSE, and R² are obtained when the number of latent variables is 6. Finally, the number of latent variables is determined as 6 for the M-CVAE model from a comprehensive point. Meanwhile, the optimal number of latent variables is also selected for other regression models.

Ν	MAE	RMSE	R ²
1	0.1223	0.1476	0.3805
2	0.1166	0.1433	0.4162
3	0.1191	0.1445	0.4068
4	0.1195	0.1482	0.3761
5	0.1152	0.1424	0.4238
6	0.1099	0.1347	0.4844
7	0.1159	0.1413	0.4329

Table 2. Evaluation indicators versus different numbers of latent variables.

The prediction results of PLS, SVR, DVAE, Supervised NDS, and the proposed M-CVAE method are shown in Figure 8. From Figure 8, it is not difficult to see that the predicted butane concentration based on the M-CVAE method is closest to the actual value in most of the process. It also can be seen that neither PLS nor SVR can fit the actual values well. The reason is that traditional machine learning methods are based on shallow learning, which cannot deeply mine complex nonlinearity in data. The M-CVAE has a deep generative model composed of multi-layer networks, which can well mine the nonlinear

characteristics of data. With DVAE and Supervised NDS as deep learning methods, there is a better fluctuation trend of tracking the actual value. However, the fitting degree of these two methods has declined with obvious fluctuation. This is because the data generation based on both the DVAE and Supervised NDS model is similar to the traditional VAE model. While data generation is based on VAE, no constraints are introduced to the model, and as a result, the resampling can be carried out in the whole latent space. This can cause instability and uncontrollability issues in data generation. It is because as the encoding area expands, the resampling range also expands due to the noise introduced in the VAE framework. This may lead to strengthen the randomness and uncertainty of resampling, increase the sampling probability far from the original data code area, and decrease the probability of the original code area. It means that the constraint of the original data to resampling is reduced, which to some extent causes the uncontrollability and randomness of data generation. As a result, the generated data (estimated original data) output from the decoder are not as close as possible to the original data. Compared with DVAE and Supervised NDS, the proposed M-CVAE method has better traceability even at obvious fluctuation, such as samples 80th-130th, 180th-230th, and 255th-350th. The reason why M-CVAE can display the superior performance of prediction is that both process variables and butane concentration can be considered in the M-CVAE model instead of just considering process variables in DVAE and Supervised NDS. In this way, nonlinearity between process variables and butane concentration can be well extracted so that it is quite helpful for the regression model to improve the performance of prediction. It is more important that M-CVAE can control the predicted value close to the actual value by a condition instead of outputting the predicted value only according to a probability distribution, which can improve the controllability and stability of the model. In addition, labels or conditions input to the encoder and decoder are changeable with the input data. That is to say, the input data are always input to the encoder and decoder together with the matching label. Therefore, the dynamic label always can specify generated data as close as possible to the current input sample. This ensures the generalization ability and robustness of the model. Furthermore, the error prediction of all the methods is shown in Figure 9 for comparison. From Figure 9, it can be seen that the proposed M-CVAE method has the smallest error in most of the process. The detailed MAE, RMSE, and R^2 are given in Table 3. From Table 3, we can see that M-CVAE has the lowest MAE and RMSE and the highest R^2 among all the methods. Therefore, it can be demonstrated that the soft sensor based on M-CVAE has optimal performance.

N	SVR	PLS	DVAE	Supervised NDS	M-CVAE
MAE	0.1565	0.1524	0.1257	0.1257	0.1146
RMSE R ²	0.1815 0.0636	0.1790 0.0893	0.1477 0.3798	0.1477 0.3798	0.1379 0.4324
	0.0000	0.0070	0.07.20	0.0770	0.1021

Table 3. Evaluation indices for SVR, PLS, DVAE, Supervised NDS and M-CVAE.

3.3.2. CO₂ Absorption Column

The Ammonia synthesis process is a common industrial process for producing NH_3 used as the basic material for Urea synthesis. In this process, NH_3 is produced as well as CO_2 so CO_2 should be further separated. Therefore, the CO_2 absorption column is an important unit in Ammonia synthesis for CO_2 separation, and CO_2 content is a key variable for quality control. The flowchart of the CO_2 absorption column is given in Figure 10.



Figure 8. The prediction results of PLS, SVR, DVAE, Supervised NDS, and M-CVAE.



Figure 9. The prediction errors of PLS, SVR, DVAE, Supervised NDS, and M-CVAE.

In this paper, 11 process variables are selected for CO₂ content prediction, which are listed in Table 4. A total of 30,000 samples are collected, in which the first 2000 samples are used as the training data set and the last 500 samples are used as the test data set. The trend of input variables and output variables is shown in Figure 11. It can be seen that the input variables and output variable have obvious fluctuation. It can be inferred that this process has a strong nonlinearity. To illustrate the nonlinearity between the input and output, the training set is divided into 100 blocks with 40 samples in each block. The DR of each block

and the DDR of two adjacent blocks are shown in Figure 12. In Figure 12, the values of DR and DDR have a significant fluctuation, which illustrates that the nonlinearity between the process variables and output variable is obvious.



Figure 10. Flowchart of the CO₂ absorption column.

Table 4. Description of the variables in CO₂ absorption column.

Process Variables	Descriptions		
variable 1	Process gas pressure entering 05E001		
variable 2	05F003 liquid level		
variable 3	05E003 outlet lean liquid temperature		
variable 4	Lean fluid flow to 05C001		
variable 5	Semi-lean fluid flow to 05C001		
variable 6	05F003 outlet process temperature degree		
variable 7	05C001 process gas inlet and outlet pressure difference		
variable 8	05C001 outlet-rich liquid temperature		
variable 9	05C001 liquid level		
variable 10	06F001 high liquid level alarm value		
variable 11	Enter the process gas pressure of unit 06		
У	Residual CO ₂ content in process gas		

Based on the training data set, the model is built. The soft sensor model based on the proposed M-CVAE, PLS, SVR, DVAE, and Supervised NDS model is built. The encoder of M-CVAE consists of five convolutional layers and a fully connected layer, and the decoder consists of five deconvolution layers and a fully connected layer. The number of convolution kernels is 32, the size of the convolution kernel is 3×3 , and the activation function is the Relu function. Similarly, the number of latent variables for the M-CVAE is determined based on the average evaluation indices of MAE, RMSE, and R². The average evaluation index based on 11 test experiments is listed in Table 5. From Table 5, it can be seen that the optimal values of MAE, RMSE, and R² are obtained when the number of latent variables is 11.





Figure 11. The trend distributions of input variables and output variables.



Figure 12. The trend of correlation between input variables and output variable.

Ν	MAE	RMSE	R ²
1	0.0024	0.0028	0.8410
2	0.0023	0.0028	0.8338
3	0.0022	0.0027	0.8475
4	0.0022	0.0027	0.8475
5	0.0023	0.0026	0.8497
6	0.0022	0.0026	0.8537
7	0.0022	0.0026	0.8548
8	0.0022	0.0026	0.8555
9	0.0022	0.0026	0.8530
10	0.0022	0.0026	0.8559
11	0.0021	0.0026	0.8575

Table 5. Evaluation indices of different latent variables.

The prediction results of PLS, SVR, DVAE, Supervised NDS, and M-CVAE are shown in Figure 13. From Figure 13, it can be seen that the prediction results based on PLS and SVR show a poor tracking performance. This is because the shallow learning method cannot extract the underlying complex nonlinearity of the data set enough. While the prediction results based on the DVAE and Supervised NDS have a significant improvement in tracking ability in the first 300 samples, it is due to the deep extraction on nonlinearity. However, it can be found that DVAE and Supervised NDS also have a weak prediction performance after about the 300th sample. Instead, the proposed M-CVAE model has an

outstanding tracking capacity throughout the process, especially in the last 200 samples with a big change. The reason is that the DCVAE and Supervised NDS model cannot solve the instability and uncontrollability problems in VAE model data generation. This is because, with the expansion of the coding area, the randomness and uncertainty of the sampling will also increase, increasing the uncontrollability and randomness of the generated data. As a result, the similarity between the generated data output by the decoder and the original data is reduced. On the contrary, the M-CVAE model constrains the sampling range of the specified label area by adding the label of the original data as a condition to the model. In this way, the generated data can be closer to the original data, and the generalization ability and robustness of the model can be ensured. Because the labels' input to the encoder and decoder will change with the input data, the input data is always input to the encoder and decoder together with the matched labels. Therefore, the label always specifies that the generated data are as close as possible to the current input sample. In this way, the M-CVAE shows superior fitting performance, even in a big fluctuation. The prediction error is further shown in Figure 14. The detailed information of MAE, RMSE, and \mathbb{R}^2 are given in Table 6. From Figure 14 and Table 6, it can be seen that the performance of prediction of the proposed M-CVAE model is superior to other methods.



Figure 13. The prediction results of each model.

Table 6. Evaluation indices for each model.

N	SVR	PLS	DVAE	Supervised NDS	M-CVAE
MAE	0.0057	0.0045	0.0036	0.0036	0.0032
RMSE	0.0062	0.0055	0.0043	0.0042	0.0041
R ²	0.1644	0.3429	0.6012	0.6237	0.6419



Figure 14. The prediction errors of each model.

4. Conclusions

In this paper, a new M-CVAE model has been proposed for soft sensor application in industrial processes. The proposed M-CVAE model can not only extract nonlinear characteristics effectively, but can also make the predicted output have the superior ability to track the actual value. Through a comparative study of two industrial cases, it was found that when the process variables change greatly, the proposed M-CVAE model can still obtain better prediction performance by adding the label of the original data as a condition to the model to constrain the resampling range of the specific label area.

For further research on soft sensors, the simultaneous extraction of the multiple characteristics underlying the data should be further considered. In this paper, nonlinearity is deeply mined to describe process characteristics. However, in real industrial processes, other data features, such as the dynamic non-Gaussian feature, are all contained in data. If the dynamic feature and non-Gaussian distribution of the latent variables can be further considered, it will be more helpful to improve the accuracy and robustness of the soft sensor model. Another important issue is that the proposed M-CVAE model can be extended to a semi-supervised model to deal with the lack of labeled samples.

Author Contributions: Conceptualization, X.T. and J.Y.; methodology, X.T. and J.Y.; software, J.Y.; validation, J.Y.; formal analysis, X.T. and J.Y.; investigation, X.T. and Y.L.; resources, X.T. and Y.L.; data curation, X.T. and Y.L.; writing—original draft preparation, J.Y.; writing—review and editing, X.T. and J.Y.; visualization, J.Y.; supervision, X.T. and Y.L.; project administration, X.T. and J.Y.; funding acquisition, X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is unavailable due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [CrossRef]
- Wang, D.; Liu, J.; Srinivasan, R. Data-driven soft sensor approach for quality prediction in a refining process. *IEEE Trans. Ind. Inform.* 2010, 6, 11–17. [CrossRef]

- 3. Zhang, X.; Ge, Z. Automatic Deep extraction of robust dynamic features for industrial big data modeling and soft sensor application. *IEEE Trans. Ind. Inform.* 2020, *16*, 4456–4467. [CrossRef]
- Dong, Y.; Qin, S.J. Regression on dynamic PLS structures for supervised learning of dynamic data. J. Process Control 2018, 68, 64–72. [CrossRef]
- 5. Ge, Z.; Gao, F.; Song, Z. Mixture probabilistic PCR model for soft sensing of multimode processes. *Chemom. Intell. Lab. Syst.* 2011, 105, 91–105. [CrossRef]
- 6. Chelgani, S.C.; Shahbazi, B.; Hadavandi, E. Support vector regression modeling of coal flotation based on variable importance measurements by mutual information method. *Measurement* **2018**, *114*, 102–108. [CrossRef]
- Daemi, A.; Alipouri, Y.; Huang, B. Identification of robust gaussian process regression with noisy input using EM algorithm. *Chemom. Intell. Lab. Syst.* 2019, 191, 1–11. [CrossRef]
- 8. Shao, W.; Ge, Z.; Song, Z.; Wang, K. Nonlinear industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines. *Control. Eng. Pract.* **2019**, *91*, 104098. [CrossRef]
- 9. Sun, Q.; Ge, Z. A Survey on deep learning for data-driven soft sensors. IEEE Trans. Ind. Inform. 2021, 17, 5853–5866. [CrossRef]
- 10. Yu, J.; Yan, X. Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network. *IEEE Trans. Cybern.* **2020**, *50*, 3998–4007. [CrossRef]
- 11. Guo, F.; Bai, W.; Huang, B. Output-relevant variational autoencoder for Just-in-time soft sensor modeling with missing data. J. Process Control 2020, 92, 90–97. [CrossRef]
- 12. Zhang, X.; Zhang, H.; Song, Z. Feature-aligned stacked autoencoder: A novel semi-supervised deep learning model for pattern classification of industrial faults. *IEEE Trans. Artif. Intell.* **2021**, *4*, 592–601. [CrossRef]
- Shang, C.; Yang, F.; Huang, D.; Lyu, W. Data-driven soft sensor development based on deep learning technique. J. Process Control 2014, 24, 223–233. [CrossRef]
- 14. Liu, Y.; Fan, Y.; Chen, J. Flame images for oxygen content prediction of combustion systems using DBN. *Energy Fuels* **2017**, *31*, 8776–8783. [CrossRef]
- 15. Yan, W.; Tang, D.; Lin, Y. A data-driven soft sensor modeling method based on deep learning and its application. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4237–4245. [CrossRef]
- 16. Yuan, X.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3235–3243. [CrossRef]
- Yuan, X.; Gu, Y.; Wang, Y.; Yang, C.; Gui, W. A deep supervised learning framework for data-driven soft sensor modeling of industrial processes. *IEEE Trans. Neural Netw. Learn. Syst.* 2020, 31, 4737–4746. [CrossRef]
- 18. Yao, L.; Ge, Z. Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application. *IEEE Trans. Ind. Electron.* **2018**, *65*, 1490–1498. [CrossRef]
- 19. Wang, K.; Gopaluni, R.B.; Chen, J.; Song, Z. Deep learning of complex batch process data and its application on quality prediction. *IEEE Trans. Ind. Inform.* 2020, *16*, 7233–7242. [CrossRef]
- Feng, L.; Zhao, C.; Sun, Y. Dual attention-based encoder–Decoder: A customized sequence-to-sequence learning for soft sensor development. *IEEE Trans. Neural Netw. Learn. Syst.* 2021, 32, 3306–3317. [CrossRef]
- 21. Doersch, C. Tutorial on variational autoencoders. arXiv 2016, arXiv:1606.05908.
- 22. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv* 2017, arXiv:1611.05148v3.
- Zhao, Q.; Honnorat, N.; Adeli, E.; Pfefferbaum, A.; Sullivan, E.V.; Pohl, K.M. Variational autoencoder with truncated mixture of gaussians for functional connectivity analysis. In Proceedings of the International Conference on Information Processing in Medical Imaging, Hong Kong, China, 2–7 June 2019; pp. 867–879.
- 24. Dilokthanakul, N.; Mediano, P.; Garnelo, M.; Lee, M.; Salimbeni, H.; Arulkumaran, K.; Shanahan, M. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv* **2016**, arXiv:1611.02648.
- 25. Liu, G.; Liu, Y.; Guo, M.; Li, P.; Li, M. Variational inference with gaussian mixture model and householder flow. *Neural Netw.* **2019**, 109, 43–55. [CrossRef]
- Shen, B.; Yao, L.; Ge, Z. Nonlinear probabilistic latent variable regression models for soft sensor application: From shallow to deep structure. *Control Eng. Pract.* 2020, 94, 104198. [CrossRef]
- Shen, B.; Ge, Z. Supervised nonlinear dynamic system for soft sensor application aided by variational auto-encoder. *IEEE Trans. Instrum. Meas.* 2020, 69, 6132–6142. [CrossRef]
- 28. Guo, F.; Xie, R.; Huang, B. A deep learning just-in-time modeling approach for soft sensor based on variational autoencoder. *Chemom. Intell. Lab. Syst.* 2020, 197, 103922. [CrossRef]
- 29. Guo, F.; Wei, B.; Huang, B. A just-in-time modeling approach for multimode soft sensor based on gaussian mixture variational autoencoder. *Comput. Chem. Eng.* 2021, 146, 107230. [CrossRef]
- Xie, R.; Jan, N.M.; Hao, K.; Chen, L.; Huang, B. Supervised variational autoencoders for soft sensor modeling with missing data. IEEE Trans. Ind. Inform. 2020, 16, 2820–2828. [CrossRef]
- Pandey, G.; Dukkipati, A. Variational methods for conditional multimodal deep learning. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 308–315.

- Sohn, K.; Yan, X.; Lee, H. Learning structured output representation using deep conditional generative models. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 3483–3491.
- 33. Fortuna, L.; Graziani, S.; Xibilia, M. Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Eng. Pract.* **2005**, *13*, 499–508. [CrossRef]
- 34. Tang, X.; Li, Y.; Xie, Z. Phase division and process monitoring for multiphase batch processes with transitions. *Chemom. Intell. Lab. Syst.* **2015**, 145, 72–83. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.