Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Developing Variational Autoencoders with Differential Entropy Soft Sensor Models for Nonlinear Processes

Dave Tanny*, Junghui Chen*, Kai Wang**

**Department of Chemical Engineering, Chung-Yuan Christian University, Chung-Li District, Taoyuan, 32023, Taiwan, R.O.C.* (*e-mail: jason@wavenet.cycu.edu.tw*)
**School of Automation, Central South University, Changsha 410083, China

Abstract: Developing a good soft sensor for prediction has been a major interest, given the time lag to obtain quality data. Deep learning based variational autoencoders (VAE) have been implemented in industrial plants because of their capacities in dealing with the complex stochastic nonlinearity with better probabilistic interpretation. However, unsupervised VAE is inapplicable to the prediction. This article proposes a nonlinear soft sensor, which is an extension of the VAE framework with differential entropy (VAE_DE) loss function to construct a prediction model. The proposed VAE_DE model structure allows all the available data to be used for training although the data consist of process-quality data pairs and/or solely process data. Also, VAE_DE enhances the prediction performance and its robustness through capturing the inter-correlations between process data and quality data in the nonlinear probabilistic model. Under the proposed framework, VAE_DE model can be used for quick quality estimates of process data with unavailable quality data. The prediction quality of the proposed method is testified through a numerical case and an industrial case.

Keywords: Differential entropy, Soft sensors, Variational Autoencoder (VAE)

1. INTRODUCTION

Soft sensors have been widely applied to chemical plants as a mean of estimation of the product quality through utilization of process variables. Traditional means of soft sensors are expressed by first principal models, which are derived from physical principles, such as mass and energy balances of the system. However, constructing first principal models require full understanding of the system. In such a situation, it is difficult to consider the complexity of chemical processes.

Meanwhile, data-driven models have been rising in popularity with the abundance of available process data. Commonly-used methods include principal component regression (PCR) (Lin et al., 2007), partial least squares (PLS), independent component analysis (Kaneko et al., 2009), artificial neural networks (ANN) (Shang et al., 2014; Yao et al., 2017), and support vector machines (SVM) (Yan et al., 2004). Some conventional methods, such as PCR and PLS, are limited to linear assumption, so they do not perform well in nonlinearity exhibited in most chemical processes.

Nonlinear models, such as SVM, and kernel methods, allow a more accurate representation of the chemical process as they directly get the global optimum of the variables through eigenvector decomposition. SVM and kernel methods generally expand the original data to a higher dimension to allow linear representation of the data based on the chosen kernel function. However, these methods, especially SVM and the kernel method, are computationally expensive. Their computational load increases exponentially as there are more samples easily collected in most chemical processes. The kernel based model may also lead to singularity during taking the eigenvalue and the eigenvector of the process variables, especially when a lot of process variables inputted into the model may not exhibit any relation to the quality variables. In addition, the performances of the kernel based models also depend on the selected kernel function to map data to a higher dimension. Also, selecting the dimension of the projection is quite tricky.

ANN is preferable as users can flexibly determine the structure layer and the activation function of each layer. It also allows the parameters to be automatically updated through the backpropagation technique. But the parameters are not directly inferentiable to describe the relation between the process and quality variables. The network performance is determined by its loss function. The network structure allows neural networks to be flexible, so the neural networks can model nonlinearities without being bounded by a specific kernel function, such as SVM and kernel methods, and the flexibility in choosing a loss function allows users to choose the way the network is trained. The next problem is how to increase the robustness of the prediction from the neural network output to correspond to the stochastic nature of chemical processes.

Variational autoencoder (VAE) models based on probabilistic deep learning have been rising in popularity because of their high capacity of dealing with complex nonlinearity. It also exhibits the stochastic nature and proves to have better probabilistic interpretation than shallow models with deterministic analysis. Albeit the promising VAE structure can provide a deep orthogonal latent variable model for

complex nonlinear processes, VAE is an unsupervised structure, which cannot be directly and easily applied to traditional soft sensor modeling problems.

In this paper, a novel differential entropy soft sensor model is proposed. It is the extension of the vanilla VAE, named variational autoencoders with differential entropy (VAE_DE) learning. In the proposed method, the VAE_DE model is trained by all the available data no matter whether the data belongs to labelled data or unlabelled data. The use of the entire dataset would not only provide consistency during training the VAE_DE model but also enhance the prediction performance and robustness by prioritizing the soft sensors to capture the inter-correlation between the process and quality data in the nonlinear probabilistic model. Under the proposed framework, the VAE_DE model can provide quick quality estimates of the unlabelled data. The details will be discussed as follows. First, the problem description is given in Section 2. The proposed VAE_DE formulation details are explained in Section 3. Subsequently, the numerical and industrial case studies for the performance assessment of the model is conducted and compared with the conventional methods in Section 4. The conclusion is made in the end of the paper.

2. Methodology

For a clear grasp of the concept of the proposed method, first the concept of VAE is discussed briefly. Then the supervised VAE prediction model is proposed; the soft sensor model training is fully explored along with both the VAE model and the proposed loss function to allow the full utilization of the whole dataset regardless whether the collected data are labelled or unlabelled.

2.1 Variational Autoencoder (VAE)

Suppose a given labelled data composed of $\mathbf{x}$ process data and $\mathbf{y}$ quality data. The key idea behind VAE is to minimize the Kullback-Leibler divergence of the approximate posterior distribution $q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})$ and the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})$,

$$\begin{aligned}&\min KL\left(q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})\right)=\\&E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})-\log p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})\right]\end{aligned}\tag{1}$$

Through Bayes' theorem, the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})$ can also be represented by

$$p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})=\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})}=\frac{p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})\,p_{\boldsymbol{\theta}}(\mathbf{y}\mid\mathbf{z})\,p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})}\tag{2}$$

Substituting Eq.(2) into Eq.(1), and rearranging the equation, one can get

$$\begin{aligned}E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})\right]&=E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})\right]+E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{y}\mid\mathbf{z})\right]\\&-KL\left(q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{z})\right)+KL\left(q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})\right)\end{aligned}\tag{3}$$

where $E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\square]$ is an expectation operator in regard to approximate posterior distribution $q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})$. However, the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})$ is usually intractable as the real marginal distribution $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$ is intractable. Therefore, instead of maximizing the marginal distribution $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$, the variational lower bound, also called evidence lower bound (ELBO), is usually maximized,

$$\begin{aligned}\max L(\mathbf{X},\mathbf{Y})&=E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z})\right]\\&+E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{y}\mid\mathbf{z})\right]\\&-KL\left(q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{z})\right)\end{aligned}\tag{4}$$

In Eq.(4), ELBO will be naturally maximized when the KL divergence (Eq.(1)) can be minimized to zero, as the KL divergence is always positive. However, it should be noticed that the above structure of the VAE model only takes the process and quality variables as inputs to the model and reconstructs back the latent variables and take them as the predicted values of the same input variables. That is, given process variables, the original VAE model cannot be used for the prediction of quality data. In this paper, the supervised VAE model is proposed with the capability of quality prediction. The following section will explain how to derive a supervised VAE network.

2.2 Variational Autoencoder with differential entropy training based prediction model (VAE_DE)

A soft sensor is built upon the assumption that a prediction value of quality data can be achieved by an accurate model with the input of the process variables. In the original VAE, another model with the process data as the input should be constructed to output the prediction of the quality data. To improve the accuracy of the prediction, the model needs a loss function that trains the model with the consideration of the existing correlation between process data and quality data. The previously defined VAE model takes in labelled data as an input. Alternatively, a new neural network model can be used to take in the correlation between $\mathbf{x}$ process variables (the input) and $\mathbf{y}$ quality variables (the output). Thus, the true and approximate posterior distribution are re-defined by

$$\begin{aligned}&\min KL\left(q_\phi(\mathbf{z},\mathbf{y}\mid\mathbf{x})\,\|\,p_\theta(\mathbf{z},\mathbf{y}\mid\mathbf{x})\right)\\&=E_{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log q_\phi(\mathbf{z},\mathbf{y}\mid\mathbf{x})-\log p_\theta(\mathbf{z},\mathbf{y}\mid\mathbf{x})\right]\end{aligned}\tag{5}$$

where

$$\begin{aligned}p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y}\mid\mathbf{x})&=p_{\boldsymbol{\theta}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,p_{\boldsymbol{\theta}}(\mathbf{y}\mid\mathbf{x})\\q_\phi(\mathbf{z},\mathbf{y}\mid\mathbf{x})&=q_\phi(\mathbf{z}\mid\mathbf{x},\mathbf{y})\,q_\phi(\mathbf{y}\mid\mathbf{x})\end{aligned}\tag{6}$$

Through Bayes' theorem, the true posterior $p_\theta(\mathbf{z},\mathbf{y}\mid\mathbf{x})$ can be represented by

$$p_\theta(\mathbf{z},\mathbf{y}\mid\mathbf{x})=\frac{p_{\boldsymbol{\theta}}(\mathbf{x}\mid\mathbf{z},\mathbf{y})\,p_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}\tag{7}$$

Assume that only **z** latent variables can be reconstructed back to **x** process data and **y** quality data. It is a reasonable assumption as, in **z** latent variables, there are common relations between **x** process variables and **y** quality variables. Thus, $p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z},\mathbf{y})$ can be re-written as

$$p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z},\mathbf{y}) = p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}) \tag{8}$$

Substituting Eq.(8) into Eq.(5):

$$\begin{aligned} E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] = {} & E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})\right] \\ & -KL\left(q_\phi(\mathbf{z},\mathbf{y}\,|\,\mathbf{x})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{z})\right) \\ & +KL\left(q_\phi(\mathbf{z},\mathbf{y}\,|\,\mathbf{x})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y}\,|\,\mathbf{x})\right) \end{aligned} \tag{9}$$

The new ELBO is defined as:

$$L(\mathbf{X}) = E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z})\right] - KL\left(q_\phi(\mathbf{z},\mathbf{y}\,|\,\mathbf{x})\,\|\,p_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{z})\right) \tag{10}$$

Expanding each term on the right-hand side of Eq.(10), a simpler form can be provided,

$$L(\mathbf{X}) = E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[L(\mathbf{X},\mathbf{Y})\right] + H\left(p_{\boldsymbol{\theta}}(\mathbf{y}\,|\,\mathbf{x})\right) \tag{11}$$

where $E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[L(\mathbf{X},\mathbf{Y})\right]$ is expectation of $L(\mathbf{X},\mathbf{Y})$ from Eq.(4) with regard to the prediction model $q_\phi(\mathbf{y}\,|\,\mathbf{x})$, and $H\left(p_{\boldsymbol{\theta}}(\mathbf{y}\,|\,\mathbf{x})\right)$ is differential entropy of the prediction network. The derivations of Eq.(11) are detailed in Appendix. Based on the defined loss function in Eq.(11), $p_{\boldsymbol{\theta}}(\mathbf{y}\,|\,\mathbf{x})$ allows the inference of quality data estimate through process data. However, to compute the term of $E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[L(\mathbf{X},\mathbf{Y})\right]$, the previously defined model in Eq.(4) needs to be used because $L(\mathbf{X},\mathbf{Y})$ requires process and quality data as inputs to the model. However, it is important to note that Eq.(11) merely focuses on the expectation of the prediction model only. It does not consider the correlation between process and quality data so that the robustness of the prediction quality estimate toward the real quality data can be weakened. Labelled data can improve the prediction model to mimic the real quality data distribution while unlabelled data can improve the performance of overall model with additional training data. Therefore, it is necessary to provide a new loss function with the advantages of both labelled and unlabelled data. This will be detailed in the next section.

2.3 VAE_DE Training phase

Given a total of N number of data, consisting of N^l number of labelled data (**x** process data and **y** quality data) and N^u number of unlabelled data (**x** process data). The labelled data is represented by $\left(\mathbf{X}^l,\mathbf{Y}\right)=\left[\left(\mathbf{x}_i,\mathbf{y}_i\right)\right]_{i=1}^{N^l}$, and the unlabelled data is given by $\left(\mathbf{X}^u\right)=\left[\left(\mathbf{x}_i\right)\right]_{i=1}^{N^u}$, where the superscript u indicates unlabelled, l indicates labelled, and subscript i indicates the index from N available samples. Based on the previous discussion, it is best to use all the available data without the need to differentiate the training process. This means that the training data for constructing the VAE_DE model consists of labelled data or unlabelled data (process data only); thus, the overall training function from the perspective of the labelled data in Eq.(4) and the unlabelled data in Eq.(11) can be represented by:

$$\max\left[\sum_{i=1}^{N^l} L\left(\mathbf{X}^l,\mathbf{Y}^l\right)+\sum_{i=1}^{N^u} L\left(\mathbf{X}^u\right)\right] \tag{12}$$

where $N = N^l + N^u$. The problem of the loss function using the labelled data in Eq.(4) is that it cannot train and obtain a prediction model at all. Thus, with the derived relation between the labelled loss function and unlabelled loss function in Eq.(11), the labelled loss function can be modified as:

$$\begin{aligned} E_{q_\phi(\mathbf{y}|\mathbf{x}^l)}\left[L\left(\mathbf{X}^l,\mathbf{Y}^l\right)\right] &= L\left(\mathbf{X}^l\right) - H\left(\ln p_{\boldsymbol{\theta}}\left(\mathbf{y}\,|\,\mathbf{x}^l\right)\right) \\ &= L\left(\mathbf{X}^l\right) + E_{q_\phi(\mathbf{y}|\mathbf{x}^l)}\left(\ln q_\phi\left(\mathbf{y}\,|\,\mathbf{x}^l\right)\right) \end{aligned} \tag{13}$$

Through Eq.(13), the prediction network can be trained by the labelled data while it cannot be trained through Eq.(4). Substituting Eq.(13) into Eq.(12) yields:

$$\begin{aligned} & \max\left[\sum_{i=1}^{N^l} L\left(\mathbf{X}^l,\mathbf{Y}\right)+\sum_{i=1}^{N^u} L\left(\mathbf{X}^u\right)\right] \\ & = \max\left[\sum_{i=1}^{N^l} L\left(\mathbf{X}^l\right)+\sum_{i=1}^{N^u} L\left(\mathbf{X}^u\right)+\sum_{i=1}^{N^l} \ln q_\phi\left(\mathbf{y}_i\,|\,\mathbf{x}_i^l\right)\right] \\ & = \max\left[\sum_{i=1}^{N} L\left(\mathbf{X}\right)+\sum_{i=1}^{N^l} \ln q_\phi\left(\mathbf{y}_i\,|\,\mathbf{x}_i^l\right)\right] \end{aligned} \tag{14}$$

With the loss function stated in Eq.(14), the additional log likelihood term of the prediction distribution expresses the difference between the predictions and the real values of available quality variables, to train the prediction model which can mimic the real quality data distribution as closely as possible.

To train the VAE_DE model based on the overall loss function in Eq.(14), the detailed graphical representation of the flowchart of the proposed VAE_DE model is shown in Fig. 1. First, all the process variables will be fed through the prediction network to generate the predicted quality variables. If the process data belongs to labelled data, the log likelihood between the prediction and the labelled counterpart is calculated (red dotted box in Figure 1); then the entropy of the generated prediction is also calculated (brown box with H). The prediction is assimilated with the process data as inputs to the VAE model encoder to obtain the mean and the covariance of the latent variables while the KL divergence is computed by the dissimilarity of the latent variables distribution to its prior distribution with the Gaussian distribution mean of 0 and the covariance of 1. Then the prior is reconstructed back to its original data space through each decoder respectively. The reparameterization trick is applied on both quality data prediction and the latent variables.

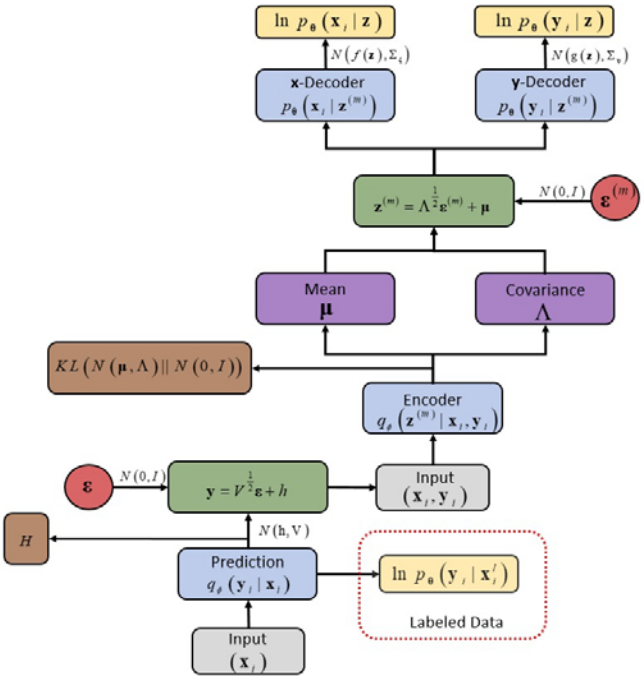


Fig. 1 VAE_DE flowchart diagram

3. Case Studies

In this section, the performance of the proposed VAE_DE method is tested against other conventional methods in a numerical case and an industrial case. The details of the implementation of each method are given below.

3.1 Numerical case

The process data and quality data are generated by two latent variables. Both of them contain noises that affect each variable.

$$\begin{aligned} x_1 &= z_2 \sin(z_1) + w_1 \\ x_2 &= z_1 \cos(z_2) + w_2 \\ y_1 &= \cos(z_1) - \sin(z_2)^2 + \frac{z_1^{1.5}}{z_1 + z_2} + w_3 \end{aligned} \tag{15}$$

where the latent variables (z_1 and z_2) as well as noises (w_1, w_2, and w_3) are distributed in Gaussian distributions with the mean and the variances listed as follows:

$$\begin{aligned} &z_1 \sim N(3, 0.5) \quad z_2 \sim N(2, 0.3) \\ &w_1 \sim N(0, 0.02) \quad w_2 \sim N(0, 0.03) \quad w_3 \sim N(0, 0.05) \end{aligned} \tag{16}$$

Through the function given above, 1,200 labelled samples and 1,200 unlabelled samples are generated for the data infested with noise and the real data without noise. Half of the samples are used as a training dataset and the other as a test dataset. The data are preprocessed by getting the normalized values through each variable's mean and variance before the data are inputted to the model.

The structure of VAE_DE (the encoder, decoder, and prediction networks) is consisted of 3 hidden layers with 30 units in each layer with tanh activation functions. The number of latent variables in the bottleneck section between the encoder and the decoder is set to be 2. In addition, to prevent negative covariance value, the covariance for the final activation function in the latent variable is softplus. The networks are trained for 400, 800, and 1,200 times based on the loss function given in Equation (12). Fig. 2 shows the prediction on the data with and without noise. To assess the prediction performance, the root mean square error (RMSE) of the test dataset is calculated for the prediction values in regard to the data with noise and the data without noise. It is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{17}$$

where y_i and $\hat{y}_i$ denote the i-th quality value and its prediction value, respectively. The quality value y_i also corresponds to the data infested with noise and the real data without noise.

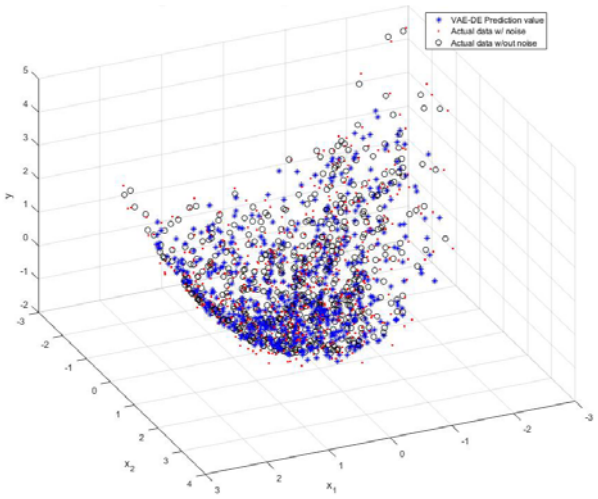


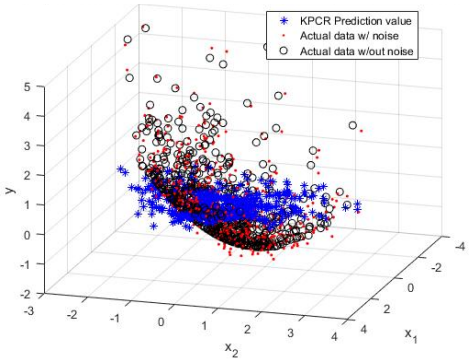
Fig. 2 The prediction of the proposed VAE_DE method compared with the real data with and without noise (1,200 iterations)

Comparing the proposed method with the other conventional methods, PPCR, PPLS, KPCR, and KPLS are used to predict the quality data. For the applications of both KPCR and KPLS, the Gaussian radial basis function is selected as the kernel to handle the nonlinear properties of the generated data. The parameters of each method are properly tuned to generate the best prediction performance. All the methods, except for VAE_DE, uses only labelled data as training data, and the remaining unlabelled data as the testing data. The prediction results of the quality data by KPCR, and KPLS are shown in Fig. 3 (a) and (b) respectively. The prediction result of remaining method will be tested by RMSE criteria.

RMSE predicted by these 5 methods are shown in Table 1. It can be observed that the prediction result of the proposed VAE_DE is becoming better with more training iterations to learn the important features to represent the quality data. Although initially the RMSE of the VAE_DE method with 400 iterations lose to the KPLS method, the currently trained

VAE_DE model still has not fully caught the data representation. When trained further, the VAE_DE model is becoming more accurate and can represent the real data without noise better than the data infested with noise. The VAE_DE result is superior because the neural network allows deep nonlinear transformation, unlike the cases with KPCR and KPLS, both of which only allow shallow nonlinear transformation. With deep nonlinear transformation through many hidden layers, the model is more capable of representing highly complex systems. However, it also needs more training to adjust parameters and catch the nonlinear behavior of systems.

(a)



(b)

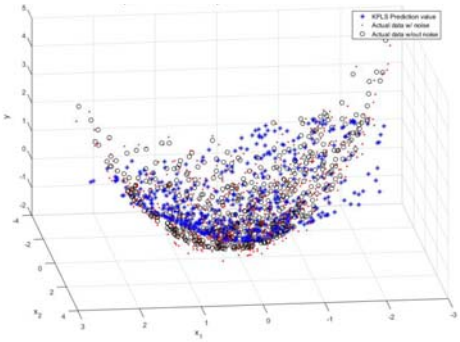


Fig. 3 Prediction results in the numerical case by: (a) KPCR; (b) KPLS

Table 1 The RMSE values of numerical case prediction result

Methods	RMSE (noise infested data)	RMSE (no noise)
PPCR	0.7878	0.7715
PPLS	1.399	1.361
KPCR	0.960	0.982
KPLS	0.557	0.530
VAE_DE(400 iters)	0.680	0.665
VAE_DE(800 iters)	0.471	0.425
VAE_DE(1200 iters)	0.419	0.345

3.2 Ammonia synthesis process

Ammonia is an essential ingredient for a lot of applications, such as the key ingredient in the production of fertilizers. The pre-decarburization process is one of the most important parts in this synthesis process. The carbon dioxide from the process gas is absorbed, and the absorbed CO_2 gas is used for the future production process. The flowchart of this process in Fig. 4 consists of 4 major devices (the feed gas separator, the PG separator, the heat exchanger, and the absorption column). The absorption column is the main device responsible for capturing CO_2 in the feed gas.

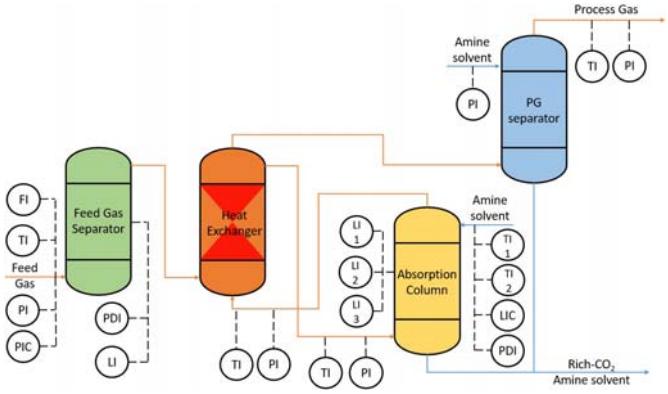


Fig. 4 The decarburization step of ammonia synthesis process

To maximize the production of ammonia, CO_2 in the process gas should be absorbed as much as possible. The goal is to minimize the amount of CO_2 residue in the process gas output, and there are 19 process variables in the process. There are 1,800 data samples; the first half of the samples have quality data, and the remainder does not. The labelled and the unlabelled data are divided in half for training and testing. Those methods, except for VAE_DE, uses only the labelled data as the training set, and the remaining unlabelled data as the testing set. Given available process data, various methods are used to predict the next CO_2 residue in the process gas. The prediction results of PPCR, PPLS, KPCR, KPLS and VAE_DE are 1.0358, 1.554, 2.01, 0.853, 0.536, respectively. The prediction result of KPLS and VAE_DE for all data is shown in Fig. 5, VAE_DE prediction is more accurate than KPLS, due to better representation of non-linear process with its deep non-linear probability model, compared to KPLS shallow non-linear representation.

4. CONCLUSIONS

This paper proposes a nonlinear probabilistic method for training a soft sensor using different forms of data no matter whether they are labelled or unlabelled. This method also provides a consistent training method for all the networks in the proposed VAE-DE model. The proposed method is shown to be superior to other methods because of its complex nonlinear representation of the system in deep neural networks.

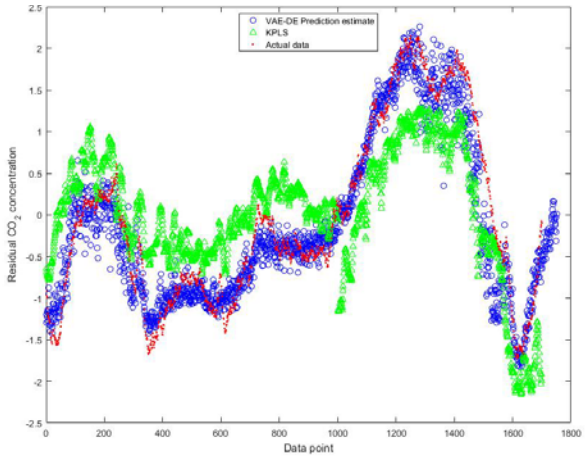


Fig. 5 VAE_DE and KPLS predictions vs. real industrial data

ACKNOWLEDGE

The authors would like to gratefully acknowledge the Ministry of Science and Technology, Taiwan, R.O.C. (MOST 106-2221-E-033-060-MY3) for the financial support.

REFERENCES

Weiwu Yan, Huihe Shao, Xiaofan Wang (2004). Soft sensing modelling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering*, 28, 8, 1489-1498

Bao Lin, Bodil Recke, Jorgen K.H. Knudsen, Sten Bay Jorgensen (2007). A systematic approach for soft sensor development. *Computers and Chemical Engineering*, 31, 419-425

Hiromasa Kaneko, Masamoto Arakawa, Kimito Funatsu (2009). Development of a new soft sensor method using independent component analysis and partial least squares. *AIChE Journal*, 55, 87-98.

Zhiqiang Ge, Zhihuan Song (2010). Semi-supervised Bayesian Method for soft sensor modelling with unlabelled data samples. *AIChE Journal*, 57, 2109-2119.

Chao Shang, Fan Yang, Dexian Huang, Wenxiang Lyu (2014). Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, 24, 223-233.

Diederik P. Kingma, Max Welling (2014). Auto-Encoding Variational Bayes. arXiv:1312.6114v10.

Liang Bao, Xiaofeng Yuan, Zhiqiang Ge (2015). Co-training partial least squares model for semi-supervised soft sensor development. Chemometrics and Intelligent Laboratory Systems, 147, 75-85.

Le Yao, Zhiqiang Ge (2017). Moving window adaptive soft sensor for state shifting process based on weighted supervised latent factor analysis. *Control Engineering Practice*, 61, 2017, 72-80.

Appendix. VAE_DE complete derivation

Expanding each term in Eq.(10):

$$E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \iint q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z})\, d\mathbf{z}d\mathbf{y}$$

$$E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = \underbrace{\iint q_\phi(\mathbf{y}|\mathbf{x}) \underbrace{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \log p_\theta(\mathbf{x}|\mathbf{z})\, d\mathbf{z}}_{E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]} d\mathbf{y}}_{E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right]}$$

$$E_{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] = E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right]$$

Expanding KL divergence term to integral format:

$$KL\left(q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \,\|\, p_\theta(\mathbf{y},\mathbf{z})\right) = \iint q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}{p_\theta(\mathbf{y},\mathbf{z})} d\mathbf{z}d\mathbf{y}$$

$$= \iint q_\phi(\mathbf{y}|\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \log \frac{q_\phi(\mathbf{y}|\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}{p_\theta(\mathbf{y}|\mathbf{z}) p_\theta(\mathbf{z})} d\mathbf{z}d\mathbf{y}$$

$$KL\left(q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \,\|\, p_\theta(\mathbf{y},\mathbf{z})\right) = \iint q_\phi(\mathbf{y}|\mathbf{x}) \underbrace{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}{p_\theta(\mathbf{z})}}_{KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \| p_\theta(\mathbf{z})\right)} d\mathbf{z}\, d\mathbf{y}$$

$$+ \iint q_\phi(\mathbf{y}|\mathbf{x}) \underbrace{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \log q_\phi(\mathbf{y}|\mathbf{x})\, d\mathbf{z}}_{E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log q_\phi(\mathbf{y}|\mathbf{x})\right]} d\mathbf{y}$$

$$- \iint q_\phi(\mathbf{y}|\mathbf{x}) \underbrace{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \log p_\theta(\mathbf{y}|\mathbf{z})\, d\mathbf{z}}_{E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right]} d\mathbf{y}$$

$$KL\left(q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \,\|\, p_\theta(\mathbf{y},\mathbf{z})\right) = \underbrace{\int q_\phi(\mathbf{y}|\mathbf{x}) \underbrace{KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \,\|\, p_\theta(\mathbf{z})\right)}_{E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \| p_\theta(\mathbf{z})\right)\right]} d\mathbf{y}}$$

$$+ \underbrace{\int q_\phi(\mathbf{y}|\mathbf{x}) E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log q_\phi(\mathbf{y}|\mathbf{x})\right] d\mathbf{y}}_{-H\left(p_\theta(\mathbf{y}|\mathbf{x})\right)}$$

$$- \underbrace{\int q_\phi(\mathbf{y}|\mathbf{x}) E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right] d\mathbf{y}}_{E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right]\right]}$$

$$KL\left(q_\phi(\mathbf{z},\mathbf{y}|\mathbf{x}) \,\|\, p_\theta(\mathbf{y},\mathbf{z})\right) = E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \,\|\, p_\theta(\mathbf{z})\right)\right]$$
$$-H\left(p_\theta(\mathbf{y}|\mathbf{x})\right) - E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right]\right]$$

Substitute the derivations back to Eq.(10):

$$L(\mathbf{X}) = E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right] - \begin{pmatrix} E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \,\|\, p_\theta(\mathbf{z})\right)\right] \\ -H\left(p_\theta(\mathbf{y}|\mathbf{x})\right) \\ -E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right]\right] \end{pmatrix}$$

$$L(\mathbf{X}) = E_{q_\phi(\mathbf{y}|\mathbf{x})}\begin{bmatrix} E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \\ +E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{z})\right] \\ \underbrace{-KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \,\|\, p_\theta(\mathbf{z})\right)}_{L(\mathbf{X},\mathbf{Y})} \end{bmatrix} + H\left(p_\theta(\mathbf{y}|\mathbf{x})\right)$$

$$L(\mathbf{X}) = E_{q_\phi(\mathbf{y}|\mathbf{x})}\left[L(\mathbf{X},\mathbf{Y})\right] + H\left(p_\theta(\mathbf{y}|\mathbf{x})\right)$$