Preprints, 10th IFAC International Symposium on
Advanced Control of Chemical Processes
Shenyang, Liaoning, China, July 25-27, 2018

Generic Process Visualization Using Parametric t-SNE

Wenbo Zhu * **Zachary Webb** * **Xianyao Han** ** **Kaitian Mao** ***
Wei Sun ** **José Romagnoli** *

* *Department of Chemical Engineering, Louisiana State University, Baton Rouge, Louisiana 70803, United States (e-mail: jose@lsu.edu)*
** *College of Chemical Engineering, Beijing University of Chemical Technology, Beijing 100029, China*
*** *Shanghai SupeZET Engineering Technology Corp,.Ltd, Shanghai, 200335, China*

Abstract: In this work, a generic process visualization method is introduced using parametric t-SNE and used to visualize real-time process information and correlations among variables on a 2D map. A deep network is used to learn the Kullback-Leibler divergence between the original high-dimensional space and the latent space. In practice, it is observed that a model trained with historical data is not robust enough to visualize shifts into unknown states. Due to the effect of greedy learning, the response of the model is biased toward the most-contributing inputs. To relieve this effect, combinatorial variation creation is applied in the training stage to allow the model to respond to each input more evenly. The proposed method is tested on the Tennessee Eastman Process (TEP) data for four types of faults. The result indicates that the proposed method outperforms conventional methods such as PCA and Isomap, and is able to provide clear visual indication of process changes.

Keywords: Data mining and multivariate statistics; Data-Driven Decision Making

1. INTRODUCTION

Distributed control systems (DCS) are essential for modern chemical industries. Operators use the DCS to monitor the operation in real time and keep the process within safe and productive domains of operation. Traditionally, the monitoring has been based on single-variable analysis with alarms to notify personnel when the measurements of selected variables drift out of safety or process specification limits. The process measurements are usually expressed in two ways: labels with values on the process diagram or a line chart tracking an individual variable on a time axis. These implementations are clear and simple to implement, but they can only provide low-level information. For chemical processes, it is often necessary to analyze correlations between multiple variables in order to determine the state of operation. Another disadvantage of the process diagram is its limited use for a single operating unit. For a chemical plant, having a monitoring system that can be easily applied to multiple units is preferred.

To improve modern data visualization technique, different methods were proposed in recent years. Polygon-based monitoring methods (Wang et al. (2017); Yiakopoulos et al. (2016)) visualize multiple variables in a single polygonal diagram, where each variable is plotted in radial direction of the polygon. Utilizing the knowledge of the operating bounds or statistic tests, the faulty condition can be visualized on any violation outside these bounds. Though such approaches are generic, they are built upon single variable statistics that are unable to extract the hidden correlation among multiple variables. Self-organizing map (SOM) methods (Zhong et al. (2016); Robertson et al. (2015)) are also popular. After training with labeled historical data, the SOM can correctly classify the operation modes. This allows the process behaviors to be visualized via a u-matrix generated from the SOM network. Training these methods requires high computational cost, making frequent adaptations to the model difficult. Without updates to the map, process data from unknown regions of the map can be misclassified, which can lead to problems in real-time monitoring.

In our work, we proposed a generic process visualization method using dimensionality reduction techniques based on deep learning, namely parametric t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten (2009), Maaten and Hinton (2008)). As a dimensionality reduction method derived from t-SNE, it uses a deep neural network to optimize the projection into the latent space by minimizing the Kullback-Leibler divergence from the original space. Although the t-SNE method is an excellent technique for data visualization in a low dimensional space, it requires a huge computational cost for the optimization. Particularly, in order to map new incoming data vectors, the optimization must be run for the entire set again. To avoid this, a deep network is used to learn the parametric projection from the original space to the latent space, so that new data vectors can be mapped more easily, hence the name parametric t-SNE.

⋆ The project is sponsored by Shanghai SupeZET Engineering Technology Co.Ltd, according to the grant, GR-00001459.

This parametric form of the t-SNE can be applied to process monitoring as a visualization of chemical process information. In the 2D space, different operating states of the process are separated on a map, and faulty states are isolated from normal regions of operation.

Nevertheless, we observed that the conventional parametric t-SNE model could be a victim of greedy learning if it was only trained with historical data. In other words, the model only learns features from the most contributing variables of the historical data (those with the most variance). Variables that show less variance in the training are muted in the model. Although this greedy result is meaningful for feature learning, it can be misleading for visualization purposes. Unknown or faulty operating states caused by abnormalities in variables which exhibit tiny variation in the historical record can be problematic, as they can be projected onto the same area of the map as known regions. To improve the conventional parametric t-SNE method, rather than only using historical data, we create variation on the given data base combinatorially.

To validate the effectiveness of the proposed method, it was tested on the benchmark process: Tennessee Eastman Process (TEP). Four TEP faults (Fault 1, 4, 11 and 14) representing different types of faults are selected in the experiment. Additionally, the mapping result is compared with that from conventional dimensionality reduction methods. To validate the hypothesis that the variation introduction enhances the model sensitivity to input variables, Sobol's method (Sobol (1993)) is utilized. The corresponding sensitivities between the conventional parametric t-SNE method and the proposed method are compared.

The remainder of the paper is organized as following. In section 2, the overall methods are introduced. In section 3, we will introduce the case study, the Tennessee Eastman process (TEP). In section 4, we will show the result tested on TEP.

2. METHOD

In this section, the overall background and method are introduced. We start from the t-SNE method, given that parametric t-SNE is just an alternative form of the t-SNE method. Then, the parametric t-SNE and the proposed improvement are discussed. In the end, Sobol's method for sensitivity analysis is used to evaluate the response of the model to each input.

2.1 t-distributed stochastic neighbor embedding (t-SNE)

t-SNE is developed from stochastic neighbor embedding (SNE) (Hinton and Roweis (2003)). It uses a Student t-distribution with a heavy-tailed probability distribution to solve the crowding problem found in the original SNE method. Denote the probability distribution in the original space as p_{ij}:

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{k\neq l}\exp(-||x_k - x_l||^2/2\sigma^2)} \tag{1}$$

And denote the distribution in the latent space as q_{ij}:

$$q_{ij} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k\neq l}\exp(-||y_k - y_l||^2)} \tag{2}$$

The cost function that minimizes the Kullback-Leibler divergences between high-dimensional space and the latent space is given as:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

Using the gradient descent method to optimize the cost function, the distribution in the original space can be expressed on the low-dimensional map.

2.2 Parametric t-SNE

Parametric t-SNE is proposed to avoid the heavy optimization of the t-SNE method when applied to the same data set. Taking advantage of the learning ability of deep networks, a feed-forward neural network is used to learn the parametric mapping from high-dimensional space. There are two stages in the training procedure, namely pretraining with restricted Boltzmann machine (RBM) (Hinton (2010); Hinton et al. (2006)) and fine-tuning using the cost function from t-SNE.

RBM Training a deep network with multiple hidden layers is challenging, because a typical deep network can contain millions of parameters. Activation functions used as nonlinear transformation (e.g. the sigmoid function) can cause gradient vanishing and exploding issues. This makes tuning extremely difficult for the parameters in earlier layers. Until mid-2000s, the development of RBM and its application of layer-wise greedy training on deep networks (Hinton and Salakhutdinov (2006); Hinton (2010)) brought breakthroughs in the parameter initialization of deep network.

The RBM is a generative artificial neural network that learns the probability distribution on a given data set using energy function models:

$$p(x) = \frac{e^{-E(x)}}{Z} \tag{4}$$

where Z is called a partition function: $Z = \sum_x e^{-E(x)}$

The standard type of the RBM has one hidden layer and one visible layer. The structure of the RBM is illustrated in Figure 1.

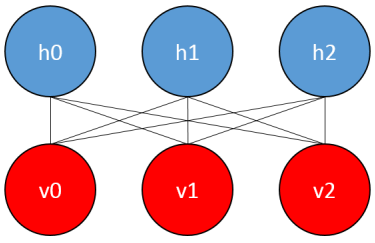


Fig. 1. Illustration of the RBM

The probability of a data vector in the hidden layer can be written as:

$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-E(x,h)}}{Z} \tag{5}$$

In the RBM, the binary energy function is defined as:

$$E(v,h) = -b'v - c'h - h'Wv \tag{6}$$

where b and c are the bias of visible and hidden layer to be learned through training. In practice, the RBM can be extended to Gaussian distribution for real data:

$$E(v,h) = \sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \boldsymbol{c'h} - \boldsymbol{h'Wv} \tag{7}$$

To train the RBM, the objective is to maximize $P(v)$, the log-likelihood function at single data point v, calculated as:

$$\begin{aligned}\frac{\partial \log P(\boldsymbol{v})}{\partial \theta} &= \sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{v}) \frac{\partial[-E(\boldsymbol{v},\boldsymbol{h})]}{\partial \theta} \\ &- \sum_{\tilde{\boldsymbol{v}}} \sum_{\boldsymbol{h}} P(\tilde{\boldsymbol{v}},\boldsymbol{h}) \frac{\partial[-E(\tilde{\boldsymbol{v}},\boldsymbol{h})]}{\partial \theta}\end{aligned} \tag{8}$$

The second term on the right hand of Equation 8 is approximated by contrastive divergence (CD-k) to accelerate the optimization process. The Gibbs sampler was utilized to sample v and h at different time steps from a Markov chain (Figure 2). In the Gibbs sampling, k represents the sampling step. Although v and h should be ideally sampled at $k = \infty$, the high computational cost make such approach impractical. In practice, k is always set as 1 to get a second-order approximation, which has been proved effective and reliable. The RBM is used in layer-wise training, and the trained RBMs are combined into a single network. More details about training a RBM can be found in Hinton (2010).

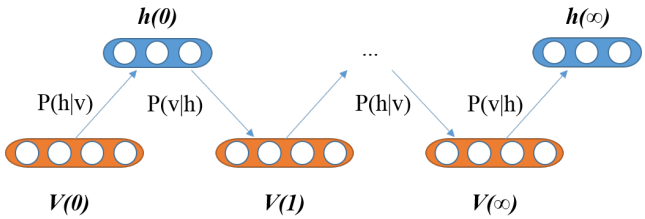


Fig. 2. Gibbs sampling and contrastive divergence of RBM training

Finetuning After pre-training that model parameters are initialized, the network is fine-tuned by the t-SNE cost function (Equation 3) using gradient-based backpropagation. In the feed-forward network, the term of q_{ij} is modified as follows:

$$q_{ij} = \frac{(1 + ||f(x_i|W) - f(x_j|W)||^2/\alpha)^{-(\alpha+1)/2}}{\sum_{k \neq l}(1 + ||f(x_k|W) - f(x_l|W)||^2/\alpha)^{-(\alpha+1)/2}} \tag{9}$$

The overall procedure of the training is summarized in Figure 3. The architecture of the network is 31-32-128-64-32-2, which is obtained from multiple tests minimizing the cost function. Adam (Kingma and Ba (2014)) optimizer is used to fine-tune the model.

The parametric t-SNE model is implemented in *python* 3.5 environment, and *tensorflow* is used as the deep learning framework for pre-training and fine-tuning.

2.3 Combinatorial variation creation

Following above procedures to train the parametric t-SNE with historical data, it was observed that the model

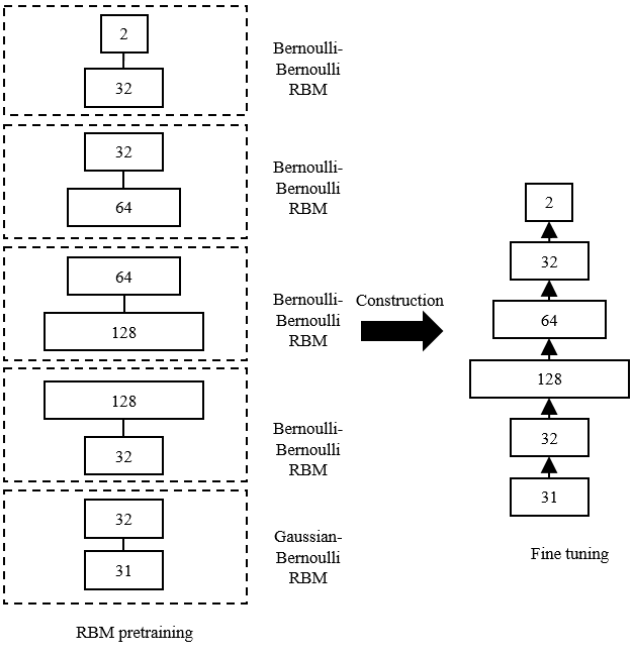


Fig. 3. Illustration of the training procedure. In our work, we use a 31-32-128-64-32-2 deep network, where 31 is the number of input variables and 2 corresponds to the 2D output dimension. For pre-training and finetuning stages, the epochs were set as 100 and 60 respectively. A configuration with four hidden layers (32, 128, 64 and 32) is used and gives the best visual performance.

can easily become a greedy learner, causing the model to respond to only a few dominating inputs. Although this type of learning is crucial for feature learning and extraction, it leaves the model weak to future faults that occur because of the muted variables. Faults unfamiliar to the model or operation modes with high contribution from the muted variables can be incorrectly projected onto areas of the map that are known to contain data from normal operation only. This lack of separation between states of operation can mislead operators. To relieve this effect, in the fine-tuning stage, we introduce random variation to make the model respond more evenly to every input variable.

After testing with multiple scenarios to introduce variation, the combinatorial variation creation is the most effective method to make the visualization more robust. The method aims to evenly add variation on each input variable in the data set. Figure 4 demonstrates the procedure of the process. The variation level δ used is 15 for the TEP data, which means a variation around 15 times the standard deviation from the original set is introduced into each selected variable. A total of 100 combinations of random subsets are created using the proposed method, where for each combination, 20% of the given data are sampled for variation introduction.

2.4 Sensitivity analysis

Sensitivity analysis is commonly used in model evaluation for a broad range of objectives including robustness testing of the model, model simplification, and in identifying the

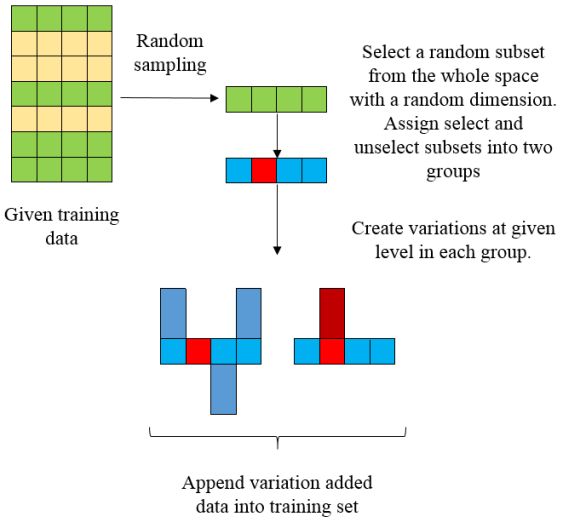


Fig. 4. Illustration of the proposed combinatorial variation creation method. From original training set, a number of data are randomly sampled for variation creation. With a random dimension d, a subset is chosen following the combinatorial rule. The variables in the subset and the rest of the unselected variables are assigned into two groups. A variation of δ times of standard deviation is added into the variables for each group.

key correlation between inputs and outputs. In this work, Sobol's method is used to quantify the greedy learning effect that causes the model to be governed by the most contributing variables, whereas the model is less sensitive to the least contributing variables. Sobol's method is a variance-based sensitivity analysis method. Under the assumption that model variables are independent, the total variance of the model output can be decomposed to the variance of each input:

$$V_y = \sum_{i=1}^{d} V_i + \sum_{i<j}^{d} V_{ij} + \ldots + V_{12\ldots d} \tag{10}$$

where V_y, V_i, V_{ij}, and n denote the variance of the output, the first order contribution of the i^{th} variable, the second order contribution from the interactive effect from i^{th} and j^{th} variables, and the number of variables correspondingly. Thus, the first, second, and total order sensitivity indices are expressed respectively as:

$$S_i = V_i/V(y) \tag{11}$$
$$S_{ij} = V_{ij}/V(y) \tag{12}$$
$$S_{Ti} = 1 - V_{\neq i}/V(y) \tag{13}$$

Generally, the variances are approximated by Monte Carlo numerical integration to reduce the computational cost. In this work, Saltilli's method(Saltelli et al. (2008)) is adopted for variance approximation.

In practice, Sobol's method is always appropriate for obtaining sensitivity for a single-output model. Therefore, in order to analyze the input sensitivity, we take advantage of the characteristics of the model that tries to optimize the KL divergences between the original space and the latent space. The model is forced to project the high dimensional space into one dimension, where Sobol's method can be applied.

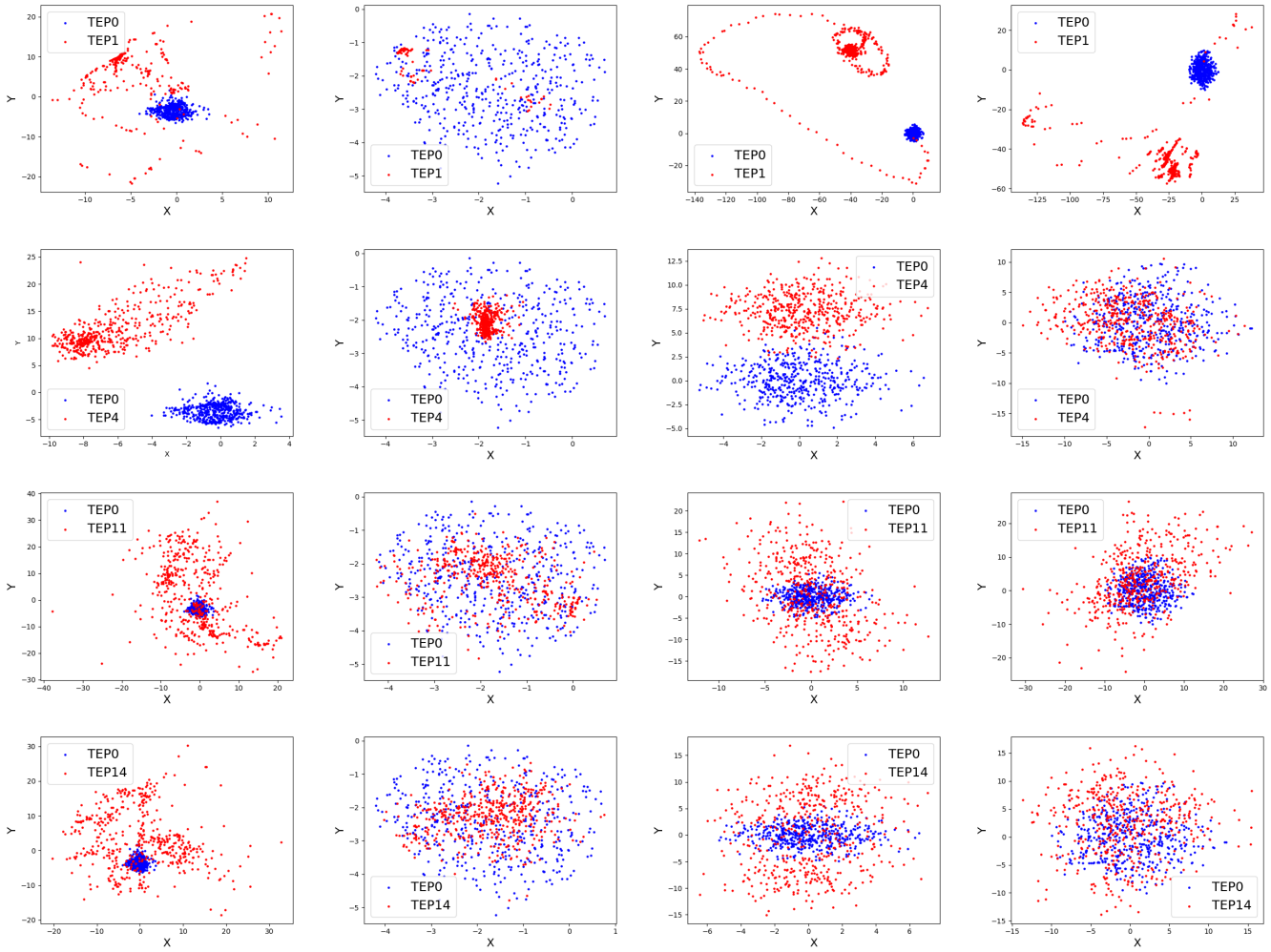
3. CASE STUDY: TENNESSEE EASTMAN PROCESS(TEP)

TEP is a realistic chemical process simulation originally developed by Downs and Vogel (Downs and Vogel (1993)). It is widely used as a benchmark to evaluate process control and monitoring tasks. From the total 53 measurements variables, we selected 31 variables for the study including 22 measured variables and 9 non-constant manipulated variables. In the proposed method, combinatorial variation creation is introduced to normal data for training. Three typical types of faults including step faults, random variations, and sticking faults (Fault 1, 4, 11 and 14) are selected to validate the visualization performance over a diverse set of process states. To simulate the real process operation, the dataset used in the test included 48 hours of operation. The first 24 hours were normal condition, and the faults were introduced in the following 24 hours. Conventional dimensionality reduction methods with linear and non-linear techniques such as PCA and Isomap are compared with the proposed method for 2D visualization. The objective is to notably distinguish normal and faulty data in the 2D map either by position or distribution difference. This feature of the 2D map can be intuitive and informative for operators in a real plant. Overlapping of the normal and fault data indicate the failure of the method to provide useful process information.

4. RESULT AND DISCUSSION

The testing results are summarized in Figure 5. For a simple step fault, fault 1, all methods achieve a good visualization effect where normal and faulty regions are well separated. Nevertheless, for a more complicated fault, only the parametric t-SNE trained with the proposed method is effective to visualize faulty data clearly on the 2D map. For step faults (fault 1 and 4), the model can separate the normal and faulty data points. For fault 11 that is caused by random variation of reactor cooling water inlet temperature, the faulty and normal data can be differentiated by their corresponding distribution that normal data are closely clustered and faulty data are dispersed around. Same result can be observed for fault 14 that is a sticking fault caused by reactor cooling water value that the difference can be observed from the distribution. For PCA and Isomap, neither of them is able to provide noticeable separation between faulty and normal data. Faulty data are overlapped on the normal region and the distribution for both classes is also similar.

Although the conventional parametric t-SNE method shows responses to faulty data which are clustered in a small region on the map, it is still unable to separate the faulty data out of the normal region. Since only normal data is provided in the training stage, the network is unable to learn the prototypes outside the normal region, which causes the failure of fault separation in the 2D space. In the proposed method, many random variations are created to simulate possible unknown conditions away from the normal condition, which balances the distribution of each variable. This addition enhances the robustness of the model when confronting possible unknown faults in the future. By this method, the learning capability of a deep



(a) Proposed parametric t-SNE (b) Native parametric t-SNE (c) PCA (d) Isomap

Fig. 5. Comparison of the 2D visualization for four TEP faults (Fault 1, Fault 4, Fault 11 and Fault 14). Each row of the subplots represents the different methods on the same TEP fault, and each column represents the projection of a certain method on different TEP faults.

network can be utilized to learn how to map data outside the normal region.

The sensitivity analysis using the Sobol's method can provide in-depth details to explain the performance difference between the proposed and the conventional parametric t-SNE method. The result is summarized in Figure 6. The sensitivity indices for each variable are averaged from multiple runs of the model with different types of random variation included in the training. Through the comparison in Figure 6, it suggests that the proposed training method using variation creation gives a more even sensitivity distribution. In other words, the model can more equally express the variation in each input instead of only responding to the most contributing variables learned from the training set. Besides, the average magnitude of the sensitivity indices in the proposed method is higher than that without random variation creation. Hence, variation in the input variables can be better expressed on the 2D map, since more variables can have impact on the output, which also explains why the proposed method improves the conventional parametric t-SNE method.

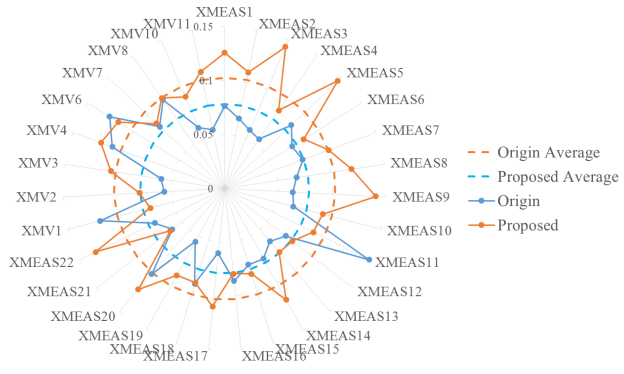


Fig. 6. Sensitivity analysis using Sobol's method to analyze the model sensitivity of each input variable. The deep network using the proposed combinatorial variation creation method is more evenly sensitive to all inputs, compared with the model trained with only historical data.

5. CONCLUSION

In this work, we introduced a process visualization method using parametric t-SNE to provide high-level process information in the form of a 2D map. Instead of monitoring a line-chart diagram for a single variable in the DCS systems, the proposed method can extract features from multiple process variables and indicate patterns corresponding to different process behaviors. A deep network is used to learn the mapping process that can retain the probability distribution of the original space. In practice, the greedy learning effect was observed, causing the model to prefer learning from variables causing process variation in the historical data. We applied combinatorial variation creation to enhance the robustness of the parametric t-SNE model. Such method could be also applied in cases that the training samples are imbalance either for the distribution of each variable or sample amounts among different classes, which could potentially lead the model to learn biased information. The overall method is tested on the TEP data set. It outperforms conventional dimensionality reduction methods in visualizing complicated faults in the TEP dataset. Such method can be applied in current DCS monitoring interfaces for any existent processes that eases the monitoring effort of multiple variables.

REFERENCES

Downs, J.J. and Vogel, E.F. (1993). A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3), 245–255.

Hinton, G. (2010). A practical guide to training restricted boltzmann machines. *Momentum*, 9(1), 926.

Hinton, G.E., Osindero, S., and Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527–1554.

Hinton, G.E. and Roweis, S.T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems*, 857–864.

Hinton, G.E. and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504–507.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Maaten, L.v.d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov), 2579–2605.

Robertson, G., Thomas, M., and Romagnoli, J.A. (2015). Topological preservation techniques for nonlinear process monitoring. *Computers & Chemical Engineering*, 76, 1–16.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.

Sobol, I.M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4), 407–414.

van der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. *RBM*, 500(500), 26.

Wang, R., Edgar, T.F., Baldea, M., Nixon, M., Wojsznis, W., and Dunia, R. (2017). A geometric method for batch data visualization, process monitoring and fault detection. *Journal of Process Control*.

Yiakopoulos, C., Gryllias, K., Chioua, M., Hollender, M., and Antoniadis, I. (2016). An on-line sax and hmm-based anomaly detection and visualization tool for early disturbance discovery in a dynamic industrial process. *Journal of Process Control*, 44, 134–159.

Zhong, B., Wang, J., Wu, H., Zhou, J., and Jin, Q. (2016). Som-based visualization monitoring and fault diagnosis for chemical process. In *Control and Decision Conference (CCDC), 2016 Chinese*, 5844–5849. IEEE.