Korean J. Chem. Eng., 38(12), 2406-2422 (2021)
DOI: 10.1007/s11814-021-0894-1

pISSN: 0256-1115
eISSN: 1975-7220

Data-driven fault detection for chemical processes using autoencoder with data augmentation

Hodong Lee*, Changsoo Kim, Dong Hwi Jeong***,†, and Jong Min Lee*,†**

*School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea
**Clean Energy Research Center, Korea Institute of Science and Technology, Seoul 02792, Korea
***School of Chemical Engineering, University of Ulsan, 93, Daehak-ro, Nam-gu, Ulsan 44610, Korea
(*Received 21 May 2021 • Revised 5 July 2021 • Accepted 5 July 2021*)

Abstract−Process monitoring plays an essential role in safe and profitable operations. Various data-driven fault detection models have been suggested, but they cannot perform properly when the training data are insufficient or the information to construct the manifold is confined to a specific region. In this study, a process monitoring framework integrated with data augmentation is proposed to supplement rare but informative samples for the boundary regions of the normal state. To generate data for augmentation, a variational autoencoder was employed to exploit its advantage of stable convergence. For the construction of the process monitoring system, an autoencoder that can extract useful features in an unsupervised manner was used. To illustrate the efficacy of the proposed method, a case study for the Tennessee Eastman process was applied. The results show that the proposed method can improve the monitoring performance compared to the autoencoder without data augmentation in terms of fault detection accuracy and delay, particularly within the feature space.

Keywords: Process Monitoring, Fault Detection and Isolation (FDI), Autoencoder, Variational Autoencoder, Data Augmentation

INTRODUCTION

Multivariate statistical process monitoring (MSPM) is an indispensable part of the successful operation of chemical processes used to guarantee the safety and quality of the products. The various MSPM methods can be classified into two approaches: prior knowledge-based methods, such as first principle equations or empirical equations, and historical data-driven methods [1]. Historical data-driven methods have the advantage of generality, and thus there is no need for process-specific domain knowledge. These have the advantage of general applicability owing to the fast and straightforward model construction in general. Data-driven process monitoring models make use of the data under normal operation in developing the monitoring statistics and defining a boundary of normal states that detect the process faults by checking whether the online monitoring statistics violate the boundary. Conventional multivariate statistical models using latent variables for process monitoring, such as principal component analysis (PCA) and partial least squares (PLS), have been widely used as dimensionality reduction methods. PCA, which defines orthogonal latent variables that maximize the variance of the original data, is used as a dimensionality reduction method for monitoring in a reduced dimensional feature space. PLS is an extension of PCA that incorporates quality variables under inspection. Independent component analysis (ICA) utilizing higher-order statistics, unlike PCA, which only employs second-order statistics such as the mean and variance, performs better on data following a non-Gaussian distribution. However, it still has certain limitations with respect to the nonlinearity of the data owing to a linearity assumption. To deal with nonlinearity, kernel PCA (KPCA) has been suggested [2]. KPCA exploits the kernel trick to map the nonlinear data into a higher dimensional linear space, such that it can perform feature extraction better than a directly applied PCA on nonlinear data. However, it is limited in that the computational complexity in the kernel method increases exponentially as the number of dimensions and samples increases. In addition, the kernel method has limitations in that it exhibits an inconsistent performance that is significantly dependent on the kernel type and hyperparameters. Autoencoders (AEs), which are a type of neural network for unsupervised dimensional reduction, have recently been suggested as a notable alternative to overcome these limitations with the help of recent advances in machine learning techniques. Various studies have demonstrated that AEs show a better performance compared to a conventional dimensionality reduction method in process monitoring [3].

Hinton [4] compared the performance of the AEs and PCA as a conventional dimensionality reduction technique for various types of data. Notable results also suggest that AEs achieve a better performance in reducing the dimensionality of the data than conventional methods, such as PCA, ICA, and KPCA when sufficient computational resources, sufficient numbers of training data, and a plausible initialization of the weight parameters are secured [4]. Since it was first reported by Hinton, many studies on process moni-

†To whom correspondence should be addressed.
E-mail: jdonghwi@ulsan.ac.kr, jongmin@snu.ac.kr

toring using AEs have been actively conducted and have proven the competitiveness of AEs when combined with nonlinear activations in terms of the effectiveness of nonlinear feature extraction in process monitoring [5,6]. Advancing from the classical form of AEs, various AEs used to cope with noisy process data have been proposed, such as denoising autoencoder (DAE) [7], contractive autoencoder (CAE) [8], and robust autoencoder [9]. DAE and CAE were used to demonstrate the improvement in monitoring performance over the basic AE and PCA for the Tennessee Eastman process (TEP), a benchmark chemical process selected as the target process in the present study [3]. As a new structure of autoencoder-based process monitoring system, parallel autoassociative neural network [10] have been proposed and demonstrated for the same benchmark process, TEP. The AEs were integrated with another averaging approximator such as k-nearest neighbor (kNN) to suggest a newly refined monitoring metric [11] or combined with a regularization method such as elastic net to enhance the robustness of the monitoring model under abundant training data [12]. Even if sufficient amounts of training data can be provided, a data-rich but information-poor problem still remains, resulting in a typical overfitting issue of the models [13]. For this reason, diverse attempts have been made to supplement information through data augmentation, which makes manifold learning robust for both overfitting and underfitting.

Data augmentation techniques can be classified into two approaches: conventional methods and generative models. Two representative methods for conventional data augmentation have been developed in the fields of image processing and computer vision applications: data warping [14] and the synthetic minority over-sampling technique (SMOTE) [15]. Data warping involves the synthesis of data by applying a deformation from intuitive features in the original data space, such as translation, rotation, and skewing from existing samples. Although SMOTE can be applied in both the data space and the feature space to produce artificial samples, it was primarily proposed to alleviate class imbalance problems during classification. Being implemented through an affine transformation in the feature space as well, SMOTE has the advantage of being applied independent of the applications owing to the fact that a feature space can represent the salient structure of the data. Wong [16] reported that both warping and SMOTE can improve the performance of a classification model.

The generative model that belongs to the neural network-based method can be categorized into two groups: variational autoencoder (VAE) and generative adversarial network (GAN). Unlike a conventional method, generative models generally estimate the underlying distribution in the feature space. Based on the feature space, new vectors, which are latent vectors, are sampled and then fed into a generator unit corresponding to each generative model to create artificial data. By leveraging the inherent manifold knowledge rather than directly manipulating the sample data in the original space, the generative modeling approach has proven its superiority in terms of the quality and effectiveness of augmentation in previous studies conducted in diverse fields [14,15]. This property becomes more significant as the number of dimensions increases because the Euclidean distance, commonly used as the distance metric, weakens the meaning as a similarity measure in the original space. To make use of the indispensable merits of stable convergence of VAE in modeling compared to those of GAN, which possesses an adversarial training process between two networks, VAE has been employed to augment the supplementary training data. Because most of the chemical process data might not violate the assumption of VAE in which the class for encoding and the prior distribution are restricted as multivariate Gaussians, it makes use of the VAE characteristic in which the latent vectors can be sampled from the explicit distribution in the feature space. This enables the manipulation of the latent vectors to reflect the intention of the data augmentation, such as selective sampling within the boundary regions of a normal distribution, which corresponds to rare samples. The capability of a selective production of artificial data to convey the intention for augmentation can contribute to the improvement of the process monitoring modeling by providing insufficient information.

Various studies have used generative models as tools for data augmentation, particularly in the field of computer vision. Several studies have improved the performance of image classifiers through data augmentation using generative models, such as VAEs, GANs, and their variants [15-17]. The models for speech recognition [21] or translation [17] can be supported by data augmentation techniques to alleviate the class imbalance problem or allow the reuse in another domain, as applied in transfer learning. Although they applied a variant of GAN to construct a generative model, and not VAE, Gao et al. [22] suggested that augmentation in the case of process data can also contribute to improvements as a classifier for process monitoring.

This research was motivated by previous studies promoting the quality of manifold learning, which is an essential part of modeling for fault detection, through data augmentation. Integrated with the idea of an exclusive augmentation of data that rarely appear but should be classified as a normal state similar to the training dataset, the proposed method suggests a framework to boost the monitoring performance for fault detection by supplementing the insufficient information of the training dataset. Based on a specific strategy to reflect the intention of the augmentation, an edge-based oversampling scheme [23] is utilized with a general transformation to explicitly aim the boundary region of the normal state within the feature space [24]. The synthetic samples generated from the latent vectors of the border of a normal region are augmented in the training dataset to promote manifold learning by imposing more weight.

The remainder of this paper is organized as follows. In Section 2, the preliminaries of the proposed method are introduced. A description of the Tennessee Eastman process which is the target process used in the case study, the data augmented monitoring methodology, and the implementation results of the TEP are presented in Section 3. In Section 4, the results of the case study are discussed. Finally, Section 5 provides concluding remarks and areas of future study.

PRELIMINARIES

1. Autoencoder

AE is an unsupervised machine-learning technique for feature

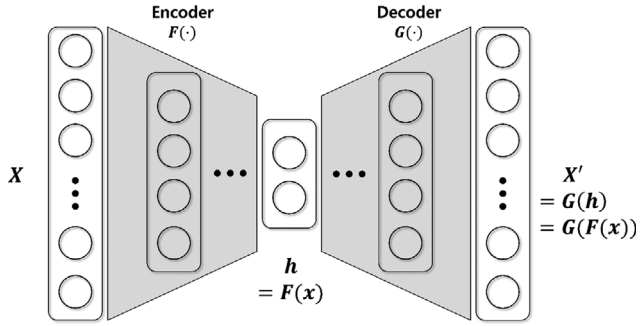

Fig. 1. Conceptual scheme of AE.

extraction. It encodes the input data onto low-dimensional latent features and reconstructs the data using only the encoded feature by squeezing the middle layer of the symmetric network, as shown in Fig. 1. The compression part of the network, which produces a latent feature, is the encoder and the opposite part, which is the decoder, applies the converse functionality. The encoder function mapping an input $x \in R^n$ into a latent vector $h \in R^m$ through general functions is as follows:

$$h = F(x) = f(W_1 \cdot x + b_1), \tag{1}$$

where W_1 is an m×n weight matrix, b_1 is an m×1 bias vector, and $f(\cdot)$ is an activation function. The activation for hidden layers typically employs nonlinear functions such as a sigmoid, tangent hyperbolic, and rectified linear units, except for the visible layers having linear activations. By adopting a bottleneck structure in the latent space, AE is guided to extract a rich representation that is advantageous to the reconstruction of the input. The reconstruction of h is as follows:

$$x' = G(h) = g(W_2 \cdot h + b_2), \tag{2}$$

where W_2 is an n×m weight matrix, b_2 is an n×1 bias vector, and $g(\cdot)$ is an activation function. The weight matrices W_1 and W_2 have distinct weight values in general, but can be tied, i.e., $W_1 = W_2^T$, in some cases.

The objective function of an AE, which is the loss function that the optimizer should minimize, has different forms depending on the data type, such as the squared error and cross-entropy. Following the typical choices in the cases of linear regression, the reconstruction loss function across a given set of training samples, D, can be represented as follows:

$$L = \min_{W,b} \Sigma_{x \in D} \|x - x\|^2 / |D| = \min_{W,b} \Sigma_{x \in D} \|x - G(F(x))\|^2 / |D|. \tag{3}$$

The network can be extended to an arbitrary number of hidden layers and nodes in both the encoding and decoding parts. Special attention is needed to determine the dimensions of networks, depending on the applications to prevent underfitting and/or overfitting. As a typical approach in machine learning, regularization methods such as weight regularization, where the objective function includes the norm of the weights [25], dropouts [26], batch normalization [27], and pruning [28,29] can implicitly help a network avoid overfitting starting from a network with sufficient model capacity.

The operations through weights and biases used to reveal the latent vector h correspond to the projection of input data from the original to feature space in PCA. If linear activation replaces the nonlinear functions, AE is reduced to PCA, which is conceptually equivalent [30]. Thus, AE is a nonlinear generalization of PCA, which is a conventional dimensionality reduction method used for purposes such as feature extraction, visualization, and data compression. Although KPCA [2], a nonlinear extension of PCA using kernel trick, can be compared with AE using the same nonlinear dimensionality reduction method, the performance of KPCA depends entirely on the type of kernel and represents poor robustness against the kernel parameters, depending on the applications. However, AE copes inherently with nonlinearity through nonlinear activation functions in each layer. Meanwhile, there exist variants of AE to improve the limitations in terms of robustness against process noise. Denoising autoencoder (DAE) [7] can improve the robustness by intentional random corruption of the input data to promote the reconstruction ability even under noisy situations. For a similar purpose, contractive autoencoder (CAE) [8] was devised by explicitly penalizing the objective function by adding a term representing the sensitivity of hidden representations to the input perturbations, $\|J_f(x)\|_F^2 = \Sigma_{ij}\left(\frac{\partial h_j(x)}{\partial x_i}\right)^2$.

The construction of statistics for process monitoring using AE is carried out using the same procedure as that used in PCA. After the network training is completed, test statistics defined in the two spaces are used to monitor the abnormality of a process. One is H^2, which is the squared scalar value of the latent vector corresponding to T^2 in PCA calculated in the feature space, and the other is the squared prediction error (SPE) in the original space [3]. Subsequently, the control limit used to characterize the normal operating region is defined by a non-parametric density estimator called kernel density estimation (KDE). Based on the KDE results for the normal operation training samples, the 95 percentile values for each space are typically determined criteria for process monitoring. After the offline training procedure based on a set of training samples is finished, the test samples are mapped into the low-dimensional manifold and reconstructed into the original dimensional space online. Process monitoring is conducted by comparing the statistics of the test data to the control limit in each space.

2. Variational Autoencoder

VAE is a popular generative model that learns the data distribution to generate new samples aside from existing data in an unsupervised manner. Once the training of VAE is completed, the latent vector z to be used as the input for the generation process is sampled in the feature space. The main objective of VAE is to generate new synthetic samples of the original space using the latent vector z sampled from a low-dimensional feature space through a generation network that corresponds to the decoder, as shown in Fig. 2. Meanwhile, it is insufficient to train a generation network to generate plausible samples with only randomly sampled vectors drawn from a prior distribution p(z), which is typically assumed as a normal distribution that possesses little information producing meaningful samples. Thus, the encoder network is introduced to provide evidence to produce a latent vector that allows the decoder to reconstruct at least the training samples well. At this point, the

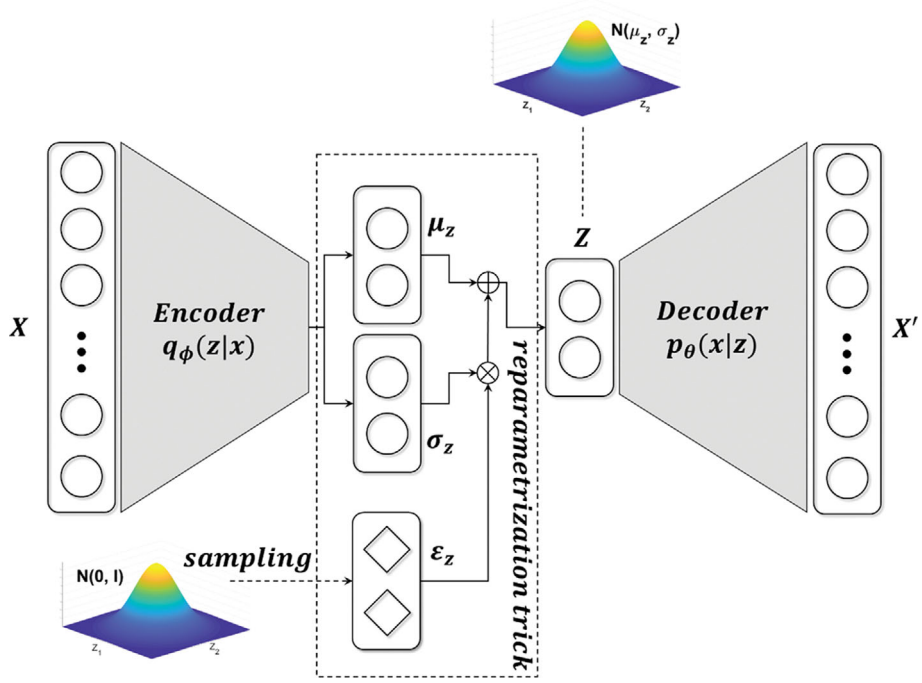

Fig. 2. Conceptual scheme of VAE and reparameterization trick.

true posterior p(z|x), which is generally intractable, is replaced by the approximated posterior $q_\phi(z|x)$ parameterized by ϕ, which is typically assumed as a multivariate Gaussian, leading to a closed-form loss term, that is the variational inference method.

As a result, the entire structure incorporating an inferential encoding network and a generative decoding network is analogous to AE in terms of compressing the input data into a low-dimensional latent space and then restoring the data. Thus, the methodology, which is an autoencoder with a variational inference method for generative modeling, is called a variational autoencoder. Given the ultimate purpose of development and the process of establishing a generative model, the basis of VAE has little to do with AE, except for the structural similarity of the final form of the objective function [24].

For a vanilla VAE, first introduced by Kingma et al. [4], the objective function is

$$\log p_\theta(x^{(i)}) = D_{KL}[q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})] + L(\theta, \phi; x^{(i)}), \quad (4)$$

where the first term on the right-hand side, which represents the variational inference process, forces the approximate posterior $q_\phi(z|x^{(i)})$ to match the true posterior $p_\theta(z|x^{(i)})$ by using Kullback-Leibler (KL) divergence, and the second term is the evidence lower bound (ELBO) on the marginal likelihood of data point i. Because the KL divergence is non-negative, the marginal likelihood is greater than the ELBO. The ELBO can be further decomposed as follows:

$$\log p_\theta(x^{(i)}) \geq L(\theta, \phi; x^{(i)})$$
$$= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z)). \quad (5)$$

Instead of directly maximizing the marginal likelihood, the ELBO is maximized with respect to both the variational parameters ϕ and the generative parameters θ. It is noteworthy that the reparameterization trick suggested by Kingma et al. [4] makes it possible for the VAE formulation to jointly optimize both parameters in the encoder and decoder by utilizing the stochastic gradient descent method, even though it includes a non-differentiable sampling process, as shown in Fig. 2. In addition, by assuming both the prior $p_\theta(z)$ and the inferential posterior $q_\phi(z|x^{(i)})$ as having a multivariate Gaussian distribution, and the generative posterior $p_\theta(x^{(i)}|z)$ as a multivariate Gaussian or Bernoulli distribution depending on the application, the ELBO can be represented as a closed form using the parameters of the encoder and decoder network. The detailed proof and formula can be found in the study by Kingma et al. and the appendix thereof [4]. In conclusion, the two terms in Eq. (5) can be, respectively, interpreted as a reconstruction error and a regularization encouraging the approximated posterior to fit into the prior, which will eventually be used as a sampling distribution.

Meanwhile, InfoVAE [31] has recently been proposed to improve the problem of vanilla VAE ignoring the latent vectors, i.e., so-called uninformative latent vectors, because it has been shown that a decoding network with sufficient capacity can take over the role of reconstructing inputs even with meaningless random vectors. Thus, the latent vectors, which must potentially retain significant information needed to restore the data, are forced to fit the prior distribution by minimizing the second term in Eq. (5). This is a fatal limitation in that a latent vector cannot contain any data features, particularly when there is a significant manipulation to impose any intention in the latent space. Zhao et al. [31] introduced an additional regularization term in the objective that allows the encoded distribution in the latent space to preserve the data features. Organizing the ELBO objective of the original VAE as an

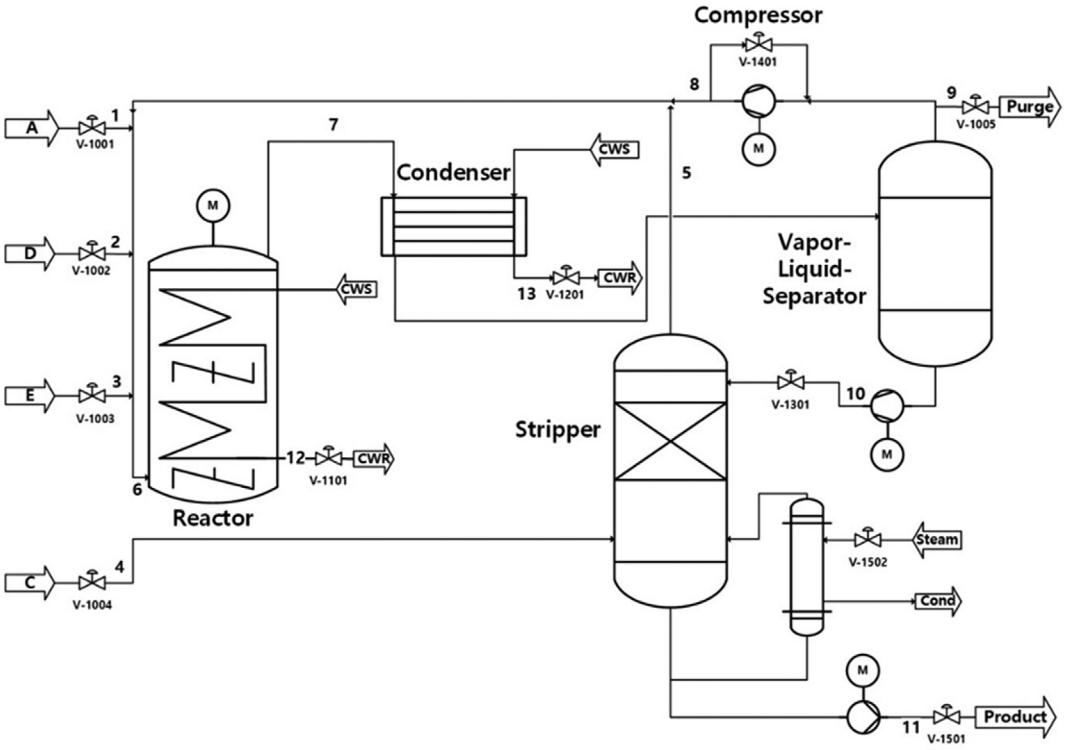

Fig. 3. Process flow diagram of Tennessee Eastman process [35].

equation,

$$L_{ELBO}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p(z))$$
$$= -D_{KL}(q_\phi(z)||p_\theta(z)) - \mathbb{E}_{q(z)}[D_{KL}(q_\phi(x^{(i)}|z)||p_\theta(x^{(i)}|z))] \quad (6)$$

The objective function of InfoVAE, including a mutual information maximization term that leads to meaningful features, is defined as follows:

$$L_{InfoVAE}(\theta, \phi; x^{(i)}) \equiv \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) + \alpha I_q$$
$$= \mathbb{E}_{p_D(x)}\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - (1-\alpha)\mathbb{E}_{p_D(x)}D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z))$$
$$-(\alpha-1)D_{KL}(q_\phi(z)||p(z)), \quad (7)$$

where the scaling parameter λ in the original study [31] was assumed to be one for simplicity. According to the proof and derivation in InfoVAE, the final form of the objective function can be computed by replacing the last term in Eq. (7) with an equivalent divergence family, i.e., the maximum mean discrepancy (MMD). Hence, an operation in the latent space can convey implications to be reflected on the generated data. It provides conditions under which operations in the learned latent space may have meaningful implications for the generated data, which cannot be utilized by an uninformative latent code.

PROPOSED MONITORING METHODOLOGY

1. Tennessee Eastman Process

The Tennessee Eastman process is a widely used benchmark chemical process for a performance comparison in process monitoring algorithms or control structures. It consists of five modules: a reactor, condenser, product separator, stripper, and compressor for the recycle stream, as shown in Fig. 3. The irreversible and exothermic gas-phase catalytic reactions of reactants A, C, D, and E occur to produce two liquid products, G and H. The following steps, including condensation, separation, and compression, recycle the unconverted reactants in the product streams and make up the fresh reactants rectified through the stripper. Some reactions involve inert gas B and byproduct F, which are primarily removed by the purge stream. A detailed description of the TEP can be found in the original suggestion of the Fortran [32] and revised MATLAB [33] models. The model used in this study contains the control strategy proposed by Ricker [34] based on the revised MATLAB model.

There are 41 measurements, i.e., 22 continuous variables and 19 composition variables from the installed analyzers. The model also includes 12 manipulated variables used in the process control. In this study, 50 variables, excluding three manipulated variables remaining as fixed values (compressor recycle valve, stripper steam valve, and agitator speed), were investigated. The target variables for the analysis are listed in Table 1. The MATLAB model modified based on the original Fortran model includes 28 pre-defined fault cases, and 8 (21-28) more fault cases were added to the 20 fault cases in the original model [32]. A total of 28 faults in the TEP cover various types, such as step change, random variation, slow drift, and sticking of a certain variable. The fault scenarios in the TEP are summarized in Table 2. The proposed methodology for process fault detection, which is described in detail in the following sections, is validated and analyzed using the TEP in the following sections.

Table 1. Process variables of TEP subject to process monitoring

Variable No.	Variable name	Variable No.	Variable name
1	A feed flowrate (stream 1)	18	Stripper temperature
2	D feed flowrate (stream 2)	19	Stripper steam flowrate
3	E feed flowrate (stream 3)	20	Compressor work
4	A & C feed flowrate (stream 4)	21	Reactor c/w outlet temperature
5	Recycle flowrate (stream 8)	22	Condenser c/w outlet temperature
6	Reactor feed rate (stream 6)	23-28	Reactor feed analysis (A-F mol%) (stream 6)
7	Reactor pressure	29-36	Purge gas analysis (A-H mol%) (stream 9)
8	Reactor level	37-41	Product analysis (D-H mol%) (stream 11)
9	Reactor temperature	42	D feed flow valve (stream 2)
10	Purge rate (stream 9)	43	E feed flow valve (stream 3)
11	Product separator temperature	44	A feed flow valve (stream 1)
12	Product separator level	45	A & C feed flow valve (stream 4)
13	Product separator pressure	46	Purge valve (stream 9)
14	Product separator under flowrate (stream 10)	47	Separator pot liquid flow valve (stream 10)
15	Stripper level	48	Stripper liquid product flow valve (stream 11)
16	Stripper pressure	49	Reactor c/w flow valve
17	Stripper under flowrate (stream 11)	50	Condenser c/w flow valve

Table 2. Process faults in TEP

No.	Description	Type
IDV(1)	A/C feed ratio, B composition constant (stream 4)	Step
IDV(2)	B composition, A/C ratio constant (stream 4)	Step
IDV(3)	D feed temperature (stream 2)	Step
IDV(4)	Reactor cooling water inlet temperature	Step
IDV(5)	Condenser cooling water inlet temperature	Step
IDV(6)	A feed loss (stream 1)	Step
IDV(7)	C header pressure loss–reduced availability (stream 4)	Step
IDV(8)	A, B, C feed composition (stream 4)	Random variation
IDV(9)	D feed temperature (stream 2)	Random variation
IDV(10)	C feed temperature (stream 4)	Random variation
IDV(11)	Reactor cooling water inlet temperature	Random variation
IDV(12)	Condenser cooling water inlet temperature	Random variation
IDV(13)	Reaction kinetics	Slow drift
IDV(14)	Reactor cooling water valve	Sticking
IDV(15)	Condenser cooling water valve	Sticking
IDV(16)	*Unknown (Deviation of heat transfer within stripper heat exchanger)	*Unknown (Random variation)
IDV(17)	*Unknown (Deviation of heat transfer within reactor)	*Unknown (Random variation)
IDV(18)	*Unknown (Deviation of heat transfer within condenser)	*Unknown (Random variation)
IDV(19)	*Unknown (recycle valve, stripper steam valve, underflow separator (stream 10), underflow stripper (stream 11))	*Unknown (Sticking)
IDV(20)	*Unknown	*Unknown
IDV(21)	A feed temperature (stream 1)	Random variation
IDV(22)	E feed temperature (stream 3)	Random variation
IDV(23)	A feed pressure (stream 1)	Random variation
IDV(24)	D feed pressure (stream 2)	Random variation
IDV(25)	E feed pressure (stream 3)	Random variation
IDV(26)	A & C feed pressure (stream 4)	Random variation
IDV(27)	Reactor cooling water pressure	Random variation
IDV(28)	Condenser cooling water pressure	Random variation

*Unknown: Uncovered by A. Bathelt in revised version of MATLAB model

2. Process Monitoring Integrated with Data Augmentation

In this section, we propose a method that makes use of the advantage of data augmentation, particularly of the boundary of the normal operation data, such that it can help the classifier generalize better in terms of the manifold learning of the normal state. The method for edge-based sampling and data generation, proposed under the term DOPING technique [23], showed an improvement of image classification for the well-known MNIST dataset. Although the study was tested in different domains and utilized a different type of generative model and classifier from this study, it revealed the effectiveness of data augmentation based on the edge of a certain class. The borderline-SMOTE [15], which is a modified minority over-sampling method used only to generate samples near the borderline of the minority class, also presents evidence of further improvements with the help of borderline samples.

In this study, we propose an approach to supplementing rare samples in the same class to mitigate the in-class data imbalance, unlike previous studies that augment the minority class to resolve the between-class imbalance. From the viewpoint of process fault detection, the proposed method was designed to augment relatively large amounts of rare samples that occur with low probability distributed within the boundary region of a normal state. As a result, by deliberately adding rare normal instances to the training data, the monitoring system can better perform in terms of increasing the fault detection rate (FDR) while keeping the false alarm rate (FAR) below the acceptable level.

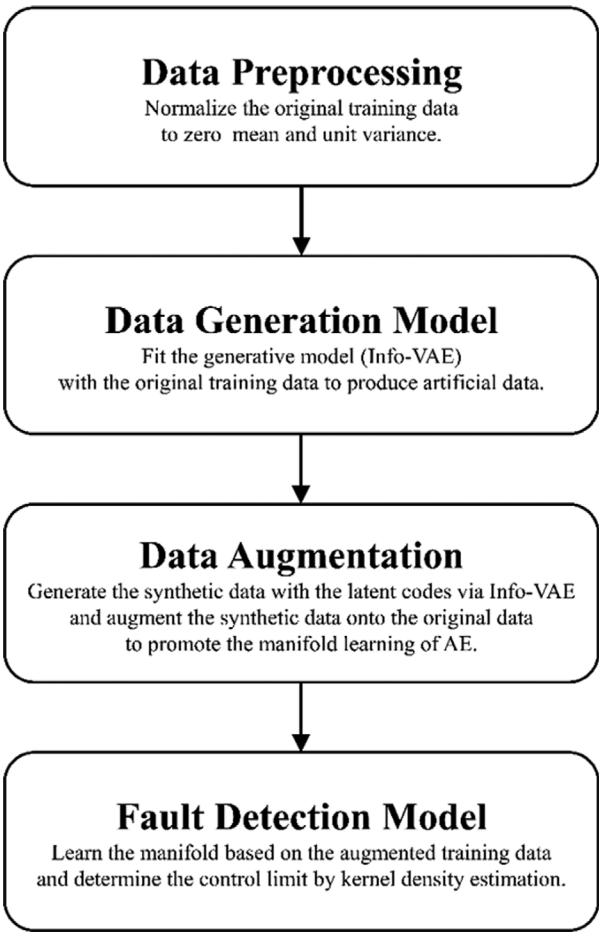

Fig. 4. Flowchart of the proposed method for process fault detection model.

The essential steps of the workflow are summarized in Fig. 4. To prepare the data at similar scales and variabilities, a preprocessing step is first required. The generative model using Info-VAE was trained on the original data to generate artificial data for augmentation. Once the generative model is prepared, various sets of data are generated by sampling in the latent space and retrieving data of the original space through the decoder network. The generated data are merged with the original data as the augmented training data for modeling the fault-detection model using AE.

2-1. Data Generation Using Info-VAE

Before employing the modeling of the generative and monitoring models, the data scaling process, which imposes equal importance against all variables in the model, works as a critical preprocessing, as in other machine learning algorithms. This is also important in terms of the stable convergence of the model, which is valid for all methods employed in this study. The standardization to scale each variable is as follows:

$$X_{scaled} = \frac{X - \mu}{\sigma}, \tag{8}$$

where X is the original variable, and μ and σ are the mean and standard deviation of each variable based on the training data, respectively. To establish the stopping criteria of the training process, the original dataset was divided into training and validation sets, each having 6,000 and 1,200 samples out of a total of 7,200 samples.

The structure of the Info-VAE used in this study is summarized in Table 3. First, the distributions of the inferential posterior, $q_\phi(z|x^{(i)})$, and the generative posterior, $p_\theta(x^{(i)}|z)$, are assumed to be multivariate Gaussians because process variables with continuous values are considered. The input layer dimension is matched with the dimensions of the benchmark process system, TEP, which has 50 variables, as suggested in Table 1. As one of the most critical parameters in the application of the autoencoders, the reduced dimensions of the latent space should be determined. Several heuristics exist to determine the dimensionality of the latent space, such as the elbow of the scree plot, the cutoff the eigenvalues greater than 1, or the cumulative percentage of explained variance (CPV). As shown in Fig. 5, the first criterion, the elbow of the scree plot, is

Table 3. Structure of the generative model using Info-VAE

Layer	Dimension	Activation	Remarks
Input	50	-	
Encoder 1	40	Leaky ReLU	Alpha: 0.2
Encoder 2 for Mean	30	Linear	
Encoder 2 for STD	30	Softplus	
Feature	30	-	
Decoder 1	40	Leaky ReLU	Alpha: 0.2
Decoder 2 for Mean	50	Linear	
Decoder 2 for STD	50	Softplus	
Output	50	-	

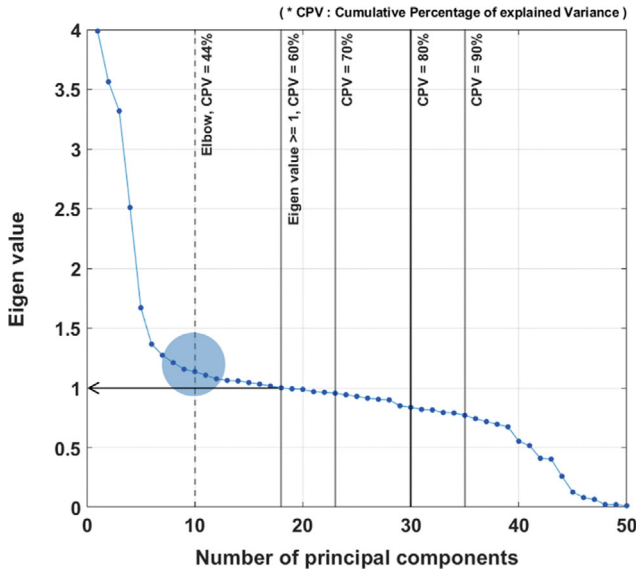

Fig. 5. Scree plot for selection of the reduced dimension of the latent space.

insufficient to account for the variance of the TEP data in the latent space because it results in excessive loss of information during dimension reduction. Therefore, a case study was performed to decide the dimension of the latent space that is the most efficient in terms of the monitoring performance. Based on the settings of the base case which will be introduced later in this section, the case study was conducted by varying the dimensionality of the latent space from 60% to 90% CPV. According to each case, the averages of the FDR over the 28 fault cases in TEP were derived and compared as Fig. 6. According to the result of the case study, it can be concluded that the 80% CPV is enough to efficiently reduce the dimension of the latent space based on the monitoring performance. Thus, the number of nodes in the bottleneck layer was set to 30. The result is reflected in the output dimension of the second layer in the encoder. In this study, the number of hidden layers between the input and output, which is another essential hyperparameter determining the performance of the model, was determined to be one in both the encoding and decoding networks to prevent an overfitting. The nonlinear activation functions for the hidden layers are set to the leaky rectified linear unit (ReLU), except for the output levels, such as the feature and reconstruction layers. In general, no activation function is adopted for the output layers in the regression models, which corresponds to the linear activations to fit the means of a multivariate Gaussian for the feature vector and reconstruction of the input. In the case of layers used to fit the standard deviations of the feature vectors or the reconstructions, another type of activation function, i.e., softplus activation, is employed to explicitly impose a positive definite constraint for the standard deviations. The relevant hyperparameters for configuring the generative model using Info-VAE are listed in Table 4. The weight parameter adjusting the relative importance between the reconstruction loss term and the replaced divergence term from the KL divergence, i.e., the MMD, should be selected to balance the relative scale of each element.

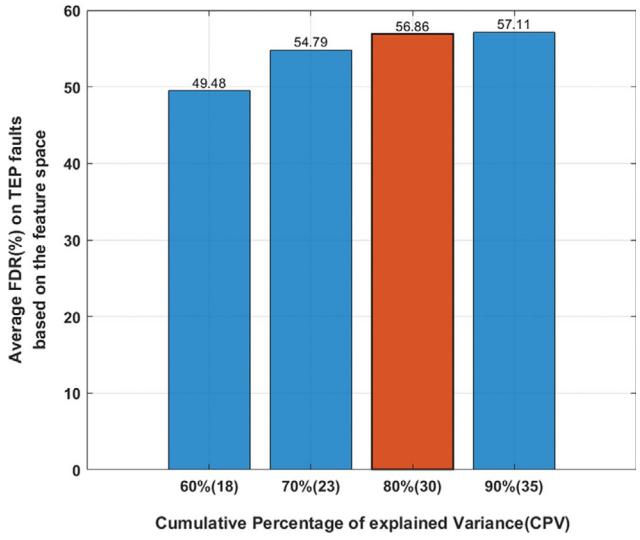

Fig. 6. Case study to decide the reduced dimension of the latent space (The dimensionality of the latent space in each case is represented in parentheses.).

Table 4. Hyperparameters of Info-VAE

Variable name	Value
MMD weight	50
Mini batch size	256
Optimizer	RMSProp
Learning rate	0.001

After the training process was finished, the Info-VAE model was used to augment the original training dataset with artificial samples. The model generates samples that can help the manifold learning of AE by emphasizing the boundary of the training data based on the latent distribution. To selectively specify the boundary of the normal samples distributed by a multivariate Gaussian, a ring-shape transformation is applied that can be easily extended to a shell of a sphere or a hypersphere within a higher space. First, random samples are extracted from the prior distribution, which has a multivariate normal distribution in a typical VAE having the same dimensionality as the feature space. A specific mapping of the samples from the Gaussian to ring-shape distribution is then applied to rearrange the sample vectors based on the latent space, such that the sample vectors suggest the meaning of the boundary region based on the original dimensional space. The mapping to the boundary is defined as follows:

$$R(z) = \frac{z}{a} + b \cdot \frac{z}{|z|}, \tag{9}$$

where a and b are responsible for the scatteredness and radius of the resulting ring, respectively.

The results of the case studies for various sets of parameters a and b based on two-dimensional Gaussian data are shown in Fig. 7. By using this mapping from the randomly sampled points from a normal distribution, as shown in Fig. 7(a), we can exclusively

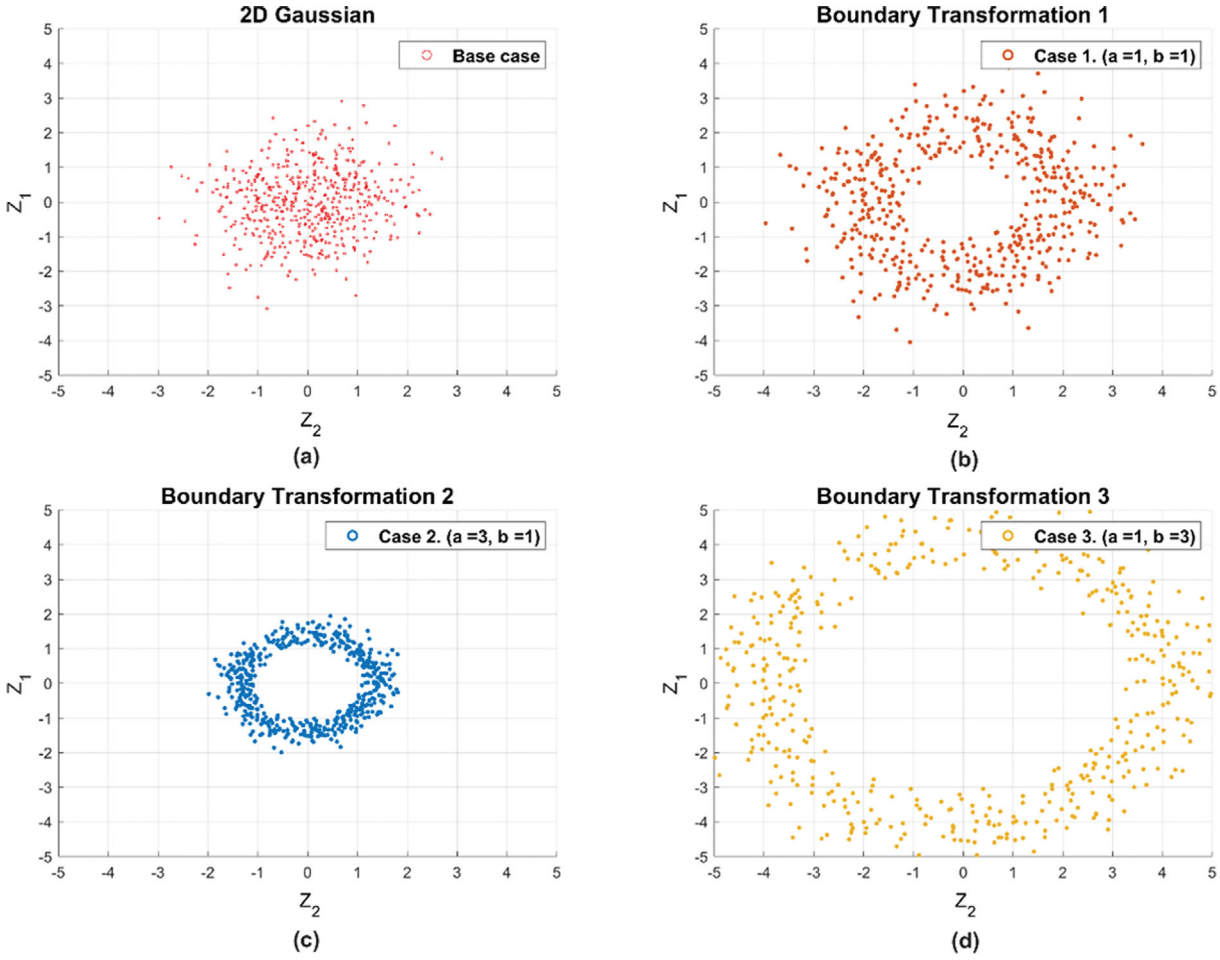

Fig. 7. Case study for various parameters of boundary transformation in 2D Gaussian data.

select the input vectors representing boundaries in the latent space, which is based on the notion that the latent space contains the inherent features of the original data. By adjusting parameters a and b in Eq. (9) to point to the objective region corresponding to the boundaries of the normal state based on the two-dimensional feature space, the desired area in the feature space can be specified as shown in Fig. 7(b), (c), and (d) depending on the purpose. Although the case study is only demonstrated for two-dimensional data, it can also be expanded into higher-dimensional data without a loss of generality.

The generated samples can be classified as distinct groups representing different regions of the normal samples to adjust the number of the different augmentation groups by manipulating the scatteredness and radius through a and b, respectively. To flexibly control the number of augmented datasets with different characteristics, a strategy that divides boundaries into several specific groups and then merges a different number of samples for each group for augmenting into the original dataset was used in this study.

2-2. Data Augmentation

The detailed methodology for augmenting the synthetic data is based on a case study of TEP. Although the dimension of the feature space is beyond the visualizable limit, the main idea of the proposed method for data augmentation can be conceptually explained in a two-dimensional space. The candidate groups for augmentation were divided into five groups, as shown in Fig. 8. The groups were chosen to be able to thoroughly cover the areas that were originally described by the prior distribution while not overlapping each other. Each group can be distinguished based on its distance from the mean.

The groups of infrequent samples that exist far from the mean have a higher weight among the augmented data to supplement the deficient information in the original data. The sample vectors near the center, such as G1, G2, and G3, as well as the outer groups such as G4 and G5 representing the boundary, are also included in the samples to generate artificial data for augmentation to avoid a data imbalance problem owing to an excessive supplementation of the boundary data indiscriminately. Instead, relatively high weights are assigned to the outer groups to emphasize the meaning of the augmentation of the boundary samples that correspond to rare normal samples. The parameters of the boundary transformation and the respective amount of data augmentation for each group are presented in Table 5.

The total number of augmented samples was designed to be half of the original training data that can maximize the final monitoring performance through augmentation. The relative numbers

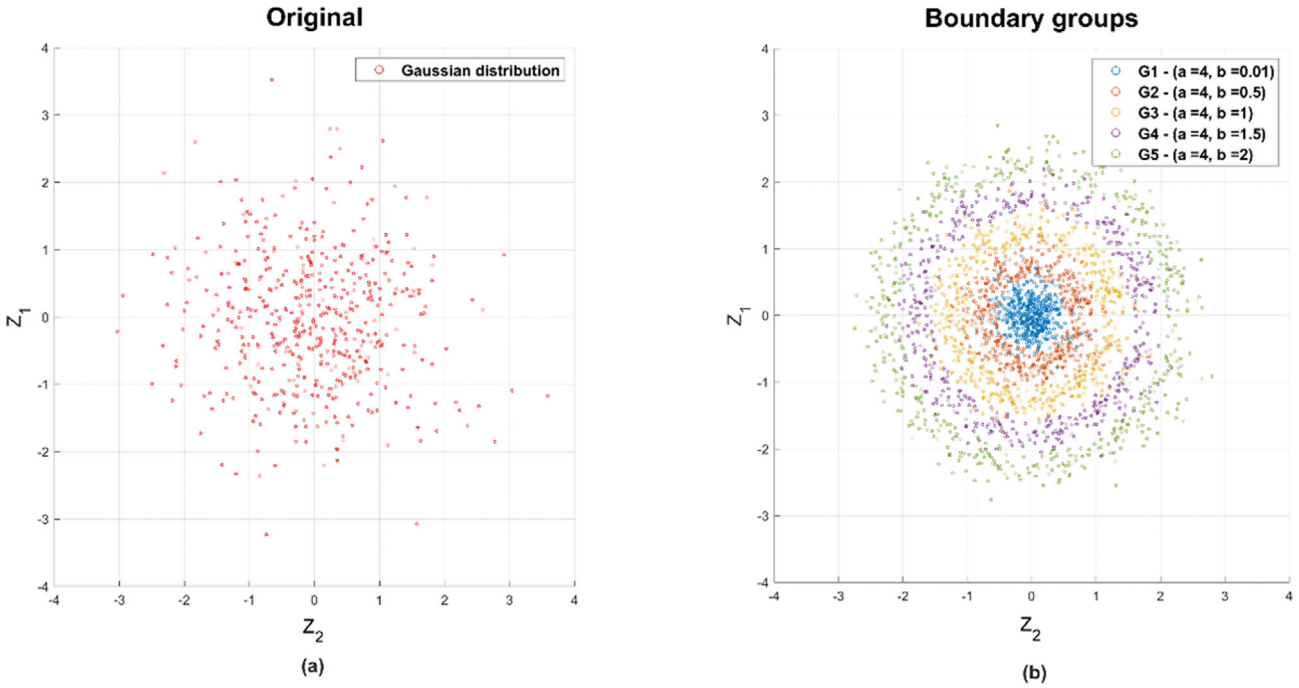

Fig. 8. (a) Sampling from 2D Gaussian distribution (b) Candidate groups of samples transformed by the boundary mapping.

Table 5. Parameters and the number of samples in each group for the TEP case study

Boundary groups	Parameters		Augmented samples
	a	b	
G1	4	0.01	200
G2	4	0.5	400
G3	4	1.0	600
G4	4	1.5	800
G5	4	2.0	1,000

Table 6. Configuration of the dataset for training the AE monitoring model

		Training data	Validation data
Original dataset		6,000	1,200
Augmented datasets	G1	200	40
	G2	400	80
	G3	600	120
	G4	800	160
	G5	1,000	200
Total		9,000	1,800

of each group in the augmented samples were set to be linearly proportional from the center to the outside. Hyperparameters such as the relative scale of the augmentation compared to the original data, the importance among the various groups, and the number of different groups summarized in Table 5 are adjustable depending on the applications.

2-3. Fault Detection Modeling Using AE

After the augmented training dataset for the construction of the fault detection system using AE was prepared, the training of the normal state for the process fault detection system was performed by defining the normal manifold to be used as a monitoring model. The 6,000 samples for the training data out of the total 7,200 samples of the original data from the TEP simulation model were set apart from the validation data after a random shuffling process, which is in accord with the structure of the AE assuming each sample as being independent. The validation data were used to determine the termination point of the training to prevent an overfitting using the early stopping criteria. Both the training and validation data in the original dataset were the same as those used in the modeling of Info-VAE, so standardization was applied as a scaling process. Because the synthetic data obtained from the generative model are scaled, the data for augmentation are attached to the original training and validation dataset resulting from the generation by Info-VAE. Finally, the detailed configuration of the training and validation datasets after the augmentation of the synthetic data is summarized in Table 6. The relative size of the total training dataset was set to five times that of the validation data, as determined through a case study for various amounts of augmentation.

As the dimensions of the inner part, the number of nodes of the feature layer was set to 30, which is the same as in the case of generative modeling using Info-VAE in Section 3. 2. 1. However, the intermediate structure of the monitoring system using AE can be set differently from that of the generative model, where the capacity of the model is limited owing to the lack of the original training data. To make full use of AE for the monitoring system, the number of hidden layers and the size of each layer can be adjusted according to the application. A case study to tune the hyperparameters, such as the number of hidden layers and nodes of the AE monitoring system, determined the final structure, as shown

Table 7. Structure of the monitoring system using AE

Layer	Dimension	Activation	Remarks
Input	50	-	
Encoder 1	46	ReLU	Alpha: 0.2; Kernel_regularizer: L2(0.2)
Encoder 2	42	ReLU	Alpha: 0.2
Encoder 3	38	ReLU	Alpha: 0.2
Encoder 4	34	ReLU	Alpha: 0.2
Feature	30	Linear	
Decoder 1	34	ReLU	Alpha: 0.2; Kernel_regularizer: L2(0.2)
Decoder 2	38	ReLU	Alpha: 0.2
Decoder 3	42	ReLU	Alpha: 0.2
Decoder 4	46	ReLU	Alpha: 0.2
Output	50	Linear	

Table 8. Hyperparameters for the training of AE

Variable name		Value
Mini batch size		256
Loss		MSE
Optimizer		Adam
Learning rate		0.001
Early stopping	min_delta	$5*10^{-4}$
	patience	320
	mode	min

Table 9. Settings for KDE and the control limits for each case

		Base case	Proposed case
Kernel type		Gaussian	Gaussian
Bandwidth (20-fold cross-validation)	H^2	6.158	2.335
	SPE	1.438	1.128
Control limits	H^2_α	57.85	63.75
	SPE_α	31.40	32.25

in Table 7. All layers employed a fully connected layer, and the weights in all cases were initialized using a truncated normal distribution. The nonlinear activation functions of the AE monitoring model used to cope with the nonlinearity of the chemical process data were set to a rectified linear unit (ReLU) with the same hyperparameters. The nonlinear activations for the output layers for the encoder and decoder of the AE monitoring model, which correspond to the feature and reconstruction layers, respectively, were not applied to leave them as linear units following the convention used in regression problems. Kernel regularization was adopted in the first layers of the encoding and decoding networks to control the weight parameters from being excessively large by penalizing them.

The additional hyperparameters used to set up the training conditions are listed in Table 8. The loss to be minimized during the training process is set by the mean squared error (MSE) between the input and its reconstruction at the end of the network. Adam with a default learning rate of 0.001 was applied as the optimizer. For the reproducibility of the monitoring system under the same conditions, early stopping criteria were introduced during the training process. The early stopping criteria are a methodology suggesting the termination of the training process if no improvements more than the minimum changes are made, that is, min_delta in Table 8, during a predefined patience epoch by monitoring the validation loss. To compare the proposed method under the same conditions as the base case, which establishes the monitoring system using only the original training data, the same specifications for the training process are applied to the proposed case, as shown in Table 8.

The configurations of the KDE, which are used to determine the control limit for the monitoring system, are presented in Table 9. Although the original process data follow a Gaussian distribution, the hidden representations and reconstructions used to obtain the monitoring statistics might not follow the same distribution after passing through AE. Hence, KDE is utilized as the general approach to estimate the probability density function of the monitoring statistics, which is the basis of the decision of the control limit in each space. The Gaussian kernel, the most common type of kernel, was used to estimate the density of each monitoring statistic. The bandwidths, which are the most significant parameters of KDE influencing the results of the estimation, were selected based on a 20-fold cross-validation to cover all data samples in determining the hyperparameter. Since the control limits are determined based on the models of the base case and the proposed case respectively, they have different values in each case, as shown in Table 9.

To monitor the process fault, two monitoring statistics are defined in the feature space and the original space, similar to that of PCA [36]. Instead of T^2 in the case of PCA, H^2 can be analogously defined based on the hidden representations in the feature space as follows:

$$H^2 = h^T \cdot h,$$
$$h = f_{En^M}(f_{En^{M-1}} \cdots (f_{En^1}(x))), \quad (10)$$

where f_{En^i} represents the i^{th} hidden layer in the encoder network, and M is the number of intermediate layers between the input and feature layers. Similar to the other statistics in PCA, the SPE can be calculated from the reconstruction error between the input and its reconstruction as

$$SPE = e^T \cdot e,$$
$$e = x - g_{De^M}(g_{De^{M-1}} \cdots (g_{De^1}(h))), \quad (11)$$

where g_{De^i} denotes the i^{th} hidden layer in the decoder network. With the proposed method, the two statistics are observed in real time against the process data for fault detection.

Once the training process is completed, the original training data under normal operating conditions are fed into the network. Based on the two statistics, H^2 and SPE, calculated based on the original training data, KDE was applied to predefine the control limits for each monitoring chart [37]. The typical choice for a significance level of α=0.05 was adopted such that the confidence limits in detecting the faulty conditions when the monitoring statistics of the new samples exceed the limits were set to 95%.

CASE STUDY AND DISCUSSION

In this section, the monitoring system based on the proposed method is tested on the TEP fault cases, and the monitoring results are analyzed. To demonstrate the advantage of data augmentation in building a fault detection system, the performance of the proposed method was compared to that of the base case, which only utilizes the original training data in constructing a monitoring system. The simulation was run for a total of 7200 samples with a sampling frequency of 0.01 hr/sample in the Simulink model, which corresponds to 72 hr of plant operation. The simulation data of the faulty condition have the same size as the training data under normal operations, although the process faults are introduced at 1000 simulation time for all cases of faulty conditions.

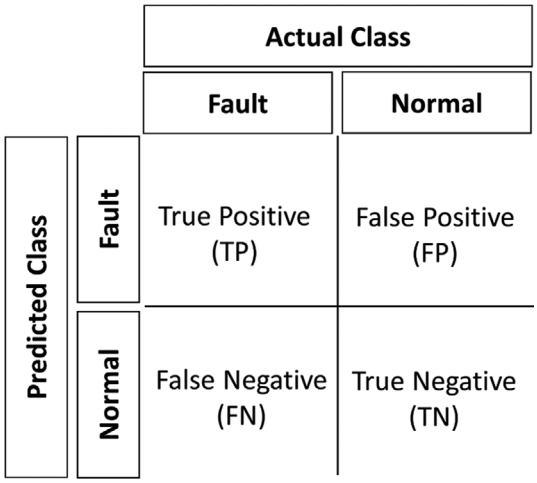

Fig. 9. Binary classification criteria based on the monitoring results of data points.

To compare the performance of the monitoring systems quantitatively, two performance metrics were set up: FDR and FAR [38]. These two metrics were defined based on the results of the binary classification test. The monitoring results of the data points can be classified into four groups, as shown in Fig. 9 [39]. FDR and FAR can be calculated based on the number of instances in each group as follows:

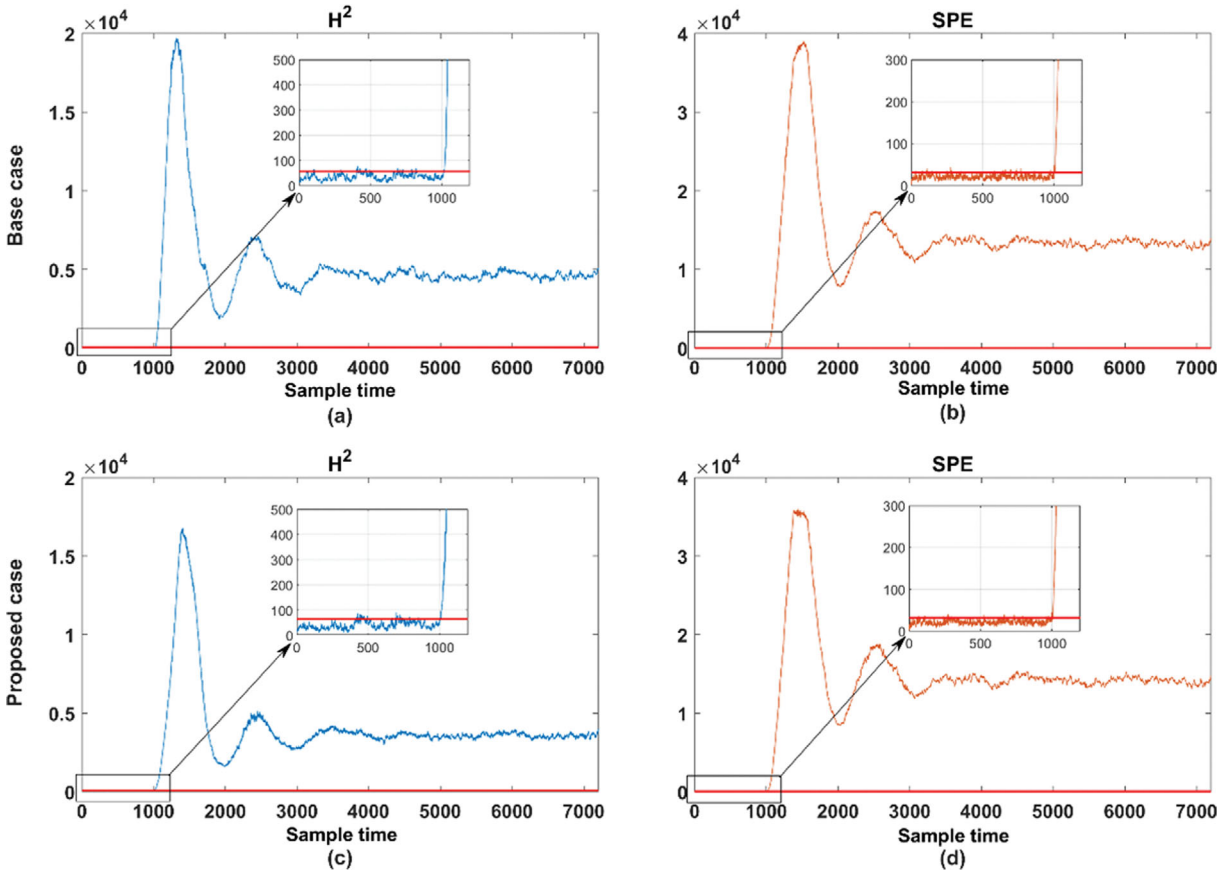

Fig. 10. Monitoring charts of fault 1 for the base case ((a) and (b)) and the proposed case ((c) and (d)).

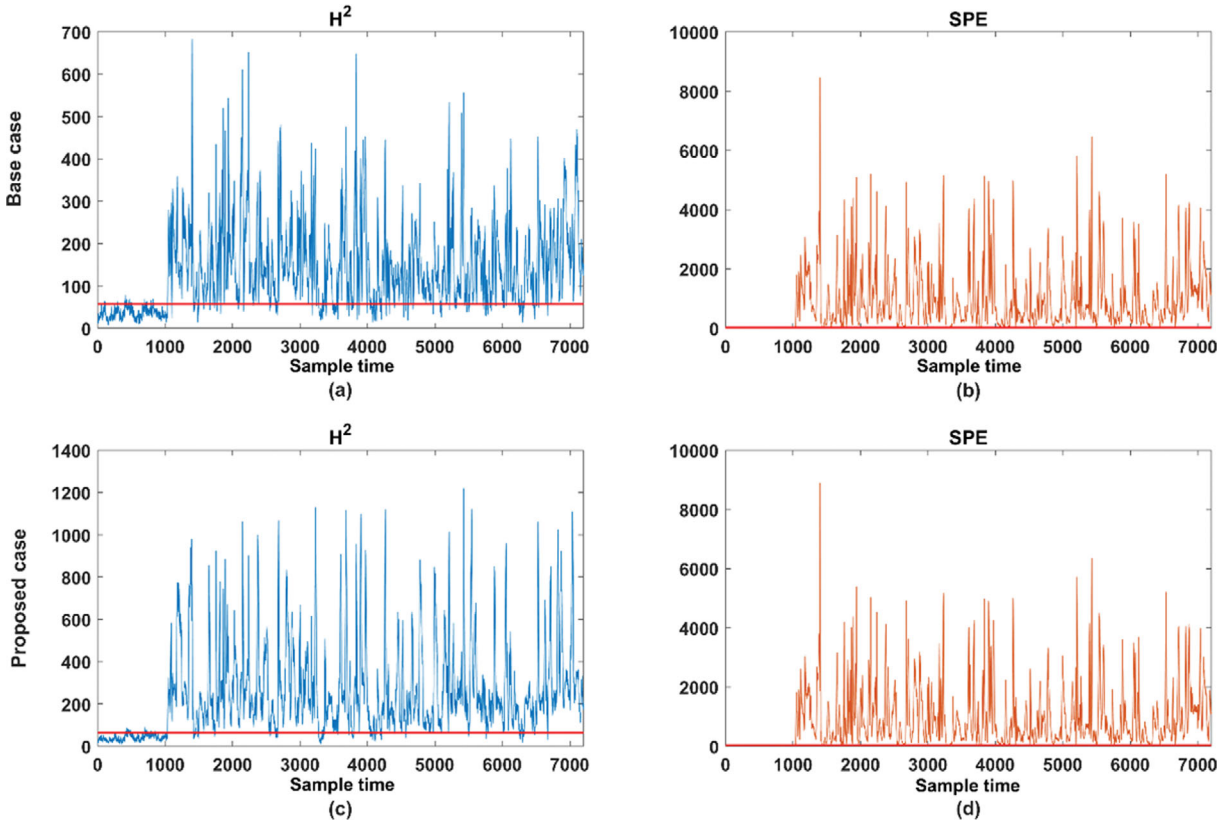

Fig. 11. Monitoring charts of fault 11 for the base case ((a) and (b)) and the proposed case ((c) and (d)).

$$\mathrm{FDR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\quad \mathrm{FAR}=\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}} \tag{12}$$

FDR is the ratio of the samples exceeding the control limit to the entire sample time since the fault has been introduced. Conversely, FAR is the number of samples falsely going beyond the control limit per total number of normal operation samples. FDR needs to be maximized on the abnormal data while keeping FAR for the normal data as low as possible, which is generally determined as 5%. These two metrics should be compared simultaneously because a monitoring system with a high FDR and high FAR under a normal state is undesirable.

For the first case, the monitoring chart of fault 1 of the TEP is shown in Fig. 10. The blue and orange lines represent the monitoring statistics of the test samples in the feature space and residual space, H^2 and SPE, respectively. The red horizontal lines in each monitoring chart are the respective control limits, H^2_α and SPE_α, as determined by KDE in Table 9. For fault 1 in the TEP, both statistics can detect the process fault immediately after the occurrence of the process anomaly, similar to other methodologies used in previous research [3]. In the investigation during the first 1000 sample times before the fault was introduced, it was confirmed that more than 95% of the samples were classified as being in a normal state, distributed within the control limits. Considering the scenario of fault 1, which incurs a step deviation of the feed ratio of streams A and C, it is obvious that the majority of the process variables deviate from their nominal values during normal operation. These results verify that the monitoring statistics in both spaces can properly define a normal manifold and differentiate the faulty process condition from it.

For fault 11, which is the random variation of the reactor cooling water inlet temperature, the monitoring performance of the proposed method does not show a significant improvement in terms of the FDR or FAR compared to the base case. However, based on the results of the monitoring charts in the feature spaces shown in Fig. 11(a) and (c), the proposed method showed a more pronounced isolation with a larger magnitude in the monitoring statistics for the faulty samples compared to the normal operation samples. The false-negative rate, similar to the type II error in the statistical analysis, was reduced from 10.9% to 5.5%. Therefore, data augmentation can improve the monitoring systems. The improvement in the feature space is also noteworthy because it is in the feature space where data augmentation is designed to emphasize the boundary region of the normal space.

The monitoring result of fault 14, which incurs the sticking of the reactor water cooling valve, is shown in Fig. 12. Although more than 7.5% of the monitoring statistics of the base case in Fig. 12(a) improperly stay below the monitoring limit since a fault occurs in the 1000 sample time, the results of the proposed method in Fig. 12(c) exceed the limit for all but 0.15% of the faulty samples. In terms of the fault detection rate, the fault was detected with high accuracy by the proposed method 99.85% of the time, with 92.37% being the base case. This demonstrates the effectiveness of the proposed method, particularly in the feature space. In addition, it can

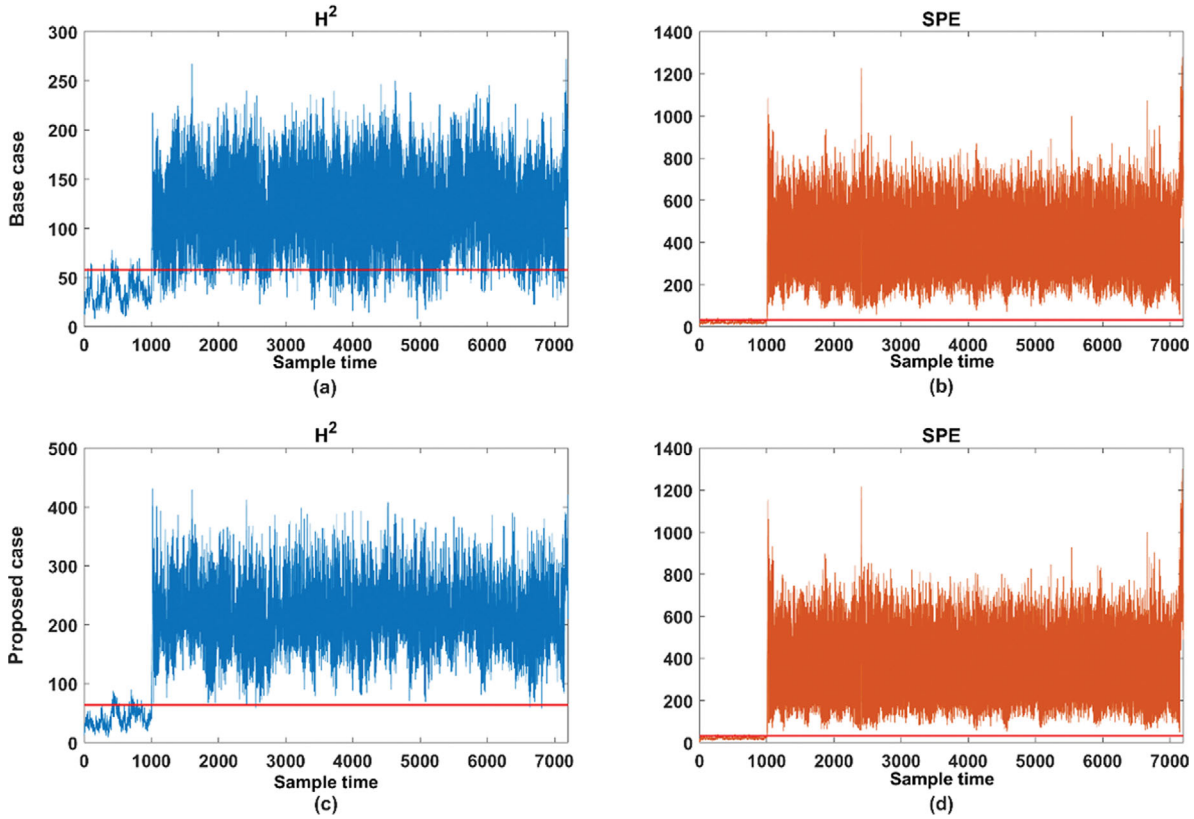

Fig. 12. Monitoring charts of fault 14 for the base case ((a) and (b)) and the proposed case ((c) and (d)).

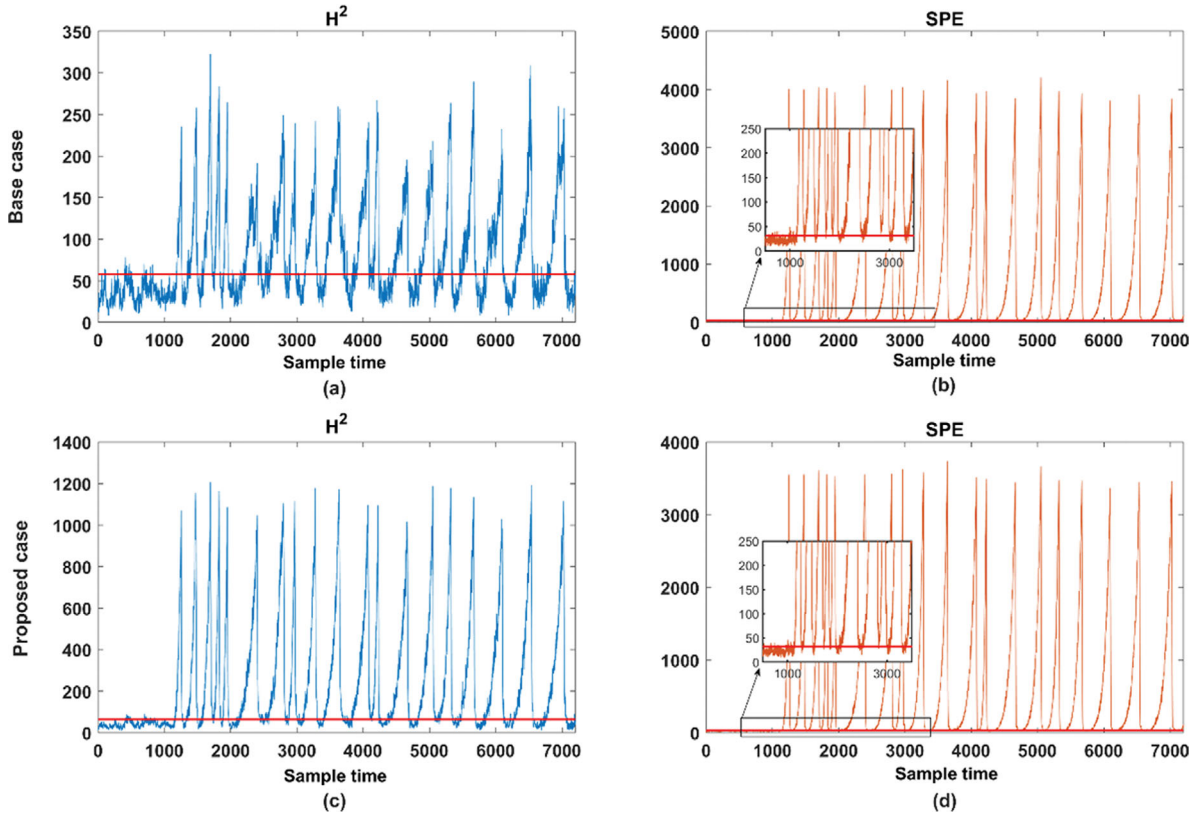

Fig. 13. Monitoring charts of fault 18 for the base case ((a) and (b)) and the proposed case ((c) and (d)).

be confirmed that the proposed method is effective with other types of faults, such as the sticking of a valve as fault 14.

For fault 18, in which the deviation of the heat transfer within the condenser occurs as a random variation type, a similar result can be observed in the monitoring result as shown in Fig. 13. The monitoring charts in both cases have common trends where the fault pushes the state far from the normal condition, followed by the control actions compensating it iteratively. Given the control scheme applied to the TEP model used in this study [34], the trends of the monitoring charts in Fig. 13 are the result of controlling the separator temperature by utilizing the condenser coolant valve. Meanwhile, the proposed method showed a distinct result, minimizing the restoration of the normal state and emphasizing the magnitude of the fault compared to the base case result, as shown in Fig. 13(a) and (c). Considering the monitoring results in the residual space, Fig. 13(b) and (d) shows a better performance than that of the feature space in both cases, and the improvement of the monitoring performance in the feature space from the base case, as shown in Fig. 13(c), can be interpreted as evidence that the data augmentation encourages manifold learning. As another advantage of the proposed method, the monitoring indices exhibit a larger magnitude of deviation in the monitoring statistics, which means that the proposed method can isolate the fault condition better. The fault detection rates for all 28 faults in the TEP are summarized in Table 10. The detection rate in the residual space, SPE, is slightly higher in the base case, but the difference is negligible considering that the base case maintains a relatively higher FAR of 8.58% on the normal operation data than that of the proposed case (6.92%). It is also noteworthy that the proposed method in the feature space outperforms the base case for most situations while maintaining a lower FAR than the base case, which means that it can distinguish between normal and abnormal states more accurately.

Table 10. FDR (%) of the base case and the proposed case for all 28 faults in the TEP model (The value in parentheses corresponds to FAR (%) in each space)

Fault No.	Base case		Proposed case	
	H^2 (4.45)	SPE (8.58)	H^2 (3.50)	SPE (6.92)
1	99.69	99.95	99.90	99.90
2	99.37	99.81	99.37	99.55
3	1.63	25.98	1.53	15.26
4	99.40	99.97	99.97	99.97
5	2.90	26.95	2.79	17.38
6	99.72	99.72	99.72	99.72
7	99.97	99.97	99.97	99.97
8	97.69	98.87	98.00	98.50
9	2.21	32.24	7.08	20.79
10	62.34	94.44	75.78	93.61
11	89.15	98.87	94.48	98.68
12	20.35	65.34	39.64	56.51
13	98.15	99.47	99.29	99.40
14	92.37	99.97	99.85	99.95
15	2.06	22.21	1.32	14.47
16	1.63	19.29	0.84	12.56
17	91.69	98.69	97.5	98.61
18	57.15	87.60	70.54	85.02
19	92.45	99.45	97.94	99.39
20	92.21	97.87	97.18	97.73
21	4.55	22.98	2.87	14.92
22	4.45	34.19	3.43	21.53
23	3.21	24.77	2.64	15.93
24	75.89	98.21	92.74	98.06
25	36.09	92.90	67.39	89.92
26	64.89	93.78	77.23	92.08
27	50.43	94.05	63.55	92.84
28	3.47	26.93	5.61	18.26

As another performance index for the monitoring system, we can investigate the detection delay, which is the time required to detect a fault for the first time since it occurred. Except for a few hard-to-detect fault cases, such as faults 3, 9, and 15, for which the monitoring system could not effectively identify the process, the detection delay was significantly reduced by the proposed method in some fault cases. In terms of minimizing the loss of profitability due to process faults and securing process safety, the proposed method can inform engineers of faults more rapidly, allowing them to handle such faults as quickly as possible. Fig. 14 shows that the delay was significantly reduced by the proposed method. Fault 10, where a random variation in the temperature of the C feed occurs, is the case with the greatest reduction in the fault detection delay while improving the detection accuracy by more than 10%. The delay in the base case was 351 samples, which corresponds to 210 min considering the sampling frequency of the TEP, whereas the proposed method can cut down on it by 168 samples, thereby reducing the fault detection delay by approximately 100 min. Even if the time when a large fault appears is equivalently assumed in terms of the monitoring statistics in the feature space, the detection delay can be reduced by 51 samples, corresponding to 30 min.

For fault 17, where the heat transfer within the reactor deviates from the nominal condition, the proposed method also shows an improvement in the detection accuracy and a delay reduction. The monitoring charts in the feature space for both cases are shown in Fig. 15. As shown in the enlarged view of the plots, the detection delay in the proposed case was decreased by 21 sample times, which corresponds to approximately 12 min; thus, the monitoring accuracy is also improved by the proposed method.

CONCLUSION

In this study, a monitoring framework was proposed that integrates manifold learning with data augmentation to supplement insufficient information for training. The main idea is to augment the synthetic data into the original training data using a generative model, Info-VAE, to supplement the training data for the construction of the fault detection system using AE. The synthetic data are aimed at representing the region of the boundary of the normal training data, which contain infrequent but informative samples in the manifold learning of the normal state for process monitoring. In addition, a generative model that can manipulate latent sample

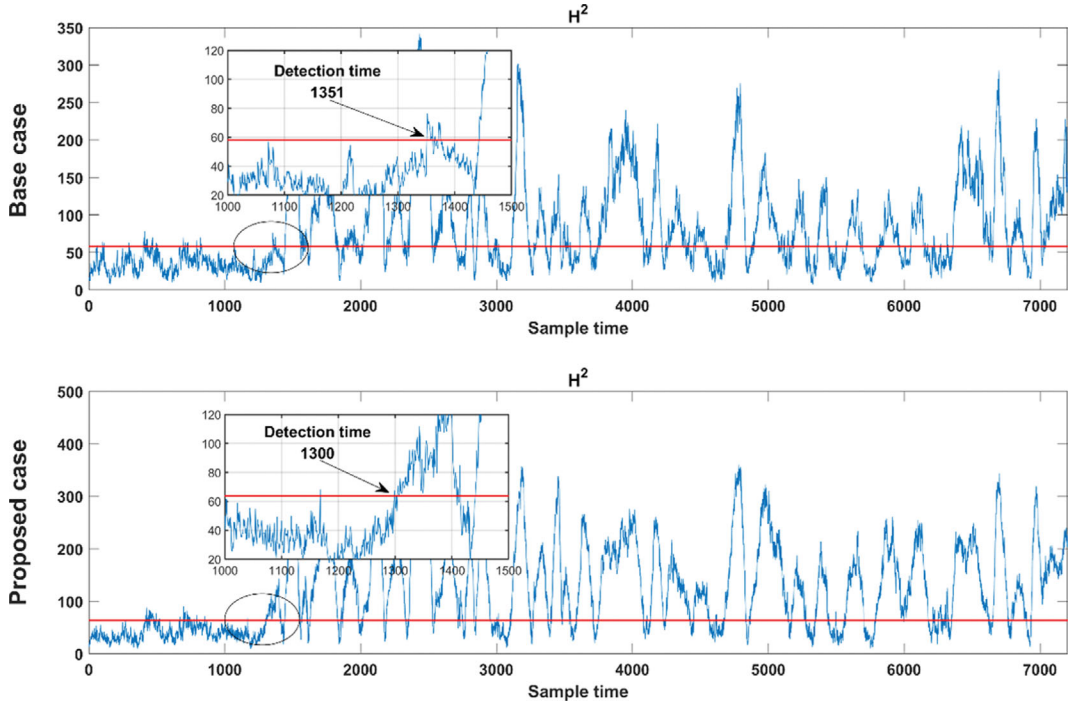

Fig. 14. Comparison of the fault detection delay to first alarm for fault 10 base case: 351 samples (210 min), proposed case: 300 samples (180 min) (A fault is introduced at 1000 samples.)

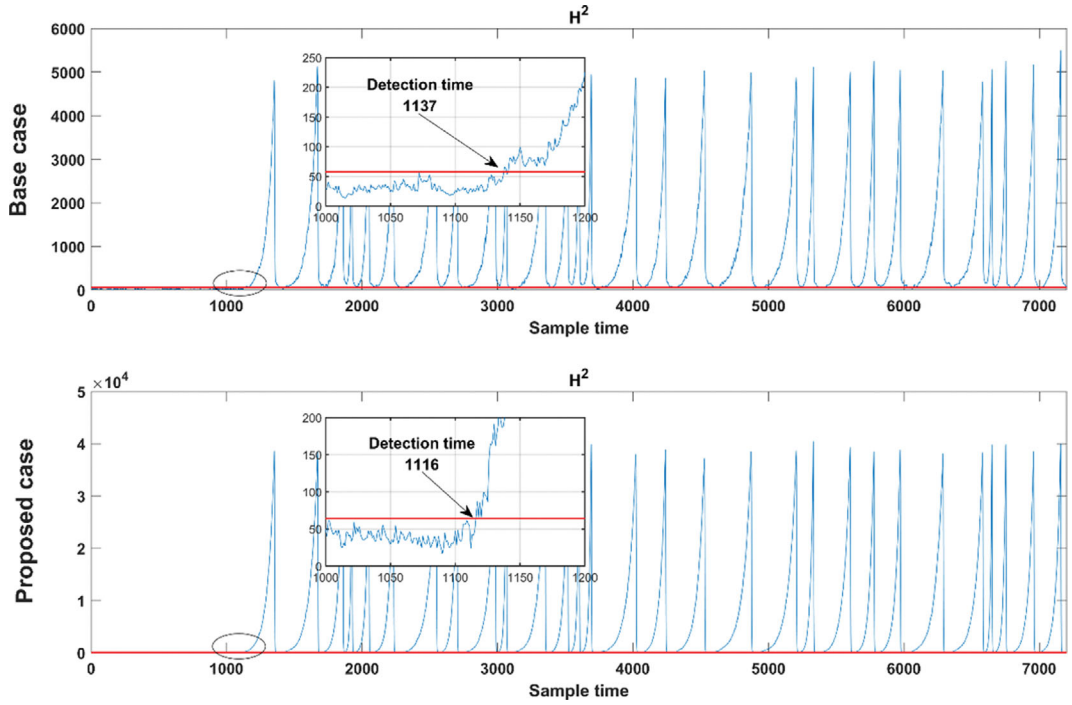

Fig. 15. Comparison of the fault detection delay to first alarm for fault 17 base case: 137 samples (82 min), proposed case: 116 samples (70 min) (A fault is introduced at 1000 samples.).

vectors based on the feature space and decode it through the generative network was utilized rather than the conventional methods using the transformation from manually engineered features; as a result, the proposed method can be applied in various domains.

To demonstrate the effectiveness of the data augmentation for the development of a monitoring system, a case study using the Tennessee Eastman process was carried out. The analysis results showed that the fault detection accuracy was improved for most fault cases

in the feature space in accord with the intention of the data augmentation, and the fault detection delay was also reduced.

However, several issues remain to be resolved for further improvements. Even if the hard-to-detect fault cases are set aside, there exist a few cases that the current monitoring system cannot effectively detect despite the data augmentation. The process dynamics, which also includes information about the process state, were not considered in this study employing a basic AE, which assumes independence between the data samples. Although a recurrent neural network (RNN) structure, such as long short-term memory and a gated recurrent unit, can consider time-series information more accurately than a conventional AE, which only utilizes the current information, it requires more weight parameters and a larger amount of training data as the dimension of the network is expanded along the time axis. Therefore, if sufficient training data are provided, the RNN structure can be combined to address such limitations in future studies.

In addition, the generative model can be further investigated to improve the fidelity of the synthetic sample for data augmentation. As the hybrid of VAE and GAN, adversarial autoencoder (AAE) [40] was proposed, which replaces the KL divergence penalizing the encoding distribution to fit the prior distribution with the discriminative network. By the modification, the assumption that the encoding posterior $q_\phi(z|x)$ should be multivariate Gaussian is no longer constrained, thus allowing the arbitrary distribution for the latent vector z. As AAE retains the structure of VAE that can fit the data distribution in the latent space to a certain distribution, selective sampling and generation such as the boundary region of the data distribution can be achieved. Thus, the proposed methodology in this study can be used in various domains by alleviating the restrictive assumption of the generative model, VAE.

ACKNOWLEDGEMENT

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2016R1A5A1009592).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

1. V. Venkatasubramanian, R. Rengaswamy, K. Yin and S. N. Kavuri, *Comput. Chem. Eng.*, **27**(3), 293 (2003).
2. J. Lee, C. Yoo, S. Wook, P. A. Vanrolleghem and I. Lee, **59**, 223 (2004).
3. W. Yan, P. Guo and Z. Li, *Chemom. Intell. Lab. Syst.*, **158**, 31 (2016).
4. A. J. Holden, *Science*, **313**, 504 (2006).
5. F. Lv, C. Wen, Z. Bao and M. Liu, *2016 Am. Control Conf.*, **2**, 6851 (2016).
6. J. Fan and W. Wang, *IEEE*, 1001 (2017).
7. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P. A. Manzagol, *J. Mach. Learn. Res.*, **11**, 3371 (2010).
8. S. Rifai, P. Vincent, X. Muller, X. Glorot and Y. Bengio, *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*, **1**, 833 (2011).
9. L. Jiang, Z. Song, Z. Ge and J. Chen, *Ind. Eng. Chem. Res.*, **56**, 26 (2017).
10. S. Heo and J. H. Lee, *Processes*, **7**, 7 (2019).
11. Z. Zhang, T. Jiang, S. Li and Y. Yang, *J. Process Control*, **64**, 49 (2018).
12. W. Yu and C. Zhao, *IEEE Trans. Control Syst. Technol.*, 1 (2019).
13. H. Zhao, *Chemom. Intell. Lab. Syst.*, **176**, 11 (2018).
14. P. Y. Simard, D. Steinkraus and J. C. Platt, *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, 958 (2003).
15. H. Han, W. Y. Wang and B. H. Mao, in *Lecture Notes in Computer Science*, **3644**, 878 (2005).
16. S. C. Wong and M. D. Mcdonnell, *2016 Int. Conf. Digit. Image Comput. Tech. Appl.*, 1 (2016).
17. T. Devries and G. W. Taylor, *arXiv Prepr. arXiv1702.05538*, 1 (2017).
18. Z. Wan, Y. Zhang and H. He, *IEEE*, 1 (2017).
19. N. Etworks, A. Storkey and H. Edwards, *arXiv preprint arXiv:1711.04340*, 1 (2017).
20. J. Jorge, R. Paredes, J. A. Sanchez and M. Bened, *VISIGRAPP (5: VISAPP)*, 96 (2018).
21. W. N. Hsu, Y. Zhang and J. Glass, *IEEE Autom. Speech Recognit. Underst. Work.*, **1**, 16 (2017).
22. X. Gao, F. Deng and X. Yue, *Neurocomputing*, **396**, 487 (2019).
23. S. K. Lim, Y. Loo, N. Tran, N. Cheung, G. Roig and Y. Elovici, *2018 IEEE Int. Conf. Data Min.*, 1122 (2018).
24. C. Mellon and U. C. Berkeley, *arXiv preprint arXiv:1606.05908*, 1 (2016).
25. A. Krogh and J. A. Hertz, *Adv. Neural Inf. Process. Syst.* (1992).
26. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *J. Mach. Learn. Res.*, **15**(1), 1929 (2014).
27. S. Ioffe and C. Szegedy, *Int. Conf. Mach. Learn.* (2015).
28. S. Han, J. Pool, J. Tran and W. J. Dally, arXiv preprint arXiv:1506.02626 (2015).
29. S. Heo and J. H. Lee, *Comput. Chem. Eng.*, **127**, 1 (2019).
30. P. Baldi and K. Hornik, *Neural Networks*, **2**(1), 53 (1989).
31. S. Zhao, J. Song and S. Ermon, *Proc. AAAI Conf. Artif. Intell.*, **33**(1), 5885 (2019).
32. J. J. Downs and E. C. Company, *Comput. Chem. Eng.*, **17**, 3 (1993).
33. A. Bathelt, N. L. Ricker and M. Jelali, *IFAC-PapersOnLine*, **48**(8), 309 (2014).
34. N. L. Ricker, *J. Process Control*, **6**(4), 205 (1996).
35. H. Lee, C. Kim, S. Lim and J. Min, *Comput. Chem. Eng.*, **142**, 107064 (2020).
36. V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri and K. Yin, *Comput. Chem. Eng.*, **27**(3), 327 (2003).
37. R. T. Samuel and Y. Cao, *Syst. Sci. Control Eng.*, **4**(1), 165 (2016).
38. D. L. Olson and D. Delen, *Advanced data mining techniques*, Springer, December (2013).
39. C. Kim, H. Lee, K. Kim, Y. Lee and W. B. Lee, *Ind. Eng. Chem. Res.*, **57**(39), 13144 (2018).
40. A. Makhzani, B. Frey and I. Goodfellow, *arXiv Prepr. arXiv1511.05644* (2014).