Proceedings of the 42nd Chinese Control Conference
July 24-26, 2023, Tianjin, China

Industrial Fault Detection Based on Time-Frequency Distillation Autoencoder

Rongyao Cai[1], Kexin Zhang[1], Yong Liu[†1]
1. Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 311121, P. R. China
E-mail: rycai@zju.edu.cn

Abstract: Data-driven feature extraction is a crucial research area in control loop performance assessment (CLPA). Deep learning is a widely used technique for building feature learning models based on neural networks (NNs). However, most NN-based CLPA methods require a large amount of labeled data and do not fully leverage the potential of frequency features. We propose a novel model called time-frequency distillation autoencoder (TFDAE) to address these limitations. The TFDAE consists of a frequency distillation encoder and a representation extraction decoder. The encoder leverages self-supervised contrastive learning to learn time features that guide the distillation of key frequency information. Additionally, a multi-kernel pooling block is incorporated in the encoder, enabling multi-scale information refinement for time feature extraction. The decoder uses the distilled information to extract informative representations and reconstruct the original input series. Taking valve stiction detection in CLPA as the evaluation task, we developed a stiction detection method based on TFDAE. Finally, We evaluate our model on the benchmark dataset: International Stiction Data Base (ISDB), and the experimental results show that TFDAE outperforms traditional knowledge-based and recent NN-based methods.

Key Words: Control loop performance assessment, self-supervised contrastive learning, feature learning, autoencoder

1 Introduction

Valve stiction detection has always been one of the essences in the assessment of control loop performance in process industries [1, 2]. Strong stiction triggers oscillations and increases transmission energy consumption and the dynamic load of equipment. Extracting discriminant features from time series signals and judging the condition based on the extracted features is the primary solution to the challenge.

Two main kinds of feature extraction categories were established in a few decades: feature engineering (FE) and representation learning (RL) [3]. FE extracts features from industrial time series information through expert experiences, or mathematical analysis [4, 5]. FE has advantages in interpretability and reliability, and it has been proven to be an effective feature extraction method in practical applications. However, with the complexity of the industrial process going on, the complicated process exacerbates the difficulty and hinders the FE method from exerting effectiveness.

Accompanied by the proliferation of research in cyber-factories, digital industrial data's scale and types have been larger and various than before for the wide utilization of intelligent sensors in plants [6]. This provides a great opportunity to construct a data-driven method that is capable of automatically learning features from industrial time series. RL, accompanied by the data-driven method based on deep learning rising, has shown huge potential in industrial processes. [7]. Compared with FE, RL is more flexible and free-manual. It can extract features automatically from complex industrial data using deep neural networks (DNNs).

However, the traditional DNNs need massive labeled data for supervised training, and the demand for labeled data is increasing. In practice, obtaining a large amount of high-quality labeled data requires professional judgment, which is time-consuming and expensive. Self-supervised learning (SSL) can significantly reduce the dependence on labeled data with a proxy task, which is an auxiliary task that generates the pseudo-labels [8]. The proxy task aims to learn the representations revealing the general information of data in the latent space. The extracted representations can be used as the input of the target task, which allows the target task to require only a small number of labeled samples for training.

Fourier transform is a common method to filter noise from time series. Moreover, it is now used as an auxiliary augmentation technique for SSL to create positive samples from the frequency perspective [9, 10]. However, traditional methods ignore the advantage of fast Fourier transform (FFT) in noise splitting and composition analysis. In this paper, we use FFT to extract the seasonal and trend features from the original time series.

Inspired by the above works, we present the novel time-frequency distillation autoencoder (TFDAE) to automatic learning representations from industrial time series. TFDAE includes a frequency distillation encoder and a representation extraction decoder. The frequency distillation encoder aims to select valuable frequency information utilizing contrastive learning. The representation extraction decoder handles the processed series to extract the target representation and reconstructs the original series. The main contribution of this work are summarized as follows:

- A novel framework named TFDAE is proposed focusing on the representation extraction task from complicated industrial processes. TFDAE consists of a frequency distillation encoder and a representation extraction decoder.
- The frequency distillation mechanism is developed to refine practical frequency information with contrastive learning in the encoder. Contrastive learning creates the consistent time feature, which distills information through frequency attention.
- Multi-kernel pooling block is introduced to refine various scale information, which helps extract the time feature containing seasonal and trend information.

This work was supported by the National Key R&D Program of China under Grant 2021YFB2012300.

- Based on TFDAE, we build a fault detection method for the valve stiction recognition task. The proposed method was evaluated on the benchmark dataset. The experimental results show that our method outperforms the traditional expert-based and recent data-driven-based methods.

The rest of this paper is organized as follows. Section 2 introduces the basic blocks and TFDAE. Section 3 provides the experimental results on the benchmark data. Finally, the conclusion summarizes this article in Section 4.

2 Methodology

2.1 Basic Blocks

Convolutional layers:A convolutional layer takes an input tensor X and applies a set of filters, each with its own set of weights, to the input. Each filter slides over the input tensor, performing a dot product at each location and producing a single value in the output tensor. A convolutional layer can be expressed as:

$$Z^p = \sum_{i=0}^{c} W^p * X_i + b^p, p = 0, ..., n, \tag{1}$$

where Z^p is the p-th channel's output; p is convolution kernel index; W^p is the weight; X is the input data and i is the input channel index; $*$ is the convolution operator.

In order to increase the overall network's nonlinearity, an activation function such as the rectified linear unit (ReLU) is commonly applied after the convolutional layer. The ReLU function sets any negative values in the feature map to zero, allowing the network to learn nonlinear relationships in the data. It is defined as:

$$ReLU(Z^p(i,j)) = max(0, Z^p(i,j)). \tag{2}$$

Average-Pooling layers: To capture multi-scale features in our framework, we employ pooling layers with varying kernel sizes to process feature maps. Specifically, we leverage this approach to extract seasonal patterns in time series data. We choose the average-pooling layer and which is denoted as:

$$Z^p = \frac{1}{n} \sum_{i=0}^{c} \mathbb{1} * X_i, \tag{3}$$

where $\mathbb{1}$ is the sum vector with dimension n.

Linear layers: Linear layers are usually used between blocks to change the dimension or length of the feature maps from preceding blocks. It can be expressed as:

$$Z = WX + b, \tag{4}$$

W is the weight matrix; X is the input vector; b represents the bias and Z is output vector.

Attention mechanism: The attention mechanism [11] uses limited attention resources, analyzing the relationship between the input sequence elements in parallel and focusing on valid information from a global perspective. The widely used method is the self-attention mechanism, which is described as follows:

$$Q_{n\times k} = X^Q_{n\times m} W^Q_{m\times k}, \tag{5}$$

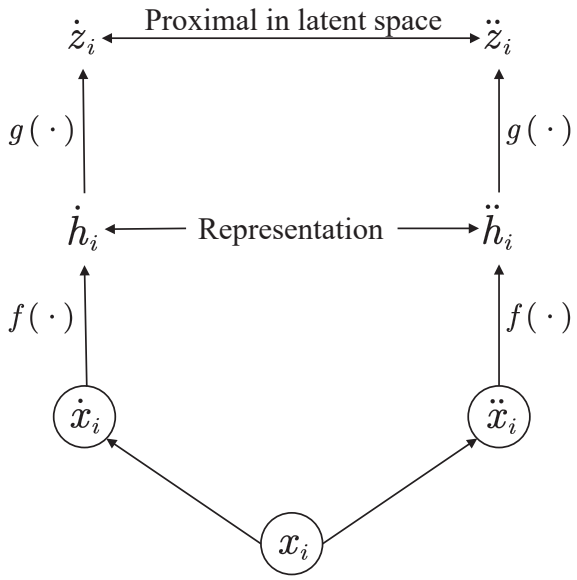

Fig. 1: Architecture of contrastive learning. $\dot{x}_i$ and $\ddot{x}_i$ are the augmented positive pair of original sample x_i, $\dot{h}_i$ and $\ddot{h}_i$ are the representations after encoder $f(\cdot)$. Mapping function $g(\cdot)$ acquires the vectors $\dot{z}_i$ and $\ddot{z}_i$, which have max similarity in latent space.

$$K_{n\times k} = X^K_{n\times m} W^K_{m\times k}, \tag{6}$$

$$V_{n\times v} = X^V_{n\times m} W^V_{m\times v}, \tag{7}$$

$$Softmax(x_i) = \frac{\exp x_i}{\sum_{j=0}^{n} \exp x_j}, \tag{8}$$

$$Attention_{n\times v} = Softmax(\frac{QK^T}{\sqrt{k}})V, \tag{9}$$

where X is the input vector, Q, K, V are the query, key, value matrix, respectively; subscript n, k, v are the dimensions; W^Q, W^K, W^V are transformation matrices; $Softmax$ is a nonlinear normalization function and $Attention$ is the target output.

2.2 Contrastive Learning

Contrastive learning is a vital branch of self-supervised learning. Suppose we have two time features $\dot{h}^T_i$ and $\ddot{h}^T_i$ from one sample x but augmented by different methods, respectively. Based on the consistent constraint, the time features derived from the same time sample are consistent even if augmented. In contrast, the time features derived from different samples are inconsistent. Especially the goal of contrastive learning is to extract the available feature and maximize the similarity between $\dot{h}^T_i$ and $\ddot{h}^T_i$, meanwhile maximizing the distance between $\dot{h}^T_i$ and $\dot{h}^T_j$ in latent space. In this paper, we use $cos < \dot{h}^T_i, \dot{h}^T_j >$ to quantum the similarity between $\dot{h}^T_i$ and $\dot{h}^T_j$.

$$sim(\dot{h}^T_i, \dot{h}^T_j) = \frac{\dot{h}^{T\top}_i \dot{h}^T_i}{|\dot{h}^T_i| \cdot |\dot{h}^T_j|}. \tag{10}$$

Fig. 1 reveals the basic structure of contrastive learning. Starting with N original samples x_i, the augmentation adds different noise into the x to create positive pair $2N$ $\tilde{x}_j$ and $\tilde{x}_j$ and the $2N-2$ augmented items from other samples are

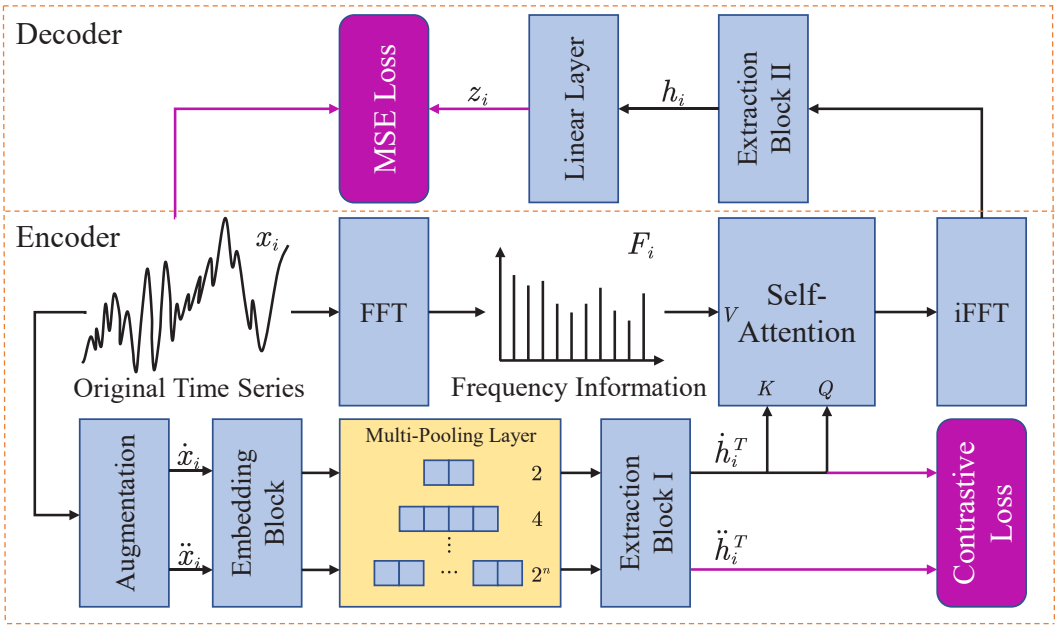

Fig. 2: Architecture of our framework. TFDAE aims to obtain the general representation h_i from the original time series x_i. To achieve this, an augmentation process is applied to the original time series x_i, creating positive pairs $\dot{x}_i$ and $\ddot{x}_i$. These pairs are transformed into time features $\dot{h}_i^T$ and $\ddot{h}_i^T$ using the Embedding Block, Average Pooling Layer, and Encoder Block. The time feature $\dot{h}_i^T$ is then used to induce the frequency information F_i, and the self-attention results from the frequency domain are transformed into the time domain using iFFT. The resulting h_i is the desired general representation, and z_i is the reconstructed time series used to approximate x_i.

considered as negative values for each positive pair. Then, encoder $f(\cdot)$ calculates the latent representation of positive pair $\tilde{x}_i$ and $\tilde{x}_j$ and $g(\cdot)$ as temporary proxy task block output the results. The similarity to the negative values is calculated using NCE loss as follows:

$$L_i = -\log \frac{\exp(sim(\dot{h}_i^T, \dot{h}_i^T)}{\sum_{k=0}^{2N} \mathbb{1}_{k \neq i} \exp(sim(\dot{h}_i^T \dot{h}_k^T)/\tau)}, \tag{11}$$

$$L_{NCE} = \frac{1}{N} \sum_{i=1}^{N} L_i, \tag{12}$$

where τ is the temperature hyperparameter, N is the number of samples in a batch.

Following training based on the NCE loss, the proxy task block is discarded, and only the encoder $f(\cdot)$ is utilized to create a latent representation for the target task. Then a linear or other layer is trained to establish the relationship between the representation obtained from the pre-trained $f(\cdot)$ and the target task, such as fault detection.

2.3 Framework Architecture

Our strategy involves using the dataset D^{pre} to pretrain the model's parameters, followed by acquiring the general representation h_i from x_i. This representation is then used to reconstruct the input for supervised few-shot training of the downstream task on D^{tune}. This approach reduces the demand for labeled data compared to directly supervised training. Moreover, the framework provides a paradigm that utilizes time domain information to guide practical frequency selection. The detailed architecture of the framework is shown in Fig. 2.

2.3.1 Frequency distillation encoder

Let x_i be the original time series and F_i be the frequency signal of x_i. Time feature $\dot{h}_i^T$ and $\ddot{h}_i^T$ are the outputs through Augmentation, Embedding Block, Multi-Pooling Layer, and Extraction Block I. Time feature $\dot{h}_i^T$ and $\ddot{h}_i^T$ are close in the latent space compared with $\dot{h}_j^T$ from other series, and contrastive learning guarantees consistency of them. Further, self-attention (5)-(9) utilizes $\dot{h}_i^T$ to instruct to distill adequate frequency information from F_i. Finally, the chosen frequency information is converted to time series by inverse fast fourier transform (iFFT) for downstream processing.

The Augmentation includes three kinds of operations: noising, flipping and scaling. x_i is transformed to $\dot{x}_i$ and $\ddot{x}_i$, which construct a positive pair.

Our method makes another contribution through the use of a Multi-Pooling Layer. This layer employs kernels with multiple sizes, which allows for the calculation of the mean in various scales. By doing so, it is able to describe both local and global trends, providing an intuitive feature in the time domain. After calculating the mean vectors for each kernel size, they are concatenated into a long vector. Extraction Block I then utilizes this vector to extract seasonal and trend information, ultimately creating the time feature $\dot{h}_i^T$.

$$v_i = \phi(x_i|2^i), \tag{13}$$

$$V = [v_1, v_2, ..., v_n], \tag{14}$$

where v_i is the output of pooling operation ϕ with kernel size 2^i for input x_i and V is the concatenated long vector.

Both the Embedding Block and Extraction Block I are composed of multiple convolutional and linear layers.

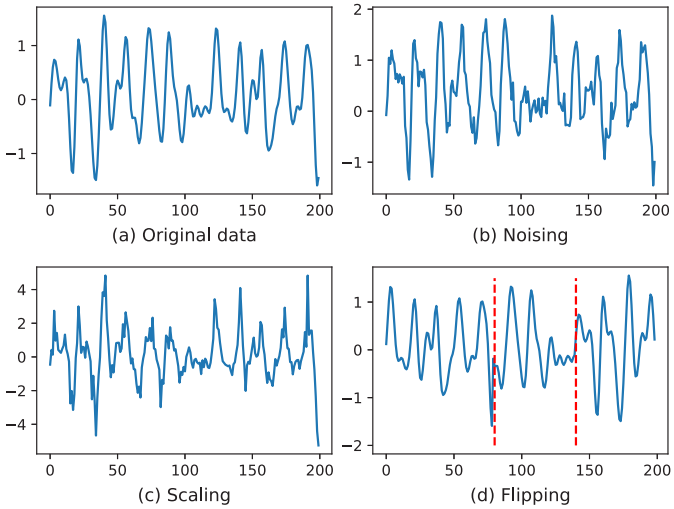

Fig. 3: Data Augmentation.

Table 1: Benchmark Industrial Loops

Loop	Type	Malfunction	Loop	Type	Malfunction
CHEM 1	Fic	Stiction	CHEM 24	Fic	Likely stiction
CHEM 2	Fic	Stiction	CHEM 26	Lev	Likely stiction
CHEM 3	Tem	Quantisation	CHEM 29	Fic	Stiction
CHEM 4	Lev	Tuning problem	CHEM 32	Fic	Likely stiction
CHEM 5	Fic	Stiction	CHEM 33	Fic	Disturbance
CHEM 6	Fic	Stiction	CHEM 34	Fic	disturbance
CHEM 10	Pre	Stiction	CHEM 58	Fic	No oscillation
CHEM 11	Fic	Stiction	MIN 1	Tem	Stiction
CHEM 12	Fic	Stiction	PAP 2	Fic	Stiction
CHEM 13	Ana	Faulty sensor	PAP 4	Con	Tight tuning
CHEM 14	Fic	Faulty sensor	PAP 5	Con	Stiction
CHEM 16	Pre	Interaction	PAP 7	Fic	Disturbance
CHEM 23	Fic	Likely stiction	PAP 9	Tem	No-stiction

2.3.2 Representation extraction decoder

The representation extraction decoder consists of two components: Extraction Block II and Linear Layer. Extraction Block II is responsible for handling the series from self-attention and producing the desired representation h_i, which is subsequently utilized for downstream tasks. The Linear Layer then maps h_i to z_i.

The loss function plays a crucial role in achieving the desired training outcome of the model. Our framework consists of two parts of loss functions. The first part is the NCE loss (12) that distills the time feature from augmented series. The second part is the mean square error (MSE) loss between the reconstructed series z_i and the original series x_i, which verifies whether the target representation h_i can accurately describe x_i. To obtain the total loss, we add the NCE loss and the MSE loss with a loss coefficient α, which controls their relative importance.

$$L_{MSE} = \frac{1}{N}\sum_{i=0}^{N}||x_i - z_i||^2, \tag{15}$$

$$L_{Tol} = \alpha \cdot L_{NCE} + (1-\alpha)\cdot L_{MSE}. \tag{16}$$

After training the model, we use a two-layer perceptron to establish the relationship between z_i and the stiction probability p_i through supervised training with a small labeled dataset.

3 Experiments and Results

3.1 International Stiction Data Base (ISDB)

The valve is the main dynamic equipment in control loop adjusting fluid flow. With the failure or aging of the valve, it is inevitable that the stiction will appear. The valve stiction may cause signal oscillation. For this reason, valve stiction detection has always been one of the research hotspots..

ISDB is a well-known benchmark dataset used to verify the effectiveness of new methods for valve stiction detection tasks. This dataset contains information gathered from various industrial processes, including chemical plants (CHEM), mining (MIN), pulp and paper mills (PAP), power plants (POW), and buildings (BAS). Following similar research in [12], we selected 26 loop information from the ISDB to evaluate our framework and trained the parameters using 55 loops. We divided the data into several samples of a fixed length for each loop and used majority voting to determine whether the loop exhibited stiction.

Table 2: Comparison Results

Method	Accuracy	NOT tested loops
Limit cycle patterns analysis[15]	2/2	24
MV(OP) patterns analysis[16]	12/14	12
MV(OP) patterns analysis[17]	14/16	10
Fuzzy clustering method[18]	2/6	20
Higher-order statistics method[19]	19/24	2
Cross-correlation method[20]	14/24	2
Statistics method[21]	16/25	1
Relay-based method[13]	17/26	0
Curve fitting method[22]	12/25	1
Waveform shape analysis[12]	11/26	0
PSD/ACF method[14]	18/26	0
Peak slope method[23]	14/25	1
Zone Segmentation Method[23]	15/25	1
Our model	19/26	0

3.2 Comparison results

We compared the performance of the proposed TFDAE against thirteen classical stiction detection methods based on expert experiments or mathematical analysis, which were evaluated on the ISDB. The comparison results are presented in Table 2. From Table 2, we observed that only four methods are applicable to all of the test loops, while the others reveal a narrower range of applications. For example, the higher-order statistics method can detect 19 loops, but two loops were not tested. Additionally, the methods proposed in [12–14] can apply to all test loops, but their accuracy is lower than our framework. TFDAE outperforms the classical methods, achieving an accuracy of 19/26, the highest among all the considered methods. This highlights the potential of DL-based models for solving complex industrial problems.

3.3 Hyperparameters study

Hyperparameters must be manually pre-defined and cannot be obtained through model training, making their selection a critical factor in determining the performance of deep learning models. In this study, we examine the impact of two key hyperparameters: loss coefficient α and sampling gap g.

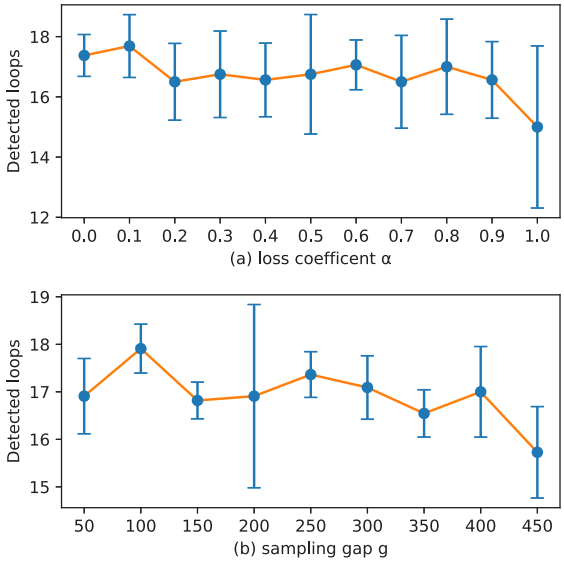

Fig. 4: Detected loops under different α and g. In figure, blue point is the mean value, vertical line is the distribution and orange line is the trend.

Fig. 4 displays the number of detected loops for various values of the loss coefficient α and sampling gap g. The results suggest that NCE loss is significantly improved when $\alpha = 0.1$, as it increases the mean value of detected loops and expands the value's amplitude. The NCE loss function (12) ensures effective model learning and features despite augmentation interference. However, an excessively large value of α can lower the mean value. We speculate that this is due to the training process focusing too much on time feature extraction and neglecting the representation's validity for reconstruction purposes.

The sampling gap g is a crucial hyperparameter that is influenced by the characteristics of the data. A smaller value of g indicates a larger amount of training data, as adjacent samples have long repeating fragments and contain similar information. However, in the contrastive learning process, augmented adjacent samples are labeled as negative, as they are from different original samples. Moreover, adjacent samples contain similar information, which reduces the effective size of the overall sampling dataset and increases the time required for model training. Conversely, a large value of g exceeding the sampling length can discard information from the original data. Therefore, selecting a suitable value of g is critical for training. As shown in Fig. 4, when the sampling length is 400, $g = 100$ yields the best mean accuracy.

4 Conclusion

This article presents a novel approach for extracting informative representations from complex industrial time series using a TFDAE framework. The proposed framework leverages a frequency distillation mechanism guided by time features to refine key frequency information, followed by representation extraction using a decoder. One of the major advantages of our approach is that it requires minimal dependency on labeled data, unlike traditional supervised learning frameworks. To demonstrate the effectiveness of the proposed approach, a fault detection model is built and evaluated on the benchmark ISDB for valve stiction detection. In future work, we plan to further explore and explain the mechanism behind the representation extraction process. The proposed framework has the potential to be applied in practical industrial applications for performance assessment and fault detection.

References

[1] R. B. di Capaci and C. Scali, "Review and comparison of techniques of analysis of valve stiction: From modeling to smart diagnosis," *Chemical Engineering Research and Design*, vol. 130, pp. 230–265, 2018.

[2] M. Jelali, B. Huang *et al.*, *Detection and diagnosis of stiction in control loops: state of the art and advanced methods*. Springer, 2010.

[3] K. Zhang, Y. Liu, Y. Gu, X. Ruan, and J. Wang, "Multiple-timescale feature learning strategy for valve stiction detection based on convolutional neural network," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1478–1488, 2022.

[4] Q. P. He and J. Wang, "Valve Stiction Quantification Method Based on a Semiphysical Valve Stiction Model," *Industrial & Engineering Chemistry Research*, vol. 53, no. 30, pp. 12 010–12 022, 2014. [Online]. Available: https://doi.org/10.1021/ie501069n

[5] R. Srinivasan, R. Rengaswamy, S. Narasimhan, and R. Miller, "Control Loop Performance Assessment. 2. Hammerstein Model Approach for Stiction Diagnosis," *Industrial & Engineering Chemistry Research*, vol. 44, no. 17, pp. 6719–6728, 2005. [Online]. Available: https://doi.org/10.1021/ie049026f

[6] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.

[7] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20 590–20 616, 2017.

[8] L. Jing and Y. Tian, "Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey," *arXiv e-prints*, p. arXiv:1902.06162, Feb. 2019.

[9] C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Tfad," *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Oct 2022. [Online]. Available: http://dx.doi.org/10.1145/3511808.3557470

[10] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Aug 2021. [Online]. Available: http://dx.doi.org/10.24963/ijcai.2021/631

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Singhal and T. I. Salsbury, "A simple method for detecting valve stiction in oscillating control loops," *Journal of Process Control*, vol. 15, no. 4, pp. 371–382, 2005.

[13] M. Rossi and C. Scali, "A comparison of techniques for automatic detection of stiction: simulation and application to industrial data," *Journal of Process Control*, vol. 15, no. 5, pp. 505–514, 2005.

[14] S. Karra and M. N. Karim, "Comprehensive methodology for

detection and diagnosis of oscillatory control loops," *Control Engineering Practice*, vol. 17, no. 8, pp. 939–956, 2009.
[15] A. S. Brásio, A. Romanenko, and N. C. Fernandes, "Detection of stiction in level control loops," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 421–426, 2015.
[16] Y. Yamashita, "An automatic method for detection of valve stiction in process control loops," *Control Engineering Practice*, vol. 14, no. 5, pp. 503–510, 2006.
[17] Y. Yoshiyuki, "Diagnosis quantification of control valves," in *2008 SICE Annual Conference*, 2008, pp. 2108–2111.
[18] M. Daneshwar and N. M. Noh, "Detection of stiction in flow control loops based on fuzzy clustering," *Control Engineering Practice*, vol. 39, pp. 23–34, 2015.
[19] M. S. Choudhury, S. L. Shah, and N. F. Thornhill, "Diagnosis of poor control-loop performance using higher-order statistics," *Automatica*, vol. 40, no. 10, pp. 1719–1728, 2004.
[20] A. Horch, "A simple method for detection of stiction in control valves," *Control Engineering Practice*, vol. 7, no. 10, pp. 1221–1231, 1999.
[21] M. Jelali, B. Huang *et al.*, *Detection and diagnosis of stiction in control loops: state of the art and advanced methods*. Springer, 2010.
[22] Q. P. He, J. Wang, M. Pottmann, and S. J. Qin, "A curve fitting method for detecting valve stiction in oscillating control loops," *Industrial & engineering chemistry research*, vol. 46, no. 13, pp. 4549–4560, 2007.
[23] J. W. Dambros, M. Farenzena, and J. O. Trierweiler, "Data-based method to diagnose valve stiction with variable reference signal," *Industrial & Engineering Chemistry Research*, vol. 55, no. 39, pp. 10 316–10 327, 2016.