2021 IEEE 10th Data Driven Control and Learning Systems Conference
May 14-16, 2021, Suzhou, China

Unsupervised Feature Learning with Data Augmentation for Control Valve Stiction Detection

Kexin Zhang[1], Yong Liu[1]
1. Institute of Intelligent Systems and Control, Zhejiang University, Hangzhou 310027, P. R. China

Abstract: This paper proposes an unsupervised feature learning approach on industrial time series data for detection of valve stiction. Considering the commonly existed characteristics of industrial time series signals and the condition that sometimes massive reliable labeled-data are not available, a new time series data transformation and augmentation method is developed. The transformation stage converts the raw time series signals to 2-D matrices and the augmentation stage increases the diversity of the matrices by performing transformation on different timescales. Then a convolutional autoencoder is used to extract the representative features on the augmented data, these new features are taken as the inputs of the traditional clustering algorithms. Unlike the traditional approaches using hand-crafted features or requiring labeled-data, the proposed strategy can automatically learn features on the time series data collected from industrial control loops without supervision. The effectiveness of the proposed approach is evaluated through the International Stiction Data Base (ISDB). Compared with the traditional machine learning methods and deep learning based methods, the experimental results demonstrate that the proposed strategy outperforms the other methods. Besides performance evaluation, we provide a visualization process of feature learning via principal component analysis.

Key Words: Unsupervised feature learning, data augmentation, convolutional autoencoder, control valve stiction

2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS) | 978-1-6654-2423-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/DDCLS52934.2021.9455535

1 Introduction

Stiction detection of a control valve has always been an essential issue in control loop performance assessment and fault diagnosis in the process industry [1, 2]. Strong stiction results in the unexpected oscillations, which increase variability in product quality, accelerate equipment wear and increase energy consumption. The key to a successful detection is to effectively extract the representative features in the industrial time series data collected from the sensors. In recent years, the smart factory has received increased attention, and the industrial data can be collected much easier and faster than ever before. These provide a new opportunity to achieve an automatic stiction detection using the data-driven methods capable of automatically learning features in the massive industrial data.

The traditional methods for stiction detection within a control valve rely heavily on the hand-crafted features and are highly application-specific. The hand-crafted features are extracted from raw industrial time series data based on the specific characteristics and mechanisms [3, 4]. A similar feature extraction strategy was also adopted in [5]. The typical stiction behavior results in a special shape or pattern in the phase plot of process variables [6, 7]. Extracting the specific features from an image encoded with raw time series data seems more intuitive. The methods based on observed features in an image are summarized as shape-based method [8–12]. Although the shape-based methods are more intuitive, extracting features is essentially a manual process rather than an automatic one. Therefore, inevitable limitations exist when applying these methods to the control loops of dynamic changing characteristics.

In recent years, deep learning (DL) provides a promising and effective solution for feature representation due to its powerful feature learning ability. For the task of valve stiction detection, Amiruddin [13] transformed the raw time samples to D values and used an artificial neural network (ANN) to detect stiction. Dambros [14] also employed simple ANN but the inputs of the network are process variable diagrams. Kamaruddin [15] replaced ANN with CNN to learn features on the designed "butterfly" shape images. These approaches achieve the better results than the traditional handcrafted feature based methods.

However, some challenges in the usage of DL in valve stiction detection still remain. On the one hand, the use of simple learning networks on the real industrial data is also susceptible to noise and the unexpected factors. For a more reliable application, the feature extraction on multiple timescales with temporal and spatial characteristics are necessary for advanced pattern recognition approaches. On the other hand, the above successful models need to satisfy a critical condition: the massive reliable labeled data with process information are available. However, this condition is sometimes a highly constraint in practice since obtaining high-quality data is time-consuming and expensive.

In this paper, we propose an unsupervised industrial time series feature learning approach to distinguish the stiction valves from the non-stiction valves. Considering the commonly existed characteristics of industrial time series signals and the condition that sometimes massive reliable labeled-data are not available, a new time series data transformation and augmentation method is developed. The transformation stage converts the raw time series signals to 2-D matrices through calculating the dynamic time warping (DTW) distance between different segments of the raw time series. The augmentation stage increases the diversity the matrices by performing the transformation on different timescales, then the augmented data are taken as the inputs of the feature learning model.

As one of the most representative DL models, convolutional neural network (CNN) has not only achieved a breakthrough improvement on image recognition and classification tasks in the past decade, but also shown promising de-

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61771193 and Key R&D Program Project of Guangdong Province (2019B010120001).

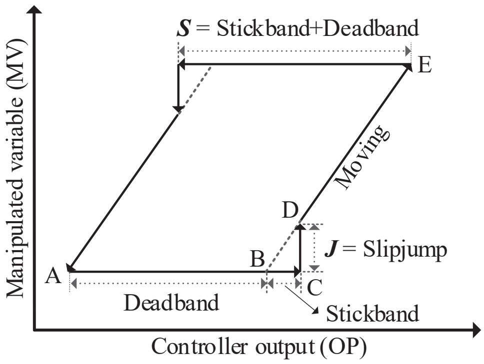

Fig. 1: Typical behavior of valve stiction.

velopment prospects in the feature representation of industrial TS data [16–18]. Moreover, Bai *et al* indicted that a simple convolutional architecture outperforms canonical recurrent neural networks (RNN) which were originally designed for processing sequence data [19]. Hence this paper use convolutional autoencoder (CAE) to achieve the feature learning on an unsupervised way. An autoencoder is a neural network that is trained to reconstruct the input. Internally, it has a hidden layer (or embedded layer) that defines a code to represent the input. The code is representative and the traditional clustering algorithms use it as the features to achieve final detection and classification.

The main insights and contributions of this paper are summarized as follows.

- A new multivariate industrial time series data transformation and augmentation methods are developed, which code the temporal information of the raw time series signals in the 2-D matrices.
- A CAE with augmented input data is used to capture the interaction between the pairs of time series, which learns the spatial information of the raw time series signals.

The rest of this paper is organized as follows. Section 2 gives the definition and characteristics of the typical valve stiction. Section 3 describes the details of the proposed approach. And the experiment is shown in section 4. The conclusions and future work are presented in Section 5.

2 Definition of Valve Stiction

A valve is an actuator and stiction was formally defined in [20], which is illustrated with the phase plot of controller output value (OP) and manipulated variable (MV) in Fig. 1. The typical behavior of valve stiction involves four stages, a deadband, a stickband, a slipjump, and a moving. Assuming a valve is in the position (A) and it sticks. As OP increases, the valve's position does not change because of the deadband (AB) and the stickband (BC). When OP overcomes the deadband and the stickband, the valve suddenly jumps to the new position (D) because of the potential energy stored in the actuator and starts to move. The same behavior occurs in the opposite direction of the valve movement. S and J quantify a stiction behavior in the two-parameters data-driven model of valve stiction [20, 21], where S = *deadband* + *stickband* and J = *slipjump*.

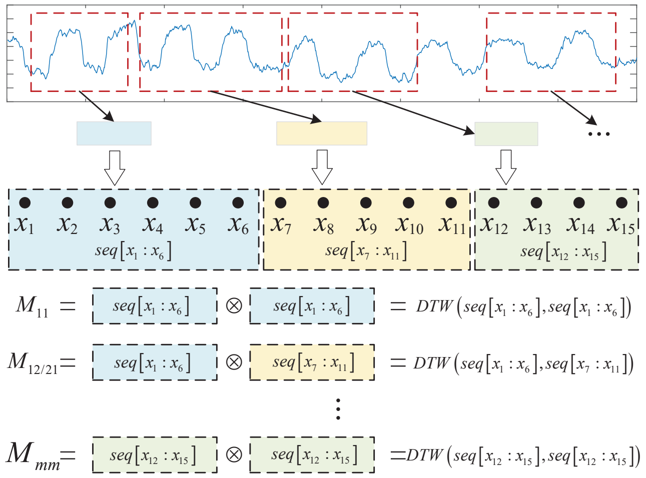

Fig. 2: Example of data conversion.

3 The Proposed Approach

3.1 Data Transformation and Augmentation

In the traditional data-driven approaches, the data transformation method is important since most of the data-driven methods cannot deal with the raw signals directly. One of the main functions of the process is to extract the features of the raw signals from the large volume of historic data. However, to extract the proper features is an exhausted work, and these features have great effects on the final results. In this paper, an effective data conversion stage is developed. The basic idea is to convert the time-domain raw signals into C-channel $M \times M$ images.

As shown in Fig. 2, for a single series, in order to obtain an $M \times M$ size image, the time-domain raw signal is divided into M segments with random lengths. Let $L(i)$, $i = 1, 2, \ldots, M$, denotes the i-th segment of the raw signal. $P(j, k)$, $j = 1, 2, \ldots, M; k = 1, 2, \ldots, M$ denotes the pixel value of the conversed image, as shown in the following equation:

$$P(j,k) = DTW(L(j), L(k)) \tag{1}$$

where $DTW(\cdot)$ denotes dynamic time warping (DTW) distance that is calculated instead of the widely used Euclidean distance. The advantage of the DTW distance is that the two series to be of equal lengths are not necessary [22]. Hence, the DTW distance is calculated by

$$DTW(Q, C) = \underset{W = w_1, \ldots, w_k, \ldots, w_K}{\arg\min} \sqrt{\sum_{k=1, w_k=(i,j)}^{K} (q_i - c_j)^2} \tag{2}$$

Note that each single series can be calculated by (1) and a $M \times M$ size image is obtained. Assume that the industrial data have C series, the image with $C \times M \times M$ size is finally obtained. The advantage of the proposed data conversion method is that the two segments to be of equal lengths are not necessary, which means the complicated temporal information can be encoded on different timescales. Moreover, this data transformation method can be used without any predefined or pretrained parameters and can reduce the reliance on expert experience as much as possible. Moreover, considering the commonly existed characteristics of industrial time series signals, we develop two different data augmentation methods based on the different timescales, which are called fixed-span augmentation and random-span augmentation, separately. The details are as follows.

Given a series $ts = [x_1, x_2, \ldots, x_n]$ and the matrix size m, which is also the number of segments of the raw series. First, we calculate the average length of the segments by

$$L = floor\left(\frac{n}{m}\right) \tag{3}$$

where $floor(\cdot)$ returns the largest integer not greater than $\frac{n}{m}$. For the fixed-span augmentation, the initial and the end indexes are calculated by

$$\begin{aligned} idx_{init} &= [0 \cdot L, 1 \cdot L, 2 \cdot L, \ldots, m \cdot L] \\ idx_{end} &= idx_{init} + L \end{aligned} \tag{4}$$

And for the random-span augmentation, the initial and the end indexes are calculated by

$$\begin{aligned} idx_{init} &= [0 \cdot L, 1 \cdot L, 2 \cdot L, \ldots, m \cdot L] \\ idx_{end} &= idx_{init} + random\left(L, \mu \cdot L\right) \end{aligned} \tag{5}$$

where μ is a integer that controls the range of the segment length. $random\left(L, \mu \cdot L\right)$ returns m integers that are greater than L but smaller than $\mu \cdot L$. Obviously, when m is fixed, selecting a larger n means encoding larger-timescale information of the raw signals, and a smaller n means that the smaller-timescale information is considered. In this study, we performed the data augmentation under different timescales n simultaneously, which provides more information about the raw signals.

3.2 Unsupervised Feature Learning

In this paper, we use a CAE to learn features. A CAE is generally composed of two blocks, corresponding to encoder $F_E(\cdot)$ and decoder $F_D(\cdot)$ respectively. The goal is to find a representative features for each input sample by minimizing the mean squared errors (MSE) between its input and output over all samples, i,3.

$$\min \frac{1}{n}\sum_{i=1}^{n} \left\| F_D\left(F_E\left(x_i\right)\right) - x_i \right\|_2^2 \tag{6}$$

For a fully connected autoencdoer,

$$\begin{aligned} F_E(x) &= \sigma(Wx) = h \\ F_D(h) &= \sigma(Ux) = \hat{x} \end{aligned} \tag{7}$$

where h is a vector and σ is activation function like ReLU, sigmoid, Tanh. After training, the embedded features h regards as the new representation of the input sample. Then h can be fed into another autoencoder to build a stacked autoencoder (SAE). To exploit the spatial structure of images, the convolutional layers replace the fully connected layers, and a convolutional autoencoder is defined as

$$\begin{aligned} F_{ConvE}(x) &= \sigma(x \circ W) = h \\ F_{ConvD}(h) &= \sigma(x \circ U) = \hat{x} \end{aligned} \tag{8}$$

where x and h are matrices or tensors, and $\circ$ is convolution operator. The stacked convolutional autoencoder can be construed in a similar way as SAE. In this paper, we use

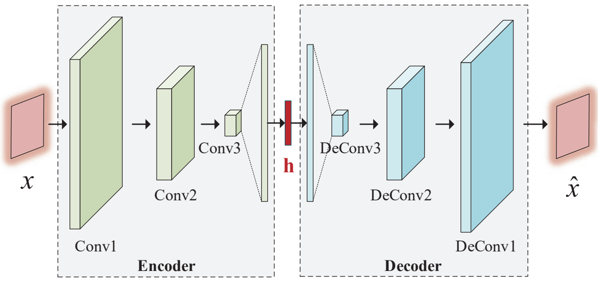

Fig. 3: The structure of convolutional autoencoder (CAE).

a convolutional autoencoder that does not need layer-wise pretraining, as shown in Fig. 3. Some convolutional layers are stacked on the input images to extract hierarchical features. Then flatten all units to form a vector, followed by a fully connected layer which is called embedded layer. The input 2-D image is thus transformed into a low dimension feature space. In order to train it in the unsupervised way, a decoder that consists of a fully connected layer and some stacked convolutional layers is built, which map the embedded layer to original image. The parameters of the encoder and the decoder are updated by minimizing the reconstruction error:

$$L_r = \frac{1}{n}\sum_{i=1}^{n} \left\| F_{D\leftarrow w'}\left(F_{E\leftarrow w}\left(x_i\right)\right) - x_i \right\|_2^2 \tag{9}$$

where n is the number of the samples.

Considering the dimension of the embedded layer is very low compared to the original input. Learning such undercomplete representations forces the autoencoder to capture the most salient features of the data. Note that the data conversion method mentioned in 3.1 capture the temporal information of a single series, and the inputs of CAEs are $C \times M \times M$ size images, thus stacked convolutional layers can capture spatial interactions between different pairs of series.

3.3 Feature Concatenation and Clustering

In this paper, we concatenate the embedded layers of CAEs on different timescales to form the final features. Given the encoder of the trained CAE and the inputs with k-th timescale, which are denoted as F_{ConvD} and x_k, the embedded layer is described as

$$h_{ts} = F_{ConvD}\left(x_k\right) \tag{10}$$

Given a set of timescales, $TS = \{ts_1, ts_2, \ldots, ts_k\}$. The fused features are denoted as

$$H = \left(h_1^T, h_1^T, \ldots, h_k^T\right)^T \tag{11}$$

where h_k denotes the embedded layer under the k-th timescale. Then the fuzzy clustering algorithm with input H is used to find different classes, the objective function is

$$L_c = \sum_{c=1}^{C} \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} m_{ic}^2 m_{jc}^2 d_{ij}}{2\sum_{j=1}^{N} m_{jc}^2} \tag{12}$$

Table 1: Main Simulation Parameters

Process	Class	S	J
$G_1(s)$	Non-stiction	[0]	[0]
$G_1(s)$	Stiction	[0.1: 0.1: 0.7, 1.0: 0.5: 5.0]	[0.0: 0.1: 1.0]
$G_1(s)$	Non-stiction	[0]	[0]
$G_1(s)$	Stiction	[0.1: 0.1: 1.5]	[0.00: 0.01: 0.05]
$G_2(s)$	Non-stiction	[0]	[0]
$G_2(s)$	Stiction	[0.1: 0.1: 0.7, 1.0: 0.5: 5.]	[0.0: 0.1: 1.0]
$G_2(s)$	Non-stiction	[0]	[0]
$G_2(s)$	Stiction	[0.1: 0.1: 1.5]	[0.00: 0.01: 0.05]

Table 2: Experimental Parameters

Type	Parameters
Matrix Size	28×28
Training Timescales	200, 400, 600
Test Timescales	75, 125, 175
Encoder	X →ReLU(Conv2d(2, 32, 3, 2)) →ReLU(Conv2d(32, 64, 3, 2)) →ReLU(Conv2d(64, 128, 3, 2)) →Linear(*,5)
Decoder	Linear(*,5) →ReLU(DeConv2d(128, 64, 3, 2)) →ReLU(DeConv2d(64, 32, 5, 2)) →DeConv2d(32, 2, 5, 2) → X

where m_{ic} represents the unknown membership of the object i in cluster c and d_{ij} is the dissimilarity between object i and j. The memberships are subject to constraints that they all must be non-negative and that the memberships for a single individual must sum to one.

4 Experiment

4.1 Experiment Settings

4.1.1 Model Training

In most cases, obtaining massive industrial data is time-consuming. Inspired by [13, 15], we first generate the additional data for model training through simulation rather than the real industrial data. The following transfer functions are considered.

$$G_1(s) = \frac{1}{0.2s} e^{-0.05s}, \tag{13}$$

$$G_2(s) = \frac{1}{0.2s+1}. \tag{14}$$

Both processes are controlled by PI controllers. In Kano's stiction model, S and J are used to control the stiction degree [21]. When both S and J are equal to zero, a valve is non-stiction and the data is labeled with "non-stiction". Conversely, when S or J are not equal to zero, the data is labeled with "stiction". The main simulation parameters for different transfer functions are listed in Table 1. After collecting the training data, a CAE network is trained using stochastic gradient descent algorithm. The training loss is shown in Fig. 4.

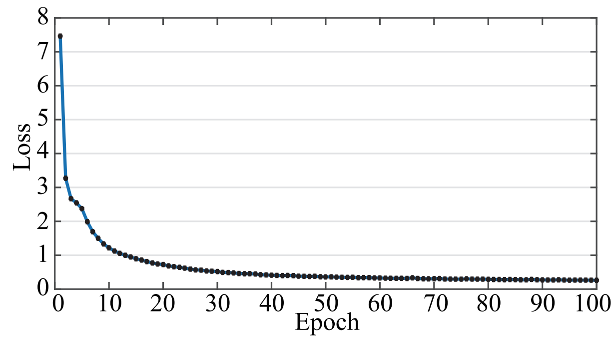

Fig. 4: Training loss.

4.1.2 Model Test

We use a benchmark dataset, The International Stiction Data Base (ISDB), to verify the effectiveness of our approach. ISDB is a comprehensive process control dataset, including self-regulating and integrating control loops. Most loops are flow, temperature, level, and pressure loops. In this paper, 70 loops are selected, in which 30 loops are stiction and 40 loops are non-stiction. The details of the loops are not shown in this paper because of the space limitation, and all information about the loops can be found in [2].

4.1.3 Model Parameters Setting

The experimental parameters mainly include the values of timescales and the model parameters. The main parameters used in our experiments are listed in Table 2. X denotes the inputs of the CAE model, which are two-channels 28×28 matrices calculated through section 3.1. Note that the timescales of training data and test data are different because the training data comes from MATLAB, and the test data is real industrial data. In our experiments, we extract features on three different timescales. In Table 2, conv2d(*) represents a convolutional operation over the 2-D format data, the parameters are the number of channels in the input, number of channels produced by the convolution, kernel size, and stride. Linear(*) denotes a fully connected layer, and the parameters are the size of each input and output.

4.1.4 Evaluation Metrics

We use clustering accuracy to evaluate the performance of the proposed approach. It is commonly used in clustering problems. Generally, the definition of the metric is

$$acc = \max_{perm \in P} \frac{1}{n} \sum_{i=0}^{n-1} 1\left(perm\left(\hat{y}_i\right) = y_i\right) \tag{15}$$

where P is the set of all permutations in $[1, 2, \ldots, K]$ where K is the number of clusters. Moreover, we use classification accuracy to evaluate the comparison methods that belong to supervised methods. The classification accuracy is defined as

$$acc_{clf} = \frac{1}{n} \sum_{i=0}^{n-1} 1\left(\hat{y}_i = y_i\right) \tag{16}$$

where $x \to 1(x)$ is the indicator function: $1\left(\hat{y}_i = y_i\right) = 1$ if $\hat{y}_i = y_i$ and 0 else.

Table 3: Comparison Results on ISDB dataset

Method	Ours	BSD-CNN	SDN	–
Accuracy	0.84	0.76	0.77	–
Method	LR	SVM	RF	Xgboost
Accuracy	0.41	0.78	0.72	0.75
Method	Kmeans	FC	HC	–
Accuracy	0.78	0.80	0.75	–

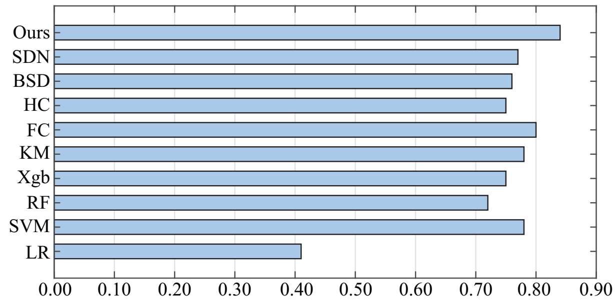

Fig. 5: Comparison Results on ISDB dataset.

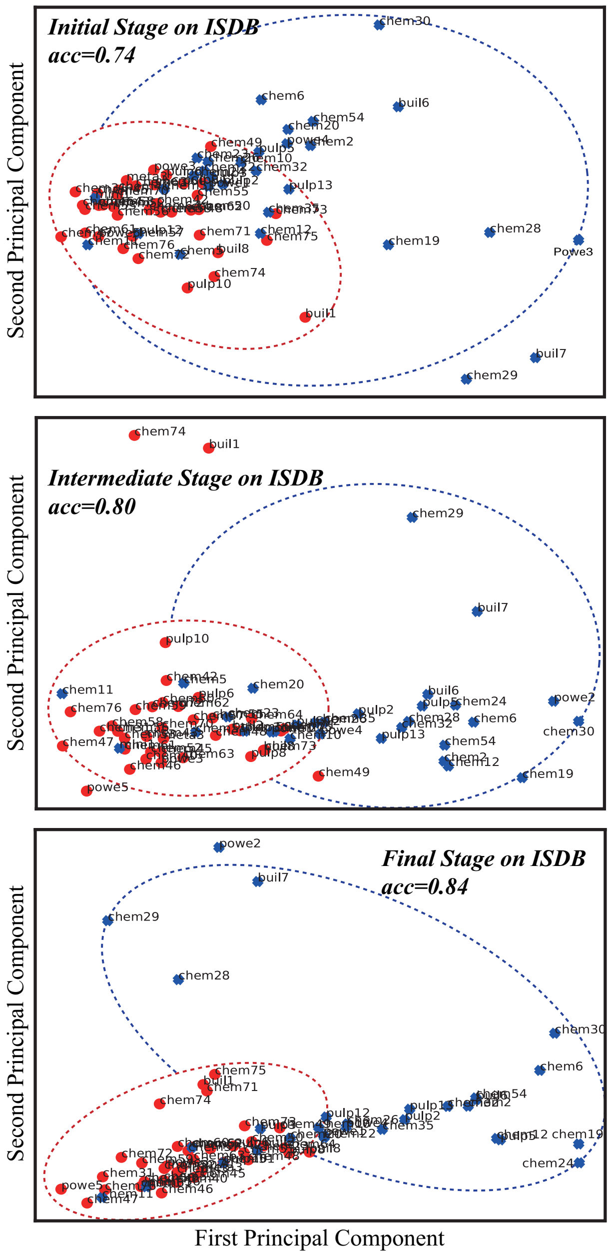

Fig. 6: Visualization of features in different stages.

4.2 Comparison Methods

In the experiments, we compare the proposed method with other methods, including four traditional machine learning methods: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM) , and Gradient Boosting (XgBoost), two DL-based methods: BSD-Convolutional Neural Network (BSD-CNN) [15], Stiction Detection Network (SDN) [13], and four classic clustering algorithm, K-means (KM), hierarchical clustering (HC), and fuzzy clustering (FC).

LR is a machine learning algorithm used for classification problems. It is a predictive analysis algorithm and based on the concept of probability. SVM and its extensions are one class of the most successful machine learning methods. It aims to seek the optimal hyperplane with the maximum margin principle in a high- or infinite-dimensional space. It has a solid theoretical foundation and good generalization ability, which results in wide applications in various fields. RF and XgBoost are both ensemble learning strategies. Ensemble learning a commonly used technique in a data science competition since model performance could always benefit from various algorithms.

BSD-CNN and SDN are both the DL-based methods that were proposed for stiction detection. BSD-CNN is based on a CNN, but the inputs of the network are butterfly shape-based (BSD) images derived from the manipulation of the standard PV and OP data. SDN is based on a multi-layer feed-forward NN and the inputs are transformations format of PV and OP operational data.

KM is one of the simplest and popular unsupervised algorithm, which identifies k number of centroids, and then allocates every data point to the nearest cluster. Each data point only belong to one cluster. FC is a soft form of clustering in which each data point can belong to more than one cluster, and memberships are introduced and indicate the degree to which data points belong to each cluster. In HC, initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until on cluster or K clusters are formed.

4.3 Experimental Results and Visualization

4.3.1 Clustering Results

We use ten comparison methods to compare with the proposed detection strategy on the ISDB dataset in the experiments. Table 3 and Fig. 5 present the experimental results with our proposed detection strategy on ISDB dataset.

Overall, the DL-based methods (BSD-CNN, SDN, and the proposed strategy) achieve higher accuracy than the most traditional supervised methods (LR, RF, Xgboost), even RF and Xgboost are ensemble methods. However, SVM shows the most performance and relatively high metric over the traditional methods and two DL-based methods (BSD-CNN, SDN), proving SVM is still a good classier. And simple KM and FC algorithms achieve relatively high accuracy than the most comparison methods, which means the data transformation method mentioned in section 3.1 has captured the representative features.

Compared to the DL-based methods, BSD-CNN and SDN are supervised networks proposed for valve detection in recent years, and our unsupervised approach is not limited to valve stiction detection but achieve the higher accuracy. We argue that two factors contributed to this result. The first factor is the diversity of the training data. The training data for our approach are generated via a developed transformation and augmentation method on multiple timescales, but the data for BSD-CNN and SDN are generated on a single timescale. The second factor is the different representation ability, BSD-CNN and SDN use the relatively simple networks of fewer layers than our CAE network. We use a CAE model to further extract the features from the transformed data, which results in a higher accuracy.

4.3.2 Visualization

In this section, we provide a visualization of feature learning process using principal component analysis, which is shown in 6. It can be seen that in the initial stage, both stiction and non-stiction loops have a dispersed distribution in the two-dimensional space. After a period of training, the most non-stiction loops lie on the left side of the feature space, and the stiction loops lies on the right side of the two-dimensional feature space. In the final stage, the distribution of non-stiction loops are gathered in a small area, and most data points of the stiction loops lie on the right and the top area. As a result, the clustering accuracy increases from 0.74 to 0.84. This visualization process proves the effectiveness of the proposed unsupervised approach.

5 Conclusions

This paper focused on unsupervised feature representation of industrial valve stiction in a control loop. We propose a learning strategy combining a new data transformation and an augmentation method and a CAE model to detect the stiction of control valves. Comparing the proposed approach with the traditional methods and DL-based methods, it showed that the proposed approach could achieve higher accuracy. In our future work, the quantification evaluation of stiction will be further discussed. And the authors will try to extend the proposed approach to general industrial multivariate time series data.

References

[1] R. Bacci 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 and B. Huang, *Detection and diagnosis of stiction in control loops: state of the art and advanced methods*. London: Springer-Verlag, 2010.

[3] M. Shoukat Choudhury, S. Shah, and N. Thornhill, "Diagnosis of poor control-loop performance using higher-order statistics," *Automatica*, vol. 40, no. 10, pp. 1719–1728, 2004.

[4] A. Zakharov, E. Zattoni, L. Xie, O. P. Garcia, and S.-L. Jämsä-Jounela, "An autonomous valve stiction detection system based on data characterization," *Control Engineering Practice*, vol. 21, no. 11, pp. 1507–1518, 2013.

[5] O. Pozo Garcia, A. Zakharov, and S. Jämsä-Jounela, "Data and reliability characterization strategy for automatic detection of valve stiction in control loops," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 769–780, May 2017.

[6] 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.

[7] C. Scali and C. Ghelardoni, "An improved qualitative shape analysis technique for automatic detection of valve stiction in flow control loops," *Control Engineering Practice*, vol. 16, no. 12, pp. 1501–1508, 2008.

[8] A. Horch, "A simple method for detection of stiction in control valves," *Control Engineering Practice*, vol. 7, no. 10, pp. 1221–1231, 1999.

[9] 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.

[10] T. Hägglund, "A shape-analysis approach for diagnosis of stiction in control valves," *Control Engineering Practice*, vol. 19, no. 8, pp. 782–789, 2011.

[11] 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.

[12] M. S. Choudhury, S. Shah, N. Thornhill, and D. S. Shook, "Automatic detection and quantification of stiction in control valves," *Control Engineering Practice*, vol. 14, no. 12, pp. 1395–1412, 2006.

[13] A. A. A. Mohd Amiruddin, H. Zabiri, S. S. Jeremiah, W. K. Teh, and B. Kamaruddin, "Valve stiction detection through improved pattern recognition using neural networks," *Control Engineering Practice*, vol. 90, pp. 63–84, 2019.

[14] J. W. Dambros, M. Farenzena, and J. O. Trierweiler, "Oscillation detection and diagnosis in process industries by pattern recognition technique," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 299–304, 2019.

[15] B. Kamaruddin, H. Zabiri, A. Mohd Amiruddin, W. Teh, M. Ramasamy, and S. Jeremiah, "A simple model-free butterfly shape-based detection (bsd) method integrated with deep learning cnn for valve stiction detection and quantification," *Journal of Process Control*, vol. 87, no. 4, pp. 1–16, 2020.

[16] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, July 2018.

[17] J. Jiao, M. Zhao, J. Lin, and C. Ding, "Deep coupled dense convolutional network with complementary data for intelligent fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9858–9867, 2019.

[18] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101–110, 2018.

[19] S. Bai, J. Zico Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv e-prints*, p. arXiv:1803.01271, Mar. 2018.

[20] M. Shoukat Choudhury, N. Thornhill, and S. Shah, "Modelling valve stiction," *Control Engineering Practice*, vol. 13, no. 5, pp. 641–658, 2005.

[21] M. Kano, H. Maruta, H. Kugemoto, and K. Shimizu, "Practical model and detection algorithm for valve stiction," *IFAC Proceedings Volumes*, vol. 37, no. 9, pp. 859–864, 2004.

[22] R. J. Kate, "Using dynamic time warping distances as features for improved time series classifications," *Data Mining and Knowledge Discovery*, vol. 30, p. 283–312, 2016.