Multiple Timescale Feature Learning Strategy for Valve Stiction Detection Based on Convolutional Neural Network

Kexin Zhang, Yong Liu, *Member, IEEE*, Yong Gu, Xiaojun Ruan, and Jiadong Wang

***Abstract*—This paper proposes a valve stiction detection strategy based on convolutional neural network (CNN). Considering the commonly existed characteristics of industrial time series signals, the strategy is developed to learn features on multiple timescales automatically. Unlike the traditional approaches using hand-crafted features, the proposed strategy can automatically learn representative features on the time series data collected from industrial control loops. The strategy is composed of two complementary data conversion methods, a mixed feature learning stage and a fusion decision stage, and it has the following merits: 1) the interaction of different pairs of time series can be effectively captured; and 2) the whole process of feature learning is automatic, and no manual feature extraction is needed. The effectiveness of the proposed strategy is evaluated through the comprehensive data, including the International Stiction Data Base (ISDB), and the real data collected from the real hardware experimental system and the industrial environment. Compared with four traditional methods and three deep learning (DL) based methods, the experimental results demonstrate that the proposed strategy outperforms the other methods. Besides performance evaluation, we give the implementation procedure of practical application of the proposed strategy and provide the detailed analysis from the perspective of the data conversion methods and the number of timescales.**

***Index Terms*—Valve stiction detection, convolutional neural network (CNN), feature learning, multiple timescale, hardware experimental system.**

I. 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 the 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 that are capable of automatically learning features from the massive industrial data.

Different approaches have been developed to detect the valve stiction over the past decades, which could be broadly classified into two categories from the perspective of feature learning, feature engineering (FE) based and representation learning (RL) data-driven based. The FE-based methods have great advantages in interpretability and effectiveness. However, they tend to be time-consuming, and reliable prior knowledge and a complete understanding of the actual process are required. In contrast, RL-based data-driven methods employ machine learning (ML) algorithms along with the collected industrial data without the intervention of domain experts, explaining why RL-based methods have received great attention.

In recent years, Deep learning (DL) has emerged as a powerful technique to directly learning distinct and abstract features from the data. DL reduces the efforts in the design of hand-crafted features and provides a way to learn representative features. Several DL-based methods had already been developed for the industrial anomaly detection and diagnosis [3]–[5], proving that the DL-based methods have better performance and higher automation level than the traditional FE-based or ML methods. However, to the best of our knowledge, a reliable application of DL on valve stiction detection is still developing in recent years. The representative literature about this topic are [6], [7], but the representation models are simple and the special input formats are required. The data conversion methods are still hand-crafted processes because the domain knowledge is needed. Therefore, it is necessary to develop a new representation model for valve stiction detection.

Shape-based detection is one of the most effective methods, it is intuitive and easy to understand. The basic idea behind it is to manually extract features from the special diagram of the process variable (PV) versus the controller output (OP) [8], [9]. Nevertheless, the traditional shape-based method still has some drawbacks. First, the detection still heavily relies on the prior knowledge. Second, the feature extraction and the final decision are separately designed and performed, both of which affect the final detection accuracy. Third, the diversity of the features is insufficient, which means the methods barely adapt to a dynamically changing industrial environment.

Alternatively, to address the above drawbacks, DL provides a promising and effective solution. As one of the most representative DL models, Convolutional neural network (CNN) has achieved a breakthrough improvement on image recognition and classification tasks, and the fundamental idea

This work was supported in part by the key R&D Project of Guangdong Province (No. 2019B010120001) and the National Key R&D Program of China (No. 2018YFB1305900).

K. Zhang, Y. Liu, and Y. Gu are with the Institute of Intelligent Systems and Control, Zhejiang University, Hangzhou, 310027, China. (corresponding author: Y. Liu; phone: +86(571) 87951445; e-mail: yongliu@iipc.zju.edu.cn; zhangkexin@zju.edu.cn; gyxm@zju.edu.cn).

X. Ruan and J. Wang are with the Zhejiang Supcon Technology Co. Ltd., Hangzhou 310053, China. (e-mail: ruanxiaojun@supcon.com; wangjiadong1@supcon.com)

behind a CNN is to extract features using multiple stacked convolutional layers with a hierarchical architecture that is similar to the processes of the human recognizing object. CNNs were initially proposed to extract features on images, and the data format of an image is the same as that used in the shape-based methods.

Inspired by the above works, a multiple timescale feature learning strategy is proposed for valve stiction detection. First, the raw time series data are converted to the 2-D format data on the fixed and the unfixed timescales. Second, the mixed data on multiple timescales are used to train two CNNs for feature learning. Finally, an ensemble decision is made by concatenating the features extracted by the trained CNNs on different timescales. The main contributions of this paper are summarized as follows.

1) A data preprocessing process involving two complementary data conversion methods is proposed to capture the interactions between the different industrial time series pairs on multiple timescales. These two conversions can effectively convert the raw time series to the 2-D format data without any prior knowledge.
2) Three successive stages, the mixed feature learning stage, the separate feature extraction stage, and the ensemble decision stage, compose the complete multiple timescale feature learning strategy for valve stiction detection. The proposed strategy can learn representative features through the CNN-based models, and no manual feature extraction is needed.
3) An end-to-end strategy of valve stiction detection is developed and the procedure of practical application is given. Moreover, the proposed strategy is evaluated through the comprehensive data, including the International Stiction Data Base (ISDB), and the real data collected from the real hardware experimental system and the industrial environment.

The rest of this paper is organized as follows. Section II shows related techniques about valve stiction detection and deep learning for fault diagnosis. Section III introduces the proposed strategy. Section IV shows the experimental results and presents the discussion. The conclusions and future work are presented in Section V.

II. Related Work

In this section, we provide a literature review related to valve stiction detection and industrial fault diagnosis.

A. Valve Stiction Detection

A valve is the moving part in a process control loop. The presence of control valve nonlinearities such as stiction, backlash or deadband, is a major cause of oscillations in control loops. Among the many types of nonlinearities in control valves, stiction is the most common and one of the long-standing problems in the process industry. From the perspective of the physical principle, stiction in industrial control valves has been defined, which differentiates it from other similar nonlinear phenomenon in a control valve, such as backlash, hysteresis, and dead-band [1], [6], [10].

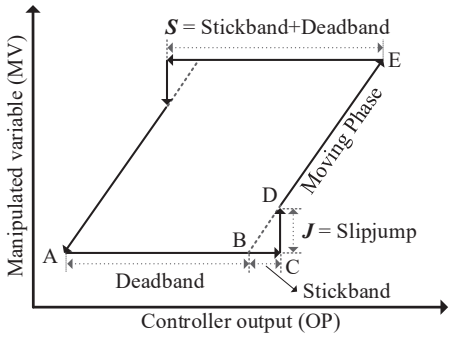


Fig. 1. Typical behavior of valve stiction.

According to the definition, stiction is characterized by two main parameters, namely S and J, where S = *deadband* + *stickband* and J = *slipjump* [10]. The typical stiction in a closed control loop is illustrated with the phase plot of OP and manipulated variable (MV), as shown in Fig. 1. The whole process involves four stages, deadband, stickband, slipjump, and the moving phase.

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 valve stiction model.

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. Choudhury [11] designed two higher-order statistics indexes, and Zakharov [12] proposed four novel data feature indexes quantifying the presence of oscillations, mean-nonstationarity, noise, and nonlinearities in a given data sequence. A similar feature extraction strategy was also adopted in [13]. Some methods were proposed to simultaneously extract related features on time and frequency domain through signal processing methods [14].

The typical stiction behavior results in a special shape or pattern in the phase plot of OP, MV, and PV [8], [9]. Extracting the specific features from the image encoded with raw time series data seems more intuitive, and it is easy to implement. The methods based on observed features in the images are summarized as shape-based methods [15], [16]. 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 on the dynamic changing environment.

A wide variety of problems can cause the failure of a detection system. In cases where the prior knowledge of a control loop is either unknown or too complicated, the use of advanced pattern recognition approaches is becoming more attractive as an alternative data-based strategy for valve stiction

detection. Amiruddin [7] transformed the raw time samples of PV-OP to D values and used an artificial neural network (ANN) to detect stiction. Dambros [17] also employed simple ANN but the inputs of the network are PV(OP) diagrams. Kamaruddin [6] replaced ANN with CNN to learn features on the designed "butterfly" shape images. The use of simple learning networks on real industrial data is also susceptible to noise and 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.

B. Deep Learning for Fault Diagnosis

As one of the representative DL methods, CNN is a special feed-forward ANN that performs convolutional operations and has a deep structure. It is one of the successful representative algorithms of deep learning and has been widely used in computer vision (CV) and natural language processing (NLP). Compared to the traditional fully connected ANN, CNN is a particular network that uses the hierarchical pattern in data and assembles more complex patterns using simpler patterns. Furthermore, local-connectivity characteristics with shared-weights scheme reduce the number of network parameters and prevent a CNN from over-fitting. CNNs initially show success in CV problems like object localization, object classifications, and image recognition. Lecun [18] designed LeNet-5 for the real-life document recognition system. Krizhevsky [19] trained a large, deep CNN to classify the 1.2 million high-resolution images and the recognition results are better than the previous state-of-the-art methods.

For the task considered in this paper, some simple models using ANNs [7], [17] and CNN [6] were proposed for valve stiction detection in control loop performance assessment. Stiction detection is essentially a fault diagnosis and failure prognosis task. Motivated by the success of deep learning in various classification and recognition tasks, lots of DL-based methods that applied to fault diagnosis tasks have been proposed [20]. These methods achieved better results than traditional data-driven methods because of the powerful feature representation ability.

For example, Yuan [21] used multiscale CNN to extract the features from the new transformed representation of the raw data. Qiu [22] proposed a multi-fusion CNN (MFCNN) which fused the original signal and its corresponding frequency information. The above methods directly deal with the time series data or its transformed variants, which ignore the interactions between different pairs of variables. Therefore some works transform the raw signals to two-dimensional data and then feed the transformed data to a DL-based network. Wen [23] proposed a signal-to-image conversion method for converting the raw signals into two-dimensional images that are more suitable for CNN processing. Xia [24] combined the raw from multiple sensors into a 2-D matrix at the data level and employed a CNN-based model for rotating machinery diagnosis. Besides CNNs, many effective ML and DL methods were proposed for fault diagnosis, including the adaptive bayesian algorithm for wind turbine bearings diagnosis [25], residual learning algorithm for rotating machinery diagnosis [26], and deep transfer learning strategy for machine health monitoring [27]. These successful applications prove the effectiveness of the DL in fault diagnosis especially the CNN-based strategies.

Inspired by these works, we focus on exploring multiple timescale feature learning strategy for valve stiction detection in control loops in this paper. We present two complementary data conversion methods for capturing the interactions between different industrial time series pairs on multiple timescales, and then develop a mixed feature learning and fusion decision method to achieve the stiction detection of a control valve.

III. Proposed Strategy For Valve Stiction Detection

This paper focuses on the stiction detection of a control valve and the running condition of a control loop is changing under multiple timescales. Therefore, we argue that the information about multiple timescales is necessary for reliable detection. In addition, the correlations between different time series pairs are critical to characterize the system status. To better extract features under multiple timescales and capture interactions of the raw observation sequences, this paper incorporates interactions of different pairs of signals under multiple timescales into the traditional CNN-based network. The proposed strategy is illustrated in Fig. 2.

A. Data Conversion

The raw industrial data collected from a control loop is in sequence with time stamps, and in this paper the time stamps are denoted by $t_1, t_2, \ldots, t_T$. In a typical control loop, the main data are collected from two sensors, i.e., OP and PV, which are denoted by $\mathbf{x}_{op} = \left\{x_{op}^{t_1}, x_{op}^{t_2}, \ldots, x_{op}^{t_T}\right\}$ and $\mathbf{x}_{pv} = \left\{x_{pv}^{t_1}, x_{pv}^{t_2}, \ldots, x_{pv}^{t_T}\right\}$, respectively. Additionally, this paper proposes to make predictions on a time span rather than each time stamp, and the prediction outcomes involve non-stiction and stiction classes. Thus, the problem could be transformed into a binary time series classification problem. The training data is represented as $\mathbf{D}_{Train} = \left\{\left(\mathbf{x}_{op}^{s_1}, \mathbf{x}_{pv}^{s_1}, y_1\right), \ldots, \left(\mathbf{x}_{op}^{s_i}, \mathbf{x}_{pv}^{s_i}, y_i\right), \ldots, \left(\mathbf{x}_{op}^{s_N}, \mathbf{x}_{pv}^{s_N}, y_N\right)\right\}$, in which $\mathbf{x}_{op}^{s_i}$ and $\mathbf{x}_{pv}^{s_i}$ represent the i-th segment of $\mathbf{x}_{op}$ and $\mathbf{x}_{pv}$, respectively. y_i represent the classification label of i-th data sequence. The test data is represented as $\mathbf{D}_{Test} = \left\{\left(\mathbf{x}_{op}^{loop_1}, \mathbf{x}_{pv}^{loop_2}\right), \ldots, \left(\mathbf{x}_{op}^{loop_m}, \mathbf{x}_{pv}^{loop_m}\right), \ldots, \left(\mathbf{x}_{op}^{loop_M}, \mathbf{x}_{pv}^{loop_M}\right)\right\}$, which means that given data $\mathbf{x}_{op}^{loop_m}$ and $\mathbf{x}_{pv}^{loop_m}$ of the m-th loop, inferring the real class of this loop.

Data preprocessing is essential since most data-driven methods cannot directly handle raw industrial data. The primary purpose of this process is to extract useful features from the historical data. However, extracting the representative features is exhausting work, and these features have significant effects on the final results. This paper develops two complementary data conversion methods to handle the sequence data on fixed and unfixed timescales.

1) Fixed Timescale Conversion: The interactions between different pairs of time series are critical to characterize the system status [28]. To represent the inter-correlations between $\mathbf{x}_{op}^{s_i}$ and $\mathbf{x}_{pv}^{s_i}$, a $n \times n$ distance matrix $\mathbf{M^i}$ is constructed

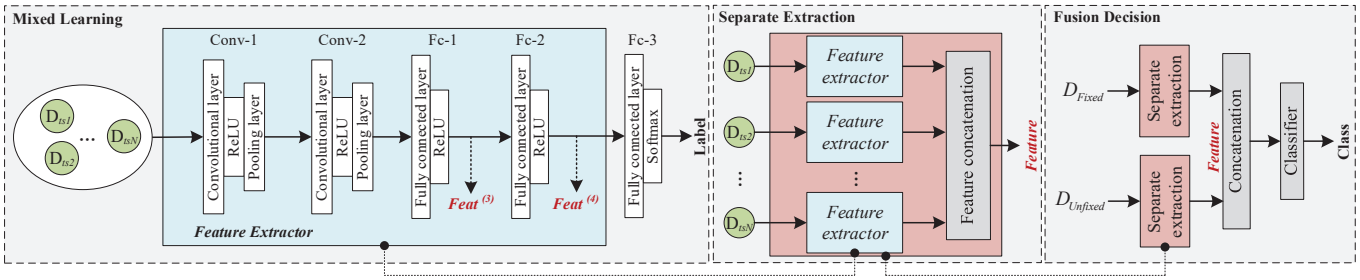


Fig. 2. Strategy of multiple timescale feature learning.

based on the pairwise Euclidean distance of two series within segment s_i. Specifically, given $\mathbf{x}_{op}^{s_i} = \left(x_{op}^0, x_{op}^1, \ldots, x_{op}^{|s_i|}\right)$ and $\mathbf{x}_{pv}^{s_i} = \left(x_{pv}^0, x_{pv}^1, \ldots, x_{pv}^{|s_i|}\right)$, where $|s_i|$ denotes the length of the segment s_i. The interaction term $m_{uv}^i \in \mathbf{M^i}$ is calculated with

$$m_{uv}^i = \sqrt{\frac{\sum_{j=1}^{|s_i|/n} \left(x_{op}^{u \leftarrow j} - x_{pv}^{v \leftarrow j}\right)^2}{(|s_i|/n)}} \tag{1}$$

where $SN = |s_i|/n$ is a re-scale factor ensuring that the segments of different lengths generate the distance matrices with the same size, i.e., $n \times n$. $u \leftarrow \sum j$ and $v \leftarrow \sum j$ indicate that an element in M^i is calculated through SN sequential time stamps in $\mathbf{x}_{op}^{s_i}$ and $\mathbf{x}_{pv}^{s_i}$. Obviously, the length of the segment s_i is not arbitrary, which is limited to integer multiples of the n. This is why this preprocessing method is called fixed timescale conversion. Although the limitation exists, it can not only capture the interactions between two time series but also automatically adapt to the sequences of specific lengths. The conversion is illustrated in Fig. 3, in which $\otimes$ denotes the operation described in (1).

2) Unfixed Timescale Conversion: In order to remedy the limitation of fixed timescale conversion method, we use another conversion method to handle the sequences of arbitrary lengths. Also given $\mathbf{x}_{op}^{s_i} = \left(x_{op}^0, x_{op}^1, \ldots, x_{op}^{|s_i|}\right)$ and $\mathbf{x}_{pv}^{s_i} = \left(x_{pv}^0, x_{pv}^1, \ldots, x_{pv}^{|s_i|}\right)$. We first construct an image with $p \times p$ pixels, and then creates a 2-D line plot of the data in $\mathbf{x}_{pv}^{s_i}$ versus the corresponding values in $\mathbf{x}_{op}^{s_i}$. The process of converting raw time series is similar to that of a human recognizing stiction. This conversion method is regarded as a feature supplement to the fixed timescale conversion method, especially for the sequences of special shapes. The conversion is illustrated in Fig. 4.

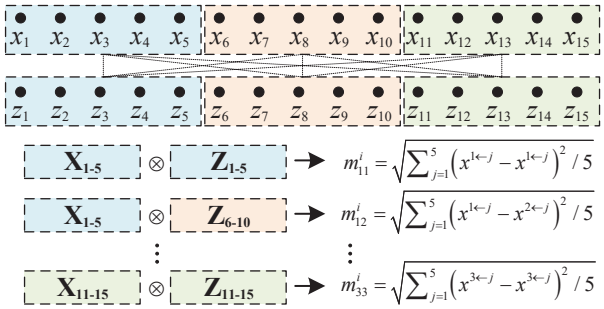


Fig. 3. An illustration of fixed timescale conversion. (Considering i-th timescale and a 3×3 distance matrix)

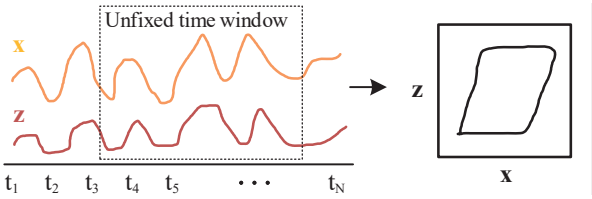


Fig. 4. An illustration of unfixed timescale conversion.

B. Convolutional Neural Network

CNN is composed of trainable multistage architectures involving linear and nonlinear operations. CNNs extract local features at lower layers and then combine them into more abstract features at higher layers. A general feedforward process is represented as follows

$$f(\mathbf{x}) = f_L\left(\ldots f_2\left(f_1\left(\mathbf{x}, \theta_1\right), \theta_2\right) \ldots, \theta_L\right) \tag{2}$$

where $\mathbf{x}$ is the input data, and the input of the network is usually 2-D format arrays or 3-D format images of shape $C \times W \times H$. $\theta_1, \theta_2, \ldots, \theta_L$ are learnable parameters such as weights and biases at each layer. $f = \{f_1, f_2, \ldots, f_L\}$ represents a series of layers that are used to extract features.

1) Convolutional Layer: The convolutional layer usually consists of a series of learnable kernels and biases. Considering the k-th feature map in the l-th layer, it can be expressed as

$$\mathbf{X}_k^l = F_{acti}\left(\sum_{k=1}^{K} \mathbf{W}_{k'k}^l * \mathbf{X}_{k'}^{l-1} + b_k^l\right) \tag{3}$$

where $k' = \{1, 2, \ldots, K'\}$ is the index of the input feature map, and $*$ is the valid 2-D convolution operator. $\mathbf{X}_{k'}^{l-1}$ denotes the input feature map and $\mathbf{W}_{k'k}^l$ is the weight matrix that applied to k'-th feature map at $l-1$-th layer. After a convolutional operation, a nonlinear activation function F_{acti} is applied to increase the nonlinear property. Rectified Linear Unit (ReLU) is widely used and proved to work far better in most of the classification tasks for its capacity to accelerate the convergence and alleviate the vanishing gradient problem [29]. ReLU is expressed as

$$F_{acti}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x}) = \max(0, \mathbf{x}). \tag{4}$$

2) Pooling Layer: Pooling operation is an important process in a CNN. Pooling is a form of nonlinear down-sampling, which reduces each map size and the network parameters, achieving spatial invariance. The average ($Avg(\cdot)$) and the max ($Max(\cdot)$) are two commonly used strategies for pooling operation. Max pooling is generally used and the definition is

$$y_{i,j}^{pool} = \max_{m=0,\ldots,kh-1;n=0,\ldots,kw-1} \left(y_{i\le i'<i+m, j\le j'<j+n}\right). \quad (5)$$

3) Fully Connected Layer: In a typical CNN, the feature learned from the last convolutional layer is often mapped to a vector, which is taken as the input of a fully connected network with the Softmax function which estimates the probability of each input sample belonging to each class. Specifically, the last layer of a typical CNN is a fully connected layer, and the number of neurons in this layer is usually the same as the number of classes. Therefore, given N input samples and C classes, the output of the network can be expressed as $\mathbf{y}^{N\times C}$, each row of the output corresponds to the output of an input sample, which is a C-dimensional vector. Then applying the softmax function on this vector, the probabilities to each class can be calculated by

$$p_{n,c} = \log\left(\frac{\exp(y_{n,c})}{\sum_{c=1}^{C}\exp\left(y_{n,c}\right)}\right). \quad (6)$$

Given the target classes of N input samples, the loss can be calculated by

$$loss\left(\mathbf{p},\mathbf{t}\right) = -\frac{1}{N}\sum\nolimits_{i=1}^{N} t_i \log p_i \quad (7)$$

where $\mathbf{t}$ is a vector corresponding to the target classes. The parameters in the network are updated by minimizing the loss function over the training stage.

C. Strategy of Multiple Timescale Feature Learning

The architecture of the complete feature learning consists of a mixed learning stage and a separate extraction decision making stage. The architecture is shown in Fig. 2.

1) Mixed Learning: Through the fixed timescale conversion and unfixed timescale conversion in section III-A, two datasets, $D_{Fix} = \left\{D_{Fix}^{(ts\_1)}, \ldots, D_{Fix}^{(ts\_i)}, \ldots, D_{Fix}^{(ts\_N_{Fix})}\right\}$ and $D_{Unfix} = \left\{D_{Unfix}^{(ts\_1)}, \ldots, D_{Unfix}^{(ts\_j)}, \ldots, D_{Unfix}^{(ts\_N_{Unfix})}\right\}$, are generated. $D_{Fix}^{(ts\_i)}$ represents the dataset generated through the fixed timescale conversion method under the i-th timescale, where $i \in \{1,2,,\ldots,N_{Fix}\}$ and $D_{Unfix}^{(ts\_j)}$ represents the dataset generated through the unfixed timescale conversion method under the j-th timescale, where $j \in \{1,2,,\ldots,N_{UnFix}\}$.

Then based on the LeNet-5 [18], we construct two CNNs to learn features on D_{fix} and D_{Unfix}. The reason for using LeNet-5 is that the basic structure of this network is relatively simple but effective, and it does not require many computing resources and can be trained only on a CPU. Two CNNs are defined as Net_{Fix} and Net_{Unfix}, respectively. In this paper, we adopt a mixed learning strategy, which means that the datasets under different timescales are used to train one CNN. For achieving that, the same input format is required. For D_{fix}, the re-scale factor SN is developed. For D_{Unfix}, we set the same pixel size for each image. On the one hand, the mixed training can strengthen the learning and generalization capabilities of the network since it is more diverse than a single timescale input. On the other hand, compared to training the different networks at each timescale, fewer parameters are required to training a CNN under all timescales. The learning stages for Net_{Fix} and Net_{Unfix} can be expressed as

$$Net_{Fix} = Learning\left(CNN\left(D_{Fix}, Label_{Fix}\right)\right) \quad (8)$$

$$Net_{Unfix} = Learning\left(CNN\left(D_{Unfix}, Label_{Unfix}\right)\right) \quad (9)$$

where $Label_*$ are the actual class sets corresponding to the D_*. Like other DL-based networks, the whole feature learning process uses training data to adapt its parameters (weights and biases) to perform the desired task and the parameters of the network are optimized using Adam optimizer [30].

2) Separate Extraction and Fusion Decision: Given the trained networks Net_{Fix} and Net_{Unfix}, the features on each timescale can be extracted, and this process is called a separate extraction stage since the operations performed on each timescale. The extracted features can be expressed as

$$feat_{Fix}^{ts\_i(l)} = Net_{Fix}\left(D_{Fix}^{ts\_i}\right)[l] \quad (10)$$

$$feat_{Unfix}^{ts\_j(l)} = Net_{Unfix}\left(D_{Unfix}^{(ts\_j)}\right)[l] \quad (11)$$

where $feat_{Fix}^{ts\_i(l)}$ represents the features of the l-th layer in the network under the i-th timescale, and $feat_{Unfix}^{ts\_j(l)}$ represents the features of the l-th layer in the network under the j-th timescale. In Fig. 2, two illustrations of the position of the features are provided, $feat^{(3)}$ and $feat^{(4)}$ are located in the output of 3-rd and 4-th layer, respectively. The final features are the concatenation of the features on different timescales, which are denoted as

$$Feats = Concat\left(\begin{array}{l} feat_{Fix}^{ts\_1}, \ldots, feat_{Fix}^{ts\_N_{Fix}}, \\ feat_{Unfix}^{ts\_1}, \ldots, feat_{Unfix}^{ts\_N_{Unfix}} \end{array}\right). \quad (12)$$

$Feats$ contain the information of the raw time series and interactions between series on the different timescales. Given a classifier clf, it is easy to predict the class of the dataset. In this paper, the classes are stiction and non-stiction. Therefore, the decision result is expressed as

$$pred = clf\left(Feats\right). \quad (13)$$

The decision is essentially an ensemble learning strategy. The features extracted on different timescales correspond to the different characteristics of a control loop. Moreover, the classifier can be set as a simple logistic regression (LR) or a neural network. The complete process of feature learning and decision is working automatically as long as the training data is provided.

IV. Experiments

In the experiments, first three DL-based methods were compared on International Stiction Data Base (ISDB), and then the proposed method was tested on a real hardware system and a real industrial environment. Total seven real valves were tested.

TABLE I
SIMULATION PARAMETERS FOR TRAINING DATA GENERATION

Process	Class	$Input_{wave}$	K_p	K_i	S	J	Std_{noise}
$G_1(s)$	Non-stiction	$\sin(2\pi t)+2$	[4: 0.2: 8]	[0.05: 0.02: 0.2]	[0]	[0]	[0.005, 0.01]
$G_1(s)$	Stiction	$\sin(2\pi t)+2$	4	0.01	[0.1: 0.1: 0.7, 1.0: 0.5: 5.0]	[0.0: 0.1: 1.0]	[0]
$G_1(s)$	Non-stiction	$\varepsilon(t-0.05)$	[4: :0.2: 8]	[0.05: 0.05: 0.2]	[0]	[0]	[0.005, 0.01]
$G_1(s)$	Stiction	$\varepsilon(t-0.05)$	6	0.2	[0.1: 0.1: 1.5]	[0.00: 0.01: 0.05]	[0]
$G_2(s)$	Non-stiction	$\sin(2\pi t)+2$	[4: 0.2: 8]	[0.1: 0.1: 0.5]	[0]	[0]	[0.005, 0.01]
$G_2(s)$	Stiction	$\sin(2\pi t)+2$	6	0.1	[0.1: 0.1: 0.7, 1.0: 0.5: 5.]	[0.0: 0.1: 1.0]	[0]
$G_2(s)$	Non-stiction	$\varepsilon(t-0.05)$	[4: 0.2: 8]	[0.05: 0.05: 0.2]	[0]	[0]	[0.005, 0.01]
$G_2(s)$	Stiction	$\varepsilon(t-0.05)$	6	0.5	[0.1: 0.1: 1.5]	[0.00: 0.01: 0.05]	[0]

A. Experimental Settings

1) Training Data Generation Through Simulation: The proposed method is working under the supervised learning strategy, so the labeled data for training the network is necessary. However, in most cases, obtaining industrial labeled data is time-consuming or even impossible. Inspired by [6], [7], we first generate the labeled data for model training through the simulation of feedback control loops rather than the real industrial data. In the simulation system, the following transfer functions are considered

$$G_1(s) = \frac{1}{0.2s} e^{-0.05s} \tag{14}$$

$$G_2(s) = \frac{1}{0.2s+1}. \tag{15}$$

Both simulation processes are controlled by PI controllers, and the noise is also considered. In Kano's stiction model, *S* and *J* are used to control the stiction degree. 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 simulation parameters for different transfer functions are listed in Table I. $Input_{wave}$ represents the input signals. Two different input signals are considered because the patterns are different under different input signals. K_p and K_i are proportional and integral gains. *S* and *J* are valve stiction parameters. Std_{noise} represents the standard deviation of the Gaussian white noise added to the processes. Because of the fewer values in parameters *S* and *J* for non-stiction loops, we set more values of K_p, K_i to ensure that the amount of data in each class is roughly equal.

2) Comparison of Different Methods: In the experiments, we compared the proposed method with other seven methods, including Logistic Regression (LR), Random Forest (RF) [31], Support Vector Machine (SVM) [32], Extreme Gradient Boosting (XgBoost) [33], LeNet-5 [18], BSD-Convolutional Neural Network (BSD-CNN) [6], Stiction Detection Network (SDN) [7]. LR is a predictive analysis algorithm and based on the concept of probability. Most importantly, LR maps predicted values to probabilities through Sigmoid function, explaining why the outcome could be interpreted as a 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. RF consists of multiple random decision trees. Two types of randomnesses are considered in the whole process. First, each tree is built on a random sample from the original data. Second, a subset of features is randomly selected for the best split. A RF makes the prediction by averaging the predictions from all the individual decision trees. Xgboost is an efficient and scalable implementation of the Gradient Boosting Machine (GBM). Compared to the general GBM, The optimization of Xgboost takes the Taylor expansion of the loss function up to the second-order, and the model uses a more regularized model formalization to control over-fitting, which provides better performance.

LeNet-5 was proposed to classify hand-written digits on bank cheques automatically. It is a representative CNN because before it was proposed, character recognition was done mostly using hand-crafted features. The LeNet-5 architecture consists of two sets of convolutional and average pooling layers, followed by a flattening covolutional layer, two fully-connected layers, and a softmax classifier.

BSD-CNN and SDN are both the DL-based methods 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 network and the inputs are transformations of PV and OP data.

3) 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 II. In Net_{Unfix}, we directly take images (.jpg) as the training and test data, and in Net_{Fix} we use 2-D matrices because the element in the matrices is calculated through (1). Since we set the size of the matrix is 50×50, the timescales of training data and test data are all multiples of 50. 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 four different timescales. In Table II, 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

TABLE II
EXPERIMENTAL PARAMETERS

Model	Main Settings
Net_{Unfix}	Net: Conv2d(3, 8, 5, 1)→ReLU()→Maxpool2d(2,2)→ Conv2d(8, 25, 5, 1)→ReLU()→Maxpool2d(2,2)→ Fc(625,120)→ReLU()→Fc(120,84)→ReLU()→Fc(82,2) $Data$: Format: .jpg Image Size: 32 × 32 pixels Training Timescale: 200, 300, 400, 600 Test Timescale: 50, 75, 100, 200
Net_{fix}	Net: Conv2d(1, 8, 5, 2)→ReLU()→Maxpool2d(2,2)→ Conv2d(8, 25, 5, 1)→ReLU()→Maxpool2d(2,2)→ Fc(225,120)→ReLU()→Fc(120,84)→ReLU()→Fc(82,2) $Data$: Format: 2-D matrix Matrix Size: 50 × 50 Training Timescale: 200, 300, 400, 600 Test Timescale: 50, 100, 150, 200

size, and stride. Fc(*) denotes a fully connected layer, and the parameters are the size of each input and output.

B. Experimental Evaluation

This paper uses the benchmark stiction dataset and a real hardware experimental system to evaluate the performance of the proposed strategy.

1) International Stiction Data Base: The International Stiction Data Base (ISDB) was provided by [2] and it is a comprehensive process control dataset, including self-regulating and integrating control loops. Most loops are flow, temperature, level, and pressure loops. Sixty loops are chosen for testing in this paper, in which 30 loops are stiction and 30 loops are non-stiction. Note that non-stiction does not mean the loop is normal. The non-stiction loops may exist other problems such as disturbance. The main information of the test loops is described in Table III, where Tem, Flo, Pre, Lev, Con, Gau denote the temperature, flow, pressure, level, concentration, and gauge control loops, respectively.

2) Hardware Experimental System: A real hardware experimental system with three control valves is used to verify our proposed strategy. The system consists of a liquid level loop and two flow loops. The process flow chart and the experimental system are shown in Fig. 5 and Fig. 6. FIC201 and FIC202 are flow loops, and LIC201 is the level loop. V201, V202, and V203 represent the valves, and M201, M202, and M203 represent the magnetic flow meters. L201 represents a pressure sensor that measures the bottom pressure of Tank 202, and then the pressure value is transformed to the liquid level. V203 and V202 control the water flow into the Tank202, and V201 controls the water flow out of the Tank202.

3) Evaluation Metrics: Three metrics, $precision$ (P), $recall$ (R), and $F1$ score ($F1$) were used to evaluate the performance of the proposed strategy. These three metrics are commonly used in classification problems. Generally, the definitions of the metrics are

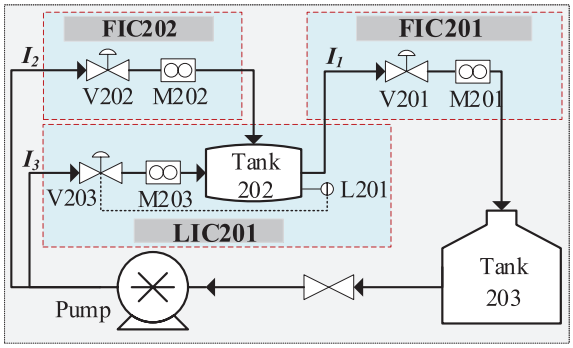
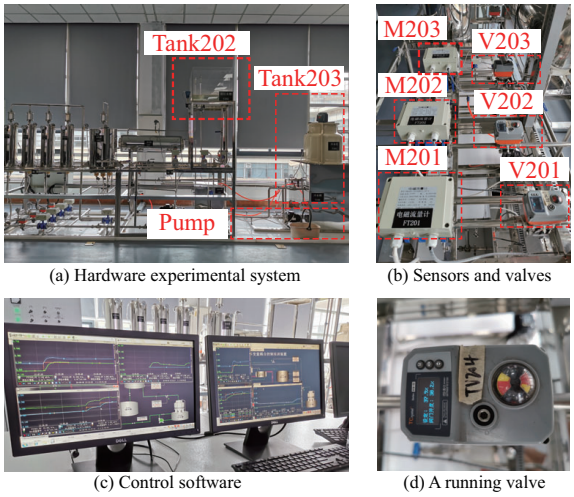


Fig. 5. The flow chart of the hardware experimental system.



(a) Hardware experimental system (b) Sensors and valves

(c) Control software (d) A running valve

Fig. 6. The real hardware experimental system.

$$P = TP/\left(TP + FP\right) \tag{16}$$

$$R = TP/\left(TP + FN\right) \tag{17}$$

$$F1 = \left(2 \cdot P \cdot R\right)/\left(P + R\right) \tag{18}$$

where TP is True Positive which represents the number of test samples that both the predicted label and the actual label are positive. FP is False Positive which represents the number of test samples that the predicted labels are positive, but the actual labels are negative. Contrary to FP, FN is False Negative which denotes the number of samples that the predicted labels are negative, but the actual labels are positive. Generally, a high P means that the predicted positive samples are correct and a high R means that most real positive samples are recognized, so both P and R should be high. $F1$ is the harmonic mean of precision and recall. Generally, the higher the $F1$, the better the model performance.

C. Experimental Results and Discussions

1) Experimental Results on ISDB: We use seven methods to compare with the proposed detection strategy on the ISDB dataset in the experiments. Besides, we deploy the model on a real hardware system, which has three valves. Table IV presents the experimental results with our proposed detection

TABLE III
MAIN INFORMATION OF TEST LOOPS

Loop	Type	T_s[s]	Comment	Loop	Type	T_s[s]	Comment	Loop	Type	T_s[s]	Comment
BAS 6	Tem	1	$Stiction$	CHEM 26	Pre	12	$Stiction$	CHEM 64	Flo	60	$No-stiction$
BAS 7	Tem	1	$Stiction$	CHEM 28	Tem	12	$Stiction$	CHEM 71	Lev	1	$No-stiction$
CHEM 1	Flo	1	$Stiction$	CHEM 29	Flo	60	$Stiction$	CHEM 72	Lev	1	$No-stiction$
CHEM 2	Flo	1	$Stiction$	CHEM 30	Flo	15	$Stiction$	CHEM 74	Lev	1	$No-stiction$
CHEM 5	Flo	1	$Stiction$	CHEM 31	Flo	15	$No-stiction$	CHEM 76	Tem	1	$No-stiction$
CHEM 6	Flo	1	$Stiction$	CHEM 32	Flo	10	$Stiction$	MET 1	Gau	0.05	$No-stiction$
CHEM 10	Pre	1	$Stiction$	CHEM 33	Flo	12	$No-stiction$	MET 3	Gau	0.05	$No-stiction$
CHEM 11	Flo	1	$Stiction$	CHEM 34	Flo	10	$No-stiction$	MIN 1	Tem	60	$Stiction$
CHEM 12	Flo	1	$Stiction$	CHEM 35	Flo	10	$Stiction$	POW 1	Lev	5	$Stiction$
CHEM 14	Flo	20	$No-stiction$	CHEM 37	Lev	12	$No-stiction$	POW 2	Lev	5	$Stiction$
CHEM 15	Pre	20	$No-stiction$	CHEM 38	Pre	10	$No-stiction$	POW 3	Lev	5	$No-stiction$
CHEM 16	Pre	20	$No-stiction$	CHEM 43	Tem	60	$No-stiction$	POW 4	Lev	5	$Stiction$
CHEM 18	Flo	12	$Stiction$	CHEM 45	Pre	60	$No-stiction$	PAP 1	Flo	1	$Stiction$
CHEM 19	Flo	12	$Stiction$	CHEM 46	Pre	60	$No-stiction$	PAP 2	Flo	1	$Stiction$
CHEM 20	Flo	1	$Stiction$	CHEM 52	Lev	60	$No-stiction$	PAP 4	Con	1	$No-stiction$
CHEM 21	Flo	12	$No-stiction$	CHEM 53	Lev	60	$No-stiction$	PAP 5	Con	0.2	$Stiction$
CHEM 22	Flo	12	$Stiction$	CHEM 54	Lev	60	$No-stiction$	PAP 6	Lev	1	$No-stiction$
CHEM 23	Flo	12	$Stiction$	CHEM 56	Flo	60	$No-stiction$	PAP 9	Tem	5	$No-stiction$
CHEM 24	Flo	12	$Stiction$	CHEM 60	Flo	60	$No-stiction$	PAP 12	Lev	15	$Stiction$
CHEM 25	Pre	12	$No-stiction$	CHEM 61	Flo	60	$No-stiction$	PAP 13	Lev	15	$Stiction$

TABLE IV
COMPARISON RESULTS ON ISDB DATASET

Method	Stiction			Non-stiction		
	P	R	$F1$	P	R	$F1$
Ours	**0.87**	**0.90**	**0.89**	**0.90**	**0.87**	**0.88**
LR	0.57	0.53	0.55	0.56	0.60	0.58
RF [31]	0.74	0.47	0.57	0.61	0.83	0.70
SVM [32]	0.68	0.70	0.69	0.69	0.67	0.68
Xgboost [33]	0.63	0.40	0.49	0.56	0.77	0.65
LeNet-5 [18]	0.80	0.80	0.80	0.80	0.80	0.80
BSD-CNN [6]	0.75	0.73	0.74	0.77	0.79	0.78
SDN [7]	0.80	0.80	0.80	0.73	0.73	0.73
Only Net_{Fix}	0.76	0.63	0.69	0.69	0.80	0.74
Only Net_{Unfix}	0.96	0.77	0.85	0.81	0.97	0.88

TABLE V
EXPERIMENTAL RESULTS ON REAL HARDWARE SYSTEM AND INDUSTRIAL ENVIRONMENTS

Loop	Stic?	Ours	LR	RF	SVM	Xgboost	LeNet-5
V201	0	0	1	1	1	1	0
V202	0	0	0	1	0	0	1
V203	0	0	0	0	0	0	0
PIC23002	0	0	0	0	0	0	1
FIC3107	0	1	0	0	0	0	1
FIC2228	1	1	0	0	1	0	1
F6304	1	1	0	1	0	1	1

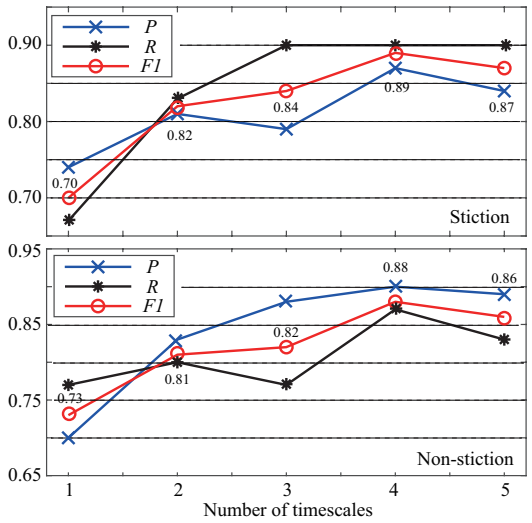


Fig. 7. Metrics on different number of timescales.

strategy on ISDB dataset. In the experiments, the DL-based methods (LeNet-5, BSD-CNN, SDN, and the proposed strategy) achieve better performance than the traditional methods (LR, RF, SVM, Xgboost). SVM shows the most balanced performance and relatively high metrics over the traditional methods in both classes, proving SVM is still a good classier. The traditional ensemble methods show unbalanced results since RF and Xgboost achieve the higher $F1$ in class non-stiction than that in class stiction.

The DL-based methods achieve better performance since they can learn effective and representative features. BSD-CNN and SDN are relatively simple networks proposed for valve stiction detection, and LeNet-5 has been proposed for about two decades. However, The LeNet-5 achieves better performance than the networks dedicated to valve stiction. We argue that two factors contributed to this result. The first factor is the diversity of the training data. The training data for LeNet-5 are generated 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 the LeNet-5. The proposed strategy derives the best performance from combining the two complementary data

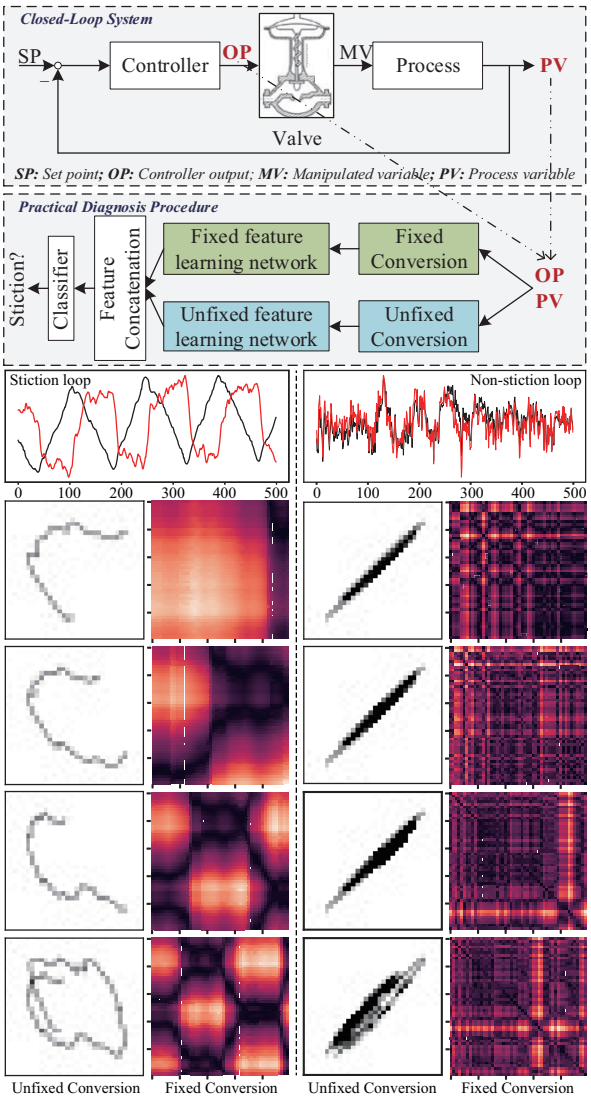


Fig. 8. Procedure for practical application and the typical input spectrum on four different timescales through the proposed data conversion methods.

conversion methods, the mixed feature learning stage and the fusion decision stage. In other words, two data conversion methods enhance the diversity of the data, mixed feature learning makes the model have better generalization ability, fusion ensemble decision-making method provides more robust and reliable results. The proposed strategy combines the advantages of traditional ML methods and DL-based methods to achieve the best performance, and the experiments confirmed the effectiveness of our strategy.

2) Discussion of Data Conversion: The discussion about the usage of data conversion methods is shown in Table IV. Net_{fix} and Net_{Unfix} represent the network with single fixed timescales conversion (III-A1) and the network with single unfixed timescales conversion (III-A2). The metrics of Net_{Unfix} are better than those of Net_{fix}, and the proposed ensemble method achieves the best results. Compared to the results of Net_{Unfix} in class non-stiction loops, the proposed ensemble method maintains the $F1$ and achieves a higher $F1$ in class stiction loops, proving the effectiveness of the complementary strategy of data conversion.

3) Discussion of number of Timescales: Considering feature fusion is an essential stage and the number of features is determined by the number of selected timescales in our proposed strategy, we discuss the influence of the number of timescales on the recognition accuracy. Intuitively, the more timescales we selected, the higher accuracy will achieve because more features are collected. The results are shown in Fig. 7. Through our experiments, we found that increasing the number of timescales is conducive to the improvement of recognition accuracy when the number of selected timescales is less than 4, but as the number of timescales continues to increase, the decrease in F1-score indicates that the performance of the approach will degrade. We argue that the diversity of features increases significantly when the number changes from 1 to 4, which is conducive to the model performance. When the number changes from 4 to 5 the redundant features are extracted, which leads to the deterioration of model performance. Overall, the multiple features ensemble strategy on industrial data is effective, but too many features will cause degradation of model performance due to feature redundancy.

D. Practical Application

The first three rows of Table V are the experimental results on the real hardware experimental system. In Table V, 0 and 1 represent non-stiction and stiction respectively. The valves within the experimental system are relatively new, and no stiction situations are reported from the operators. The decision results made by our strategy are non-stiction, which are consistent with the real labels.

In addition to the benchmark dataset and the real hardware experimental system, we further tested our proposed strategy in the real industrial environments. We provide the procedure for practical application, as shown in Fig 8. The whole application consists of a closed-loop system and the diagnosis procedure. OP and PV data are collected from the closed-loop system and they are taken as the inputs of the diagnosis framework. Through two data conversion methods mentioned in section III-A, the raw data are convert to images and matrices on different timescales. Then the trained feature learning networks are used to extract features from the transformed images and matrices. Finally, concatenating the features from different networks and using a traditional classifier, which achieving the final diagnosis of the closed-loop system. Moreover, we provide the typical input spectrum that includes the raw signals and the transformed signals via the proposed data conversion methods. The images in the columns on the left are the typical input spectrum of stiction loops, and the images on the right belong to non-stiction loop.

We collected industrial data from four valves and compared with five other methods, including LR, RF, SVM, Xgboost, and LeNet-5. We did not compare with BSD-CNN and SDN since the implementation codes of these two methods are not available. The results are shown in the last four rows of Table V. It can be seen that our strategy and LeNet-5 can correctly recognize the two stiction control loops (FIC2228 and F6304), but LeNet-5 give also two additional false positives (PIC23002

and FIC3107), so its global performance is worse. And our proposed strategy correctly recognizes the condition of six valves, which is better than other methods.

The last decade has witnessed the great success of DL, and this paper proposes two complementary data conversion methods along with a DL-based feature learning architecture to tackle industrial time series data for valve stiction detection. The experimental results on the benchmark dataset and the real industrial environments indicate that the proposed framework is comparative in this monitoring task.

V. Conclusion

This paper focused on multiple timescale feature learning of industrial control loop data and proposed a learning strategy for valve stiction detection. The key to the proposed strategy is combining two complementary data conversion methods, the mixed feature learning stage, and the fusion decision stage. Comparing the proposed CNN-based strategy with the traditional methods and DL-based methods showed that the proposed strategy could achieve higher and more reliable decision results. Moreover, the experiments on the real hardware system and actual industrial environment proved the effectiveness of the proposed strategy in practice. In our future work, the quantification evaluation of valve stiction will be further discussed. Additionally, the authors will try to extend the learning paradigm toward unsupervised learning since obtaining reliable industrial labeled data is not very easy in some industrial cases.

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] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke, "Deep learning for infrared thermal image based machine health monitoring," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 151–159, 2018.

[4] X. Li, X. Jia, Y. Wang, S. Yang, H. Zhao, and J. Lee, "Industrial remaining useful life prediction by partial observation using deep learning with supervised attention," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2241–2251, 2020.

[5] J. Wang, P. Fu, L. Zhang, R. X. Gao, and R. Zhao, "Multilevel information fusion for induction motor fault diagnosis," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2139–2150, 2019.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] L. Xie, X. Lang, A. Horch, and Y. Yang, "Online oscillation detection in the presence of signal intermittency," *Control Engineering Practice*, vol. 55, pp. 91–100, 2016.

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

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

[17] 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.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[20] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications-a survey of recent literature," *IEEE Transactions on Reliability*, pp. 1–21, 2019.

[21] J. Yuan and Y. Tian, "A multiscale feature learning scheme based on deep learning for industrial process monitoring and fault diagnosis," *IEEE Access*, vol. 7, pp. 151 189–151 202, 2019.

[22] W. Qiu, Q. Tang, J. Liu, and W. Yao, "An automatic identification framework for complex power quality disturbances based on multifusion convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3233–3241, 2020.

[23] 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.

[24] 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.

[25] M. Rezamand, M. Kordestani, R. Carriveau, D. S. Ting, and M. Saif, "An integrated feature-based failure prognosis method for wind turbine bearings," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1468–1478, 2020.

[26] W. Zhang, X. Li, and Q. Ding, "Deep residual learning-based fault diagnosis method for rotating machinery," *ISA Transactions*, vol. 95, pp. 295 – 305, 2019.

[27] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446–2455, 2019.

[28] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz inverse covariance-based clustering of multivariate time series data," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 215–223. [Online]. Available: https://doi.org/10.1145/3097983.3098060

[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 15, 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec. 2014.

[31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

Kexin Zhang received the B.S. and the M.S. degrees in engineering from Chine University of Geosciences, Wuhan, China, in 2016 and 2019, respectively. He is studying for the Ph.D degree in engineering with College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His major research interests include data mining and machine learning.

Yong Liu (M'11) received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2001 and 2007, respectively.

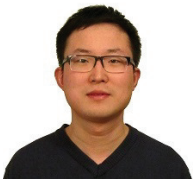
He is currently a Professor with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has authored or coauthored more than 30 research papers in machine learning, computer vision, information fusion, and robotics. His current research interests include machine learning, robotics vision, information processing, and granular computing.

Yong Gu received the B.S. degree in control theory and engineering and the Ph.D. degree in control theory and engineering from Zhejiang University, Hangzhou, China, in 1996 and 1999, respectively.

He is currently a Associate Professor with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has authored or coauthored more than 10 research papers in advanced process control, system identification, soft sensor, and artificial neural networks. His current research interests include machine learning, information processing, advanced process control and optimization.

Xiaojun Ruan received the B.S. degree in chemical engineering from Beijing University of Chemical Technology, Beijing, China, 2015 and the M.S. degree in chemical engineering with IT from Loughborough University, Leicestershire, U.K., 2016.

He is an algorithm engineer in Zhejiang Supcon Software Co., Ltd in Hangzhou, Zhejiang. His major research interests include data mining, process control and optimization.



Jiadong Wang received the B.S. and M.S. degrees in control engineering from the East China University of Sci. and Tech., Shanghai, China and the PhD degree in electrical and computer engineering from the University of Alberta, Edmonton, Canada, in 2005, 2008 and 2013, respectively.

He is currently a product manager at SUPCON, China, and he also works as an Adjunt Professor at the Nanjing Tech University, Nanjing, China. His majot research interests include PID and MPC control theory and applications.