Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 3197–3202

Industrial Fault Detection using Contrastive Representation Learning on Time-series Data

Kexin Zhang * **Rongyao Cai** * **Yong Liu** *,†

* *Institute of Intelligent Systems and Control, Zhejiang University, Hangzhou, 310027, China. (e-mail: yongliu@iipc.zju.edu.cn)*

Abstract: Deep learning (DL) has been known as one of the effective techniques for building data-driven fault detection methods. The successful DL-based methods require the condition that massive labeled data are available, but this is sometimes an inevitable obstacle in real industrial environments. As one of the solutions, autoencoders (AEs) are widely adopted since AEs can extract features from unlabeled data. However, some challenges in AE-based fault detection methods remain, such as the design of encoder architecture, the computational cost, and the usage of the limited labeled data. This paper proposes a new industrial fault detection method through learning instance-level representation of time-series based on the self-supervised contrastive learning framework (SSCL). The proposed method uses dilated-causal-convolution-based encoder-only architecture to extract the information from industrial time-series data. A new data augmentation method for time-series data is proposed based on the temporal distance distribution, which is used to construct positive pairs in SSCL. Moreover, the encoder is alternately trained by the new weighted contrastive loss and the traditional classification loss. Finally, the experiments are conducted on the industrial data set and a semi-physical system, showing the effectiveness of the proposed method.

Keywords: Industrial fault detection, time-series, contrastive learning, convolution-based encoder, semi-physical system.

1. INTRODUCTION

Industrial fault detection is essentially a task extracting underlying patterns from massive industrial data. Deep learning (DL) has been known as one of the effective techniques for building data-driven models. Unlike the traditional knowledge-based models in which the mechanism or system structures are required, the DL-based methods can extract information from the massive data with relatively little prior knowledge or without domain knowledge. Moreover, most DL-based methods deal with time-series data because most industrial data are collected from sensors that monitor process conditions at each time step, such as temperature, pressure, and other process variables. Therefore, DL-based fault detection methods for time-series data have attracted more attention.

In recent years, many fault detection methods have been developed based on DL techniques and their impressive performance attracts more researchers Wen et al. (2018); Qiu et al. (2020); Liu et al. (2020); Hu et al. (2020); Wen et al. (2022); Zhang et al. (2022). Generally, the above DL-based detection methods require to meet the condition that massive labeled data with fault information are available. However, this condition is sometimes an inevitable obstacle in practice because obtaining sufficient labeled data is time-consuming. Moreover, the trained DL-based models with insufficiently labeled data tend to produce unsatisfying detection results. Therefore, some unsupervised DL-based fault detection methods have been proposed. As one of the most representative unsupervised DL models, autoencoders (AEs) have strong abilities of learning representation using unlabeled data, which are more suitable for practical applications. For example, a denoising autoencoder (DAE) constructed by a fully connected neural network was proposed for fault detection of wind turbines (Jiang et al., 2018), a one-dimensional residual convolutional autoencoder (1-DRCAE) was developed for gearbox fault detection (Yu and Zhou, 2020), and a sliding-window convolutional variational autoencoder (SWCVAE) was proposed for anomaly detection of industrial robots (Chen et al., 2020).

Undoubtedly, these AE-based unsupervised methods reduce the dependence on labeled data. However, some challenges in AE-based fault detection methods remain. First, the encoders are not designed for time-series, thus losing contextual information along the time dimension. Especially in the AE based on the stacked fully connected layers (Jiang et al., 2018), the data at each time-step is assumed to be independent. Although the fully connected layers are replaced with the convolutional layers in (Yu and Zhou, 2020; Chen et al., 2020), the setup of the kernel width and the network depth for 1D-CNN-based methods and the inevitable time-series transformation for 2D-CNN-based methods are still important issues. Second, the traditional AE-based model consists of an encoder network and

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

† Corresponding author: Yong Liu (yongliu@iipc.zju.edu.cn)

10.1016/j.ifacol.2023.10.1456

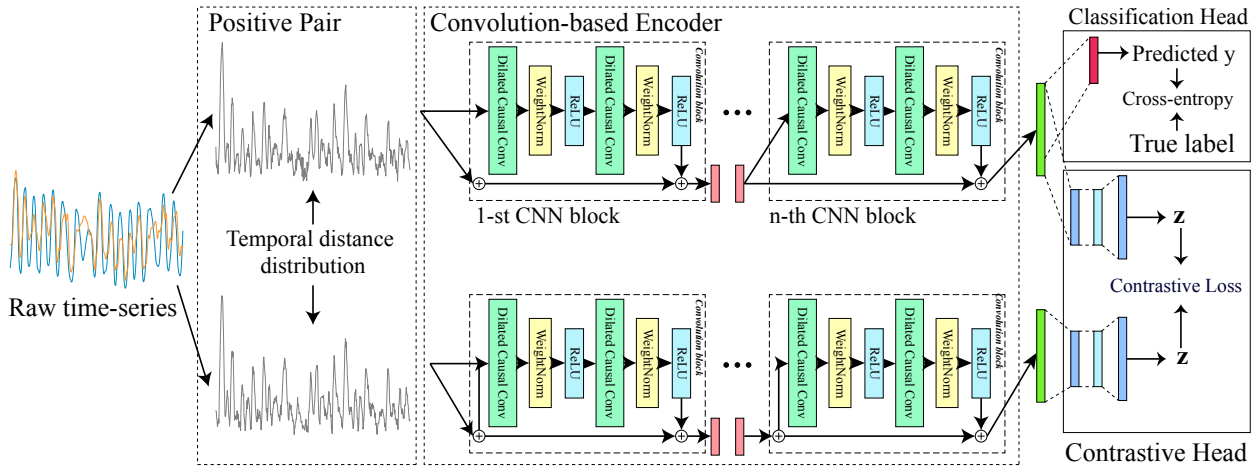

Fig. 1. The overall architecture of the proposed detection framework

a decoder network. These two networks need to be jointly trained using the reconstruction constraint, which causes a high computational cost, and the reconstruction constraint may fail to produce useful representations. Third, although collecting sufficient labeled data is difficult in real industrial environments, a small number of labeled data is accessible. Most AE-based methods drop the labeled data entirely, but the limited labeled data are also helpful.

This paper seeks to develop an encoder-only fault detection method when the limited labeled data are available. To this end, we propose a new industrial fault detection method through learning instance-level representations of time-series based on the contrastive learning framework. The main contributions of this paper are:

(1) The temporal distance distribution of the time-series data is first defined. It can be used as the data augmentation technique in the time-series contrastive learning framework.
(2) A new industrial fault detection framework is developed based on the convolution-based encoder, which combines the weighted contrastive loss and the traditional cross-entropy loss to train the encoder, making full use of the limited labeled data and the considerable unlabeled data.
(3) The extensive experiments are conducted on the industrial benchmark data set and the semi-physical system, demonstrating the effectiveness of the proposed framework.

The rest of this paper is organized as follows. Section 2 introduces the details of the proposed method. Section 3 provides the comprehensive experiments. Section 4 gives the experiments on the semi-physical system. Finally, conclusions are drawn in Section 5.

2. METHODOLOGY

2.1 Problem definition and detection framework

Given a set of multivariate time series (MTS) $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ of N instances, containing normal instances and faulty instances. The goal is to learn a nonlinear embedding function f_θ that maps each $\mathbf{X}_i$ to its instance-level representation $\mathbf{h}_i \in \mathbb{R}^K$ that best describes itself, where K is the dimension of representation vectors. Each time-series data is represented as $\mathbf{X}_i \in \mathbb{R}^{T \times D}$, where T is the number of measurements over time and D is the number of features.

The overall detection framework is shown in Fig. 1. It consists of three main components: (1) a new data transformation method for time-series data; (2) a convolution-based encoder; and (3) a classification head and a contrastive head. The data transformation component transforms the raw time-series into a temporal distance distribution series. It is a new method to build positive pairs in the contrastive learning framework. The convolution-based encoder maps the raw time-series and its augmented sample to the more suitable representations for the downstream detection tasks. The classification and contrastive heads are the positions where the cross-entropy loss and the improved contrastive loss are applied. The details of the three components are described as follows.

2.2 Convolution-based encoder

CNN has been demonstrated to be performant on time-series data. For example, combined with a conversion method converting signals into two-dimensional images, a new convolutional neural network (CNN) based on LeNet-5 was proposed for motor bearing detection (Wen et al., 2018). A multifusion CNN (MFCNN) that combines raw signal information and physical features was developed for intelligent identification of multiple power quality disturbances (Qiu et al., 2020). For motor fault detection, a multiscale kernel-based ResCNN (MK-ResCNN) architecture is proposed (Liu et al., 2020). In this paper, a dilated-causal-convolution-based encoder that is more suitable for time-series data is used as the feature extractor.

The encoder is based on stacks of dilated causal convolutions, which can map time-series of different lengths to representations of the same dimension. The causal operator ensures that there can be no information leakage from the future to the past, and the dilated operator is used to deal with the time-series tasks requiring longer history information. Following the work of (Yu and Koltun, 2016; Franceschi et al., 2019), the dilated convolutions have an

exponentially sizeable receptive field rather than a linear receptive field, and the dilated convolution operation D on time-step t is defined as

$$D(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{t-d \cdot i}, \tag{1}$$

where d is the dilation factor, k is the kernel size and $t - d \cdot i$ accounts for the involved time-steps of the past. Dilation is equivalent to introducing a fixed step between two adjacent filter taps. In addition to the dilated causal convolutions, each encoder layer includes weight normalizations and ReLUs. Residual connections are also used for solving the convergence problem. The output is then provided to a global max-pooling over the dimension of time, which aggregates all temporal information in a fixed-size representation vector, that is

$$\mathbf{h}_i = f_\theta(\mathbf{X}_i). \tag{2}$$

Then, $\mathbf{h}_i$ is sent to the projection head $g(\cdot)$ and the classification head $c(\cdot)$, which are denoted as

$$\mathbf{z}_i = g(\mathbf{h}_i) = W_g^{(2)}\text{ReLU}((W_g^{(1)}\mathbf{h}_i)), \tag{3}$$

$$\mathbf{o}_i = c(\mathbf{h}_i) = (W_c^{(1)}\mathbf{h}_i), \tag{4}$$

where $g(\cdot)$ maps the $\mathbf{h}_i$ to the space $(\mathbf{z}_i)$ where the contrastive constraint is applied. $\mathbf{o}_i$ is the output of the classification head. This paper builds both the projection head and the classification head using a multi-layer perceptron with one hidden layer. The predicted label of $\mathbf{X}_i$ is computed by

$$\tilde{y}_i = \text{argmax}(\text{Softmax}(\mathbf{o}_i)), \tag{5}$$

where Softmax$(\cdot)$ is the Softmax function that normalize the $\mathbf{o}_i$ to a probability distribution over predicted output classes. argmax$(\cdot)$ is an operation that returns the indices of the maximum value of Softmax$(\mathbf{o}_i)$.

2.3 Training encoder with weighted contrastive loss

Generally, the encoder is trained using the criterion called cross-entropy (CE) loss that is widely used in supervised classification. It is defined as

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\text{Softmax}(\mathbf{o}_i)), \tag{6}$$

where y_i is the true label and p_i is the predicted probability. A well-trained encoder requires a large amount of labeled training data. However, collecting such data is difficult in some industrial environments. Therefore, the additional contrastive constraint is added to the training process in this paper. Contrastive learning is a widely used method in a self-supervised context. It encourages learning representations by maximizing agreement between positive pairs via a contrastive loss in the latent space. The learning objective is expressed as

$$\begin{aligned}\mathcal{L}_c = &- \mathbb{E}_{\mathbf{z}_i,\mathbf{z}_i^{pos}}[\log(\mathcal{S}(\mathbf{z}_i, \mathbf{z}_i^{pos}))] \\ &- \mathbb{E}_{\mathbf{z}_i,\mathbf{z}_i^{neg}}[\log(1 - \mathcal{S}(\mathbf{z}_i, \mathbf{z}_i^{neg}))],\end{aligned} \tag{7}$$

where, $\mathcal{S}(\cdot)$ is the similarity between instance-level representation $\mathbf{z}_i$ and its positive and negative representations $\mathbf{z}_i^{pos}$ and $\mathbf{z}_i^{neg}$. The similarity is measured by

$$\mathcal{S}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|, \tag{8}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are ℓ_2 normalized representation vectors. However, (7) suffer from the problem of sampling bias because randomly drawing negative examples from a batch may be the samples that are similar to the positive samples (Tonekaboni et al., 2021). In our context, this can happen when the samples in a random batch belong to the same class. Therefore, the improved learning objective is expressed as

$$\begin{aligned}\mathcal{L}_c = &\mathbb{E}_{\mathbf{z}_i,\mathbf{z}_i^{pos}}[-\log(\mathcal{S}(\mathbf{z}_i, \mathbf{z}_i^{pos}))] \\ &- \mathbb{E}_{\mathbf{z}_i,\mathbf{z}_i^{neg}}[(1-w) \times \log(1 - \mathcal{S}(\mathbf{z}_i, \mathbf{z}_i^{neg})) \\ &+ w \times \log(\mathcal{S}(\mathbf{z}_i, \mathbf{z}_i^{neg}))],\end{aligned} \tag{9}$$

where w is the hyperparameter that controls the probability of the negative samples belonging to the positive samples. In this paper, the encoder is trained using $\mathcal{L}_c$ and $\mathcal{L}_s$, the final loss is defined as

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_c. \tag{10}$$

2.4 Positive pairs construction using temporal distance distribution transformation

Data augmentation is a widely used strategy for constructing positive pairs in contrastive learning. However, the time-series augmentation is still a challenge because the temporal information may be destroyed when using the traditional augmentation methods, such as window cropping or slicing (Wen et al., 2021). This paper proposed a new time-series augmentation method to construct positive pairs.

For a MTS $\mathbf{X}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,T}\}$ with T measurements, where $\mathbf{x}_{i,k} \in \mathbb{R}^D$ is a D-dimensional vector. A temporal distance matrix G is constructed and the elements in matrix describe the temporally-weighted similarities between time-steps. In other words, G is constructed by first computing the distances in the feature space and then modulating them by their temporal distance. The distance in feature space is defined as

$$G_i^f(k, k^{'}) = 1 - \mathcal{S}\langle \mathbf{x}_{i,k}, \mathbf{x}_{i,k'} \rangle, \ k, k' = 1, 2, \ldots, T, \tag{11}$$

where $G_i^f(k, k^{'})$ represents the feature space distance between time-steps k and $k^{'}$. $\mathcal{S}\langle \mathbf{x}_{i,k}, \mathbf{x}_{i,k'} \rangle$ denotes the cosine distance as shown in (8). Although (8) can measure the similarity between any two time steps, the temporal positions of them is not considered. Based upon the intuition that two consecutive (or very close) time steps are more likely to be similar (a small distance), the temporal factor is defined as

$$G_i^t(k, k^{'}) = \exp(\mu \cdot |k - k^{'}|/T), \tag{12}$$

where μ is a parameter to control the sensitivity of the time difference between the time-steps, the term $|k-k^{'}|/T$ provides a weighing mechanism relative to the time-series length. Then the modulated distance between any two time steps is computed as follows

$$W_i(k, k^{'}) = G_i^f(k, k^{'}) \cdot G_i^t(k, k^{'}), \ k, k' = 1, 2, \ldots, T. \tag{13}$$

$W_i(k, k') \in \mathbb{R}^{T \times T}$ is the temporal distance of time-step k and k'. The temporal distance distribution of k-th time-step is defined as

$$\hat{\mathbf{X}}_{i,k} = \frac{1}{T}\sum_{k'=1}^{T} W_i(k, k'), \tag{14}$$

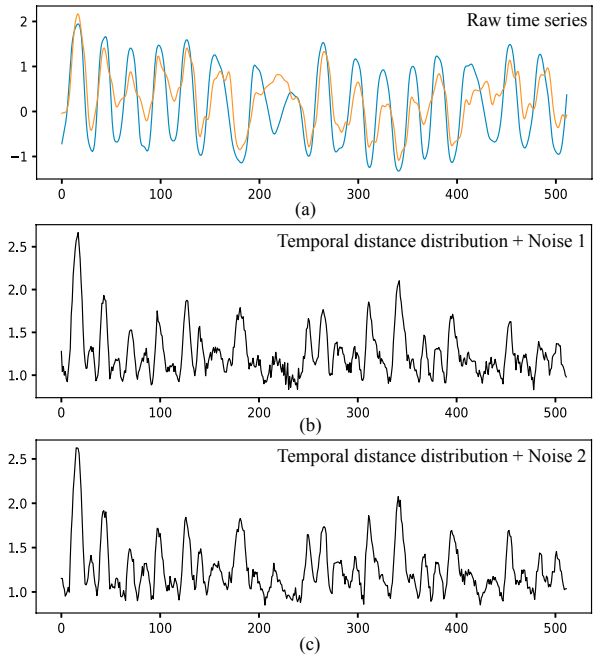

Fig. 2. Temporal distance distribution Transformation. (a) Raw time-series. (b) Temporal distance distribution + $\mathcal{N}_1$. (c)Temporal distance distribution + $\mathcal{N}_2$.

where $\mathbf{X}_{i,k} \in \mathbb{R}^{T\times 1}$. $\mathbf{X}_{i,k}$ is the average temporal distance between time-step k and other time-steps in the same series. Then the noise is added to the temporal distance distribution, i.e.,

$$\hat{\mathbf{X}}_i^{\mathcal{N}_1} = \hat{\mathbf{X}}_i + \mathcal{N}_1(0, 0.03), \quad \hat{\mathbf{X}}_i^{\mathcal{N}_2} = \hat{\mathbf{X}}_i + \mathcal{N}_2(0, 0.03), \quad (15)$$

where $\mathcal{N}_*(mean, std)$ is the operator that produces the noise drawn from normal distributions. In our method, $\hat{\mathbf{X}}_i^{\mathcal{N}_1}$ and $\hat{\mathbf{X}}_i^{\mathcal{N}_2}$ construct a positive pair. An example is shown in Fig 2.

3. EXPERIMENTS ON BENCHMARK DATA SET

3.1 Benchmark data set

The data set for evaluating the proposed method is the International Stiction Data Base (ISDB), which is supported by (Jelali and Huang, 2010) and is a well-known benchmark for validation of novel techniques concerning control loop performance assessment. These loops were collected from various process industries, including chemical plants (CHEM), pulp and paper mills (PAP), buildings (BAS), mining (MIN), and power plants (POW). In this paper, we select 59 loops as training data and 26 loops as test data, and the length of time-series is 600 timesteps. The main information of the test loops is described in Table 1, where *Tem*, *Fic*, *Pre*, *Lev*, *Con*, *Ana* denote the temperature, flow, pressure, level, concentration, and analyzer control, respectively.

3.2 Detection results

In the first experiments, the detection results that use all training data are provided, which are also used for comparison with other fault detection approaches. Specifically, 59 loops (20 stiction and 39 no-stiction samples) are available for training networks.

Table 1. Test Loops in the ISDB Data Set

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	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*	Non-stiction
CHEM 11	*Fic*	Stiction	MIN 1	*Tem*	Stiction
CHEM 12	*Fic*	Stiction	PAP 2	*Fic*	Stiction
CHEM 13	*Ana*	Faulty sensor	PAP 4	*Con*	Deadzone
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*	Non-stiction

Table 2. Detection results for 26 loops

Loop	Result	Loop	Result	Loop	Result
CHEM 1	✓	CHEM 11	✓	CHEM 26	✓
CHEM 2	✓	CHEM 12	✓	CHEM 29	✓
CHEM 3	✓	CHEM 13	✓	CHEM 32	✓
CHEM 4	✓	CHEM 14	×	CHEM 33	✓
CHEM 5	✓	CHEM 16	✓	CHEM 34	×
CHEM 6	✓	CHEM 23	✓	CHEM 58	✓
CHEM 10	✓	CHEM 24	✓	MIN 1	✓
PAP 2	✓	PAP 4	×	PAP 5	✓
PAP 7	✓	PAP 9	×	–	–

Table 3. Comparison with other stiction detection methods

Method	Best accuracy	NOT tested
Higher-order statistics	19/24	2
Statistics-based method	16/25	1
Relay-based method	17/26	0
Curve fitting method	12/25	1
Waveform shape analysis	11/26	0
PSD/ACF	18/26	0
Peak slope method	14/25	1
Zone segmentation	15/25	1
BSD-CNN	20/26	0
D-value ANN	19/24	2
Ours	**22/26**	0

The detection results presented in Table 2. Our method gives 22 correct detection out of 26 test loops, except CHEM 14, PAP 4, PAP 9, and CHEM 34. The accuracy score is 22/26=0.846. Therefore, the effectiveness of the proposed fault detection method is initially shown. To further highlight the capability of the proposed method, the accuracy score of the proposed method with other detection methods is presented in Table 3.

3.3 Comparison results

In our experiments, eight traditional methods and two DL-based methods for the ISDB data set are collected from (Jelali and Huang, 2010; Bacci di Capaci and Scali, 2018; Mohd Amiruddin et al., 2019; Kamaruddin et al., 2020). Eight traditional methods can be broadly classified into four categories: cross-correlation-function-based, limit-cycle-patterns-based, nonlinearity-detection-based, and waveform-shape-based. Two DL-based methods were developed based on a convolutional network and

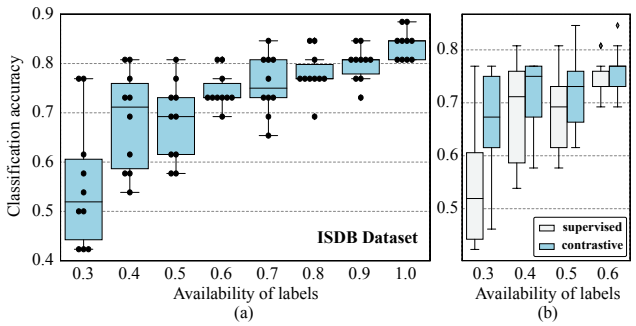

Fig. 3. Detection accuracy under different availabilities of labels.

a multilayer perceptron. The comparison results are shown in Table 3. It can be seen that only five methods can be performed on all 26 test loops, which means our proposed method applies to a wider range of processes. Besides, the methods proposed in (Mohd Amiruddin et al., 2019; Kamaruddin et al., 2020) are DL-based methods, and the accuracy scores are lower than our method. It shows that our proposed approach outperforms state-of-the-art DL-based stiction detection methods. Our method has the highest accuracy score compared with the considered methods, i.e., 22/26 (0.846).

3.4 Fault detection with the limited labeled data

The performance of the DL-based fault detection methods depends on the number of labeled training data. The following experiments show that the proposed method achieves higher accuracy scores than the purely supervised training manner when only the limited training data are available.

Availability of labels In this part, the model is first trained with different availability of training data, and only classification loss is considered. The results and statistics are presented in Fig. 3(a) and Table 4. For the ISDB data set, we set 0.3, 0.4, 0.5, and 0.6 as the proportion of the labeled data, and the corresponding numbers of training samples are 17, 23, 29, and 35. It is easy to see that the accuracy scores improve with the increased labeled data. Based on the statistics results in Table 4, the standard deviations decrease at the same time. We argue that for the ISDB data set, when the availability of labels is equal to 0.3, 0.4, 0.5, and 0.6, it is difficult to achieve acceptable accuracy scores through a single classification loss.

Effectiveness of weighted contrastive loss In order to show the effectiveness of the additional contrastive constraint, the experiments were conducted when the limited training data were available. The classification results and the statistics are shown in Fig. 3(b) and Table 4. It demonstrates that the additional contrastive constraint improves the accuracy scores when the limited labeled data are available.

4. EXPERIMENTS ON SEMI-PHYSICAL SYSTEM

A semi-physical system is further used to show the effectiveness of our method. The hardware system consists of a liquid level loop and two flow loops. The flow chart and

Table 4. Statistics of the full supervised training mode and the supervised-contrastive training mode with different proportion of labeled data

Mode	*Stat*	Availability of labels			
		0.3	0.4	0.5	0.6
CE only	mean	0.5539	0.6846	0.6807	0.7461
	std	0.1309	0.1006	0.0811	0.0371
CE + Contrastive	mean	0.6538	0.7153	0.7230	0.7615
	std	0.1087	0.0729	0.0720	0.0436

Table 5. Experimental Results on semi-physical system and Industrial Environments

Loop	Malfunction	Detection	Correct?
FIC201	Normal	Non-stiction	YES
FIC202	Normal	Non-stiction	YES
LIC201	Normal	Stiction	NO
PIC23002	External disturbance	Non-stiction	YES
FIC3107	Normal	Non-stiction	YES
F6304	Stiction	Stiction	YES

the experimental system are shown in Figure. 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. Moreover, three real control loops, PIC23002, FIC3107, and F6304, are collected from the real industrial environments, in which PIC23002 is a pressure control loop, and it is affected by the unknown external disturbances, FIC3107 is a flow control loop, and its state is normal, and F6304 are flow control loops, and they were recorded as stiction. The raw time-series recordings of these loops are shown in Fig 5.

The experimental results on the semi-physical system and the real industrial environments are shown in Table 5. The three valves in this hardware system are relatively new, so there are no stiction records. The detection results made by our method are stiction for two flow control loops, which are consistent with the real conditions. However, false detection occurs on loop LIC201. The detection results of the other three collected industrial loops are shown in the last three rows of Table V. It can be seen that our method gives the correct detection on these three control loops. Although loop PIC23002 was affected by unknown external disturbances, our method also provides a reliable detection.

5. CONCLUSION

This paper developed a new industrial fault detection framework based on the proposed instance-level feature learning method, which can be added to the traditional supervised detection framework. The feature learning method is based on the contrastive learning framework, but the temporal distance distribution transformation was proposed to construct positive pairs for time-series data.

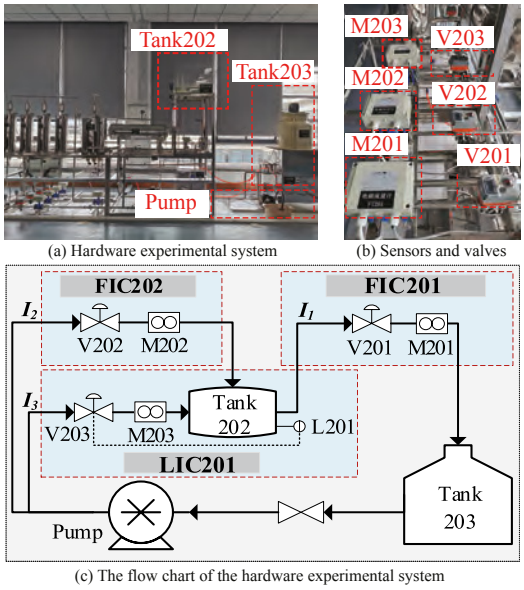

Fig. 4. The semi-physical system.

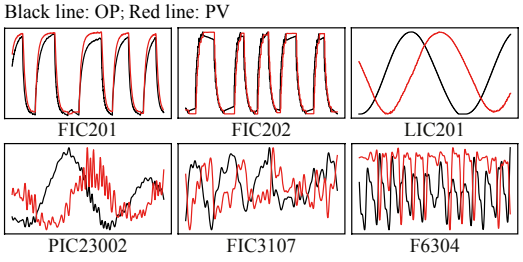

Fig. 5. Raw recordings for six real loops.

Additionally, the weighted contrastive loss was used to alleviate the sampling bias problem. The proposed detection framework is performed on the benchmark data set ISDB, the semi-physical system, and three real industrial control loops. The extensive experimental results indicated that the proposed framework achieves better results than state-of-the-art methods and improves the detection results when labeled data are insufficient.

REFERENCES

Bacci di Capaci, R. and Scali, C. (2018). Review and comparison of techniques of analysis of valve stiction: From modeling to smart diagnosis. *Chemical Engineering Research and Design*, 130, 230–265. doi: https://doi.org/10.1016/j.cherd.2017.12.038.

Chen, T., Liu, X., Xia, B., Wang, W., and Lai, Y. (2020). Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. *IEEE Access*, 8, 47072–47081. doi: 10.1109/ACCESS.2020.2977892.

Franceschi, J.Y., Dieuleveut, A., and Jaggi, M. (2019). Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, volume 32.

Hu, Z.X., Wang, Y., Ge, M.F., and Liu, J. (2020). Data-driven fault diagnosis method based on compressed sensing and improved multiscale network. *IEEE Transactions on Industrial Electronics*, 67(4), 3216–3225. doi: 10.1109/TIE.2019.2912763.

Jelali, M. and Huang, B. (2010). *Detection and diagnosis of stiction in control loops: state of the art and advanced methods*. London: Springer-Verlag. doi:10.1007/978-1-84882-775-2.

Jiang, G., Xie, P., He, H., and Yan, J. (2018). Wind turbine fault detection using a denoising autoencoder with temporal information. *IEEE/ASME Transactions on Mechatronics*, 23(1), 89–100. doi: 10.1109/TMECH.2017.2759301.

Kamaruddin, B., Zabiri, H., Mohd Amiruddin, A., Teh, W., Ramasamy, M., and Jeremiah, S. (2020). 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*, 87(4), 1–16. doi: https://doi.org/10.1016/j.jprocont.2020.01.001.

Liu, R., Wang, F., Yang, B., and Qin, S.J. (2020). Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions. *IEEE Transactions on Industrial Informatics*, 16(6), 3797–3806. doi:10.1109/TII.2019.2941868.

Mohd Amiruddin, A.A.A., Zabiri, H., Jeremiah, S.S., Teh, W.K., and Kamaruddin, B. (2019). Valve stiction detection through improved pattern recognition using neural networks. *Control Engineering Practice*, 90, 63–84. doi: https://doi.org/10.1016/j.conengprac.2019.06.008.

Qiu, W., Tang, Q., Liu, J., and Yao, W. (2020). An automatic identification framework for complex power quality disturbances based on multifusion convolutional neural network. *IEEE Transactions on Industrial Informatics*, 16(5), 3233–3241. doi:10.1109/TII.2019.2920689.

Tonekaboni, S., Eytan, D., and Goldenberg, A. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations*.

Wen, L., Li, X., Gao, L., and Zhang, Y. (2018). A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*, 65(7), 5990–5998. doi: 10.1109/TIE.2017.2774777.

Wen, L., Xie, X., Li, X., and Gao, L. (2022). A new ensemble convolutional neural network with diversity regularization for fault diagnosis. *Journal of Manufacturing Systems*, 62, 964–971. doi: https://doi.org/10.1016/j.jmsy.2020.12.002.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. (2021). Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 4653–4660. doi:10.24963/ijcai.2021/631.

Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*.

Yu, J. and Zhou, X. (2020). One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. *IEEE Transactions on Industrial Informatics*, 16(10), 6347–6358. doi: 10.1109/TII.2020.2966326.

Zhang, K., Liu, Y., Gu, Y., Ruan, X., and Wang, J. (2022). Multiple-timescale feature learning strategy for valve stiction detection based on convolutional neural network. *IEEE/ASME Transactions on Mechatronics*, 27(3), 1478–1488. doi:10.1109/TMECH.2021.3087503.