12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Multi-Operating Condition Time Series Anomaly Detection Based on Domain Adaptation

Xiaojun Yang[1], Zhihua Xiong[1], Hao Ye[1], Tongshuai Zhang[1,2]

[1] *Department of Automation, Tsinghua University, Beijing, 100084, China*
(*e-mail: yangxj21@mails.tsinghua.edu.cn*, *zhxiong@tsinghua.edu.cn*, *haoye@tsinghua.edu.cn*).

[2] *Beijing Hyetec Technology Co., Ltd., Beijing 100085, China* (*Email:ztnone@qq.com*).

Abstract: Learning from time series data in industrial scenarios enables the detection and classification of anomalies or faults in equipment and production processes. In industrial settings, variations in production equipment parameters or raw materials lead to changes in production operating conditions, resulting in the multi-operating condition characteristics of the data and placing higher requirements on anomaly detection models. This paper introduces domain adaptation and contrastive learning methods for multi-operating condition time series data, and designs an end-to-end model structure to enhance the performance of time series anomaly detection. The objective loss function incorporates the maximum mean discrepancy (MMD) and contrastive loss functions. The proposed approach is validated and analyzed on a simulated dataset.

Keywords: multi-operating condition; time series; anomaly detection; domain adaptation

1. INTRODUCTION

In the era of digitization and automation, data generated from industrial processes contains valuable information, which can be used to predict production line yields and detect anomalies in industrial equipment or processes. Anomalies in production processes refer to situations that deviate from normal production conditions, such as equipment faults or deviations in equipment parameter settings. It is crucial to identify and detect potential anomalies and faults as early as possible.

Time series data is a commonly found data format in factory data, in addition to image data. The temporal dependency of time series data requires different analysis approaches compared to image and text data. Time series data is an important research focus in academia, including prediction, anomaly detection, and classification. In industrial scenarios, time series data presents complex and variable conditions, such as unlabeled data, multi-operating condition, and known source domain data or target domain data with unknown labels.

To address these challenges, we introduce domain adaptation methods from transfer learning (Weiss et al., 2016). In this paper, we focus on studying time series data with multi-operating conditions. Drawing inspiration from domain adaptation techniques in the visual domain and incorporating contrastive learning algorithms, an end-to-end deep learning model is designed for anomaly detection in multi-operating conditions of time series data. The proposed model is validated and analyzed using a simulated dataset. The key contribution of this work is twofold.

1) A Cross Domain-Contrastive Learning (CdCl) method is proposed to solve anomaly detection under multi-operating conditions by training a model based on known information and performing fault diagnosis on unlabeled data in the target domain.

2) The proposed model constructs positive sample pairs of time series through different kinds of encoders, and retains only domain adaptive label classifiers while removing domain classifiers, which can ensure performance while reducing model complexity.

The paper is structured as follows: Section 2 provides a detailed explanation of the methodology, Section 3 presents an analysis of the experiment conducted on an industrial process dataset, and Section 4 concludes the paper.

2. METHODS

2.1 Problem Definition

Anomaly refers to behavior patterns in data that significantly deviate from normal conditions, and detecting these inconsistent patterns is known as anomaly detection (Chandola et al., 2009). In general, anomaly detection in the industrial domain often refers to fault diagnosis. Multi-operational conditions are crucial scenarios faced by industrial data, where variations in raw materials and external environmental factors can cause abrupt changes in the production process, resulting in multimodal characteristics (Liu and Qin, 2016).

Unlike other research fields in machine learning, the term "modality" here does not refer to diverse patterns of different data representations such as images or videos, but is defined as changes in operating points. To avoid confusion, we refer to these variations in industrial process operating points as multi-operational conditions. Under different operational conditions, data may exhibit multimodal distributions, and fault detection models built based on one condition may exhibit higher false alarm rates in other conditions (Bi et al., 2022).

Although deep learning methods have been successfully applied in the field of fault diagnosis (Saeed et al., 2020), such approaches impose higher requirements on training data.

Firstly, an abundant and high-quality dataset is required, and secondly, the training and testing data should come from the same operational conditions. Only by simultaneously satisfying these conditions can the effectiveness of the fault diagnosis model be ensured. However, in practical situations, due to various environmental factors, the collected data often comes from different operational conditions, and there is a significant class imbalance in the collected data.

Transfer learning can guide deep learning models to learn domain-invariant feature representations, i.e., narrowing the data distribution differences among different domains to adapt to the differences between source domain and target domain. For instance, within the realm of object recognition, a self-supervised domain adaptation method is proposed to evaluate the multi-source domain adaptation dataset (Xu et al., 2019), which consists of 7 object categories and 4 domains (i.e. photos, art paintings, cartoons, and sketches). One of the domains is selected as the target domain for training the model, while the other three domains are used as the source domain.

Different datasets from various operating conditions can be considered as different domains, but the objective of anomaly detection still remains consistent. The methods mentioned above can also be applied to time series data, offering a solution for fault detection in scenarios involving multiple operating conditions (Wu and Zhao, 2020).

In transfer learning, we treat the known labeled data from the operating conditions as the source domain, and the unlabeled data as the target domain. Therefore, our problem and its objective are defined as: provided with labeled data from the source domain and unlabeled data from the target domain, we aim to train a model that brings the source and target domain data closer together in the feature space. Consequently, the classifier trained on the source domain for fault classification should perform effectively on the target domain.

2.2 Transfer Learning

Domain: It refers to the primary focus of learning, which includes two main components: the feature space X and the probability distribution $P(X)$ that generates the data (Pan and Yang, 2009). Typically, D is used to represent a domain, while P represents a probability distribution. In the context of transfer learning, two fundamental domains are involved specifically: the source domain D^s and the target domain D^t.

Task: It refers to the learning objective, which consists of two main components: the label space Y and the function $f(\cdot)$ that associates inputs with their corresponding labels (Pan and Yang, 2009). Y is commonly used to represent the label space, and $f(\cdot)$ or $P(y|x)$ represents a learning function. The class spaces for the source domain and the target domain are denoted as Y^s and Y^t, respectively.

$$T^s = \{Y^s, P_s(y|x)\}, \quad T^t = \{Y^t, P_t(y|x)\}. \qquad (1)$$

If two domains are consistent, i.e., $D^s = D^t$ and $T^s = T^t$, traditional machine learning methods can be adopted to address this issue, where D^s serves as the training set and D^t serves as the testing set. However, if $D^s \neq D^t$ or $T^s \neq T^t$, a model trained on D^s may not perform well on D^t. When there exists a correlation between the source domain and the target domain, the knowledge and information gathered from the source domain can be utilized to improve the predictive capabilities of the target task learning function $P_t(y|x)$. This process is known as transfer learning (Csurka, 2017).

2.3 Domain Adaptation

Domain adaptation (DA) is one of the most prominent problems in transfer learning, which aims to learn classifiers for unlabeled or unseen data in the target domain using labeled data from one or multiple related source domains (Csurka, 2017). The general assumption in domain adaptation is:

$$P_s(x) = P_t(x), \quad P_s(y|x) \neq P_t(y|x). \qquad (2)$$

The objective of domain adaptation is to identify common features across different distributions and transfer invariant information (Csurka, 2017). Domain adaptation are commonly observed in the field of image analysis, such as images from different scenes or with different styles. Time series data can also exhibit multiple sources, such as varied motor loads or different operational settings in a manufacturing process.

The most commonly used distance measure in domain adaptation is maximum mean discrepancy (MMD) (Gretton et al., 2007). It calculates the distance between data from the source domain D^s and the target domain D^t after mapping them to a reproduced kernel *Hilbert* space. The detailed definition of MMD is as follows (Gretton et al., 2007):

$$MMD(X^s, X^t) = \left\| \frac{1}{n^s} \sum_{i=1}^{n^s} \phi(x_i^s) - \frac{1}{n^t} \sum_{j=1}^{n^t} \phi(x_j^t) \right\|_{\mathcal{H}} \qquad (3)$$

Where $\mathcal{H}$ is the kernel *Hilbert* space, ϕ is the mapping function typically using a kernel function, $X, Y \to \mathcal{H}$, x_i^s represents the source domain data, n^s is the number of source domain data, x_j^t represents the target domain data, and n^t is the number of target domain data, respectively.

MMD applies a mapping technique that transforms the data into a reproduced kernel *Hilbert* space and calculates the difference between the source and target domain data in this space. This difference is then used as part of the loss function. This method aims to minimize the discrepancy between the source and target domains, bringing them closer together in the same feature space. This allows classifiers trained on the source domain data to be effective on the target domain data as well. A model, called as DAFD (Lu et al., 2017), is proposed to address cross-domain learning in the field of fault diagnosis. Specifically, in most domain adaptation models, two types of classifiers are used, named a label classifiers and domain classifiers. In our model, we removed the domain classifier because we hope to achieve a good fault classifier and reduce the complexity of the model.

2.4 Contrastive Learning

Contrastive learning emphasizes extracting shared features among analogous samples and discerning disparities between distinct categories.

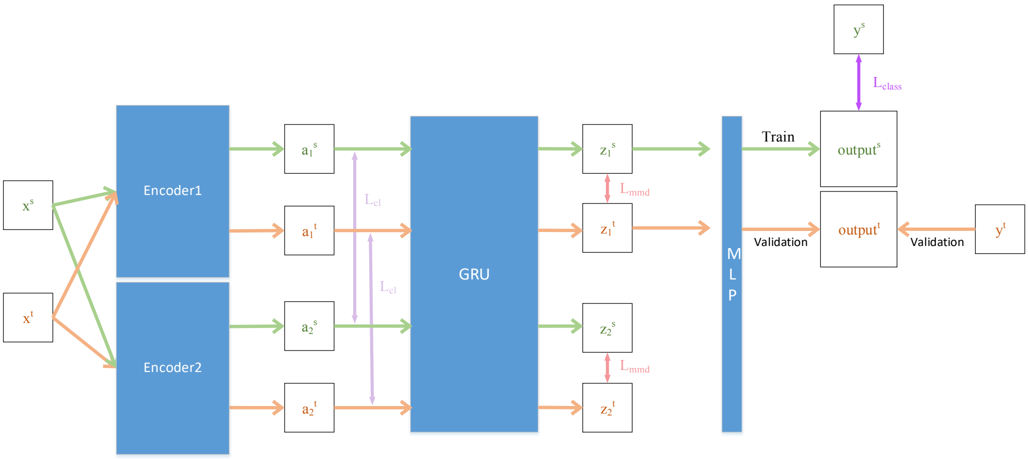

Figure 1. Model Structure

For an input sample x, there exist similar samples $x+$ and dissimilar samples $x-$. The aim of contrastive learning is to learn an encoder f that can minimize the distance between positive samples while maximizing the distance between x and its negative samples. Video domain adaptation tasks were studied by using a cross-modal contrastive learning framework (Kim et al., 2021). Two objective losses, called as cross-modal contrastive loss and cross-domain contrastive loss, were used to learn better domain adaptation methods for feature representation. Contrastive learning was also applied to temporal data, and various novel data augmentation techniques were designed. SimCLR-TS model was proposed for contrastive learning on time series data (Pöppelbaum, 2022).

The key to contrastive learning lies in constructing positive and negative sample pairs. To create positive sample pairs in the context of time series data construction, there are three common methods, including sequence consistency, time consistency, and transformation consistency. Transformation consistency involves enhancing the time series through different transformations, such as scaling and permutation. In the visual domain, rotation and cropping are commonly used methods. In the case of time series data, various data augmentation techniques were proposed (Pöppelbaum, 2022), but the left-right flipping for time series data cannot be explained in terms of the physical meaning of the time series. We adopt different encoder architectures, namely linear layer encoding and gate recurrent unit (GRU) temporal feature encoding, to capture features from various viewpoints of the input data and form positive sample pairs.

2.5 Model Structure

The proposed method is an end-to-end model called as cross domain-contrastive learning (CdCl). The model architecture is shown in Figure 1, and the algorithm flow is described in *Algorithm* 1.

The core idea of the model is to encode the original data using different encoders to obtain positive sample pairs for contrastive learning, which serves as the first step of feature extraction. The data is then fed into the GRU layer of a time series deep learning model to reduce the distance between the source domain and target domain in the feature space through cross-domain learning. In this step, we use the MMD as part of the loss function. In the classifier part, we construct a two-layer fully connected network to classify faults based on the learned features from the source domain, using cross-entropy loss as the loss function. The trained model is then used to classify faults in the target domain and evaluate the classification performance on the source domain data.

In Figure 1, X^d represents the data, ${a_1}^d$ represents the encoded data, ${z_1}^d$ represents the feature vectors, $output^d$ represents the predicted labels, and y^d represents the true labels, where the superscript $d = s$ or t indicates the source domain and target domain, respectively. L_{cl} denotes the contrastive learning loss, L_{cd} denotes the domain adaptation loss, and L_{class} denotes the cross-entropy loss for the source domain labels. The overall loss function L_{total} of the model is defined as follows:

$$L_{total} = L_{class} + \alpha_{cd} * L_{cd} + \alpha_{cl} * L_{cl} \tag{4}$$

$$L_{class} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{nc}\log(p_{nc}) \tag{5}$$

$$L_{cd} = \frac{1}{\mathrm{m}}\sum_{\sigma=1}^{m}\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}} k_{\sigma}^{rbf}\,(x_i^s, x_i^t)$$
$$= \frac{1}{\mathrm{m}}\sum_{\sigma=1}^{m}\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}} \exp(-\left\|x_i^s - x_i^t\right\|_2/2\sigma^2) \tag{6}$$

$$L_{cl} = -log\frac{\exp(q*k_+/\tau)}{\sum_{i=0}^{k}\exp(q*k_i/\tau)} \tag{7}$$

Where α_{cd} represents the weight of the domain adaptation loss function, and α_{cl} represents the weight of the contrastive learning loss function, N is the number of data, C is the number of total classes, y_{nc} is the label, p_{nc} is the probability, k_{σ}^{rbf} is *Gaussian* kernel, q *and* k_+ are positive pairs, respectively.

The accuracy of the model in the source and target domains is defined as：

$$src\_acc = \frac{Number\ of\ correctly\ classified\ source\ domain\ data}{Number\ of\ source\ domain\ data} \tag{8}$$

$$tar\_acc = \frac{Number\ of\ correctly\ classified\ target\ domain\ data}{Number\ of\ target\ domain\ data} \tag{9}$$

The procedure of CdCl model is described in detail as follows.

Algorithm 1: CdCl Model

Train

Input: labeled source data $\{X^s, Y^s\}$, and unlabeled target data $\{X^t\}$, α_{cd}, α_{cl};
Output: trainded model
for epoch = 1 to N **do**
for each mini-batch **do**
① get encoder data ${a_1}^s$,${a_2}^s$ of x^s by Encoder1 and Encoder 2 respectively, ${a_1}^t$,${a_2}^t$ of x^t, calculate the contrastive learning loss L_{cl};
② get feature data ${z_1}^s$, ${z_2}^s$, ${z_1}^t$, ${z_2}^t$ by GRU-layer, calculate the mmd loss L_{cd};
③ get prediction $output^s$ and calculate cross entropy loss L_{class} between $output^s$ and y^s;
④ Back-propagate L_{total} and update model parameters

$$L_{total} = L_{class} + \alpha_{cd} * L_{cd} + \alpha_{cl} * L_{cl}$$

end for
end for

Test

Input: source data$\{X^s\}$, target data $\{X^t\}$;
Output: source data label $\{Y^s\}$ and src_acc, target data label $\{Y^t\}$ and tar_acc

$$src\_acc = \frac{N_{correct\ prediction\ of\ X^s}}{N_{total\ source\ data}}$$

$$tar\_acc = \frac{N_{correct\ prediction\ of\ X^t}}{N_{total\ target\ data}}$$

3. EXPERIMENT

3.1 TE Process Dataset

The Tennessee Eastman (TE) process (Bathelt et al., 2015) is a dataset commonly used to validate the effectiveness of anomaly detection algorithms. The variables in this process exhibit evident coupling and nonlinear relationships, and the process response properties change over time.

The revised TE model is used here to generate data with multi-operating conditions (Bathelt et al., 2015). The TE process consists of 53 process monitoring variables, including 12 manipulated variables and 41 variables related to monitoring. The monitoring variables can be further divided into 22 continuous variables and 19 analytical variables. The TE process has a basic operation case and six operation modes. Due to difficulties in generating data for mode 2 and mode 5, this study only uses data from the basic case (called as mode 0), mode 1, mode 3, mode 4, and mode 6.

In the simulation, the sampling interval was set to 36 seconds (i.e., 100 sampling points per hour). We generated 1000 samples for each class in all five modes. Under each mode, normal data and 21 types of fault data were generated. A detailed description of the modes has been shown in Table 1. In this study, we focused on anomaly detection and fault classification for several fault types (1,2,10,12,13,20) and normal data (labeled as 0). We classified each fault separately from normal data and taked the average result. The normal and fault data in different modes are shown in Figure 2.

Our experiment was set up with known data and labels from multiple source domains. We aimed to perform fault classify-cation on a single unknown labeled target domain data, and evaluated the classification performance of the statistical model on both the source and target domains.

Table1. Optimal steady-state values in multi-operating modes

Manipulated variables(u)	Base case	Mode1 (50/50)	Mode3 (90/10)	Mode4 (50/50)	Mode6 (90/10)
D Feed, %	63.05	62.94	89.13	*100.00*	*100.00*
E Feed, %	53.98	53.15	8.38	86.72	9.44
A Feed, %	24.64	26.25	19.11	49.48	21.54
A+C Feed, %	61.30	60.57	51.37	96.60	57.64
Recycle value, %	22.21	*1.00*	77.62	*1.00*	71.17
Purge value, %	40.06	25.77	9.50	48.74	10.65
Separator value, %	38.10	32.27	29.15	60.96	32.69
Stripper value, %	46.53	46.44	39.43	74.52	44.25
Steam value, %	47.45	*1.00*	*1.00*	*1.00*	*1.00*
Reactor coolant, %	41.11	35.99	35.55	60.79	40.54
Condenser coolant, %	18.11	12.43	99.00	35.53	99.00
Agitator speed, %	50.00	*100.00*	*100.00*	*100.00*	*100.00*

Table2. Performance of the model under different parameters

Source	Target	α_{cd}	α_{cl}	*src_acc*	*tar_acc*
1,3,4,6	0	0	0	0.95	0.54
1,3,4,6	0	0	1	0.94	0.65
1,3,4,6	0	1	0	0.95	0.86
1,3,4,6	0	1	1	0.93	0.88
0,3,4,6	1	0	0	0.95	0.68
0,3,4,6	1	0	1	0.94	0.73
0,3,4,6	1	1	0	0.94	0.90
0,3,4,6	1	1	1	0.93	0.90
0,1,4,6	3	0	0	0.95	0.81
0,1,4,6	3	0	1	0.94	0.80
0,1,4,6	3	1	0	0.94	0.86
0,1,4,6	3	1	1	0.93	0.90
0,1,3,6	4	0	0	0.94	0.64
0,1,3,6	4	0	1	0.93	0.70
0,1,3,6	4	1	0	0.94	0.89
0,1,3,6	4	1	1	0.92	0.90
0,1,3,4	6	0	0	0.94	0.72
0,1,3,4	6	0	1	0.94	0.78
0,1,3,4	6	1	0	0.94	0.89
0,1,3,4	6	1	1	0.93	0.86

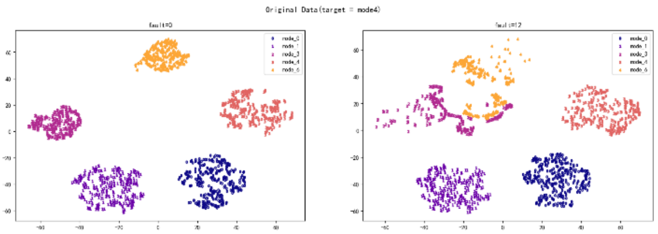

Figure 2. Normal (left) and Fault 12 (right) data in different modes

3.2 Ablation experiment

The model underwent ablation experiments on multiple source domains and a single target domain, and the results are shown in Table 2.

It can be found from Table 2 that if the domain adaptive part is removed, the model's performance in the target domain is extremely poor, and the domain adaptive part greatly solves operating conditions; Adding contrastive learning can keep the model's performance on source domain data to a certain extent, but introducing contrastive learning increases the depth and complexity of the model, and there may be over-fitting on easily distinguishable faults, resulting in a slight decrease in performance.

3.3 Analysis

Figure 3 shows the raw data, indicating that each fault is mixed and each mode is dispersed. Figure 4, Figure 5 and Figure 6 show the dimensionality reduction visualization of the data, with different colors representing different faults, and normal data in blue. After adding domain adaptation and contrastive losses, the model plays a role in bringing data from different domains closer, while also bringing similar data closer. This enables the model to classify faults in the target domain data.

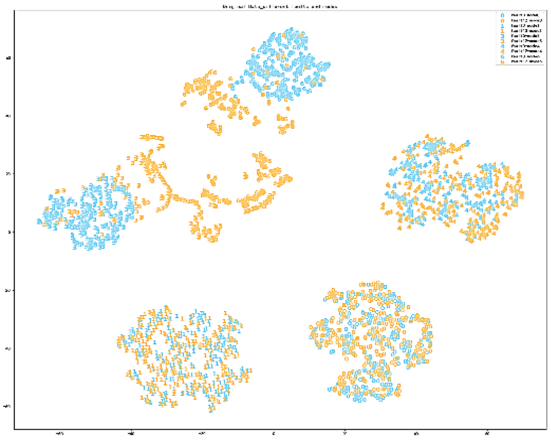

Figure 3. Dimensionality reduction visualization of raw data

(Different colors represent different types of faults, and different numbers represent different operating conditions)

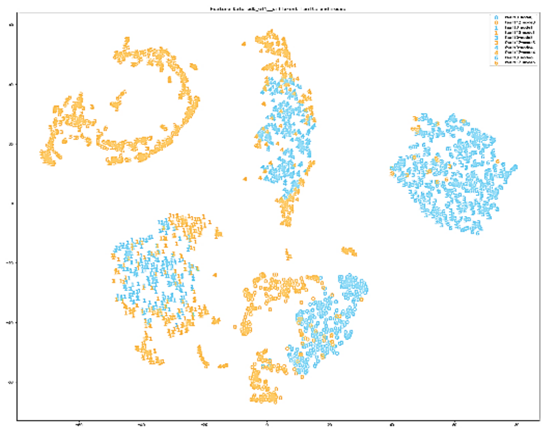

Figure 4. Dimensionality reduction visualization (α_{cd}=0, α_{cl}=1)

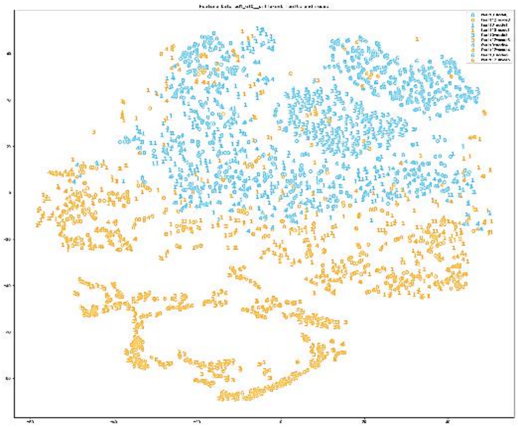

Figure 5. Dimensionality reduction visualization (α_{cd}=1, α_{cl}=0)

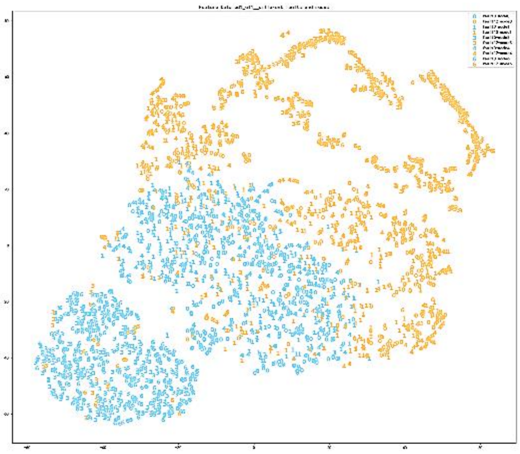

Figure 6. Dimensionality reduction visualization (α_{cd}=1, α_{cl}=1)

4. CONCLUSIONS

This paper proposed an end-to-end model architecture that addresses the problem of anomaly detection in multi-operating condition time series data in industrial scenarios. The approach incorporated domain adaptation and contrastive learning methods and experiments were conducted on a simulated dataset to demonstrate the necessity of domain adaptation and the importance of contrastive learning. We applied strategies mitigate overfitting, such as regularization techniques or model complexity control. However, overfitting was indeed a concern in complex models, which was inevitable in deep learning. In future, further research is needed to explore the integration of domain adaptation and contrastive learning.

ACKNOWLEDGEMENT

This work was supported by National Key R&D Program of China (2021YFB3301200).

REFERENCES

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3, 1-40.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 1-58.

Liu, Q., and Qin, S. (2016). Research prospects for big data modeling in process industry. *Journal of Automation*, 42(02), 161-171.

Saeed, H. A., Peng, M. J., Wang, H., and Zhang, B. W. (2020). Novel fault diagnosis scheme utilizing deep learning networks. *Progress in Nuclear Energy*, *118*, 103066.

Bi, X., et al. (2022). One step forward for smart chemical process fault detection and diagnosis. *Computers and Chemical Engineering*, 164.

Xu, J., Xiao, L., and López, A. M. (2019). Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7, 156694-156706.

Wu, H., and Zhao, J. (2020). Fault detection and diagnosis based on transfer learning for multimode chemical processes. *Computers and Chemical Engineering*, 135.

Pan, S. J., and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.

Csurka, G. (2017). Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint* arXiv:1702. 05374.

Gretton, A., Fukumizu, K., Teo, C., et al. (2007). A kernel statistical test of independence. *Advances in neural information processing systems*, 20.

Lu, W., et al. (2017). Deep Model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3), 2296-2305.

Kim, D., Han, Y.-H., Zhuang, B., Yu, X., et al. (2021). Learning cross-modal contrastive features for video domain adaptation. *ICCV* 2021.

Pöppelbaum, J., Chadha, G. S., and Schwung, A. (2022). Contrastive learning based self-supervised time-series analysis. *Applied Soft Computing*, 117.

Bathelt, A., Ricker, N. L., and Jelali, M. (2015). Revision of the Tennessee Eastman process model. *IFAC Papers online*, 48(8), 309-314.

Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee-Eastman Challenge process. *Computers and Chemical Engineering*, 19(9), 949-959.