

Deep Transfer Learning based Multisource Adaptation Fault Diagnosis Network for Industrial Processes

Zheng Chai, Chunhui Zhao*

State Key Laboratory of Industrial Control Technology,
College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China
(Corresponding author*, email: chaizheng@zju.edu.cn, chzhao@zju.edu.cn)

Abstract: In industrial processes, there are generally multiple data sources generated from different working conditions, which can provide different fault diagnosis knowledge to the target application. In this paper, a multisource adaptation diagnosis network (MADN) method is proposed to transfer the diagnostic knowledge existed in multiple sources to the target. First, a stacked-autoencoder based feature generator is pretrained to extract feature representations from the process data acquired from diverse working conditions. Then, domain discriminators are developed to reduce the distribution discrepancy between the target domain and each of the sources in an adversarial way. The domain discrimination ability, on the other hand, also reveals the different importance of the source domains. Thus, the fault classifiers can be assembled to identify the fault types of the unlabeled target data. The superiority of the proposed method is verified using a real-world process.

Keywords: Adversarial learning, deep neural network, fault diagnosis, industrial process, multi-source domain adaptation.

1. INTRODUCTION

With the increment of complexity of modern industries, fault diagnosis methods have been extensively developed to diagnose the occurred faults and ensure the operating safety (Pilario and Cao, 2018; Iqbal et al., 2019; Zhao et al., 2020; Chen et al., 2019). In recent years, machine learning based fault diagnosis approaches, such as principal component analysis (Gajjar et al., 2018), exponential discriminant analysis (Yu and Zhao, 2019), support vector machines (SVM) (Deng et al., 2017), random forest (RF) (Chai and Zhao, 2020), and deep convolutional neural networks (Wu and Zhao, 2018), have been successfully applied to industrial process. These methods use massive data collected from diverse sensors and need no prior information about the process, offering effective approaches to the fault diagnosis problem. However, such methods have a strong assumption that the training data (source domain) and future test data (target domain) should have the same distribution. Unfortunately, such an assumption cannot hold well in real industrial cases as the working condition of process can vary with the progress of production, leading to a certain discrepancy between the source and target data.

To address this problem, several domain adaptation based deep neural network methods have been developed to enable cross-domain fault diagnosis. As a typical case of transfer learning (Pan and Yang, 2010), domain adaptation aims to learn transferrable features that have low distribution discrepancy across domains. Thus, the models learned from the source data can remain effective on the target. In recent years, the metrics of domain discrepancy like the maximum mean discrepancy (Lu et al., 2017; Wen et al., 2019) or

covariance discrepancy (Wang et al., 2019) have been embedded into the deep models to reduce the domain bias and learn domain-uninformative features. However, these methods consider that there is only single source domain in the process. In many industrial applications, however, there are massive historical data collected from various working conditions, which can provide different fault diagnosis knowledge to the target application. The distributions of these domains are different from not only the target domain, but also from each other. The single-source adaptation methods can suffer from some limitations when applied to the multisource scenarios directly. On one hand, as domain shifts can also exist across different source domains, it is not appropriate to simply combine the sources as single one. On the other hand, the sources have different distributions and can provide different diagnostic knowledge to the target, and thus the different importance of multiple sources in the transferring task should be revealed and considered.

To address the above problems, in this paper, we focus on the multisource domain setting for industrial process fault diagnosis, and a multisource adaptation diagnosis network (MADN) method is proposed. In MADN, the distribution of the target domain is expected to be aligned with that of the multisource domains as much as possible, enabling the source fault classifiers effective on the target domain. Note that different domains have an identical fault label space, as the unshared categories may lead to a negative transfer (Pei et al., 2018). Specifically, MADN consists of three modules, namely, a feature generator, multiple fault classifiers, and multiple domain discriminators. First, we pretrain the multi-layered stacked autoencoders (SAE) as the backbone of the proposed MADN, which can extract the high-level feature representations of the multidimensional process data. Based

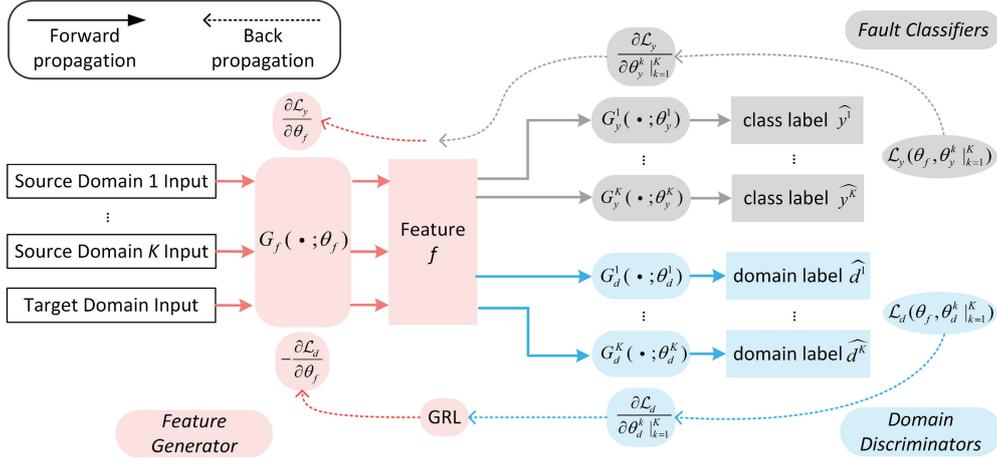


Fig. 1. The framework of the proposed MADN.

on the pretrained model, multiple fault classifiers are developed to discriminate different fault types of samples in each source to make the features fault-sensitive, and the domain discriminators are used to ensure that the features are domain-indistinguishable. After that, the fault classifiers trained on multiple adapted source domains can be assembled and re-weighted to diagnose the target samples. The key contribution of this work is twofold.

1) An MADN method is proposed to leverage and fuse the diagnostic knowledge from multiple sources to enhance the performance of fault diagnosis in industrial processes.

2) The SAE is pretrained to extract high-level representations from process data, and the discrepancy within each domain pair is optimized. Thus the fault classifiers can be assembled for effective target diagnosis.

The structure of the paper is organized as follows. Section 2 describes the methodology in detail. In Section 3, the case study on an industrial process is demonstrated. Conclusions are drawn in Section 4.

2. METHODOLOGY

In this section, the definitions and notations are presented first. Then, details of the proposed MADN and its training and online diagnosis procedure are introduced.

2.1 Definitions and Notations

Given K source domains $\mathcal{D}_S = \{\mathcal{D}_S^k\}_{k=1}^K$ with labelled fault data $\{\mathbf{X}_S^k, \mathbf{Y}_S^k\}_{k=1}^K$ and a target domain \mathcal{D}_T with unlabelled fault data $\{\mathbf{X}_T\}$, where different domains correspond to different working conditions in an industrial process. $\mathbf{x}_i \in \mathbf{X}$ is a sample vector composed of different sensor measurements, and $y_i \in \mathbf{Y}$ is the corresponding fault label. The joint distributions of different domains are different, i.e.,

$P(\mathbf{X}_S^1, \mathbf{Y}_S^1) \neq P(\mathbf{X}_S^2, \mathbf{Y}_S^2) \neq \dots \neq P(\mathbf{X}_S^K, \mathbf{Y}_S^K) \neq P(\mathbf{X}_T, \mathbf{Y}_T)$. The goal of multisource cross-domain fault diagnosis is to leverage the diagnostic knowledge existed in multiple sources \mathcal{D}_S , and develop an ensembled diagnosis model, which can identify the fault types of target samples precisely. It is noted that all the domains have an identical fault label space with C types of fault.

2.2 The Framework of MADN

The framework of the proposed MADN is shown in Fig. 1. There are three main parts: a feature generator, multiple fault classifiers, and multiple domain discriminators. The feature representations are extracted by the feature generator from the raw inputs, which are fed into the other two modules. On one hand, the features are input to the fault classifiers corresponding to different source domains to enable reliable fault classification results on the source domains. On the other hand, the features are input to the multiple domain discriminators, and each discriminator corresponds to a domain pair composed of the target data and one of the sources. The domain discriminator aims at distinguishing whether the features come from the source or the target domain accurately. On the contrary, the feature generator is expected to extract features that can confuse the discriminator as much as possible. The adversary between these two parts yields an equilibrium that the extracted features from both domains are indistinguishable. The specific modules in the MADN are given as follows.

1) Feature Generator

To extract high-level representations from the raw process data, an SAE based deep feature generator G_f parameterized by θ_f is exploited. SAE is a popular deep structure, which has shown prominent performance in the field of feature extraction of industrial process data in recent years (Yuan et al., 2018; Chai and Zhao, 2020). Specifically, SAE uses the autoencoder (AE) (Bengio et al., 2013) as the basic single-layer module to stack the deep structure. An AE encodes a

sample \mathbf{x}_i as a feature vector $e(\mathbf{x}_i)$, which can be decoded as $\hat{\mathbf{x}}_i$ that has minimum reconstruction error compared with \mathbf{x}_i . Denote \mathbf{w}_e as the encoder network weights and b_e as the corresponding bias, and then $e(\mathbf{x}_i)$ can be defined as

$$e(\mathbf{x}_i) = f_e(\mathbf{w}_e^T \mathbf{x}_i + b_e) \quad (1)$$

where f_e is the sigmoid activation function.

Based on the encoded feature $e(\mathbf{x}_i)$, the reconstructed $\hat{\mathbf{x}}_i$ output by the decoder network can be obtained by:

$$d(e(\mathbf{x}_i)) = f_d(\mathbf{w}_d^T e(\mathbf{x}_i) + b_d) \quad (2)$$

where \mathbf{w}_d denotes the weights of the decoder network, and b_d is the bias. f_d is the sigmoid activation function.

An AE aims to reconstruct the input using the representation $e(\mathbf{x}_i)$. Thus, the L2 reconstruction loss function is optimized in the AE training phase:

$$\mathcal{L}_{AE} = \|\mathbf{x}_i - d(e(\mathbf{x}_i))\|^2. \quad (3)$$

A single AE is generally structured as a one-hidden-layer network. To obtain the deep structure, a common way is to stack multiple AEs to obtain a stacked-AE structure. Specifically, after training an AE using (3), the decoder of the current AE is removed and the encoded $e(\mathbf{x}_i)$ is exploited as the input of next AE. Thus the SAE model G_f can be pretrained in an unsupervised manner.

2) Fault Classifiers

Following the feature generator, the fault classifiers $G_y^k |_{k=1}^K$ parameterized by $\theta_y^k |_{k=1}^K$ can be achieved using a fully-connected layer with softmax operator. Note that different domains share an identical fault label space. Thus, with the learned feature representation, the c th output ($c=1,2,\dots,C$) of the k th fault classifier $G_y^{k,c}$ can be obtained by

$$G_y^{k,c}(G_f(\mathbf{x}_i)) = \frac{\exp(\mathbf{v}_{k,c}^T G_f(\mathbf{x}_i))}{\sum_{j=1}^C \exp(\mathbf{v}_{k,j}^T G_f(\mathbf{x}_i))} \quad (4)$$

where $\mathbf{v}_{k,c}$ indicates the weights connected to the c th output neuron in the k th classifier.

With the predictions of G_y^k obtained, the cross-entropy loss \mathcal{L}_y^k is exploited as the optimization objective of the fault classifier:

$$\begin{aligned} \mathcal{L}_y^k(\theta_f, \theta_y) &= \frac{1}{n_S^k} \sum_{\mathbf{x}_i \in \mathcal{D}_S^k} \mathcal{L}_y^k(G_y^k(G_f(\mathbf{x}_i)), y_i) \\ &= -\frac{1}{n_S^k} \sum_{i=1}^{n_S^k} \sum_{c=1}^C \left[1\{y_i = c\} \log \frac{\exp(\mathbf{v}_{k,c}^T G_f(\mathbf{x}_i))}{\sum_{j=1}^C \exp(\mathbf{v}_{k,j}^T G_f(\mathbf{x}_i))} \right] \end{aligned} \quad (5)$$

where $1\{y_i = c\}$ is an indicator function. n_S^k indicates the number of samples in the k th source domain.

Then the loss of the fault classifiers can be calculated with the following equation:

$$\mathcal{L}_y(\theta_f, \theta_y) = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_y^k(\theta_f, \theta_y^k). \quad (6)$$

3) Domain Discriminators

The domain adversary training strategy designed by Ganin et al. suggests to use a domain discriminator to identify the domain label of samples, i.e., distinguishing a sample belongs to the source or the target domain (Ganin et al., 2016). Here, each source domain \mathcal{D}_S^k and the target domain \mathcal{D}_T can compose a domain pair $\{\mathcal{D}_S^k, \mathcal{D}_T\}$. A sample in the pair can be labelled as 0 if it belongs to \mathcal{D}_S^k while labelled as 1 if it belongs to \mathcal{D}_T . Thus, the domain discriminator aims to perform binary classification. Formally, the k th domain discriminator G_d^k learns a logistic regressor to map the embedded feature $\mathbf{f}_i = G_f(\mathbf{x}_i)$ into the domain label space, i.e., $\{0, 1\}$. The fully-connected layer is used as the domain discriminator. Denote the weights and the corresponding bias of the fully-connected layer as \mathbf{u} and z respectively, and then the output of G_d^k can be calculated as follows

$$G_d^k(\mathbf{f}_i, \theta_d^k) = \text{sigmoid}(\mathbf{u}^T G_f(\mathbf{x}_i) + z). \quad (7)$$

The logistic regression loss of G_d^k can then be obtained by

$$\begin{aligned} \mathcal{L}_d^k(\theta_f, \theta_d^k) &= \frac{1}{n_S^k + n_T} \sum_{\mathbf{x}_i \in (\mathcal{D}_S^k \cup \mathcal{D}_T)} \mathcal{L}_d^k(G_d^k(G_f(\mathbf{x}_i)), d_i) \\ &= -\frac{1}{n_S^k + n_T} \sum_{\mathbf{x}_i \in (\mathcal{D}_S^k \cup \mathcal{D}_T)} [d_i \log G_d^k(G_f(\mathbf{x}_i)) + \\ &\quad (1-d_i) \log(1-G_d^k(G_f(\mathbf{x}_i)))] \end{aligned} \quad (8)$$

where d_i is the domain label, which is set as 1 if \mathbf{x}_i is sampled from the source, or set as 0 if sampled from the target.

Then the loss of the multiple domain discriminators can be obtained as follows:

$$\mathcal{L}_d(\theta_f, \theta_d) = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_d^k(\theta_f, \theta_d^k). \quad (9)$$

2.3 The Training Strategy of MADN

In this part, the overall optimization objective and training strategy of the MADN are presented. First, the unsupervised SAE based feature extractor is pre-trained to initialize the parameters of the feature generator with the loss function in (3). Then, by combining the two optimization objectives in (6) and (9), the overall objective of MADN can be written as:

$$\mathcal{L}(\theta_f, \theta_y^k |_{k=1}^K, \theta_d^k |_{k=1}^K) = \mathcal{L}_y(\theta_f, \theta_y) - \lambda \mathcal{L}_d(\theta_f, \theta_d) \quad (10)$$

where λ is the parameter that balances the multisource fault classification and the adversarial domain adaptation.

The objective of MADN is to optimize the parameters $\theta_f, \theta_y^k |_{k=1}^K, \theta_d^k |_{k=1}^K$ such that:

$$\begin{aligned} (\hat{\theta}_f, \hat{\theta}_y^k |_{k=1}^K) &= \arg \min_{\theta_f, \theta_y^k |_{k=1}^K} \mathcal{L}(\theta_f, \theta_y^k |_{k=1}^K, \theta_d^k |_{k=1}^K) \\ \hat{\theta}_d^k |_{k=1}^K &= \arg \max_{\theta_d^k |_{k=1}^K} \mathcal{L}(\theta_f, \theta_y^k |_{k=1}^K, \theta_d^k |_{k=1}^K). \end{aligned} \quad (11)$$

In (11), the parameters $\hat{\theta}_y^k |_{k=1}^K$ of the fault classifiers are optimized to make the fault classification more accurate, i.e., minimize the fault classification loss. For $\hat{\theta}_f$, on one hand, it is optimized to minimize the fault classification error \mathcal{L}_y . On the other hand, it aims to confuse the domain discriminator such that the discriminator is prone to produce a wrong domain discrimination result. Thus, $\hat{\theta}_f$ is also optimized to maximize the domain discrimination loss \mathcal{L}_d . Finally, the parameters $\hat{\theta}_d^k |_{k=1}^K$ of the domain discriminators are optimized to improve the domain-discrimination performance, i.e., minimize \mathcal{L}_d . Thus, the feature generator and the domain discriminator in the proposed MADN are trained in an adversarial way.

The optimization objective in (11) can be solved by exploiting the stochastic gradient descent (SGD) algorithm (Goodfellow et al., 2016). Specifically, the parameters $\theta_f, \theta_y^k |_{k=1}^K, \theta_d^k |_{k=1}^K$ are updated as follows:

$$\begin{aligned} \theta_f &\leftarrow \theta_f - \varepsilon \left(\frac{\partial \mathcal{L}_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f} \right) \\ \theta_y &\leftarrow \theta_y - \varepsilon \frac{\partial \mathcal{L}_y}{\partial \theta_y} \\ \theta_d &\leftarrow \theta_d - \varepsilon \frac{\partial \mathcal{L}_d}{\partial \theta_d}. \end{aligned} \quad (12)$$

It is noticed that for θ_f , the signs of the partial derivative from the classification loss \mathcal{L}_y and the discrimination loss \mathcal{L}_d are opposite. Thus, the gradient reversal layer (GRL) (Ganin et al., 2016) is used to implement this adversarial process. The GRL has no impact on the feed-forward process, while the gradient direction is reversed by multiplying the gradients with -1 in the backward propagation process.

2.4 Diagnosis Procedure

The diagnosis flowchart using the proposed MADN is depicted in Fig. 2. In the training phase, the labelled fault data in K different working conditions and the unlabelled target domain data are collected, in which the fault labels of the unlabelled data are required to be predicted. Then, the training strategy (10) and (12) is used to train the MADN model. Finally, K fault classifiers can be obtained from K source domains. Note that different sources have different similarities with the target domain, and thus using some metrics to reweigh the multiple source classifiers to boost the target learning has become a common practice (Xu et al. 2018). It is noted that after the adaptation, if two domains in a

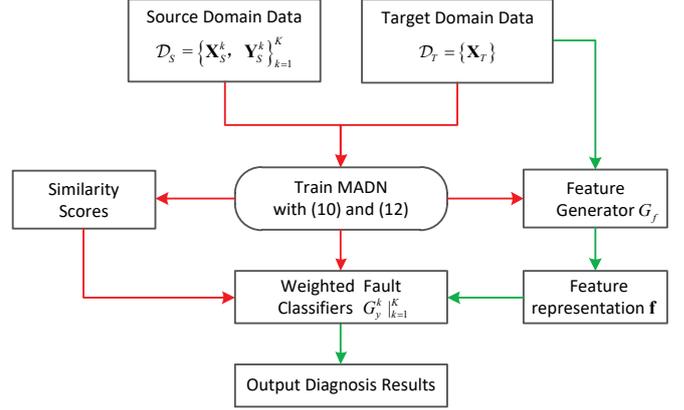


Fig. 2. The diagnosis flowchart of the proposed MADN. The red and green lines represent the training and testing phases, respectively.

pair are close to each other, the corresponding domain discriminator is more likely to produce a wrong result. Thus, it is a natural idea to use the discrimination loss as a metric to measure the similarity between two domains in a domain pair, as a larger discrimination loss indicates higher similarity:

$$w_k = \frac{\mathcal{L}_d^k(\theta_f, \theta_d^k)}{\sum_{j=1}^K \mathcal{L}_d^j(\theta_f, \theta_d^j)}. \quad (13)$$

In the testing stage, first, the high-level representation \mathbf{f} can be obtained using G_f . Then, \mathbf{f} is fed into the K fault classifiers, and each classifier can output a probability distribution over the label space of C fault classes. Inspired by Xu et al., the multiple distributions can be then assembled using similarity scores (Xu et al. 2018). Thus the class with the maximum probability is the predicted diagnostic result. Formally, the prediction can be obtained as follows:

$$\hat{y} = \arg \max_c \left[\sum_{k=1}^K \left(\frac{\exp(\mathbf{v}_{k,c}^T \mathbf{f})}{\sum_{j=1}^C \exp(\mathbf{v}_{k,j}^T \mathbf{f})} w_k \right) \right]_{c=1}^C \quad (14)$$

where $\mathbf{v}_{k,c}$ indicates the weights connected to the c th output neuron in the k th classifier.

3. CASE STUDY

In this section, the superiority of the proposed MADN is validated on a real-world industrial multiphase flow process dataset collected from Cranfield University (Ruiz-Cárcel et al., 2015). The multiphase flow facility is designed to provide a measured and controlled multiphase flow system. Water, oil, air, or their mixtures can be used as input to the facility, which are finally separated using a multiphase separator. A sketch map of this multiphase flow facility is given in Fig. 3.

In the multiphase flow case, all the data were sampled at a frequency of 1 Hz. There are 24 process variables measured in the system. Besides, two process inputs including the air

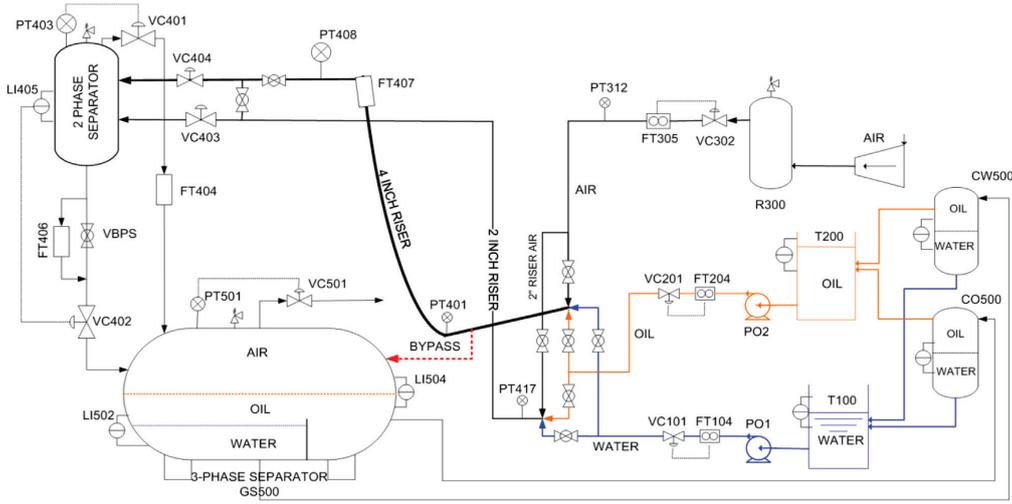


Fig. 3. The sketch map of the multiphase flow facility.

and water flow rate setpoints are also recorded. Different setpoint values can generate different working conditions of the TPF process, resulting in different domains. In this paper, three domains with different setpoints are used. In detail, in the first domain, the water flow rate is varying all the time. In the second domain, the water flow rate is fixed as 2 kg/s. In the third domain, the water flow rate is fixed as 3.5 kg/s. These three different domains are denoted by domains A, B, and C in this section. Four kinds of faults, including the air line blockage, the water line blockage, the top separator input blockage, and the open direct bypass are considered, which are denoted by Fault 1 to Fault 4, respectively. According to the three domains, three multisource transfer tasks are designed, including $A+B \rightarrow C$, $A+C \rightarrow B$, and $B+C \rightarrow A$. Taking $A+B \rightarrow C$ as an example, it indicates that the domains A and B serve as the sources, which are transferred to the domain C, i.e., the target domain.

To validate the effectiveness of the proposed MADN, four methods, including deep neural network for domain adaptation in fault diagnosis (DAFD) (Lu et al., 2017), SAE, RF, and SVM, in which the first method is the typical single-

source single-target domain adaptation diagnosis method and the last three methods are extensively used for fault diagnosis in recent years (Feng and Zhao, 2021), are selected as the comparison methods. The number of trees in RF is 100. The kernel of SVM is polynomial. For SAE, DAFD, and the proposed MADN, three AEs are stacked as the basic feature extractor. The architecture is fixed as FC(24)-FC(100)-FC(100)-FC(50), where FC(a) means that there are a neurons in the fully-connected layer. The architecture of the fault classifier in the three methods is fixed as FC(50)-FC(4). The architecture of the domain discriminator in the proposed MADN is fixed as FC(50)-FC(10)-FC(2). The stochastic gradient descent algorithm is used to optimize the loss function of the neural network and the early stopping strategy is used. The trade-off parameter λ in (10) changes gradually from 0 to 1 using $\lambda = 2 / (1 + \exp(-10 \times t)) - 1$, where t is the linear learning progress varying from 0 to 1. The learning rate is set as 0.01, and the batch size is set as 256. To more comprehensively evaluate the diagnostic performance of the proposed method, two metrics including accuracy and F1 score are exploited, which have been widely used in previous work (Lu et al., 2017; Chai and Zhao, 2020).

Table 1. Performance comparison for different methods in TPF case (%)

Methods	A+B→C		A+C→B		B+C→A		Mean Accuracy	Mean F1 Score
	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score		
The proposed MADN	90.13	90.08	71.29	67.18	81.09	81.18	80.84	79.48
DAFD	74.21	70.09	63.75	62.67	75.51	74.41	71.16	69.06
SAE	76.58	76.24	51.14	44.38	71.60	71.92	66.44	64.18
RF	67.52	66.64	44.20	35.98	70.37	67.91	60.70	56.84
SVM	72.67	71.52	46.90	30.21	45.76	37.31	55.11	46.35

The cross-domain diagnosis performance on the three tasks is shown in Table 1. First, we compare the performance of the two deep adaptation methods including the proposed MADN and DAFD. It can be observed that the proposed MADN shows 80.84% mean accuracy and 79.48% mean F1 score over the three tasks, which are significantly better than the

comparison method DAFD. The potential reason is that the DAFD simply mixes the two source domains as one and does not consider the discrepancy between the two sources. While for the propose DADN, the discrepancy within each domain pair is considered, and the different importance are used to assemble the fault classifiers, yielding improved performance.

Then, for the rest deep model SAE and the shallow models RF and SVM, we can observe that these three methods show inferior performance in comparison with the deep adaptation methods. Because these traditional machine learning methods assume that the distributions of the training and testing data are identical, which is challenged in industrial applications with various working conditions. Finally, in comparison with the shallow models RF and SVM, the deep model SAE shows better performance on such cross-domain fault diagnosis task, demonstrating the effectiveness of the SAE in modelling industrial process data.

4. CONCLUSION

In this paper, we focus on the multisource cross-domain fault diagnosis problem, and propose a novel method termed as multisource adaptation diagnosis network (MADN). In MADN, the SAE structure is utilized to extract high-level representations. Multiple domain discriminators are utilized to make the learned features within each domain pair transferrable, and thus the fault discrimination capacity learned from multiple sources can remain effective on the target data, with different importance of the sources in the transferring task revealed. Experiment on a real-world multiphase flow dataset verifies the performance of the proposed MADN.

ACKNOWLEDGEMENTS

This work was supported by National Key R&D Program of China (2019YFC1908100).

REFERENCES

- Pilario, K. E. S. and Cao, Y. (2018). Canonical variate dissimilarity analysis for process incipient fault detection. *IEEE Transactions on Industrial Informatics*, 14, 5308-5315.
- Iqbal, R., Maniak, T., Doctor, F., and Karyotis, C. (2019). Fault detection and isolation in industrial processes using deep learning approaches. *IEEE Transactions on Industrial Informatics*, 15, 3077-3084.
- Zhao, C., Chen, J., Jing, H. (2020). Condition-driven data analytics and monitoring for wide-range nonstationary and transient continuous processes. *IEEE Transactions on Automation Science and Engineering*, doi: 10.1109/TASE.2020.3010536.
- Chen, H., Jiang, B., Ding, S. X., Lu, N., and Chen, W. (2019). Probability-relevant incipient fault detection and diagnosis methodology with applications to electric drive systems. *IEEE Transactions on Control Systems Technology*, 27, 2766-2773.
- Gajjar, S., Kulahci, M., and Palazoglu, A. (2018). Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control*, 67, 112-128.
- Yu, W. and Zhao, C. (2019). Online fault diagnosis in industrial processes using multimodel exponential discriminant analysis algorithm. *IEEE Transactions on Control System Technology*, 27, 1317-1325.
- Deng, F., Guo, S., Zhou, R., and Chen, J. (2017). Sensor multifault diagnosis with improved support vector machines. *IEEE Transactions on Automation Science and Engineering*, 14, 1053-1063.
- Chai, Z. and Zhao, C. (2019). Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification. *IEEE Transactions on Industrial Informatics*, 16, 54-66.
- Wu, H. and Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers and Chemical Engineering*, 115, 185-197.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22, 1345-1359.
- Lu, W., Liang, B., Cheng, Y., Meng, D., Yang, J., and Zhang, T. (2017). Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 63, 2296-2305.
- Wen, L., Gao, L., and Li, X. (2019). A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49, 136-144.
- Wang, X., He, H., and Li, L. (2019). A hierarchical deep domain adaptation approach for fault diagnosis of power plant thermal system. *IEEE Transactions on Industrial Informatics*, 15, 5139-5148.
- Pei, Z., Cao Z., Long, M., and Wang, J. (2018). Multi-adversarial domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 3934-3941.
- Yuan, X., Huang, B., Wang, Y., Yang, C., and Gui, W. (2018). Deep learning based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Transactions on Industrial Informatics*, 14, 3235-3423.
- Chai, Z. and Zhao, C. (2020). A fine-grained adversarial network method for cross-domain industrial fault diagnosis. *IEEE Transactions on Automation Science and Engineering*, 17, 1432-1442.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1798-1828.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press, Cambridge, MA, USA.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17, 2096-2030.
- Xu, R., Z. Chen, W. Zuo, J. Yan, and L. Lin. (2018). Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7976-7985.
- Ruiz-Cárcel, C., Cao, Y., Mba, D., Lao, L., and Samuel, R. (2015). Statistical process monitoring of a multiphase flow facility. *Control Engineering Practice*, 42, 74-88.
- Feng, L. and Zhao, C. (2021). Fault description based attribute transfer for zero-sample industrial fault diagnosis. *IEEE Transactions on Industrial Informatics*, 17, 1852-1862.