Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-5 (2020) 37–42

Transfer Learning for Smart Manufacturing: A Stepwise Survey

Shufei Li* and Pai Zheng*

** Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China*

(E-mail: shufei.li@connect.polyu.hk; pai.zheng@polyu.edu.hk)

Abstract: Nowadays, industrial companies embrace the cutting-edge artificial intelligence (AI) techniques to achieve smart manufacturing over the entire organization. However, effective data collection and annotation still remain as a big challenge in many manufacturing scenarios. Transfer learning, serving as a breakthrough of learning sharing knowledge and extracting latent features from scarce data, has attracted much attention. Transfer learning in literature mainly focuses on the definitions and mechanisms of interpretation while lacking a systematic implementation scheme for manufacturing. To fulfill this gap and facilitate industrial resource use efficiency, this paper attempts to systematize strategies of transfer learning in today's smart manufacturing in a step-by-step manner. Twenty representative transfer learning works are investigated from the perspectives of manufacturing activities along the engineering product lifecycle. Meanwhile, the potential availability of industrial dataset is also briefly introduced. It is hoped this research can provide a clear guide for both academics and industrial practitioners to design appropriate learning approaches according to their own industrial scenarios.

Keywords: Transfer learning, smart manufacturing, domain adaptation, manufacturing intelligence

1. INTRODUCTION

Currently, with the emergence of Industrial Internet of Things (IIoT), cyber-physical-human systems become a prevailing paradigm with a huge amount of data generated in modern factories. These data contain useful information and knowledge on design, manufacturing, distribution and usage stages across the engineering product lifecycle. Hence, it is necessary to learn latent representations from industrial data, which immerse manufacturing activities with condition awareness and decision-making capabilities.

As a prevailing approach for pattern classification, deep learning is regarded as a powerful solution to extract knowledge and make appropriate decision from industrial data (Li, ZHENG and Zheng, 2020). Nevertheless, deep learning-based computing intelligence relies heavily on enormous amounts of data, which has to cover all possible scenarios for one manufacturing task. It is often expensive, time-consuming, and even unrealistic in many industrial activities (Arinez *et al.*, 2020). To overcome such bottleneck and achieve better resource efficiency, transfer learning enables to construct deep neural networks using pre-trained models or transfer knowledge between different feature spaces, avoiding training models from scratch and annotating data collected across different working conditions, machines, products and users (i.e., industrial scenario variances).

In general, strategies of transfer learning include: instance-based transfer learning, feature-based transfer learning, model-based transfer learning and relation-based transfer learning (Zhuang *et al.*, 2019). Note that the main objective in industrial tasks is domain adaptation, which focuses on reducing the distribution difference of data collected among industrial scenario variances (i.e., instances between the source domain and the target domain). In this context, solutions mainly consist of finetune, adaptation layer and generative adversarial nets for deep learning models, which extract general features and share latent weights between the source domain and the target domain (e.g., different machines) for knowledge transferring. However, most people who engaged in the traditional manufacturing industry have few experiences on constructing transfer learning networks for smart manufacturing process. Therefore, this research aims to provide a stepwise survey on the transfer learning cases adopted in manufacturing activities, so that assist academics and industrial practitioners can readily design their own appropriate learning approaches accordingly.

The remaining is organized as follows. Section 2 introduces the manufacturing intelligence in industrial activities and the available datasets of each scenario. Transfer learning-based solutions are then introduced to transfer knowledge between industrial tasks. Recommended network architectures and preferred transfer leaning-based training strategies are given as a guideline in Section 3. Finally, conclusions and future research directions are highlighted in Section 4.

2. APPLICATIONS OF TRANSFER LEARNING IN INDUSTRIAL SCENARIOS

2.1 Domain Adaptation for Multi-source Manufacturing Data

10.1016/j.ifacol.2021.04.081

As aggregated sensory data along engineering lifecycle is obtained across different operation conditions or manufacturing configurations enabled by the IIoT, those industrial data suffer variances on feature space and probability distribution. Traditional deep learning methods fail to generate intelligence since these models have to be trained and tested on data from the same domain, by annotating sufficiently typical data coving all possible conditions (e.g., user/manufacturing/product/sensor etc.). It remains as a big challenge for most industrial scenarios with large time consumption, high price cost and much annotation difficulty resulted from labelling multi-source data. Hence, domain adaptation plays a critical role in minimising the impact of distribution differences of those data and further learning useful knowledge.

For a manufacturing activity, source domain D^s can be obtained by labelling a small amount of data on a single working condition or one machine. The source domain D^s consists of a feature space χ^s and a marginal distribution $P(X^s)$, i.e., $D^s=\left(\chi^s,P(X^s)\right)$, where X^s denotes the obtained dataset. Data collected in real production process is regarded as the target domain D^t, which is defined as $D^t=\left(\chi^t,P(X^t)\right)$. Because of the different conditions, machines or sensors, dataset between the source domain and the target domain are not in the same distribution. Normally, the source domain is referred as $D^s=\left\{\left(X_i^s,Y_i^s\right)\right\}$, where $X_i^s\in\chi^s$ is a data sample and Y_i^s is its corresponding label. While the target domain is denoted as $D^t=\left\{\left(X_i^t\right)\right\}$, where $X_i^t\in\chi^t$ is an unlabelled data example. Note that it is assumed that the source and target domains share the domain-invariant feature spaces and label spaces, as data from different working conditions or machines only differ in their probability distributions. Therefore, the objective of transfer learning is to train a classifier f with the source domain data, where the classifier f: $X^t\mapsto Y^t$ can predict the label Y_i^t for a target data X_i^t.

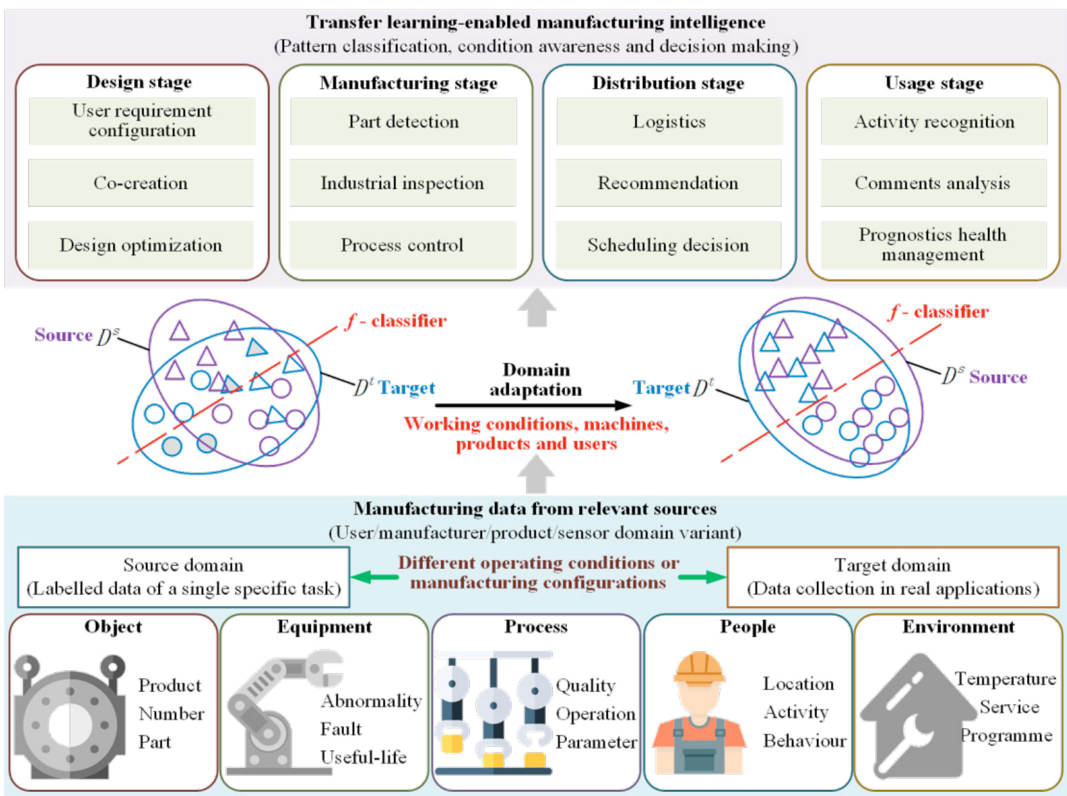

Fig. 1. An overview of transfer learning-enabled manufacturing intelligence throughout engineering lifecycle.

2.2 Transfer Learning-enabled Manufacturing Intelligence and Dataset Availability

Transfer learning-enabled manufacturing intelligence can be achieved via domain adaptation between data across different industrial scenario variances, including finetune, adaptation layer and generative adversarial nets, as presented in Fig. 1 and Table 1. It mainly aims to construct sharing feature representations between following variant conditions, e.g., from old configurations to new one, from historical records to real one, from simulation to physics, and from other objects to target one. In products' design stage (Lin et al., 2018), data knowledge from old configurations can be shared in lithography simulation scenarios via finetune. In the manufacturing stage, public datasets, such as ImageNet and COCO (Rendall et al., 2018), can be regarded as source data for some object detection tasks. For scheduling decision-making during the distribution stage, historical production records can be referred as source dataset to fine-tune prediction model in the production progress (Huang et al., 2019). In the usage stage over products' lifecycle, especially for the PHM task, there are a large number of available public datasets, e.g., CWRU, IMS, Bogie and Crack datasets (Li et al., 2020). Therefore, various methods have been deployed to improve the capacity of the target prediction function in the target domain (i.e., real applications) using the knowledge learned from the source domain (i.e., simulation data or other machines' data).

In this context, transfer learning can adapt a deep learning model that has been trained via a source domain, to a relevant target domain in which data is scare. Simulation data, historical data and data of almost the same objects (e.g., machines, humans, products) are preferred as the source domain. Meanwhile, despite the listed application scenarios, there is still more potential cases (see Fig. 1), of which transfer learning can assist production activities and boost manufacturing intelligence. With the supplement of the application scenarios, the selection program for source dataset and corresponding knowledge-transferring methods can be provided to enterprises as the assisted guideline for service deploying in the production process.

Table 1. Transfer learning-based applications along manufacturing activities

	Scenarios	Source data	Strategy	Method	Ref.
Design optimization	Lithography simulation	Old lithography configurations	Model-based	Finetune	(Lin *et al.*, 2018)
Part detection	Pellet classification	ImageNet	Model-based	Finetune	(Rendall *et al.*, 2018)
Process control	Tool selection for CNC machines	Process case samples	Instance-based	Finetune	(Zhou *et al.*, 2018)
Scheduling decision	Production progress prediction	Historical production data	Model-based	Finetune	(Huang *et al.*, 2019)
Prognostics health management (PHM)	Fault diagnosis in a car production line	Simulation data	Feature-based	Adaptation layer	(Xu *et al.*, 2019)
	Fault diagnosis of rotating machines	CWRU, IMS, Bogie and Crack datasets	Feature-based	Generative adversarial nets	(Li *et al.*, 2020)
	Prognostics of cutting tool	Labelled data of another tool	Model-based	Adaptation layer	(Sun *et al.*, 2019)

Table 2. Preferred transfer learning architecture

Architecture		Data	Ref.
Sparse Autoencoder (SAE)	Two hidden layers	Two hyperspectral datasets (around 42776 samples for each dataset) – Images	(Deng *et al.*, 2019)
	Three hidden layers	Vibration signals collected with 4 motor loads (2000 samples with 4000 sample lengths) – Signals	(Wen, Gao and Li, 2019)
	Five hidden layers	4000 samples of Device Electrocardiogram during the production process (each sample containing 196 data points) – Signals	(Xu *et al.*, 2019)
Denoising Autoencoder (DAE)	Two hidden layers	4 categories in ImageNet (2000 samples for each category) – Images	(Zhu *et al.*, 2019)
VGG16	Frozen the first 3 convolution blocks	6000 samples for 6 working conditions of motor (each sample containing 1024 data points) – Signals	(Shao *et al.*, 2019)
	Initialized weight of ImageNet	5923 pellet images with two classifications – Images	(Rendall *et al.*, 2018)
ResNet	Three residual blocks	Five hyperspectral datasets (around 42776 samples for each dataset) – Images	(Zhao *et al.*, 2020)
	Four residual blocks	8000 samples of 980 mask clips (each sample consisting an image and centre threshold) – Images	(Lin *et al.*, 2019)

3. STEP-WISE APPLICATION IN PRACTICE

3.1 Data Scale and Transfer Learning Architecture

To efficiently extract sharing domain-invariant features between source and target domains, the transfer learning-based network architecture should be designed according to the scale, modality, and format of available data, as illustrated in Table 2. For example, sparse Autoencoder (SAE) with three hidden layers was utilized to extract sharing invariant latent feature spaces of vibration signals between four different working conditions (Wen, Gao and Li, 2019). A 16-layer convolutional neural network (CNN)-stacked network (i.e. VGG16) was re-trained with target image data to fine-tune the feature extraction ability in a part detection system (Rendall et al., 2018). For 3D images containing RGB colours and threshold information, residual neural network

(ResNet) was adopted to classify invariant features from old lithography configurations to a target one (Lin *et al.*, 2019). These typical works provide a benchmark that indicates the data scale requirement for transfer learning-based models, when the model is trained by architectures from AE, VGG16 to ResNet in a single modality. Besides, stacked AE architectures are normally employed to extract latent information of 1D-format data, including time-series signals of fault diagnostics and prognostics, text structures of production data (Huang *et al.*, 2019), and recordings of the production process (Shao *et al.*, 2019). While stacked CNN architectures are almost utilized to classify sharing features from 2D-format data (time-frequency data or images) or 3D-format data (depth images).

3.2 Annotation Data and Transfer Learning Procedure

Based on the availability of data between source and target domains, three transfer learning procedures can be employed as solutions for domain adaptation over industrial data, i.e., finetune, adaptation layer and generative adversarial nets.

Finetune-based transfer learning mainly transfers knowledge at the model/parameter level, and it is effective when there are sufficient annotation datasets in the source domain but only a small amount of annotation data in the target domain. A typical finetune-based deep CNN architecture is shown in Fig. 2. As the convolutional operation (*w*x)* for 2D data in CNN is spontaneous without confusion, a finetune-based feature extractor which can learn knowledge from 1D data is illustrated as an example here, i.e., 1D convolution and pooling operations. Let $X_i^D \in R^n$ be a data sample from $\left(\chi^s;\chi^t\right)$ as the input, where n is the length of data points, i.e. $X_i^D=\left[x_i^{(1)},x_i^{(2)},\ldots,x_i^{(n)}\right]$. Correspondingly, a 1D convolution kernel is referred as $K_j^c \in R^{\omega} \mid j=1,2,\ldots,m$ with filter size w, where m is the number of filters. The convolutional (conv) layer is to take a dot product between the kernel K_j^c and the input X_i^D to extract features,

$$X_i^{D,C}=f\left(X_i^D;\theta^C\right)=\sigma_r\left(\sum_{j=1}^{m}K_j^c\bullet x_i^{(k):(k+w-1)}+b^C\right) \tag{1}$$

Where $\bullet$ is the dot product, $b^C \in R^{\omega}$ is the bias term and σ_r is the non-linear activation function (e.g., Relu). By sliding the kernel K_j^c over X_i^D for $k=1$ to $k=n-w+1$, output vector $X_i^{D,C} \in R^{n-w+1}$ can be obtained,

$$X_i^{D,C}=\left[x_{i,c}^{(1)},x_{i,c}^{(2)},\ldots,x_{i,c}^{(n-w+1)}\right] \tag{2}$$

Next is the pooling operation, which is achieved by connecting a pooling layer with the convolutional layer. Let take the max pooling function as an example,

$$X_i^{D,P}=\max\left\{x_{i,c}^{(l\times s):((l+1)\times s)} \mid l=1,2,\ldots,(n-w+1)/s\right\} \tag{3}$$

Where s is the pooling length, and $X_i^{D,P} \in R^{(n-w+1)/s}$ is the pooling output vector. After blocks of convolution and pooling operations, fully connected (FC) layer is connected to flatten the output,

$$X_i^{D,FC}=flatten\left[down\left(X_i^{D,P},s\right)\right] \tag{4}$$

Where, $flatten[\bullet]$ is the flatten function, and $down(\bullet)$ is shown in (3). Then softmax regression can be selected as the activation function for the last FC layer and predict a confidence score of label for each data sample. For one *K*-label classification task, the probability of data sample X_i^D belonging to the q-th label is,

$$p\left(Y_i^D=q \mid X_i^{D,FC};\theta^F\right)=\frac{\exp\left(w_q^F X_i^{D,FC}+b_q^F\right)}{\sum_{q=1}^{k}\exp\left(w_q^F X_i^{D,FC}+b_q^F\right)} \tag{5}$$

Where $\theta^F=\left\{w_q^F,b_q^F\right\}$ denotes weight and bias terms of the last FC layer. As low-level layers normally pay attention to extract general features (e.g., objects' corner), finetune-based transfer learning often use pre-trained weights (e.g., weight from ImageNet or COCO) to initial networks, avoiding training from scratch. Besides, these low-level layers can also be trained by source data which subjects to almost the same distribution, to improve extraction ability. Then, these layers are frozen, and weights of high-level layers is fine-tuned via the target domain dataset for knowledge transferring. Due to the existence of public large-scale labelling data or nearly zero-cost simulation data, application scenarios of finetune-based transfer learning approaches include part detection, industrial inspection, tool selection and scheduling decision.

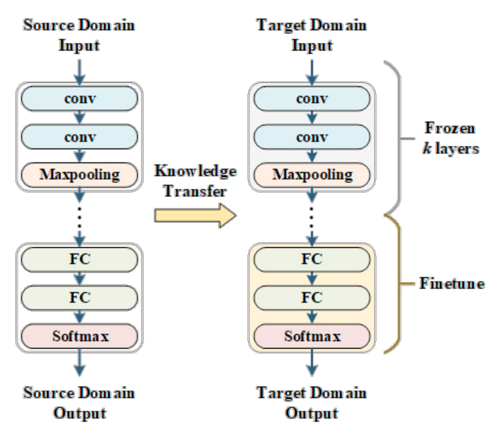

Fig. 2. Finetune-based transfer learning procedure.

The core of the adaptation layer-based transfer learning approach is to reduce the discrepancy of extracted features between source and target domains, as shown in Fig. 3 (a). By minimizing domain loss and classification errors, the extractor of feature extraction layers can focus on extracting domain-invariant features. The loss function L of adaptation layer-based transfer learning is defined as：

$$L=L_c\left(D^s,Y^s\right)+\lambda L_a\left(D^s,D^t\right) \tag{6}$$

Where $L_c\left(D^s,Y^s\right)$ is the classification loss on annotation data of the source domain, $L_a\left(D^s,D^t\right)$ is the domain loss on the discrepancy penalty, λ balances the weights of these two losses. For a dataset with K labels, the classification errors can be defined as,

$$L_c=-\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{K}I\left[Y_i^D=j\right]\log\left(P\left(Y_i^D=j\mid X_i^{D,FC};\theta^F\right)\right)\right] \\ =-\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{K}I\left[Y_i^D=j\right]\log\frac{\exp\left(w_j^F X_i^{D,FC}+b_j^F\right)}{\sum_{l=1}^{K}\exp\left(w_l^F X_i^{D,FC}+b_l^F\right)}\right] \tag{7}$$

Where N is the batch size of training samples, and $I[\bullet]$ denotes an indicator function. The domain loss normally adopts maximum mean discrepancy (MMD) distance (Guo *et al.*, 2019), which is defined as,

$$D_H\left(D^s,D^t\right):=\sup_{\phi\in H}\left\{E_{X^s\sim p}\left[\phi\left(X_i^{s,FC}\right)\right]-E_{X^t\sim q}\left[\phi\left(X_i^{t,FC}\right)\right]\right\} \tag{8}$$

Where ":=" means "define" and sup$(\cdot)$ is the supremum of one set. The reproducing kernel Hilbert space (RKHS) is denoted as H and $\phi(\bullet)\in H$ means mapping data to feature spaces in RKHS. With n_s data samples X_i^s from the source domain and n_t ones X_i^t from the target domain, the distance between $p\left(X_i^{s,FC}\right)$ and $q\left(X_i^{t,FC}\right)$ is calculated as,

$$D=\left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\phi\left(X_i^{s,FC}\right)-\frac{1}{n_t}\sum_{j=1}^{n_t}\phi\left(X_j^{t,FC}\right)\right\|_H^2 \\ =\frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\left\langle\phi\left(X_i^{s,FC}\right),\phi\left(X_j^{s,FC}\right)\right\rangle_H+\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\left\langle\phi\left(X_i^{t,FC}\right),\phi\left(X_j^{t,FC}\right)\right\rangle_H \\ -\frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\left\langle\phi\left(X_i^{s,FC}\right),\phi\left(X_j^{t,FC}\right)\right\rangle_H \tag{9}$$

As $\left\langle\phi(x),\phi(y)\right\rangle_H$ can always be calculated by kernel function $k(x,y)$ in RKHS. Taking Gaussian kernels as an example, the unbiased estimation of D_H is calculated as follows,

$$\hat{D}\left[X^s,X^t\right]=\frac{1}{N(N-1)}\sum_{i\neq j}^{N}\exp\left(-\left\|X_i^{s,FC}-X_j^{s,FC}\right\|^2/2\sigma^2\right) \\ +\frac{1}{N(N-1)}\sum_{i\neq j}^{N}\exp\left(-\left\|X_i^{t,FC}-X_j^{t,FC}\right\|^2/2\sigma^2\right) \\ -\frac{2}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\exp\left(-\left\|X_i^{s,FC}-X_j^{t,FC}\right\|^2/2\sigma^2\right) \tag{10}$$

Therefore, the loss function L is rewritten as,

$$L=L_c+\lambda\hat{D} \tag{11}$$

As shown in Fig. 3 (a), let θ_f and θ_c be parameters for the feature extractor and the classifier. The optimization object can be denoted as,

$$L\left(\theta_f^*,\theta_c^*\right)=\min_{\theta_f,\theta_c}L_c\left(\theta_f,\theta_c\right)+\lambda\hat{D}\left(\theta_f\right) \tag{12}$$

With the learning rate α, parameters θ_f and θ_c of the adaptation layer-based model can be updated as follows,

$$\theta_f\leftarrow\theta_f-\alpha\left(\frac{\partial L_c}{\partial\theta_f}+\lambda\frac{\partial\hat{D}}{\partial\theta_f}\right) \\ \theta_c\leftarrow\theta_c-\alpha\frac{\partial L_c}{\partial\theta_c} \tag{13}$$

In this context, adaptation layer-based transfer learning can be employed in scenarios, of which data of the target domain lack annotation information. It's an effective solution to solve the large differences of data distribution between source and target domains, or difficulty of data annotation. Industrial applications include user requirement configuration, activity recognition, comments analysis and PHM, as sensory data of these organisations suffer huge discrepancy.

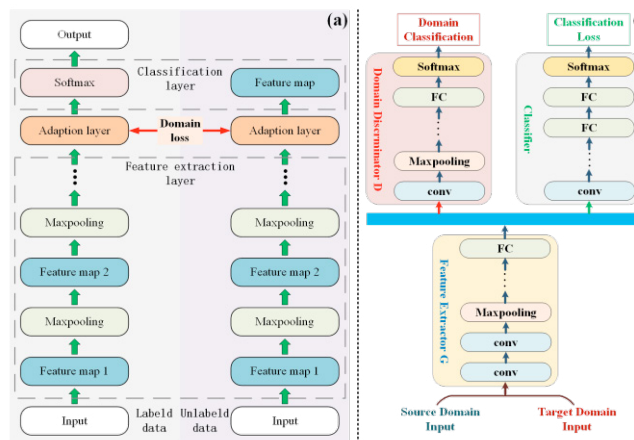

Fig. 3. Adaptation layer (a) and generative adversarial net (b).

Generative adversarial nets contain feature extractor G, domain discriminator D and classifier, as shown in Fig. 3 (b). Feature extractor G continuously learns domain-invariant features between source dataset and target datasets, aiming to make the domain discriminator D unable to distinguish which domain one data comes from. The loss function of generative adversarial nets is defined as,

$$L=L_c\left(D^s,Y^s\right)+\mu L_d\left(D^s,D^t\right) \tag{14}$$

Where, the optimization objective is to minimize the category classification error $L_c\left(D^s,Y^s\right)$ and maximize the domain classification error $L_d\left(D^s,D^t\right)$ respectively, and the $L_c\left(D^s,Y^s\right)$ is the same as (7). The domain classification cost can be donated as,

$$L_d=-\frac{1}{n_s}\sum_{i=1}^{n_s}l_d^i\left(X_i^{s,FC}\right)-\frac{1}{n_t}\sum_{j=1}^{n_t}l_d^j\left(X_j^{t,FC}\right) \tag{15}$$

Where, the l_d is referred as follows,

$$l_d=\frac{1}{n}\sum_{i=1}^{n}\left(d_i\log\frac{1}{d\left(X_i^{D,FC}\right)}+\left(1-d_i\right)\log\frac{1}{1-d\left(X_i^{D,FC}\right)}\right) \tag{16}$$

Where, d_i is the ground truth label of the belonging domain for the i-th sample $X_i^{D,FC}$, and $d\left(X_i^{D,FC}\right)$ is the predicted one for the sample. Similarly, adding a parameter θ_d for domain discriminator D, the model can be updated like (13). In this way, generative adversarial nets-based transfer learning further improves the ability to learn domain-invariant features between source and target domains. This approach fits knowledge transfer between different working conditions in diagnostics and prognostics applications, of which sensory signals data are almost subject to the same distribution.

4. CONCLUSIONS AND DISCUSSION

This research provides a stepwise guide to both engineers and researchers who concern about transfer learning-enabled manufacturing intelligence along any stage of the engineering product lifecycle. The objective of transfer learning in industrial activities, i.e., domain adaptation for multi-source manufacturing data, were depicted, and application scenarios and corresponding source domain datasets were sorted out. Moreover, transfer learning-based network architectures and training strategies were recommended as the guideline for practitioners based on existing industrial applications. In this way, more transfer learning-based solutions may readily extract domain-invariant features and learn sharing knowledge between different operating conditions or manufacturing configuration in industrial activities. Despite the abovementioned contributions, two promising future research are highlighted here, namely 1) exploring the impact of simulation data generated by cyber-physical-human systems or digital twin on the performance of transfer learning-based models, and 2) studying transferability between source domains and target domains in an interpretable manner to avoid a negative transfer.

REFERENCES

Arinez, J. F. *et al.* (2020) ‘Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook’, *Journal of Manufacturing Science and Engineering*, 142(11), pp. 1–16. doi: 10.1115/1.4047855.

Deng, C. *et al.* (2019) ‘Active Transfer Learning Network: A Unified Deep Joint Spectral-Spatial Feature Learning Model for Hyperspectral Image Classification’, *IEEE Transactions on Geoscience and Remote Sensing*. IEEE, 57(3), pp. 1741–1754. doi: 10.1109/TGRS.2018.2868851.

Guo, L. *et al.* (2019) ‘Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines with Unlabeled Data’, *IEEE Transactions on Industrial Electronics*. IEEE, 66(9), pp. 7316–7325. doi: 10.1109/TIE.2018.2877090.

Huang, S. *et al.* (2019) ‘A Two-Stage Transfer Learning-Based Deep Learning Approach for Production Progress Prediction in IoT-Enabled Manufacturing’, *IEEE Internet of Things Journal*. IEEE, 6(6), pp. 10627–10638. doi: 10.1109/JIOT.2019.2940131.

Li, S., ZHENG, P. and Zheng, L. (2020) ‘An AR-Assisted Deep Learning Based Approach for Automatic Inspection of Aviation Connectors’, *IEEE Transactions on Industrial Informatics*, pp. 1–1. doi: 10.1109/tii.2020.3000870.

Li, Xiang *et al.* (2020) ‘Diagnosing Rotating Machines With Weakly Supervised Data Using Deep Transfer Learning’, *IEEE Transactions on Industrial Informatics*, 16(3), pp. 1688–1697. doi: 10.1109/TII.2019.2927590.

Lin, Y. *et al.* (2018) ‘Data efficient lithography modeling with residual neural networks and transfer learning’, *Proceedings of the International Symposium on Physical Design*, pp. 82–89. doi: 10.1145/3177540.3178242.

Lin, Y. *et al.* (2019) ‘Data Efficient Lithography Modeling with Transfer Learning and Active Data Selection’, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. IEEE, 38(10), pp. 1900–1913. doi: 10.1109/TCAD.2018.2864251.

Rendall, R. *et al.* (2018) ‘Image-based manufacturing analytics: Improving the accuracy of an industrial pellet classification system using deep neural networks’, *Chemometrics and Intelligent Laboratory Systems*, 180(July), pp. 26–35. doi: 10.1016/j.chemolab.2018.07.001.

Shao, S. *et al.* (2019) ‘Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning’, *IEEE Transactions on Industrial Informatics*. IEEE, 15(4), pp. 2446–2455. doi: 10.1109/TII.2018.2864759.

Sun, C. *et al.* (2019) ‘Deep Transfer Learning Based on Sparse Autoencoder for Remaining Useful Life Prediction of Tool in Manufacturing’, *IEEE Transactions on Industrial Informatics*. IEEE, 15(4), pp. 2416–2425. doi: 10.1109/TII.2018.2881543.

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(1), pp. 136–144. doi: 10.1109/TSMC.2017.2754287.

Xu, Y. *et al.* (2019) ‘A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning’, *IEEE Access*. IEEE, 7, pp. 19990–19999. doi: 10.1109/ACCESS.2018.2890566.

Zhao, X. *et al.* (2020) ‘Classification of small-scale hyperspectral images with multi-source deep transfer learning’, *Remote Sensing Letters*. Taylor & Francis, 11(4), pp. 303–312. doi: 10.1080/2150704X.2020.1714772.

Zhou, J. *et al.* (2018) ‘Tool selection method based on transfer learning for CNC machines’, *Mechanical Sciences*, 9(1), pp. 123–146. doi: 10.5194/ms-9-123-2018.

Zhu, Y. *et al.* (2019) ‘Transfer learning with deep manifold regularized auto-encoders’, *Neurocomputing*. Elsevier B.V., 369, pp. 145–154. doi: 10.1016/j.neucom.2019.08.078.

Zhuang, F. *et al.* (2019) ‘A Comprehensive Survey on Transfer Learning’, (10), pp. 1–27. Available at: http://arxiv.org/abs/1911.02685.