# A New Particle-Swarm-Optimization-Assisted Deep Transfer Learning Framework With Applications to Outlier Detection in Additive Manufacturing

Jingzhong Fang, Zidong Wang, Weibo Liu, Linwei Chen, and Xiaohui Liu

**Abstract**

In wire arc additive manufacturing (WAAM), the electric arc is an essential part of the welding equipment, which serves as the heat source and is directed by the current and voltage. The working status of the electric arc is an important factor in determining the quality of the fabricated components. During the welding process, the current and voltage may change abruptly due to some abnormalities in the operating conditions, which may affect the working status and thereby affect the quality of products. Such abnormal changes in the current and voltage can be treated as outliers. In order to identify outliers in current and voltage to further improve the welding process, in this paper, a novel deep-transfer-learning-embedded outlier detection approach is developed for WAAM. A new domain adaptation strategy is designed where the cross-domain discrepancies of the marginal distribution and conditional distribution are minimized. Specifically, two separate coefficients are introduced to adjust the conditional domain discrepancies of normal instances and outliers with the purpose of alleviating the data imbalance problem. The particle swarm optimizer is employed to adjust the hyper-parameters. The developed deep transfer learning framework is exploited in designing a new outlier detector with application to WAAM. The proposed approach is exploited in real-world industrial data collected through the WAAM process. Experimental results demonstrate that the proposed outlier detection approach outperforms the standard deep-learning-based outlier detector approach and the standard transfer-learning-embedded outlier detection approach in terms of detection accuracy.

**Index Terms**

Outlier detection, deep transfer learning, particle swarm optimization, industrial data analysis, additive manufacturing.

## I. Introduction

By identifying abnormal instances, outlier detection has been widely used in industrial data analysis. As a promising technique, outlier detection has been successfully exploited in various areas, e.g., medical engineering, finance, electrical engineering, and manufacturing [1]–[4]. Thanks to its powerful feature extraction ability, deep learning (DL) has been extensively utilized in outlier detection, which contributes to the rapid development of DL-based outlier detection methods. Recently, numerous DL algorithms have been presented for tackling various outlier detection tasks [5], [6].

Owing to the data-hungry nature of DL, a vast amount of training data with high-quality labels are required to build a reliable DL-based outlier detection model [7]–[10]. By using DL methods, training data and testing data should be independent and identically distributed (i.i.d.). Nevertheless, it is difficult to guarantee that the collected data obey the i.i.d. rule in real-world problems. Fortunately, such constraint

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 820776 (INTEGRADDE), the Engineering and Physical Sciences Research Council (EPSRC) of the UK, Brunel University London BRIEF funding, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. (*Corresponding author: Zidong Wang*)

J. Fang, Z. Wang, W. Liu, L. Chen and X. Liu are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (email: Zidong.Wang@brunel.ac.uk)

could be relaxed with the assistance of transfer learning (TL) which attempts to fully leverage the knowledge discovered from the source domain for analyzing the target domain [11], [12]. Among existing transfer learning techniques, deep TL (DTL) techniques have been widely adopted due mainly to their strong feature extraction and knowledge transfer abilities.

Serving as a popular class of DTL techniques, domain adaptation (DA) has been developed to reduce the cross-domain distribution discrepancy [13], [14]. Recently, various DA methods have been introduced, which mainly include metric learning DA (MLDA) approaches, reconstruction-based DA approaches, and adversarial learning DA approaches [15], [16]. Among them, the MLDA approaches could quantify the cross-domain distribution discrepancy with the assistance of statistical approaches. Recognised as a well-known distance metric, in recent years, the maximum mean discrepancy (MMD) has been extensively exploited in MLDA due to its computational efficiency and flexibility [17]. In many existing MMD-based MLDA approaches, the influences of the marginal distribution and conditional distribution are assumed to be the same, which neglects the difference between source and target data with regards to data characteristics. In this case, the trained DL model may be biased. To balance the influences of various distribution discrepancies and inspired by [18], [19], we will introduce weighting factors into the loss function of MLDA methods.

Data imbalance is a widespread problem in outlier detection, which makes it difficult to build a reliable detection model [20]–[22]. In most real-world scenarios, the quantity of normal instances is much larger than that of abnormal instances (i.e., outliers). In this paper, we separately consider the distribution discrepancies of outliers and normal instances in the proposed deep DA (DDA) strategy by introducing two independent coefficients (i.e., hyper-parameters). The coefficients are employed to balance the influences of the normal data and outliers to alleviate the data imbalance problem.

Hyper-parameters are generally chosen based on experimental experience, which is time-consuming [23], [24]. A reasonable solution is to apply optimization techniques to select optimal hyper-parameters. Thanks to its fast convergence rate and low implementation difficulty, particle swarm optimization (PSO) has been successfully adopted to choose optimal hyper-parameters [25]–[27]. It is therefore reasonable to employ the PSO algorithm to automatically tune the hyper-parameters.

To sum up, this paper proposes a PSO-embedded DTL framework for outlier detection on industrial data. The utilized data set is collected by multiple sensors deployed in a wire arc additive manufacturing (WAAM) pilot line. The contributions of this paper lie in the following threefold:

1) a novel DA strategy is proposed, where the weighting factors are designed to balance i) the marginal distribution discrepancy loss and ii) the conditional distribution discrepancy loss of both normal instances and outliers to tackle the data imbalance problem;
2) the PSO algorithm is employed to select the optimal weighting factors automatically; and
3) the developed approach is deployed to detect the outliers on real-world industrial data obtained through the WAAM process. Experimental results demonstrate the superiority of the developed PSO-embedded DTL outlier detection approach over two benchmark approaches.

The remaining sections of this paper are organized as follows. Related work of DA is discussed in Section II. In Section III, the developed DTL-based framework is introduced. Experiment settings, network configurations, and results are presented in Section IV. Finally, conclusions are presented in Section V.

## II. Related Work

The DA strategy aims to align the source and target domains by minimizing the cross-domain distribution discrepancy based on the conditional distribution $P(Y|X)$. In general, calculating $P(Y|X)$ directly is a difficult mission. Many existing DA strategies employ the Bayes' theorem to obtain $P(Y|X)$ [18], [28].

MMD is a powerful approach to quantify the distribution discrepancy between the source domain and target domain [17]. The MMD is calculated by mapping the original data into a reproducing kernel Hilbert space (RKHS). The MMD of $P\left(X\right)$ between the source domain and target domain can be calculated by:

$$D\left(X^s, X^t\right) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi\left(x_i^s\right) - \frac{1}{m}\sum_{i=1}^{m}\phi\left(x_i^t\right) \right\|_{\mathcal{H}}^2 \tag{1}$$

where $X^s$ and $X^t$ represent the source and target data, respectively; $n$ and $m$ are the total number of instances in the source and target domains, respectively; $x_i^s$ and $x_i^t$ indicate the $i$th samples in the source and target domains, respectively; $\phi(\cdot)$ denotes a mapping from the original space into the RKHS; and $\|\cdot\|_{\mathcal{H}}^2$ is the squared norm in the RKHS.

It is worth mentioning that in the DA strategy for the binary classification task, the marginal distributions $P\left(Y\right)$ of the source and target domains are the same, and the MMD of $P\left(X|Y\right)$ between the source and target domains (i.e., $D\left(X^s|Y^s, X^t|Y^t\right)$) can be calculated by accumulating the MMDs of the instances with same classes between the source and target domains. $D\left(X^s|Y^s, X^t|Y^t\right)$ can be calculated by:

$$\begin{aligned} &D\left(X^s|Y^s, X^t|Y^t\right) \\ =& \left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi\left(x_{i,1}^s\right) - \frac{1}{m_1}\sum_{i=1}^{m_1}\phi\left(x_{i,1}^t\right) \right\|_{\mathcal{H}}^2 \\ &+ \left\| \frac{1}{n_2}\sum_{j=1}^{n_2}\phi\left(x_{j,2}^s\right) - \frac{1}{m_1}\sum_{j=1}^{m_2}\phi\left(x_{j,2}^t\right) \right\|_{\mathcal{H}}^2 \end{aligned} \tag{2}$$

where $n_1$ and $n_2$ represent the total numbers of source domain instances in class 1 and class 2, respectively; $m_1$ and $m_2$ represent the total number of target domain instances in class 1 and class 2, respectively; $x_{i,1}^s$ and $x_{i,2}^t$ indicate the $i$th samples of the first class in the source and target domains, respectively; and $x_{j,1}^s$ and $x_{j,2}^t$ are the $j$th samples of the second class in the source and target domains, respectively.

## III. The DTL-Based Outlier Detection Approach

### A. Motivation

In some existing DDA-based classification methods, the loss function is designed by considering the classification loss and the domain loss. Recently, many researchers have focused on balancing the weights between the classification loss and the domain loss [28]–[30]. Nevertheless, the scale of distribution discrepancies between the source data and target data should also be considered. Using hyper-parameters to adjust the weight of each term in the loss function has been proven to be a fast and easy approach. In [28], the hyper-parameters have been introduced to adjust the weight among the domain loss, the regularization term, and the training loss. In [29], the hyper-parameters have been utilized to balance not only the weight among the domain loss, the regularization term, and the training loss, but also the weight between the different domain losses.

Note that data imbalance is a frequent problem in data analysis, which could lead to a biased model with unsatisfactory results. Recently, a number of strategies have been proposed in order to tackle the data imbalance challenge [19], [31]. In [31], a weighted MMD has been proposed to alleviate the data imbalance problem. In [19], the data imbalance problem has been tackled by adaptively changing the weights of samples from different classes. Specifically, the marginal distributions are assumed to be unchanged when calculating the MMD of $P(Y|X)$, which may affect the classification accuracy when dealing with other data sets. Motivated by the above discussions, the weights of different domain losses and the weights to

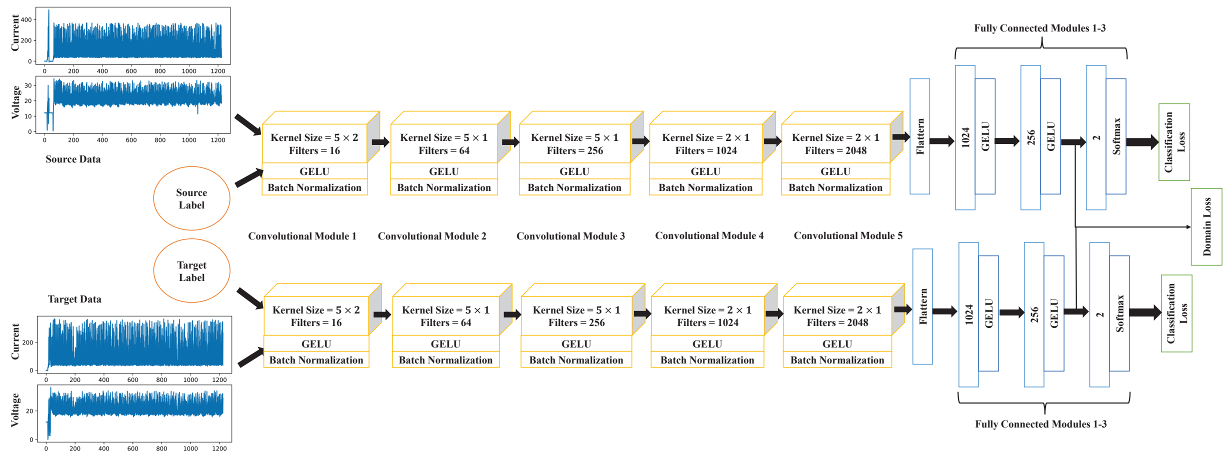


Fig. 1. The proposed PSO-embedded DTL outlier detection approach

tackle the imbalance problem are considered at the same time so as to design a proper loss function in this paper.

The selection of hyper-parameters is usually manual and based on experimental experience, which is time-consuming and inefficient [32]. In fact, selecting the optimal hyper-parameters is an optimization problem. As a popular optimization algorithm, the PSO algorithm (which is a famous evolutionary computation technique) has been widely exploited in various optimization tasks because of its high efficiency and easy implementation, which seems to be an appropriate solution to select the optimum hyper-parameters automatically.

In this paper, a novel PSO-embedded DTL framework is developed, where a new DA strategy is put forward to reduce the cross-domain discrepancy by adjusting the weight of marginal distribution discrepancy loss and the conditional distribution discrepancy loss. In addition, two separate coefficients are introduced to balance the conditional domain discrepancies of normal instances and outliers with the hope to alleviate the data imbalance problem. The hyper-parameters are tuned by the PSO algorithm automatically.

### *B. The PSO-Embedded DTL Framework*

The developed framework is displayed in Fig. 1. The convolutional neural network (CNN) is utilized for feature extraction, which comprises five convolutional modules and three fully connected (FC) layers. In each convolutional module, there is one convolutional layer followed by the Gaussian error linear unit (GELU) mapping and batch normalization. In the first and the second FC layers, the activation function is GELU. As a popular activation function, GELU shows better robustness and generalization ability than some existing activation functions (e.g., ReLU and Leakly-ReLU) in some sense [33]. As such, GELU is selected as the activation function. In the third FC layer, the softmax activation function is adopted.

### *C. Loss Function*

*1) Classification Loss:* The binary cross-entropy is applied to calculate the classification loss. In this paper, the classification loss includes the classification loss of the source data $L_c^s$ and the target data $L_c^t$.

$L_c^s$ and $L_c^t$ are given by:

$$L_c^s = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i^s \log\left(p_i^s\right) + \left(1-y_i^s\right)\log\left(1-p_i^s\right)\right] \tag{3}$$

$$L_c^t = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i^t \log\left(p_i^t\right) + \left(1-y_i^t\right)\log\left(1-p_i^t\right)\right] \tag{4}$$

where $n$ and $m$ are the source and target sample numbers, respectively; $p_i^s$ and $p_i^t$ represent the probabilities when the predicted label is the same as the real label of the $i$th sample in the source and target domains, respectively; $y_i^s$ and $y_i^t$ are the real labels of the $i$th sample in the source and target domains, respectively.

*2) Domain Loss:* The domain loss between source and target domains contains two parts (i.e., marginal distribution discrepancy $L_{\mathrm{MMD}}^{P(X)}$ and conditional distribution discrepancy $L_{\mathrm{MMD}}^{P(X|Y)}$), which can be calculated by:

$$L_{\mathrm{MMD}}^{P(X)} = \left\|\frac{1}{n}\sum_{k=1}^{n}\phi\Big(\psi\left(x_k^s\right)\Big) - \frac{1}{m}\sum_{k=1}^{m}\phi\Big(\psi\left(x_k^t\right)\Big)\right\|_{\mathcal{H}}^2 \tag{5}$$

$$\begin{aligned} L_{\mathrm{MMD}}^{P(X|Y)} &= \mu_1 L_{\mathrm{MMD}}^{P(X|Y=Normal)} + \mu_2 L_{\mathrm{MMD}}^{P(X|Y=Outlier)} \\ &= \mu_1\left\|\frac{1}{n_N}\sum_{i=1}^{n_N}\phi\Big(\psi\left(x_i^s\right)\Big) - \frac{1}{m_N}\sum_{i=1}^{m_N}\phi\Big(\psi\left(x_i^t\right)\Big)\right\|_{\mathcal{H}}^2 \\ &+ \mu_2\left\|\frac{1}{n_O}\sum_{j=1}^{n_O}\phi\Big(\psi\left(x_j^s\right)\Big) - \frac{1}{m_O}\sum_{j=1}^{m_O}\phi\Big(\psi\left(x_j^t\right)\Big)\right\|_{\mathcal{H}}^2 \end{aligned} \tag{6}$$

where $L_{\mathrm{MMD}}^{P(X|Y=Normal)}$ ($L_{\mathrm{MMD}}^{P(X|Y=Outlier)}$) denotes the domain discrepancy loss of normal data (outliers) between source and target domains; $\mu_1$ and $\mu_2$ are two coefficients; $n_N$ and $n_O$ represent the normal data and outliers in the source domain, respectively; $m_N$ and $m_O$ are the normal data and outliers in the target domain, respectively; $x_k^s$, $x_k^t$ represent the $k$th sample of the data in source and target domains, respectively; $x_i^s$ and $x_i^t$ are the $i$th sample of the normal data in source and target domains, respectively; $x_j^s$ and $x_j^t$ are the $j$th sample of outliers in source and target domains, respectively; and $\psi(\cdot)$ represents the distributions of deep features of $P(\cdot)$. $\psi(\cdot)$ is learned by five convolutional modules and the first two fully connected modules in the designed CNN model. In this paper, the Gaussian kernel is employed [34], which is given as follows:

$$k(a,b) = \exp\left(\frac{-\left\|a-b\right\|^2}{2\sigma^2}\right) \tag{7}$$

where $a$ and $b$ are two random samples; $\|\cdot\|$ is the Euclidean distance; and $\sigma$ represents the kernel's width.

*3) Overall Loss Function:* The entire loss function of the proposed outlier detection approach is given as follows:

$$\begin{aligned} L = {} & L_c^s + \delta L_c^t + \varepsilon L_{\mathrm{MMD}}^{P(X)} + \mu_1 L_{\mathrm{MMD}}^{P(X|Y=Normal)} \\ & + \mu_2 L_{\mathrm{MMD}}^{P(X|Y=Outlier)} + \lambda\left\|\omega\right\|_2 \end{aligned} \tag{8}$$

where $\|\cdot\|_2$ is the $L_2$ norm to avoid the overfitting problem; $\omega$ denotes the weights of the CNN; $\delta$ and $\varepsilon$ are weighting factors to balance the domain discrepancy loss of normal instances and outliers; and $\lambda$ is the penalty factor.

*Remark 1:* During the WAAM process, the working status is normal most of the time. The current and voltage will be abnormal only when some unexpected changes occur (e.g., arc instability and surface

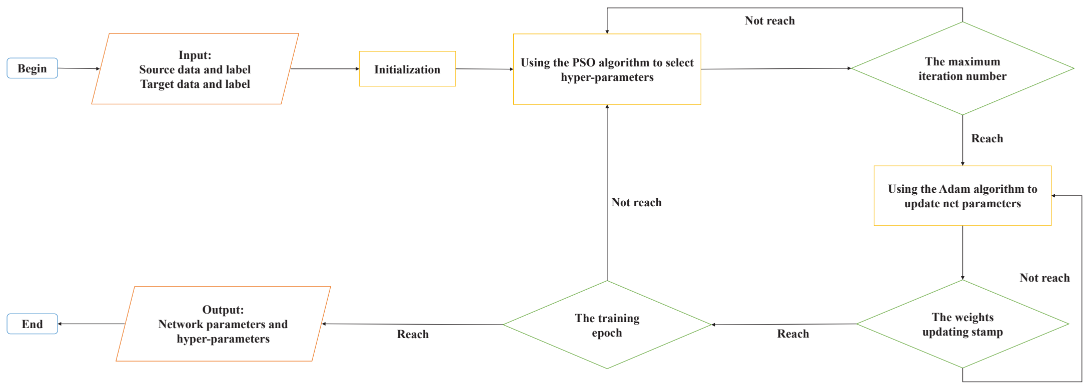


Fig. 2. The training procedure of the developed framework

contaminations). In this case, the WAAM data set seriously suffers from class imbalance problem, where the number of normal instances is much more than that of outliers. To alleviate the impact of class imbalance, in this paper, two coefficients (i.e., $\mu_1$ and $\mu_2$) are introduced. The weight between $L_{\mathrm{MMD}}^{P(X|Y=Normal)}$ and $L_{\mathrm{MMD}}^{P(X|Y=Outlier)}$ is balanced by adjusting $\mu_1$ and $\mu_2$, which improves the performance of the model on WAAM outlier detection.

### *D. Parameter Optimization*

*1) Basic PSO:* In the basic PSO algorithm, each particle in the swarm is a candidate solution, which is directed by its personal best location and the global best location (i.e., pbl and gbl) found by the entire swarm in a $D$-dimensional problem space. The updating equations of the velocity and position of the $\alpha$th particle at the $\beta$th iteration and the $d$th dimension (i.e., $v_{\alpha,\beta}^d$ and $x_{\alpha,\beta}^d$) are given as follows:

$$\begin{aligned} v_{\alpha,\beta+1}^d &= wv_{\alpha,\beta}^d + c_1r_1\left(pbl_{\alpha,\beta}^d - x_{\alpha,\beta}^d\right) \\ &\quad + c_2r_2\left(gbl_{\beta}^d - x_{\alpha,\beta}^d\right) \\ x_{\alpha,\beta+1}^d &= x_{\alpha,\beta}^d + v_{\alpha,\beta+1}^d \end{aligned} \tag{9}$$

where $w$ denotes the inertia weight; $c_1$ and $c_2$ represent two acceleration coefficients; $r_1$ and $r_2$ are two random numbers selected from $[0, 1]$; $pbl_{\alpha,\beta}^d$ denotes the pbl found by the $\alpha$th particle itself; $gbl_{\beta}^d$ is the gbl of all particles.

*2) Optimization Strategy:* The basic PSO algorithm is used for selecting proper hyper-parameters ($\sigma$, $\delta$, $\varepsilon$, $\mu_1$, and $\mu_2$) in this paper. The CNN weights $\omega$ are optimized by the Adam algorithm. During the training process, the hyper-parameters and net parameters are updated automatically. The training procedure of the method is depicted in Fig. 2.

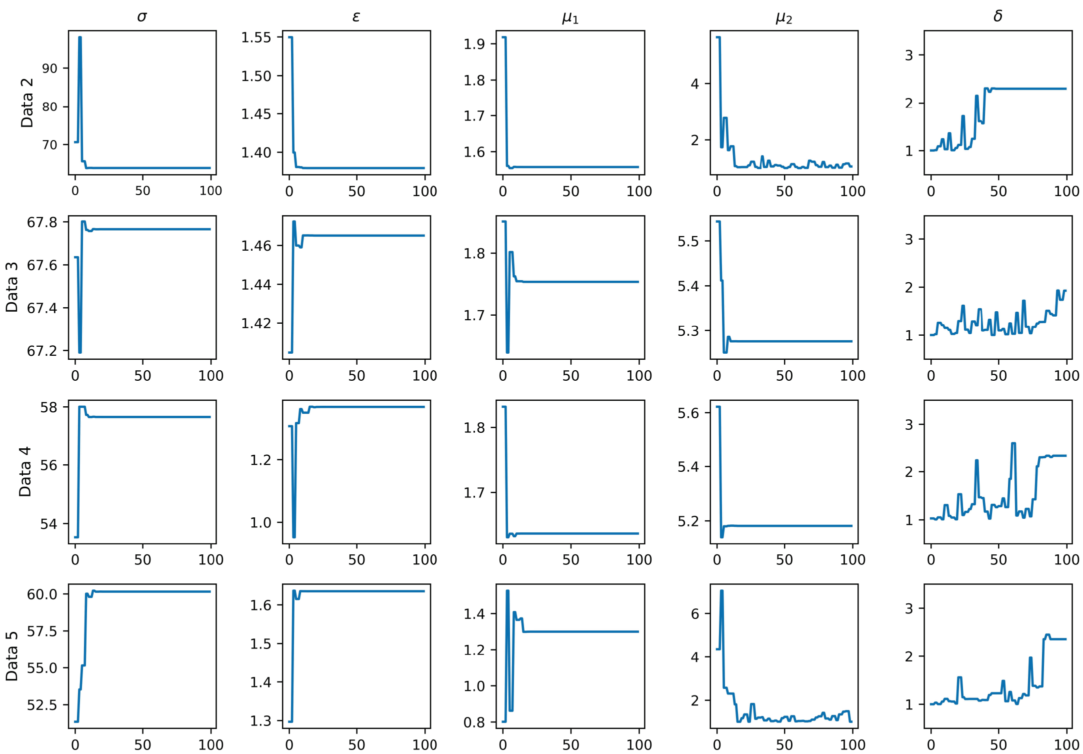


Fig. 3. The optimization process of the developed framework when Data 1 is the source data

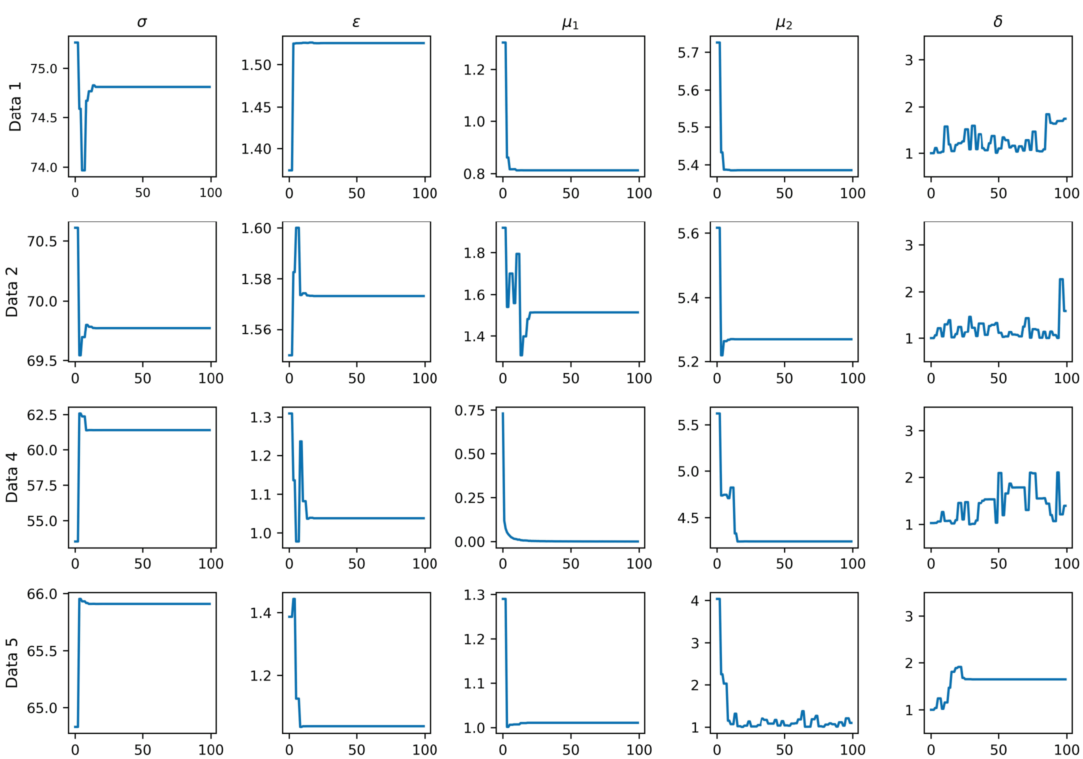


Fig. 4. The optimization process of the developed framework when Data 3 is the source data

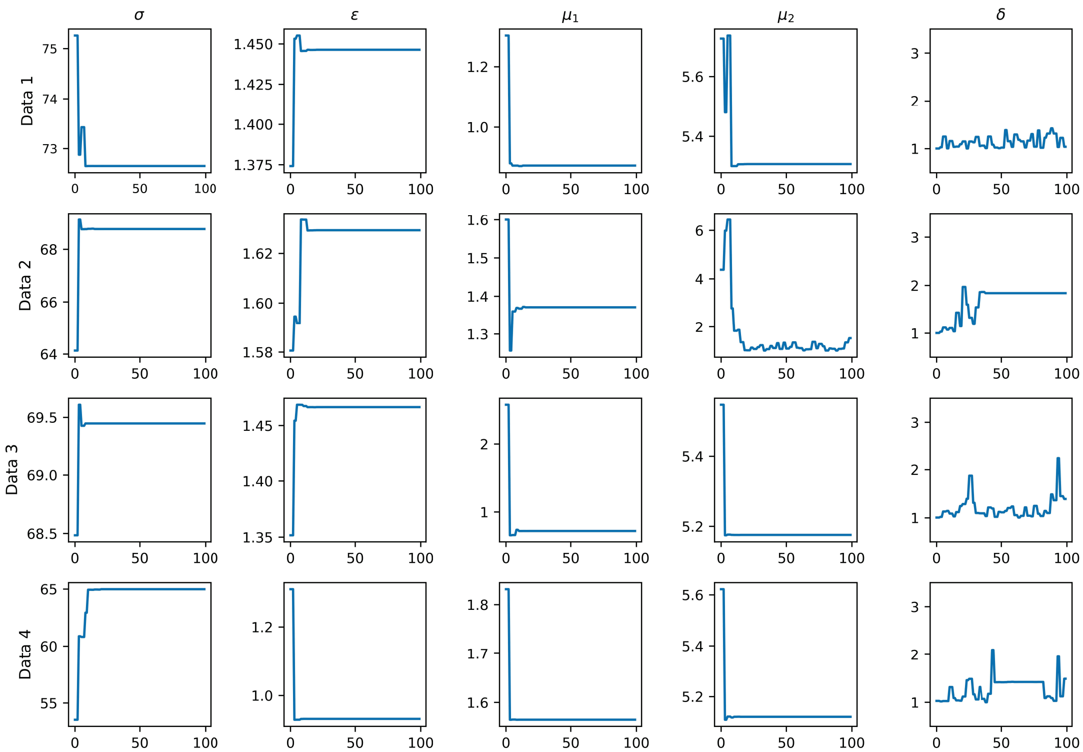


Fig. 5. The optimization process of the developed framework when Data 5 is the source data

# IV. WAAM Outlier Detection

Additive manufacturing (AM) has become a big breakthrough in industrial manufacturing, which has great advantages in building 3-D components and rapid prototyping [35]–[37]. In recent years, AM technology has been employed in a lot of fields (e.g., aerospace engineering, electrical engineering and biomedical engineering) [38]–[41]. As one of the most popular AM methods, WAAM is a wire-based direct energy deposition method, which has significant merits including high deposition and low cost [42]. It is worth mentioning that the electric arc is an important part of WAAM, which serves as the heat source and mainly directed by the total current and arc voltage. The working status of the electric arc is a key factor in deciding the quality of the fabricated components [43]. A common way to track the working status of the electric arc is to monitor the current and voltage because they reflect important information and are easy to be measured. Unfortunately, during the WAAM process, current/voltage may change irregularly and rapidly owing to abnormalities in the operating conditions (e.g., geometrical deviations and unstable metal transfer), which may have bad impacts on the working status, and affect the quality of products eventually. Recently, detecting current/voltage outliers and taking follow-up automatic corrective actions has become a useful way to ensure the WAAM working status [2].

## A. Data Sets

*1) Data Sets Description:* The data used in this experiment is collected from the WAAM process, which is deployed in Sweden. The collected data are classic time-series data which are divided into 5 data sets based on different welding strategies. Each data set represents an individual manufacturing task and contains 98000 instances and 5 attributes including the time stamp, the welding current, the welding voltage, and the operating time of the entire welding process. The manual labels of the data are annotated based on the expert knowledge, which contains two classes, namely “Normal” and “Outlier”.

*2) Data Pre-Processing:* In the experiment, the abnormal data points (e.g., missing or null) are removed to obtain clean data. To enhance the difference between the instances, the values of current and voltage are transformed to their cubed. Each data set is divided into several consecutive segments with the purpose of extracting and analyzing the features of time, where each segment (contains 56 instances) corresponds to a label (i.e., "Normal" and "Outlier"). The label of each segment depends on the label of each instance in the segment. Specifically, the label of the segment will be "Normal" if the labels of all the instances in this segment are "Normal", otherwise, the label of this segment will be "Outlier". In the experiment, the labels are transformed to one-hot formats for ease of classification.

TABLE I
THE OUTLIER DETECTION RESULTS USING SELECTED APPROACHES

| Evaluation Metrics | Approach | T1 (Data1, Data2) | T2 (Data2, Data4) | T3 (Data3, Data1) | T4 (Data4, Data5) | T5 (Data5, Data3) |
|---|---|---|---|---|---|---|
| Accuracy (%) | CNN | 78.85 | 92.46 | 99.00 | 96.38 | 95.23 |
| | CNN+DDA | 97.62 | 99.08 | 99.11 | 98.15 | 98.38 |
| | Developed Approach without Regularization | 97.04 | 98.18 | 97.69 | 98.17 | 97.25 |
| | **Developed Approach** | **97.69** | **99.15** | **99.46** | **98.62** | **99.08** |
| Precision (%) | CNN | 45.12 | 70.18 | 96.14 | 95.46 | 93.34 |
| | CNN+DDA | 94.65 | 98.52 | 96.15 | 97.55 | 96.15 |
| | Developed Approach without Regularization | 96.34 | 95.23 | 95.94 | 97.48 | 95.33 |
| | **Developed Approach** | **96.67** | **98.86** | **96.87** | **98.33** | **97.20** |
| Sensitivity (%) | CNN | 38.19 | 92.34 | 98.25 | 95.34 | 75.11 |
| | CNN+DDA | 90.94 | 95.69 | 98.68 | 96.82 | 94.94 |
| | Developed Approach without Regularization | 89.18 | 94.35 | 96.72 | 96.14 | 94.05 |
| | **Developed Approach** | **91.34** | **96.55** | **98.71** | **97.23** | **95.57** |

### *B. Model Training*

*1) Parameters Setting and Model Configuration:* The system of the experiment is Ubuntu 20.04.5. The GPU is NVIDIA RTX A6000 with 49 GB memory. The experiment conducted is based on Python 3.10.9, CUDA 11.7. The configurations of 5 convolutional modules and 3 FC layers are displayed in Fig. 1. In the experiment, the developed approach is compared with the CNN-based approach and the basic DTL-based approach with the purpose of verifying the performance of the developed approach. The ablation experiment is also conducted to verify the effectiveness of the regularization term in the developed approach. Note that the same network architecture and configurations of the CNN are employed for comparison.

In the experiment, the hyper-parameters ($\sigma$, $\delta$, $\varepsilon$, $\mu_1$, and $\mu_2$) are selected automatically by the standard PSO algorithm. The constraint intervals of hyper-parameters are set to be $[0, 100]$, $[1, 5]$, $[0, 3]$, $[0, 3]$, and $[1, 10]$, respectively. In order to put more consideration to the target data, the minimum of $\delta$ is set to be 1. The maximum of $\mu_2$ is set to be 12, which aims to tackle the data imbalance problem. Based on the experimental experience, the parameter $\lambda$ of the $L_2$ regularization term is set as 0.00001. In the standard PSO algorithm, the particle number is 25. The control parameters (e.g., $w$, $c_1$, and $c_2$) are set to be 0.6, 1.5, and 1.5, respectively.

*2) Model Training and Testing:* In this paper, two different data sets are selected as the source and target data sets which both contain all segments and labels of the pre-processed data sets. The training data consists of two parts, which are the source and target training data. To be specific, the source training data is the same as the source data. The target training data is the first 450 segments of the target data. Through the training process, the source and target training data are fed to the CNN at the same time. The classification loss is calculated based on the source and target training data. The domain loss is calculated based on the deep features of the source and target training data. The numbers of training epoch and PSO iteration are set to be 40 and 100, respectively. The weights updating stamp is set to be 50. The testing data is the rest of the target data excluding the target training data. The testing data is input to the trained model after the training process. The output of the testing process is the predicted labels of the testing data.

### *C. Results and Discussion*

*1) Experimental Results:* In the experiment, one data set is selected as the source data set, and other different data sets are selected as the target data sets. In this paper, we select three data sets (Data1, Data3 and Data5) as the source data sets in turn. The optimization process of the developed approach is shown in Figs. 3-5, where columns 1-5 represent $\delta$, $\varepsilon$, $\mu_1$, $\mu_2$, and $\sigma$, respectively, and rows 1-4 represent 4 target data sets.

In this paper, 5 outlier detection tasks (i.e., T1, T2, T3, T4 and T5) are explored by utilizing the designed approach. In each task, two different data sets are selected randomly. In the experiment, three evaluation metrics (i.e., accuracy, precision, and sensitivity) are employed for performance evaluation [44]. The accuracy, precision, and sensitivity of three approaches on 5 outlier detection tasks are summarized in Table I. As mentioned previously, in each outlier detection task, the source and target data sets are randomly selected from 5 data sets. In the table, the selected source and target data in each task are described as "(source data, target data)". According to Table I, the developed framework obtains the highest values in accuracy, precision, and sensitivity in all 5 tasks. The results also demonstrate the effectiveness of the regularization term of the developed approach.

The optimal values of 5 hyper-parameters selected by the PSO algorithm in each outlier task are listed in Table II. We can see in Tables I-II that the discovered hyper-parameters are different in each outlier detection task, which indicates that the solution space of each task is quite different. Thereby, there is a need to apply such kind of hyper-parameter optimization strategy for analyzing the sensor data collected in various working status. Comparing with two benchmark methods, the proposed PSO-embedded DTL outlier detection approach has better detection accuracy and exhibits satisfactory generalization ability.

TABLE II
THE OPTIMAL VALUES OF HYPER-PARAMETERS IN EACH OUTLIER TASK

| Task | $\delta$ | $\varepsilon$ | $\mu_1$ | $\mu_2$ | $\sigma$ |
|---|---|---|---|---|---|
| T1 | 2.30 | 1.38 | 1.56 | 1.06 | 63.90 |
| T2 | 1.07 | 1.14 | 1.50 | 4.84 | 57.51 |
| T3 | 1.73 | 1.53 | 0.81 | 5.39 | 74.81 |
| T4 | 1.54 | 1.01 | 1.17 | 5.08 | 72.06 |
| T5 | 1.28 | 1.47 | 0.72 | 5.18 | 69.45 |

## V. CONCLUSION

In this paper, a novel PSO-embedded DTL framework for outlier detection has been developed. A new DA strategy has been put forward to minimize the cross-domain discrepancies of both the marginal distribution and the conditional distribution. To be specific, in order to alleviate the data imbalance problem, the conditional domain discrepancies of normal instances and outliers have been adjusted by two separate coefficients. The PSO algorithm has been adopted to adjust the hyper-parameters of the proposed method. The developed DTL framework has been applied in the outlier detection tasks on WAAM data sets. Experimental results have shown satisfactory performance in the detection accuracy of the proposed approach. The developed framework has shown promising performance of outlier detection on the WAAM data sets collected from real-world manufacturing processes. Future work can be summarized into following five aspects: 1) applying the developed framework to the real-world outlier detection tasks in WAAM; 2) developing advanced optimization algorithms for hyper-parameters selection [45]–[47]; 3) adjusting the model structure [48], [49]; 4) designing appropriate regularization terms for the model [50], [51]; and 5) deploying the proposed framework to some other outlier detection tasks [52], [53].

## REFERENCES


[1] H. Alimohammadi and S. N. Chen, Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis, *Expert Systems with Applications*, vol. 191, art. no. 116371, 2022.

[2] J. Fang, Z. Wang, W. Liu, S. Lauria, N. Zeng, C. Prieto, F. Sikström and X. Liu, A new particle swarm optimization algorithm for outlier detection: Industrial data clustering in wire arc additive manufacturing, *IEEE Transactions on Automation Science and Engineering*, in press, DOI: 10.1109/TASE.2022.3230080.

[3] M. Gupta, J. Gao, C. C. Aggarwal and J. Han, Outlier Detection for Temporal Data: A Survey, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250-2267, 2014.

[4] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H. T. Zhang, H. Ding and J. Goncalves, Data driven discovery of cyber physical systems, *Nature Communications*, vol. 10, no. 1, pp. 1-9, 2019.

[5] A. Boukerche, L. Zheng and O. Alfandi, Outlier detection: Methods, models, and classification, *ACM Computing Surveys*, vol. 53, no. 3, pp. 1-37, 2020.

[6] G. Pang, C. Shen, L. Cao and A. V. D. Hengel, Deep learning for anomaly detection: A review, *ACM Computing Surveys*, vol. 54, no. 2, pp. 1-38, 2021.

[7] N. Yu, R. Yang and M. Huang, Deep common spatial pattern based motor imagery classification with improved objective function, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 73-84, 2022.

[8] G. Zhao, Y. Li and Q. Xu, From emotion AI to cognitive AI, *International Journal of Network Dynamics and Intelligence*, vol. 1, no. 1, pp. 65-72, 2022.

[9] D. Yang, J. Lu, H. Dong and Z. Hu, Pipeline signal feature extraction method based on multi-feature entropy fusion and local linear embedding, *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 407-416, 2022.

[10] C. M. Liang, Y. W. Li, Y. H. Liu, P. F. Wen and H. Yang, Segmentation and weight prediction of grape ear based on SFNet-ResNet18, *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 722-732, 2022.

[11] S. J. Pan and Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2009.

[12] K. Weiss, T. M. Khoshgoftaar and D. D. Wang, A survey of transfer learning, *Journal of Big Data*, vol. 3, art. no. 9, 2016.

[13] M. Long, H. Zhu, J. Wang and M. I. Jordan, Deep transfer learning with joint adaptation networks, In: *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, Aug. 2017, pp. 2208-2217.

[14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko and T. Darrell, Deep domain confusion: Maximizing for domain invariance, *arXiv preprint arXiv:1412.3474*, 2014.

[15] Y. Himeur, S. Al-Maadeed, H. Kheddar, N. Al-Maadeed, K. Abualsaud, A. Mohamed and T. Khattab, Video surveillance using deep transfer learning and deep domain adaptation: Towards better generalization, *Engineering Applications of Artificial Intelligence*, vol. 119, art. no. 105698, 2023.

[16] C. Fan, Y. Jin, P. Liu and W. Zhao, Transferable visual pattern memory network for domain adaptation in anomaly detection, *Engineering Applications of Artificial Intelligence*, vol. 121, art. no. 106013, 2023.

[17] S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, Domain adaptation via transfer component analysis, *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199-210, 2011.

[18] D. Wu, V. J. Lawhern, S. Gordon, B. J. Lance and C. T. Lin, Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (OwARR), *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1522-1535, 2017.

[19] J. Wang, Y. Chen, S. Hao, W. Feng and Z. Shen, Balanced distribution adaptation for transfer learning, In: *Proceedings of the 2017 IEEE International Conference on Data Mining*, New Orleans, USA, Nov. 2017, pp. 1129-1134.

[20] J. M. Johnson and T. M. Khoshgoftaar, Survey on deep learning with class imbalance, *Journal of Big Data*, vol. 6, art. no. 27, 2019.

[21] F. Thabtah, S. Hammoud, F. Kamalov and A. Gonsalves, Data imbalance in classification: Experimental evaluation, *Information Sciences*, vol. 513, pp. 429-441, 2020.

[22] Y. Liu, G. Yang, S. Qiao, M. Liu, L. Qu, N. Han, T. Wu, G. Yuan, T. Wu and Y. Peng, Imbalanced data classification: Using transfer learning and active sampling, *Engineering Applications of Artificial Intelligence*, vol. 117, art. no. 105621, 2023.

[23] M. Claesen and B. D. Moor, Hyperparameter search in machine learning, *arXiv preprint arXiv:1502.02127v2*, 2015.

[24] R. Andonie, Hyperparameter optimization in learning systems, *Journal of Membrane Computing*, vol. 1, pp. 279-291, 2019.

[25] J. Fang, W. Liu, L. Chen, S. Lauria, A. Miron and X. Liu, A survey of algorithms, applications and trends for particle swarm optimization, *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, 2023.

[26] E. H. Houssein, A. G. Gad, K. Hussain and P. N. Suganthan, Major advances in particle swarm optimization: Theory, analysis, and application, *Swarm and Evolutionary Computation*, vol. 63, art. no. 100868, 2021.

[27] X. Luo, Y. Yuan, S. Chen, N. Zeng and Z. Wang, Position-transitional particle swarm optimization-incorporated latent factor analysis, *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3958-3970, 2022.

[28] M. Long, J. Wang, G. Ding, S. J. Pan and P. S. Yu, Adaptation regularization: A general framework for transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076-1089, 2014.

[29] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang and Q. Yang, Transfer learning with dynamic distribution adaptation, *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 1, pp. 1-25, 2020.

[30] Q. Qian, Y. Qin, Y. Wang and F. Liu, A new deep transfer learning network based on convolutional auto-encoder for mechanical fault diagnosis, *Measurement*, vol. 178, art. no. 10935, 2021.

[31] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu and W. Zuo, Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, Jul. 2017, pp. 2272-2281.

[32] G. Ma, Z. Wang, W. Liu, J. Fang, Y. Zhang, H. Ding and Y. Yuan, Estimating the state of health for lithium-ion batteries: A particle swarm optimization-assisted deep domain adaptation approach, *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 7, pp. 1530-1543, 2023.

[33] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), *arXiv preprint arXiv:1606.08415*, 2016.

[34] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf and A. Smola, A kernel method for the two-sample-problem, In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2006, pp. 513-520.

[35] I. Gibson, D. Rosen and B. Stucker, *Additive Manufacturing Technologies*, Switzerland: Springer, 2021.

[36] K. Tafakkori, R. Tavakkoli-Moghaddam and A. Siadat, Sustainable negotiation-based nesting and scheduling in additive manufacturing systems: A case study and multi-objective meta-heuristic algorithms, *Engineering Applications of Artificial Intelligence*, vol. 112, art. no. 104836, 2022.

[37] Y. Yuan, G. Ma, C. Cheng, B. Zhou, H. Zhao, H. T. Zhang and H. Ding, A general end-to-end diagnosis framework for manufacturing systems, *National Science Review*, vol. 7, no. 2, pp. 418-429, 2020.

[38] B. Blakey-Milner, P. Gradl, G. Snedden, M. Brooks, J. Pitot, E. Lopez, M. Leary, F. Berto and A. Plessis, Metal additive manufacturing in aerospace: A review, *Materials & Design*, vol. 209, art. no. 110008, 2021.

[39] R. Kumar, M. Kumar and J. S. Chohan, The role of additive manufacturing for biomedical applications: A critical review, *Journal of Manufacturing Processes*, vol. 64, pp. 828-850, 2021.

[40] X. Sui, Julia R. Downing, M. C. Hersam and J. Chen, Additive manufacturing and applications of nanomaterial-based sensors, *Materialstoday*, vol. 48, pp. 135-154, 2021.

[41] F. Pourkamali-Anaraki, T. Nasrin, R. E. Jensen, A. M. Peterson and C. J. Hansen, Evaluation of classification models in limited data scenarios with application to additive manufacturing, *Engineering Applications of Artificial Intelligence*, vol. 126, art. no. 106983, 2023.

[42] C. Xia, Z. Pan, J. Polden, H. Li, Y. Xu, S. Chen and Y. Zhang, A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system, *Journal of Manufacturing Systems*, vol. 57, pp. 31-45, 2020.

[43] N. A. Rosli, M. R. Alkahari, M. F. Abdollah, S. Maidin, F. R. Ramli and S. G. Herawan, Review on effect of heat input for wire arc additive manufacturing process, *Journal of Materials Research and Technology*, vol. 11, pp. 2127-2145, 2021.

[44] C. Wang, Z. Wang, W. Liu, Y. Shen and H. Dong, A novel deep offline-to-online transfer learning framework for pipeline leakage detection with small samples, *IEEE Transactions on Instrumentation and Measurement*, vol. 72, art. no. 3503913, 2023.

[45] L. Xu, J. Du, B. Song and M. Cao, A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots, *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 134-141, 2022.

[46] H. Li, P. Wu, N. Zeng, Y. Liu and F. E. Alsaadi, A survey on parameter identification, state estimation and data analytics for lateral flow immunoassay: From systems science perspective, *International Journal of Systems Science*, vol. 53, no. 16, pp. 3556-3576, 2022.

[47] L. Yang and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing*, vol. 415, pp. 295-316, 2020.

[48] F. Yang, J. Li, H. Dong and Y. Shen, Proportional-integral-type estimator design for delayed recurrent neural networks under encoding-decoding mechanism, *International Journal of Systems Science*, vol. 53, no. 13, pp. 2729-2741, 2022.

[49] X. Li, Q. Song, Z. Zhao, Y. Liu and F. E. Alsaadi, Optimal control and zero-sum differential game for Hurwicz model considering singular systems with multifactor and uncertainty, *International Journal of Systems Science*, vol. 53, no. 7, pp. 1416-1435, 2022.

[50] Y. Li, C. Fu, F. Ding, Z. Huang and G. Lu, AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization, In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, Jun. 2020, pp. 11923-11932.

[51] X. Li, A. Abuduweili, H. Shi, P. Yang, D. Dou, H. Xiong and C. Xu, Semi-supervised transfer learning with hierarchical self-regularization, *Pattern Recognition*, vol. 144, art. no. 109831, 2023.

[52] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu and G. Chen, ECOD: Unsupervised outlier detection using empirical cumulative distribution functions, *IEEE Transactions on Knowledge and Data Engineering*, in press, DOI: 10.1109/TKDE.2022.3159580.

[53] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C. Lin and G. Fortino, Hyperparameter optimization in learning systems, *Information Fusion*, vol. 65, pp. 13-20, 2021.