This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

A Transfer-Learning-Assisted Role-Differentiated Approach for Industrial Outlier Detection under Label Noise

Jingzhong Fang[1], Zidong Wang[1], Weibo Liu[1], and Xiaohui Liu[1]
[1] Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom.
E-mails:{Jingzhong.Fang, Zidong.Wang, Weibo.Liu2, Xiaohui.Liu}@brunel.ac.uk

***Abstract*—In this paper, a novel role-differentiated learning with noisy label (RD-LNL) approach is proposed for industrial outlier detection. A leader-follower-inspired sample selection (LFSS) strategy is introduced to choose relatively "clean samples" for establishing a robust outlier detector against label noise. Specifically, a pre-trained deep learning model is employed as the leader network to guide the training of two follower deep learning models via a joint training manner, where a selection metric is designed to facilitate sample selection by leveraging both the training dynamics and the prediction discrepancy among the models. To further enhance the possibility of selecting potential clean samples, an adaptive selection scheme is put forward to adaptively adjust the clean sample selection ratio throughout the model training process by making full use of the loss characteristics of the samples. The proposed outlier detection approach is exploited in a real-world industrial outlier detection task with application to wire arc additive manufacturing (WAAM). Experimental results demonstrate the effectiveness of the developed RD-LNL approach for WAAM outlier detection in terms of detection accuracy.**

***Index Terms*—Outlier detection, learning with noisy labels, industrial data analysis, wire arc additive manufacturing.**

I. Introduction

In the era of big data, deep learning (DL) techniques have achieved great success across various domains. Notably, label quality is crucial for ensuring both the predictive accuracy and generalization ability of DL models. In many real-world applications, the obtained raw data are normally unlabeled and require annotation by human experts or automated tools. Due to the unexpected errors in automated labeling tools and mis-operation by human annotators, the obtained training data may exist data with incorrect labels. Such label noise would corrupt the training process and mislead the mapping between the feature space and the target space, thereby affecting the performance and reliability of the trained DL model.

To build a trustworthy DL model using data with label noise, learning with noisy labels (LNL) has become a recently popular topic in DL [1]. Serving as a prominent class of LNL approaches, sample selection focuses on identifying correctly-labeled samples (i.e., clean samples) for model training by using specific metrics (e.g., the training loss and similarity metrics). Compared with other classes of LNL approaches such as robust training and label correction, sample selection is capable of decreasing the negative influences from samples with label noise (i.e., noisy samples) and avoiding the misleading effects arising from label correction errors. A major advantage of sample selection is its simplicity and ease of implementation, which leads to the widespread adoption of sample selection in various applications for handling label noise in recent years [2]–[5].

Sample selection may suffer from incorrect selection, where clean samples are misclassified as noisy samples, especially when applied to large or complex data sets with label noise. During the model training process, noisy samples would cause error accumulation and introduce model biases, which results in degraded model performance. Recently, a number of sample selection variants have been proposed to improve the selection performance by introducing advanced training strategies or developing new network architectures [6]–[8]. For instance, in [7], the O2U-Net has been proposed for identifying noisy and clean samples by making use of the loss variation of each sample at different training stages. In [6], a well-known LNL approach named Co-teaching has been proposed, where the Siamese network structure is employed. Particularly, the potential clean samples identified by each network are utilized to update its peer network. The Siamese network structure has been widely adopted in sample selection by leveraging the unique learning capability of each network, offering complementary views on samples and improving selection accuracy. Unlike single-network methods, the Siamese-network-based sample selection (SNSS) approach leverages model diversity to reduce confirmation bias and improve robustness against label noise. Apart from the Co-teaching approach, various SNSS approaches, including Co-teaching+ [9], JoCoR [4], and DivideMix [8] have also achieved notable success in tackling the noisy label problem.

To reduce the computational cost, many SNSS methods employ identical and simple subnetworks, which would unfortunately constrain the model's learning and representation capabilities, especially when handling complex data sets. Moreover, identical architectures sometimes fail to provide diverse multi-view representations of data patterns, thereby limiting the network complementarity and impairing the model's ability to discriminate clean and noisy samples. Inspired by coordination strategies in multi-agent systems, the leader-follower framework proposed in [10] is a seemingly promising solution to break symmetry among subnetworks and promote diverse yet coordinated model learning behaviors. In the leader-

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

follower framework, role differentiation among subnetworks is achieved, where the leader network incorporates additional domain-specific knowledge (which is imparted to the follower networks to guide their training), thereby improving the overall capacity to identify and select clean samples.

The clean sample selection ratio (CSSR) plays a significant role in sample selection, which brings a trade-off between the inclusion of clean samples against the exclusion of noisy samples. Many existing sample selection approaches use a constant CSSR based on experimental experience, which demands domain knowledge and cannot guarantee consistent performance on various data sets with multi-modal inputs or partially labeled data. In Co-teaching, a dynamic selection strategy has been designed, where the CSSR decreases according to the stage of training to gradually exclude noisy data and retain clean samples. Unfortunately, the dynamic selection strategy in Co-teaching overlooks the variability in the training process across different data sets, potentially leading to premature exclusion of valuable samples or delayed removal of noisy ones. To enhance selection efficiency and fully utilize potentially clean samples, a reasonable idea is to adaptively adjust the CSSR based on both the training state information and the loss characteristics.

Motivated by above discussions, in this paper, a role-differentiated LNL (RD-LNL) approach is put forward for industrial outlier detection. A leader-follower-inspired sample selection (LFSS) strategy is proposed for identifying potential clean samples. Then, the chosen clean samples are fed into the deep neural networks (DNNs) for outlier detection. In particular, three DNNs are trained collaboratively. Here, a leader network is pre-trained on clean auxiliary data and guides the joint training of two follower networks by fully leveraging its domain knowledge. A selection metric is introduced that integrates both the training dynamics and the prediction divergence across the DNNs. According to the selection metric, an adaptive selection scheme is proposed to adaptively adjust the CSSR during the training process.

The contributions of this paper lie in the following threefold:

1) A LFSS strategy is proposed for training the outlier detection model with selected potential clean samples, where a leader network and two follower networks are trained jointly for sample selection by leveraging the domain knowledge of the leader network.
2) An adaptive selection scheme is introduced to adjust the CSSR adaptively through the model training process, which makes use of the loss characteristics of the training samples.
3) The developed RD-LNL approach is successfully applied to a real-world industrial outlier detection application using data collected from a wire arc additive manufacturing (WAAM) pilot line. Experimental results verify the practical utility of the developed RD-LNL approach for handling data with label noise.

The remaining sections of this paper are organized as follows. In Section II, the developed RD-LNL approach is introduced. Experimental results for WAAM outlier detection under label noise are presented in Section III. Finally, conclusions are presented in Section IV.

II. Methodology

A. Motivation

Many SNSS approaches use identical network architectures with random initialization to encourage diverse learning behaviors and reduce confirmation bias [4], [6]. During the model training, samples are selected based on the prediction consensus of the networks. Nevertheless, identical architectures with random initialization may still lack sufficient representational diversity, thereby decreasing the selection quality. As a role-differentiated framework, the leader-follower framework is capable of integrating domain-specific knowledge and model diversity to support effective sample selection [10]. Besides, the variation between the leader and follower networks brings multiple perspectives of the learning process, which helps reduce confirmation bias and improve the reliability of selected samples.

As a critical factor, the CSSR directly determines how many samples are used for model training at each training epoch, and thus governs the balance between preserving clean samples and excluding noisy samples. It is worth mentioning that most existing approaches adopt the fixed selection ratio or the dynamic selection ratio based on training epochs. Such selection ratios may overlook the fact that training dynamics vary significantly across data sets, and thus decrease the generalization ability of the approaches. To address the above mentioned limitation, a seemingly reasonable solution is to consider both the training epochs and the sample loss characteristics to adaptively select clean samples during training.

In this paper, an RD-LNL approach is developed for industrial outlier detection with noisy labels. An LFSS strategy is introduced to select clean samples to train the outlier detector. To be specific, a pre-trained network is employed as the leader network and is utilized to guide the joint training process with two follower networks. A selection metric is designed for sample selection by integrating both training behavior and prediction divergence across the networks. Based on the selection metric, an adaptive selection scheme is put forward to improve the quality of selected samples by adaptively adjusting the CSSR during the training process.

B. Overview of the RD-LNL Approach

The developed RD-LNL approach is shown in Fig. 1. The RD-LNL approach consists of three networks: Network 3 serves as the leader, while Network 1 and Network 2 act as followers.

The Transformer is utilized as the backbone of the leader network in order to capture long-range dependencies within the time series and improve the generalization capability. The Transformer is composed of several identical encoder layers, where each layer includes a multi-head attention mechanism and a lightweight 1D convolutional module to enhance local temporal feature extraction.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

The convolutional neural network (CNN) serves as the backbone of the follower networks, enabling efficient training and effective extraction of local features. Each CNN consists of multiple standard convolutional modules, each containing a convolutional layer followed by batch normalization. Residual connections are employed in both the encoder layers and the convolutional modules to facilitate gradient flow and stabilize training. Three identical multi-layer perceptrons (MLPs) are employed as the classifiers, each attached to the output of one network.

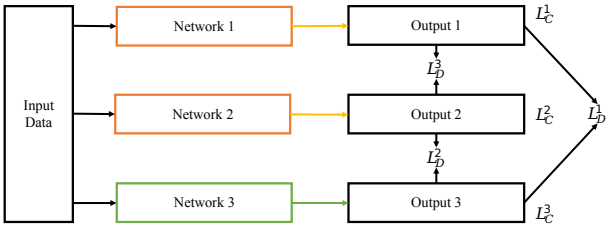

Fig. 1. The proposed RD-LNL approach

C. Loss Function

1) Classification Loss: The cross-entropy is applied to compute the classification loss for Network 1, Network 2, and Network 3 (i.e., L_C^1, L_C^2, and L_C^3, respectively), which are given by:

$$L_C^1 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(\hat{y}_{ij}^1), \tag{1}$$

$$L_C^2 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(\hat{y}_{ij}^2), \tag{2}$$

$$L_C^3 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(\hat{y}_{ij}^3), \tag{3}$$

where N indicates the sample size of current mini-batch; M denotes the total number of classes; y_{ij} is the jth element of the label vector corresponding to the ith sample; and $\hat{y}_{ij}^1$, $\hat{y}_{ij}^2$, and $\hat{y}_{ij}^3$ are the jth element of the predicted output for the ith sample of Network 1, Network 2, and Network 3, respectively.

2) Prediction Divergence: To mitigate the confirmation bias of each network and enhance collaborative sample selection performance, in this paper, the prediction divergences between all pairs of networks are calculated, where the Kullback-Leibler (KL) divergence and Jensen–Shannon (JS) divergence are employed. Specifically, the KL divergence is utilized to measure the output discrepancies between the leader network and each of the two follower networks (i.e., L_D^{13} and L_D^{23}), and the JS divergence is applied to calculate the output discrepancy between two follower networks (i.e., L_D^{12}). The prediction divergences between all pairs of networks are calculated by:

$$L_D^{13} = \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(P_i^1 \parallel P_i^3) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} P_{ij}^1 \log\left(\frac{P_{ij}^1}{P_{ij}^3}\right), \tag{4}$$

$$L_D^{23} = \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}(P_i^2 \parallel P_i^3) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} P_{ij}^2 \log\left(\frac{P_{ij}^2}{P_{ij}^3}\right), \tag{5}$$

$$\begin{aligned} L_D^{12} &= \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{JS}}(P_i^1 \parallel P_i^2) \\ &= \frac{1}{2}\left[\sum_{i=1}^{N} D_{\mathrm{KL}}(P_i^1 \parallel Q_i) + \sum_{i=1}^{N} D_{\mathrm{KL}}(P_i^2 \parallel Q_i)\right], \end{aligned} \tag{6}$$

where $Q_i = \frac{1}{2}(P_i^1 + P_i^2)$ is the average distribution of the two follower networks for the ith sample; L_D^{13} and L_D^{23} denote the prediction discrepancy between Network 1 and Network 3, and Network 2 and Network 3, respectively; L_D^{12} represents the prediction discrepancy between Network 1 and Network 2; P_i^1, P_i^2 and P_i^3 are the distribution of the output probability of the ith sample from Network 1, Network 2, and Network 3, respectively; and P_{ij}^1, P_{ij}^2 and P_{ij}^3 are the output probability for the ith sample with the jth class of Network 1, Network 2, and Network 3, respectively.

Remark 1: Network 3, denoted as the leader network, is obtained via pre-training and fine-tuning. Compared with the follower networks that are trained directly on the noisy labeled data set, the leader network has better feature representation and discriminative capabilities. It is worth mentioning that KL divergence measures the information loss introduced when approximating one probability distribution with another, making the direction of comparison a critical factor due to its asymmetric nature. To guide the follower networks toward the prediction behavior of the leader, in this paper, the prediction divergences L_D^{13} and L_D^{23} are calculated according to (4) and (5). Furthermore, the JS divergence is applied in calculating L_D^{12} based on (6) due to its symmetric and bounded nature to ensure balanced comparison of the output prediction between the two follower networks.

3) Overall Loss Function: The overall loss functions of the three networks (i.e., L_1, L_2, and L_3) are given by:

$$L_1 = \alpha_1 L_C^1 + \beta_1 L_D^{13} + \gamma_1 L_D^{12}, \tag{7}$$

$$L_2 = \alpha_2 L_C^2 + \beta_2 L_D^{23} + \gamma_2 L_D^{12}, \tag{8}$$

$$L_3 = L_C^3, \tag{9}$$

where α_1, α_2, β_1, β_2, γ_1 and γ_2 are hyper-parameters for balancing the contributions of the classification loss and the prediction divergences.

D. Training Scheme

Fig. 2 demonstrates the training scheme of the RD-LNL approach.

1) Pre-training of Leader Network: In the proposed RD-LNL approach, the feature extraction and representation abilities of the leader network are significantly important. Thanks to its generalization and knowledge transfer capability, transfer learning becomes an effective solution to obtain the leader network using a clean auxiliary data set. The leader network is pre-trained on the clean source domain to learn transferable representations for the downstream noisy task.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

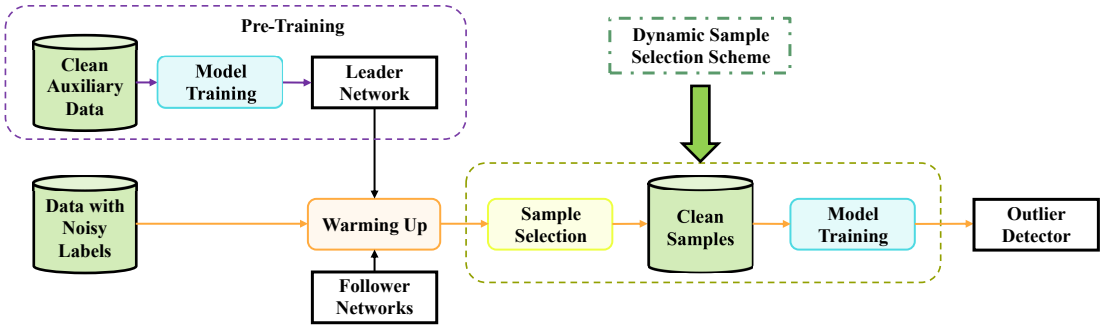

Fig. 2. The training process of the RD-LNL approach

2) Joint Training on Noisy Labeled Data: After obtaining the leader network, the follower networks are trained jointly with the leader network. The joint training process consists of two main stages, namely warm-up and sample selection. Specifically, the leader network is initialized using the weights obtained through pre-training, and the follower networks are initialized randomly. In the warm-up stage, the leader network and the follower networks are trained on the noisy data for several epochs to ensure a stable initialization and learn initial representations. During warm-up, the leader network is fine-tuned to adapt its pre-trained knowledge to the noisy target domain. In the sample selection stage, the clean samples are selected using the adaptive selection scheme in each epoch and are then used to update three networks.

3) The Adaptive Selection scheme: In this paper, the "small-loss criterion" is utilized for sample selection. To be specific, a selection metric is introduced for selecting potential clean samples based on three networks, where the loss values of each sample from three networks during the training are leveraged for distinguishing clean samples from noisy ones. The selection metric of each sample $L_S(i)$ can be calculated by:

$$L_S(i) = \mu_1 L_1(i) + \mu_2 L_2(i) + \mu_3 L_3(i), \tag{10}$$

where $L_1(i)$, $L_2(i)$, and $L_3(i)$ are per-sample loss of three networks; and μ_1, μ_2, and μ_3 are three hyper-parameters for balancing the contributions of each term.

The CSSR $R(e)$ is adjusted adaptively during the training process based on the training epoch and selection metric, which is calculated by:

$$R(e) = \max\left\{\left(1 - \frac{e}{E}\cdot\hat{r}\right)\cdot\frac{\sigma_e}{\sigma_0},\ 1-\hat{r}\right\}, \tag{11}$$

where e and E are the current and total training epochs, respectively; σ_e and σ_0 denote the standard deviation of values of selection metric across all samples in the mini-batch at the eth epoch and the initial training epoch, respectively; and $\hat{r}$ is the estimated noise ratio derived from the validation set.

4) Training Procedure: The training process of the RD-LNL approach is detailed in Algorithm 1. After the training process, the leader network is then utilized for outlier detection tasks.

III. WAAM Outlier Detection

Recognized as a cost-effective additive manufacturing technique, WAAM fabricates metal parts by employing an electric arc as the heat source to deposit wire feedstock in a layer-by-layer manner. WAAM is highly applicable for producing large and complex metal structures. So far, WAAM has been deployed in a wide range of industrial applications owing to its high material efficiency and flexible design capability [11]. It should be noted that the electric arc plays an important role in WAAM, which significantly affects the process stability. In real-world scenarios, the characteristics of the wire arc such as welding current and voltage may experience sudden changes, which can negatively affect the melt pool behavior and result in defects in the fabricated parts. The aim of the experiments is to detect such sudden changes (i.e., outliers) in order to ensure the manufacturing quality and further improve the manufacturing process.

A. Data Preparation

1) WAAM Data Sets: The data sets used in this study are obtained from a real-world WAAM pilot line, where each data set records the key process parameters (e.g., welding current and welding voltage) of a distinct WAAM task. Fig. 3 presents a visualization of the WAAM data. Ground-truth labels are manually annotated by domain experts, classifying each sample as either "Normal" or "Outlier" based on operational characteristics. The detailed description of the WAAM data sets can be found in [12].

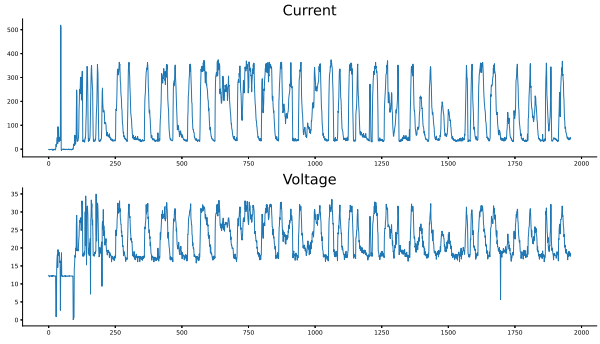

Fig. 3. Visualizations of the current and voltage of the WAAM data

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

Algorithm 1: Training Procedure of the RD-LNL Approach

Input: Noisy training data set $\tilde{\mathcal{D}}$, clean auxiliary data set $\mathcal{D}$, number of pre-training epochs E_P, number of warm-up epochs E_W, number of training epochs E, batch size B, hyper-parameters μ_1, μ_2 and μ_3.

Output: Model parameters of three networks θ_1, θ_2 and θ_3.

```
Randomly initialize θ1.
for e ← 0 to E_P do
    for n ← 0, 1, ··· , |D|/B do
        Randomly draw a mini-batch.
        Calculate L3 based on (9).
        Update θ1.
    end
end
Retain θ1 and randomly initialize θ2 and θ3.
for e ← 0 to E_W do
    for n ← 0, 1, ··· , |D̃|/B do
        Randomly draw a mini-batch.
        Calculate L1, L2 and L3 based on (7), (8) and (9).
        Update θ1, θ2 and θ3.
    end
end
Retain θ1, θ2 and θ3.
for e ← 0 to E do
    for n ← 0, 1, ··· , |D̃|/B do
        Randomly draw a mini-batch.
        Calculate selection metric L_S(i) based on (10).
        Calculate the selection ratio R(e) based on (11).
        Obtain clean samples D̃_C.
        Calculate L1, L2 and L3 on D̃_C based on (7), (8) and (9).
        Update θ1, θ2 and θ3.
    end
end
```

2) Data Pre-Processing: In this paper, a sliding window method is applied to segment each data set, with a window size of 10 and a stride of 5. The label for each segment is determined based on the labels of its constituent instances: a segment is considered "Normal" only if all included instances are labeled as such; otherwise, it is marked as an "Outlier". For classification purposes, all labels are converted into one-hot encoded vectors. Each processed data set is subsequently divided into training and testing subsets in a 70 : 30 ratio. To ensure scale consistency and enhance model performance, min-max normalization is applied to both the training sets and the testing sets.

B. Implementation Details

1) Model Configurations: The Transformer encoder of the leader network consists of 3 encoder layers. The CNNs of the two leader networks both consist of 5 convolutional modules. The classifier of each network is a 3-layer MLP. To ensure fair and robust evaluation, all hyper-parameters are selected through grid search on a separate validation set. The hyper-parameters are tuned within predefined ranges based on the best validation performance, and the same selection procedure is applied consistently across all selected approaches. To be specific, the hyper-parameters α_1, α_2, β_1, β_2, γ_1 and γ_2 in the loss functions are set as 0.35, 0.35, 0.7, 0.7, 0.25 and 0.25, respectively. The hyper-parameters μ_1, μ_1 and μ_1 in the selection metric are set to be 0.6, 0.2 and 0.2, respectively. The Adam optimizer is adopted and the learning rate is configured as 0.0001. The epoch numbers of the pre-training process, the warm-up process and the training process are 30, 10 and 50, respectively.

2) Experimental Settings: All experiments are performed under a consistent environment: Ubuntu 20.04.6, PyTorch 2.5.1, Python 3.9.21, and CUDA 12.3, using hardware with an NVIDIA RTX A6000 GPU (48 GB) and an Intel Xeon Silver 4214R processor to ensure fair comparisons. Each experiment is repeated five times, and the mean values of all evaluation metrics are presented to minimize the effect of random variation.

To verify the performance of the developed RD-LNL approach, two standard DL-based outlier detection approaches and two representative LNL approaches (e.g., Co-teaching and JoCoR) are selected for comparison. Specifically, in two standard DL-based outlier detection approaches, the CNN and the Transformer are employed as the backbone and connected with two identical classifiers, respectively.

Four evaluation metrics (e.g., accuracy, precision, recall, and F1-score) are applied to evaluate the effectiveness of the proposed RD-LNL approach for outlier detection under label noise. In the experiments, a clean auxiliary data set is selected for pre-training the leader network, and the other four data sets are used as the noisy training data. Extensive experiments are conducted under various noise ratios denoted by ϕ to evaluate the outlier detection performance under different levels of label noise. Each data set is evaluated under noise ratios ϕ of 30%, 40% and 50%.

C. Results and Discussion

The outlier detection results of the selected approaches on four WAAM data sets with various noise ratios are summarized in Table I. As shown by the results, the RD-LNL approach shows competitive performance across four tasks under various noise ratios. Detailed analysis of the results is provided below based on the noise ratio.

In the experiments under a noise ratio of 30%, the selected LNL approaches as well as two standard approaches all exhibit satisfactory performance on four data sets. It should be noticed that Tasks 1 and 2, the JoCoR reaches higher

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI: 10.1109/ICAC65379.2025.11196672, 2025 30th International Conference on Automation and Computing (ICAC)

TABLE I
OUTLIER DETECTION RESULTS ON NOISY WAAM DATA SETS WITH DIFFERENT NOISE RATIOS

Approaches	Metrics (%)	Task 1			Task 2			Task 3			Task 4		
		30%	40%	50%	30%	40%	50%	30%	40%	50%	30%	40%	50%
Co-teaching	Accuracy	96.77	89.01	74.76	95.22	88.43	72.11	95.12	88.82	73.57	93.80	88.65	73.52
	Precision	96.10	90.82	50.33	94.26	82.04	52.19	95.86	82.33	53.81	88.54	78.93	48.90
	Recall	92.05	90.37	77.62	93.64	88.59	76.36	93.06	88.97	69.31	94.22	79.90	70.55
	F1 Score	94.28	90.38	62.28	93.05	86.87	58.15	94.11	87.76	60.31	90.45	70.47	49.73
JoCoR	Accuracy	97.13	93.98	79.00	96.09	93.90	75.50	95.63	93.12	77.20	94.63	92.01	69.58
	Precision	97.29	93.50	48.55	96.00	93.01	46.01	96.74	91.49	45.75	89.91	79.16	34.01
	Recall	92.30	89.18	66.41	95.12	93.43	79.85	95.53	92.61	90.08	96.42	94.55	95.53
	F1 Score	94.57	91.39	56.16	95.68	92.65	60.59	96.01	92.41	61.09	93.12	87.12	50.16
Standard (CNN)	Accuracy	87.24	61.90	39.23	85.46	58.92	25.07	73.72	53.34	30.64	71.14	56.98	10.63
	Precision	83.29	62.79	55.53	80.48	75.82	36.90	69.53	56.28	42.27	69.24	59.14	26.49
	Recall	88.09	85.34	31.71	93.12	45.49	46.81	90.79	78.26	45.96	72.62	36.87	18.94
	F1 Score	90.02	72.35	40.37	87.45	56.84	41.21	81.71	66.41	43.72	80.64	53.67	22.24
Standard (Transformer)	Accuracy	89.16	63.36	41.65	87.66	60.96	27.19	75.71	55.34	32.24	72.39	58.35	12.74
	Precision	85.76	64.44	56.93	82.15	77.83	38.44	70.44	58.56	43.71	70.00	61.52	28.25
	Recall	89.40	86.11	34.05	94.70	47.73	47.69	91.70	79.54	47.33	73.41	38.94	20.94
	F1 Score	92.09	74.33	42.77	89.92	58.52	42.55	82.27	67.56	45.38	81.79	55.74	24.39
RD-LNL (Ours)	Accuracy	96.98	94.20	85.52	95.40	94.07	81.80	95.85	92.85	82.49	95.16	93.85	78.90
	Precision	96.17	93.85	86.96	96.95	94.04	86.80	96.82	92.95	88.60	96.70	93.15	87.96
	Recall	97.02	93.02	87.64	95.25	92.11	81.71	95.66	94.33	85.85	96.66	92.27	88.37
	F1 Score	96.37	95.09	86.40	96.28	94.23	84.22	96.14	92.70	86.55	96.29	93.22	87.78

accuracy than the RD-LNL. On the other two tasks, the RD-LNL approach outperforms the selected approaches in terms of detection accuracy and F1 score. At a noise level of 40%, both standard methods exhibit noticeable performance drops. According to the results, the selected LNL approaches still show reliable performance. On task 3, the accuracy of the JoCoR is higher than RD-LNL. Nevertheless, the RD-LNL approach maintains superior performance across the remaining tasks, which confirms its effectiveness in detecting outliers under noisy labels.

In the scenario with a 50% noise ratio, the RD-LNL approach also demonstrates strong performance, while all selected approaches suffer significant degradation across the four datasets. In particular, the accuracy and F1 scores of the selected approaches do not exceed 80% and 63%, respectively. In contrast, RD-LNL achieves results on all datasets, highlighting its robustness in the presence of moderate label noise.

IV. CONCLUSION

In this paper, an RD-LNL approach has been developed for industrial outlier detection under noisy labels. An LFSS strategy has been proposed to train the outlier detector. A selection metric has been designed for identifying clean samples. In addition, an adaptive selection scheme has been put forward based on the selection metric for adjusting the CSSR. Experimental results for outlier detection on real-world data sets have demonstrated satisfactory performance of the developed approach.

REFERENCES

[1] J. Shin, J. Won, H. Lee and J. Lee, A review on label cleaning techniques for learning with noisy labels, *ICT Express*, vol. 10, no. 6, pp. 1315-1330, 2024.

[2] C. Cheng, X. Liu, B. Zhou and Y. Yuan, Intelligent fault diagnosis with noisy labels via semisupervised learning on industrial time series, *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 7724-7732, 2023.

[3] X. Qin, P. Yao, M. Liu, X. Cheng, F. Shi and L. Guo, Robust classification of incomplete time series with noisy labels, In: *Proceedings of the 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Tianjin, China, May. 2024, pp. 2620-2625.

[4] H. Wei, L. Feng, X. Chen and B. An, Combating noisy labels by agreement: A joint training method with co-regularization, In: *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, Jun. 2020, pp. 13723-13732.

[5] S. Wu, T. Zhou, Y. Du, J. Yu, B. Han and T. Liu, A time-consistency curriculum for learning from instance-dependent noisy labels, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 7, pp. 4830-4842, 2024.

[6] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang and M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montreal, Canada, Dec. 2018, vol. 31.

[7] J. Huang, L. Qu, R. Jia and B. Zhao, O2U-Net: A simple noisy label detection approach for deep neural networks, In: *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 3325-3333.

[8] J. Li, R. Socher and S. C. H. Hoi, DivideMix: Learning with noisy labels as semi-supervised learning, *arXiv preprint arXiv:2002.07394*, 2020.

[9] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang and M. Sugiyama, How does disagreement help generalization against label corruption? In: *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, Jun. 2019, pp. 7164-7173.

[10] S. Dai, S. He, X. Chen and X. Jin, Adaptive leader–follower formation control of nonholonomic mobile robots with prescribed transient and steady-state performance, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3662-3671, 2020.

[11] A. Hamrani, F. Bouarab, A. Agarwal, K. Ju and H. Akbarzadeh, Advancements and applications of multiple wire processes in additive manufacturing: a comprehensive systematic review, *Virtual and Physical Prototyping*, vol. 18, no. 1, art. no. e2273303, 2023.

[12] J. Fang, Z. Wang, W. Liu, L. Chen and X. Liu, A new particle-swarm-optimization-assisted deep transfer learning framework with applications to outlier detection in additive manufacturing, *Engineering Applications of Artificial Intelligence*, vol. 131, art. no. 107700, 2024.