Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

Label-Noise-Resistant Time-Series Classification With Self-Supervised Label Correction

Yimeng He, Zidong Wang *Fellow, IEEE*, Weibo Liu *Member, IEEE*, Jingzhong Fang *Member, IEEE*, Linwei Chen, and Zhihuan Song

***Abstract*—The reliable operation of industrial systems requires not only the prompt detection of faults but also their accurate classification into the appropriate categories. At present, numerous data-driven industrial fault detection and diagnosis models, which have been developed based on historical fault data, frequently neglect the issue of label noise. When labels are corrupted by noise, a significant degradation in the performance of industrial fault detection models can be observed. In this paper, a label-noise-resistant time-series classification method based on consistency-driven label correction, referred to as LNRTSC, is proposed. First, an attention-based temporal correlation-enhanced (TCE) encoder is introduced to extract low-dimensional representations of industrial time series. Then, label confidence, which is assessed based on local label consistency, is utilized to correct noisy labels during training. In addition, a two-stage self-supervised enhancement strategy is designed to guarantee the reliability of the corrected labels. Specifically, a reconstruction loss term is introduced to assist feature extraction in the warming-up stage, and a newly-designed contrastive loss term is added to the loss function for the LNL training stage, which mitigates the effect of false negatives. Finally, the effectiveness of the LNRTSC method is validated on the Tennessee Eastman process and the SEU-gearbox dataset. When compared to peer methods, the LNRTSC approach demonstrates substantial improvements in fault classification performance on corrupted data.**

***Index Terms*—Learning with noisy labels, label correction, self-supervised learning, time-series classification, industrial fault detection.**

I. Introduction

Industrial fault classification plays a critical role in industrial fault detection and diagnosis, which ensures the safety, reliability, and efficiency of industrial production [4], [20], [29]. As the modern industrial systems become increasingly complex, even minor faults may lead to substantial economic losses or safety hazards. The accurate classification of faults into their respective categories facilitates prompt fault diagnosis and system recovery [10], [27], [34].

With the widespread adoption of large-scale sensor networks and Industrial Internet of Things technologies, the volume of available industrial process data has expanded exponentially, facilitating the advancement and application of data-driven fault classification approaches [11], [13], [15], [26]. As deep learning technologies continue to evolve, the fault classification problem is predominantly formulated as a time-series classification task by most data-driven approaches [2], [21]. These methods aim to train a classifier based on representations extracted from industrial process time series and the corresponding fault labels.

Among the most widely adopted deep learning techniques, one-dimensional convolutional neural networks (1D CNNs) are particularly effective in hierarchical information extraction and capturing local patterns, making them extensively utilized for fault identification in equipment such as motors and turbines [9], [10], [35], [37]. Long short-term memory (LSTM) networks, which can flexibly retain, update, or discard information, effectively model long-term dependencies, rendering them suitable for detecting faults over extended time periods [38]. Transformer-based models [25], leveraging self-attention mechanisms, are capable of handling long-range dependencies while enabling parallel computation, thereby delivering superior performance on large-scale datasets from complex industrial systems. Overall, these deep learning-based approaches facilitate accurate, scalable, and automated fault detection and diagnosis in modern industrial applications. However, existing deep learning-based fault detection and diagnosis methods mostly rely on monitoring statistics (e.g., reconstruction errors) to assess the statistical differences between faulty and normal samples, while seldom explicitly using changes in long-term temporal dependencies as diagnostic criteria. Therefore, the fault detection performances of these methods are limited and unreliable in complex noisy situations.

The aforementioned data-driven methods usually exhibit satisfactory performance in time-series analysis and fault classification, but their effectiveness is highly dependent on the quality of historical data and labels. However, in real-world scenarios, label noise is a prevalent issue that can be introduced by various factors (e.g., expert disagreements, sensor-level ambiguity, inconsistent labeling protocols, process drifts, abnormal operating conditions and ambiguous boundary definitions between classes), thereby compromising the reliability of training data [30], [31]. Consequently, the development of a label-noise-resistant time-series classification model is essential for enhancing the robustness of industrial fault detection and diagnosis.

This work was supported in part by the National Natural Science Foundation of China under Grant 62473103, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany *(Corresponding author: Zhihuan Song)*.

Yimeng He is with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: 12032033@zju.edu.cn).

Zidong Wang, Weibo Liu, Jingzhong Fang and Linwei Chen are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom (e-mail: Zidong.Wang@brunel.ac.uk).

Zhihuan Song is with Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China (e-mail: songzhihuan@gdupt.edu.cn).

Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

To mitigate the adverse effects of noisy labels on deep learning models, numerous *learning with noisy labels* (LNL) methods have been proposed in recent years [30], [31], which can primarily be categorized into two groups based on their underlying principles. The first category consists mainly of sample selection [14], [17] and label correction methods [1], [33], both of which aim to enhance data quality. In sample selection-based methods, clean and noisy samples are initially identified using loss values or other metrics. The noisy samples are then gradually excluded during training. For instance, in [14], samples with small loss values have been considered informative and used to update the models. While sample selection-based methods have gained popularity in addressing the issue of noisy labels, they are less suitable when data availability is limited. To maximize the utilization of information within a given dataset, label correction-based methods first assess the label confidence for each sample and then refurbish noisy labels based on the model predictions. For example, in [1], the proposed LNL method has been developed to estimate label confidence by evaluating local label consistency, and subsequently correct the labels using both the label confidence and predictive values. However, after training several epochs of warming up, the model's predictions may still be inaccurate for some hard samples that are difficult to be classified, which compromises the reliability of the corrected labels. Considering the limitations mentioned above, various methods have been developed to minimize the impact of noisy labels in different ways, mainly by refining model architectures [12], designing specialized loss functions [23], and incorporating regularization techniques [19].

The paper aims to address the time-series classification problem under label noise and propose a label-noise-resistant time-series classification method (LNRTSC). Given that the dependencies between time steps reflect the underlying dynamic characteristics of multivariate time series, the LNRTSC includes a temporal correlation-enhanced (TCE) encoder. The TCE encoder innovatively incorporates the multi-head attention [24] score matrix as an explicit feature, which is concatenated with the extracted features, enabling the classifier to learn the variations in long-term dependencies under different fault conditions. In addition, to effectively mitigate the impact of noisy labels while maximizing data utilization, the proposed LNRTSC method is developed within a consistency-driven label correction framework. Furthermore, a two-stage self-supervised enhancement strategy is designed to guarantee the reliability of the corrected labels. Specifically, during the warming-up stage, a reconstruction loss term is introduced to assist feature extraction. After that, a novel contrastive loss term is added to the loss function for the LNL training stage, which mitigates the effect of false negatives.

The novelty of our method can be summarized as follows:

1) A novel attention-based temporal correlation-enhanced encoder (TCE) is introduced to produce low-dimensional representations for industrial time series.
2) A consistency-driven label correction framework is proposed to evaluate label confidence and refurbish the labels for model training.
3) A two-stage self-supervised enhancement strategy is designed to guarantee the reliability of the corrected labels.

The remainder of this paper is structured as follows. Section II provides an overview of recent related work on LNL. Section III details the module structures and underlying principles of the proposed LNRTSC method. In Section IV, the Tennessee Eastman process (TEP) and the SEU-gearbox dataset are utilized to validate the effectiveness of LNRTSC. Finally, conclusions and directions for future research are presented in Section V.

II. Related Work

In this section, we introduce the recent LNL-related studies with focus on approaches that integrate various learning techniques and demonstrate competitive performance in addressing the challenges posed by noisy labels.

A representative approach, known as Co-teaching, has been introduced in [14], where two networks are trained simultaneously, with each selecting small-loss samples to update its peer network. Another notable method, joint training with co-regularization (JoCoR), has been proposed in [36]. The JoCoR method jointly trains two neural networks using a loss function that combines conventional supervised learning loss with a co-regularization term, encouraging agreement between the models. The intrinsic similarity, which is measured by the contrastive InfoNCE loss, has been utilized by the co-learning approach [32], to mitigate the impact of noisy labels on model performance. In [16], a twins contrastive learning framework has been proposed, which employs two twin branches and leverages contrastive learning to pull together representations of the same samples while pushing apart representations of different samples, thereby obtaining more robust and discriminative representations. In the aforementioned work, the positive pairs are various transformations of one certain sample, while the negative pairs are transformations of two different samples. However, this technique aggregates same-class samples relatively slowly, and its effectiveness may be limited when applied to time-series data. In [28], a supervised contrastive learning strategy has been employed to directly impose supervision in the hidden space for label correction. In this approach, samples sharing the same corrected soft labels are treated as positive pairs, while those with different corrected soft labels are considered negative pairs. However, the proposed strategy in [28] may be affected by false negatives, as the initial soft labels used to form positive pairs are not always correct. To solve the challenges, our LNRTSC method newly designs the positive and the negative pairs based on corrected labels and nearest sample neighbors to avoid false negative problem.

The DivideMix approach has been proposed in [22], which formulates noisy label learning as a semi-supervised problem by separating samples into clean and noisy sets using a two-component Gaussian Mixture Model (GMM) over their losses. Unlike DivideMix, the proposed LNRTSC method fits the local label consistency (LLC) scores and evaluates label confidence using the Dirichlet Process Gaussian Mixture Model (DP-GMM) [3]. DP-GMM can automatically determine the number of Gaussian components, making it more flexible and better suited for complex datasets with hard samples. Recent studies have explored multi-granularity strategies to mitigate the impact of noisy labels in fault diagnosis. For example, the MgCNL method has incorporated a label-confidence evaluation technique based on granular-ball computing [5], enabling the selection of high-confidence samples from noisy datasets for supervised learning and thus avoiding the negative impact of noisy labels. Multi-granularity label-correction approaches such as multi-granularity cluster fusion (MgCF) [6], the multi-granularity labeling (MgL) strategy [7], and the multi-granularity ball-intra fusion (MgBIF) [8] have also been proposed to more effectively suppress the misleading effects of noisy labels.

To evaluate the effectiveness of the LNRTSC method, the JoCoR, the Co-teaching and the DivideMix methods are selected as comparative approaches. To address the time-series classification problem, the Transformer and the 1D CNN are employed as backbone networks for both methods, denoted as JoCoR (CNN), JoCoR (Transformer), Co-teaching (CNN), Co-teaching (Transformer), and DivideMix (CNN), and DivideMix (Transformer).

III. Proposed Method

Since merely filtering out samples with noisy labels would inevitably lead to the loss of valuable information, the proposed LNRTSC method is developed within a consistency-driven label correction framework. The LNRTSC method consists of two stages. In the first stage, the model is warmed up for several epochs using the original noisy labels. After this warming-up stage, the labels are corrected based on label confidence [1] and model predictions, and the corrected labels are then used to supervise the LNL training stage. The overall pipeline of the proposed LNRTSC method is shown in Algorithm 1. To facilitate effective feature extraction, an attention-based temporal correlation-enhanced (TCE) encoder is newly designed to learn high-quality low-dimensional representations of time series. Additionally, a two-stage self-supervised enhancement strategy is employed to ensure the reliability of the corrected labels, introducing a reconstruction loss term to enhance feature extraction and a novel contrastive loss term to mitigate false negatives.

A. Problem statement

The paper aims to develop a novel label-noise-resistant time-series classification method, which is capable of accurately categorizing time series data into its respective category while mitigating the impact of the noisy labels.

The established model of our method consists of three modules: a TCE encoder $g(\cdot)$, a classifier $f(\cdot)$ and a decoder $h(\cdot)$, as illustrated in Fig. 1. Considering the temporal dependencies of the time series data, the input is serialized before transmitted into $g(\cdot)$, the serialized sample is denoted as $\boldsymbol{x}_{i:(i+T)}$ (T is the length of the sliding window). The noisy training dataset is denoted as $\tilde{D} = \{\boldsymbol{x}_{i:(i+T)}, \tilde{\boldsymbol{y}}_i\}_{i=1}^{N}$, where $\tilde{\boldsymbol{y}}_i$ is one-hot vector of noisy label $\tilde{y}_i \in \{1, \cdots, C\}$. N is the number of training samples, and C is the total number of classes.

Algorithm 1: LNRTSC Method

```
Input: Noisy dataset $\tilde{D} = \{\boldsymbol{x}_{i:(i+T)}, \tilde{\boldsymbol{y}}_i\}_{i=1}^{N}$, epochs for
       warming up $E_{\text{warmup}}$, total epochs $E$, batch size
       $N_b$, hyperparameters $k$, $\eta$.
Output: Model's parameters $\theta$
Randomly initialize $\theta$
for $e \leftarrow 0$ to $E_{warmup}$ do
    Train($\tilde{D}$, $\theta$) with loss $\mathcal{L}_{warm}$ in (9)
end
for $e \leftarrow 0$ to $E$ do
    for $i \leftarrow 0$ to $N$ do
        Obtain $\mathcal{N}_k(i)$ and $\mathcal{S}im(\mathcal{N}_k(i))$ based on the
          learned representation $z_i = g(\boldsymbol{x}_{i:(i+T)})$
        Obtain every sample's local label consistency
          $LLC_i = \sum_{j \in \mathcal{N}_k(i)} \mathbb{1}[\tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{y}}_j]$
    end
    Fit GMM on $\{LLC_i\}_{i=1}^{N}$ and obtain per-sample
      label confidence $\mathcal{W} = \{w_i\}_{i=1}^{N}$
    for $n \leftarrow 0, 1, \cdots, \frac{|\tilde{D}|}{N_b}$ do
        Randomly draw a mini-batch
          $\{(\boldsymbol{x}_{i:(i+T)}, \tilde{\boldsymbol{y}}_i, w_i)\}_{i=1}^{N_b}$
        for $i \leftarrow 0$ to $N_b$ do
            $\hat{\boldsymbol{y}}_i = f(g(\boldsymbol{x}_{i:(i+T)}))$
            $\boldsymbol{y}_i^{*} = w_i \tilde{\boldsymbol{y}}_i + (1 - w_i)\hat{\boldsymbol{y}}_i$
        end
        Calculate loss $\mathcal{L}_{lnl} = \{l^i\}_{i=1}^{N_b}$ in (12)
        Select top $\eta\%$ loss $\mathcal{L}' = \arg\max_{\mathcal{L}' \in \mathcal{L}_{lnl}} \sum_{l_i \in \mathcal{L}'} l^i$
          $s.t. |\mathcal{L}'| = |\mathcal{L}_{lnl}| * \eta\%$
        Update $\theta$ according to the gradient
          $\nabla \frac{1}{|\mathcal{L}'|} \sum_{l^i \in \mathcal{L}'} l^i$
    end
end
```

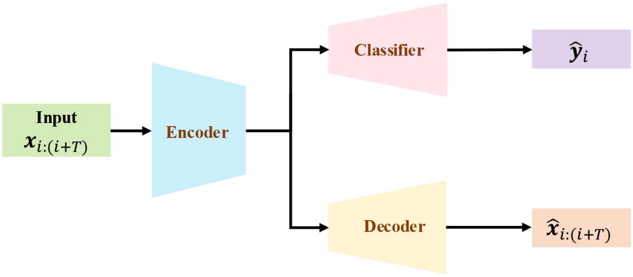


Fig. 1. The overall architecture of the model.

During the warming-up stage, the encoder $g(\cdot)$, classifier $f(\cdot)$, and decoder $\phi(\cdot)$ are trained jointly. In contrast, during the LNL training, only the parameters of the encoder $g(\cdot)$ and classifier $f(\cdot)$ are updated jointly, while the parameters of the decoder $\phi(\cdot)$ remain unchanged. Notably, to assess the robustness of the model against corrupted labels, the model is trained on noisy training datasets and evaluated on an unseen clean testing dataset.

The structures of the encoder $g(\cdot)$, the classifier $f(\cdot)$ and the decoder $\phi(\cdot)$ are presented in III-B. The procedures of the label

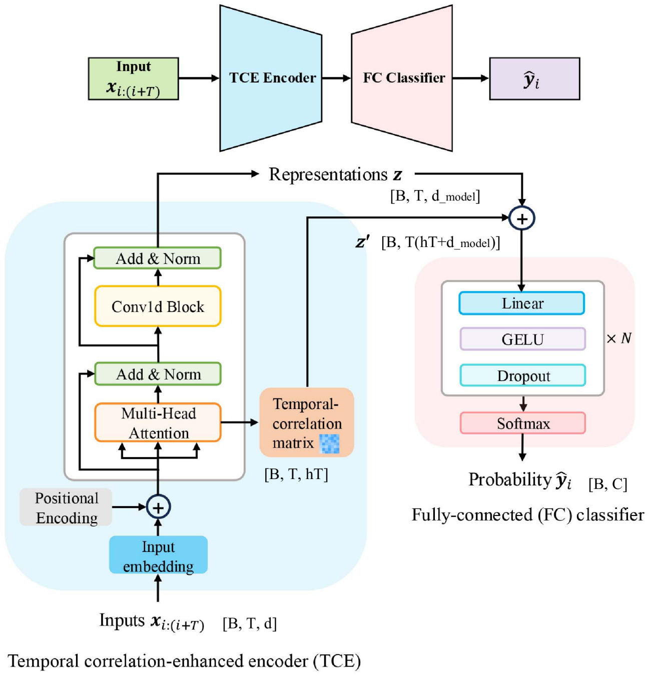


Fig. 2. The structures of the TCE and the FC classifier.

confidence assessment and the label correction are illustrated in III-C. The designed loss functions for the warming up and the LNL training are introduced in III-D respectively.

B. A novel encoder design

To extract the low-dimensional representations which fully reflect the underlying dynamic characteristics of the serialized input samples, the TCE encoder employs an embedding module with 1D CNNs to capture local dependencies, and utilizes the self-attention mechanism to extract long-term temporal dependencies. Moreover, the multi-head attention score matrix is then concatenated with the extracted features as extra explicit features, enabling the classifier to learn the variations in long-term dependencies under different fault conditions.

The detailed structure of the TCE encoder and the data flow are displayed as the blue block in Fig. 2. The raw samples are serialized to incorporate several previous time steps by using time window, which can be denoted as $\boldsymbol{x}_{i:(i+T)}$. T is the length of time window, d denotes the number of raw features. B denotes the batch size. The serialized input samples are embedded using 1D CNNs and concatenated with the positional encoding [25]. Then, the obtained embeddings H_0 are transmitted into a multi-head self-attention block. The multi-head self-attention block projects the queries Q, keys K and values V totally h times with different, learned linear projections to d_{model}/h dimensions. The formulations are displayed as follows:

$$\begin{aligned} &Q = K = V = H_0 \\ &Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V \end{aligned} \tag{1}$$

where Q, K, V are all equal to the values of the input embeddings H_0. $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ are the learned linear projection matrices, $d_k = d_v = d_{model}/h$. After that, the scaled dot-product function in (2) is performed on Q_i, K_i, V_i in parallel.

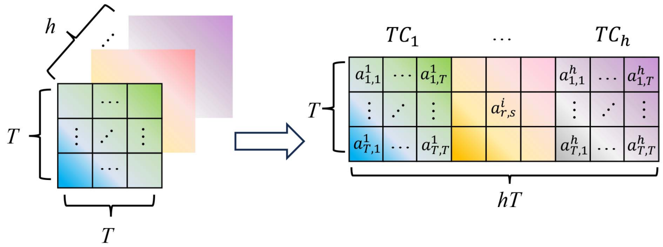


Fig. 3. The temporal correlation matrix (TCM).

$$TC_i = \mathrm{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}}) \tag{2}$$

$$\mathrm{head}_i = TC_i \cdot V_i. \tag{3}$$

We denote the attention score matrix, representing the temporal correlations, as TC_i, and the d_v-dimensional output values as head$_i$ $(i = 1, \cdots, h)$. The head$_i$ $(i = 1, \cdots, h)$ are then concatenated together and once again projected, followed by a residual connection and a layer normalization.

After that, the data is transmitted to the second block which consists of two 1D convolutional layers, a residual connection and a layer normalization. Z denotes the yielding representations from the TCE encoder.

Before transmitted into the classifier, the obtained representations Z is concatenated with the temporal correlation matrix (TCM). As shown in Fig. 3, the TCM is reshaped from the multi-head attention scores $TC_i = \{a^i_{r,s}\}(i = 1, \cdots, h; r = 1, \cdots, T; s = 1, \cdots, T)$, where $a^i_{r,s}$ represents the correlation intensity between timestamp r and timestamp s in the subspace i. The shape of the TCM is $T \times hT$, and $Z' \in \mathbb{R}^{T(hT+d_{model})}$. The first row of the TCM represents the correlation intensity between timestamp 1 and all timestamps in the h subspaces.

The concatenated features Z' is fed into the classifier, which is composed of a stack of identical layers (including a linear layer, a GeLU and a dropout) and a Softmax layer. The output of the classifier is a vector in which each element indicates the probability of the sample belonging to a specific category. In the warming-up stage, the encoded representations are also transmitted to the decoder, the decoder structure is same as the classifier. The output of the decoder is the reconstructed values of the serialized input samples.

C. A consistency-driven label correction framework

The local label consistency (LLC) metric is calculated to evaluate the label confidence and correct labels, denoted as a consistency-driven label correction framework. Firstly, k nearest neighbors of each sample are selected based on the pair-wise cosine similarities of the representations obtained by the TCE encoder, which has a computational complexity of $O(N^2)$:

$$\begin{aligned} &\mathrm{sim}(z_i, z_j) = z_i^T z_j / \|z_i\| \|z_j\| \\ &\text{where} \quad \boldsymbol{z_i} = g(\boldsymbol{x}_{i:(i+T)}) \end{aligned} \tag{4}$$

Then, the LLC of the i-th sample can be calculated as the fraction of the nearest k neighbors that share the same class label with sample i:

$$LLC_i = \sum_{j \in \mathcal{N}_k(i)} \mathbb{1}[\tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{y}}_j] \tag{5}$$

where $\mathbb{1}[\cdot]$ is the indicator function. It equals 1 if the equation in it is true. Otherwise, it equals 0. In the experiments, k is determined according to the dataset size and class density (k=30). Under high noise rates, k can be appropriately increased to enhance the robustness of the LLC calculation.

In every epoch, we calculate the LLC scores of samples in the method mentioned above. It can be inferred that the LLC scores of clean samples are more likely to be larger than those of noisy samples. Based on this, the two-component GMM is used to fit the LLC scores in many researches. To adjust the method to more general situations, this paper employs the Dirichlet process to automatically determine the number of components in the GMM, denoted as DP-GMM [3]. In the experimental results in Section IV, the number of components in the GMM is automatically set to 2, which is sufficient to distinguish between clean and noisy samples even in the situation with a high noise rate.

The probability belonging to the GMM component with a bigger absolute mean is used as the label confidence w_i for the i-th sample. Then, we correct the noisy labels based on the obtained label confidence w_i and the predictions from the model $\hat{\boldsymbol{y}}_i$:

$$\boldsymbol{y}_i^* = w_i \tilde{\boldsymbol{y}}_i + (1 - w_i) \hat{\boldsymbol{y}}_i \tag{6}$$

where $\boldsymbol{y}_i^*$ is the one-hot vector of the corrected label, $\tilde{\boldsymbol{y}}_i$ is the original noisy label, and $\hat{\boldsymbol{y}}_i$ is the predictive probability vector from the classifier.

In summary, to enhance the model's robustness to label noise, we follow the procedures outlined in Algorithm 1. Specifically, the label confidence values for all samples are firstly evaluated using (4)-(5). Then, in each training epoch, the labels of each batch are corrected as a weighted sum of the original noisy labels and the predicted labels as (6), where the weights are the label confidence values. These corrected labels, represented as probability vectors, are subsequently used to compute the LNL training loss in (12) and update the model parameters.

It should be noted that we do not explicitly relabel the samples with the corrected labels. Instead, the probability vectors obtained by (6) are directly used for model training, which improves the classifier's performance in terms of accuracy, precision, recall, and F1 score.

D. A two-stage self-supervised enhancement strategy

A two-stage self-supervised enhancement strategy is developed to guarantee the reliability of the corrected labels. In the initial stage of training, the model tends to focus on learning useful information that leads to a rapid reduction in loss. Therefore, we warm up the model using noisy labels for several epochs. The loss function for warming up is denoted in (7)-(9), including the cross-entropy loss term in (7) and the mean-squared-error (MSE) loss term in (8). The MSE loss term is introduced to guide the reconstruction of the input and assist feature extraction by using the self-supervised information.

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N_b} \sum_{i=1}^{N_b} \sum_{c=1}^{C} \tilde{y}_{i,c} \, \log(\hat{y}_{i,c}) \tag{7}$$

$$\mathcal{L}_{\text{mse}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \|\boldsymbol{x}_{i:(i+T)} - \hat{\boldsymbol{x}}_{i:(i+T)}\|_2 \tag{8}$$

$$\mathcal{L}_{\text{warm}} = \alpha \mathcal{L}_{\text{ce}} + \beta \mathcal{L}_{\text{mse}} \tag{9}$$

where N_b denotes the batch size. $\tilde{\boldsymbol{y}}_i$ is the one-hot vector of the noisy label. $\tilde{y}_{i,c}$ denotes the c-th element of $\tilde{\boldsymbol{y}}_i$, which represents the probability of sample i belonging to the class c; $\hat{y}_{i,c}$ is the element of the predictive probability vector from the classifier $\hat{\boldsymbol{y}}_i$. The $\boldsymbol{x}_{i:(i+T)}$ denotes the serialized input, $\hat{\boldsymbol{x}}_{i:(i+T)}$ denotes the reconstructed values of the input from the decoder. α, β are the parameters to balance the cross-entropy loss term $\mathcal{L}_{\text{ce}}$ and the MSE loss term $\mathcal{L}_{\text{mse}}$.

After the warming-up stage, the cross-entropy loss is also employed as the first term of the loss function for the LNL training, which is denoted as l_{ce}^i in (10).

$$l_{\text{ce}}^i = -\sum_{c=1}^{C} y_{i,c}^* \, \log(\hat{y}_{i,c}) \tag{10}$$

where C is the total number of the classes. $\boldsymbol{y}_i^*$ is the one-hot vector of the corrected label. $y_{i,c}^*$ denotes the c-th element of $\boldsymbol{y}_i^*$, which represents the probability of sample i belonging to the class c. As the original labels are corrupted by noise, the corrected labels are used to calculate the cross-entropy loss.

As mentioned in (6), the labels are refurbished by the label confidence and the predictive probability vectors. However, after the warming up, the model may still not be able to provide accurate category probability for classification of hard samples, thus the refurbished labels of these samples are not reliable. Inspired by the self-supervised learning techniques [23], a newly-designed contrastive loss term, denoted as l_{cts}^i in (11), is added to the loss function for the LNL training, in order to improve the quality of the corrected labels and accelerate the LNL training process.

$$l_{\text{cts}}^i = -\frac{1}{\sigma} \sum_{j \in \mathbb{S}_{i,\sigma}} \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k \in \{k | k \notin \mathbb{S}_{i,\sigma}, y_k^* \neq y_i^*\}} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{11}$$

where z_i denotes the representation obtained by the TCE encoder. σ denotes the number of the selected similar instances for each sample. $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity function. $\exp(\cdot)$ denotes the exponential function. τ is the temperature coefficient that adjusts the sensitivity of similarity scores.

In (11), the pair-wise cosine similarities in the feature space are calculated to find the top-σ most similar instances for each sample. The top-σ similar instances, denoted as $\mathbb{S}_{i,\sigma}$, are regarded as positive pairs for sample i. The left samples which simultaneously have the different corrected labels with sample i, are regarded as negative pairs for sample i, denoted as $\{k | k \notin \mathbb{S}_{i,\sigma}, y_k^* \neq y_i^*\}$. Compared with original contrastive loss, the proposed contrastive term pulls the similar instances together and mitigates the effect of false negatives by fully utilizing the information of the feature representations [18].

The loss function for the LNL training process is shown in (12), including the cross-entropy loss term l_{ce}^i in (10) and the contrastive loss term l_{cts}^i in (11).

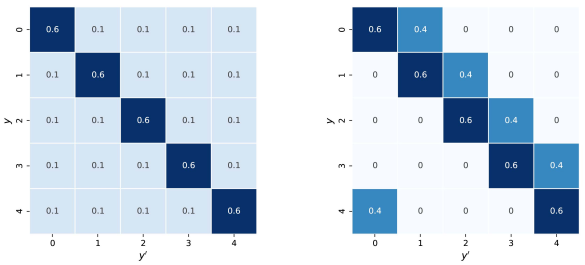
$$l^i = \lambda_1 l_{\text{ce}}^i + \lambda_2 l_{\text{cts}}^i \tag{12}$$

where λ_1, λ_2 are the parameters to balance these two items, generally set as $\lambda_1 = 1, \lambda_2 = 1$. In order to train a robust model, we only select the largest η% items from $\mathcal{L}_{lnl} = \{l^i\}_{i=1}^{N_b}$ for each batch [1], to update the parameters of the encoder and the classifier. The reason to add the selection operation is that the samples with small loss values are more likely to be clean, thus their corrected labels are not meaningful for the training.

IV. Case Studies

A. Tennessee Eastman process dataset

1) Process descriptions and noise settings: The Tennessee Eastman simulation platform [29] is developed based on a practical chemical reaction process. The whole TEP dataset is composed of 22 simulations, including one in normal state and the other 21 in different fault conditions. In this section, the TEP dataset has been applied to test the performance of our method. A sliding window is used to serialize the samples, incorporating previous timestamps. The window length is set to 50, and the step size is set as 1. We split the serialized dataset into a training set and a testing set with a ratio of 8:2. Since collecting large-scale datasets with precisely quantified real noise is practically infeasible, we injected controlled synthetic label noise into the training set [31]. After training, we utilized the clean-labeled testing set to evaluate the model. Our experiments are conducted on a server equipped with an NVIDIA RTX A6000 GPU (48 GB memory) and an Intel(R) Xeon(R) Silver 4214R CPU. All approaches are implemented using PyTorch 2.5.1, Python 3.9.21, and CUDA 12.3.



(a) Symmetric. (b) Pairflip.

Fig. 4. The examples for the two types of label noise.

Two types of noise are adopted for the experiments: symmetric noise and pairflip noise [33], [34]. The symmetric noise refers to a type of label noise where each true label is randomly flipped to any other classes with equal probability, which mimics manual annotation mistakes or sensor malfunctions. The noise level is typically controlled by a noise rate, which determines the probability of a label being flipped and enables reproducible and interpretable evaluation of the model. For example, as shown in Fig. 4(a), assuming there are five classes in total and the noise rate is 40%, only 60% of the labels remain correct, while the other 40% are uniformly mislabeled to the other four classes, each with a probability of 10%. The pairflip noise is a structured form of noise where a sample's label is only flipped to a specific class instead of any random class. This type of noise mimics class-dependent systematic errors induced by process drifts, abnormal operating conditions, or ambiguous boundary definitions between classes. For example, as shown in Fig. 4(b), when the noise rate is 40%, 40% of the labels are flipped to the next class of their true class. In the following experiments, the noise rate for the symmetric noise is set to 20%, 50%, 60%, and 70%; the noise rate for the pairflip noise is set to 20% and 40%.

2) Experimental results: On various setting of noise, we assessed the fault classification performance of the model by calculating the following evaluation metrics: accuracy (Acc), precision (Pre), recall (Rec) and F1 score (F1). In order to prove the superiority of the proposed LNRTSC method, the empirical risk minimization (ERM) method and the other six LNL methods are chosen to conduct comparative experiments. The hyperparameter settings are presented as follows:

(1) Co-teaching (CNN) [14]: The number of the out channels for the CNN layers are respectively set to [256, 512, 128, 64]. The kernel size, stride, and padding are set to 3, 1, and 2, respectively. The batch size is 128. The learning rate is initialized at 0.0005. The networks are trained for 10 epochs.

(2) Co-teaching (Transformer) : The d_model of the transformer is set to 512, and the number of attention heads is set to 8. The numbers of hidden neurons in the MLP are set to [512, 256, 128], respectively. The batch size is 32, and the learning rate is initialized at 0.0005. The networks are trained for 10 epochs.

(3) JoCoR (CNN) [36]: The regularization coefficient is set to 0.1. The other hyperparameters are kept the same as Co-teaching (CNN).

(4) JoCoR (Transformer) : The regularization coefficient is set to 0.1. The other hyperparameters remain the same as those in Co-teaching (Transformer).

(3) DivideMix (CNN) [22]: The hyperparameters are kept the same as Co-teaching (CNN).

(4) DivideMix (Transformer): The hyperparameters remain the same as those in Co-teaching (Transformer).

(5) ERM (CNN): This approach follows the standard empirical risk minimization framework without any specific operations for handling noisy labels. The hyperparameter setting is kept the same as Co-teaching (CNN).

(6) LNRTSC: The batch size is set to 64, and the learning rate is set to 0.0005. The model is firstly be warmed for 2 epochs, and then trained with corrected labels for 8 epochs. The *d_model* of the TCE encoder is set to 512, and the number of attention heads is set to 8. The numbers of hidden neurons in the FC classifier are set to [512, 256, 128]. The number of neighbors k is set to 30. The hyperparameters η, α, β, τ, σ, λ_1, and λ_2 are set as 80, 1., 0.1, 0.5, 3, 1., and 0.8, respectively, which can be

TABLE I
THE LNRTSC OUTPERFORMS THE BASELINES ACROSS ALL NOISE SETTINGS ON TEP DATASET

Noise type	Noise rate	Co-teaching (CNN)				Co-teaching (Transformer)				JoCoR (CNN)				JoCoR (Transformer)			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.994	0.994	0.994	0.994	0.991	0.991	0.991	0.991	0.991	0.991	0.991	0.991	0.992	0.992	0.992	0.992
	50%	0.981	0.982	0.982	0.981	0.972	0.976	0.973	0.973	0.915	0.928	0.918	0.915	0.989	0.989	0.989	0.989
	60%	0.920	0.886	0.925	0.903	0.958	0.964	0.959	0.960	0.781	0.824	0.783	0.785	0.936	0.949	0.937	0.937
	70%	0.732	0.663	0.740	0.685	0.875	0.861	0.875	0.863	0.729	0.757	0.737	0.717	0.714	0.719	0.716	0.705
Pairflip	20%	0.989	0.990	0.990	0.990	0.994	0.994	0.993	0.994	0.988	0.988	0.988	0.988	0.988	0.988	0.988	0.988
	40%	0.949	0.954	0.949	0.948	0.942	0.947	0.943	0.943	0.864	0.864	0.862	0.841	0.959	0.966	0.962	0.961
Average		0.928	0.912	0.930	0.917	0.955	0.956	0.956	0.954	0.878	0.892	0.880	0.873	0.930	0.934	0.931	0.929

Noise type	Noise rate	DivideMix (CNN)				DivideMix (Transformer)				ERM (CNN)				LNRTSC			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.994	0.994	0.994	0.994	0.995	0.995	0.996	0.995	0.999	0.999	0.999	0.999	1.	1.	1.	**1.**
	50%	0.975	0.977	0.973	0.972	0.983	0.981	0.983	0.983	0.974	0.979	0.975	0.976	0.999	0.999	0.999	**0.999**
	60%	0.965	0.963	0.962	0.963	0.975	0.977	0.972	0.975	0.881	0.891	0.883	0.882	0.995	0.995	0.996	**0.995**
	70%	0.846	0.853	0.827	0.841	0.870	0.866	0.869	0.868	0.752	0.749	0.759	0.743	0.974	0.978	0.974	**0.975**
Pairflip	20%	0.991	0.991	0.991	0.991	0.993	0.994	0.992	0.992	0.960	0.961	0.960	0.960	0.999	0.999	0.999	**0.999**
	40%	0.955	0.943	0.959	0.944	0.958	0.958	0.957	0.957	0.715	0.741	0.714	0.707	0.976	0.978	0.976	**0.976**
Average		0.957	0.954	0.951	0.951	0.956	0.955	0.955	0.955	0.880	0.887	0.882	0.878	0.991	0.992	0.991	**0.991**

adjusted based on the noise characteristics of the data.

Table I reports the Acc, Pre, Rec, and F1 scores of the selected methods on the TEP dataset, averaged over five repeated experiments. The highest F1 scores under each noise setting are bolded, and the second-highest are underlined. Standard ERM performs well only under low noise rate (e.g., 20%) and degrades rapidly at higher noise levels, with F1 scores of 0.743 under 70% symmetric noise and 0.707 under 40% pairflip noise. Co-teaching and JoCoR improve over ERM, especially when combined with a transformer encoder, yet both struggle under extreme noise (e.g., 70% symmetric noise). In contrast, our proposed LNRTSC consistently achieves the highest F1 scores across all noise settings, reaching 0.975 and 0.976 under 70% symmetric noise and 40% pairflip noise respectively, surpassing the second-best models by 0.107 and 0.015. Averaged over all noise conditions, LNRTSC outperforms DivideMix (Transformer) by 0.036. These results demonstrate that our method effectively handles diverse label noise and maintains strong classification performance even under severe noise.

3) Ablation study: To evaluate the effectiveness of the three components of LNRTSC, including the TCE encoder, the label correction (LC) and the self-supervised loss functions, we performed comprehensive ablation study on the TEP dataset.

As illustrated in Table II, the sole replacement of 1D CNNs with the TCE encoder does not significantly enhance the classification performance of ERM. However, by comparing the classification results of CNN + LC and TCE + LC, it can be seen that the F1 score of the TCE + LC is markedly superior under 40% pairflip noise conditions, while the performance of the two methods remains comparable in other noise settings. This finding suggests that the TCE exhibits a strong capability in capturing underlying temporal correlations and effectively mitigating the structured bias induced by pairflip noise. By comparing the performance of ERM (CNN) with CNN + LC and ERM (TCE) with TCE + LC, it is evident that the label correction based on label confidence considerably enhances classification performance under label noise. By comparing the performance of Transformer encoder (TransEnc) + LC and TCE + LC, it can be observed that the TCE obtains higher Acc and F1 scores than TransEnc + LC in most conditions, indicating that TCE better captures temporal dependencies in noisy time-series data and performs effectively under high noise rate. Specifically, the F1 scores show an increase of about 0.2~0.3 under 70% symmetric noise and 40% pairflip noise. Comparing to TCE + LC, the average F1 score of the LNRTSC increases by 0.003, and the improvement is particularly evident when the noise rate is high. ALL in all, the results of the ablation experiments proves the effectiveness of the three key parts of the proposed LNRTSC, including the TCE encoder, label correction and the newly-designed loss functions.

B. SEU-Gearbox dataset

1) Process descriptions and noise settings: The SEU-gearbox dataset is collected from Drivetrain Dynamic Simulator. The working condition with rotating speed-load configuration is set to be 20-0. There are eight types of signals recorded in the dataset, including motor vibration, vibration of planetary gearbox in x, y and z directions, motor torque, vibration of parallel gear box in x, y and z directions. There are totally five labels for the samples, which are Health, Chipped, Miss, Root and Surface. We split the dataset into a training set and a testing set with a ratio of 8:2. The noise settings are the same as the TEP dataset.

2) Experimental results: Table III summarizes the fault classification performance (Acc, Pre, Rec, F1) of LNRTSC, Co-teaching, JoCoR, DivideMix, and ERM with Transformer and CNN backbones on the SEU-gearbox dataset under symmetric and pairflip noise. While all methods perform well at low noise levels (20% and 50% symmetric noise, 20% and 40% pairflip noise), LNRTSC consistently achieves perfect scores (almost 1.000) across all metrics. This is partly because the dataset is relatively small, with only five categories (including fault and normal conditions), making the classification task relatively simple, and the negative effects of 20% and 50% label noise can be largely resisted by the models themselves. As the noise rate increases, LNRTSC demonstrates remarkable robustness. Under 70% symmetric noise, it maintains an F1 score of 0.998, whereas DivideMix (Transformer), typically the second-best, drops below 0.820, Co-teaching (Transformer) decreases to 0.737, and JoCoR (Transformer) to 0.804. Averaged across all noise conditions,

Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

TABLE II
THE ABLATION STUDY FOR DIFFERENT COMPONENTS OF THE LNRTSC ON TEP DATASET

Noise type	Noise rate	ERM (CNN)				ERM (TCE)				CNN + LC				TransEnc + LC				TCE + LC				LNRTSC			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.999	0.999	0.999	0.999	0.991	0.991	0.991	0.991	0.999	0.999	0.999	0.999	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**
	50%	0.974	0.979	0.975	0.976	0.963	0.965	0.965	0.964	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**
	60%	0.881	0.891	0.883	0.882	0.876	0.884	0.881	0.880	0.997	0.997	0.997	0.997	0.990	0.989	0.990	0.990	0.989	0.989	0.989	0.989	0.995	0.995	0.996	0.995
	70%	0.752	0.749	0.759	0.743	0.723	0.742	0.732	0.726	0.975	0.977	0.975	**0.975**	0.961	0.959	0.964	0.960	0.966	0.967	0.967	0.966	0.974	0.978	0.974	**0.975**
Pairflip	20%	0.960	0.961	0.960	0.960	0.947	0.952	0.950	0.949	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**	0.999	0.999	0.999	**0.999**
	40%	0.715	0.741	0.714	0.707	0.762	0.787	0.760	0.755	0.923	0.927	0.922	0.921	0.967	0.967	0.971	0.968	0.974	0.976	0.974	0.973	0.976	0.978	0.976	**0.976**
Average		0.880	0.887	0.882	0.878	0.877	0.887	0.880	0.878	0.982	0.983	0.982	0.982	0.986	0.985	0.987	0.986	0.988	0.988	0.988	0.988	0.991	0.992	0.991	**0.991**

TABLE III
THE LNRTSC OUTPERFORMS THE BASELINES ACROSS ALL NOISE SETTINGS ON SEU-GEARBOX DATASET

Noise type	Noise rate	Co-teaching (CNN)				Co-teaching (Transformer)				JoCoR (CNN)				JoCoR (Transformer)			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.996	0.997	0.996	0.996	0.998	0.998	0.998	0.998	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
	50%	0.994	0.994	0.994	0.994	0.997	0.996	0.996	0.996	0.995	0.995	0.995	0.995	0.997	0.997	0.997	0.997
	60%	0.973	0.984	0.963	0.971	0.975	0.982	0.965	0.974	0.971	0.971	0.968	0.968	0.984	0.984	0.984	0.984
	70%	0.712	0.703	0.716	0.705	0.745	0.736	0.742	0.737	0.745	0.743	0.747	0.742	0.799	0.813	0.798	0.804
Pairflip	20%	0.999	0.999	0.999	0.999	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
	40%	0.996	0.995	0.996	0.995	0.998	0.998	0.998	0.998	0.997	0.996	0.997	0.996	0.997	0.997	0.996	0.996
Average		0.945	0.946	0.944	0.944	0.952	0.952	0.950	0.951	0.951	0.950	0.951	0.950	0.962	0.965	0.962	0.963

Noise type	Noise rate	DivideMix (CNN)				DivideMix (Transformer)				ERM (CNN)				LNRTSC			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.998	0.997	0.998	0.998	0.999	0.998	0.999	0.999	0.996	0.996	0.995	0.995	1.	1.	1.	**1.**
	50%	0.996	0.996	0.996	0.996	0.998	0.997	0.998	0.997	0.992	0.992	0.992	0.992	0.999	0.999	0.999	**0.999**
	60%	0.970	0.970	0.969	0.970	0.983	0.982	0.986	0.982	0.965	0.971	0.964	0.965	0.998	0.998	0.998	**0.998**
	70%	0.729	0.722	0.725	0.723	0.821	0.827	0.813	0.820	0.650	0.540	0.646	0.562	0.998	0.998	0.998	**0.998**
Pairflip	20%	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	**1.**
	40%	0.996	0.996	0.996	0.996	0.997	0.997	0.997	0.997	0.993	0.993	0.993	0.993	1.	1.	1.	**1.**
Average		0.948	0.947	0.947	0.947	0.966	0.967	0.965	0.966	0.934	0.916	0.933	0.919	0.999	0.999	0.999	**0.999**

LNRTSC achieves an average F1 score of 0.999, exceeding the second-best baseline, DivideMix (Transformer), by more than three percentage points. These results highlight LNRTSC's robustness across various noise settings.

3) Visualization analysis: In order to better illustrate the principles of the proposed LNRTSC model, we display the evolution of label confidence and feature extraction through multiple visualizations. Furthermore, we denote the class corresponding to the largest element of the probability vector in (6) as the corrected label, and we evaluate the accuracy of label correction in each epoch.

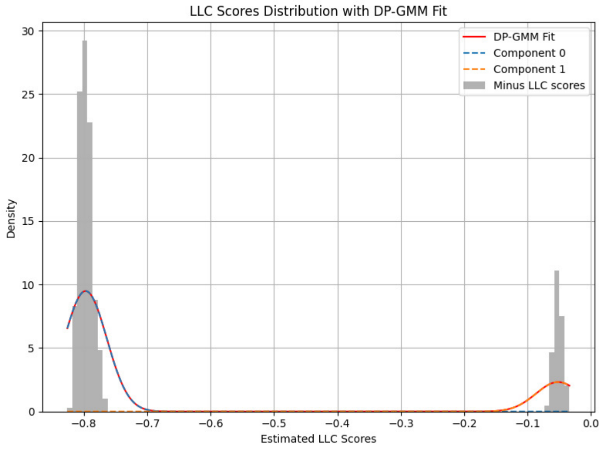


Fig. 5. The LLC scores distribution with DP_GMM fit (Symmetric with 20% noise).

The distributions of LLC scores are displayed in Fig. 5 and Fig. 6, which correspond respectively to the samples with symmetric label noise at noise rates of 20% and 70%. In these figures, the x-axis represents the negative of the estimated LLC scores calculated using (5), the y-axis represents the sample density for different LLC scores. To observe the shape of the distribution in detail, we divide it into 100 gray histogram bins and compute the sample density for each bin. The distribution of the LLC scores is fitted by the DP_GMM, shown as a red solid line. Through the Dirichlet process, two Gaussian components are ultimately extracted, represented by the blue dashed line and the orange dashed line respectively. The two Gaussian components are well separated, indicating that the two-component GMM is sufficient to fit the distribution of

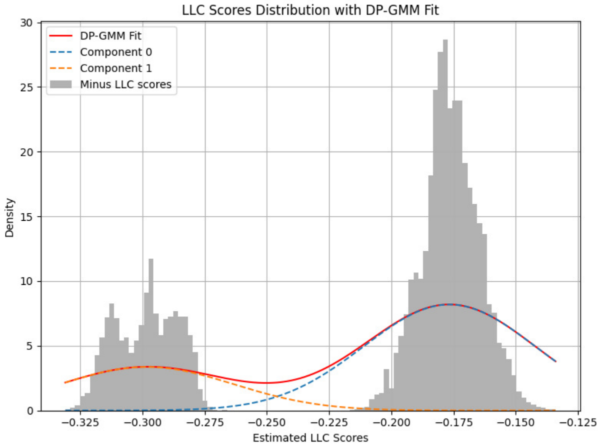


Fig. 6. The LLC scores distribution with DP_GMM fit (Symmetric with 70% noise).

TABLE IV
LABEL CORRECTION PERFORMANCE OF THE LNRTSC METHOD ACROSS EPOCHS UNDER 70% SYMMETRIC NOISE

Metric	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10
V-measure score	0.847	0.682	0.683	0.683	0.684	0.823	0.953	0.962	0.971	**0.975**
correction accuracy	0.599	0.597	0.597	0.600	0.597	0.812	0.860	0.967	0.984	**0.986**
classification accuracy	0.613	0.613	0.605	0.613	0.820	0.859	0.973	0.988	0.989	**0.994**
F1 Score	0.502	0.502	0.499	0.502	0.760	0.824	0.972	0.988	0.989	**0.994**

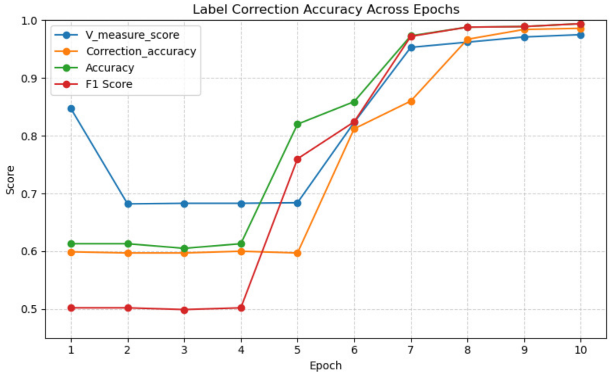


Fig. 7. The label correction performance across epochs under 70% symmetric noise.

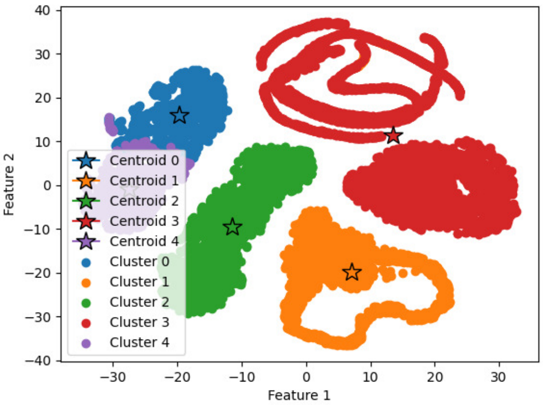


Fig. 8. The t-SNE visualization of the features processed by the TCE encoder under 70% symmetric Noise (without LNL training).

the LLC scores even in cases with a high noise rate.

As depicted in these figures, the LLC scores of the samples can be regarded as coming from two different Gaussian distributions. It can be assumed that the LLC scores of clean samples are generally larger than those of noisy samples. Therefore, the component which has the smaller mean (with larger absolute value) corresponds to the clean sample, while the component which has the bigger mean (with smaller absolute value) corresponds to the noisy sample. The ratio of the areas under the curves of the two components (the ratio of the areas of the two sets of histograms) corresponds to the ratio of the number of clean samples to noisy samples, which can be denoted as (1-γ): γ, where γ represents the noise rate. Based on the above analysis, it is reasonable to use the probability of belonging to the GMM component with the larger absolute mean as the label confidence w for the samples.

In each LNL training epoch, the labels of each batch are corrected as a weighted sum of the original noisy labels and the predicted labels as (6), where the weights are the calculated label confidence values. It should be noted that we do not explicitly relabel the samples with the corrected labels. Instead, we use the probability vectors calculated by (6) directly for model training, which improves the classifier's performance. In order to visualize how the label correction benefits the classification performance, we assign the largest element of the probability vector in (6) as the corrected label (by 'argmax' operation) and calculate the accuracy of the corrected labels before each training epoch. The accuracy of the correction labels (70% symmetric noise) across epochs are displayed in Table IV, as well as the classification accuracy and F1 scores. It also can be figured out that the classification accuracy and F1 scores are improved with the increasing label correction rate, as shown in Fig. 7. It should be noted that the 'argmax' operation is not a continuous function, thus the label correction rate would be a little delayed, compared with the classification accuracy and the F1 scores.

Meanwhile, we cluster the features (70% symmetric noise) extracted by the TCE encoder using the K-means method, and reduce their dimensions into two dimensions via the t-SNE method. Fig. 8 and Fig. 9 respectively show the clustering results of the features which are extracted by the TCE encoder without LNL training and with LNL training. By comparing the two figures, it is evident that after LNL training, the extracted features of samples from different classes become more clearly separable. In addition, the V-measure score for each epoch is recorded in Table IV and Fig. 7. It can also be observed that the clustering accuracy of K-means increases progressively with the improvement of label correction accuracy and classification accuracy.

4) Sensitivity analysis: The detailed explanations of hyperparameters, the hyperparameter adjustment strategy, and the results of the sensitivity experiments are presented. The experimental results demonstrate that the model performance remains stable within a certain range of hyperparameter values. According to the approach principles and the sensitivity analysis experimental results in Tables V~VII, we recommend default values or ranges for the following hyperparameters: $\eta = 80$, $\beta \in [0.1, 1.]$, $\lambda_2 \in [0.1, 1.]$, and $\sigma = 20$.

η represents the proportion of samples in each batch whose loss values are used to update the model parameters. We sort the samples in descending order of their loss values and select the top η% for back propagation. When the noise rate is high, most samples tend to have inaccurate initial classification

Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

TABLE V
SENSITIVITY ANALYSIS OF HYPERPARAMETER (α, β) ON THE LNRTSC PERFORMANCE

Noise type	Noise rate	(α, β) : (1, 0)				(α, β) : (1, 0.1)				(α, β) : (1, 0.5)				(α, β) : (1, 1)				(α, β) : (1, 1.5)			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.986	0.986	0.985	0.985	1.	1.	1.	**1.**	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.989	0.989	0.988	0.989
	70%	0.978	0.982	0.977	0.976	0.998	0.998	0.998	**0.998**	0.966	0.970	0.968	0.968	0.975	0.976	0.974	0.974	0.968	0.970	0.967	0.966
Pairflip	40%	0.995	0.995	0.995	0.995	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	0.996	0.996	0.996	0.996
Average		0.986	0.988	0.986	0.985	0.999	0.999	0.999	**0.999**	0.988	0.990	0.989	0.989	0.991	0.992	0.991	0.991	0.984	0.985	0.984	0.984

TABLE VI
SENSITIVITY ANALYSIS OF HYPERPARAMETER (λ_1, λ_2) ON THE LNRTSC PERFORMANCE

Noise type	Noise rate	(λ_1, λ_2) : (1, 0)				(λ_1, λ_2) : (1, 0.1)				(λ_1, λ_2) : (1, 0.5)				(λ_1, λ_2) : (1, 1)				(λ_1, λ_2) : (1, 1.5)			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	0.997	0.998	0.997	0.997	1.	1.	1.	**1.**	0.999	0.998	0.999	0.999	1.	1.	1.	**1.**	0.999	0.999	0.999	0.999
	70%	0.993	0.993	0.993	0.993	0.998	0.998	0.997	**0.998**	0.997	0.996	0.996	0.996	0.998	0.998	0.998	**0.998**	0.989	0.990	0.987	0.989
Pairflip	40%	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	0.999	0.999	0.998	0.999
Average		0.997	0.997	0.997	0.997	0.999	0.999	0.999	**0.999**	0.999	0.998	0.998	0.998	0.999	0.999	0.999	**0.999**	0.996	0.996	0.995	0.996

TABLE VII
SENSITIVITY ANALYSIS OF HYPERPARAMETER σ ON THE LNRTSC PERFORMANCE

Noise type	Noise rate	σ : 5				σ : 15				σ : 20				σ : 25				σ : 50			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Symmetric	20%	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**
	70%	0.984	0.985	0.983	0.984	0.986	0.986	0.986	0.986	0.998	0.998	0.998	**0.998**	0.995	0.995	0.995	0.995	0.986	0.987	0.986	0.986
Pairflip	40%	0.998	0.999	0.997	0.997	1.	1.	1.	**1.**	1.	1.	1.	**1.**	1.	1.	1.	**1.**	0.994	0.994	0.993	0.994
Average		0.994	0.995	0.993	0.994	0.995	0.995	0.995	0.995	0.999	0.999	0.999	**0.999**	0.998	0.998	0.998	0.998	0.993	0.994	0.993	0.993

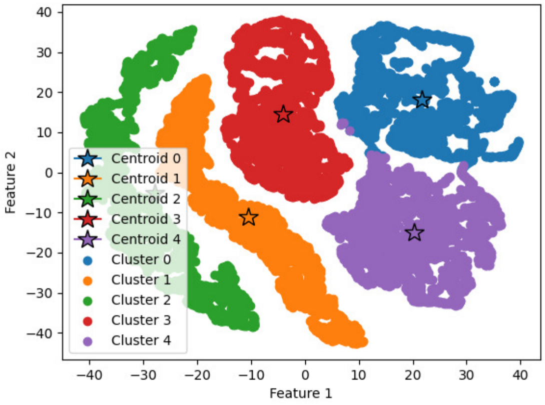


Fig. 9. The t-SNE visualization of the features processed by the TCE encoder under 70% symmetric Noise (with LNL training).

results and relatively large loss values. In this case, a larger η (e.g., 80, 90, or 100) is recommended.

τ represents the temperature coefficient in the contrastive loss term (11), which controls the sharpness of the Softmax function. A small τ will cause the model to focus excessively on the most similar samples, whereas a large τ will lead to an overly flat probability distribution.

α and β are used to balance the weights of the two loss terms in the warming-up stage, shown in (7)-(9). Their values can be adjusted according to the noise rate. In general, we set $\alpha = 1$. When the noise rate is high, β can be appropriately increased to emphasizing the reconstruction loss term, thereby leveraging more unsupervised feature information to guide model learning.

λ_1 and λ_2 control the relative weights of the two loss terms in the LNL training phase, shown in (10)-(12). We generally set λ_1 to 1. When the noise rate is high, certain sample labels are difficult to be right corrected at the early stage of training. In such cases, λ_2 can be appropriately increased to leverage the contrastive loss term for self-supervised learning.

σ denotes the number of positive pairs in the contrastive loss function in (11). If σ is too small, the number of positive pairs is insufficient, which weakens the aggregation of positive samples in the feature space. If σ is too large, it may introduce positive pairs from other classes (false positives), reducing the effectiveness of contrastive learning. Therefore, it is generally recommended to set σ to be approximately (batch_size/num_class). We found through numerous experiments that this setting ensures a moderate number of positive pairs, maintaining stable training while preventing the introduction of false positives.

V. CONCLUSIONS AND FUTURE WORK

The paper has proposed a noise-resistant method for time-series classification called LNRTSC, which is used to effectively classify time-series data into its correct category while reducing the influence of noisy labels. In the LNRTSC, a novel attention-based temporal correlation-enhanced (TCE) encoder has been introduced to produce low-dimensional representations for long time series. For LNL training, the noisy labels have been corrected according to the label confidence which is calculated based on the local label consistency of the nearest neighbors in the representative space. A two-stage self-supervised enhancement strategy has been specifically

Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

designed to assist the warming-up stage and the LNL training stage respectively, in order to enhance the quality of the corrected labels. The LNRTSC method has been validated on the TEP dataset and the SEU-gearbox dataset. The experimental results have indicated that the LNRTSC can achieve significant improvements in fault classification performance on corrupted data, compared with other LNL methods.

Although the LNRTSC has demonstrated satisfactory performance in time-series classification under label noise, there are still some issues that can be further improved in the future. Firstly, we are considering to incorporate the structure-aware confidence estimation into the label correction framework. Besides, we plan to extend the proposed framework to broader labeling scenarios. Since our method is based on confidence estimation and soft label refinement, it can be naturally adapted to semi-supervised, soft-labeled, and heterogeneous label settings by assigning pseudo labels or extending label dimension, to adapt to more complex industrial situations.

References

[1] M. Chen, Y. Zhao, B. He, Z. Han, J. Huang, B. Wu, and J. Yao, Learning with noisy labels over imbalanced subpopulations, *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[2] Z. Chen, L. Zhang, J. Tang, J. Mao and W. Sheng, Conditional generative adversarial net based feature extraction along with scalable weakly supervised clustering for facial expression classification, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 4, art. no. 100024, 2024.

[3] W. Chen, X. Wang, Z. Cai, C. Liu, Y. Zhu, and W. Lin, *DP-GMM clustering-based ensemble learning prediction methodology for dam deformation considering spatiotemporal differentiation. Knowledge-Based Systems*, vol. 222, art. no. 106964, 2021.

[4] G.-F. Cui, L.-B. Wu and M. Wu, Adaptive event-triggered fault-tolerant control for leader-following consensus of multi-agent systems, *International Journal of Systems Science*, vol. 55, no. 15, pp. 3291–3303, 2024.

[5] F. Dunkin, X. Li, H. Li, G. Wu, C. Hu, and S. S. Ge, MgCNL: A sample separation approach via multi-granularity balls for fault diagnosis with the interference of noisy labels, *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 7748-7761, 2024.

[6] F. Dunkin, X. Li, H. Li, G. Wu, C. Hu, L. Yu, X. Lu and S. S. Ge, Wisdom via Multiple Perspectives: A Multigranularity Clusters Fusion Approach for Fault Diagnosis With Noisy Labels, *IEEE/ASME Transactions on Mechatronics*, 2025.

[7] F. Dunkin, X. Li, Z. Zhang, K. Wang, T. Gao, G. Wu, and Z. Li, Certainty From Uncertainty: Multigranularity Labeling Inspired by Quantum Collapse for Learning With Noisy Labels in Fault Diagnosis, *IEEE Transactions on Industrial Informatics*, vol. 21, no. 8, pp. 6443-6454, 2025.

[8] F. Dunkin, X. Li, C. Hu, G. Wu, H. Li, X. Lu, and Z. Zhang, Like draws to like: A Multi-granularity Ball-Intra Fusion approach for fault diagnosis models to resists misleading by noisy labels, Advanced Engineering Informatics, vol. 60, art. no. 102425, 2024.

[9] F. Deng, Y. Ming and B. Lyu, CCE-Net: causal convolution embedding network for streaming automatic speech recognition, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 3, art. no. 100019, 2024.

[10] K. Feng, Y. Xu, Y. Wang, S. Li, Q. Jiang, B. Sun, J. Zheng, and Q. Ni, Digital twin enabled domain adversarial graph networks for bearing fault diagnosis, *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 1, pp. 113-122, 2023.

[11] W. Fang, B. Shen, A. Pan, L. Zou and B. Song, A cooperative stochastic configuration network based on differential evolutionary sparrow search algorithm for prediction, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2314481, 2024.

[12] J. Goldberger and E. Ben-Reuven, Training deep neural-networks using a noise adaptation layer, in *Proceedings of the International conference on learning representations*, 2017.

[13] M. Gong, L. Sheng and D. Zhou, Robust fault-tolerant stabilisation of uncertain high-order fully actuated systems with actuator faults, *International Journal of Systems Science*, vol. 55, no. 12, pp. 2518–2530, 2024.

[14] 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, *Advances in neural information processing systems*, vol. 31, 2018.

[15] Y. He, X. Kong, L. Yao, and Z. Ge, Neural network weight comparison for industrial causality discovering and its soft sensing application, *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 8817–8828, 2022.

[16] Z. Huang, J. Zhang, and H. Shan. Twin contrastive learning with noisy labels, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[17] 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 IEEE/CVF international conference on computer vision*, pp. 3326–3334, 2019.

[18] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, Boosting contrastive self-supervised learning with false negative cancellation, in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2785–2795, 2022.

[19] A. Iscen, J. Valmadre, A. Arnab, and C. Schmid, Learning with neighbor consistency for noisy labels, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4672–4681, 2022.

[20] F. Jin, L. Ma, C. Zhao and Q. Liu, State estimation in networked control systems with a real-time transport protocol, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2347885, 2024.

[21] H. Jmila, M. I. Khedher and M. A. El-Yacoubi, The promise of applying machine learning techniques to network function virtualization, *International Journal of Network Dynamics and Intelligence*, vol. 3, no. 4, art. no. 100020, 2024.

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

[23] S. Li, X. Xia, S. Ge, and T. Liu, Selective-supervised contrastive learning with noisy labels, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 316–325, 2022.

[24] Y. Li, Z. Zhou, C. Sun, X. Chen, and R. Yan, Variational attention-based interpretable transformer network for rotary machine fault diagnosis, *IEEE transactions on neural networks and learning systems*, vol. 35, no. 5, pp. 6180–6193, 2024.

[25] Y. Liu, H. Wu, J. Wang, and M. Long, Non-stationary transformers: Exploring the stationarity in time series forecasting, *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[26] G. Ma, Z. Wang, W. Liu, J. Fang, Y. Zhang, H. Ding, and Y. Yuan, A two-stage integrated method for early prediction of remaining useful life of lithium-ion batteries, *Knowledge-Based Systems*, vol. 259, art. no. 110012, 2023.

[27] B. Qu, D. Peng, Y. Shen, L. Zou and B. Shen, A survey on recent advances on dynamic state estimation for power systems, *International Journal of Systems Science*, vol. 55, no. 16, pp. 3305–3321, 2024.

[28] J. Ouyang, C. Lu, B. Wang, and C. Li, Supervised contrastive learning with corrected labels for noisy label learning, *Applied Intelligence*, vol. 53, no. 23, pp. 29378–29392, 2023.

[29] J. Qian, Z. Song, Y. Yao, Z. Zhu, and X. Zhang, A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes, *Chemometrics and Intelligent Laboratory Systems*, vol. 231, art. no. 104711, 2022.

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

[31] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022.

[32] C. Tan, J. Xia, L. Wu, and S. Z. Li, Co-learning: Learning from noisy labels with self-supervision, *Proceedings of the 29th ACM international conference on multimedia*, pp. 1405–1413, 2021.

[33] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, Joint optimization framework for learning with noisy labels, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5552–5560, 2018.

[34] C. Wang, Z. Wang, and H. Dong, A novel prototype-assisted contrastive adversarial network for weak-shot learning with applications: Handling weakly labeled data, *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 533–543, 2024.

Citation information: DOI: 10.1109/TII.2025.3626697, IEEE Transactions on Industrial Informatics

[35] Y. Wang, C. Wen and X. Wu, Fault detection and isolation of floating wind turbine pitch system based on Kalman filter and multi-attention 1DCNN, *Systems Science & Control Engineering*, vol. 12, no. 1, art. no. 2362169, 2024.

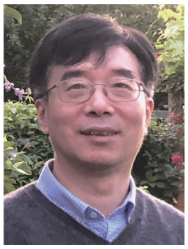
[36] H. Wei, L. Feng, X. Chen, and B. An, Combating noisy labels by agreement: A joint training method with co-regularization, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13726–13735, 2020.

[37] Y. Xue, R. Yang, X. Chen, Z. Tian, and Z. Wang, A novel local binary temporal convolutional neural network for bearing fault diagnosis, *IEEE Transactions on Instrumentation and Measurement*, vol. 72, art. no. 3525013, 2023.

[38] Z. Yang, R. Jia, P. Wang, L. Yao, and B. Shen, Supervised attention-based bidirectional long short-term memory network for nonlinear dynamic soft sensor application, *ACS omega*, vol. 8, no. 4, pp. 4196–4208, 2023.

Yimeng He received the B.Eng. degree from the School of Automation, Central South University, Changsha, China, in 2020, and received a Ph.D. degree in control science and engineering at the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2025.

Her research interests include causal discovery, data-driven industrial modeling, machine learning, and deep learning.



Zidong Wang (Fellow, IEEE) received the B.Sc. degree in mathematics in 1986 from Suzhou University, Suzhou, China, and the M.Sc. degree in applied mathematics in 1990 and the Ph.D. degree in electrical engineering in 1994, both from Nanjing University of Science and Technology, Nanjing, China.

He is currently Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the U.K. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published a number of papers in international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for *International Journal of Systems Science*, the Editor-in-Chief for *Neurocomputing*, the Editor-in-Chief for *Systems Science & Control Engineering*, and an Associate Editor for 12 international journals including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics—Part C. He is a Member of the Academia Europaea, a Member of the European Academy of Sciences and Arts, an Academician of the International Academy for Systems and Cybernetic Sciences, a Fellow of the IEEE, a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.

Weibo Liu (Member, IEEE) received the B.Eng. degree in electrical engineering from the Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool, U.K, in 2015, and the Ph.D. degree in artificial intelligence in 2020 from the Department of Computer Science, Brunel University London, Uxbridge, U.K.

He is currently a Lecturer in the Department of Computer Science, Brunel University London, Uxbridge, U.K. His research interests include intelligent data analysis, evolutionary computation, machine learning, deep learning and transfer learning. He serves as an Associate Editor for the Journal of Ambient Intelligence and Humanized Computing and the Journal of Cognitive Computation. He is a very active reviewer for many international journals and conferences.

Jingzhong Fang received his B.Eng. degree in automation from Shandong University of Science and Technology, Qingdao, China, in 2020, and the M.Sc. degree in data science and analytics from Brunel University London, Uxbridge, U.K., in 2021. He is currently pursuing the Ph.D. degree in Computer Science at Brunel University London, Uxbridge, U.K.

His research interests include data analysis and deep learning techniques.

Linwei Chen received her B.Eng. degree in Electrical and Electronic Engineering from the University of Warwick, Coventry, U.K., in 2022. She is currently pursuing the Ph.D. degree in Computer Science at Brunel University London, Uxbridge, U.K.

Her research interests include transfer learning and optimization.

Zhihuan Song received the B.Eng. and M.Eng. degrees in industrial automation from the Hefei University of Technology, Anhui, China, in 1983 and 1986, respectively, and the Ph.D. degree in industrial automation from Zhejiang University, Hangzhou, China, in 1997. Since 1997, he has been with the Department of Control Science and Engineering, Zhejiang University, where he was first a Postdoctoral Research Fellow, then an Associate Professor, and is currently a Professor. He has published more than 200 papers in journals and conference proceedings. His research interests include the modeling and fault diagnosis of industrial processes, analytics and applications of industrial big data, and advanced process control technologies.