Adaptive Wavelet Normalization Network for Soft Sensor Modeling in Nonstationary Industrial Processes

Xiaoli Wang, Yulong Wang, Fanlei Lu, Jiayi Zhou, *Member, IEEE*, Liyang Qin, and Chunhua Yang, *Fellow, IEEE*

***Abstract*—In industrial processes, accurate, real-time soft sensor modeling of key product indices is essential for optimal process control and improved product quality. However, the nonstationary characteristics inherent in industrial data, i.e., data distribution drift over time, cause significant challenges to soft sensor modeling. Existing normalization methods in the modeling procedure often rely on global static statistics, which fail to represent local statistical variations and result in poor predictive performance for soft sensor models in nonstationary environments. To overcome this limitation, we propose a novel plug-and-play normalization method named adaptive wavelet normalization network (AWNN) for soft sensor modeling in nonstationary industrial processes. Our method first utilizes the wavelet transform to identify nonstationary points and adaptively slices the input time series. Subsequently, fine-grained normalization is applied to each segment based on its local statistical properties. Finally, an attention-based BiLSTM predicts the statistical properties of the corresponding output window. This prediction enables accurate adaptive denormalization of the soft sensor model's output. Extensive experiments validated AWNN's effectiveness, demonstrating its ability to significantly improve the predictive performance across diverse soft sensor architectures. Moreover, it consistently outperforms traditional normalization methods when dealing with nonstationary data. Its plug-and-play nature facilitates seamless integration, offering a versatile and effective solution for improving soft sensor reliability under complex industrial environments.**

***Index Terms*—Adaptive data normalization, froth flotation, nonstationary process, soft sensor modeling, wavelet transform.**

Received 22 May 2025; revised 30 July 2025; accepted 4 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U23A20329, in part by the Open Research Project of the State Key Laboratory of industrial Control Technology, China under Grant ICT2024B69, in part by the Kunming Aca-demician Workstation under Grant YSZJGZZ-2023011, and in part by the High Performance Computing Center of Central South University. Paper no. TII-25-3375. *(Corresponding author: Liyang Qin.)*

The authors are with the School of Automation, Central South University, Changsha 410083, China (e-mail: xlwang@csu.edu.cn; lvguima@csu.edu.cn; lufanlei@csu.edu.cn; jiayizhou@csu.edu.cn; qly520@csu.edu.cn; ychh@csu.edu.cn).

Digital Object Identifier 10.1109/TII.2025.3609103

I. Introduction

IN INDUSTRIAL processes, accurate real-time monitoring of the key product index is one of the main objectives pursued by enterprises to optimize process control and improve product quality. This demand has driven the rapid development of soft sensor technology over the past few decades [1]. Soft sensors predict difficult-to-measure variables using easily observable process data. Accurate soft sensor models provide reliable information for subsequent process control, thereby playing a significant role in the closed-loop control of industrial operations.

Two fundamental soft sensor techniques are first-principle models and data-driven models. The former require extensive process knowledge to establish mechanistic equations. However, developing a universally effective, long-term first-principle model is often too complex. Traditional data-driven models (e.g., support vector regression) are low-cost and efficient for simple datasets [2]. With developments in hardware and industrial automation, process data have become large, dynamic, high-dimensional, and structurally diverse. These complexities in the data make it challenging for traditional machine learning models to learn effective patterns, thus limiting their performance [3].

To address the challenges posed by these complex data characteristics, extensive research has applied deep learning-based methods to soft sensor modeling [4]. For example, to handle issues, such as high nonlinearity, dynamics, and noise in industrial data, Geng et al. [5] used an improved gated convolutional neural network with multihead attention to model short-term patterns and time-dynamic correlations of different variables. Xie et al. [6] designed a supervised variational autoencoder (SVAE) to extract quality-related features and capture high-level representations, and improved the robustness of the deep SVAE through adversarial training. The industrial process data typically manifest as time series, so some studies have focused on capturing temporal variations in variables during soft sensor modeling. Liu et al. [7] proposed a supervised bidirectional long short-term memory (BiLSTM) network to extract dynamic temporal latent information from process variables, which significantly improved the predictive performance. Yuan et al. [8] introduced a spatiotemporal attention LSTM network using spatial attention to identify relevant input variables and temporal attention to discover relevant timesteps for prediction. Analyzing the process and incorporating domain knowledge can

also benefit soft sensor modeling and its interpretability [9], [10]. Zhou et al. [11] proposed a soft sensor modeling framework based on spatiotemporal deep LSTM embedded with domain knowledge for predicting particle size in the grinding process. This method first performs time delay analysis to align variables and selects auxiliary variables. Using domain knowledge of grinding, the entire process is decoupled into multiple subprocesses and a spatiotemporal structural model is constructed for soft sensor modeling.

Although these models are carefully designed with specific network architectures to handle various inherent characteristics of industrial data, such as correlation, temporal pattern, and process knowledge, another significant and universal challenge in complex process industries is nonstationary, which is often caused by fluctuations in the properties of raw materials and results in changes in production conditions. From the data perspective, nonstationary typically manifests as a drift in the statistical properties of data over time, making it difficult for deep learning-based models to learn effective and generalizable patterns, thereby severely limiting the applicability of existing deep learning-based soft sensor models.

To address nonstationarity in batch processes monitoring, Zhao et al. [12] designed statistical quantities based on static and dynamic monitoring, which provide comprehensive identification capability by detecting deviations and small disturbances that may be masked by closed-loop control. Zhu et al. [13] proposed a multihop attention graph convolutional network that adaptively learns variable coupling relationships to mitigate soft sensor model degradation in nonstationary environments. Liu et al. [14] introduced an adaptive working condition selection strategy for manifold learning. This strategy incorporates attention distance calculation to handle nonstationary features caused by changes in raw material properties. However, although these studies noted the performance degradation of soft sensor model due to nonstationary, they have not addressed the fundamental cause of nonstationary at the data level, which is the drift of the statistical properties. The method in Zhu et al.'s [13] work focused on building more accurate relationships between variables to develop models. The method in Liu et al.'s [14] work added a regularization term to address nonstationarity during network training.

In fact, soft sensor modeling is particularly sensitive to shifts in statistical properties, as shown in Fig. 1(a). The accuracy of predictions depends not only on the precision of the soft sensor model itself, but also largely on the data normalization process [15]. Existing studies typically employ global normalization methods, such as Z-score normalization, but this uniform approach has significant shortcomings in nonstationary industrial processes. As shown in Fig. 1(b), industrial data within the input window may have inherently different statistical distributions, and the statistical properties between the input and output windows may also differ. Therefore, existing normalization methods in soft sensor modeling have the following deficiencies:

1) *Lack of Fine-Grained Dynamic Normalization Processing:* Using global means and standard deviations for normalization cannot accurately reflect the dynamic distribution of industrial data. Such static normalization methods may lead to distortion in the normalized data, preventing the model from effectively capturing dynamic changes in the process, thereby affecting model training and prediction accuracy.
2) *Neglect of Distribution Drift in the Output Window:* The use of global statistical properties for denormalization of the output window is inappropriate. Since the statistical characteristics of the output window may change with the process, using fixed global parameters for denormalization can result in predictions that deviate from the actual distribution, reducing the reliability and accuracy of soft sensor predictions.

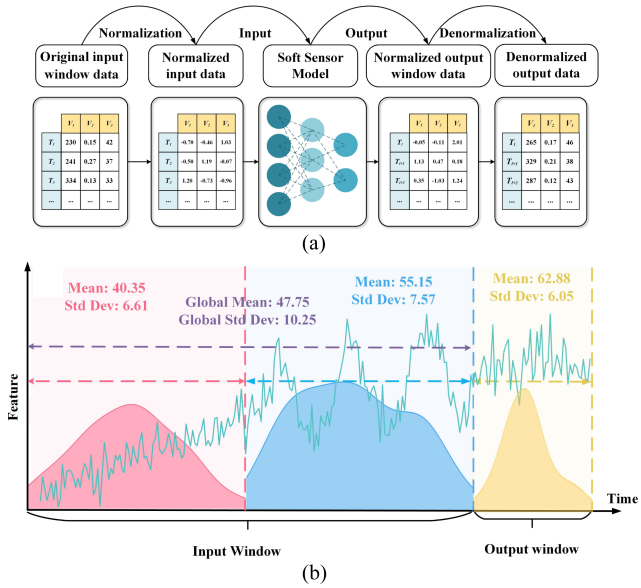

Fig. 1. Soft sensor in nonstationary processes. (a) Typical deep learning-based soft sensor workflow. (b) Distribution drift in input and output windows.

The dynamics and complexity of nonstationary industrial data make it difficult for existing deep learning-based soft sensor models to effectively capture these changing patterns, resulting in inaccurate predictions and unreliable subsequent control. Therefore, there is a need for a universal and highly adaptable normalization method that is capable of coping with the distribution drift issue of industrial data, thereby achieving more reliable and precise soft sensor predictions in complex industrial environments.

In this article, we propose the adaptive wavelet normalization network (AWNN), a plug-and-play adaptive normalization method that is suitable for soft sensor modeling in nonstationary industrial processes. AWNN enhances the adaptability and predictive performance of deep learning soft sensor models by addressing data distribution drift through fine-grained adaptive normalization and denormalization of industrial time-series data. Specifically, AWNN uses wavelet transforms to detect nonstationary points in the input time-series data. It then slices the data based on these points, calculates the statistical properties of each slice, and normalizes them separately. Next, AWNN constructs an attention-based BiLSTM network to model and predict

the statistical properties of the output window. The predicted statistical properties are used to denormalize the soft sensor model's output, thereby obtaining more accurate prediction results. We validated the effectiveness of AWNN using a dataset from a typical nonstationary flotation process. The results show that AWNN can obtain more accurate and reliable predictions, which are critical for process optimization and control in industrial operations. The contributions are as follows.

1) We propose a novel adaptive normalization framework that directly addresses distribution drift. By modeling local statistical properties, AWNN performs fine-grained normalization and denormalization, overcoming the limitations of traditional global methods in nonstationary environments.
2) AWNN is designed as a universal plug-and-play module. It can be seamlessly integrated with diverse deep learning-based soft sensor models without modifying their internal architectures, significantly enhancing their ability to adapt to nonstationary data and improving their overall predictive accuracy.
3) Through two types of comparison experiments on a real flotation plant dataset, we demonstrate that AWNN not only substantially improves the performance of various soft sensor models, but also consistently outperforms existing normalization methods, showing its practical value and high potential for industrial applications.

The rest of this article is organized as follows. Section II provides preliminary background on soft sensors in nonstationary industrial processes and discusses related work on data normalization methods in deep learning. Section III presents the detailed methodology of the AWNN. Section IV describes the experimental setup, including the dataset preparation, evaluation metrics, and implementation details, followed by a presentation and discussion of the comparative results. Finally, Section V concludes this article, summarizes the key findings and suggests directions for future research.

II. Preliminary

A. Soft Sensors for Nonstationary Industrial Process

With the continuous advancement of industrial IoT technology, soft sensors have become indispensable intelligent tools in modern industrial systems [16]. By utilizing easily measurable variables as inputs, soft sensors infer the values of variables that are difficult or impossible to measure directly [2]. Data-driven methods, particularly those employing deep learning techniques, have demonstrated immense potential due to their ability to handle high-dimensional and nonlinear data. Consequently, these methods have been widely adopted in process monitoring, quality prediction, and optimal control. In nonstationary industrial processes, the data generated during operation exhibit complex dynamic characteristics due to factors, such as fluctuations in raw material properties, equipment performance degradation, and operational adjustments.

A typical example is soft sensor modeling for concentrate grade in froth flotation. Flotation is a widely used method in mineral processing to separate valuable minerals from ore. The concentrate grade refers to the quality of the extracted mineral in the flotation process. Real-time monitoring of the concentrate grade is crucial for ensuring the efficiency and stability of the flotation, as it provides immediate feedback for operational adjustments. In recent years, machine vision techniques are usually combined with soft sensor modeling for grade prediction. By capturing froth images from the flotation cells, visual features, such as froth size, froth color, and froth speed can be extracted. These features provide real-time information about the state of the froth and are closely related to the grade. Incorporating these visual features into data-driven soft sensor models enhances the ability to predict concentrate grade more accurately [17]. However, the dynamic properties of raw ores cause nonstationarity in the process data. These fluctuations affect the attachment behavior of mineral particles and the stability of the froth, resulting in nonstationarity and posing significant challenges to current deep learning-based soft sensor models.

B. Data Normalization Methods in Deep Learning

Data normalization is a critical step in training deep learning models across various fields. It eliminates the influence of different units and scales, balances feature weights, and thereby improves the model's stability and generalization. Traditional normalization methods, such as Z-score normalization typically assume that the data distribution is static. When dealing with nonstationary data, global statistical measures cannot represent temporal variations and may fail to adapt to dynamic changes in the data distribution. To address this, Ogasawara et al. [15] proposed an adaptive normalization method that transforms nonstationary time series into stationary series by partitioning them into nonoverlapping windows and extracting statistical features within these windows. Passalis [18] designed deep adaptive input normalization (DAIN), a deep learning layer that adaptively adjusts normalization to handle changes in data distribution, improving predictive performance without relying on fixed statistics. Kim et al. [19] introduced reversible instance normalization (RevIN), which performs normalization and denormalization through learnable affine transformations. It removes statistical information in the input layer and restores it in the output layer. Fan et al. [20] proposed a coefficient network that maps input sequences to two learnable coefficients, namely, a level coefficient representing the overall scale and a scaling coefficient representing the fluctuation amplitude, thereby alleviating distribution drift in time series forecasting.

While these methods address distribution drift to some extent, they still rely on fixed statistical properties within the input window for normalization, which means they overlook fine-grained local changes in time series data in industrial processes. Therefore, existing methods remain insufficient in handling local nonstationarity and cannot fully address complex distribution drift problems in industrial soft sensors.

III. Methodology

The overall architecture of the proposed AWNN is illustrated in Fig. 2. AWNN employs wavelet decomposition to perform multiscale analysis, decomposing input data into components

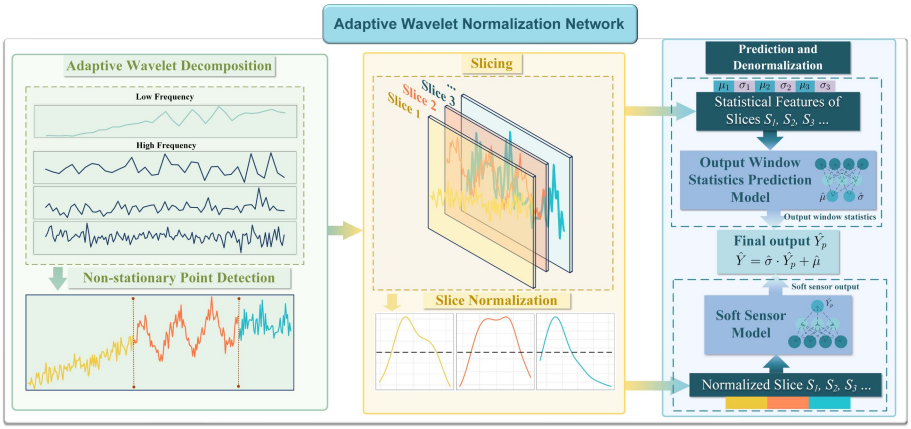

Fig. 2. Architecture of the AWNN.

at different frequencies to capture global trends and local fluctuations. By detecting nonstationary points, the data are segmented into relatively stationary slices. Each slice is individually normalized to eliminate distribution differences over time. An attention-based BiLSTM is then used to predict the statistical properties of the output window. Finally, the output of the soft sensor model is denormalized using the predicted statistical properties.

A. Adaptive Wavelet Slicing

Wavelet transformation not only exhibits excellent localization properties in both the time and frequency domains but is also particularly well-suited for analyzing complex nonstationary signals. To address the challenges posed by local statistical variations in nonstationary process data, AWNN employs wavelet transformation to identify nonstationary points within the input time series data and partition it into multiple subsequences, i.e., slices. The scheme comprises three stages: Wavelet decomposition, nonstationary point detection, and slice normalization.

1) Wavelet Decomposition: First, we perform wavelet decomposition on the data. Wavelet decomposition is a multiresolution analysis method that can simultaneously capture both global trends and local fluctuations by decomposing the signal into frequency components at different scales [21]. Specifically, the discrete wavelet transform (DWT) decomposes the input data into low-frequency components A_j and high-frequency components D_j, which represent the global smooth trend and local variations of the data, respectively. Starting with the original signal as the initial approximation $A_0[k] = x[k]$, the decomposition at each subsequent level j is calculated by applying a low-pass filter h and a high-pass filter g to the approximation coefficients A_{j-1} from the previous level, followed by down-sampling

$$A_j[n] = \sum h[k-2n] \cdot A_{j-1}[k] \tag{1}$$

$$D_j[n] = \sum g[k-2n] \cdot A_{j-1}[k] \tag{2}$$

where $x[k]$ is the input data. The approximation coefficients A_j are generated by applying h to the approximation coefficients of the previous level. This process retains the slower variations of the signal, effectively capturing the global smooth trend at scale j. Meanwhile, the detail coefficients D_j are obtained using g to the same input as the low-pass filter. These coefficients capture the rapid changes and local details removed by the low-pass filter. Due to the inherent time-localization property of wavelet functions, D_j primarily reflect localized variations, transients, or singularities occurring around specific time points within the signal [22]. The choice of the specific wavelet filters should consider the characteristics of the industrial data (e.g., periodicity and smoothness level). This recursive process allows us to analyze the signal's characteristics across multiple scales simultaneously.

2) Nonstationary Point Detection: To identify nonstationary points, the multiscale coefficients obtained from wavelet decomposition need to be further processed. Nonstationary points often appear as sudden changes or significant variations in the high-frequency detail coefficients $D_j[k]$. Therefore, we detect nonstationary points by analyzing these detail coefficients. The specific process is as follows: For each scale j, we define the energy $E_j[k]$ as the squared magnitude of the detail coefficient

$$E_j[k] = |D_j[k]|^2. \tag{3}$$

Using the square quantifies the local energy density of high-frequency variations and emphasizes larger coefficients, making it a sensitive indicator for detecting abrupt changes or local irregularities often associated with nonstationary points [21]

$$E_j^{\text{norm}}[k] = \frac{E_j[k] - \mu_{E_j}}{\sigma_{E_j}} \tag{4}$$

where μ_E and σ_E are the mean and standard deviation of the energy sequence $E_j[k]$ at scale j. At each time point k, aggregate the normalized energies across all scales to obtain the total energy $E_{\text{total}}[k]$. Then, set an energy threshold θ. When $E_{\text{total}}[k] > \theta$, a nonstationary point is detected at time k. This signifies a point, where the aggregated high-frequency energy, indicating local signal irregularity, exceeds a statistically determined level. The choice of θ is based on the statistical properties

of the data

$$E_{\text{total}}[k] = \sum_j E_j^{\text{norm}}[k], \theta = \mu_{E_{\text{total}}} + \rho \cdot \sigma_{E_{\text{total}}} \tag{5}$$

where $\mu_{E_{\text{total}}}$ and $\sigma_{E_{\text{total}}}$ are the mean and standard deviation of the $E_{\text{total}}[k]$, ρ is a tuning parameter. The original time series is divided into $n+1$ slices $\{S_0, S_1, \ldots, S_n\}$ by the detected nonstationary points $\{t_k | E_{\text{total}}[t_k] > \theta\}$. Within each slice, the statistical properties of the data are relatively stable, which provides a solid basis for subsequent normalization processing.

3) Slice Normalization: After detecting nonstationary points and dividing the time series into multiple relatively stationary slices S_i, we perform separate normalization processing for each slice. This approach eliminates statistical distribution differences between different slices, enabling the model to accurately capture the local dynamic characteristics of the data. For slice S_i, which consists of N_i data points $x_k^{(i)}$, calculate the mean and standard deviation, and standardize the data within each slice using its own statistical properties

$$\begin{cases} \mu_i = \frac{1}{N_i}\sum_{k\in S_i} x_k, \sigma_i \quad = \sqrt{\frac{1}{N_i}\sum_{k\in S_i}(x_k-\mu_i)^2} \\ \tilde{x}_k = \frac{x_k-\mu_i}{\sigma_i}. \end{cases} \tag{6}$$

This slice-based normalization method ensures that the normalized data across slices have consistent scales and distribution characteristics, which facilitates model learning and capturing local dynamic characteristics. However, the detection of nonstationary points at multiple scales may lead to very short slices, we proposed a smoothing strategy at the slice boundaries to avoid discontinuities introduced by frequent switching of normalization parameters. For adjacent slices S_i and S_{i+1}, we define a transition interval Δt near the slice boundary t_{i+1}. Within the interval $[t_{i+1} - \Delta t, t_{i+1}]$, we perform a smooth transition of the normalization parameters

$$\begin{cases} \mu_s(k) = \omega(k)\mu_i + [1-\omega(k)]\mu_{i+1} \\ \sigma_s(k) = \omega(k)\sigma_i + [1-\omega(k)]\sigma_{i+1} \end{cases} \tag{7}$$

$$\omega(k) = \frac{t_{i+1} - k}{\Delta t} \tag{8}$$

where $\omega(k)$ is a smoothing weight function defined over the transition interval, satisfying $0 \le \omega(k) \le 1$, such that $\omega(k)$ approaches 1 as k approaches $t_{i+1} - \Delta t$ (end of slice S_i) and approaches 0 as k approaches t_{i+1} (start of slice S_{i+1}). By this method, the normalization parameters in the transition interval move gradually from the statistical parameters of the previous slice to those of the next slice, avoiding abrupt changes and ensuring smoothness in the normalized data. Furthermore, to handle instances where the adaptive slicing process might generate segments shorter than a practical minimum length $N_{\min}$, which could lead to unreliable statistical calculations, we employ a merging strategy. Specifically, any slice S_i with length $N_i < N_{\min}$ is merged with its adjacent slice before calculating the statistical properties used for normalization. This ensures robustness against overly fragmented segmentation.

B. Prediction of the Statistical Features in the Output Window

After completing the adaptive slicing and normalization of the input sequence, we need to predict the mean and standard deviation of the data in the output window. This prediction is essential for accurate denormalization of the soft sensor model's output during the inference phase. Instead of using the normalized input series directly as inputs, we utilize the statistical features of each slice as inputs to the statistical prediction model. This design reduces the complexity of the model, emphasizes the influence of statistical features on the output, and enhances training efficiency and predictive performance.

1) Statistics Prediction Model: The statistics prediction model aims to learn the relationship between the statistical features of the input slices and those of the output window. We propose a BiLSTM enhanced with residual connections and an attention mechanism to capture temporal dependencies and assign dynamic importance weights to different timesteps.

Let the input sequence be divided into S slices, and the statistical properties of these slices constitute the input, which is $X_s = \{(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_S, \sigma_S)\}$. Note that although each element in this sequence has a fixed 2-D feature vector, the sequence length S can vary depending on the nonstationary points identified in the input window. The input X_s is then fed into a BiLSTM layer to capture bidirectional temporal features. The BiLSTM's recurrent architecture, which shares weights across all time steps, endows it with the inherent capability to process sequences of varying lengths. The hidden states h_s produced by each of these layers for a given slice are then combined through concatenation

$$\begin{cases} \overrightarrow{h_s} = \text{LSTM}_{fwd}(x_s, \overrightarrow{h_{s-1}}) \\ \overleftarrow{h_s} = \text{LSTM}_{bwd}(x_s, \overleftarrow{h_{s+1}}) \\ h_s = [\overrightarrow{h_s}; \overleftarrow{h_s}] \in \mathbb{R}^{2d} \end{cases} \tag{9}$$

where d is the number of hidden units in each LSTM layer. The BiLSTM's objective here is distinct from standard pointwise forecasting. Instead, it learns a mapping from the sequence of statistical properties X_s (derived from the current input window) to the overall statistical properties of the target output window. The bidirectional processing captures comprehensive temporal patterns within the already observed X_s sequence, enabling a more informed prediction of the output window's characteristics. We, then, introduce a residual connection to enhance the model's training efficiency and generalization ability. The input statistical properties are flattened into a vector $x_{\text{flat}} = [\mu_1, \sigma_1, \mu_2, \sigma_2, \ldots, \mu_S, \sigma_S] \in \mathbb{R}^{2S}$, then apply a linear projection to match the dimensionality of the BiLSTM output

$$x_{\text{proj}} = W_{\text{proj}} x_{\text{flat}} + b_{\text{proj}} \tag{10}$$

where $W_{\text{proj}} \in \mathbb{R}^{2d \times 2S}$, $b_{\text{proj}} \in \mathbb{R}^{2d}$. Next, we incorporate an attention mechanism to dynamically adjust the weights of each time step, which allows the model to learn the importance of each hidden state automatically. The attention scores e_s for each hidden state h_s are calculated as follows:

$$e_s = \mathbf{v}^\top \tanh(W_h h_s + b_h) \tag{11}$$

where $W_h \in \mathbb{R}^{d_a \times 2d}$, $b_h \in \mathbb{R}^{d_a}$, and $\mathbf{v} \in \mathbb{R}^{d_a}$ are learnable parameters, and d_a is the dimensionality of the attention layer. The attention scores are normalized using the softmax function to obtain the attention weights α_s, and the output vector c is computed as

$$\alpha_s = \frac{\exp(e_s)}{\sum_{k=1}^{S} \exp(e_k)}, c = \sum_{s=1}^{S} \alpha_s h_s + x_{\text{proj}}. \tag{12}$$

Finally, the output vector c is passed through fully connected layers to predict the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the output window

$$\hat{\mu} = w_\mu^\top c + b_\mu, \hat{\sigma} = \text{ELU}(w_\sigma^\top c + b_\sigma) + \delta \tag{13}$$

where $w_\mu, w_\sigma \in \mathbb{R}^{2d}$, and $b_\mu, b_\sigma \in \mathbb{R}$ are learnable parameters, and $\delta > 0$ is a small positive constant to ensure $\hat{\sigma}$ is positive. The exponential linear unit (ELU) activation function is defined as

$$\text{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0. \end{cases} \tag{14}$$

2) Denormalizing the Output of the Soft Sensor Model: In nonstationary industrial processes, the statistical properties of data can drift over time. Traditional denormalization methods use global statistical parameters, which fail to reflect the current data distribution, potentially leading to biases in the predictions. By utilizing the predicted statistical properties of the output window, we can adaptively adjust the denormalization process to accurately reflect the future state of the process and compensate for distribution changes due to nonstationarity

$$\hat{y} = \hat{y}_s \times \hat{\sigma} + \hat{\mu}. \tag{15}$$

where $\hat{y}$ is the final prediction after denormalization, representing the predicted value on the original scale. $\hat{y}_s$ is the normalized prediction output from the soft sensor model. This adaptive denormalization method effectively addresses data distribution shifts in nonstationary industrial processes, enhancing the robustness and reliability of the soft sensor model in practical applications.

C. Two-Stage Training Strategy

Considering the distinct functionalities and objectives of the statistical prediction model and the soft sensor model, as shown in Fig. 3, we divide the training process into two stages. This division ensures that each part of the model can learn effectively and perform its role optimally. The primary goal of the first stage is to train the statistical prediction model independently. This model focuses on learning patterns and relationships between the statistical properties of the input slices and those of the output window. The accuracy of the statistical prediction directly affects the effectiveness of the adaptive denormalization process and, consequently, the overall predictive performance of the soft sensor model. We treat the training of the statistical prediction model as an independent optimization problem, aiming to minimize the loss between the predicted and actual statistical properties of the output window. The loss function used is the mean squared error

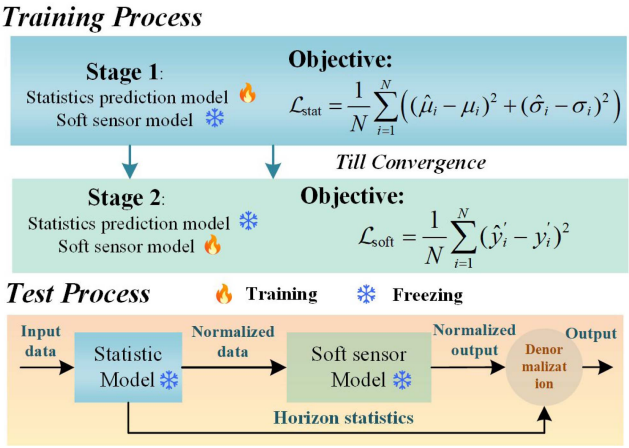

Fig. 3. Training and testing process of AWNN.

(MSE)

$$\mathcal{L}_{\text{stat}} = \frac{1}{N}\sum_{i=1}^{N}\left((\hat{\mu}_i - \mu_i)^2 + (\hat{\sigma}_i - \sigma_i)^2\right). \tag{16}$$

After training the statistical prediction model, we proceed to train the soft sensor model in the second stage. The parameters of the statistical prediction model are frozen and not updated further. The soft sensor model aims to learn the mapping from the normalized input data to the normalized output. The loss function used is also the MSE. Such a two-stage training strategy clearly separates tasks, with the statistical prediction model handling output window statistics and the soft sensor model mapping normalized inputs to outputs. In addition, decoupling optimization objectives reduces complexity, improves convergence speed, and overall model performance

$$\mathcal{L}_{\text{soft}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i' - y_i')^2. \tag{17}$$

D. Complexity Analysis

The computational complexity of the AWNN module primarily stems from three components. First, the adaptive wavelet slicing stage, including the DWT for decomposition and subsequent calculations for nonstationary point detection and segment statistics, exhibits a linear complexity of $\mathcal{O}(N)$ with respect to the input window length N. Second, the prediction of statistical features using the attention-based BiLSTM operates on a sequence of S segment statistics (where $S \ll N$). Its complexity is approximately $\mathcal{O}(S \times d^2)$, which is dependent on the number of segments S and the BiLSTM's hidden dimension d. Third, the normalization and denormalization steps involve elementwise operations with a complexity of $\mathcal{O}(N)$. Overall, AWNN introduces computational overhead that scales linearly with the input window length N and polynomially with the hyperparameters of the statistics prediction model operating on a reduced sequence length S. In comparison to potentially more complex core soft sensor models, the computational overhead of AWNN remains relatively moderate, primarily due to the

efficient wavelet analysis and the operation of the BiLSTM on a reduced number of segments.

IV. Industrial Applications

In this section, we validate the effectiveness of the proposed AWNN in a real froth flotation industrial process, using the concentrate grade soft sensor modeling as a case study. We conduct two types of experiments to assess the performance of AWNN. In the model embedding experiments, AWNN is embedded into different deep learning-based soft sensor models to evaluate its effectiveness in improving model performance. In the normalization comparison experiments, AWNN is compared with various normalization methods to demonstrate its superiority in addressing data nonstationarity.

A. Dataset Preparation and Evaluation

In the current field of flotation concentrate grade soft sensor modeling, a common approach is to use features extracted from froth images as inputs. We collected data spanning from May 10, 2022, to August 15, 2022 from a froth flotation plant, with measurements recorded at 30-min intervals, yielding a total of 5335 samples [23]. The input features obtained include froth depth, flow rate, froth density, mineral load on froth, froth size distribution, grayscale value, RGB channel values, froth speed and froth stability. These features are closely related to the production efficiency of the flotation process and the concentrate grade. The target variable for prediction is the concentrate grade, which reflects the quality of the product in the flotation process. The dataset was split using a time-based approach to prevent data leakage: 70% was allocated for training, 10% for validation, and the remaining 20% for testing.

To illustrate the inherent nonstationary characteristics of this industrial dataset, Fig. 4 provides a visual analysis of the target variable, concentrate grade, over a representative period. Fig. 4(a) shows the raw time series, and Fig. 4(b) depicts the evolution of its local mean and standard deviation. As observed in Fig. 4(b), both the local mean and standard deviation exhibit significant drifts and fluctuations over time, which shows the time-varying nature of the data's statistical properties. This nonstationarity, likely driven by factors, such as changing ore properties and operational adjustments, presents a key challenge for building robust soft sensor models.

To conduct a comprehensive comparative analysis, we employed six evaluation metrics to assess the performance of different methods: Root mean squared error (RMSE), mean absolute error (MAE), MSE, mean absolute percentage error (MAPE), coefficient of determination (R^2) and residual standard error (RSE). All experiments were conducted on a computing platform equipped with an NVIDIA Tesla V100 GPU with 32 GB of memory. The models were programmed in Python 3.10 and implemented using the PyTorch framework.

B. Experiment Settings for Comparison of Different Models

To thoroughly validate the effectiveness of the proposed AWNN, we conducted two sets of experiments. In the embedded

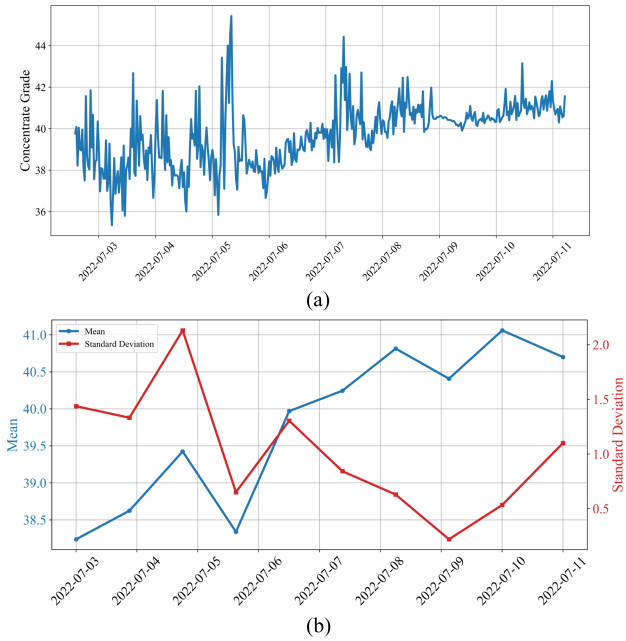

Fig. 4. Dataset nonstationarity analysis. (a) Raw concentrate grade time series. (b) Evolution of its local mean and standard deviation over the different periods.

comparison experiments, we selected several state-of-the-art deep learning-based models and integrated AWNN into them. These models are listed in the Table I. By replacing their original normalization components with AWNN, we evaluated the impact of AWNN on their performance for the task of soft sensor modeling. In the normalization comparison experiments, we used Autoformer as the baseline soft sensor model. By comparing the predictive performance of AWNN with various popular normalization techniques implemented within Autoformer, we assessed its advantages in handling nonstationary industrial process data.

In these experiments, important hyper-parameters were carefully selected to ensure fair and comprehensive evaluations. We explored input window sizes L ranging from 12 to 200 to determine the optimal length of the historical data. The number of hidden units of BILSTM N_l, was selected from 50 to 1000 to assess the impact on model performance. Further hyperparameter analysis details of these key hyperparameters are provided in the Section IV-D. The output window size (denoted as H) was set to $\{2, 10, 24\}$. These specific horizons were chosen based on their practical relevance for process monitoring and control adjustments in the flotation process, informed by operational experience regarding the typical time scales of ore property variations and control cycles. In general applications, the optimal H might be treated as a hyperparameter determined by validation or specific task requirements. The batch size was set to be 64, and models were trained for up to 40 epochs, with early stopping based on validation loss to prevent overfitting and ensure generalization. We used the Adam optimizer to minimize the loss function. The learning rate was initially set to 0.001 and reduced by a factor of 0.1 if the validation loss plateaued for 5 consecutive epochs. The results can be found in the following section.

TABLE I
EMBEDDED MODELS AND NORMALIZATION METHODS

Embedded models	**Description**
LSTNet [24]	A network combining CNN and RNN to extract short-term local dependencies and long-term patterns, respectively.
MSGNET [25]	A multiscalegraph neural network, capturing interseries correlations across different scales.
Autoformer [26]	A transformer model using auto-correlation decomposition.
Informer [27]	A transformer model with a sparse attention mechanism.
Crossformer [28]	A transformer model that uses a two-stage attention mechanism and hierarchical encoder-decoder architecture.
Normalization methods	**Description**
Z-Score	The most common method that standardizes the data by subtracting the mean and dividing by the standard deviation.
Robustscaler	Scales the data using the median and the interquartile range, reducing the influence of outliers.
DAIN [18]	Normalizes data by adaptively learning the data distribution, improving performance in nonstationary and multimodal tasks.
RevIN [19]	Addresses the distribution shift problem by normalizing and denormalizing data using a learnable affine transformation.
DishTS [20]	Uses a dual-network approach to capture intra-space and inter-space distribution changes.

TABLE II
PERFORMANCE OF MODELS WITH AND WITHOUT AWNN ACROSS DIFFERENT H

Model	H	wAWNN / woAWNN (Decrease%)					
		MSE	MAE	RMSE	MAPE	R^2	RSE
LSTNet	2	**0.350**/0.499 (30%)	**0.393**/0.486 (19%)	**0.591**/0.704 (16%)	**1.928**/2.410 (20%)	**0.696**/0.566 (–23%)	**0.551**/0.659 (16%)
	10	**0.408**/0.556 (27%)	**0.427**/0.512 (17%)	**0.639**/0.744 (14%)	**2.045**/2.569 (20%)	**0.646**/0.517 (–25%)	**0.595**/0.695 (14%)
	24	**0.495**/0.618 (20%)	**0.471**/0.543 (13%)	**0.704**/0.866 (19%)	**2.250**/2.782 (19%)	**0.572**/0.466 (–23%)	**0.654**/0.731 (11%)
MSGNET	2	**0.326**/0.368 (11%)	**0.358**/0.393 (9%)	**0.571**/0.606 (6%)	**2.229**/2.318 (4%)	**0.756**/0.725 (–4%)	**0.494**/0.525 (6%)
	10	**0.414**/0.524 (21%)	**0.398**/0.484 (18%)	**0.644**/0.742 (13%)	**2.283**/2.370 (4%)	**0.692**/0.610 (–13%)	**0.555**/0.624 (11%)
	24	**0.563**/0.712 (21%)	**0.464**/0.580 (20%)	**0.751**/0.844 (11%)	**2.867**/3.183 (10%)	**0.586**/0.476 (–23%)	**0.644**/0.724 (11%)
Autoformer	2	**0.349**/0.403 (13%)	**0.391**/0.437 (11%)	**0.591**/0.653 (9%)	**1.925**/2.204 (13%)	**0.696**/0.650 (–7%)	**0.551**/0.592 (7%)
	10	**0.406**/0.504 (19%)	**0.426**/0.498 (14%)	**0.637**/0.710 (10%)	**2.032**/2.453 (17%)	**0.648**/0.563 (–15%)	**0.594**/0.661 (10%)
	24	**0.493**/0.627 (21%)	**0.470**/0.567 (17%)	**0.702**/0.792 (11%)	**2.235**/2.781 (20%)	**0.574**/0.458 (–25%)	**0.653**/0.736 (11%)
Informer	2	**0.338**/0.410 (18%)	**0.365**/0.411 (11%)	**0.581**/0.640 (9%)	**2.315**/2.802 (17%)	**0.747**/0.693 (–8%)	**0.503**/0.554 (9%)
	10	**0.425**/0.553 (23%)	**0.416**/0.465 (11%)	**0.652**/0.743 (12%)	**2.412**/2.984 (19%)	**0.684**/0.589 (–16%)	**0.562**/0.640 (12%)
	24	**0.616**/0.664 (7%)	**0.505**/0.511 (1%)	**0.785**/0.851 (8%)	**3.233**/3.515 (8%)	**0.547**/0.512 (–7%)	**0.673**/0.699 (4%)
Crossformer	2	**0.318**/0.350 (9%)	**0.346**/0.385 (10%)	**0.564**/0.590 (4%)	**2.100**/2.308 (9%)	**0.762**/0.738 (–3%)	**0.488**/0.512 (5%)
	10	**0.409**/0.447 (9%)	**0.393**/0.473 (17%)	**0.639**/0.665 (4%)	**2.227**/2.441 (9%)	**0.696**/0.668 (–4%)	**0.551**/0.577 (5%)
	24	**0.560**/0.595 (6%)	**0.460**/0.510 (10%)	**0.749**/0.773 (3%)	**2.862**/2.868 (-)	**0.588**/0.562 (–5%)	**0.642**/0.662 (3%)

Decrease percentages are shown in parentheses; for R^2, a negative percentage indicates an improvement.

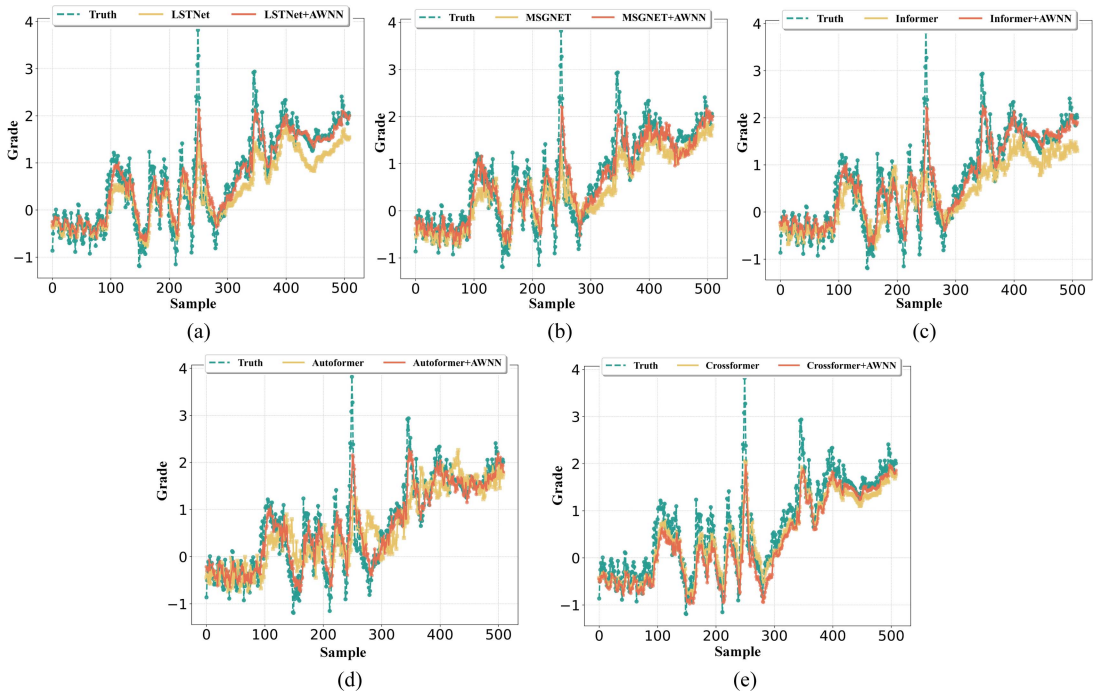

Fig. 5. Performance Comparison of Models with and without AWNN (H=10). (a) LSTNet. (b) MSGNET. (c) Informer. (d) Autoformer. (e) Crossformer.

TABLE III
PERFORMANCE METRICS OF DIFFERENT NORMALIZATION METHODS

H	2/10/24					
Metrics	MSE	MAE	RMSE	MAPE	R^2	RSE
Z-Score	0.44/0.47/0.56	0.44/0.46/0.51	0.66/0.70/0.75	6.67/4.43/4.97	0.55/0.51/0.42	0.67/0.70/0.76
Robustscaler	0.40/0.50/0.62	0.44/0.50/0.56	0.63/0.71/0.79	2.13/2.34/2.71	0.65/0.50/0.46	0.60/0.67/0.74
DAIN	0.40/0.50/0.63	0.44/0.50/0.56	0.64/0.71/0.79	2.20/2.45/2.78	0.65/0.56/0.46	0.59/0.66/0.74
RevIN	0.40/0.45/0.56	0.42/0.46/0.51	0.63/0.67/0.75	5.44/4.43/5.50	0.59/0.54/0.43	0.64/0.68/0.76
DishTS	0.36/0.42/0.51	0.40/0.43/0.49	0.60/0.65/0.72	1.96/2.08/2.27	0.68/0.64/0.56	0.56/0.60/0.67
AWNN	**0.35/0.41/0.49**	**0.39/0.42/0.47**	**0.59/0.64/0.70**	**1.93/2.03/2.24**	**0.70/0.65/0.57**	**0.55/0.59/0.65**

C. Results and Discussions

The embedded experimental results are summarized in Table II and Fig. 5, which show the performance of five models with (w/ AWNN) and without (w/o AWNN) the integration of the AWNN across different H. It can be seen that across all models and H, integrating AWNN consistently leads to improved performance. This is evident from the lower error metrics (MSE, MAE, RMSE, and MAPE) and higher R^2 when AWNN is used. The reductions in MSE and RMSE indicate that AWNN enhances the models' ability to capture the variance in the data, and decreases in MAE and MAPE suggest improved accuracy and reduced average prediction errors. For instance, Crossformer with AWNN at $H = 2$ achieves an MSE of 0.318 compared to 0.350 without AWNN, indicating that AWNN helps models capture immediate temporal patterns more effectively. As H increases, the challenges of nonstationary become more pronounced, but AWNN continues to help models mitigate these effects. LSTNet's MAPE at $H = 24$ decreases from 2.782% without AWNN to 2.250% with AWNN, highlighting consistent improvements. These consistent improvements across different models and H demonstrate AWNN's effectiveness in addressing the challenges posed by data distribution shifts and nonstationary inherent in industrial data.

It can also be seen that AWNN improves both transformer-based models (Autoformer, Informer, and Crossformer) and recurrent or graph neural network-based models (LSTNet and MSGNET), demonstrating its universality and applicability across different deep learning-based architectures. This is particularly valuable in industrial contexts where model selection may be constrained by specific requirements. From Fig. 5, it is also clear that models with AWNN are more effective in capturing sharp fluctuations in the data, which are often characteristic of nonstationary industrial processes. This ability to adapt to the dynamic nature of complex industrial data further highlights AWNN's effectiveness in addressing nonstationarity.

The performance for each normalization method across different H is summarized in Table III. It can be seen that AWNN consistently outperforms traditional normalization methods (Z-Score and RobustScaler) and advanced techniques designed to handle distribution shifts (DAIN, RevIN, and DishTS). The superior performance of AWNN can be attributed to its adaptive approach, which dynamically adjusts normalization based on local statistical properties identified through wavelet analysis. By detecting nonstationary points and applying fine-grained normalization, AWNN effectively aligns data distributions, leading to more accurate predictions. Although methods, such as DishTS also addresses distribution shifts, AWNN's combination of wavelet-based slicing and output statistical prediction offers better performance in capturing intricate patterns of nonstationary data. This further highlights AWNN's unique ability to enhance model robustness in dynamic industrial environments.

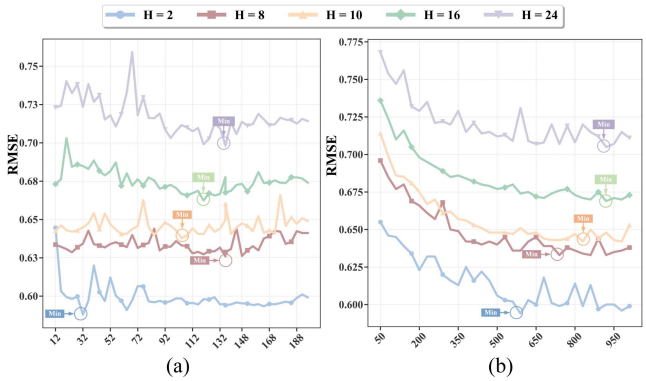

Fig. 6. Effect of input window size and BiLSTM unit number on model performance. (a) Input window size L. (b) BiLSTM unit number N_l.

D. Hyperparameter Sensitivity Analysis

To comprehensively evaluate the influence of hyperparameters on model performance, we conducted experiments focusing on two critical hyperparameters: The input window size (L) and the number of BiLSTM units (N_l). All experiments utilized the Autoformer as the base model and were conducted on the same training and testing datasets, with RMSE employed as the primary evaluation metric. The experimental results are depicted in Fig. 6. From Fig. 6(a), it is evident that L has a significant impact on the model's predictive performance. For shorter prediction horizons (e.g., $H = 2$), the model achieves the lowest RMSE when $L = 32$, indicating that a moderate input window captures sufficient historical information for accurate short-term predictions. As L increases beyond this point, RMSE shows little change, suggesting that enlarging L does not provide substantial additional benefit for short-term forecasting due to the diminishing relevance of older data points. In contrast, for longer prediction horizons (e.g., $H = 24$), the model attains the lowest RMSE at larger input window sizes, implying that long-term predictions benefit from incorporating a broader historical context to capture extended temporal dependencies. However, the reduction in RMSE with increasing L is limited, and beyond

a certain value, further increases yield negligible gains. This suggests that while a larger input window can help capture long-term dependencies, the model's capacity to leverage this information may be constrained by factors, such as noise accumulation and overfitting to irrelevant patterns. Therefore, for each prediction horizon, there exists an optimal input window size that balances the need for historical information with the risk of incorporating noise and unnecessary complexity. From Fig. 6(b), it can be seen that as N_l increases, the RMSE generally decreases, indicating that a larger model capacity enhances the ability to learn complex temporal patterns. For short-term predictions ($H = 2$), RMSE decreases significantly as N_l increases up to a certain point (e.g., $N_l = 600$), after which the improvement tapers off. For longer prediction horizons (e.g., $H = 24$), the minimum RMSE corresponds to higher values of N_l (around $N_l = 900$), reflecting the need for greater model complexity to handle more intricate temporal dependencies. However, the rate of RMSE reduction diminishes as N_l increases further, and may even begin to rise. This trend suggests that beyond a certain capacity, the model becomes prone to overfitting, and additional units contribute little to improving performance, while increasing computational overhead.

V. Conclusion

This article addressed the significant challenge posed by nonstationary data distributions in industrial soft sensor modeling, which often degrade the performance of deep learning approaches. We introduced the AWNN, a novel plug-and-play module designed to enhance model adaptability through a fine-grained, adaptive normalization and denormalization process. AWNN uniquely leverages wavelet analysis for adaptive segmentation and employs an attention-based BiLSTM to explicitly predict future output statistics, thus enabling accurate denormalization.

Experimental results showed that integrating AWNN significantly boosts the predictive accuracy and the robustness of various deep learning soft sensor models when handling nonstationary data, and outperforms conventional and other adaptive normalization methods. As an effective and versatile solution, AWNN offers a practical way to mitigate distribution drift without altering the core sensor model architecture, thereby improving the reliability of soft sensors in dynamic industrial environments by tackling data distribution shifts. Future work will focus on extending AWNN's application to diverse industrial processes, potentially combining it with online learning or model adaptation techniques to address broader nonstationary challenges, such as distribution shifts and evolving input–output relationships, and to improve responsiveness to rapid changes.

References

[1] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[2] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Trans. Ind. Inform.*, vol. 17, no. 9, pp. 5853–5866, Sep.. 2021.

[3] Q. Jiang, X. Yan, H. Yi, and F. Gao, "Data-driven batch-end quality modeling and monitoring based on optimized sparse partial least squares," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4098–4107, May 2020.

[4] J. M. M. de Lima and F. M. U. de Araujo, "Ensemble deep relevant learning framework for semi-supervised soft sensor modeling of industrial processes," *Neurocomputing.*, vol. 462, pp. 154–168, Oct. 2021.

[5] Z. Geng, Z. Chen, Q. Meng, and Y. Han, "Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1521–1529, Mar. 2022.

[6] Y. Xie, J. Wang, S. Xie, and X. Chen, "Adversarial training-based deep layer-wise probabilistic network for enhancing soft sensor modeling of industrial processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 972–984, Feb. 2024.

[7] C. F. Lui, Y. Liu, and M. Xie, "A supervised bidirectional long short-term memory network for data-driven dynamic soft sensor modeling," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[8] X. Yuan, L. Li, Y. A. W. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4404–4414, May 2021.

[9] R. Guo, H. Liu, G. Xie, Y. Zhang, and D. Liu, "A self-interpretable soft sensor based on deep learning and multiple attention mechanism: From data selection to sensor modeling," *IEEE Trans. Ind. Inform.*, vol. 19, no. 5, pp. 6859–6871, May 2023.

[10] H. Zhang, Z. Tang, Y. Xie, Z. Yin, and W. Gui, "ES-Net: An integration model based on encoder–decoder and Siamese time series difference network for grade monitoring of zinc tailings and concentrate," *IEEE Trans. Ind. Electron.*, vol. 70, no. 11, pp. 11819–11830, Nov. 2023.

[11] J. Zhou, C. Yang, X. Wang, and S. Cao, "A soft sensor modeling framework embedded with domain knowledge based on spatio-temporal deep LSTM for process industry," *Eng. Appl. Artif. Intell.*, vol. 126, 2023, Art. no. 106847.

[12] C. Zhao, W. Wang, C. Tian, and Y. Sun, "Fine-scale modeling and monitoring of wide-range nonstationary batch processes with dynamic analytics," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8808–8818, Sep. 2021.

[13] K. Zhu and C. Zhao, "Dynamic graph-based adaptive learning for online industrial soft sensor with mutable spatial coupling relations," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9614–9622, Sep.. 2023.

[14] C. Liu, Y. Wang, C. Yang, H. Leung, and X. Yin, "Adaptive attention-driven manifold regularization for deep learning networks: Industrial predictive modeling applications and beyond," *IEEE Trans. Ind. Electron.*, vol. 71, no. 10, pp. 13439–13449, Oct. 2024.

[15] E. Ogasawara, L. C. Martinez, D. de Oliveira, G. Zimbrão, G. L. Pappa, and M. Mattoso, "Adaptive normalization: A novel data normalization approach for non-stationary time series," in *Proc. Int. Jt Conf. Neural. Netw., Barcelona, Spain*, Barcelona, Spain, 2010, pp. 1–8.

[16] J. A. Alzubi et al., "Hashed Needham Schroeder industrial IoT based cost optimized deep secured data transmission in cloud," *Measurement*, vol. 150, 2020. Art. no. 107077.

[17] X. Wang, C. Song, C. Yang, and Y. Xie, "Process working condition recognition based on the fusion of morphological and pixel set features of froth for froth flotation," *Miner. Eng.*, vol. 128, pp. 17–26, Nov. 2018.

[18] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Deep adaptive input normalization for time series forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3760–3765, Sep. 2020.

[19] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *Proc. Int. Conf. Learn. Represent., Virtual, Online*, 2022, pp. 1–25.

[20] W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu, "Dish-ts: A general paradigm for alleviating distribution shift in time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2023, pp. 7522–7529.

[21] J. P. Muszkats, S. A. Seminara, and M. I. Troparevsky, *Applications of Wavelet Multiresolution Analysis*. Cham, Switzerland: Springer, 2020, pp. 59–64.

[22] Y. Wei and H. Wang, "Wavelet integrated attention network with multi-resolution frequency learning for mixed-type wafer defect recognition," *Eng. Appl. Artif. Intell.*, vol. 121, 2023, Art. no. 105975.

[23] Y. Wang, X. Wang, J. Zhou, C. Yang, and Y. Yang, "Long sequence multivariate time-series forecasting for industrial processes using SASGNN," *IEEE Trans. Ind. Inform.*, vol. 20, no. 10, pp. 12407–12417, Oct. 2024.

[24] G. Lai, W. C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2018, pp. 95–104.

[25] W. Cai, Y. Liang, X. Liu, J. Feng, and Y. Wu, "MSGNET: Learning multi-scale inter-series correlations for multivariate time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, 2024, pp. 11141–11149.
[26] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Proces. Syst.*, Virtual, Online, 2021, pp. 22419–22430.
[27] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Virtual, Online, 2021, pp. 11106–11115.
[28] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *Int. Conf. Learn. Represent.*, Kigali, Rwanda, 2023, pp. 1–21.

Xiaoli Wang received the Ph.D. degree in control theory and control engineering from Central South University, Changsha, China, in 2011.

She was a Visiting Scholar with the Department of Energy Institute, Texas A&M University, TX, USA, from December 2016 to December 2017. She is currently a Full Professor with the School of Automation, Central South University. Her research interests include modeling and optimal control of the complex industrial processes, machine learning and pattern recognition for application in industrial processes, etc.

Yulong Wang received the B.Eng. degree in electronic information engineering and the M.Eng. degree in mineral processing from the Taiyuan University of Technology, Taiyuan, China, in 2018 and 2022. He is currently working toward the Ph.D. degree in control science and engineering with the School of Automation, Central South University, Changsha, China.

His current research interests include data modeling and control of complex industrial processes, data analysis, and machine learning.

Fanlei Lu received the B.Eng. degree in detection guidance and control technology in 2016, and the M.Eng. degree in control science and engineering in 2020, both from the School of Automation, Central South University, Changsha, China, where he is currently working toward the Ph.D. degree in control science and engineering with the School of Automation.

His research interests include modeling and optimal control of flotation industrial processes.

Jiayi Zhou (Member, IEEE) received the B.Eng. degree in automation and the M.Eng. degree in control science and engineering from Chongqing University, Chongqing, China, in 2014 and 2017, respectively, and the Ph.D. degree in control science and engineering from the School of Automation, Central South University, Changsha, China, in 2024.

She is currently a Postdoctoral Researcher with the School of Automation, Central South University. Her research interests include modeling and optimal control of complex industrial processes and soft sensor modeling.

LiYang Qin received the B.Eng. degree in electronic information science and technology and the M.Eng. degree in electronics and communication engineering, both from the School of Physics and Electronics, Central South University, Changsha, China. He is currently working toward the Ph.D. degree in control science and engineering with the School of Automation at the same university.

His research interests include deep learning-based time series analysis methods, deep learning-based industrial process modeling techniques, and knowledge engineering.

Chunhua Yang (Fellow, IEEE) received the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2002.

She is currently a Professor with the School of Automation, Central South University. Her research interests include modeling and control of complex industrial processes and intelligent control systems.