Engineering Applications of Artificial Intelligence 133 (2024) 108361

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A task-oriented deep learning framework based on target-related transformer network for industrial quality prediction applications

Yalin Wang, Rao Dai, Diju Liu [*], Kai Wang, Xiaofeng Yuan, Chenliang Liu

School of Automation, Central South University, Changsha, 410083, China

ARTICLE INFO

Keywords:
Deep learning
Target-related transformer
Quality prediction
Industrial process

ABSTRACT

Executing various production tasks is critical to the safe operation and efficient production of industrial processes. As one of them, the detection task of key quality variables directly affects the operation optimization and decision-making of industrial processes, but it is severely limited by the harsh environment and detection instruments. Therefore, the real-time prediction task of key quality variables becomes the basis for optimal control of industrial processes. To address this issue, this paper proposes a task-oriented deep learning framework based on a target-related transformer (TR-Former) network for industrial quality prediction tasks. Specifically, a new target-related self-attention (TR-SA) mechanism is developed to guide feature learning by adding attention scores between task-related target variables and other variables. As a result, the learned features in this instance will be guaranteed to be relevant to the target variable and useful for the quality prediction task. Moreover, the long-range dynamics of industrial process data can also be captured, which can further improve the prediction performance of the model. Finally, extensive experiments were conducted on two industrial processes to validate the superiority of the proposed method in terms of quality prediction tasks. The experimental results demonstrate that the proposed TR-Former method exhibits an improvement ranging from 3% to 13% in the mean absolute error indicator compared to the traditional transformer and other state-of-the-art methods.

1. Introduction

Quality prediction tasks are of paramount importance in ensuring the safe, stable, and efficient operation of industrial processes (Fink et al., 2020; Wang et al., 2023a; Yao and Ge, 2023). Many control tasks necessitate a certain level of measurement precision for their execution (Cheng et al., 2022; Sun et al., 2023). While primary physical quantities, such as temperature, pressure, and flow, can be monitored using hardware sensors, key quality variables, such as the concentration of specific ions in a substance, are challenging to measure in real-time through sensor technology (Liu et al., 2022a; Wang et al., 2023b). Obtaining these quality variables often requires on-site sampling and laboratory analysis, leading to significant time lags and the inability to perform operations within a limited time, which affects the real-time control of industrial processes (He et al., 2023; Liu et al., 2023a; Zhou et al., 2022a). Therefore, soft sensor modeling techniques are developed to address this challenge (Jiang et al., 2021; Sun and Ge, 2021). These techniques establish mathematical models between quality variables and available auxiliary variables, enabling the estimation of quality variables that are difficult to measure in real time (Ren et al., 2022a; Shang et al., 2014).

Soft sensor modeling techniques typically fall into two categories: mechanism-based and data-driven methods (Kadlec et al., 2009). Mechanism-based models demand comprehensive knowledge of process reaction mechanisms, which are often challenging to obtain in complex industrial processes (Liu and Xie, 2020). Benefiting from the widespread application of distributed control systems and machine vision technology in industrial processes, a large amount of data is collected, making it possible for industrial big data to move from technology exploration to practical application (Zhou et al., 2022b). Consequently, data-driven soft sensing modeling techniques are gaining significant attention and are widely applied, particularly in time series analysis, fault detection, and more. This meets the growing need for data accuracy and reliability in industrial settings (Tao et al., 2023). Commonly used machine learning methods such as principal component analysis (PCA) (Dong and Qin, 2018), partial least squares (PLS) (Yang et al., 2021) and support vector machines (SVM) (Herceg et al., 2019) have achieved certain success in industrial processes. However, these are shallow algorithms,

[*] Corresponding author.

E-mail addresses: ylwang@csu.edu.cn (Y. Wang), raodai@csu.edu.cn (R. Dai), djliu@csu.edu.cn (D. Liu), kaiwang@csu.edu.cn (K. Wang), yuanxf@csu.edu.cn (X. Yuan), lcliang@csu.edu.cn (C. Liu).

https://doi.org/10.1016/j.engappai.2024.108361
Received 5 February 2023; Received in revised form 15 November 2023; Accepted 27 March 2024
Available online 6 April 2024

which are only suitable for online estimation of some highly correlated quality variables in industrial processes.

With the increasing complexity of modern industrial processes, deep learning has been extended to industrial data-driven soft sensor modeling with its powerful feature extraction capabilities. Algorithms with simple structures and easy implementation, such as stacked autoencoder (SAE), first receive attention (Sun and Ge, 2022). For example, Yuan et al. (2018) proposed a variable-weighted stacked autoencoder (VW-SAE) that assigns different weights to input variables according to their relevance to the target. Nevertheless, these methods are static, lacking the capability to effectively model dynamic industrial processes. Convolutional neural network (CNN) and long short-term memory network (LSTM), as commonly used deep learning algorithms, have also been introduced into the realm of data-driven soft sensor modeling (Ren et al., 2022b). For example, Xia et al. (2021) proposed a stacked gated recurrent unit-recurrent neural network (GRU-RNN), which can improve training efficiency and robustness. Lai et al. (2018b) proposed long-and short-term time-series network (LSTnet), and used CNN and RNN to extract local dependencies and long-term patterns between variables.

Unfortunately, these methods are limited by the model structure and receptive field size, rendering them incapable of capturing long-range dependent features commonly observed in industrial processes. The main reason for this phenomenon is that the receptive field size of CNN and the recurrent structure of LSTM severely limit the data extraction ability of long-term range dependencies (Lim and Zohren, 2021; Zerveas et al., 2021a). Therefore, researchers have explored the introduction of non-recursive global-aware transformer algorithms into industrial process modeling. For example, Li et al. (2019) proposed a LogSparse transformer (LogTrans) model for electricity prediction tasks. Zhou et al. (2020) proposed an Informer model for multiple industrial prediction tasks by further improving the huge memory consumption of attention calculations in transformers.

Although transformer-based methods have been extended to industrial data modeling, they still cannot achieve optimal performance in the face of different task requirements (Liu et al., 2022b; Wu et al., 2020). The main reason for this lies in the lack of an effective feature filtering mechanism, which results in the features extracted by the model tending to the overall characteristics of the data and ignoring the task characteristics. The lack of task feature guidance leads to features that are relevant to the task but do not fit into the overall trend (such as some features in the time series that respond to peaks or valleys) being given lower weights in feature extraction after feature extraction, and thus not being represented in the deeper features. This also leads to the fact that the extracted features contain the noise of industrial process data, missing important feature information related to the task. Therefore, achieving excellent results usually requires fine-tuning hyperparameters or extensive pre-training, significantly increasing the difficulty of algorithm engineering applications. Especially taking the quality prediction task as an example, the model training process should include quality variables closely related to the quality prediction task.

To solve the above problems, this paper proposes a task-oriented deep learning framework based on the TR-Former algorithm. In short, the proposed framework is expected to enhance the performance of the original transformer algorithm, particularly in critical quality prediction tasks in industrial processes. The main contributions of this paper are as follows.

1) A novel target-related self-attention mechanism (TR-SA) is designed to introduce task information into the modeling process, preventing task-related information from being overwhelmed during feature extraction.
2) An efficient TR-SA-based transformer architecture named TR-Former is devised for layer-by-layer extraction of task-related features to enhance data-driven modeling performance for dynamic industrial processes.
3) The proposed TR-Former method is applied to two real industrial process datasets, affirming its superior performance relative to several existing methods.

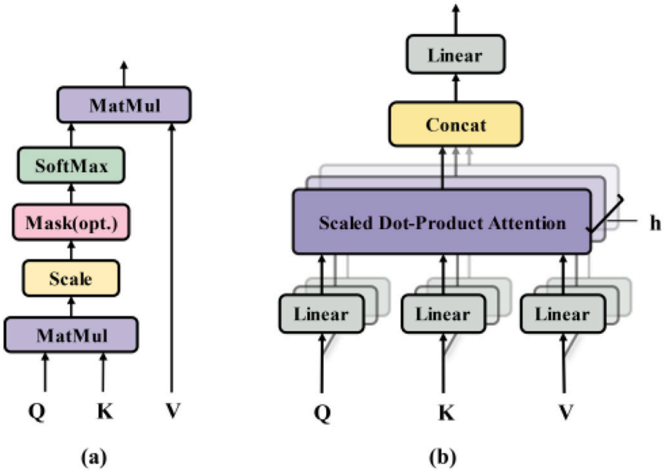


Fig. 1. Self-attention mechanism. (a) Scaled dot-product attention. (b) Multi-head attention.

The rest of this paper is arranged as follows. First, Section II provides a brief overview of the self-attention mechanism and the transformer network. Section III introduces the proposed TR-Former network and its quality prediction framework. After that, two practical industrial cases are used to demonstrate the prediction performance of the proposed TR-Former in Section IV and Section V. Finally, Section VI provides the concluding remarks and suggestions for future work.

2. Preliminaries

2.1. Self-attention mechanism

The self-attention mechanism (SA) is a special form of the traditional attention mechanism, which seeks to extract the complex correlations in the data (Shaw et al., 2018). Fig. 1 briefly depicts two attention types of SA, including scaled dot-product attention and multi-head attention.

Generally, query matrix $\mathbf{Q} \in \mathbb{R}^{N_q \times d_q}$ and key-value pairs $\{\mathbf{K} \in \mathbb{R}^{N_k \times d_k}, \mathbf{V} \in \mathbb{R}^{N_v \times d_v}\}$ are set as the inputs of SA, where N_q and $N_k = N_v$ denote a total number of samples, $d_q = d_k$ and d_v denote their corresponding dimensions.

The goal of the scaled dot-product attention shown in Fig. 1(a) is to extract the features related to the query matrix from key-value pairs. The detailed calculation formula is as follows

$$\mathbf{Z}_S = \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{SoftMax}\left(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathbf{V} \tag{1}$$

where $\mathbf{Z}_S \in \mathbb{R}^{N_q \times d_v}$ represents the extracted similar features, $\mathrm{Attention}(\cdot)$ and $\mathrm{SoftMax}(\cdot)$ denote the scaled dot-product attention process and the SoftMax normalization process, respectively.

The goal of multi-head attention shown in Fig. 1(b) is to extract richer similarity information by extending the similarity measure of a single space to multiple different subspaces. The specific calculation procedures are as follows

$$\begin{aligned} &\mathbf{Z}_{\mathrm{M}} = \mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\ &= \mathrm{Concat}(\mathbf{head}_1, \dots, \mathbf{head}_{\mathrm{h}})\mathbf{W}^O \end{aligned} \tag{2}$$

$$\mathbf{head}_{\mathrm{i}} = \mathrm{Attention}\left(\mathbf{Q}\mathbf{W}_{\mathrm{i}}^{Q}, \mathbf{K}\mathbf{W}_{\mathrm{i}}^{K}, \mathbf{V}\mathbf{W}_{\mathrm{i}}^{V}\right) \tag{3}$$

where $\mathbf{Z}_{\mathrm{M}} \in \mathbb{R}^{N_q \times d_{\mathrm{model}}}$ represents the final obtained features, d_{model} rep-

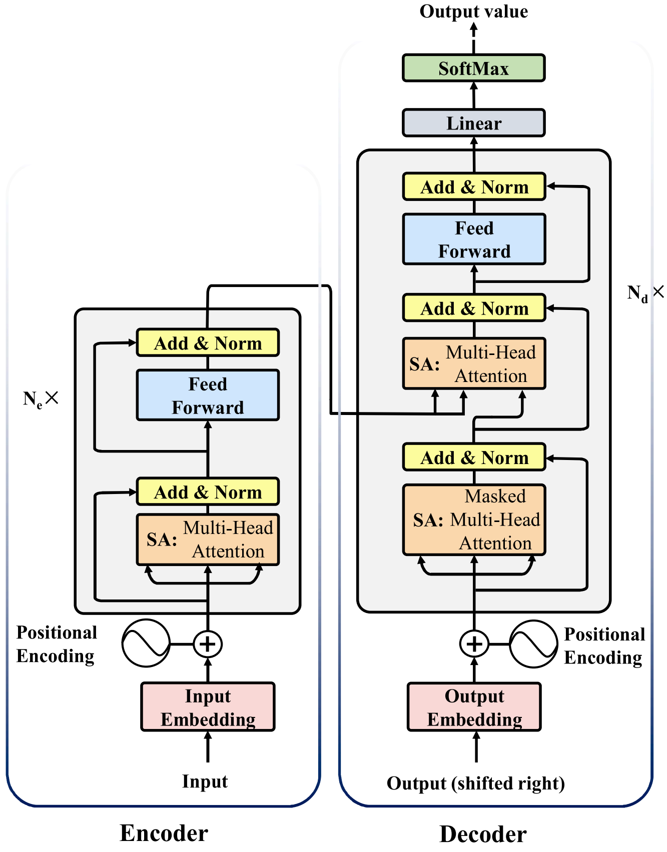


Fig. 2. The schematic diagram of transformer network.

resents the dimension. $\mathbf{head}_i \in \mathbb{R}^{N_q\times d_v}$ represents the extracted similar features in $i-$th subspace. h represents the number of subspaces. $\mathbf{W}_i{}^Q \in \mathbb{R}^{d_q\times(d_{\mathrm{model}}/h)}$, $\mathbf{W}_i{}^K \in \mathbb{R}^{d_k\times(d_{\mathrm{model}}/h)}$ and $\mathbf{W}_i{}^V \in \mathbb{R}^{d_v\times(d_{\mathrm{model}}/h)}$ are the weight matrices for mapping to $i-$th subspace. $\mathbf{W}^O \in \mathbb{R}^{d_{\mathrm{model}}\times d_{\mathrm{model}}}$ is the output weight matrix. MultiHead$(\cdot)$ and Concat$(\cdot)$ denote the multi-head attention process and the matrix stitching along the vector dimension, respectively. Since multi-head attention can obtain more diverse features, it has become one of the most used types.

2.2. Transformer network

The transformer network is a neural network with encoder-decoder architecture (Vaswani et al., 2017). Its encoder and decoder are stacked by multiple independent feature extractors. The simple schematic diagram of the transformer network is shown in Fig. 2.

Assuming that the input is $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{N_x}] \in \mathbb{R}^{N_x\times d_x}$, where N_x and d_x denote the number and dimension of samples, respectively. Then the input is first mapped to a high-dimensional space to enhance its representation ability in the input embedding module, which is expressed as

$$\mathbf{X}_{\mathrm{E}} = \text{Input Embedding}(\mathbf{X}) = \mathbf{X}\mathbf{W}_{\mathrm{E}} + \mathbf{b}_{\mathrm{E}} \tag{4}$$

where $\mathbf{X}_{\mathrm{E}} \in \mathbb{R}^{N_x\times d_{\mathrm{model}}}$ represents the obtained data, $\mathbf{W}_{\mathrm{E}} \in \mathbb{R}^{d_x\times d_{\mathrm{model}}}$ denotes the weight matrix, $\mathbf{b}_{\mathrm{E}} \in \mathbb{R}^{d_{\mathrm{mod}\,el}}$ denotes the bias parameter.

Afterwards, the $\mathbf{X}_{\mathrm{E}}$ is added positional encoding to improve the sequence position sensitivity, which is described as

$$\mathbf{X}_{\mathrm{PE}} = \begin{cases} PE(pos,2i) = \sin\left(pos/10000^{2i/d}\right) \\ PE(pos,2i+1) = \cos\left(pos/10000^{2i/d}\right) \end{cases} \tag{5}$$

$$\mathbf{X}_{\mathrm{R}} = \mathbf{X}_{\mathrm{E}} + \mathbf{X}_{\mathrm{PE}} \tag{6}$$

where $pos \in [1, N_x]$ represents the sample position, $i \in [1, d_{\mathrm{mod}\,el}/2]$ denotes its $i-$th dimension. $\mathbf{X}_{\mathrm{PE}} \in \mathbb{R}^{N_x\times d_{\mathrm{mod}\,el}}$ and $\mathbf{X}_{\mathrm{R}} \in \mathbb{R}^{N_x\times d_{\mathrm{mod}\,el}}$ represent the position information matrix and obtained data with position information, respectively.

The obtained $\mathbf{X}_{\mathrm{R}}$ is then sent into N_e stacked encoders to extract deep features. Take the first encoder as an example, $\mathbf{X}_{\mathrm{R}}$ first generates query matrix and key-value pairs. The detailed calculation procedures are as follows

$$\mathbf{Q} = \mathbf{X}_{\mathrm{E}}\mathbf{W}_{\mathrm{Q}} + \mathbf{b}_{\mathrm{Q}} \tag{7}$$

$$\mathbf{K} = \mathbf{X}_{\mathrm{E}}\mathbf{W}_{\mathrm{K}} + \mathbf{b}_{\mathrm{K}} \tag{8}$$

$$\mathbf{V} = \mathbf{X}_{\mathrm{E}}\mathbf{W}_{\mathrm{V}} + \mathbf{b}_{\mathrm{V}} \tag{9}$$

where $\mathbf{W}_{\mathrm{Q}}$, $\mathbf{W}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{V}}$ represent the weight matrices. $\mathbf{b}_{\mathrm{Q}}$, $\mathbf{b}_{\mathrm{K}}$ and $\mathbf{b}_{\mathrm{V}}$ represent the bias parameters. The complex correlation within them is extracted by SA, and the calculation process is the same as Eqs. (2) and (3).

After that, the nonlinear representation of features obtained by SA in the feed-forward layer is further strengthened. Notably, residual connection and layer normalization are added to the two modules to address the vanishing gradient issue. The calculation process is

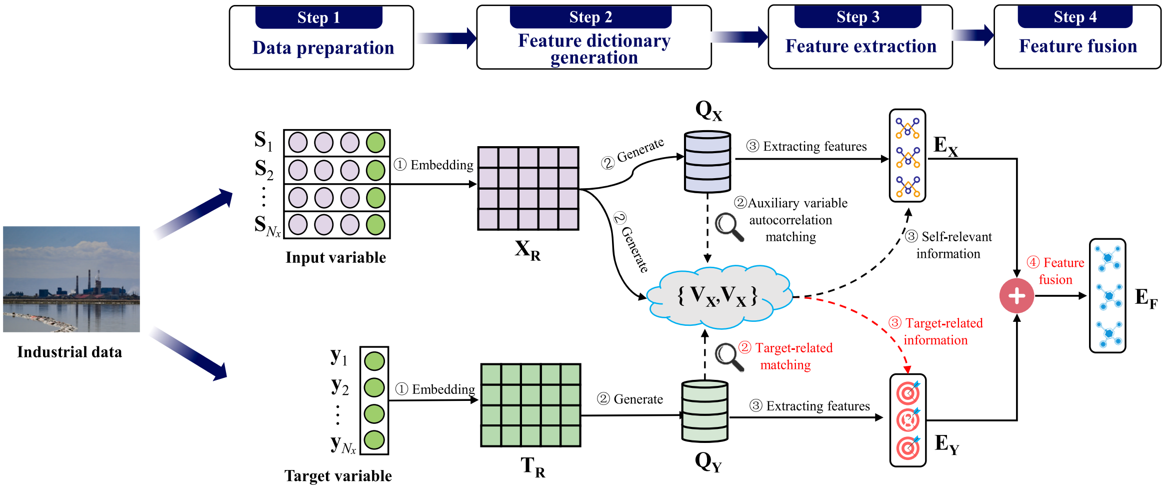


Fig. 3. The illustration of target-related self-attention mechanism.

described as

$$\mathbf{X}_{\mathrm{A}} = \mathrm{LN}(\mathbf{X}_{\mathrm{E}} + \mathbf{Z}_{\mathrm{M}}) \tag{10}$$

$$\mathrm{Feed}(\mathbf{X}_{\mathrm{A}}) = \max(0, \mathbf{X}_{\mathrm{A}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{11}$$

$$\mathbf{X}_{\mathrm{Feed}} = \mathrm{LN}(\mathbf{X}_{\mathrm{A}} + \mathrm{Feed}(\mathbf{X}_{\mathrm{A}})) \tag{12}$$

where $\mathbf{X}_{\mathrm{A}} \in \mathbb{R}^{N_x \times d_{\mathrm{mod}\ el}}$ and $\mathbf{X}_{\mathrm{Feed}} \in \mathbb{R}^{N_x \times d_{\mathrm{mod}\ el}}$ represent the output of SA and the output of final encoder. $\mathrm{LN}(\cdot)$ and $\mathrm{Feed}(\cdot)$ represent the layer normalization and the feed-forward layer, respectively. $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias parameters. $\max(\cdot)$ means take the maximum value. Afterwards, $\mathbf{X}_{\mathrm{Feed}}$ is repeatedly used as the input of the next encoder until the end of the N_e encoders.

Although the decoder process of the transformer is similar to its encoder, it is different in two details. The first is that the decoder of the transformer adds a lower triangular matrix to the first SA to mask future information. The other is that the obtained encoded output will be input into the second SA of the encoder to generate key-value pairs.

3. Proposed target-related transformer network for quality prediction applications

This section details the design of the target-related self-attention mechanism and the proposed target-related transformer network based on it. Then, the quality prediction framework based on the proposed network is introduced. The comparison methods and evaluation indicators are also provided for quality prediction applications.

3.1. Target-related self-attention mechanism

The traditional self-attention mechanism, as the core of the transformer network, usually models and learns the global feature information of the data. However, it pays more attention to the covariate feature learning of data and does not build a task-oriented deep learning network, which is not conducive to its application in real industrial processes. Especially in the task of industrial quality prediction, the target variable is crucial to the calculation of the self-attention mechanism, which can make the subsequent feature learning beneficial to enhancing the prediction performance of the model. To solve this problem, this study proposes a novel target-related self-attention (TR-SA) mechanism to comprehensively explore the correlation between covariates and quality variables to compensate for the shortcomings of the original self-attention mechanism in feature extraction. The detailed illustration of TR-SA is shown in Fig. 3. It mainly consists of four steps, namely data preparation, feature dictionary generation, feature extraction, and feature fusion.

Step 1 ***Data Preparation***: To make full use of data samples collected in industrial processes, the obtained data samples are first divided into input samples and target samples. Then, they are fed into TR-SA and mapped to obtain covariable representation $\mathbf{X}_{\mathbf{R}}$ and target variable representation $\mathbf{T}_{\mathbf{R}}$.

Step 2 ***Feature dictionary generation***: According to the calculation process of the traditional attention mechanism like Eqs. (7)–(9), covariable representation $\mathbf{X}_{\mathbf{R}}$ is first mapped into three representation subspaces to generate query matrix $\mathbf{Q}_{\mathbf{X}}$ and key-value pairs $\{\mathbf{K}_{\mathbf{X}}, \mathbf{V}_{\mathbf{X}}\}$. The autocorrelation matching of them is carried out. At the same time, target variable representation $\mathbf{T}_{\mathbf{R}}$ only generates the target query matrix $\mathbf{Q}_{\mathbf{Y}}$ and performs target-related matching with key-value pairs.

Step 3 ***Feature extraction***: Through autocorrelation matching and target-related matching, two different attention information are obtained, which are called self-relevant information and target-related information. The specific calculation formulas are given as follows

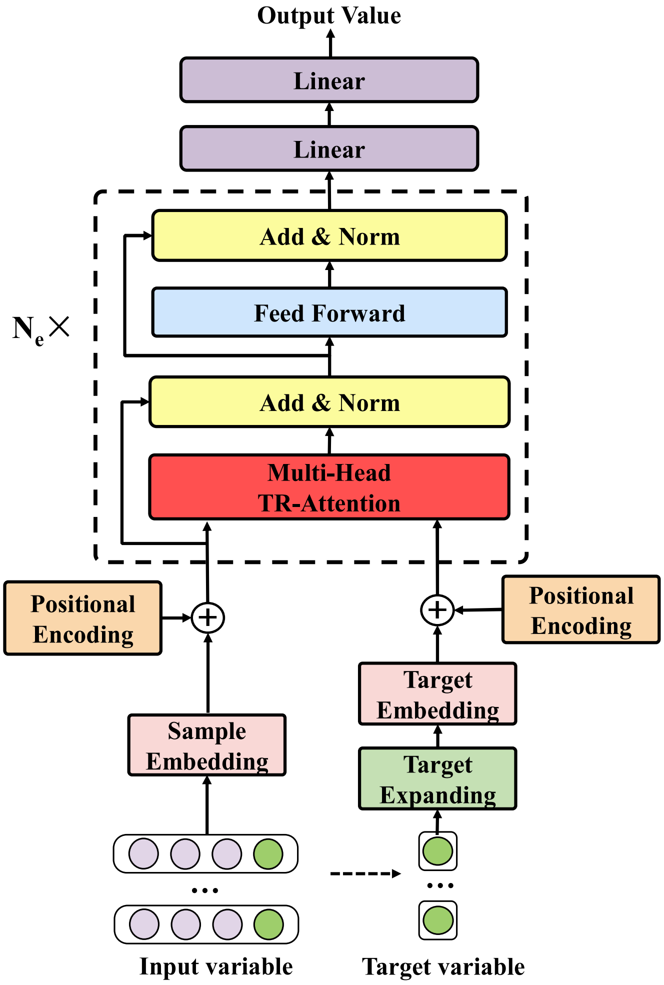


Fig. 4. The diagram of TR-Former.

$$\mathbf{E}_{\mathbf{X}} = \mathrm{MultiHead}(\mathbf{Q}_{\mathbf{X}}, \mathbf{K}_{\mathbf{X}}, \mathbf{V}_{\mathbf{X}}) \tag{13}$$

$$\mathbf{E}_{\mathbf{Y}} = \mathrm{MultiHead}(\mathbf{Q}_{\mathbf{Y}}, \mathbf{K}_{\mathbf{X}}, \mathbf{V}_{\mathbf{X}}) \tag{14}$$

where $\mathbf{E}_{\mathbf{X}}$ represents the eigenmatrix within input samples, $\mathbf{E}_{\mathbf{Y}}$ represents the eigenmatrix between input samples and target samples.

Step 4 ***Feature fusion***: The final feature information $\mathbf{E}_{\mathbf{F}}$ is obtained by adding the two eigenmatrices of self-relevant information and target-related information, which is given as

$$\mathbf{E}_{\mathbf{F}} = \mathbf{E}_{\mathbf{X}} + \mathbf{E}_{\mathbf{Y}} \tag{15}$$

3.2. Target-related transformer network

The quality prediction task in industrial processes is critical for the guidance of field workers to optimize their operations. Therefore, to improve the performance of industrial quality prediction, this study proposes a task-oriented target-related transformer (TR-Former) network. Specifically, the feature extraction capability of the network is enhanced by replacing the multi-head self-attention mechanism in the traditional transformer network with the proposed TR-SA. Unlike natural language processing tasks, in industrial quality prediction applications, the output of the encoder no longer requires feature representation. Therefore, the TR-Former proposed in this paper removes the decoder module, which can reduce the amount of calculation and eliminate the accumulation of errors caused by the decoder module. The specific structural diagram of TR-Former is shown in Fig. 4.

To perform the quality prediction task, it is assumed that the input at the current moment of the TR-Former network contains the sample variable matrix $\mathbf{X}_{\mathrm{S}} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t] \in \mathbb{R}^{t \times d_x}$ and the target variable vector $\mathbf{T} = [\mathbf{x}_{1d_x}, \mathbf{x}_{2d_x}, \ldots, \mathbf{x}_{td_x}] \in \mathbb{R}^t$. In particular, causal convolution is introduced into data embeddings to better match the time-series relationships of data samples (Li et al., 2019). Hence, the mapping matrix $\mathbf{X}_{\mathrm{E}} \in \mathbb{R}^{t \times d_{\mathrm{mod}\ el}}$ is obtained by convolving the input sample variable matrix $\mathbf{X}_{\mathrm{S}}$, which is described as follows

$$\mathbf{X}_{\mathrm{E}} = Conv(\mathbf{X}_{\mathrm{S}}) \tag{16}$$

Similarly, position coding is added to the mapping matrix to obtain the representation matrix $\mathbf{X}_{\mathrm{R}}$ according to $\mathbf{X}_{\mathrm{R}} = \mathbf{X}_{\mathrm{E}} + \mathbf{X}_{\mathrm{PE}}$. To fit the structure of the network, the target value vector $\mathbf{T}$ is expanded to the same dimension as $\mathbf{X}_{\mathrm{S}}$ before embedding, which is described as

$$\mathbf{T}_{\mathrm{S}} = Expanding(\mathbf{T}) \tag{17}$$

where $Expanding(\cdot)$ means repeating the input data t times to get $\mathbf{T}_{\mathrm{S}} \in \mathbb{R}^{t \times t}$. Based on the same embedding and positional encoding, the target representation matrix $\mathbf{T}_{\mathbf{R}} \in \mathbb{R}^{t \times d_{\mathrm{mod}\ el}}$ can also be obtained. The goal of dimension augmentation of $\mathbf{T}$ is to calculate TR-SA, so that the query matrix generated by it can match the key-value pairs of the sample matrix. After dimension expansion, $\mathbf{T}$ is mapped to the projection of different spaces by a learned weight matrix, which can enhance the expression and generalization ability of the target vector.

Finally, the attention result is calculated according to the matching pattern in TR-SA. The rest of the calculation part is the same as the encoder of the original transformer. The predicted value is obtained by adding two continuously connected fully connected layers at the end of all encoder modules. The calculation procedures are expressed as follows

$$\mathbf{Y}_{\mathrm{out}} = \mathbf{X}_{\mathrm{out}} \mathbf{W}_{\mathrm{out}} + \mathbf{b}_{\mathrm{out}} \tag{18}$$

$$\widetilde{\mathbf{y}} = \mathbf{W} \mathbf{Y}_{\mathrm{out}} + \mathbf{b} \tag{19}$$

where $\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{d_{\mathrm{mod}\ el} \times 1}$ and $\mathbf{W} \in \mathbb{R}^{1 \times t}$ represent the weight matrix of two fully connected layers. $\mathbf{b}_{\mathrm{out}}$ and $\mathbf{b}$ represent the bias parameters respectively. $\mathbf{Y}_{\mathrm{out}} \in \mathbb{R}^{t \times 1}$ represents a single vector obtained after the first transformation. $\widetilde{\mathbf{y}}$ represents the final prediction value. Then, the backpropagation algorithm is used to train the whole network by minimizing the mean square error between the true value $\mathbf{y}_i$ and the predicted value $\widetilde{\mathbf{y}}_i$. The loss function is designed as

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|\widetilde{\mathbf{y}}_i - \mathbf{y}_i\|^2 \tag{20}$$

In summary, the overall training algorithm of TR-Former is described in Algorithm 1.

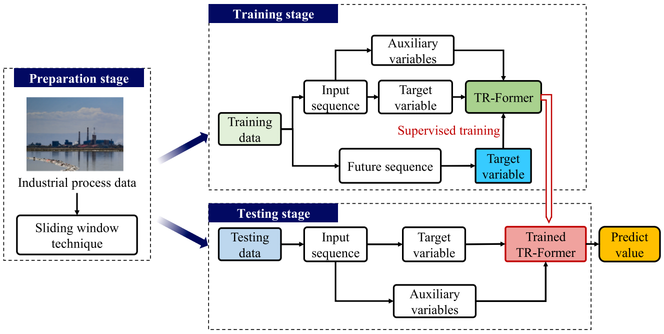
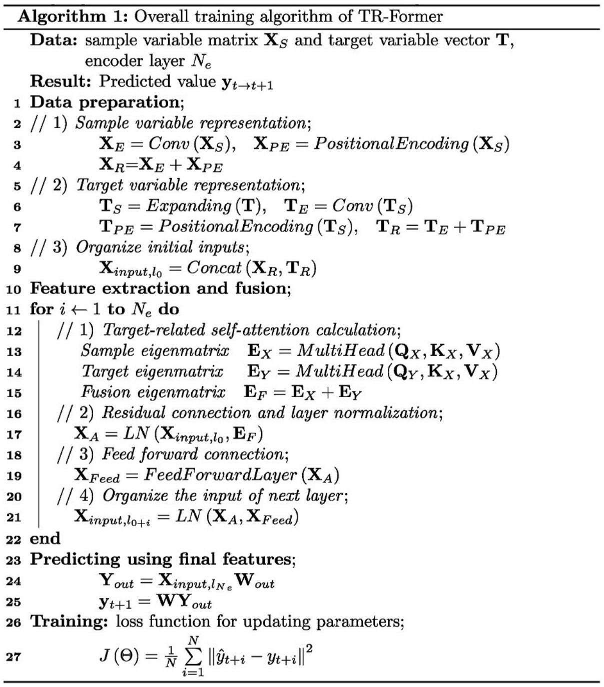


Fig. 5. The quality prediction framework based on TR-Former.

Algorithm 1. Overall training algorithm of TR-Former



```
Algorithm 1: Overall training algorithm of TR-Former
Data: sample variable matrix $\mathbf{X}_S$ and target variable vector $\mathbf{T}$,
      encoder layer $N_e$
Result: Predicted value $\mathbf{y}_{t \to t+1}$
Data preparation;
// 1) Sample variable representation;
        $\mathbf{X}_E = Conv(\mathbf{X}_S), \quad \mathbf{X}_{PE} = PositionalEncoding(\mathbf{X}_S)$
        $\mathbf{X}_R = \mathbf{X}_E + \mathbf{X}_{PE}$
// 2) Target variable representation;
        $\mathbf{T}_S = Expanding(\mathbf{T}), \quad \mathbf{T}_E = Conv(\mathbf{T}_S)$
        $\mathbf{T}_{PE} = PositionalEncoding(\mathbf{T}_S), \quad \mathbf{T}_R = \mathbf{T}_E + \mathbf{T}_{PE}$
// 3) Organize initial inputs;
        $\mathbf{X}_{input,l_0} = Concat(\mathbf{X}_R, \mathbf{T}_R)$
Feature extraction and fusion;
for $i \leftarrow 1$ to $N_e$ do
    // 1) Target-related self-attention calculation;
        Sample eigenmatrix  $\mathbf{E}_X = MultiHead(\mathbf{Q}_X, \mathbf{K}_X, \mathbf{V}_X)$
        Target eigenmatrix  $\mathbf{E}_Y = MultiHead(\mathbf{Q}_Y, \mathbf{K}_X, \mathbf{V}_X)$
        Fusion eigenmatrix  $\mathbf{E}_F = \mathbf{E}_X + \mathbf{E}_Y$
    // 2) Residual connection and layer normalization;
      $\mathbf{X}_A = LN(\mathbf{X}_{input,l_0}, \mathbf{E}_F)$
    // 3) Feed forward connection;
      $\mathbf{X}_{Feed} = FeedForwardLayer(\mathbf{X}_A)$
    // 4) Organize the input of next layer;
      $\mathbf{X}_{input,l_{0+i}} = LN(\mathbf{X}_A, \mathbf{X}_{Feed})$
end
Predicting using final features;
        $\mathbf{Y}_{out} = \mathbf{X}_{input,l_{N_e}} \mathbf{W}_{out}$
        $\mathbf{y}_{t+1} = \mathbf{W}\mathbf{Y}_{out}$
Training: loss function for updating parameters;
        $J(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|\hat{y}_{t+i} - y_{t+i}\|^2$
```

3.3. Quality prediction framework based on TR-former

The TR-Former network proposed in this paper fully considers the guiding role of the target variable in feature extraction, which can not only capture the dependence between samples but also extract the significant features between input variable samples and target variable samples. Therefore, for the quality prediction task of the actual industrial process, it will be more effective to build a quality prediction framework for TR-Former. Fig. 5 illustrates the three stages of building a quality prediction framework based on the TR-Former network, including the preparation stage, the training stage, and the testing stage. The details are described below.

3.3.1. Preparation stage

Step 1 Collect and normalize data obtained through sensor measurements and laboratory analysis from real industrial processes.

Step 2 A sliding window technique is utilized to construct the dataset, where each window contains the input sequence of data and the corresponding predicted sequence.

Step 3 According to the experimental requirements, the collected data are divided into training data sets and testing data sets.

3.3.2. Training stage

Step 1 The auxiliary and target variables of the current input time series data are used to pre-train the parameters of the network.

Step 2 The target variable for future input time series data to supervise fine-tuning of all network parameters.

Step 3 The optimal model is obtained through multiple iterative training and the network parameters are saved.

3.3.3. Testing stage

Step 1 The auxiliary variable of the input sequence in the test data set is fed into the trained TR-Former network to obtain the prediction results.

Step 2 The true label values corresponding to the input sequences in the testing data set are used to evaluate the prediction performance of the model.

3.4. Comparison methods

To verify the advantages and practicability of the proposed TR-Former network, the following five methods are also adopted to construct the quality prediction framework for comparison under the same experimental conditions.

1) LogTrans (Li et al., 2019): This model utilizes convolutional self-attention to generate queries and keys using causal convolutions in the self-attention layer.
2) Informer (Zhou et al., 2020): This model combines the self-attention distilling and the generative style decoder.
3) Long-and short-term time-series network (LSTnet) (Lai et al., 2018a): This model combines CNN and RNN to extract short-term local dependency patterns among variables.
4) Transformer-based framework for multivariate time series (MVTTrans) (Zerveas et al., 2021b): Unsupervised pre-training is introduced into multivariate time series framework for the first time.
5) Variable-wise weighted SAE (VWSAE) (Yuan et al., 2018): This model is a typical nonlinear deep learning model by introducing target-dependent variable weights in SAE.
6) Principal component regression (PCR) (Xiong and Shi, 2018): This model is a typical linear regression model through regression analysis with principal components as independent variables.
7) Stacked autoencoder (SAE) (El-allaly et al., 2020; Hinton and Salakhutdinov, 2006): The autoencoder is stacked to enable better feature extraction capabilities.

3.5. Evaluation indicators

To quantitatively evaluate the effect of the prediction model applied in industrial processes, root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and coefficient of determination (R^2) are used in this study. These evaluation indicators are all quantitative indicators recognized in the field that can accurately assess the performance of time series prediction tasks (Liu et al., 2023b; Tang et al., 2023). The detailed calculation formulas of these four evaluation indicators are as follows:

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{N} \|\tilde{\mathbf{y}}_i - \mathbf{y}_i\|^2 \Big/ N} \tag{21}$$

$$\mathrm{MAE} = \sum_{i=1}^{N} |\mathbf{y}_i - \bar{\mathbf{y}}_i| \Big/ N \tag{22}$$

$$\mathrm{MAPE} = \sum_{i=1}^{N} \left|\frac{\mathbf{y}_i - \bar{\mathbf{y}}_i}{\mathbf{y}_i}\right| \Big/ N \tag{23}$$

$$\mathrm{R}^2 = 1 - \sum_{i=1}^{N} (\mathbf{y}_i - \tilde{\mathbf{y}}_i)^2 \Big/ \sum_{i=1}^{N} (\mathbf{y}_i - \bar{\mathbf{y}})^2 \tag{24}$$

where $\bar{\mathbf{y}} = \sum_{i=1}^{N} \mathbf{y}_i / N$ represents the mean of all labeled target data. RMSE and MAE are usually used to measure the absolute size of the deviation between the real value and the predicted value. MAPE measures the relative magnitude of the deviation. The smaller the three values, the higher the accuracy of the corresponding prediction model. R^2 is mainly used to measure the fitting degree of the model. The closer its value is to 1, the better the performance of the predictive model.

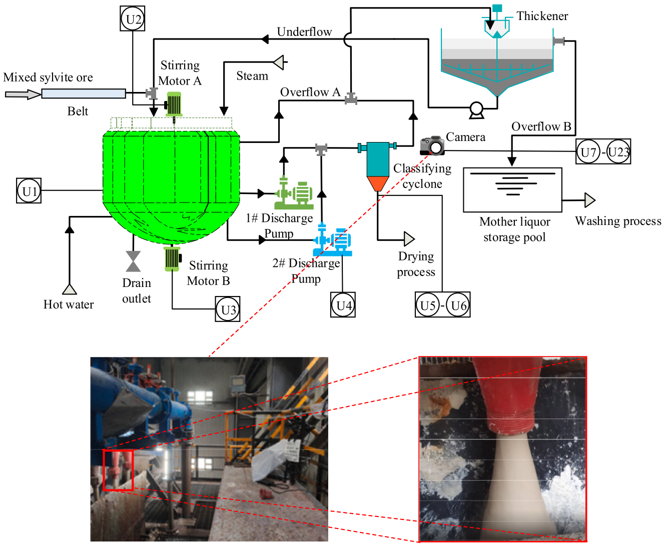


Fig. 6. The flowchart of the sylvite crystallization process.

3.6. Experimental configuration

The programming language used in this experiment is Python 3.6, with the PyTorch framework version 1.10.1. The experimental environment is equipped with an Intel(R) Core (TM) i5-10210U CPU (1.60 GHz) and an Nvidia RTX 2080Ti GPU. To ensure the fairness of the experiment, all comparative models were run in the same experimental environment and configuration.

4. Industrial application to the sylvite crystallization process

4.1. Process description of sylvite crystallization

Sylvite crystallization is an important process for chemical companies to produce potassium salt products. It utilizes the crystallization law of the pentagonal diagram to separate solid solutes like potassium sulfate crystals from liquid solutions like potassium sulfate solution to form crystals like potassium sulfate. Fig. 6 shows the flow chart of potassium sulfate crystallization in a large Salt Lake chemical company in China. First, the potassium chloride produced in the previous process is mixed and washed with kainite and then sent to the slurry distributor. In the distributor, the mixed slurry solution undergoes conversion and crystallization. To further concentrate, the crystalline slurry is subjected to centrifugation and drying in the cyclone group to obtain the final potassium salt crystal. Real-time monitoring of sylvite crystallization directly determines several important indicators such as product recovery rate and product quality.

Due to the complex mechanism of the sylvite crystallization process and the limitation of measuring instruments in severe environments, the stable and efficient production of the entire sylvite crystallization process is highly dependent on real-time monitoring of the discharge concentration of the grading cyclone, as shown in the red dotted line box in Fig. 6. However, due to the obstruction of on-site production conditions and measuring instruments, the concentration measurement of emissions currently relies on manual determination by operators. This type of operation has the disadvantages of high subjectivity, high labor intensity, and poor real-time monitoring. Therefore, it is necessary to establish a data-driven quality prediction model to predict discharge concentration in real time for the guidance of on-site workers. To ensure the accuracy of the model, an industrial camera is deployed at the outlet of the grading cyclone to monitor the shape of the discharged slurry in real time to obtain concentration-related image parameters. The detailed installation and deployment in the real industrial site can be

Table 1
Auxiliary variables in the sylvite crystallization process.

Tag	Variable description	Tag	Variable description
U1	Liquid level	U7	Energy
U2	Stirring current A	U8	Contrast
U3	Stirring current B	U9	Variance
U4	Pump current	U10	Entropy
U5	Pressure of outlet	U11	Area
U6	Temperature of outlet	U12 – U23	Fourier descriptors

Table 2
Hyperparameter combination of TR-Former network based on prediction of grading cyclone discharge concentration.

Parameter	Description	Value	Range
sql	Input sample length	2	From 2 to 10
$d_{\text{mod } el}$	Embedding dimension	512	{256, 512, 1024}
h	Multi-head number	3	From 2 to 8
N_e	Encoder layer number	3	From 2 to 6
d_q	Query vector dimension	32	{24, 32, 64, 128}
d_{ff}	Nonlinear dimension	1024	{512, 1024, 2048}
k	Convolution kernel size	7	{3, 5, 7}

Table 3
Comparison results of all comparative methods for predicting discharge concentration on the sylvite crystallization.

Method	RMSE	MAE	MAPE	R^2
TR-Former	**0.4615**	**0.3552**	**1.0282**	**0.5356**
LogTrans	0.5309	0.3833	1.0360	0.3855
Informer	0.5256	0.3928	1.2094	0.3977
LSTnet	0.5566	0.4072	1.2375	0.3245
MVTTrans	0.5413	0.4281	1.6113	0.2802
VWSAE	0.5768	0.4690	1.2783	0.2746
PCR	0.5815	0.4588	1.5064	0.1893
SAE	1.0242	0.776	4.1275	0.0004

seen in Fig. 6. Based on prior experience in artificial knowledge and image recognition, 6 process variables and 17 image features are used as auxiliary variables to construct the prediction model framework. U1–U6 represent 6 process parameters closely related to the crystallizer. After preprocessing the image data obtained by the industrial camera, the texture and shape characteristics of the export slurry image are extracted. U7–U10 are the four texture feature variables, U11 is the relative area of the outlet, and U12–U23 are Fourier descriptions of the first 12 image features. The detailed description of these variables is listed in Table 1.

4.2. Dataset and hyperparameter setting

The sylvite crystallization process dataset used in this study was obtained from an actual industrial process. Process variable data is obtained from the sensors at a sampling frequency of once every 10 min. Image data is obtained from on-site industrial cameras on-site at a sampling frequency of 30 min. A total of 175 labeled samples were collected from industrial sites over a period of about two months. To make full use of the data obtained, process variables U1–U4 of the first two moments of each labeled sample are added as a supplement to the labeled samples. In this case, the input variables of the constructed quality prediction model are increased to 31.

Then, all the obtained data are normalized for confidentiality measures. The first 125 data samples were selected as training data and the last 50 data samples as test data. For each piece of data, a separate sample is constructed using the sliding window technique, and the width of the window is the input sequence length plus the predicted sequence length, that is, *sql* plus 1. Finally, the trial-and-error technique is utilized to select the optimal hyperparameter combination for all models under many experiments. The learning rate is set as 0.0002. The number of model training iterations is set as 200. Other optimal hyperparameter combinations are listed in Table 2.

4.3. Experimental results and discussion

The specific comparative experimental results used to predict discharge concentration on the sylvite crystallization dataset are listed in Table 3. It can be easily seen from the results in Table 3 that the prediction performance of SAE and PCR is poor, especially with larger values for performance indices such as RMSE, MAE, and MAPE, and a smaller R^2. This is mainly because both are static and linear models, which cannot capture the complex nonlinear characteristics of industrial process data. Although VW-SAE is also a static method, its performance is much better than that of SAE and PCR due to the incorporation of variable-related guidance, while the task-oriented modeling approach employed in this paper is to further refine the variable guidance and associate it with the modeling task. The improved performance of VWSAE also demonstrates that the modeling ideas of the paper are reasonable and effective in the industrial process and have the potential to be extended to other methods. Among the dynamic methods, this paper compares LSTnet, MVTTrans, Informer, LogTrans, and so on. Among them, LSTnet is a recursive dynamic model based on RNN structure, and it has general performance. This is because the recursive structure generates a cumulative error problem during feature extraction, leading to its inability to extract long-range historical features. In contrast, MVTTrans, Informer, LogTrans, which are based on transformer parallel structure, can avoid the above problems with maintaining the dynamics, but their results are still unsatisfactory. This is because they lack the guidance of task-related information in the feature extraction process, resulting in the extracted features tending to reflect the overall average characteristics rather than benefiting the modeling task. For this reason, the TR-Former proposed in this paper addresses this pain point by introducing task goals into the modeling process, thus guiding the model to be more inclined to extract features that are beneficial to the task. Thus it achieves optimal performance.

In order to intuitively compare the prediction performance on the sylvite crystallization dataset, the comparison curves between the prediction data of all comparative methods and the true data are shown in Fig. 7. It can be seen that PCR, SAE, VWSAE, etc. static methods are difficult to track the rapidly changing curves, and have large errors in most of the peaks or valleys, indicating that they are unable to sense the pattern of change in the curves. In contrast, dynamic methods such as LSTnet, MVTTrans, Informer, LogTrans, etc. have better performance, which can extract the trend information of the curves, but it is still difficult to achieve high accuracy at the peaks or valleys of large changes. This also indicates that the features extracted by these methods are more averaged and overall structured rather than integrated with the modeling objectives. In comparison, the proposed TR-Former in this paper improves the modeling accuracy of the model at drastic changes by introducing targets, which makes its overall performance much better than other methods.

Further, to demonstrate the effectiveness of the proposed method, the boxplot of the absolute errors of eight methods for predicting discharge concentration on the testing dataset is shown in Fig. 8. The upper edge of the blue box represents the upper quartile, and the lower edge represents the lower quartile. The red line in the middle represents the median of the entire dataset. It can be seen from Fig. 8 that the absolute error distribution of TR-Former on the testing dataset is more concentrated and closer to zero, which indirectly indicates that its prediction accuracy is better and more stable. In addition, the prediction errors of the proposed TR-Former are also more concentrated in a small range of intervals, which explains the better overall modeling of the proposed method.

To verify the practical performance of the model, we conducted an online trial run during actual production processes over a week.

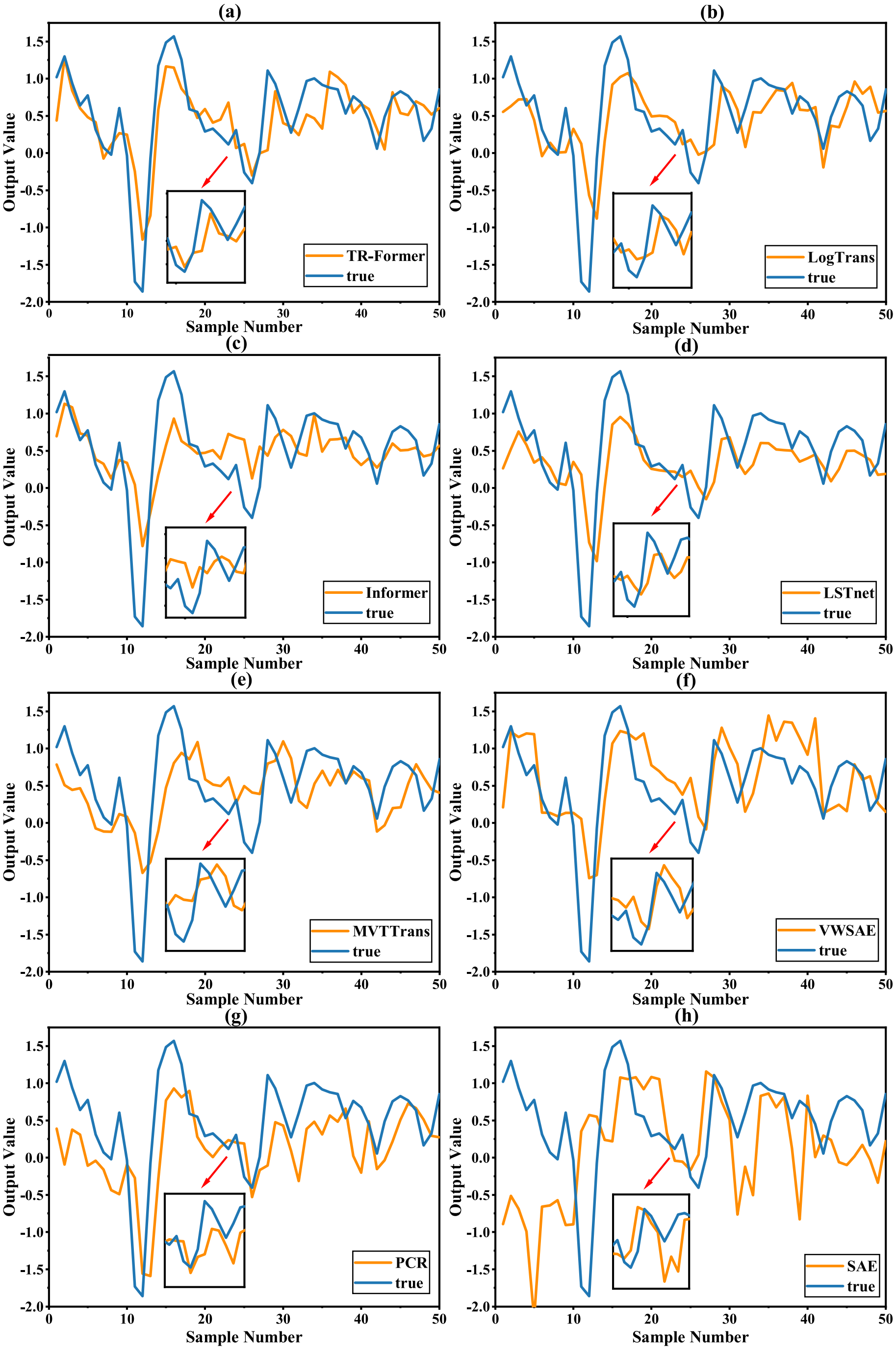


Fig. 7. Detailed prediction curves for predicting discharge concentration on the sylvite crystallization dataset. (a) TR-Former. (b) LogTrans. (c) Informer. (d) LSTnet. (e) MVTTrans. (f) VWSAE. (g) PCR. (h) SAE.

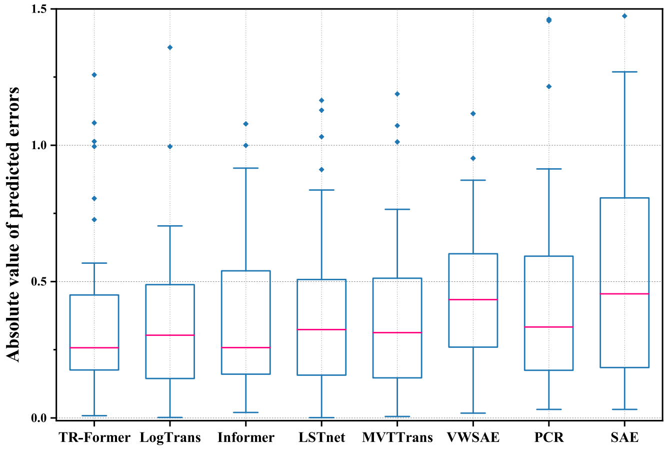


Fig. 8. Boxplot of the absolute errors of eight methods for predicting discharge concentration on the sylvite crystallization dataset.

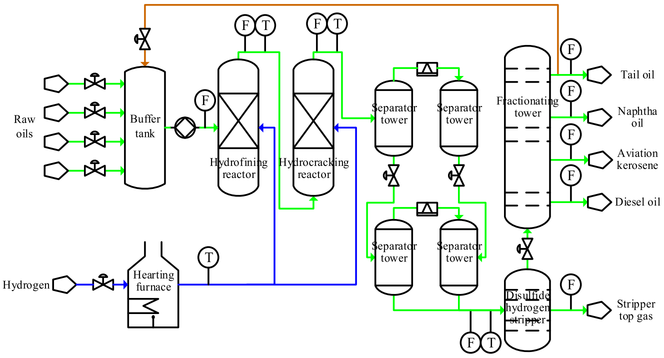


Fig. 9. The flowchart of the hydrocracking process.

Throughout this trial, the model generated concentration predictions every half hour, which were utilized to guide on-site operations. Despite the outstanding performance of the TR-Former on the sylvite crystallization dataset, achieving extremely precise concentration predictions in the real production environment remained challenging. This is understandable, as field data typically exhibit higher levels of noise and variability compared to the data used for model training. Nonetheless, this does not undermine the ability of the TR-Former to provide valuable reference points to production personnel, assisting them in operational decision-making. This substantiates the practical advantages of the smooth implementation of real industrial processes, fostering the prospect of extending the application of the TR-Former to additional hydrocyclone clusters in the crystallization workshop.

5. Industrial application to the hydrocracking process

5.1. Process description of hydrocracking

Hydrocracking is the main process for producing petroleum in refining and petrochemicals. It is a petroleum refining process that converts heavy oil into light oil through a catalytic cracking reaction under the action of heating, high hydrogen pressure, and a catalyst. Fig. 9 shows a simple hydrocracking flow chart of a chemical company in China. Naphtha, one of the products of hydrocracking, is a light oil mainly used as a chemical raw material. Compared with catalytic cracking, hydrocracking significantly improves product yield and quality, but consumes a lot of hydrogen and requires a lot of investment. In order to evaluate whether the current production meets the requirements of high efficiency and low consumption, the C5 content in light naphtha is usually used as a monitoring index. However, due to the limitations of measuring instruments, C5 content can only be obtained by laboratory off-line detection. This has a great lag and cannot guide the production in time. Therefore, it is of great practical significance to establish a quality prediction model to estimate the C5 content in the production process. According to the artificial experience and mechanism knowledge, 43 process variables are selected as auxiliary variables to establish an effective quality prediction model. The specific variable description is shown in reference (Liu et al., 2023b).

Table 4
Hyperparameter combination of TR-Former network based on prediction of C5 Content.

Parameter	Description	Value	Range
sql	Input sample length	6	From 2 to 10
$d_{mod\ el}$	Embedding dimension	256	{256, 512, 1024}
h	Multi-head number	2	From 2 to 8
N_e	Encoder layer number	2	From 2 to 6
d_q	Query vector dimension	128	{24, 32, 64, 128}
d_{ff}	Nonlinear dimension	512	{512, 1024, 2048}
k	Convolution kernel size	3	{3, 5, 7}

Table 5
Comparison results of all comparative methods for predicting C5 content on the hydrocracking process.

Method	RMSE	MAE	MAPE	R^2
TR-Former	**0.2746**	**0.1799**	**0.3739**	**0.9358**
LogTrans	0.2852	0.1989	0.3817	0.9301
Informer	0.3057	0.2075	0.3814	0.9204
LSTnet	0.3140	0.1927	0.3953	0.9160
VWSAE	0.3371	0.2159	0.4543	0.9032
MVTTrans	0.3421	0.2181	0.4300	0.9003
PCR	0.3809	0.2223	0.3868	0.8802
SAE	0.4967	0.3115	0.5728	0.7788

5.2. Dataset and hyperparameter setting

The hydrocracking process dataset used in this study was obtained from a petrochemical plant in China. A total of 2600 labeled samples were collected from industrial sites at a sampling frequency of once every 12 h over a period of about three years.

Then, all the obtained data are normalized for confidentiality measures. The first 2300 data samples are selected as training data, and the last 300 data samples are selected as test data. Similarly, the sliding window technique is used to build independent samples for the model, and the trial-and-error technique is used to select the optimal combination of hyperparameters for all models in multiple experiments. The optimal hyperparameter combination of TR-Former is listed in Table 4.

5.3. Experimental results and discussion

Table 5 gives the indicators of the eight methods for predicting the C5 content in the hydrocracking process. Likewise, since SAE, PCR, and VWSAE are static models, it is difficult to capture the dynamic characteristics of the data, which leads to their poor prediction performance. The prediction effect of MVTTrans is improved compared with the shallow models. Although LSTnet can capture both long-term and short-term patterns of time series, it cannot achieve satisfactory prediction results due to the vanishing gradient due to too long series. Furthermore, because of the sparsity mechanism, the prediction performance of Informer and LogTrans can be improved a little, but they still do not achieve the best prediction performance. The proposed TR-Former considers both the autocorrelation dynamic characteristics of the process data and the role of the target variable to guide the prediction task, which makes it achieve the best prediction performance.

Intuitively, Fig. 10 depicts the prediction curves of all models and

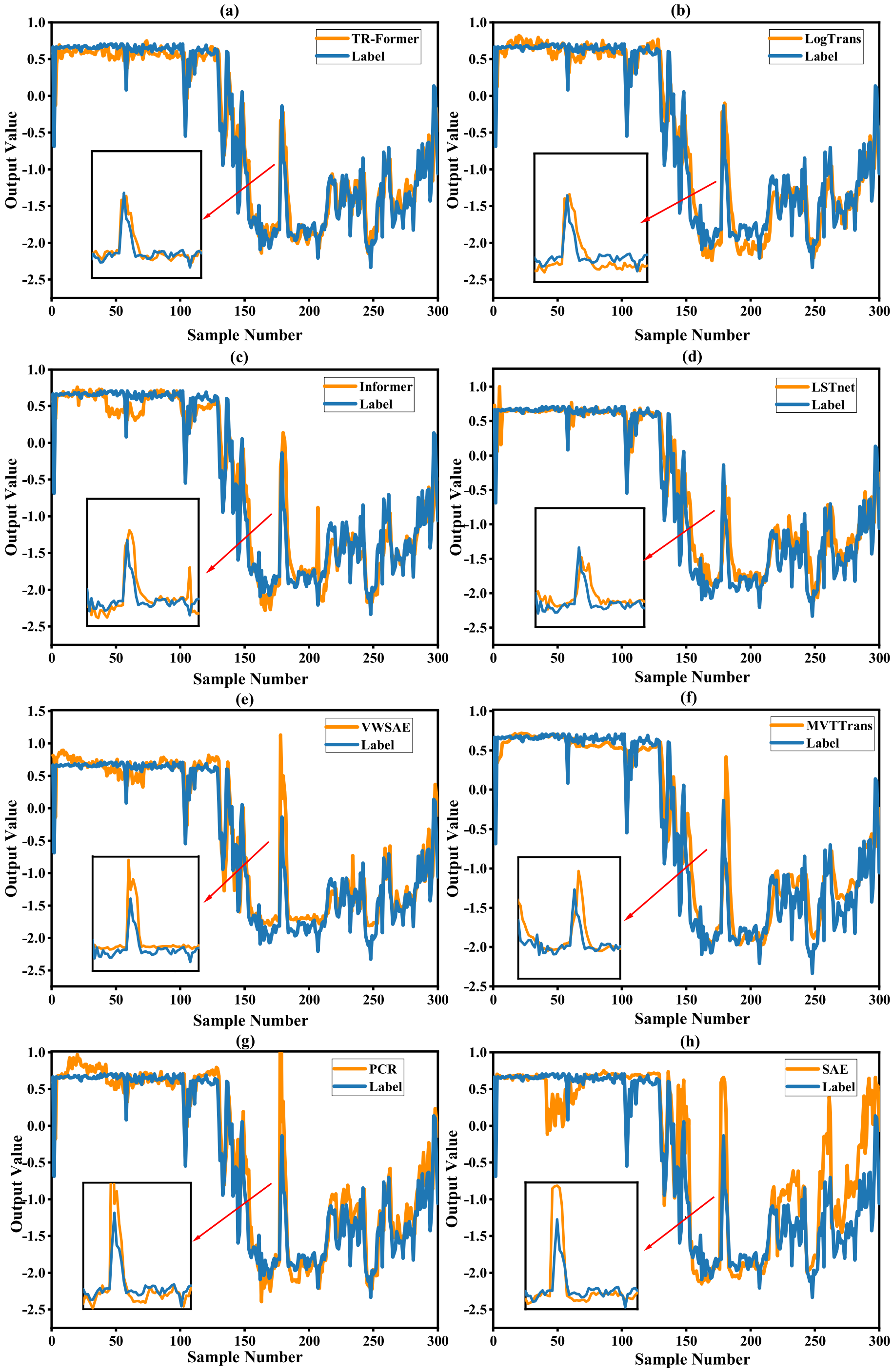


Fig. 10. Detailed prediction curves for prediction C5 content on the hydrocracking dataset. (a) TR-Former. (b) LogTrans. (c) Informer. (d) LSTnet. (e) VWSAE. (f) MVTTrans. (g) PCR. (h) SAE.

true curves on the testing dataset. It can be seen that the TR-Former can achieve better prediction results than other methods both under the wave curve and in the sharp change of the peak area. In addition, Fig. 11 shows the boxplot of the absolute errors of all comparative methods for predicting C5 content on the testing dataset. It can be seen that the prediction error distribution of TR-Former is more concentrated and its median line is the lowest, which also proves that the proposed method has the best prediction effect in predicting C5 content. It is worth noting that the C5 dataset spans a broad time range, essentially covering the entire production cycle. As a result, all models deliver satisfactory results. However, TR-Former distinguishes itself by consistently demonstrating higher predictive accuracy and fitting performance across different intervals.

In summary, based on the extensive experiments and detailed analyses presented above, it can be concluded that the TR-Former method proposed in this paper is better suited for industrial quality prediction tasks when compared to existing advanced time series algorithms and basic methods. Its consistency and enhanced performance on two industrial datasets underscore its robustness and applicability.

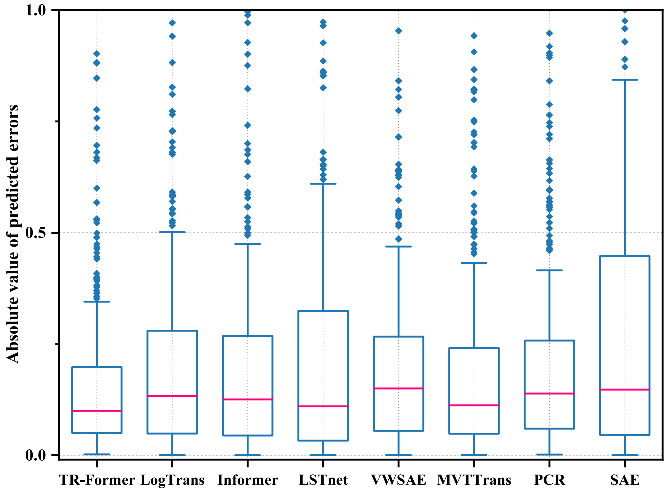


Fig. 11. Boxplot of the absolute errors of eight methods for predicting C5 content on the hydrocracking dataset.

6. Conclusion

This paper focuses on developing a task-oriented deep learning framework for the important quality prediction task in industrial processes. Based on the traditional transformer network, a novel quality prediction framework based on the target-related transformer (TR-Former) network is constructed from the perspective of quality prediction tasks by combining the advantages of long-range dynamic features and attention correlation. Extensive experiments are constructed on two industrial datasets to verify the predictive performance of the proposed method. It can be seen from the four prediction evaluation indicators that the proposed method has improved compared to other typical methods.

In the future, we will actively seek collaborations to address predictive challenges arising from uncertain factors in various aspects of industrial processes. These challenges encompass the fusion of multimodal data, including process data, audio, and images, as well as refining models to accommodate data with varying sampling frequencies. Furthermore, we will explore the application of task-specific deep learning frameworks to tackle diverse industrial production tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 92267205, 62103063, 62373378 and 61988101 and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2021RC4054.

References

Cheng, P., Wang, H., Stojanovic, V., Liu, F., He, S., Shi, K., 2022. Dissipativity-based finite-time asynchronous output feedback control for wind turbine system via a hidden Markov model. Int. J. Syst. Sci. 53, 3177–3189.

Dong, Y., Qin, S.J., 2018. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. J. Process Control 67, 1–11.

El-allaly, E.-d., Sarrouti, M., En-Nahnahi, N., Alaoui, S.O.E., 2020. A LSTM-based method with attention mechanism for adverse drug reaction sentences detection. In: Advanced Intelligent Systems for Sustainable Development (AI2SD'2019) Volume 2-Advanced Intelligent Systems for Sustainable Development Applied to Agriculture and Health, pp. 17–26.

Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.-J., Ducoffe, M., 2020. Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng. Appl. Artif. Intell. 92, 103678.

He, Y.-L., Li, X.-Y., Ma, J.-H., Zhu, Q.-X., Lu, S., 2023. Attribute-relevant distributed variational autoencoder integrated with LSTM for dynamic industrial soft sensing. Eng. Appl. Artif. Intell. 119, 105737.

Herceg, S., Ujević Andrijić, Ž., Bolf, N., 2019. Development of soft sensors for isomerization process based on support vector machine regression and dynamic polynomial models. Chem. Eng. Res. Des. 149, 95–103.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

Jiang, Y., Yin, S., Dong, J., Kaynak, O., 2021. A review on soft sensors for monitoring, control, and optimization of industrial processes. IEEE Sensor. J. 21, 12868–12881.

Kadlec, P., Gabrys, B., Strandt, S., 2009. Data-driven soft sensors in the process industry. Comput. Chem. Eng. 33, 795–814.

Lai, G., Chang, W.-C., Yang, Y., Liu, H., 2018a. Modeling long- and short-term temporal patterns with deep neural networks. In: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 95–104.

Lai, G., Chang, W.-C., Yang, Y., Liu, H., 2018b. Modeling long-and short-term temporal patterns with deep neural networks. In: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 95–104.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., Yan, X., 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Neural Information Processing Systems (NeurIPS), pp. 5243–5253.

Lim, B., Zohren, S., 2021. Time-series forecasting with deep learning: a survey. Phil. Trans. Math. Phys. Eng. Sci. 379, 20200209.

Liu, C., Wang, K., Wang, Y., Yuan, X., 2022a. Learning deep multi-manifold structure feature representation for quality prediction with an industrial application. IEEE Trans. Ind. Inf. 18, 5849–5858.

Liu, D., Wang, Y., Liu, C., Wang, K., Yuan, X., Yang, C., 2023a. Blackout missing data recovery in industrial time series based on masked-former hierarchical imputation framework. IEEE Trans. Autom. Sci. Eng. 1–13.

Liu, D., Wang, Y., Liu, C., Yuan, X., Yang, C., 2023b. Multirate-former: an efficient transformer-based hierarchical network for multi-step prediction of multirate industrial processes. IEEE Trans. Instrum. Meas. 1, 1.

Liu, D., Wang, Y., Liu, C., Yuan, X., Yang, C., Gui, W., 2022b. Data mode related interpretable transformer network for predictive modeling and key sample analysis in industrial processes. IEEE Trans. Ind. Inf. 1–12.

Liu, Y., Xie, M., 2020. Rebooting data-driven soft-sensors in process industries: a review of kernel methods. J. Process Control 89, 58–73.

Ren, L., Liu, Y., Huang, D., Huang, K., Yang, C., 2022a. MCTAN: a novel multichannel temporal attention-based network for industrial health indicator prediction. IEEE Transact. Neural Networks Learn. Syst. 1–12.

Ren, L., Wang, T., Laili, Y., Zhang, L., 2022b. A data-driven self-supervised LSTM-DeepFM model for industrial soft sensor. IEEE Trans. Ind. Inf. 18, 5859–5869.

Shang, C., Yang, F., Huang, D., Lyu, W., 2014. Data-driven soft sensor development based on deep learning technique. J. Process Control 24, 223–233.

Shaw, P., Uszkoreit, J., Vaswani, A., 2018. Self-attention with relative position representations. arXiv preprint arXiv:.02155.

Sun, P., Song, X., Song, S., Stojanovic, V., 2023. Composite adaptive finite-time fuzzy control for switched nonlinear systems with preassigned performance. Int. J. Adapt. Control Signal Process. 37, 771–789.

Sun, Q., Ge, Z., 2021. A survey on deep learning for data-driven soft sensors. IEEE Trans. Ind. Inf. 17, 5853–5866.

Sun, Q., Ge, Z., 2022. Gated stacked target-related autoencoder: a novel deep feature extraction and layerwise ensemble method for industrial soft sensor application. IEEE Trans. Cybern. 52, 3457–3468.

Tang, Y., Wang, Y., Liu, C., Yuan, X., Wang, K., Yang, C., 2023. Semi-supervised LSTM with historical feature fusion attention for temporal sequence dynamic modeling in industrial processes. Eng. Appl. Artif. Intell. 117, 105547.

Tao, H., Qiu, J., Chen, Y., Stojanovic, V., Cheng, L., 2023. Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. J. Franklin Inst. 360, 1454–1477.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, Y., Li, S., Liu, C., Wang, K., Yuan, X., Yang, C., Gui, W., 2023a. Multiscale feature fusion and semi-supervised temporal-spatial learning for performance monitoring in the flotation industrial process. IEEE Trans. Cybern. 1–14.

Wang, Y., Liu, C., Wu, H., Sui, Q., Yang, C., Gui, W., 2023b. Revolutionizing flotation process working condition identification based on froth audio. IEEE Trans. Instrum. Meas. 72, 9513012.

Wu, N., Green, B., Ben, X., O'Banion, S., 2020. Deep transformer models for time series forecasting: the influenza prevalence case. arXiv e-prints, arXiv:2001.08317.

Xia, M., Shao, H., Ma, X., Silva, C.W.d., 2021. A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. IEEE Trans. Ind. Inf. 17, 7050–7059.

Xiong, W., Shi, X., 2018. Soft sensor modeling with a selective updating strategy for Gaussian process regression based on probabilistic principle component analysis. J. Franklin Inst. 355, 5336–5349.

Yang, X., Liu, X., Xu, C., 2021. Robust mixture probabilistic partial least squares model for soft sensing with multivariate laplace distribution. IEEE Trans. Instrum. Meas. 70, 1–9.

Yao, L., Ge, Z., 2023. Causal variable selection for industrial process quality prediction via attention-based GRU network. Eng. Appl. Artif. Intell. 118, 105658.

Yuan, X., Huang, B., Wang, Y., Yang, C., Gui, W., 2018. Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. IEEE Trans. Ind. Inf. 14, 3235–3243.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C., 2021a. A transformer-based framework for multivariate time series representation learning. In: 27th ACM SIGKD Conference on Knowledge Discovery and Data Mining, pp. 2114–2124.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C., 2021b. A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2114–2124.

Zhou, C., Tao, H., Chen, Y., Stojanovic, V., Paszke, W., 2022a. Robust point-to-point iterative learning control for constrained systems: a minimum energy approach. Int. J. Robust Nonlinear Control 32, 10139–10161.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2020. Informer: beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Association for the Advancement of Artificial Intelligence (AAAI).

Zhou, J., Wang, X., Yang, C., Xiong, W., 2022b. A novel soft sensor modeling approach based on difference-LSTM for complex industrial process. IEEE Trans. Ind. Inf. 18, 2955–2964.