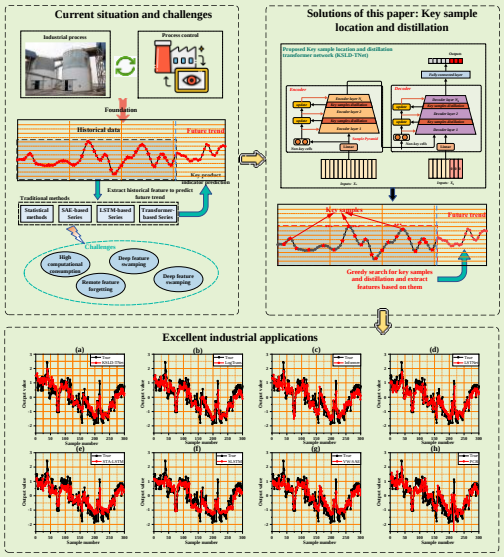
KSLD-TNet: Key Sample Location and Distillation Transformer Network for Multi-Step Ahead Prediction in Industrial Processes

Diju Liu, Yalin Wang, *Senior Member, IEEE*, Chenliang Liu, *Graduate Student Member, IEEE*, Xiaofeng Yuan, *Member, IEEE*, Kai Wang, *Member, IEEE*

***Abstract*—The multi-step ahead prediction of crucial quality indicators is the cornerstone for optimizing and controlling industrial processes. The accurate multi-step ahead prediction over long prediction horizons holds great potential for improving production performance in industrial processes. However, extracting historical features presents a significant obstacle in achieving this objective. Recent advancements have demonstrated that transformer networks offer a promising technical solution to this challenge. Nevertheless, the lack of a sample simplification mechanism makes deep feature extraction difficult. It requires a lot of computational costs, which makes the traditional transformer network less applicable in industrial processes. To explore strategies to overcome these obstacles and enhance the suitability of transformer networks for effective multi-step ahead prediction, this paper proposes a novel key sample location and distillation transformer network (KSLD-TNet). Specifically, it first locates key samples with strong interactions using the attention score matrix. Then, non-key samples are filtered out layer by layer in the KSLD-TNet encoder-decoder structure. In this way, the number of input samples for each layer can be lowered exponentially, reducing the difficulty and calculation amount of deep feature extraction significantly. It is worth noting that this paper also designs an information storage structure to avoid information loss during the sample distillation process. Two industrial process datasets are utilized to construct extensive experiments to demonstrate the effectiveness of the proposed method.**



***Index Terms*—Industrial process; deap learning; key sample location and distillation transformer; multi-step ahead prediction.**

I. Introduction

INTELLIGENT and efficient industrial process control relies heavily on real-time measurement of key product indicators, which in turn facilitate guide feedback adjustments [1-3]. Regrettably, most of these indicators, such as ion content and substance concentration, necessitate acquisition through laboratory analysis [4, 5]. This will result in a huge time delay, thereby engendering a considerable reduction in the efficacy of industrial process control. In this case, the soft sensor modeling technique, which utilizes easily measured variables such as temperature and pressure to construct predictive models of key product indicators, becomes an acceptable alternative [6, 7]. Due to the increasing complexity of the industrial process mechanism, the mechanism-driven modeling technology is difficult to adapt to the current complex industrial process. On the contrary, data-driven modeling methods have developed rapidly in industrial applications [8-10]. Commonly used methods include principal component regression (PCR) [11], partial least squares regression (PLSR) [12] and support vector machines (SVM) [13]. However, these shallow methods are difficult to reflect the nonlinearity of the process. Therefore, deep learning has been applied to data-driven modeling due to its powerful nonlinear feature extraction capability.

Various deep learning methods and their variants, such as deep belief network (DBN) [14], stacked autoencoder (SAE) [15] and long and short-term memory (LSTM) [16], have emerged in this research area. For example, Yuan et al. [17]

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 92267205 and Grant 61988101, in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022JJ20079, Grant 2021RC4054, Grant 2021JJ10065, and Grant 2020RC3003, in part by the Central South University Innovation-Driven Research Program, Changsha in China under grant 2023CXQD054, and in part by the CAAI-Huawei MindSpore Open Fund under grant CAAIXSJLJJ-2022-001A. *(Corresponding author: Chenliang Liu.)*

Diju Liu, Yalin Wang, Chenliang Liu, Xiaofeng Yuan, and Kai Wang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: djliu@csu.edu.cn, ylwang@csu.edu.cn, lcliang@csu.edu.cn, yuanxf@csu.edu.cn, kaiwang@csu.edu.cn).

designed a variable weighted stacked autoencoder (VW-SAE) algorithm to extract output related features and applied it in the industrial debutanizer process. Shi et al. [18] proposed the state transition-long short-term memory network (ST-LSTM) to solve the problem of multiple models and double sampling cycles in industrial modeling. In addition, some more advanced methods, such as transfer learning and capsule networks, have been introduced to industrial process modeling. For example, Li et al. [19] summarized the theory, applications, and challenges of deep transfer learning in industrial processes, which provide guidance for related research. Huang et al.[20] proposed a deep adversarial capsule network (DACN) to embed multidomain generalization into the intelligent compound fault diagnosis, which improves task performance. Further, Li et al. [21] proposed an interpretable WavCapsNet based on capsule network theory to improve the transparency and interpretability of the composite fault diagnosis processes.

With the advancement of carbon peaking and carbon neutralization strategies, key product indicator predicting industrial processes faces new challenges, which involve higher accuracy and long-range multi-step ahead prediction [22, 23]. Lai et al. [24] tried to design an LSTM-based method called Long- and -short time-series network (LSTNet) to solve the problem of mixing long-term and short-term dependency patterns in time series multi-step ahead prediction. However, its performance on industrial datasets is not ideal. Fortunately, the attention-based transformer network can improve the performance of multi-step ahead prediction of industrial processes to a certain extent [25]. It innovatively uses the attention mechanism to connect each sample directly, which can minimize the information transmission path and maximize information preservation. Hence, the transformer-based models have achieved numerous achievements in long-range multi-step ahead prediction [26]. For example, Zhou et al. [27] proposed an Informer network for long-range multi-step ahead prediction of industrial electric power dataset by introducing a self-attention mechanism pyramid. Li et al. [28] designed the LogTrans network to further improve the performance of the transformer network in the prediction problem by changing the single-point matching method in the attention calculation to local matching.

However, there are still plenty of issues in practical industrial applications of transformer-based methods. The most prominent point is the lack of a sample simplification mechanism in the calculation process. In traditional transformer-based methods, long-term historical samples are repeatedly extracted features in multi-layer networks without any difference, which leads to serious problems of difficulty in deep feature extraction and increased computation [29]. It is well-known that fluctuations in raw materials or changes in operating conditions over periods of time mostly cause large variations in industrial processes. This means that the existence of key samples in the historical samples, which contain the main information that causes future output to change over time. That is to say, fully extracting key sample features is enough to grasp the output change pattern. However, the traditional transformer is fed all historical samples in each layer and then violently retrieved from them to find valuable key features. It can avoid the loss of information in the shallow layer. But in the deep layer, a small amount of key information is buried beneath a large amount of useless information, making extraction of key deep features extremely difficult. Furthermore, the computational complexity of the attention mechanism is proportional to the square of the number of samples. Repeating the input of all samples will result in enormous computational consumption, making it difficult to deploy in industrial sites with limited computing power.

To address the aforementioned problems, this paper proposes an innovative key sample location and distillation transformer network (KSLD-TNet) for key product indicator prediction. Our proposed method mainly focuses on the location and feature learning of key samples to improve the performance of multi-step ahead prediction. The effectiveness of the proposed method is verified on two real industrial process datasets. Specifically, the main contributions of this paper are as follows:

1) An attention score matrix-based key sample location (KSL) strategy is designed to locate key samples with strong interactions in the long-range historical sample set.

2) To gradually extract and enhance key information from historical samples, a layer-by-layer key sample self-distillation strategy is proposed.

3) The recursive information storage structure is conceived to reduce the information loss in the distillation process of key samples.

4) An improved prediction framework based on KSLD-TNet is proposed to improve the prediction accuracy by reducing the difficulty of feature extraction and the computational effort.

II. TRANSFORMER

The transformer is a type of encoder-decoder neural network with a self-attention mechanism (SA) as its core, which is shown in Fig. 1. It mainly includes modules such as residual connection, layer normalization, and feedforward network.

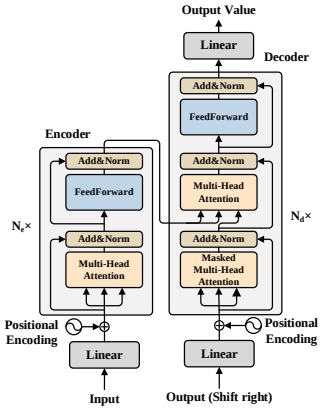


Fig. 1. The schematic diagram of a transformer.

Firstly, the input $\mathbf{X}_e \in \mathbb{R}^{n \times d_x}$ is projected into the high-dimensional space, where n is the number of samples. In addition, the positional encoding is introduced to enhance representability, which is described as follows

$$\mathbf{X}_P = \mathbf{X}_e \mathbf{W}_e^T + PE\left(\mathbf{X}_e \mathbf{W}_e^T\right) \tag{1}$$

where $\mathbf{X}_P \in \mathbb{R}^{n\times d_{\text{model}}}$ and $\mathbf{W}_e \in \mathbb{R}^{d_{\text{model}}\times d_x}$ denote the input of the encoder and the mapping parameter matrix, respectively. d_x and d_{model} denote the input and mapping data dimensions, respectively. $PE(\bullet)$ represents the calculation method of the positional encoding. Since the purpose of $PE(\bullet)$ is to generate a coding matrix for marking the sample positions, its inputs are sample position and variable position elements. These two elements are denoted as $pos \in [0, n]$ and $i \in [0, d_{\text{model}}]$. Hence, the definition of $PE(\bullet)$ is described as follows:

$$PE\left(\mathbf{X}_e \mathbf{W}_e^T\right) = \begin{cases} \sin\left(\dfrac{pos}{10000^{i/d_{\text{model}}}}\right) & i \text{ is even} \\ \cos\left(\dfrac{pos}{10000^{i-1/d_{\text{model}}}}\right) & \text{otherwise} \end{cases} \tag{2}$$

Then, $\mathbf{X}_P$ is fed into the encoder stacked by N_e identical blocks. In each block, the input is fed to a multi-head attention module and layer normalized, which is expressed as

$$\mathbf{X}_M = LN\left(MHA\left(\mathbf{X}_P\mathbf{W}_Q^T, \mathbf{X}_P\mathbf{W}_K^T, \mathbf{X}_P\mathbf{W}_V^T\right) + \mathbf{X}_P\right)$$
$$\text{where} \begin{cases} \mathbf{Q} = \mathbf{X}_P\mathbf{W}_Q^T \\ \mathbf{K} = \mathbf{X}_P\mathbf{W}_K^T \\ \mathbf{V} = \mathbf{X}_P\mathbf{W}_V^T \end{cases} \tag{3}$$

$$MHA(\mathbf{Q},\mathbf{K},\mathbf{V}) = concat\left(head_1, \cdots, head_h\right)\mathbf{W}_o \tag{4}$$

$$head_i = \text{softmax}\left(\left(\mathbf{Q}\mathbf{W}_{q,i}\right)\left(\mathbf{K}\mathbf{W}_{k,i}\right)^T \Big/ \sqrt{d_k}\right)\mathbf{V} \tag{5}$$

where $\mathbf{X}_M \in \mathbb{R}^{n\times d_{\text{model}}}$ denotes the obtained matrix, $LN(\bullet)$ denotes the layer normalization, $\mathbf{W}_Q \in \mathbb{R}^{d_q\times d_{\text{model}}}$, $\mathbf{W}_K \in \mathbb{R}^{d_k\times d_{\text{model}}}$ and $\mathbf{W}_V \in \mathbb{R}^{d_v\times d_{\text{model}}}$ denote the parameter matrices. $\mathbf{Q}$ and $\{\mathbf{K},\mathbf{V}\}$ denote the queries and key-value pairs.

Next, $\mathbf{X}_M$ is fed to a feedforward network module to extract nonlinear features, which is expressed as

$$\mathbf{X}_F = LN\left(\mathbf{X}_M + \left(\max\left(0, \mathbf{X}_M\mathbf{W}_{F,1}^T\right)\right)\mathbf{W}_{F,2}^T\right) \tag{6}$$

where $\mathbf{X}_F \in \mathbb{R}^{n\times d_{\text{model}}}$ denotes the obtained data, $\mathbf{W}_{F,1} \in \mathbb{R}^{d_{ff}\times d_{\text{model}}}$ and $\mathbf{W}_{F,2} \in \mathbb{R}^{d_{\text{model}}\times d_{ff}}$ denote the parameter matrices, d_{ff} denotes the hidden dimension. By now, a single block in the encoder has been finished. The final deep feature information $\mathbf{X}_O \in \mathbb{R}^{n\times d_{\text{model}}}$ is obtained until all blocks are finished.

Like the encoder, the decoder also consists of N_d identical modules. The difference is that each block of the decoder has two multi-head attention, one for extracting similarities between decoder input and the other for extracting useful information from encoder input.

III. Proposed Transformer Network

This section first describes the designed key sample location strategy based on multi-head attention. Then, the proposed key sample location and distillation transformer network and its prediction framework are described in detail.

A. Key sample location strategy based on multi-head attention

The term "key sample" refers to a representative sample that can best characterize the industrial processes over a period. Key sample location (KSL) means identifying key samples in a dataset, which can provide a basis for effectively reducing the difficulty and computation of feature extraction. However, the biggest challenge of KSL lies in measuring the inter-sample similarities, but none of the existing metrics have sufficient representation capability. To address this problem, this paper proposes a novel key sample location strategy based on multi-head attention, as shown in Fig. 2.

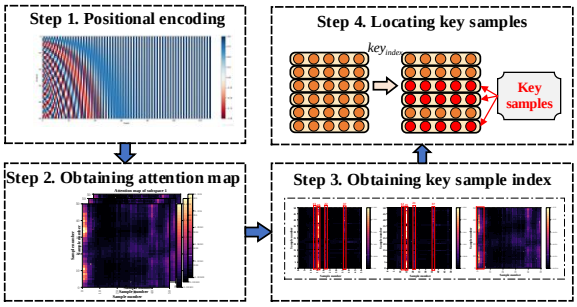


Fig. 2. The illustration of the key sample location strategy.

The overall structure of KSL-MHA can be divided into 4 steps: positional coding, obtaining attention map, obtaining key samples index, and locating key samples. The first two steps are similar to the functions of MHA in Section II, as shown in Eq (4) and Eq (5). They are used to position-encode samples and then obtain attention maps. The specific expression is as follows:

$$\begin{aligned} \mathbf{X}_{attention} &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \\ &= \text{softmax}\left(\frac{\left(\mathbf{X}_p\mathbf{W}_Q^T\right)\left(\mathbf{X}_p\mathbf{W}_K^T\right)^T}{\sqrt{d_k}}\right) \end{aligned} \tag{7}$$

$$\mathbf{X}_A = \mathbf{X}_{attention}\bullet\mathbf{V} = \mathbf{X}_{attention}\bullet\left(\mathbf{X}_p\mathbf{W}_V^T\right) \tag{8}$$

where $\mathbf{X}_{attention} \in \mathbb{R}^{n\times n}$ denotes the attention map matrix obtained using the attention mechanism. It is then used to multiply the value matrix $\mathbf{V} = \mathbf{X}_p\mathbf{W}_V^T$ to obtain the aggregated historical information features $\mathbf{X}_A$ in the subspace. It is worth noting that the attention graph can be represented by a heat map, where each column represents the similarity of a key to all queries.

In step 3, the average similarity between each key of $\mathbf{K}$ and all queries $\mathbf{Q}$ is first calculated as

$$Avg_{sim} = mean_{col}\left(\mathbf{X}_{attention}\right) \tag{9}$$

where $Avg_{sim} \in \mathbb{R}^{1\times n}$ denotes the average of all attention scores, $mean_{col}(\bullet)$ means calculating the average of the columns of a matrix. Subsequently, the samples with high average similarity between keys and queries are defined as key samples, which are

determined by the following formulas

$$Sort_{sim} = sort_{dec}\left(Avg_{sim}\right) \tag{10}$$

$$key_{index} = \arg\left(Sort_{sim}, \alpha\right) \tag{11}$$

where $sort_{dec}(\bullet)$ represents the operation of sorting in descending order. $Sort_{sim} \in \mathbb{R}^{1\times n}$ represents the sequence of average similarity obtained after sorting. $\alpha \in [0,1]$ denotes the ratio of the key sample size to the total dataset. key_{index} denotes the index of key samples. Notably, the selection of key samples is performed in a descending order of significance based on their attention similarity, which effectively avoids missing or absent instances. Since MHA has multiple parallel subspaces, the final key samples are the union of the key samples of these subspaces, which is expressed as

$$Key_{index} = key_{index_1} \cup \cdots key_{index_i} \cdots \cup key_{index_h} \tag{12}$$

where Key_{index} and key_{index_i} denote the final key sample index and the key sample index of the ith subspace, respectively. Finally, the key samples $\mathbf{X}_{key}$ in the dataset can be obtained as

$$\mathbf{X}_{key} = \mathbf{X}_A\left[Key_{index}\right] \tag{13}$$

The success of KSL-MHA is centered on the query-key-value sample similarity matching approach, which is analogous to the book search process, as shown in Fig. 3. One sample is described by three vectors, namely query (like query demands), key (like book tags), and value (like book content). The effect of query and key is to carry information between samples and extract useful information from the corresponding samples if they match closely. This implies that if a key highly matches all the query requirements, then the information contained in the corresponding sample is useful for all queries, that is, this sample is a key sample. It is analogous to the "Control Science" book in Fig. 3 that satisfies the query demands of "Process Identification", "Soft Sensor" and "Pattern Recognition", that is, these books are the key books required for this task.

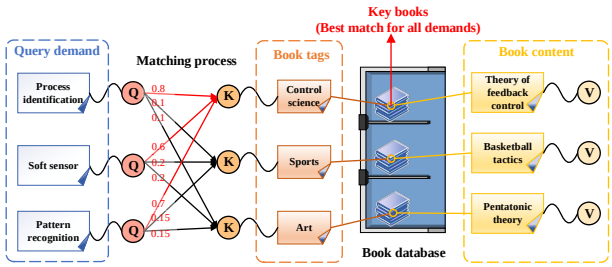
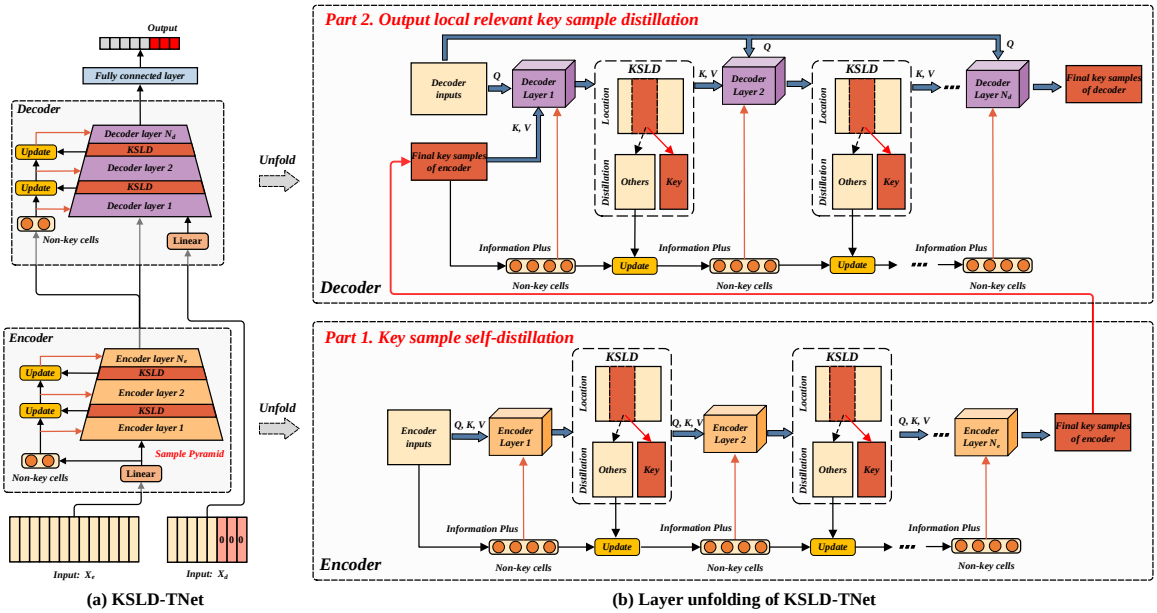


Fig. 3. The illustration of the analogy between KSL and book search process.

B. Key sample location and distillation transformer network

To reduce the complexity of deep feature extraction and computation, this paper further designs a key sample location and distillation transformer network (KSLD-TNet) with KSL-MHA as the core, as shown in Fig. 4(a). It is a typical pyramidal deep network, an encoder-decoder structure that attempts to strip historical key samples from the dataset and enhance them layer by layer. Because this process is similar to the distillation purification process, it is known as key sample distillation. Fig. 4(b) shows the hierarchical unfolding of KSLD-TNet in detail. It can be seen that KSLD-TNet consists of two parts: key sample self-distillation and output local relevant key sample distillation. Their detailed descriptions are given as follows.



(a) KSLD-TNet (b) Layer unfolding of KSLD-TNet

Fig. 4. The illustration of KSLD-TNet. (a) KSLD-TNet. (b) Layer unfolding of KSLD-TNet.

1) Key sample self-distillation: The objective of key sample self-distillation is to identify the representative key samples from the long-range historical samples. Suppose the encoder input data denotes as $\mathbf{X}_e \in \mathbb{R}^{n_e \times d_x}$, which contains a large number of historical samples ($n_e \geq 20$). Thus, $\mathbf{X}_e$ brings a lot of redundant information. The encoder part first performs the localization and self-distillation of key samples through the similarity calculation of the historical samples

themselves.

The input $\mathbf{X}_e$ passes through the first encoder layer formed by KSL-MHA, which is described as

$$\mathbf{X}_{l_1}, Key_{index,l_1} = EncoderLayer_1\left(Concat\left(\mathbf{X}_e, \mathbf{C}_{e,l_0}\right)\right) \quad (14)$$

$$\mathbf{C}_{e,l_0} = mean\left(\mathbf{X}_e\right) \quad (15)$$

where $\mathbf{C}_{e,l_i} \in \mathbb{R}^{1\times d_x}$ denotes the non-key aggregated sample that encompasses the non-key sample information in ith layer to reduce information loss. Its initial value $\mathbf{C}_{e,l_0}$ is the mean value of all historical samples. $EncoderLayer_1(\bullet)$ denotes the layer calculation process of the first encoder, which is described in Eqs. (8~14). $\mathbf{X}_{l_1} \in \mathbb{R}^{n_e\times d_x}$ and Key_{index,l_1} denote the obtained data after KSL-MHA and the index of key samples, respectively. By now, the key samples $\mathbf{X}_{key,l_1}$ and non-key samples $\mathbf{X}_{non-key,l_1}$ filtered in the first layer is obtained as

$$\mathbf{X}_{key,l_1} = \mathbf{X}_{l_1}\left[Key_{index,l_1}\right] \quad (16)$$

$$\mathbf{X}_{non-key,l_1} = \mathbf{X}_{l_1}\left[Key^{C}_{index,l_1}\right] \quad (17)$$

where Key^{C}_{index,l_1} denotes the complimentary set of Key_{index,l_1}. After that, $\mathbf{C}_{e,l_0}$ is updated by $\mathbf{X}_{non-key,l_1}$ as

$$\mathbf{C}_{e,l_1} = \left(1-\beta_{update}\right)\mathbf{X}_{non-key,l_1} + \beta_{update}\mathbf{C}_{e,l_0} \quad (18)$$

$$\beta_{update} = f\left(mean\left(\mathbf{X}_{non-key,l_1}\right)\right) \quad (19)$$

where $\beta_{update} \in [0,1]$ represents how many non-key aggregated samples are updated in the previous layer, $f(\bullet)$ is the real-valued mapping. Then, the filtered key samples and non-key aggregated samples from the first layer are concatenated and used as input $\mathbf{X}_{input,l_2}$ to the second layer for further enhancement of the key samples, which is described as

$$\mathbf{X}_{input,l_2} = Concat\left(\mathbf{X}_{key,l_1}, \mathbf{C}_{e,l_1}\right) \quad (20)$$

Afterwards, each layer recursively selects the key samples in the same way until all N_e encoders are finished. At last, the final extracted key samples $\mathbf{X}_{key,l_{N_e}}$ are obtained.

2) Output local relevant key sample distillation: The obtained key samples after self-distillation are the most representative of the long-range historical samples, but they are not certainly correlated with the output. Therefore, $\mathbf{X}_{key,l_{N_e}}$ is further distilled in the decoder of KSLD-TNet to ensure that the extracted key samples are also closely related to the output.

Specifically, the nearest samples $\mathbf{X}_{(t-n_d+1)\to t} \in \mathbb{R}^{n_d\times d_x}$ corresponding to the output $\mathbf{y}_{t\to t+p} \in \mathbb{R}^{p\times 1}$ in the time dimension and the auxiliary variables $\mathbf{X}_{t\to t+p}$ of the output jointly form the selection benchmark, which is expressed as follows

$$\mathbf{X}_d = Concat\left(\mathbf{X}_{(t-n_d+1)\to t}, \mathbf{X}_{t\to t+p}\right) \quad (21)$$

Generally, the values of the auxiliary variables at future moments are also unknown, so they are prefilled with zeros. After that, $\mathbf{X}_d$ is used as the query in KSL-MHA, while $\mathbf{X}_{key,l_{N_e}}$ is used as the key-value pair to further distill the key samples. The distillation process is similar to the self-distillation process except that the query is replaced. Finally, the output layer is added for prediction after obtaining the final key sample $\mathbf{X}_{key,l_{N_d}}$, which is described as

$$\hat{\mathbf{y}}_{t\to t+p} = \mathbf{X}_{key,l_{N_d}}\mathbf{W}_O^T \quad (22)$$

where $\mathbf{W}_O$ denotes the parameter matrix.

So far, the forward propagation of KSLD-TNet has finished. The parameters of the whole network are then updated with the backward error between the obtained predicted output and the actual output through the gradient descent algorithm. The loss function is expressed as

$$J(\Theta) = \frac{1}{2p}\sum_{i=1}^{p}\left\|\hat{y}_{t+i} - y_{t+i}\right\|^2 \quad (23)$$

where Θ denotes the learnable parameter set of KSLD-TNet.

C. Prediction framework based on KSLD-TNet

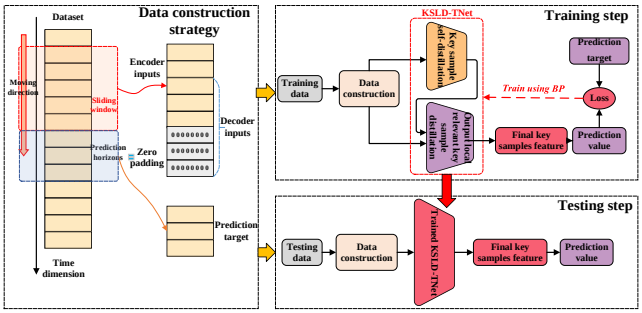


Fig. 5. The flowchart of the KSLD-TNet-based prediction framework.

To provide a more intuitive and lucid comprehension of the prediction framework based on the proposed KSLD-TNet method, Fig. 5 illustrates a simple flowchart outlining the complete method implementation. The dashed box on the left-hand side of Fig. 5 shows the strategy used for constructing input and output datasets. First, a sliding window of a specific length is used to traverse the time dimension within the acquired two-dimensional historical dataset with a uniform step size. Then, this windowed dataset is partitioned into training and test sets. Within each window sample, the initial data samples are utilized to construct the input for each time step in the framework of KSLD-TNet. Notably, within the decoder inputs of KSLD-TNet, the inclusion of segments comprising zero-valued vectors is imperative. These segments, mirroring the length of the multi-step ahead prediction horizons, serve as prediction placeholders. Their presence enables the model to discern the duration of the prediction horizon and the temporal sequencing of each sample concerning historical data points.

Following this, KLSD-TNet utilizes the proposed key sample localization technique to extract basic features from historical window samples. This process facilitates deriving

the true value of the zero-valued placeholder vectors, thereby enabling the model to perform multi-step ahead predictions. In the training phase, the pivotal product indicators corresponding to the final time point within each windowed sample serve as the labels for supervised training in the multi-step ahead prediction. Conversely, during the testing phase, these product indicators can be used to serve as authentic references for the assessment of the predictive precision of the model.

To quantitatively evaluate the prediction performance of the models, the root mean square error (RMSE) and the mean absolute error (MAE) are chosen, which are defined as

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{p}\left(\hat{y}_i - y_i\right)^2 \Big/ p} \tag{24}$$

$$\mathrm{MAE} = \sum_{i=1}^{p}\left|\hat{y}_i - y_i\right| \Big/ p \tag{25}$$

Generally, the better the performance of the model, the smaller their values will be.

IV. Industrial Applications

In this section, the proposed KSLD-TNet method is validated on two real industrial processes. The start-of-the-art methods, including LogTrans [28], Informer [27], LSTnet [24], STA-LSTM [30], SLSTM [31], VW-SAE [17] and PCR [32], are used for performance comparison. These seven methods cover most types of time series modeling methods such as traditional statistical learning methods (PCR), static deep networks (VW-SAE), dynamic deep networks (SLSTM, STA-LSTM, LSTnet), and recent transformer-based methods (Informer, LogTrans). The comparison with these methods not only can more effectively and comprehensively clarify the effectiveness and superiority of the proposed methods of this paper but also can discover the advantages and disadvantages of the existing types of methods from the results. The detailed description of these methods can be found in the introduction section of this paper.

All simulation experiments are performed on Python 3.7 with torch 1.8 and NVIDIA GeForce RTX 3060 Laptop GPU. Then, the description of two industrial processes used in this paper and the corresponding analysis of experimental results are further given.

A. Mixed potassium washing process

The mixed potassium washing process is one of the important technologies in the potassium salt production process of the Salt Lake Chemical Industry. Its purpose is to fully mix the crude potassium and soft potassium produced in the previous process and utilize the E2 mother liquor to wash out as many impurity ions as possible, such as sodium and magnesium. Fig. 6 illustrates its brief schematic diagram, which mainly includes the washing and filtration parts. The washing part consists of 4 washing tanks connected in series, in which crude potassium and soft potassium are fully mixed, and most impurity ions are dissolved into the E2 mother liquor. In the filtration part, the belt filter further filters out the impurity, and the washed mixed potassium is dried. Finally, the filtered mixed potassium is transported by belt to the next process. In the operation control of the whole washed mixed

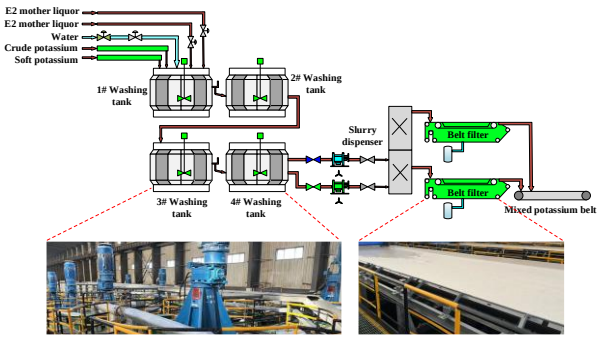


Fig. 6. The schematic diagram of the mixed potassium washing process.

Table I Description of auxiliary variables in the mixed potassium washing process

Location	Input	Variable description	Time lag (minute)
Feed section	1	1#E2 mother liquor pump flow	40-50
	2	3#E2 mother liquor pump flow	
	3	Freshwater flow	
	4	Crude potassium uptake	
	5	Soft potassium uptake	
	6	Potassium content of crude potassium filter cake	
	7	Sulfate content of crude potassium filter cake	
	8	Potassium content of soft potassium filter cake	
	9	Sulfate content of soft potassium filter cake	
Sink section	10	1# washing tank agitation current	25-40
	11	2# washing tank agitation current	
	12	3# washing tank agitation current	
	13	4# washing tank agitation current	
	14	5# washing tank agitation current	
	15	6# washing tank agitation current	
	16	7# washing tank agitation current	
	17	8# washing tank agitation current	
	18	8# washing tank level	
	19	4# washing tank level	
Slurry section	20	1# slurry distributor pressure	15-20
	21	2# slurry distributor pressure	
	22	3# slurry distributor pressure	
	23	4# slurry distributor pressure	
Outfeed section	24	1# mixed potassium slurry pump current	5-10
	25	2# mixed potassium slurry pump current	
	26	3# mixed potassium slurry pump current	
	27	4# mixed potassium slurry pump current	
	28	5# mixed potassium slurry pump current	
	29	6# mixed potassium slurry pump current	

potassium process, the sulfate ion content is an important evaluation parameter. It can directly perceive the performance of this production process. However, traditional hardware sensors cannot measure sulfate ion content in realtime, which is very unfavorable to field operators. Therefore, based on the process mechanism and expert experience, 29 easily measurable variables were selected as input to build the soft sensor model to provide the real-time prediction of sulfate ion content. Details of these auxiliary variables and their corresponding time lags are given in Table I. It is worth noting that the time lag problem of these variables has been preprocessed before being used to coordinate the predicted sulfate ion content.

In this study, a total of 4043 samples with a sampling interval of 1 hour were collected in an actual chemical enterprise in China during the period from January 2022 to June 2022. To verify the performance of the model, the first 3600 samples are taken as the training dataset and the last 443 samples are taken as the testing dataset. Meanwhile, the standardization method is used to eliminate the effect of data magnitude, and the trial-and-error method is used to find the best combination of hyperparameters. The hyperparameter combinations for the optimal performance of the proposed method under multiple experiments are shown in Table II. For a fair comparison with other methods, they are also used to conduct many experiments and give the optimal experimental results.

The detailed experimental results of the eight methods are given in Table III, which includes the results for four prediction window lengths. It is easy to observe that PCR and VW-SAE perform the worst. This is mainly because these two static methods lack the ability to capture the dynamic change patterns of the time series, which makes its performance multi-step ahead prediction difficult to guarantee. Although

Table II The optimal hyperparameters of KSLD-TNet for predicting the sulfate ion content in the mixed potassium washing process

Symbol	Description	Predict window length			
		1	2	4	8
N_e	Encoder layers	2	4	3	2
N_d	Decoder layers	5	2	1	4
α	Key sample ratio	0.75	0.6	0.65	0.7
d_{model}	Mapping dimension	1024	512	512	512
h	Subspace number	4	2	3	2
d_{ff}	Nonlinear dimension	2048	2048	2048	2048
d_q	Query dimension	32	64	64	32
d_v	Value dimension	32	64	64	32

Table III Comparison results of eight methods for predicting the sulfate ion content in the mixed potassium washing process

Method	Metrics	Predict window length			
		1	2	4	8
KSLD-TNet	RMSE	**0.2911**	**0.4508**	**0.6666**	**0.7961**
	MAE	**0.2244**	**0.3459**	**0.5134**	**0.6150**
LogTrans	RMSE	0.3207	0.4613	0.6989	0.8066
	MAE	0.2531	0.3578	0.5291	0.6373
Informer	RMSE	0.3156	0.4847	0.7099	0.8237
	MAE	0.2505	0.3836	0.5717	0.6446
LSTNet	RMSE	0.3664	0.5124	0.7227	0.8035
	MAE	0.2862	0.3887	0.5695	0.6392
STA-LSTM	RMSE	0.3524	0.5170	0.7159	0.8536
	MAE	0.2762	0.3941	0.5531	0.6841
SLSTM	RMSE	0.3885	0.5399	0.7461	0.8692
	MAE	0.3051	0.4147	0.5867	0.6916
VW-SAE	RMSE	0.3763	0.5497	0.7576	0.8876
	MAE	0.2966	0.4176	0.5773	0.6947
PCR	RMSE	0.3786	0.5538	0.7870	0.9685
	MAE	0.2954	0.4180	0.5999	0.7658

LSTNet, which combine attention mechanisms with LSTM, may solve this problem to a certain extent. It can be seen that their performance degrades rapidly in long-range multi-step

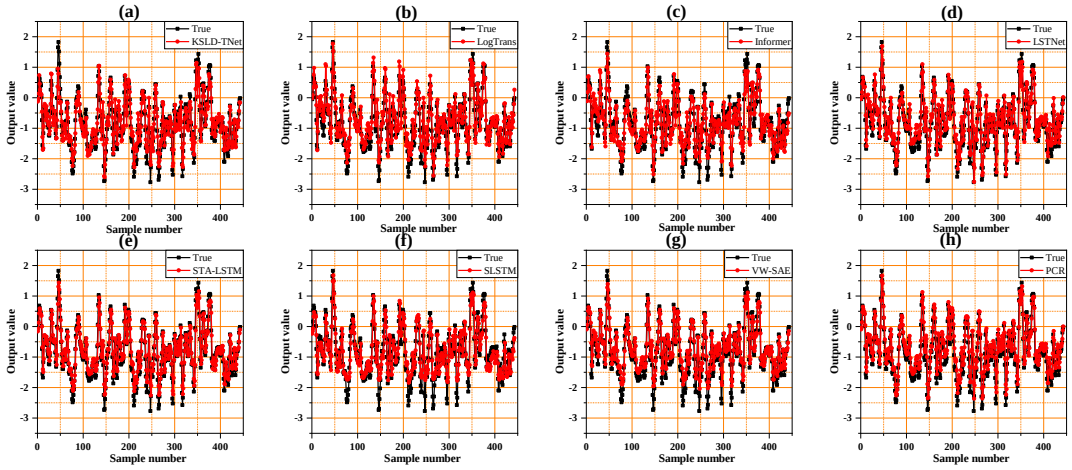


Fig. 7. The prediction curves of eight methods for predicting the sulfate ion content in the mixed potassium washing process: (a) KSLD-TNet; (b) LogTrans; (c) Informer; (d) LSTNet; (e) STA-LSTM; (f) SLSTM; (g) VW-SAE; (h) PCR.

LSTM is a dynamic model that is superior to the above two methods, it still cannot obtain satisfactory results due to its recursive structure, which makes it difficult to obtain long-term historical evolution patterns. STA-LSTM and ahead prediction, especially when the prediction window is 8. Instead, the transformer-based methods LogTrans and Informer, which abandon the recursive structure, achieve better performance in most cases. Although the single

attention structure helps LogTrans and Informer to extract a longer range of historical information, it also creates the dilemma that a large amount of redundant information overwhelms useful information, making deep feature extraction difficult. This makes their predictive performance still suboptimal. Differently, the proposed KSLD-TNet further designs a transformer-based key sample distillation strategy to eliminate redundant information, which enables it to achieve optimal performance in all cases.

In order to show the differences between all methods more intuitively, Fig. 7 gives the prediction curves of eight methods with a prediction window length of 1. It is evident that the static models PCR and VW-SAE are difficult to track the fluctuations of the time series. Although LSTM-based methods, such as SLSTM, STA-LSTM, and LSTNet, are somewhat better, they are still difficult to track accurately at drastic change points. The transformer-based methods LogTrans and Informer can track these fluctuation points more accurately. In contrast, the proposed KSLD-TNet can obtain more accurate prediction results under most time series fluctuations. In summary, KSLD-TNet has better time series tracking ability than other baseline methods.

Further, the ablation experiments of the sample distillation mechanism are carried out to verify its effectiveness. Specifically, the proposed KSLD-TNet is compared with vanilla transformer with the same structure but without key sample location and distillation strategy in terms of performance and running time. The detailed experimental results are shown in Table IV. It can be easily seen that KSLD-TNet greatly improves the inference speed compared with vanilla transformer and has a certain degree of improvement in prediction performance. This is mainly because the key sample location and distillation strategy of KSLD-TNet can filter out a large number of non-key samples and reduce the difficulty of deep feature extraction. In addition, Fig. 8 displays the training curves of the two methods for ablation experiments. It can be seen that KSLD-TNet still has good convergence performance after distilling out most of the non-key samples.

Table IV Ablation experiments of KSLD-TNet for predicting the sulfate ion content in the mixed potassium washing process

Method	Metrics	Predict window length			
		1	2	4	8
KSLD-TNet (With KSLD)	RMSE	**0.2911**	**0.4508**	**0.6666**	**0.7961**
	MAE	**0.2244**	**0.3459**	**0.5134**	**0.6150**
	Running time (s)	**3.7398**	**1.4643**	**1.3226**	**1.4033**
Vanilla transformer (Without KSLD)	RMSE	0.3250	0.6270	0.7104	0.8051
	MAE	0.2569	0.4856	0.5539	0.6343
	Running time (s)	5.4665	3.1553	2.5133	3.1772

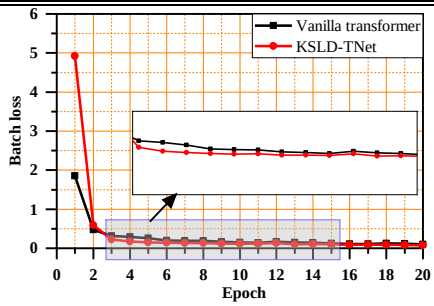


Fig. 8. The training curve comparison of KSLD-TNet and vanilla transformer.

B. Hydrocracking process

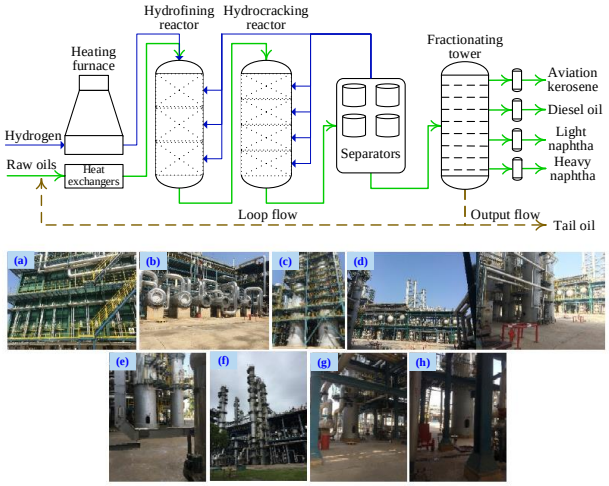


Fig. 9. The schematic diagram of the hydrocracking process: (a) heating furnace; (b) heat exchangers; (c) hydrofining reactor; (d) separators; (e) hydrogen desulfurization stripper; (f) fractionating tower; (g) debutanizer; (h) naphtha fractionator.

The hydrocracking process has become the core of modern refinery technology, which is widely used in the world. Fig. 9 illustrates the brief schematic diagram of the hydrocracking process in a Chinese petrochemical plant. The main purpose of the hydrocracking process is to mix heavy feedstock oil and hydrogen and then pass through the feed system, reaction system, high and low pressure separation system, and fractionation system in sequence to obtain aviation kerosene, diesel oil, light naphtha, and heavy naphtha oil and other products. The real-time measurement of the final distillation point of aviation kerosene is one of the key factors to ensure the quality of products. However, due to the limitations of the production environment and measuring instruments, this key quality variable cannot be obtained in real-time. Therefore, based on the process mechanism and worker experience, 43 easily measurable variables were selected as input to build the soft sensor model to provide the real-time prediction of the final distillation point of aviation kerosene. The detailed description of these variables and their time lags are shown in Table V. Similarly, to guarantee time alignment between the input variables and the prediction variable, the dataset has been preprocessed prior to its utilization.

Table V Description of auxiliary variables in the hydrocracking process

Location	Input	Variable description	Time lag (minute)
Feed section	1	Total feed	26-33
	2	Total inlet temperature of hydrogen furnace	
	3	Total exports of hydrogen furnace	
	4	Total amount of water	
Refined section	5	Inlet temperature of hydrotreating reactor	19-26
	6	Top temperature of the first bed of	

		hydrotreating reactor	
	7	Central temperature of the first bed of hydrotreating reactor	
	8	Bottom temperature of the first bed of hydrotreating reactor	
	9	Top temperature of the second bed of hydrotreating reactor	
	10	Central temperature of the second bed of hydrotreating reactor	
	11	Bottom temperature of the second bed of hydrotreating reactor	
	12	Top temperature of the third bed of hydrotreating reactor	
	13	Central temperature of the third bed of hydrotreating reactor	
	14	Bottom temperature of the third bed of hydrotreating reactor	
Slurry section	15	Bottom temperature indication of hydrotreating reactor tower	11-19
	16	Pressure difference of hydrotreating reactor	
	17	Inlet temperature of hydrocracking reactor	
	18	Top temperature of the second bed of hydrocracking reactor	
	19	Top temperature of the third bed of hydrocracking reactor	
Outfeed section	20	Top temperature of the fourth bed of hydrocracking reactor	10-11
	21	Pressure difference of hydrocracking reactor	
	22	Heat and low pressure flow from heat and low pressure oil tank to desulfurization stripper	
	23	Top pressure of heat and low pressure oil tank	
Hydrogen desulfurization section	24	Cold and low pressure flow from cold and low pressure oil tank to heat exchanger	9-10
	25	New hydrogen flow from new hydrogen compressor production to crack traffic	
	26	Back flow of desulfurization stripper	
Main fractionator section	27	Bottom liquid flow of desulfurization stripper	1-9
	28	Flow from top gas reflux of tank stripper to heat exchanger of debutanizer	
	29	Top pressure of main fractionator	
	30	Top reflux flow of main fractionator	
	31	Steam flow of stripper of main fractionator	
	32	Bottom temperature of main fractionator	
	33	Middle output flow of main fractionator	
Main fractionator side line section	34	Middle reflux temperature of main fractionator	0-1
	35	Tail oil circulating flow of main fractionator	
	36	Heavy naphtha flow from heavy naphtha reflux tank to heat exchanger	
	37	Output flow of aviation kerosene stripper	
	38	Top temperature of diesel stripper	
	39	Bottom temperature of diesel stripper	
	40	Top pressure of debutanizer	
	41	Top reflux flow of debutanizer	
	42	Top temperature of naphthalene fractionator	
	43	Bottom heavy naphtha flow of naphthalene fractionator	

In this study, a total of 1900 samples with a sampling interval of 12 hours were collected in an actual chemical enterprise in China from December 2015 to November 2018. To construct the comparative experiments, the first 1600 samples are taken as the training dataset and the last 300 samples are taken as the testing dataset. Like the previous industrial application, the trial-and-error method is used to find the optimal parameter combination of the proposed method, which is shown in Table VI. Also, the comparison methods do a lot of experiments to find the optimal experimental results.

Table VI The hyperparameters of KSLD-TNet for predicting the final distillation point of aviation kerosene in the hydrocracking process

Symbol	Description	Predict window length			
		1	2	4	8
N_e	Encoder layers	3	3	3	2
N_d	Decoder layers	1	1	5	5
α	Key sample ratio	0.7	0.6	0.6	0.9
d_{model}	Mapping dimension	512	1024	512	1024
h	Subspace number	5	4	4	4
d_{ff}	Nonlinear dimension	2048	2048	512	2048
d_q	Query dimension	64	64	32	32
d_v	Value dimension	64	64	32	32

Table VII Comparison results of eight methods for predicting the final distillation point of aviation kerosene in the hydrocracking process

Method	Metrics	Predict window length			
		1	2	4	8
KSLD-TNet	RMSE	**0.4072**	**0.4222**	**0.4977**	**0.5713**
	MAE	**0.3189**	**0.3381**	**0.3841**	**0.4587**
LogTrans	RMSE	0.4264	0.4770	0.5199	0.6115
	MAE	0.3281	0.3832	0.4181	0.5100
Informer	RMSE	0.4320	0.5182	0.5786	0.6362
	MAE	0.3368	0.4196	0.4707	0.5186
LSTNet	RMSE	0.4209	0.5184	0.5600	0.6290
	MAE	0.3282	0.3948	0.4421	0.5123
STA-LSTM	RMSE	0.4244	0.4831	0.5205	0.6198
	MAE	0.3431	0.3879	0.4191	0.5057
SLSTM	RMSE	0.4324	0.4718	0.5219	0.6126
	MAE	0.3348	0.3734	0.4206	0.4967
VW-SAE	RMSE	0.4349	0.4801	0.5497	0.6246
	MAE	0.3544	0.3873	0.4496	0.5120
PCR	RMSE	0.5600	0.6236	0.6537	0.7207
	MAE	0.3991	0.4172	0.4659	0.5608

The detailed experimental results of the eight methods of the hydrocracking process are shown in Table VII. It can still be found that the performance of the static methods PCR and VW-SAE is the worst. This further proves that the multi-step ahead prediction task of time series requires the assistance of dynamic evolutionary patterns. However, SLSTM, LSTNet and STA-LSTM have enhanced information extraction capabilities due to the introduction of attention mechanisms. But this improvement is limited and degrades sharply in long-range multi-step ahead prediction. Although the transformer-based methods LogTrans and Informer achieve better performance in most cases, they are still inferior to KSLD-TNet. In contrast, the proposed KSLD-TNet achieves the best predictive performance based on the key sample location and distillation strategy designed in this paper.

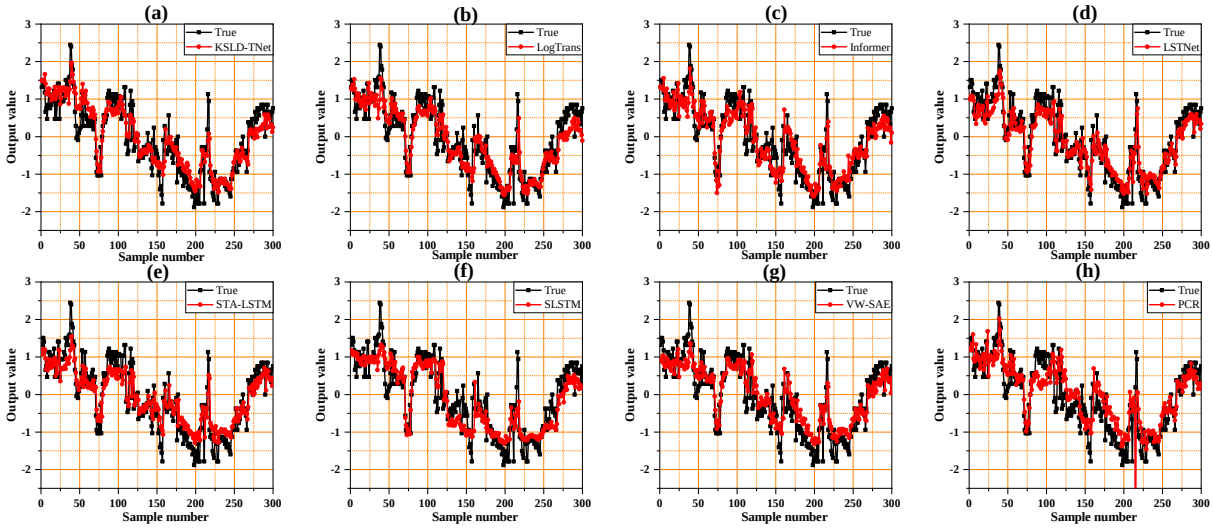


Fig. 10. The prediction curves of eight methods for predicting the final distillation point of aviation kerosene in the hydrocracking process: (a) KSLD-TNet; (b) LogTrans; (c) Informer; (d) LSTNet; (e) STA-LSTM; (f) SLSTM; (g) VW-SAE; (h) PCR.

Intuitively, Fig. 10 gives the prediction curves of all methods with a prediction window length of 1. It can be found that the proposed KSLD-TNet can obtain the best tracking performance in each period of the time series compared to other methods. This further indicates the superiority of KSLD-TNet.

Similarly, ablation experiments were constructed to verify the effectiveness of the proposed key sample location and distillation strategy. The corresponding experimental results are shown in Table VIII. It is easy to see that in most cases, especially when the prediction window length is large, KSLD-TNet is superior to the vanilla transformer without key sample location and distillation strategy in terms of prediction performance and calculation speed. In addition, the training curves of the above two methods shown in Fig. 11 clearly show that KSLD-TNet also has good convergence performance. To sum up, the KSLD-TNet proposed in this paper performs outstanding in all aspects and has good application potential in industrial processes.

Table VIII Ablation experiments of KSLD-TNet for predicting the final distillation point of aviation kerosene in the hydrocracking process

Method	Metrics	Predict window length			
		1	2	4	8
KSLD-TNet (With KSLD)	RMSE	**0.4072**	**0.4222**	**0.4977**	**0.5713**
	MAE	**0.3189**	**0.3381**	**0.3841**	**0.4587**
	Running time (s)	**2.8355**	**1.1082**	**1.0632**	**1.4523**
Vanilla transformer (Without KSLD)	RMSE	0.4402	0.5048	0.5634	0.6188
	MAE	0.3533	0.3950	0.4568	0.5089
	Running time (s)	3.5890	1.8414	2.6684	2.9095

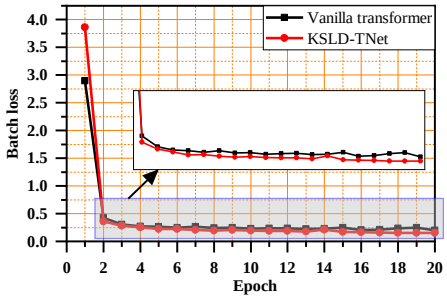


Fig. 11. The training curve comparison of KSLD-TNet and vanilla transformer.

V. Conclusion

In this study, a novel lightweight deep learning model based on key sample location and distillation transformer network (KSLD-TNet) is proposed, which can effectively streamline feature extraction and enhance the extraction of key sample information from the dataset. Through the localization and distillation of key samples, an innovative prediction framework based on a traditional transformer network is designed to improve the multi-step ahead prediction accuracy of industrial processes from the perspective of book search. Two real industrial datasets have demonstrated the superior performance of the proposed prediction framework. The proposed method has benefits in multi-step ahead prediction accuracy and model calculation efficiency compared to state-of-the-art methods. Since the sample simplification mechanism of the proposed method can reduce the amount of model computation, it is more suitable for industrial big data environments. In future research, we will consider how to use localized key samples for augmentation to enhance model performance in the context of small sample data.

REFERENCES

[1] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A review on soft sensors for monitoring, control, and optimization of industrial processes," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12868-12881, 2021.

[2] C. Liu, Y. Wang, C. Yang, and W. Gui, "Multimodal data-driven reinforcement learning for operational decision-making in industrial processes," *IEEE/CAA Journal of Automatica Sinica*, early access, 2023, doi: 10.1109/JAS.2023.123741.

[3] Y. S. Perera, D. A. A. C. Ratnaweera, C. H. Dasanayaka, and C. Abeykoon, "The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 121, Art. no. 105988, 2023.

[4] B. Shen, L. Yao, Z. Yang, and Z. Ge, "Mode Information Separated β-VAE Regression for Multimode Industrial Process Soft Sensing," *IEEE Sensors Journal*, vol. 23, no. 9, pp. 10231-10240, 2023.

[5] Y. Wang, C. Liu, H. Wu, Q. Sui, C. Yang, and W. Gui, "Revolutionizing flotation process working condition identification based on froth audio," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, Art. no. 9513012, 2023.

[6] Z. Y. Ding, J. Y. Loo, S. G. Nurzaman, C. P. Tan, and V. M. Baskaran, "A Zero-Shot Soft Sensor Modeling Approach Using Adversarial Learning for Robustness Against Sensor Fault," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5891-5901, 2023.

[7] D. Liu, Y. Wang, C. Liu, K. Wang, X. Yuan, and C. Yang, "Blackout missing data recovery in industrial time series based on masked-former hierarchical imputation framework," *IEEE Transactions on Automation Science and Engineering*, early access, 2023, doi: 10.1109/TASE.2023.3287895.

[8] Y. Wang, Q. Sui, C. Liu, K. Wang, X. Yuan, and G. Dong, "Interpretable prediction modeling for froth flotation via stacked graph convolutional network," *IEEE Transactions on Artificial Intelligence*, early access, 2023, doi: 10.1109/TAI.2023.3240114.

[9] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853-5866, 2021.

[10] Y. Wang, D. Liu, C. Liu, X. Yuan, K. Wang, and C. Yang, "Dynamic historical information incorporated attention deep learning model for industrial soft sensor modeling," *Advanced Engineering Informatics*, vol. 52, Art. no. 101590, 2022.

[11] A. Memarian, S. K. Varanasi, and B. Huang, "Mixture robust semi-supervised probabilistic principal component regression with missing input data," *Chemometrics and Intelligent Laboratory Systems*, vol. 214, Art. no. 104315, 2021.

[12] J. Zheng and Z. Song, "Mixture modeling for industrial soft sensor application based on semi-supervised probabilistic PLS," *Journal of Process Control*, vol. 84, pp. 46-55, 2019.

[13] S. Herceg, Ž. Ujević Andrijić, and N. Bolf, "Development of soft sensors for isomerization process based on support vector machine regression and dynamic polynomial models," *Chemical Engineering Research and Design*, vol. 149, pp. 95-103, 2019.

[14] P. Lian, H. Liu, X. Wang, and R. Guo, "Soft sensor based on DBN-IPSO-SVR approach for rotor thermal deformation prediction of rotary air-preheater," *Measurement*, vol. 165, Art. no. 108109, 2020.

[15] C. Liu, K. Wang, Y. Wang, and X. Yuan, "Learning deep multimanifold structure feature representation for quality prediction with an industrial application," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 5849-5858, 2022.

[16] Y. Tang, Y. Wang, C. Liu, X. Yuan, K. Wang, and C. Yang, "Semi-supervised LSTM with historical feature fusion attention for temporal sequence dynamic modeling in industrial processes," *Engineering Applications of Artificial Intelligence*, vol. 117, Art. no. 105547, 2023.

[17] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep Learning-Based Feature Representation and Its Application for Soft Sensor Modeling With Variable-Wise Weighted SAE," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3235-3243, 2018.

[18] X. Shi, Y. Li, Y. Yang, B. Sun, and F. Qi, "Multi-models and dual-sampling periods quality prediction with time-dimensional K-means and state transition-LSTM network," *Information Sciences*, vol. 580, pp. 917-933, 2021.

[19] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, and K. Gryllias, "A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges," *Mechanical Systems and Signal Processing*, vol. 167, Art. no. 108487, 2022.

[20] R. Huang, J. Li, Y. Liao, J. Chen, Z. Wang, and W. Li, "Deep Adversarial Capsule Network for Compound Fault Diagnosis of Machinery Toward Multidomain Generalization Task," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, Art. no. 3506311, 2021.

[21] W. Li, H. Lan, J. Chen, K. Feng, and R. Huang, "WavCapsNet: An Interpretable Intelligent Compound Fault Diagnosis Method by Backward Tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, Art. no. 3519811, 2023.

[22] M. K. B. Islam, M. A. H. Newton, J. Rahman, J. Trevathan, and A. Sattar, "Long range multi-step water quality forecasting using iterative ensembling," *Engineering Applications of Artificial Intelligence*, vol. 114, Art. no. 105166, 2022.

[23] Y. Wang, S. Li, C. Liu, K. Wang, X. Yuan, C. Yang, and W. Gui, "Multiscale feature fusion and semi-supervised temporal-spatial learning for performance monitoring in the flotation industrial process," *IEEE Transactions on Cybernetics*, early access, 2023, doi: 10.1109/TCYB.2023.3295852.

[24] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95-104.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.

[26] D. Liu, Y. Wang, C. Liu, X. Yuan, C. Yang, and W. Gui, "Data mode related interpretable transformer network for predictive modeling and key sample analysis in industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 9, pp. 9325-9336, 2023.

[27] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI*, 2021.

[28] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5243–5253, 2019.

[29] D. Liu, Y. Wang, C. Liu, X. Yuan, and C. Yang, "Multirate-Former: An Efficient Transformer-based Hierarchical Network for Multi-step Prediction of Multirate Industrial Processes," *IEEE Transactions on Instrumentation and Measurement*, Art. no. 3331407, 2023.

[30] 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 Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4404-4414, 2021.

[31] X. Yuan, L. Li, and Y. Wang, "Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3168-3176, 2020.

[32] A. Memarian, S. K. Varanasi, B. Huang, and G. Slot, "Smart optimization with PPCR modeling in the presence of missing data, time delay and model-plant mismatch," *Chemometrics and Intelligent Laboratory Systems*, vol. 237, Art. no. 104812, 2023.

Diju Liu received the B.Eng. degree in automation from Central South University, Changsha, in 2021, China. He is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation, Central South University, Changsha, China.

His research interests include industrial Big Data, and process modeling.

Yalin Wang (Senior Member, IEEE) received the B.Eng. degree in automation and Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 1995 and 2001, respectively.

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

Chenliang Liu (Graduate Student Member, IEEE) received the B.Eng. degree in automation from the School of Automation, Harbin University of Science and Technology, Harbin, China, in 2019. He is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation, Central South University, Changsha, China.

His research interests include deep learning, industrial process data analysis, and optimal decision-making of complex industrial processes.

Xiaofeng Yuan (Member, IEEE) received the B.Eng. degree in automation and Ph.D. degree in control science and engineering from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2011 and 2016, respectively.

From November 2014 to May 2015, he was a visiting scholar with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada He is currently a Professor with the School of Automation, Central South University. His research interests include deep learning and artificial intelligence, machine learning and pattern recognition, industrial process soft sensor modeling, process data analysis, etc.

Kai Wang (Member, IEEE) received the B.Eng. degree in automation and Ph.D. degree in control science and engineering from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2014 and 2019, respectively.

He was a Visiting Scholar with the Department of Chemical and Biological Engineering, The University of British Columbia, Vancouver, BC, Canada. He is currently an Associate Professor with the School of Automation, Central South University, Changsha, China. His research interests include industrial data analytics, process health management, and machine learning.