Front. Chem. Sci. Eng. 2024, 18(4): 42
https://doi.org/10.1007/s11705-024-2403-7

RESEARCH ARTICLE

A hybrid spatial-temporal deep learning prediction model of industrial methanol-to-olefins process

Jibin Zhou[1], Xue Li[1], Duiping Liu[2], Feng Wang[3], Tao Zhang (✉)[1], Mao Ye (✉)[1], Zhongmin Liu[1]

1 National Engineering Research Center of Lower-Carbon Catalysis Technology, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian 116023, China
2 Yulin Innovation Institute for Clean Energy, Clean Energy Innovation Institute of Chinese Academy of Sciences, Yulin 719053, China
3 Xuelang Industrial Intelligence Technology Co., Ltd., Wuxi 214000, China

Abstract Methanol-to-olefins, as a promising non-oil pathway for the synthesis of light olefins, has been successfully industrialized. The accurate prediction of process variables can yield significant benefits for advanced process control and optimization. The challenge of this task is underscored by the failure of traditional methods in capturing the complex characteristics of industrial processes, such as high nonlinearities, dynamics, and data distribution shift caused by diverse operating conditions. In this paper, we propose a novel hybrid spatial-temporal deep learning prediction model to address these issues. Firstly, a unique data normalization technique called reversible instance normalization is employed to solve the problem of different data distributions. Subsequently, convolutional neural network integrated with the self-attention mechanism are utilized to extract the temporal patterns. Meanwhile, a multi-graph convolutional network is leveraged to model the spatial interactions. Afterward, the extracted temporal and spatial features are fused as input into a fully connected neural network to complete the prediction. Finally, the outputs are denormalized to obtain the ultimate results. The monitoring results of the dynamic trends of process variables in an actual industrial methanol-to-olefins process demonstrate that our model not only achieves superior prediction performance but also can reveal complex spatial-temporal relationships using the learned attention matrices and adjacency matrices, making the model more interpretable. Lastly, this model is deployed onto an end-to-end Industrial Internet Platform, which achieves effective practical results.

Received November 1, 2023; accepted December 19, 2023;
online March 15, 2024

E-mails: zhangtao1206@dicp.ac.cn (Zhang T),
maoye@dicp.ac.cn (Ye M)

Keywords methanol-to-olefins, process variables prediction, spatial-temporal, self-attention mechanism, graph convolutional network

1 Introduction

Since 2010, the methanol-to-olefins (MTO) process has been commercialized in China as a novel technique for producing light olefins (e.g., ethylene and propylene) from non-petroleum feedstock [1,2]. Currently, 16 industrial MTO units with a total production capacity of ~9300 kt·a^{-1} have come into operation, making MTO one of the leading industrial routines for light olefins production [2]. However, the MTO process still encounters challenges such as intuitive decision-making, high energy consumption, and low levels of automation. In light of Industry 4.0 and the area of Big Data [3], the intelligent development of MTO processes is imperative to put on the agenda. Indeed, artificial intelligence (AI) technologies, such as machine learning and deep learning, have been successfully applied in chemical engineering research [4−6]. For example, Wang et al. [7] proposed a data-driven framework for optimizing the operation of the industrial MTO process using real-world industrial data sets. Zhang et al. [8] developed a method to predict and optimize the performance of industrial MTO reactors by integrating numerical simulations with machine learning techniques. Beyond that establishing a robust prediction model is of paramount importance for modern chemical plants, as it can provide valuable insights into future trends of process variables and play a crucial role in fault diagnosis, process monitoring, operation control, and optimization [9,10]. In this case, soft sensor technology, which estimates the key quality variables that are difficult to measure based on the easily measured process

variables, has been extensively studied [11−13]. In addition, predicting the future states of both key process variables and measurable process parameters simultaneously can effectively facilitate decision-making for operators. For example, when abrupt changes in key variables are predicted, operators can promptly trace the source and take appropriate actions based on predicted values of the related process parameters. From this perspective, this work intends to forecast the dynamic trends of the multi-process variables by constructing a multivariate time series prediction model for industrial MTO processes. However, the accurate prediction remains a formidable challenge due to the dynamic spatial-temporal correlations of the process variables [14,15]. Besides, in practice, timely adjustment to operating conditions is necessary to cope with market fluctuations or external disturbances. This makes the process data no longer follow the same distribution, which further exacerbates the difficulty of modeling and consequently deteriorates the prediction performance [16].

Several methods, including the mechanism models and the data-driven models, have attracted tremendous concerns from researchers for decades. Due to the intricate nature of modern industrial chemical processes and the high dimensionality of process variables, mechanism models that rely on the detailed reaction mechanism or extensive prior knowledge are always tricky and even inadequate [10,17]. Classical statistical methods, such as vector auto-regression (VAR) and auto-regressive integrated moving average, have been also proven unsuitable for multivariate time series prediction due to their inability to account for sophisticated dynamic nonlinear relationships [18]. With the rapid development of advanced monitoring sensors and distributed control system, massive amounts of process data can be stored, making data-driven models represented by machine learning and deep learning become the mainstream of current research [19]. Up to now, data-driven models have been extensively employed and achieved satisfactory prediction performances for multiple industrial processes [13,14,20−22]. Among them, recurrent neural networks (RNN) [23], long short-term memory (LSTM) [24], gated recurrent unit (GRU) [25], and convolutional neural network (CNN) [26] are commonly used methods. For example, LSTM and GRU have demonstrated successful applications in the field of electric load prediction [20]. Two CNN models are utilized for operational trend prediction in an industrial methanol production unit [21,27]. Although satisfactory prediction performances have been achieved, the extraction of long-term dependencies remains a challenge [28]. Thanks to the flexible approach in selecting and representing the information of time series, the attention mechanism [29] has been widely adopted for multivariate time series prediction, demonstrating a solid capability for long-term feature extraction [30,31]. Besides, self-attention (SA) [32], as a variant of the attention mechanism, can dynamically adjust the importance of each time series and facilitate capturing the interdependencies among multivariate time series [33−35]. For example, Huang et al. [34] proposed a dual SA network (DSANet) for multivariate time series prediction, in which the SA module was used to capture the dependencies among variables. The experimental results demonstrated that DSANet achieved state-of-the-art performance. Additionally, owing to the exceptional spatial modeling capabilities, graph convolutional networks (GCN) [36] have shown remarkable competitiveness in multivariate time series prediction [37−39]. Specifically, the process variables can be viewed as nodes in the graph, and the edges described by the adjacency matrix can be interpreted as the correlations among process variables. The construction of the adjacency matrix can be achieved through diverse approaches, including Granger causality [40], transfer entropy [41], dynamic time warping (DTW) [42], and Pearson correlation (PC) [41]. Furthermore, considering the dynamic spatial correlations between nodes, capturing the spatial dependency with a fixed adjacency matrix is particularly arduous. Therefore, GCNs with adaptive adjacency matrices have been proposed and achieved outstanding prediction performances [38,43].

Intuitively, ensemble models can potentially exhibit complementary advantages by organically integrating these deep learning models. Inspired by this idea, a novel ensemble model named CSA-MGCN is proposed in this study, which incorporates a data normalization technique, CNN, SA mechanism, and GCN. Data distribution shift poses a significant challenge to the multivariate time series prediction and inevitably compromises the robustness of the model. Therefore, a data normalization method called reversible instance normalization (RevIN) [44] is first adopted to mitigate the influence of non-stationary information. Then a deep learning component that combines CNN, SA mechanism, and GCN is utilized to model the spatial-temporal dependencies of process variables. Experimental results of an industrial MTO process testified our model outperformed the baseline models. Furthermore, this model is applied to an Industrial Internet Platform to verify its effectiveness and practicability. The primary contributions of this work are delineated as follows:

(1) We proposed a novel ensemble deep learning framework to simultaneously address the issues of temporal shift, nonlinearity, and dynamics inherent in industrial processes. Experimental results demonstrated that each block within the ensemble model contributed effectively to the overall outcome.

(2) We constructed a module with integrated CNN and SA mechanism to effectively extract temporal features. Experimental results confirmed the capability of this approach in capturing both long-term and short-term temporal patterns.

(3) We employed both static and adaptive strategies to

model spatial dependencies from different perspectives. Experimental results validated that the adaptive strategy could reveal undiscovered relations, which were not discernible through the static strategy, thereby enhancing the efficacy of correlation extraction.

(4) We evaluated our proposed model on an actual industrial MTO process and achieved state-of-the-art results, and then deployed it onto an end-to-end Industrial Internet Platform.

2 Experimental methods

2.1 Model architecture

In this section, the architecture of the proposed model is elaborated. As depicted in Fig. 1, the data in a sliding window is first processed by RevIN [44] to mitigate the influence of data distribution shift. Then the information in temporal and spatial dimensions is treated with different methods. Specifically, temporal dependencies are captured by integrating CNN and the SA mechanism, while spatial dependencies are extracted through the utilization of the multi-graph convolution network (MGCN). Subsequently, the extracted features from both temporal and spatial dimensions are fused as input to a fully connected (FC) network for final prediction. Lastly, the output is denormalized by RevIN again to obtain the final results.

2.1.1 RevIN for addressing the data distribution shift

Deep learning-based prediction models always require data to satisfy the assumption of the same distribution [44], yet in practical industrial plants, operation always needs to adapt to the dynamic market demands or unknown external disturbances, resulting in nonstationary characteristics. Thus the premise of identical data distribution is no longer tenable, inevitably compromising the accuracy of the model. Recently, RevIN, a data normalization method proposed by Kim et al. [44], has emerged as an effective solution to address the distribution shift problem and is easily applicable to other prediction models at a minimal cost. Hence, in this work, RevIN is leveraged to suppress nonstationary information at the input layer and then restore it at the output layer.

First, the mean and standard deviation of $x_k^{(i)} \in \mathbb{R}^T$ of the input data are calculated according to Eqs. (1) and (2):

$$E_t\left[x_{kt}^{(i)}\right] = \frac{1}{T_x}\sum_{j=1}^{T_x} x_{kj}^{(i)}, \tag{1}$$

$$\mathrm{Var}\left[x_{kt}^{(i)}\right] = \frac{1}{T_x}\sum_{j=1}^{T_x}\left(x_{kj}^{(i)} - E_t\left[x_{kt}^{(i)}\right]\right)^2. \tag{2}$$

Then, the input data $x^{(i)}$ are normalized by the following:

$$\hat{x}_{kt}^{(i)} = \gamma_k\left(\frac{x_{kt}^{(i)} - E_t\left[x_{kt}^{(i)}\right]}{\sqrt{\mathrm{Var}\left[x_{kt}^{(i)}\right] + \varepsilon}}\right) + \beta_k, \tag{3}$$

where $\gamma, \beta \in \mathbb{R}^K$ are learnable parameters. The normalized data are then fed as input into the following deep learning models.

2.1.2 Combined CNN and SA mechanism for extracting temporal features

The strengths of CNN and SA mechanism are synergistically employed to fully extract the temporal patterns in the data, as depicted in Fig. 1. It mainly comprises three gated linear units (GLU) [45] with different convolutional kernels ($1 \times S1$, $1 \times S2$, and $1 \times S3$) and a SA mechanism.

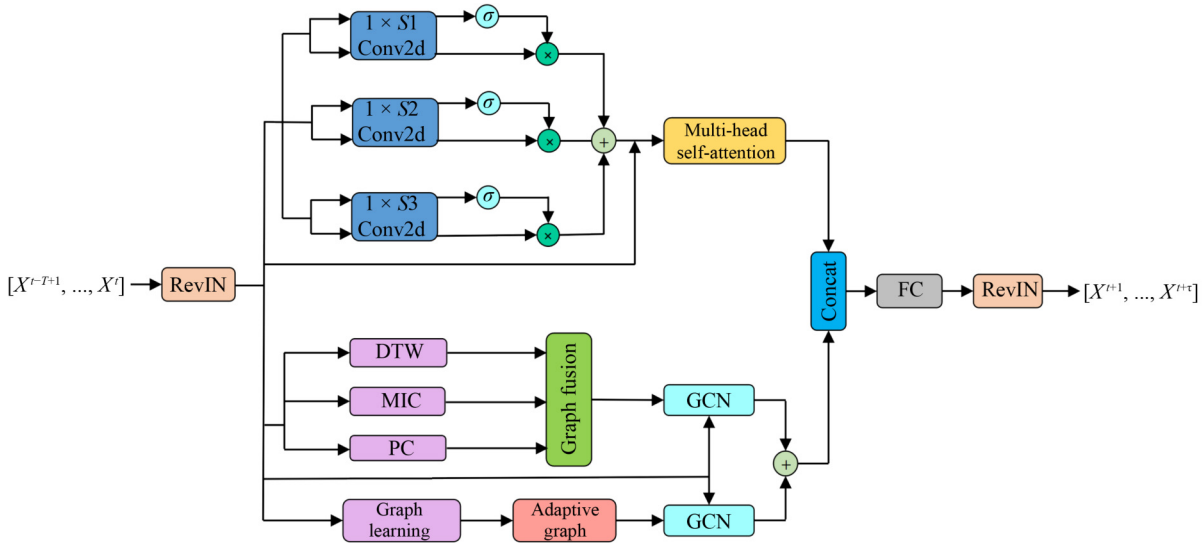

Fig. 1 The framework of the CSA-MGCN model.

Since the size of the convolutional kernel has a significant influence on the model performance, an excessively large size may result in the loss of local information while an overly small size may fail to fully extract correlations. Therefore, in this study, $S1$, $S2$, and $S3$ are designed as 1, L, and T respectively to capture features at different time scales; here L is a hyperparameter and T is the time window size. Specifically, kernel ($1 \times T$) is utilized to extract time-invariant patterns from all time steps of the univariate time series, and kernel (1×1) is used to extract the trends and the interactions of data within each individual time step, providing a dynamic weight for each time step to enhance the nonlinear representations [46], and the local temporal patterns are modeled using the kernel ($1 \times L$) [34]. Then, the learned multi-scale temporal patterns of different levels are concatenated to yield the ultimate representation:

$$C^{\text{out}} = \text{Concat}(C^1, C^2, C^3), \tag{4}$$

where the "Concat" denotes the tensor concatenation operation. C^{out} is the final representation of multi-scale GLU, $\{C^1, C^2, C^3\}$ are scale-specific representations. In addition, to tackle the issue of gradient vanishing, a residual connection is established between the original input and the C^{out}. Then input S^{input} of the SA mechanism is obtained by:

$$S^{\text{input}} = \text{Concat}(X, C^{\text{out}}). \tag{5}$$

Compared to RNNs, the SA mechanism is more adept at capturing long-term dependencies in a univariate time series by assigning different weights to each timestamp regardless of the distance [32]. Therefore, following CNN, a multi-head SA mechanism is designed to further capture the temporal patterns. In general, the SA mechanism consists of multiple attention layers running in parallel, and for each single head SA layer, the attention weight can be obtained by conducting the scaled dot-product and softmax normalization with query vector (Q), key vector (K), and value vector (V) [32].

$$Q = S^{\text{input}} W^Q,\ K = S^{\text{input}} W^K,\ V = S^{\text{input}} W^V, \tag{6}$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_m}}\right)V, \tag{7}$$

where W^Q, W^K, and W^V are trainable matrices and $1/\sqrt{d_m}$ is the scaling factor.

Multi-head attention projects Q, K, and V through h different linear transformations, followed by the concatenation of different attention results:

$$\text{Head}_i = \text{Attention}\left(QW_i^Q,\ KW_i^K,\ VW_i^V\right), \tag{8}$$

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{Head}_1, \text{Head}_2, \cdots, \text{Head}_h)\, W^O, \tag{9}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_m \times d_k}$ and $W^O \in \mathbb{R}^{d_m \times d_m}$ are parameters to learn, and $d_k = d_m/h$. A residual connection followed by a layer normalization is then inserted [32].

2.1.3 MGCN for extracting spatial features

To construct a graph, process variables can be treated as nodes and pairwise inter-variable correlations as edges. The primary challenge of applying GCN to multivariate time series prediction lies in constructing a suitable adjacency matrix that can effectively capture the spatial correlation. Most GCNs reported in the literature only use predefined adjacency matrices, but such static adjacency matrices fail to describe all possible correlations between process variables. Therefore, this work proposes the MGCN that integrates static and adaptive strategies to adequately model spatial correlations from diverse perspectives.

The interactions among process variables are typically fixed during the stable operation of an industrial process. In this case, the graph structure can be represented by a static adjacency matrix. Here, instead of simply using a single adjacency matrix, three adjacency matrices are constructed from different perspectives using DTW, maximal information coefficient (MIC), and PC. For any two variables, DTW is used to measure their similarity, MIC models their nonlinear association, and PC detects their linear relationships. Subsequently, a fused graph is obtained by element-wise weighted summation of the adjacency matrices from these three graphs. To ensure the normalization of the fusion result, Softmax is added to the weight matrix [47]. As described in Eqs. (10)–(12), the final adjacency matrix is obtained through a weighted sum of all three static adjacencies:

$$F = \sum_{i=1}^{l} w_i' \tilde{A}_i, \tag{10}$$

$$w_1',\ w_2',\ \ldots, w_l' = \text{Softmax}(w_1,\ w_2,\ \ldots, w_l), \tag{11}$$

$$\tilde{A} = I - D^{-1/2} A D^{-1/2}, \tag{12}$$

where $\tilde{A} \in \mathbb{R}^{N \times N}$is the Laplacian matrix, $A \in \mathbb{R}^{N \times N}$ is the fused adjacency matrix, $D \in \mathbb{R}^{N \times N}$ is the degree matrix, $I \in \mathbb{R}^{N \times N}$ is the identity matrix, N is the number of nodes, l is the number of the static adjacency matrix, and F is the result of graph fusion.

The interactions among process variables will change for nonstationary stages of industrial production. Therefore, a graph learning module is applied to generate an adaptive matrix to automatically model the interactions. It has been reported that the adaptive adjacency matrix can be obtained by randomly initializing two node embedding dictionaries with learnable parameters $E_1, E_2 \in \mathbb{R}^{N \times D}$ [38].

$$A_{\text{adj}} = \text{Softmax}(\text{ReLu}(E_1 E_2^T)), \tag{13}$$

where E_1 is the source node embedding and E_2 is the target node embedding. The spatial dependency weights can be obtained by multiplying E_1 and E_2. Then the ReLU and Softmax functions are applied to eliminate the weak connections and normalize the adaptive adjacency matrix.

During training, E_1 and E_2 will be updated automatically.

After determining the adaptive adjacency matrix, GCN performs feature aggregation of neighboring nodes:

$$\hat{Z} = \tilde{A}\hat{X}W, \tag{14}$$

where W denotes the trainable parameters. $\hat{X} \in \mathbb{R}^{N\times M}$ is the attribute information matrix, and $\hat{Z}$ refers to the output of GCN layers, N stands for the number of nodes, and M represents the feature dimension.

The spatial features extracted from MGCN are then fused with temporal features extracted from the SA mechanism, serving as input to a FC network to complete the prediction:

$$y = \mathrm{FC}(O_{\text{temporal}} + O_{\text{spatial}}), \tag{15}$$

where O_{temporal} denotes the extracted temporal features and O_{spatial} represents the extracted spatial features. Lastly, the model output y is denormalized [44]:

$$\hat{y}_{kt}^{(i)} = \sqrt{\mathrm{Var}\left[x_{kt}^{(i)}\right] + \varepsilon} \cdot \left(\frac{y_{kt}^{(i)} - \beta_k}{\gamma_k}\right) + E_t\left[x_{kt}^{(i)}\right], \tag{16}$$

where $\hat{y}^{(i)}$ is the ultimate prediction of the model.

2.2 MTO process

MTO process can convert methanol to light olefins using catalysts under a suitable operating temperature and pressure. The remarkable catalytic performance of SAPO-34 zeolite with CHA structure has rendered it a widely used catalyst in industrial MTO processes [1,2]. However, the formation of coke is unavoidable during the reaction, eventually leading to the deactivation of catalysts [1,2]. To maintain continuous production, timely regeneration of the coked catalysts is imperative for restoring their activity. Therefore, in industrial MTO processes, as sketched in Fig. 2, the adoption of a fluidized bed reactor-regenerator is embraced [1,2]. Figure 2 presents a simplified flow diagram of the reaction and regeneration unit in a practical MTO plant in China with the layout of key sensors. It mainly contains five operating parts: methanol feed system, catalyst circulation system, reactor unit, regenerator unit, and product analyzer system. The gasified methanol enters the reactor and reacts with the regenerated catalyst, and then the product gases are sent to the product analyzer for further analysis. Meanwhile, the coked SAPO-34 catalysts are recycled back to the regenerator via the riser and subjected to a combustion process with air to restore the catalyst activity before being recycled into the reactor. Yields of ethylene and propylene and 24 factors influencing variables, including operation temperatures, pressures, feed flows, and catalyst properties, are collected as inputs to the model, and their dynamic trends are then estimated. A total of 4463 samples, with a sampling interval of 2 h, are collected. The data set is split in chronological order with the training set (80%), validation set (10%), and test set (10%). The *k*-nearest neighbor imputation method is employed to address a limited number of missing values [48].

Table 1 provides a concise description of the selected process variables, and these process variables can be classified into five main categories: variables related to the product, variables related to the temperature, variables related to the pressure, variables related to the catalyst, and variables related to feedstock (e.g., methanol feed, steam feed, and C4 feed).

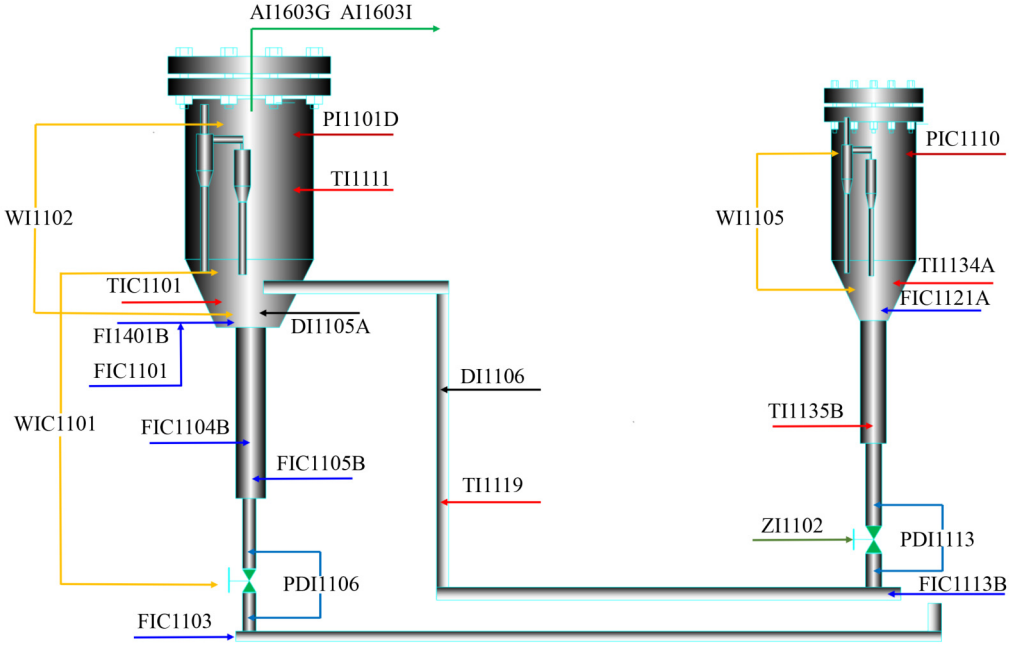

Fig. 2 Flow diagram of the reaction-regeneration unit of the MTO process.

Table 1 Description of the selected process variables

Variables	Description	Unit	Variables	Description	Unit
FI1401B	Methanol feed	$t \cdot h^{-1}$	TIC1101	Reactor temperature	°C
TI1111	Dilute phase temperature of the reactor	°C	PI1101D	Reactor pressure	MPa
WI1102	Catalyst inventory in the reactor	t	WIC1101	Catalyst density in the reactor	t
DI1105A	Catalyst density of the dense phase in the reactor	$kg \cdot m^{-3}$	TI1134A	Regenerator temperature	°C
PIC1110	Regenerator pressure	MPa	WI1105	Catalyst inventory in the regenerator	t
WZ1101	Catalyst inventory in the reactor and regenerator	t	FIC1104B	Upper stripping steam feed	$Nm^3 \cdot h^{-1}$
FIC1105B	Lower stripping steam feed	$Nm^3 \cdot h^{-1}$	FIC1113B	Steam delivery feed	$Nm^3 \cdot h^{-1}$
ZI1102	Value of slide valve of regenerated catalysts	%	DI1106	Regenerated catalyst density	$kg \cdot m^{-3}$
TI1119	Regenerated catalyst temperature	°C	TI1135B	Lower stripping temperature	°C
FIC1121A	Air feed	$Nm^3 \cdot h^{-1}$	FIC1001	C4 feed	$kg \cdot h^{-1}$
FIC1103	Nitrogen feed	$Nm^3 \cdot h^{-1}$	Q_PDI1113	Catalyst circulation rate	t·h
PDI1113	Pressure drop of the slide valve of regenerated catalysts	kPa	PDI1106	Pressure drop of the standby valve of coked catalysts	kPa
AI1603G	Ethylene yield	%	AI1603I	Propylene yield	%

2.3 Training details

All the experiments conducted in this work are carried out on a Linux-base system with Inter® Xeon Gold 5222 CPU (3.80 GHz), 8 GB RAM, and GeForce RTX 2080Ti, and the codes are implemented by Python 3.7 with PyTorch-GPU 1.3.1. Adam optimizer is adopted to train the model. The learning rate is set to 0.001, which reduces by 0.1 every 20 epochs. The number of training iterations is 500 epochs. Moreover, early stopping, the patience of 60, is implemented to avoid overfitting. The random seed is set to 2022 for the reproducibility of the experimental results. The prediction horizon h spans from 1 to 5, indicating the prediction time ranging from 2 to 10 h.

2.4 Evaluation metrics

Two commonly used evaluation metrics, mean absolute error (MAE) and mean absolute percentage error (MAPE), are employed to evaluate the performances. MAE measures the overall error of prediction results, while MAPE reflects the degree of deviation. Generally, a lower value indicates better model performance.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{17}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{18}$$

where $\hat{y}_i$ is the predicted value, y_i is the ground truth value, and N is the number of the test data set.

2.5 Baseline models

Several baseline models have been selected to highlight the performance of our model, which can be roughly categorized into three groups:

(1) Conventional methods: typical statistics method VAR and RNN variants (LSTM and GRU).

(2) Attention-based methods: LSTNet [49], which integrates CNN, RNN, and attention mechanism to effectively extract the short-term and long-term patterns for time series; and DSANet [34], which utilizes two parallel convolutional modules to capture the complex temporal dependencies and employs a SA module to model spatial dependencies.

(3) Graph-based methods: STGCN [37], Graph WaveNet [38], and MTGNN [43]. Thereinto, STGCN utilizes the Graph CNNs and gated CNNs to extract the spatial-temporal dependencies [37], Graph WaveNet models the spatial-temporal correlations by combining the self-adaptive GCN with the dilated causal convolution [38], and MTGNN uses a graph learning module to extract the relationships between variables and GCN and temporal convolution module to capture the spatial and temporal dependencies [43].

3 Results and discussion

3.1 Hyperparameter determination

In this section, the hyperparameters (time window size T and the adjustable kernel size L) that affect the prediction performance were first determined. The length of T was set to {4, 8, 12, 16, 20, 24, 28}, and the T corresponding to the lowest MAE and MAPE values was selected as the optimal hyperparameter. The performances of one-step prediction as a function of T are depicted in Fig. 3(a). It can be observed that the MAE value peaked at T = 28, indicating that the prediction performance would degrade once T was too long. In addition, from T of 4, MAE and MAPE values reached the minimum values at 16 and then gradually increased. This could be attributed to the inadequate capture of temporal characteristics when T was too small. Although a larger T could increase historical information, it would also pose challenges in training the model to capture the long-term dependencies

[50]. From Fig. 3(b), the performance of the model was also highly sensitive to the kernel size L. Deviations from an optimal value of L = 8 resulted in a deterioration of performance. Therefore, the optimal hyperparameters were determined to be T = 16 and L = 8.

3.2 Performance evaluation

Table 2 lists the evaluation results of all models for prediction horizons of 1 step (2 h), 3 steps (6 h), and 5 steps (10 h). The best results for each horizon are highlighted in bold, while the sub-optimal results are underlined. As evidenced by Table 2, VAR exhibited inferior performance due to its inability to deal well with the nonlinear dependencies between process variables. Although LSTM and GRU achieved better performances by a large margin compared to VAR, their performances were still unsatisfactory due to the neglect of spatial dependencies among process variables. Furthermore, models with both spatial and temporal correlations considered, such as attention-based and graph-based modes, gave better results than LSTM and GRU which only took temporal dependencies into account. The prediction performance was significantly improved for graph-based models, in which the spatial dependencies and temporal dependencies were explicitly captured by the use of graphs and CNNs. It was worth noting that the attention-based models outperformed the graph-based models on the whole. Specifically, DSANet exhibited the most superior performance among all baseline modes, which was justifiable considering that the SA module in DSANet could learn the dynamic spatial dependencies. Clearly, the proposed CSA-MGCN model achieved the best performances in all evaluation metrics for all prediction horizons thanks to its ability to learn both the static and adaptive structures while adequately modeling the temporal patterns. Detailedly, in comparison with DSANet, CSA-MGCN exhibited a greater improvement as the prediction increased, indicating its superiority in long-term prediction. For instance, the improvement of MAPE increased from 8.13% at prediction horizon 1 to 11.23% at prediction horizon 5.

Figure 4 illustrates the performances of different models across prediction horizons from 1 to 5. More broadly, CSA-MGCN achieved the best performance for almost all horizons, with a more pronounced improvement as the prediction horizon increased. In addition, prediction difficulty intensified with longer horizons, leading to an increase in prediction error. However, the performances of CSA-MGCN deteriorated much slower than other models. Overall, these results demonstrated that the satisfactory prediction performance achieved by CSA-MGCN can be attributed to its ability to accurately capture spatial and temporal correlations.

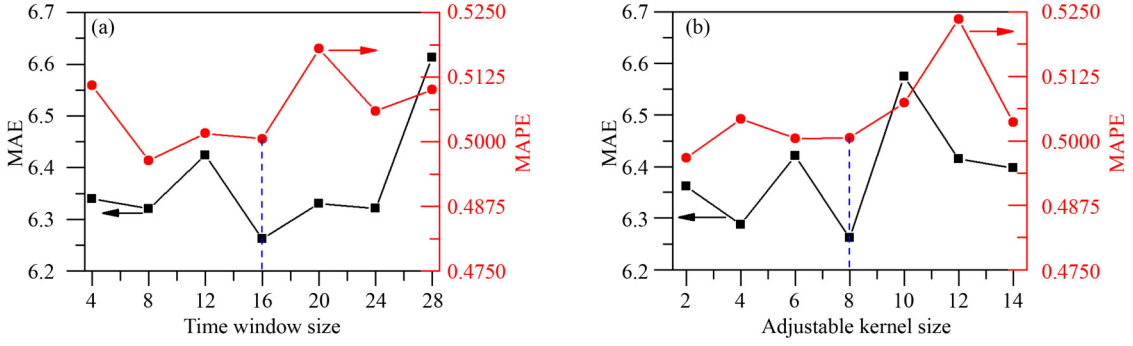

Fig. 3 Performances of the CSA-MGCN model with different (a) time window sizes T and (b) kernel size L on the test data set.

Table 2 Evaluation metrics of models for different prediction horizons on the test data set

	Models	Horizon = 1 (2 h)		Horizon = 3 (6 h)		Horizon = 5 (10 h)	
		MAE	MAPE	MAE	MAPE	MAE	MAPE
Classical	VAR	15.25	1.21%	88.37	0.29%	96.63	4.09%
	LSTM	24.39	1.40%	27.57	0.23%	37.70	2.98%
	GRU	12.94	0.94%	23.35	0.15%	22.96	2.08%
Attention	DSANet	6.65	0.54%	9.32	0.62%	10.73	0.75%
	LSTNet	7.67	0.62%	10.08	0.70%	11.85	0.85%
Graph	Graph WaveNet	8.31	0.70%	13.36	0.85%	16.53	1.19%
	STGCN	7.91	0.81%	11.27	0.86%	12.97	1.10%
	MTGNN	7.25	0.71%	10.93	0.79%	11.85	0.87%
This work	CSA-MGCN	**6.33**	**0.49%**	**9.10**	**0.58%**	**10.28**	**0.66%**
Improvements		+4.83%	+8.13%	+2.41%	+7.05%	+4.20%	+11.23%

The comparison of the prediction values of the CSA-MGCN model and the ground truth values for six typical process variables (AI1603I, AI1603G, TIC1101, TI1134A, TI1119, and DI1106) at prediction horizon of 1 are plotted in Fig. 5, providing a more comprehensive visual assessment of the model's performance. From Figs. 5(a), 5(b), 5(e), and 5(f), it could be observed that the prediction results were reliable, and CSA-MGCN consistently provided the closest predictions to the ground truth for both steady and mutation stations. Despite some deviation observed from Figs. 5(c) and 5(d), the prediction curves could still track the trend of the real values. In summary, the exceptional performances of the CSA-MGCN model enabled operators to anticipate the dynamic trends of process variables in advance, thereby significantly contributing to subsequent decision-making and early warning.

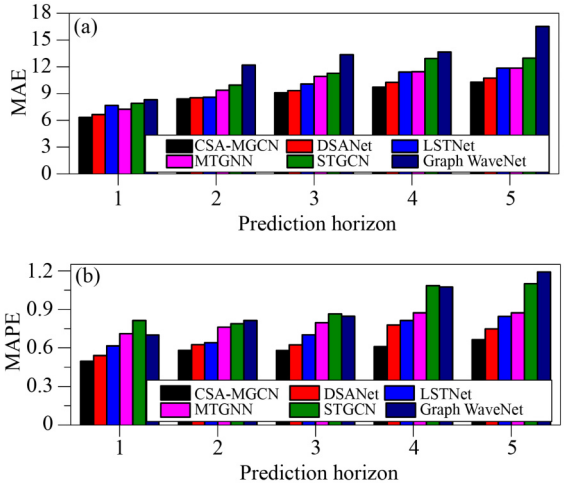

Fig. 4 Prediction performance comparison at each horizon (one prediction horizon denotes 2 h): (a) MAE and (b) MAPE.

3.3 Ablation study

In this section, an ablation study with five variants was conducted to explore the effectiveness and contribution of each component in CSA-MGCN: (1) w/oCNN: the multi-scale CNN module was removed from CSA-MGCN; (2) w/oSelf: the SA mechanism was discarded from CSA-MGCN; (3) w/oStatic: the static graph module was abandoned from CSA-MGCN; (4) w/oAdaptive: the adaptive graph module was removed from CSA-MGCN; (5) w/oRevIN: the RevIN module was removed from CSA-MGCN, and the "max-min" normalization method was adopted.

The MAE and MAPE of CSA-MGCN and its five variants for different prediction horizons are shown in Fig. 6. Notably, as depicted in Fig. 6(a), the model without RevIN (w/oRevIN) resulted in a significant decline in model performance as the prediction horizon increased. For example, when the prediction horizon was

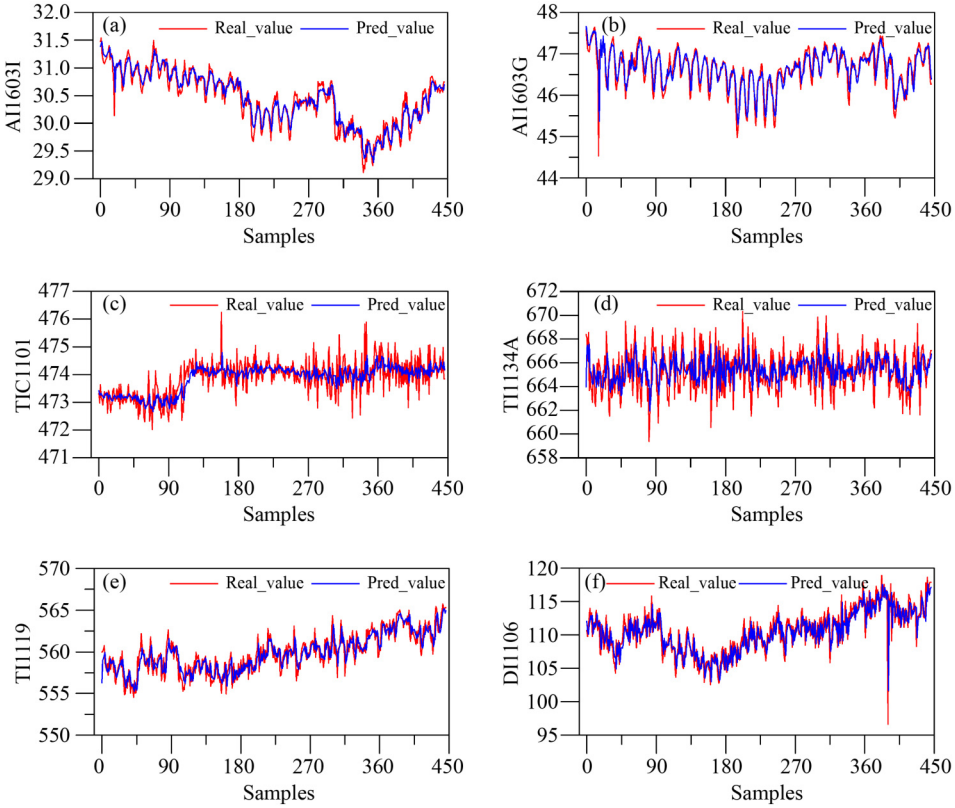

Fig. 5 Predicted and ground truth values of variables (a) AI1603I, (b) AI1603G, (c) TIC1101, (d) TI1134A, (e) TI1119, and (f) DI1106 at prediction horizon = 1.

1, MAE values of CSA-MGCN and w/oRevIN models were 6.33 and 7.17, respectively. As the prediction horizon increased to 5, MAE values increased to 10.28 and 11.86. This result highlights the effectiveness of the RevIN module in mitigating temporal distribution shifts between training and test data sets. In other words, when modeling, it should prioritize addressing data distribution shifts to avoid suboptimal prediction performances. Furthermore, as illustrated in Fig. 6(b), the substantial degradation in performance testified that the model without the adaptive graph module (w/oAdaptive) failed to accurately capture the spatial dependencies among process variables. This conclusion could also be drawn from the model without the static graph module (w/oStatic). Besides, the long-term prediction performance of the model without the CNN module (w/oCNN) remained relatively stable, which was comparable to that of CSA-MGCN. However, in terms of short-term prediction, its performance was inferior to that of CSA-MGCN, indicating the superiority of CNN in capturing short-term dynamic features. Likewise, the model without the SA mechanism module (w/oSelf) demonstrated a decline in performance as prediction horizons increased. Undoubtedly, CSA-MGCN achieved the best results for almost all prediction horizons. Based on the aforementioned analysis, it can be confirmed that all components contributed to the effectiveness and robustness of the model.

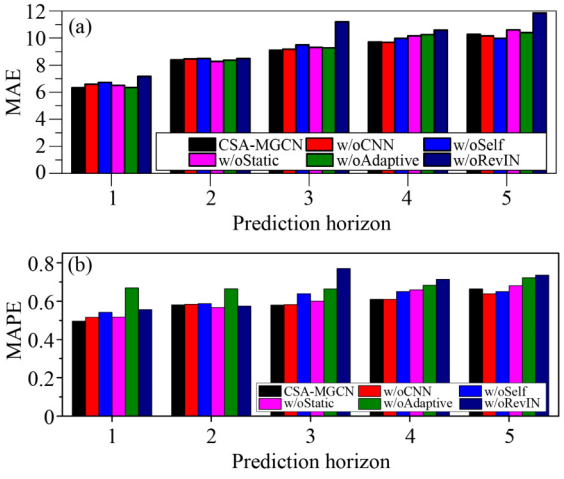

Fig. 6 Ablation test results.

3.4 Model interpretability

In this section, the weight heatmap of the adjacency matrices and the attention scores were analyzed to offer insights into the model interpretability. As shown in Fig. 7, the dimensions of the weight heatmap of adjacency matrices indicate the process variables indexes. Each cell in the heatmap reflects the degree of correlation between two variables, with lighter colors indicating a high level of correlation. Figures 7(a)–7(e) display the visualization of three static adjacency matrices, the fused adjacency matrix, and the adaptive graph at prediction horizon of 1. As can be seen from Figs. 7(a)–7(c), diverse relationships among process variables could be captured from various perspectives using different static adjacency matrices. The fused adjacency matrix was adjusted to incorporate information from the three static adjacency matrices based on the comparison between Fig. 7(d) and Figs. 7(a)–7(c). Additionally, Fig. 7(e) revealed that an adaptive learning strategy could reveal neglected relationships in static structures. Therefore, utilizing multiple graphs could significantly enhance the ability to model

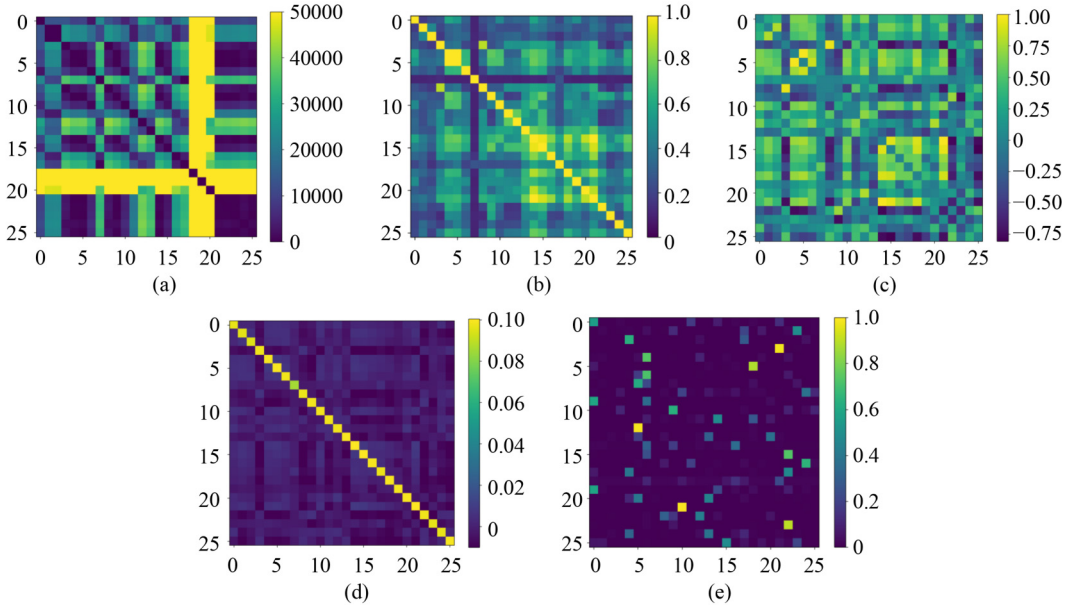

Fig. 7 Weight heatmaps of the adjacency matrices and attention matrix: (a) DTW, (b) MIC, (c) PC, (d) the fused static adjacency matrix, and (e) the adaptive matrix.

spatial dependencies.

The temporal attention weights of the last layer of different time steps on the test data set are visualized in Fig. 8, where the horizontal and vertical coordinates stand for the corresponding time steps. Lighter colors within each cell indicate greater influence, while the darker colors suggest smaller or no response. As shown in Figs. 8(a)–8(h), diverse attention patterns across different heads were observed, which was conducive to mining potential information.

3.5 Industrial application with an industrial internet platform

To demonstrate the practical application of the CSA-MGCN model in industrial plants, we deployed it on the Xuelang Suanpan Platform. Xuelang Suanpan Platform is a one-stop hybrid modeling and real-time computing system, supporting joint computing of industry mechanism models, data models, AI models, and business models. By leveraging the components and toolboxes to realize the reuse and expansion of templates. In different scenarios, only specific components need to be replaced or adjusted to achieve rapid expansion and adaptation.

In Suanpan Platform, users can construct the application for training and prediction processes by dragging components onto the back panel and setting the connection between components. The actual modeling process can then be visualized by building its business logic on the back panel. Figure 9 shows the back panel corresponding to the practical application of the model, including an OPC data acquisition and data storage components, a preprocessing component, a component containing the proposed CSA-MGCN model, a component for evaluation metrics, and a component for visualization with the front panel. Meanwhile, if the performance of the model based on the evaluation metrics is unsatisfactory, the application can retrain the model to improve its accuracy.

After building the back panel, the results can be visualized on the front panel by setting some specific configurations. Figure 10 gives the visualization results of the CSA-MGCN model at prediction horizon of 1.

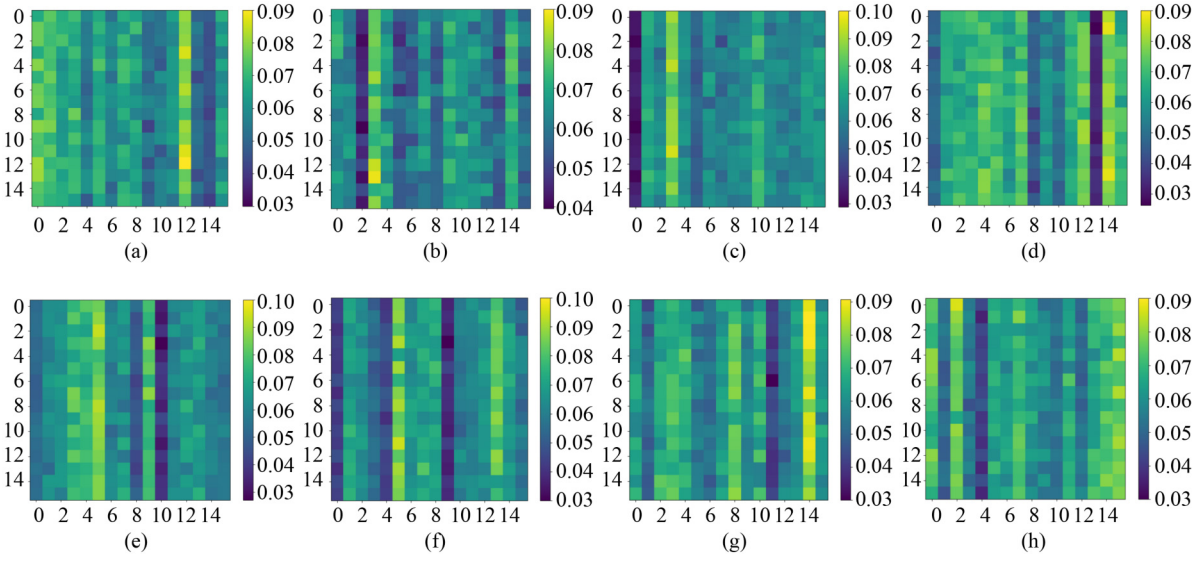

Fig. 8 (a–h) The attention weights of the 8 heads in the last layer on the test data set.

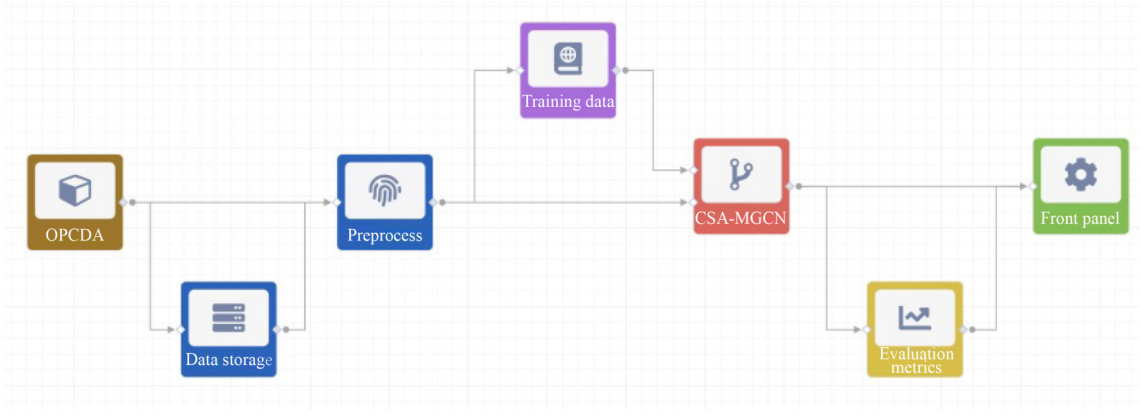

Fig. 9 The back panel corresponding to the practical application of the model.

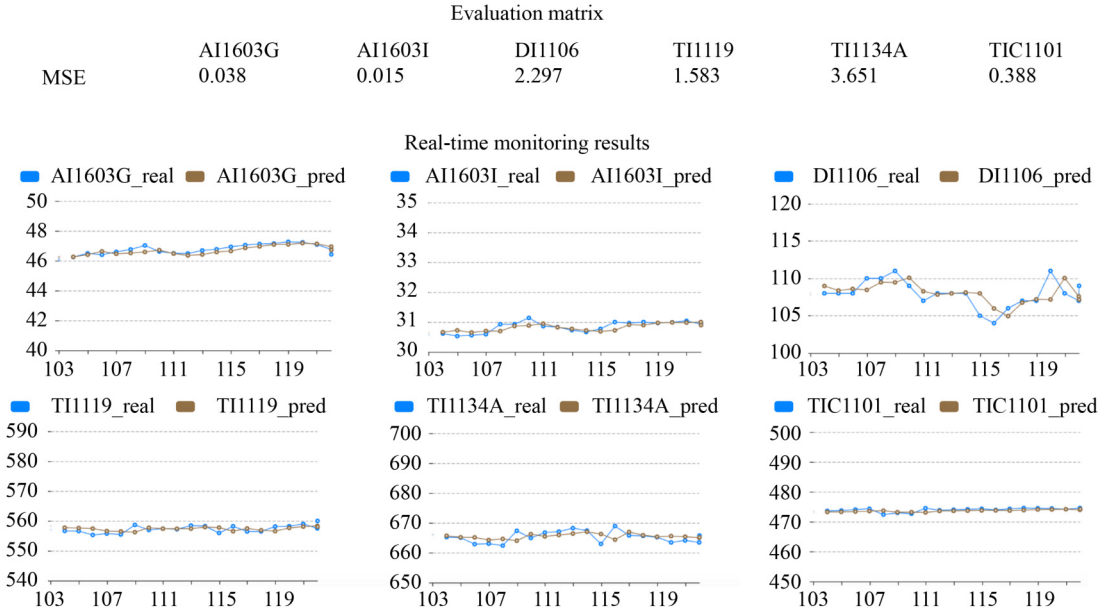

Fig. 10 The front panel corresponding to visualization results.

Concretely, it displayed the real-time monitoring results for these six indicators, as well as the detailed evaluation matrices of these indicators, which could help operators judge whether these indicators are normal and assist in operational decision-making. The real-time dynamic trends of these indicators were provided in the attached video. This work lays the technical foundation for us to implement the model in an actual MTO plant.

4 Conclusions

The prediction of process variables is critical for the construction of modern intelligent industry; however, it remains a challenging task due to the dynamic spatial-temporal dependencies among process variables and temporal distribution shift. In this work, we proposed a hybrid deep learning model for multivariate time series prediction in the industrial MTO process. By integrating data normalization, CNN, SA mechanism, and MGCN, the model is capable of effectively addressing the data distribution shift issues and capturing complex spatial-temporal dependencies. Compared to baseline models, our model achieves state-of-the-art performance in multi-process variables prediction. Furthermore, the obtained attention matrices and adjacency matrices can reveal the spatial-temporal dependencies, enhancing the interpretability of the model. Importantly, the successful implementation of the model on an industrial internet platform establishes a solid foundation for its application to MTO plants in the subsequent step.

Competing interests The authors declare that they have no competing interests.

Acknowledgements We thank the financial support from the National Natural Science Foundation of China (Grant No. 21991093), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA29050200), the Dalian Institute of Chemical Physics (DICP I202135), and the Energy Science and Technology Revolution Project (Grant No. E2010412).

References

1. Zhou J, Gao M, Zhang J, Liu W, Zhang T, Li H, Xu Z, Ye M, Liu Z. Directed transforming of coke to active intermediates in methanol-to-olefins catalyst to boost light olefins selectivity. Nature Communications, 2021, 12(1): 17
2. Ye M, Tian P, Liu Z M. DMTO: a sustainable methanol-to-olefins technology. Engineering, 2021, 7(1): 17–21
3. Li C Q, Chen Y Q, Shang Y L. A review of industrial big data for decision making in intelligent manufacturing. Engineering Science and Technology an International Journal, 2022, 29: 101021
4. Pirdashti M, Curteanu S, Kamangar M H, Hassim M H, Khatami M A. Artificial neural networks: applications in chemical engineering. Reviews in Chemical Engineering, 2013, 29(4): 205–239
5. Chiang L H, Braun B, Wang Z, Castillo I. Towards artificial intelligence at scale in the chemical industry. AIChE Journal, 2022, 68(6): e17644
6. Zhu L T, Chen X Z, Ouyang B, Yan W C, Lei H, Chen Z, Luo Z H. Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. Industrial & Engineering Chemistry Research, 2022, 61(28): 9901–9949
7. Wang Z Q, Wang L, Yuan Z H, Chen B Z. Data-driven optimal operation of the industrial methanol to olefin process based on relevance vector machine. Chinese Journal of Chemical Engineering, 2021, 34: 106–115

8. Zhang H L, Zhu A Q, Xu J, Ge W. Gas-solid reactor optimization based on EMMS-DPM simulation and machine learning. Particuology, 2024, 89: 131–143
9. Yao L, Ge Z Q. Big data quality prediction in the process industry: a distributed parallel modeling framework. Journal of Process Control, 2018, 68: 1–13
10. Sun Q Q, Ge Z Q. A Survey on deep learning for data-driven soft sensors. IEEE Transactions on Industrial Informatics, 2021, 17(9): 5853–5866
11. Yuan X F, Jia Z Z, Li L, Wang K, Ye L J, Wang Y L, Yang C H, Gui W H. A SIA-LSTM based virtual metrology for quality variables in irregular sampled time sequence of industrial processes. Chemical Engineering Science, 2022, 249: 117299
12. Lee Y S, Chen J H. Developing semi-supervised latent dynamic variational autoencoders to enhance prediction performance of product quality. Chemical Engineering Science, 2023, 265: 118192
13. Yang F, Sang Y S, Lv J C, Cao J. Prediction of gasoline yield in fluid catalytic cracking based on multiple level LSTM. Chemical Engineering Research & Design, 2022, 185: 119–129
14. Li J C, Yang B, Li H G, Wang Y J, Qi C, Liu Y. DTDR–ALSTM: extracting dynamic time-delays to reconstruct multivariate data for improving attention-based LSTM industrial time series prediction models. Knowledge-Based Systems, 2021, 211: 106508
15. Hao X, Huang G, Li Z, Zheng L, Zhao Y. A spatio-temporal data decoupling convolution network model for specific surface area prediction in cement grind process. ISA Transactions, 2023, 135: 380–397
16. Zhao C H. Perspectives on nonstationary process monitoring in the era of industrial artificial intelligence. Journal of Process Control, 2022, 116: 255–272
17. Jiang Y C, Yin S, Dong J W, Kaynak O. A review on soft sensors for monitoring, control, and optimization of industrial processes. IEEE Sensors Journal, 2021, 21(11): 12868–12881
18. De Gooijer J G, Hyndman R J. 25 years of time series forecasting. International Journal of Forecasting, 2006, 22(3): 443–473
19. Kuo Y H, Kusiak A. From data to big data in production research: the past and future trends. International Journal of Production Research, 2019, 57(15–16): 4828–4853
20. Kumar S, Hussain L, Banarjee S, Reza M. Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster. In: 2018 Fifth International Conference on Emerging Applications of Information Technology. New York: IEEE, 2018, 1–4
21. Wang Y J, Ren Y M, Li H G. Symbolic multivariable hierarchical clustering based convolutional neural networks with applications in industrial process operating trend predictions. Industrial & Engineering Chemistry Research, 2020, 59(34): 15133–15145
22. Yan F, Yang C J, Zhang X M. DSTED: a denoising spatial-temporal encoder-decoder framework for multistep prediction of burn-through point in sintering process. IEEE Transactions on Industrial Electronics, 2022, 69(10): 10735–10744
23. Connor J T, Martin R D, Atlas L E. Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks, 1994, 5(2): 240–254
24. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780
25. Cho K, Van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078, 2014
26. O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv:1511.08458, 2015
27. Wang Y J, Zhang Y C, Wu Z, Li H G, Christofides P D. Operational trend prediction and classification for chemical processes: a novel convolutional neural network method based on symbolic hierarchical clustering. Chemical Engineering Science, 2020, 225: 115796
28. Zhou J, Cui G Q, Hu S D, Zhang Z Y, Yang C, Liu Z Y, Wang L F, Li C C, Sun M S. Graph neural networks: a review of methods and applications. AI Open, 2020, 1: 57–81
29. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014
30. Yin X, Han Y, Sun H, Xu Z, Yu H, Duan X. A multivariate time series prediction schema based on multi-attention in recurrent neural network. In: 2020 IEEE Symposium on Computers and Communications (ISCC). New York: IEEE, 2020, 1–7
31. Yang Y, Xiong Q, Wu C, Zou Q, Yu Y, Yi H, Gao M. A study on water quality prediction by a hybrid CNN-LSTM model with attention mechanism. Environmental Science and Pollution Research International, 2021, 28(39): 55129–55139
32. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need. In: Advances in Neural Information Processing Systems. New York: Curran Associates Inc., 2017
33. Fu X B, Gao F, Wu J, Wei X Y, Duan F W. Spatiotemporal attention networks for wind power forecasting. In: 2019 International Conference on Data Mining Workshops. New York: IEEE, 2019, 149–154
34. Huang S T, Wang D L, Wu X, Tang A. Dsanet: dual self-attention network for multivariate time series forecasting. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. New York: Association for Computing Machinery, 2019, 2129–2132
35. Wu N, Green B, Ben X, O'Banion S. Deep transformer models for time series forecasting: the influenza prevalence case. arXiv:2001.08317, 2020
36. Scarselli F, Gori M, Tsoi A C, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Transactions on Neural Networks, 2009, 20(1): 61–80
37. Yu B, Yin H T, Zhu Z X. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv:1709.04875, 2017
38. Wu Z H, Pan S R, Long G D, Jiang J, Zhang C Q. Graph wavenet for deep spatial-temporal graph modeling. arXiv:1906.00121, 2019
39. Lu B, Gan X Y, Jin H M, Fu L Y, Zhang H S. Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. New York: Association for Computing Machinery, 2020, 1025–1034

40. Amornbunchornvej C, Zheleva E, Berger-Wolf T. Variable-lag granger causality and transfer entropy for time series analysis. ACM Transactions on Knowledge Discovery from Data, 2021, 15(4): 1–30
41. Xu H Y, Huang Y D, Duan Z H, Feng J, Song P Y. Multivariate time series forecasting based on causal inference with transfer entropy and graph neural network. arXiv:2005.01185, 2020
42. He K W, Chen X, Wu Q, Yu S, Zhou Z. Graph attention spatial-temporal network with collaborative global-local learning for citywide mobile traffic prediction. IEEE Transactions on Mobile Computing, 2022, 21(4): 1244–1256
43. Wu Z H, Pan S R, Long G D, Jiang J, Chang X J, Zhang C Q. Connecting the dots: multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: Association for Computing Machinery, 2020, 753–763
44. Kim T, Kim J, Tae Y, Park C, Choi J H, Choo J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations, 2022
45. Jin G Y, Xi Z X, Sha H Y, Feng Y H, Huang J C. Deep multi-view spatiotemporal virtual graph neural network for significant citywide ride-hailing demand prediction. arXiv:2007.15189, 2020
46. Li D F, Lin K X, Li X T, Liao J B, Du R, Chen D Q, Madden A. Improved sales time series predictions using deep neural networks with spatiotemporal dynamic pattern acquisition mechanism. Information Processing & Management, 2022, 59(4): 102987
47. Chai D, Wang L, Yang Q. Bike flow prediction with multi-graph convolutional networks. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: Association for Computing Machinery, 2018, 397–400
48. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R B. Missing value estimation methods for DNA microarrays. Bioinformatics, 2001, 17(6): 520–525
49. Lai G K, Chang W C, Yang Y M, Liu H X. Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st international ACM SIGIR Conference on Research & Development in Information Retrieval. New York: Association for Computing Machinery, 2018, 95–104
50. Fan J, Zhang K, Huang Y, Zhu Y, Chen B. Parallel spatio-temporal attention-based TCN for multivariate time series prediction. Neural Computing & Applications, 2023, 35(18): 13109–13118