Artificial Intelligence Chemistry 2 (2024) 100067

Contents lists available at ScienceDirect

Artificial Intelligence Chemistry

journal homepage: www.journals.elsevier.com/artificial-intelligence-chemistry

Spatial-temporal self-attention network based on bayesian optimization for light olefins yields prediction in methanol-to-olefins process

Jibin Zhou [a], Duiping Liu [b], Mao Ye [a,*], Zhongmin Liu [a]

[a] National Engineering Research Center of Lower-Carbon Catalysis Technology, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian 116023, China
[b] Yulin Zhongke Innovation Institute for Clean Energy, Clean Energy Innovation Institute of Chinese Academy of Sciences, Yulin 719199, China

ARTICLE INFO

Keywords:
Methanol-to-olefins
Light olefins yields prediction
Self-attention mechanism
Bayesian optimization

ABSTRACT

Methanol-to-olefins (MTO), as an alternative pathway for the synthesis of light olefins (ethylene and propylene), has gained extensive attention. Accurate prediction of light olefins yields can effectively facilitate process monitoring and optimization, as they are significant economic indexes and stable operation indicators of the industrial MTO process. However, the nonlinearity and dynamic interactions among process variables pose challenges for the prediction using traditional statistical methods. Additionally, physical-based methods relying on first-principle theory are always limited by an insufficient understanding of reaction mechanisms. In contrast, data-driven methods offer a viable solution for the prediction based solely on process data without requiring extensive process knowledge. Therefore, in this work, a data-driven approach that integrates spatial and temporal self-attention modules is proposed to capture complex interactions. Furthermore, Bayesian optimization is employed to determine the optimum hyperparameters and enhance the accuracy of the model. Studies on an actual MTO process demonstrate the superior prediction performance of the proposed model compared to baseline models. Specifically, 24 process variables are selected as the high-dimensional inputs, and yields of ethylene and propylene, as the low-dimensional outputs, are successfully predicted at various prediction horizons ranging from 2 to 8 h.

1. Introduction

As basic chemicals, light olefins (ethylene and propylene) are conventionally produced by petrochemical processes such as naphtha steam cracking and fluid catalytic cracking [1]. In recent years, methanol-to-olefins (MTO) has opened an alternative route for the synthesis of light olefins using methanol as the feedstock, which can be easily obtained from non-oil resources such as coal, natural gas, and biomass [2,3]. The first MTO industrial plant, using the technology developed by Dalian Institute of Chemical Physics (DICP), Chinese Academy of Sciences, was successfully commissioned in 2010 [2,3]. By the end of 2021, nearly 30 industrial MTO units have been licensed, making MTO one of the primary industrial routines for the production of light olefins [3]. As an emerging industrial process, quickly embracing the digitization is one of the challenges faced by the MTO process under the framework of Industry 4.0. In particular, accurate prediction of light olefins yields is currently a top priority, as it can provide a reliable basis for decision-making and give operators sufficient time to carry out effective measures by analyzing the predicted trend changes.

To build a robust prediction model, some long-standing issues need to be addressed. Firstly, industrial processes are often non-stationary due to the inherent dynamics and diversified operational conditions [4,5]. Secondly, the prediction difficulty is further raised by the nonlinearity and dynamic interactions among process variables. Mathematical methods relying on reaction mechanisms and kinetics are often impractical when applied to real-world industrial processes due to the ambiguity of physical laws between inputs and outputs, as well as the complex structures of industrial plants [6]. Classical statistical methods, such as autoregressive moving average (ARMA) [7] and autoregressive integrated moving average (ARIMA) [8], have been extensively utilized for time series prediction; however, these methods assume a linear correlation between future and historical data [9]. Consequently, they are not suitable for industrial processes and their prediction performances are unsatisfactory.

Recently, data-driven methods, represented by machine learning and deep learning, have emerged as an appealing approach for time series prediction, which can automatically learn the intricate mappings between inputs and outputs directly from the data without relying on any

* Corresponding author.
E-mail address: maoye@dicp.ac.cn (M. Ye).

https://doi.org/10.1016/j.aichem.2024.100067
Received 28 August 2023; Received in revised form 1 March 2024; Accepted 22 April 2024
Available online 30 April 2024

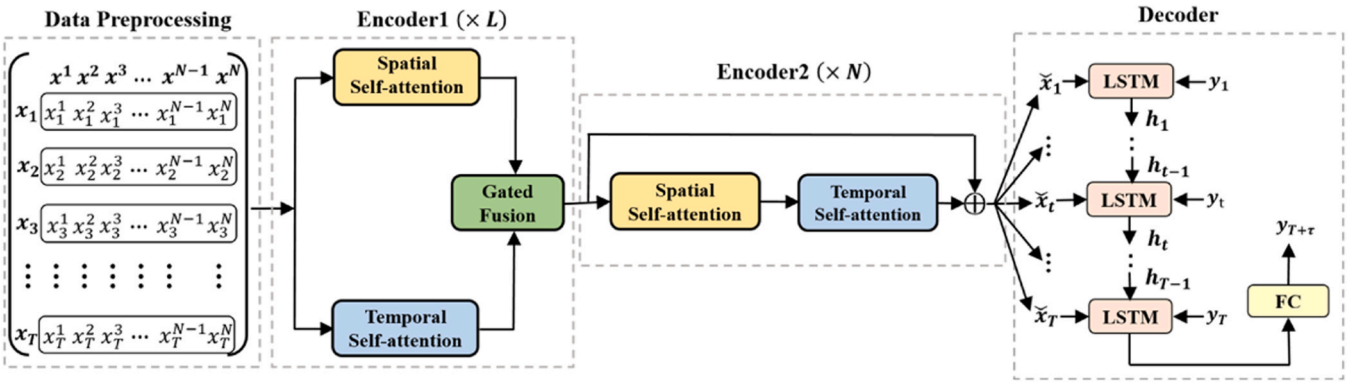

Fig. 1. The framework of the proposed STSA model.

predefined mechanisms [10–12]. In general, constructing a data-driven model necessitates an abundance of data. Fortunately, with the widespread application of distributed control systems, it has become feasible to store and collect huge process data, thereby providing a guarantee for the realization of such models. Indeed, data-driven methods have been extensively applied to multiple industrial processes [13–15]. Among them, recurrent neural network (RNN) [16] and its two popular variants, long short-term memory (LSTM) [17] and gated recurrent unit (GRU) [18], have demonstrated superior performance [13,15]. For example, Kumar et al. proved that both LSTM and GRU could deal well with the nonlinear and seasonal issues in electric load prediction [13]. The LSTM-based model, as reported by Wang et al., outperformed traditional methods in predicting the periodic energy consumption of the cooling system [15]. Despite their satisfactory prediction performance, the application of RNNs is limited due to the lack of parallelization capability [19]. Convolutional neural network (CNN) is another widely used approach and has exhibited great promise in the domain of time series prediction. Wang et al. proposed two CNN-based models utilizing the symbolic hierarchical clustering method and successfully implemented them for the operational trend prediction in an industrial methanol production unit [20,21]. Besides, attention-based models have been extensively employed in time series prediction with remarkable performances, enabling the models to automatically focus on the important parts by allocating different weights using the attention mechanism [22–25]. For instance, Qin et al. proposed a dual-stage attention based recurrent neural network (DA-RNN) to capture the long-term temporal dependency by selecting the relevant exogenous time series for multivariate prediction [22]. Aliabadi et al. testified that the attention-based RNN network gave a better performance than baseline methods in multi-step prediction of chemical process status [23]. Similarly, Li et al. adopted an attention-based LSTM model for time series prediction of an industrial methanol synthesis process [24]. Yang et al. coupled the attention mechanism with CNN-LSTM to accurately predict the water quality variables [25]. A hierarchical attention-based recurrent highway network (HRHN) was subsequently proposed by Tao et al., which has achieved outstanding performance in stock movement prediction [26]. As a variant of the attention mechanism, the self-attention mechanism has also garnered enormous interest in the field of time series prediction due to its ability to access any part of history regardless of the distance [27], rendering it more suitable for recurring patterns with long-term dependencies [28–31]. For example, Bi et al. employed two parallel self-attention layers to capture both the spatial correlations between process variables and the temporal dependencies of time series [31]. Although the aforementioned methods have achieved the desired performances, ongoing research is still dedicated to enhancing prediction performance.

Considering the diverse factors influencing the light olefins yields, including reaction conditions (temperature and pressure), catalyst properties, and feedstock quality, as well as their historical state, in this study, a deep learning model based on the self-attention mechanism is proposed. Concretely, spatial and temporal self-attention (STSA) modules are interwoven to capture the dynamic spatiotemporal correlations among process variables and differentiate weights of different time steps and process variables. For convenience, the proposed model is abbreviated as STSA. Furthermore, given the fact that deep learning models often involve multiple hyperparameters with significant impacts on prediction performance, it is crucial to employ an appropriate optimization algorithm for selecting the optimal hyperparameters. The effectiveness of Bayesian optimization in determining the optimal hyperparameters has been extensively demonstrated in previous studies [32,33]. Therefore, in this work, Bayesian optimization is adopted to identify the optimum hyperparameters. Finally, the experimental results of an actual industrial MTO process demonstrated that the prediction performances of the proposed model are considerably enhanced compared to baseline models.

2. Methodology

As illustrated in Fig. 1, the proposed STSA model, is composed of four parts: a data preprocessing module, two encoder modules, and a decoder module. In the data preprocessing module, a data normalization technique is used to mitigate the scale effect. Then the time series data is converted into a 2-D matrix with a specific time window size, where each row represents the process variables at a particular time step and each column denotes the time series data of a process variable. In the first encoder module (Encoder1), the parallel spatial and temporal self-attention module is initially applied to extract both spatial and temporal information. These extracted features are then fused using a gated convolution, resulting in a new set of features with spatial-temporal encodings. Subsequently, to further extract the spatiotemporal dependencies, the stack spatial-temporal self-attention module is performed in the second encoder module (Encoder2) [34]. As the future trends are also strongly dependent on their historical states, in the decoder module (Decoder), LSTM and a fully connected network (FC) are used to decode the historical information of the target variables and the output of the Encoder2 to accomplish the final prediction.

2.1. Data preprocessing

In this work, the process variables are divided into two series: the exogenous series $\mathbf{X}$ comprising the selected process variables such as reaction conditions, feedstock quality, and catalyst properties; and the target series $\mathbf{y}$ representing the yields of ethylene and propylene. As shown in Fig. 1, $\mathbf{X} \in R^{T\times N}$, where T is the time window size and N is the number of selected process variables. $x^k \in R^T$ is considered as the k-th time series, and $x_t \in R^N$ is the vector of exogenous series values at time t. $\mathbf{y} = (y_1, y_2, \ldots, y_T)^T \in R^{T\times 2}$ with $y_t \in R^2$. The future values across τ time steps can be predicted:

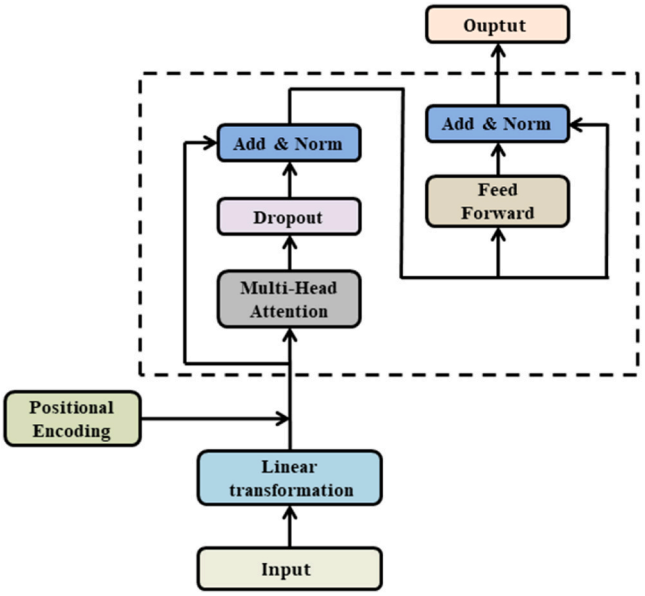

Fig. 2. The structure of the self-attention mechanism.

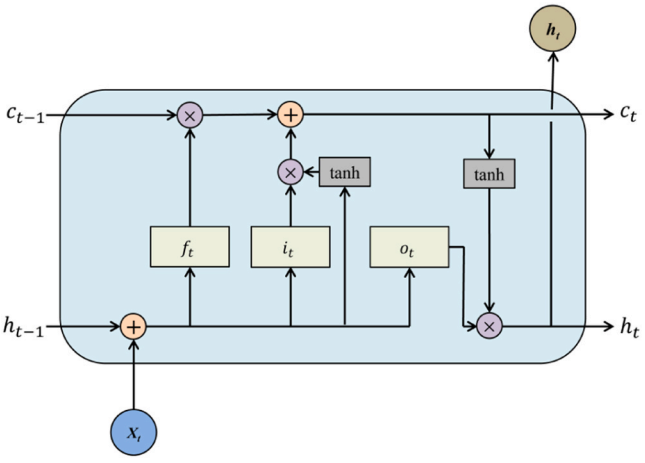

Fig. 3. The basic structure of LSTM.

$$\hat{y}_{T+\tau} = f(y_1, \quad y_2, \quad \ldots, \quad y_T, \quad x_1, \quad x_2, \quad \ldots, \quad x_T) \tag{1}$$

where $f(\bullet)$ is a nonlinear prediction function, and $\hat{y}_{T+\tau}$ represents the predicted values over the next τ time steps.

The "min-max" normalization technique is adopted to eliminate the scale effect, whereby the values of different process variables are mapped onto a range from 0 to 1:

$$X^n = \frac{X - \min(X_t)}{\max(X_t) - \min(X_t)} \tag{2}$$

where X^n denotes the values after normalization. X_t represents the training dataset. The predicted values can be converted back to the original units with denormalization.

$$\hat{y}_t = \quad \hat{y}_t^n[\max(y_t) - \min(y_t)] + \min(y_t) \tag{3}$$

where $\hat{y}_t^n$ represents the predicted values, $\max(y_t)$, $\min(y_t)$ are the maximum and minimum values of the target variables in the training dataset, respectively. $\hat{y}_t$ signifies the corresponding predicted values after denormalization.

2.2. Self-attention mechanism

As reported, the self-attention mechanism can offer a flexible approach for selecting and representing univariate sequences by learning its relationships with other learned representations, including itself, thereby enhancing its capacity to capture the internal correlation within data [27]. As shown in Fig. 2, the self-attention mechanism consists of a linear transformation module, a positional encoding module, a multi-head attention module, and a fully connected feed-forward network.

2.2.1. Linear transformation

The original input can be mapped to a high-dimension vector d_m through the FC network in the linear transformation. To be specific, in the spatial self-attention block:

$$I_t = \quad x_t W_I \tag{4}$$

where $I_t \in R^{N \times d_m}$ and $x_t \in R^{N \times 1}$ are the output and input vectors at timestamp t, respectively. $W_I \in R^{1 \times d_m}$ is the weight matrix that needs to be learned. In the temporal self-attention block:

$$I_k = \quad x_k W_o \tag{5}$$

where $I_k \in R^{T \times d_m}$ and $x_k \in R^{T \times 1}$ are the output and input vectors for the process variable k, respectively. $W_k \in R^{1 \times d_m}$ is the learnable weight matrix.

2.2.2. Positional encoding

In the absence of the recursive mechanism, the self-attention mechanism captures the sequence information using positional encoding. The type of positional encoding utilized in this work is identical to that used in the vanilla Transformer [27]:

$$PE_{(pos, \quad 2i)} = \sin\left(pos \Big/ 10000^{\frac{2i}{d_m}}\right) \tag{6}$$

$$PE_{(pos, \quad 2i+1)} = \cos\left(pos \Big/ 10000^{\frac{2i}{d_m}}\right) \tag{7}$$

where *pos* is the position and *i* is the dimension of the encoding vector. Then the positional encoding matrix is added to the input embedding as the subsequent input.

2.2.3. Multi-head attention

By employing multi-head attention, the model can focus on different positions from different representation subspaces [27]. The multi-head attention consists of several attention layers running in parallel. Hereinto, *Q*, *K*, and *V* represent the query, key, and value matrices, respectively. By utilizing these three matrixes, the attention weight can be obtained through the scaled dot-product attention function [27]:

$$Attention(Q, \quad K, \quad V) = softmax\left(\frac{QK^T}{\sqrt{d_m}}\right)V \tag{8}$$

where $\frac{1}{\sqrt{d_m}}$ is the scaling factor and accounts for the numerical stability.

Subsequently, the multi-head attention can be obtained by concatenating the different attention weights together as:

$$Multihead(Q, \quad K, \quad V) = Concat(head_1, head_2, \ldots, head_h)W^O \tag{9}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{10}$$

where $W_i^Q \in R^{d_m \times d_k}$, $W_i^K \in R^{d_m \times d_k}$, $W_i^V \in R^{d_m \times d_k}$ and $W^O \in R^{d_m \times d_m}$ are the trainable parameters, h is the number of head and $d_k = d_m/h$.

2.2.4. Feed-forward network

The feed-forward network (*FFN*) consists of two linear

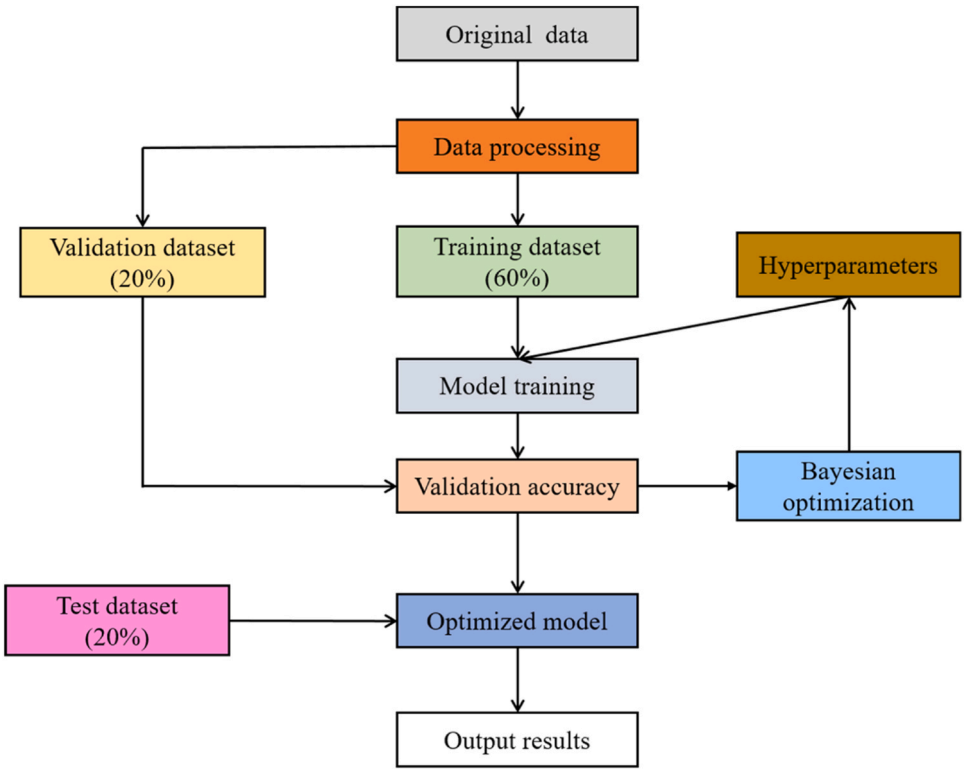

Fig. 4. Computational flowchart of this work.

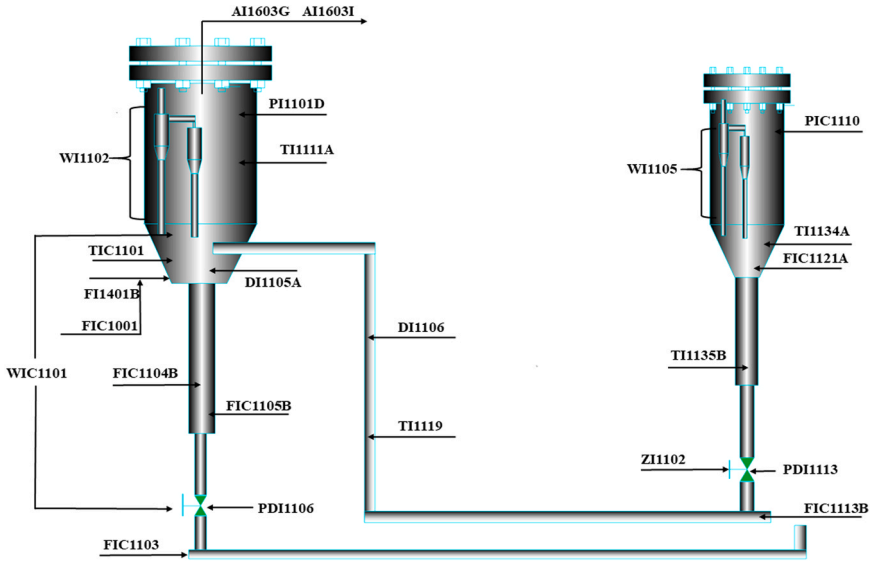

Fig. 5. Flow diagram of the reaction-regeneration unit of the MTO process [48].

transformations with a rectified linear unit (*ReLu*) activation function in between [27]:

$$FFN(x) = [ReLu(xW_1 + b_1)]W_2 \quad + b_2 \tag{11}$$

where $W_1 \in R^{d_m \times d_{ffn}}$ and $W_2 \in R^{d_{ffn} \times d_m}$ are the learnable weight matrices and d_{ffn} is the dimensionality of the inner layer.

Moreover, in order to enhance the effectiveness of the network and mitigate the issue of vanishing gradient, residual connection [35] and layer normalization [36] techniques are commonly adopted [27]. The residual connection can be defined by adding up *X* and *F(X)* [35]:

$$Output = ReLu(X + F(X)) \tag{12}$$

Layer normalization (LN) [36] can normalize the inputs across features:

$$LN(x) = \alpha \odot \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{13}$$

For the spatial self-attention block, the output can be obtained:

$$ATTN_t = LN(I_t + Multihead(Q, \quad K, \quad V)) \tag{14}$$

$$O_t = LN(ATTN_t + FFN(ATTN_t)) \tag{15}$$

Table 1
Description of the operational variables of the MTO process.

Variables	Description	Unit	Variables	Description	Unit
FI1401B	Methanol feed	t/h	TIC1101	Reactor temperature	°C
TI1111A	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/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/h
FIC1105B	Lower stripping steam feed	Nm^3/h	FIC1113B	Steam delivery feed	Nm^3/h
ZI1102	Value of slide valve for regenerated catalyst	%	DI1106	Regenerated catalyst density	kg/m^3
TI1119	Regenerated catalyst Temperature	°C	TI1135B	Lower stripping temperature	°C
FIC1121A	Air feed	Nm^3/h	FIC1001	C4 feed	kg/h
FIC1103	Nitrogen feed	Nm^3/h	Q_PDI1113	Catalyst circulation rate	t/h
PDI1113	Pressure drop of the slide valve of the regenerated catalyst	kPa	PDI1106	Pressure drop of the standby valve of the coked catalyst	kPa
AI1603G	Ethylene yield	%	AI1603I	Propylene yield	%

Likewise, for the temporal self-attention block, the output can be obtained:

$$ATTN_k = LN(I_k + Multihead(Q,\quad K,\quad V))\tag{16}$$

$$O_k = LN(ATTN_k + FFN(ATTN_k))\tag{17}$$

where $O_t \in R^{N\times d_m}$ and $O_k \in R^{T\times d_m}$ are the outputs of the spatial and the temporal self-attention blocks, respectively.

2.3. Spatial-temporal self-attention

The spatial self-attention block primarily focuses on extracting correlations between process variables, while it overlooks the capture of temporal correlations. Similarly, the temporal self-attention block effectively models the dynamic dependencies along the time dimension and lacks the ability to capture spatial interactions. Indeed, the spatial and temporal dynamics of process variables are intricately interdependent. Therefore, STSA interleaves both the spatial and temporal self-attention blocks to comprehensively model the coupled spatial-temporal interactions in a single framework. As illustrated in Fig. 1, Encoder1 employs a parallel spatial-temporal self-attention layer to extract the interdependencies across both temporal and spatial dimensions. Specifically, the embedded input is concurrently passed through the spatial self-attention block and temporal self-attention block to model the spatial interaction among process variables and capture the temporal correlations of individual process variables. Subsequently, these two types of extracted features are merged using a gated convolutional network explained in Section 2.4, yielding an integrated feature with spatial-temporal encodings. In Encoder2, a stacked spatial-temporal self-attention layer is utilized to further model the spatio-temporal correlations. Initially, the spatial self-attention block is used to capture spatial interaction with temporal information, followed by utilizing the temporal self-attention block to enhance the output spatial embeddings by incorporating temporal attention [34].

2.4. Gated convolution

According to the reports, gated convolution can learn a dynamic feature selection mechanism for each channel and each spatial location [37,38]. In gated convolution, two different convolution filters are used to perform the convolution operation in parallel. One convolution output is activated by a sigmoid function, while the other is activated by an alternative activation function, and then the element-wise multiplication is implemented [37,38]. In general, gated convolution can be defined as follows [37]:

$$Gating_{y,x} = \sum\sum W_g \bullet I\tag{18}$$

$$Feature_{y,x} = \sum\sum W_f \bullet I\tag{19}$$

$$O_{y,x} = \varnothing(Feature_{y,x}) \odot \sigma(Gating_{y,x})\tag{20}$$

where σ is the sigmoid function. $\varnothing$ represents the *ReLU* activation function in this study. W_g and W_f represent two different convolution

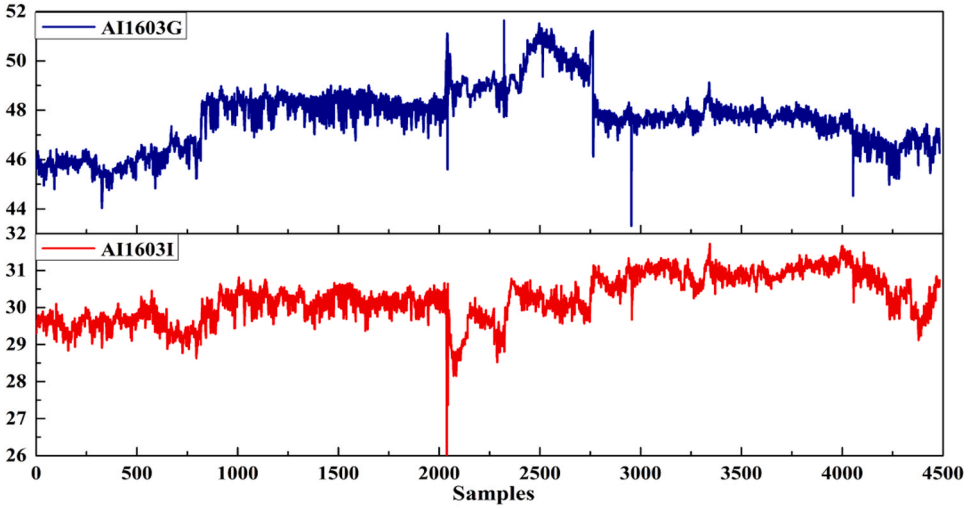

Fig. 6. Distribution of the yields of ethylene (AI1603G) and propylene (AI1603I).

Table 2
Hyperparameters and their corresponding search bounds involved in this work.

Hyperparameter	Range	Type	Hyperparameter	Range	Type
bs	[8 64]	Discrete	lr	[0.0001 0.05]	Continuous
T	[5 30]	Discrete	d_m	[32 512]	Discrete
d_{ffn}	[64 1024]	Discrete	n_{layer}	[1 8]	Discrete
n_{head}	[1 8]	Discrete	L	[1 6]	Discrete
N	[1 6]	Discrete			

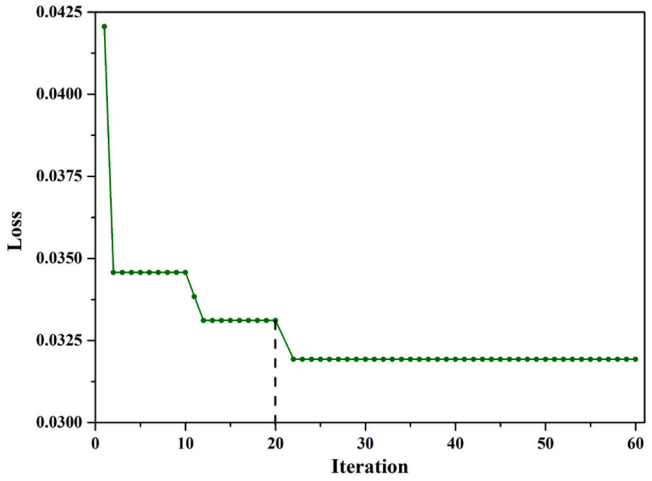

Fig. 7. Evolution of the loss versus the iterations.

Table 3
Optimum Hyperparameters determined by Bayesian optimization.

Hyperparameter	Value	Hyperparameter	Value
bs	60	lr	0.006832
T	24	d_m	75
d_{ffn}	1733	n_{layer}	1
n_{heads}	3	L	4
N	4		

filters. ⊙ denotes the element-wise multiplication.

2.5. LSTM

As a variant of RNN, LSTM can effectively learn the dependency by controlling the information flow and previous state through three gates (input gate, output gate, and forget gate) and a memory cell [17,39]. Specifically, the input gate determines the information from the new input to the memory cell, the forget gate defines the limit up to which a value is saved in the memory, and the output gate governs the information output from the memory [17,39]. The basic structure of LSTM is shown in Fig. 3.

In LSTM, the output at time t depends on the input at time t and its previous hidden states:

$$c_t, \quad h_t = LSTM(h_{t-1}, c_{t-1}, x_t) \tag{21}$$

The update of LSTM can be calculated as follows [17]:

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{22}$$

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{23}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_0) \tag{24}$$

$$c_t = \quad f_t \odot \quad c_{t-1} + i_t \odot \tanh(W_c[x_t, h_{t-1}] + b_c) \tag{25}$$

$$h_t = \quad o_t \odot \tanh(c_t) \tag{26}$$

where x_t, h_t, s_t are the input, hidden state, and cell state at time step t, respectively; i_t, f_t, o_t represent the output of the input gate, forget gate, and output gate at time step t, respectively. W_i, W_f, W_o are the corresponding convolutional kernels of the input gate, forget gate, and output gate. b_i, b_f and b_o represent the bias of the input gate, forget gate and output gate. $[x_t, h_{t-1}]$ is a concatenation of the previous hidden state h_{t-1} and the current input x_t. *LSTM* in Eq.21 represents an LSTM unit. The symbol ⊙ represents the element-wise multiplication, and the symbol σ is the logistic sigmoid function.

2.6. Bayesian optimization

It is widely acknowledged that the model performance can be significantly enhanced by selecting appropriate hyperparameters. The manual determination of hyperparameters, however, is always difficult and time-consuming. Consequently, the pursuit of efficient and automated hyperparameter selection has garnered considerable attention. As of now, various methods have been proposed [40]. Among them, optimization methods such as grid search [41] and random search [42] are commonly used but suffer from low efficiency and are prone to fall into local optima. Bayesian optimization (BO), which utilizes a probabilistic surrogate model and an acquisition function, has demonstrated its superiority over other optimization algorithms by fully taking advantage of the historical evaluations [43]. The probabilistic surrogate model, typically the Gaussian Process (GP), is utilized to predict the expectation and uncertainty of each point. The acquisition function is adopted to balance the exploration and exploitation of the search space to select the next point [43]. Lower confidence bound (LCB) and expected improvement (EI) are the two most widely used acquisition functions.

The optimization process in BO is performed as Eq.27:

$$x^* = \underset{x \in X}{argminf}(x) \tag{27}$$

where x* represents the optimal hyperparameter determined by BO, X denotes the high-dimensional hyperparameter space to be searched, f(x) is the objective function, and *argmin* refers to the process of finding the minimum value. The BO procedure is iteratively performed for a fixed number of iterations to discover a hyperparameter combination that minimizes f(x).

2.7. Computational flowchart

The computational flowchart is drawn in Fig. 4. Firstly, the data processing as described in Section 2.1 is executed, followed by dividing the processed data into three parts: training dataset, validation dataset, and test dataset. Among these, the training dataset is used to train the model while BO is applied to determine optimal hyperparameters that yield the best performance on the validation dataset. Finally, the saved model with optimal hyperparameters is verified using the test dataset. The sequential execution of the training process continues at each time step until reaching either the maximum epoch or patience of the early stopping.

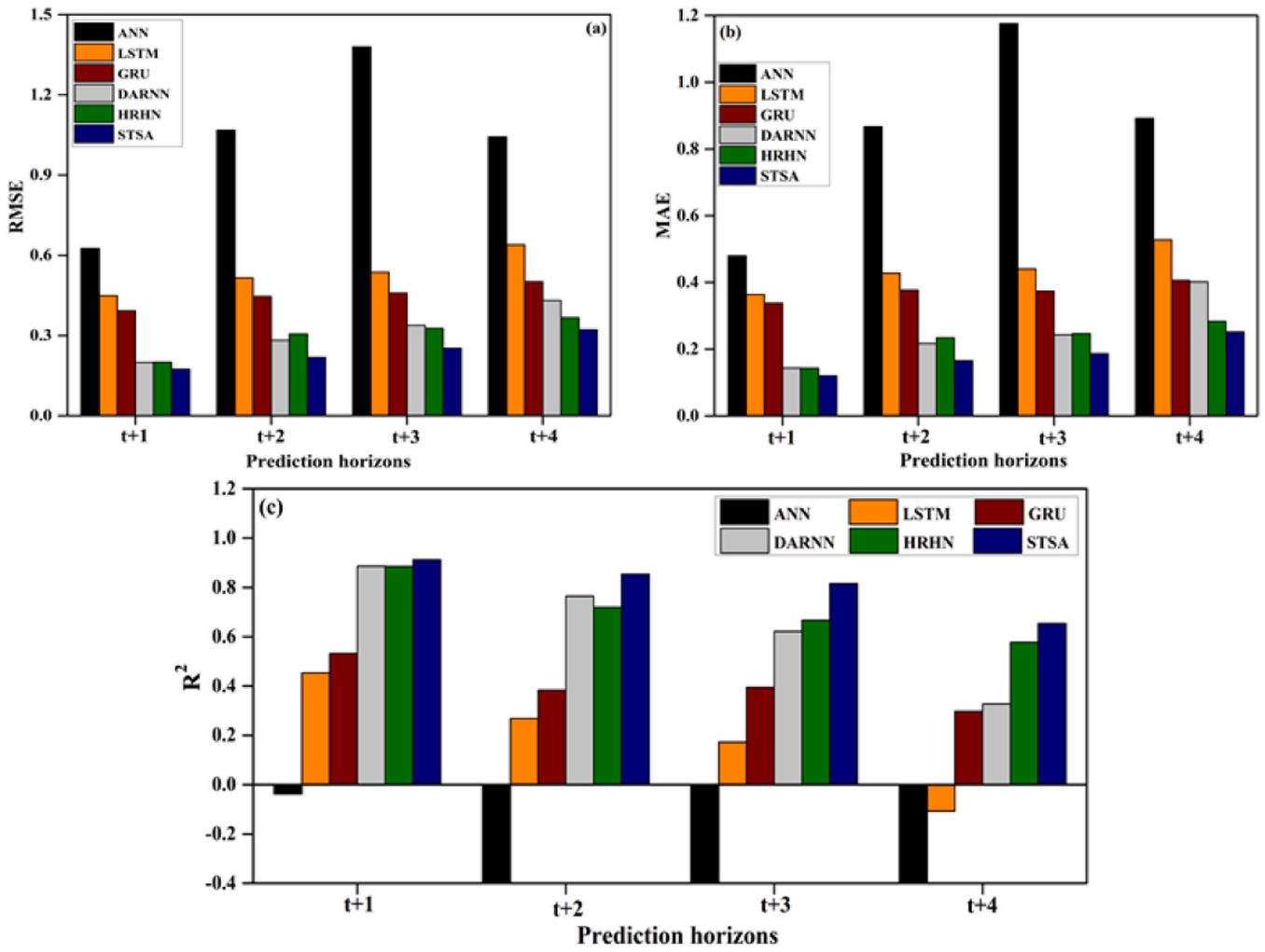

Fig. 8. RMSE (a), MAE (b), and R^2 (c) values of different models for different prediction horizons.

2.8. Evaluation metrics

To quantitatively evaluate the prediction performance, three popular metrics are utilized, namely root mean squared error (RMSE), mean absolute error (MAE), and determination coefficient (R^2). Lower values of RMSE and MAE indicate a better model fit, while a higher value of R^2 suggests a superior model fit.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_t^i - \tilde{y}_t^i\right)^2} \tag{28}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_t^i - \tilde{y}_t^i\right| \tag{29}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_t^i - \tilde{y}_t^i\right)^2}{\sum_{i=1}^{N}\left(y_t^i - \bar{y}\right)^2} \tag{30}$$

where y_t and $\tilde{y}_t$ represent the true value and predicted value at time t, respectively. $\bar{y}$ is the mean value of the true values. N is the number of samples in the test dataset.

Adam optimizer [44] is adopted to train the model. The number of training iterations is 500 epochs. Additionally, early stopping with the patience of 60 is implemented to avoid overfitting. All experiments conducted in this work are carried out on a Linux-based system with 2 Tesla V100 GPUs, and the codes are implemented by Python 3.7 with pytorch-gpu 1.3.1. The random seed is set to 123 for the reproducibility of the experimental results.

3. Methanol-to-olefins process

As a novel industrial route for the production of light olefins, the MTO process employs SAPO-34 zeolite as the catalyst owing to its excellent catalytic performance. However, coke formation over the SAPO-34 catalyst is unavoidable, which eventually leads to the deactivation of the catalyst [2,3]. To ensure continuous production, the deactivated catalyst must be regenerated in time to recover its activity. As a consequence, in the industrial MTO process, the fluidized bed reactor-regenerator is adopted [2,3]. Fig. 5 presents a simplified flow diagram of the reaction and regeneration unit in an operational MTO plant with a designed sensor layout [48]. The process mainly contains five parts: a methanol feed system, catalyst circulation system, reactor system, regenerator system, and product analyzer system. The gasified methanol is introduced into the reactor, where it reacts with the regenerated catalysts. Subsequently, the resulting product gases are sent to the product analyzer for further analysis, meanwhile, the coked catalysts are recycled back to the regenerator through a riser and react with air to recover the activity. Finally, these regenerated catalysts are returned to the reactor via another riser. Since there are numerous process variables that have significant influences on light olefins yields. In this work, 24

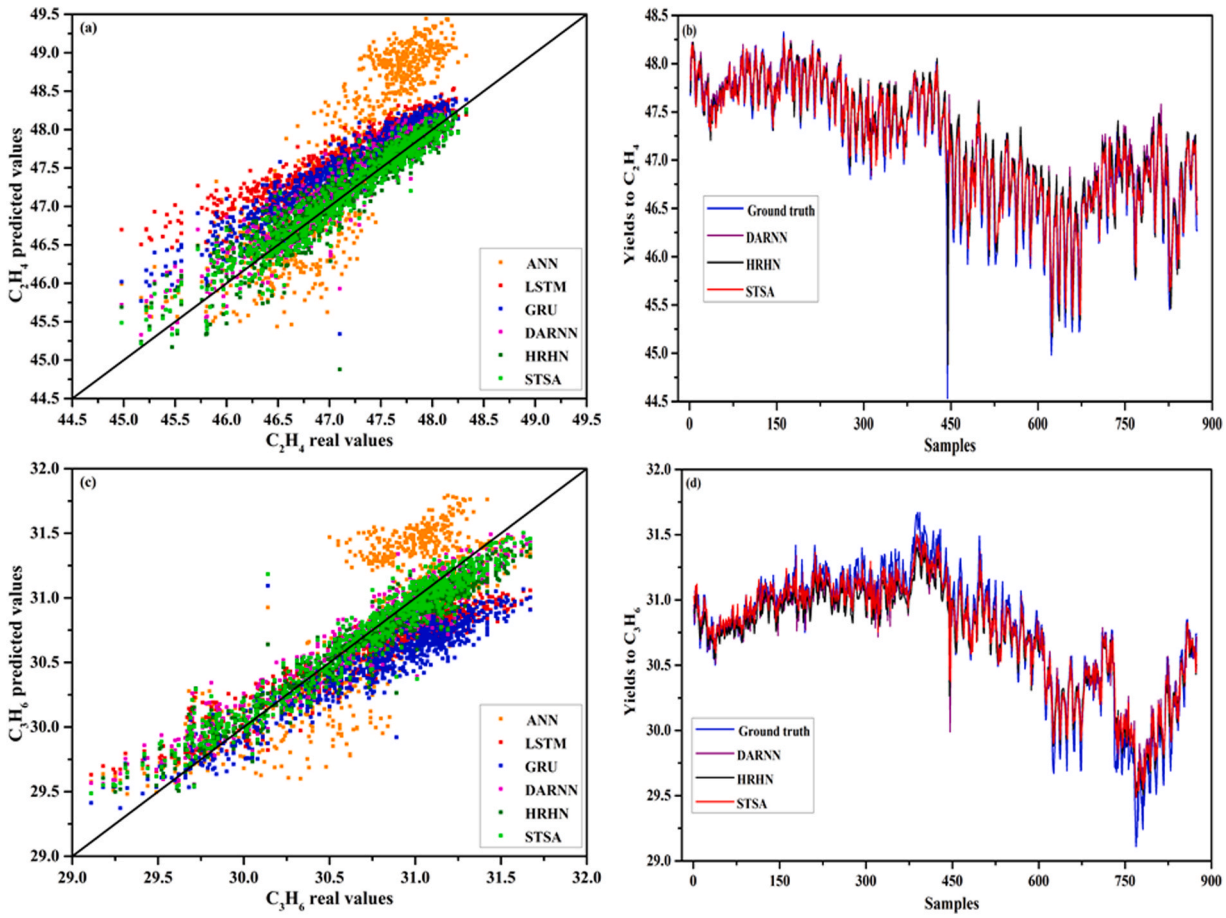

Fig. 9. Visualization results of C_2H_4 yield (a-b) and C_3H_6 yield (c-d) of different models for t+1 step prediction.

influential factors are selected as the exogenous variables based on expert knowledge, and the yields of ethylene and propylene are selected as the target variables.

Table 1 gives a brief description of the selected process variables, which can be categorized into four categories: variables related to temperature, variables related to pressure, variables related to the catalyst properties, and other manipulated variables (e.g., methanol feed, steam feed, and C4 feed).

A total of 4463 samples are collected with a sampling interval of 2 hours, and a few missing values are filled using the *k* nearest neighbor imputation method [45]. Subsequently, the dataset was partitioned into three subsets in chronological order: the initial 60% of samples for training, the subsequent 20% for validation, and the remaining 20% for testing. The partial historical data distribution of the target variables and exogenous variables are visualized in Fig. 6 and Figure S1-S2.

As illustrated in Fig. 6, the ethylene yield (AI1603G) exhibits fluctuations ranging from 44% and 52%, and the propylene yield (AI1603I) fluctuates between 26% and 32%. Besides, compared to the simulated data, industrial process data presents more intricate characteristics that pose a significant challenge for prediction. For example, anomalies such as data drift occasionally occur, which manifest as sudden increases or decreases as seen in Figure S1-S2. It is worth noting that the data depicted in Figure S1-S2 has been normalized for confidentiality reasons.

4. Results and discussions

4.1. Hyperparameters determination

Since the prediction performance of the model heavily relies on the hyperparameters, it is necessary to determine the optimal hyperparameter in advance. As a result, BO was first employed for hyperparameter determination in this study. Hyperparameters that significantly influenced the prediction performance and required tuning mainly included batch size *bs*, learning rate *lr*, time window size *T*, dimension size d_m, the dimensionality of inner-layer d_{ffn}, number layers of multi-head attention n_{layer}, number head of multi-head attention n_{head}, numbers of the first encoder *L* and the second encoder *N*. Table 2 gives the search spaces of these hyperparameters. For discrete hyperparameters, the interval was set to 1.

The BO procedure was executed using the Python suite named GPyOpt [46], which utilized GP as the probabilistic surrogate model and EI as the acquisition function. The evolution of the loss versus the iterations is plotted in Fig. 7. It was evident from the convergence diagram depicted in Fig. 7 that BO arrived at the convergence point after the 20th iteration. The corresponding optimal hyperparameters determined by BO are listed in Table 3.

4.2. Performance evaluation

In this section, we have conducted a comprehensive evaluation of the proposed model and compared it with other baseline models, including artificial neural networks (ANN), LSTM, GRU, DA-RNN, and HRHN, to showcase the superiority of our model. The results for t+1 to t+4 step with prediction horizons ranging from 2 to 8 h are presented in Fig. 8. As summarized, the proposed model outperformed all other baseline models across all prediction steps, giving the lowest values of RMSE and MAE, as well as the highest values of R^2. It was noteworthy that as the prediction horizon increased, all models experienced a decline in performance due to the complex spatial-temporal dependencies and the

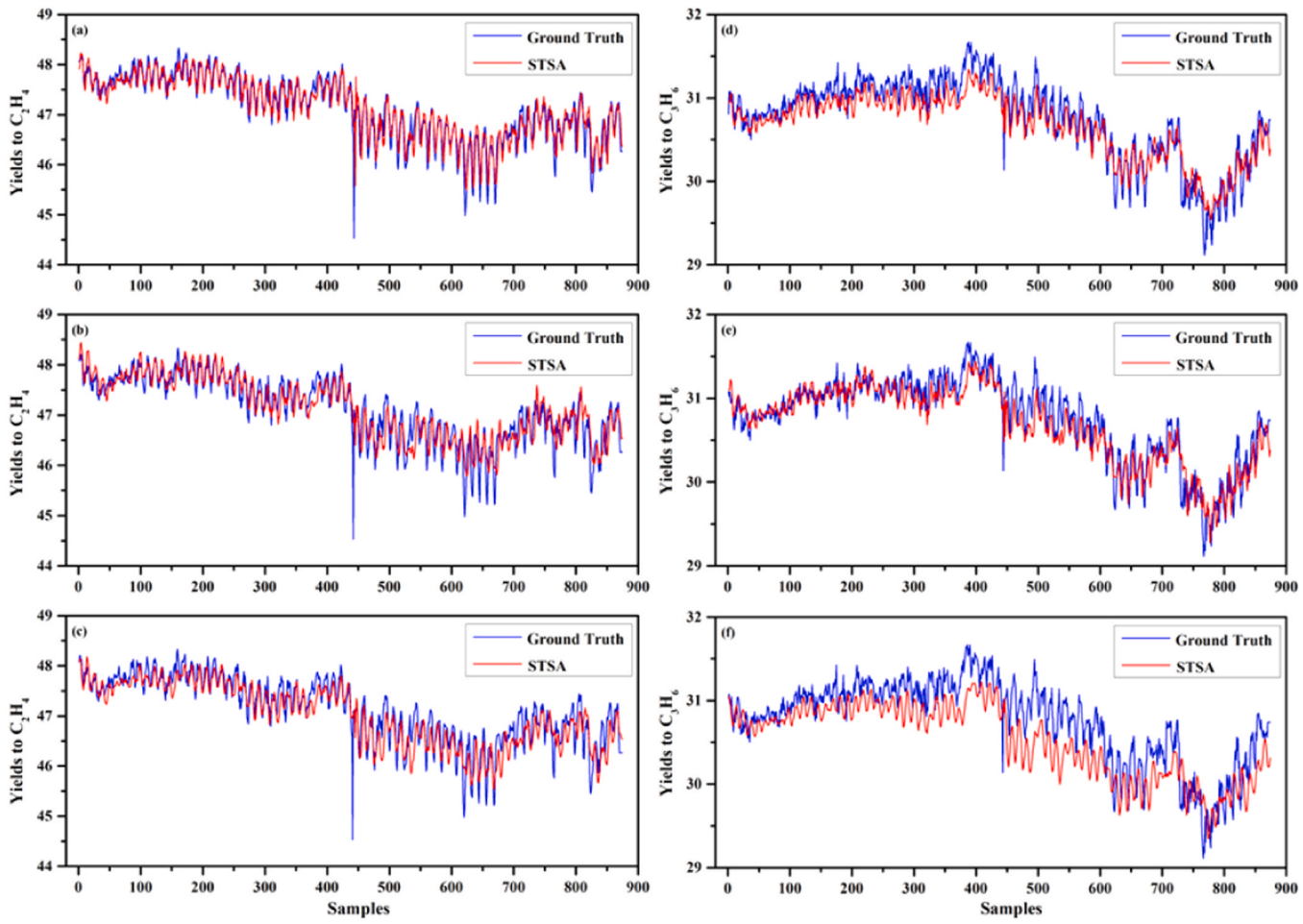

Fig. 10. Visualization results of C_2H_4 yield and C_3H_6 yield with the STSA model for t+2 step prediction (a, d), t+3 step prediction (b, e), and t+4 step prediction (c, f).

inherent difficulty of long-term prediction [47].

Concretely, the accuracy of the ANN model was observed to be the lowest, indicating its potential unsuitability for multivariate time series prediction. This limitation may be attributed to the fact that ANN is only a point-to-point mapping, which disregards the temporal characteristics inherent in time series data. As a result, ANN fails to leverage historical information to inform the future. The LSTM and GRU models both exhibited acceptable performances in t+1 step prediction owing to their capability of considering sequence information. However, their performances significantly deteriorated as the prediction horizon increased. For example, R^2 values were 0.454 and 0.533 for the t+1 step prediction, respectively, but dropped to −0.108 and 0.297 for the t+4 step prediction. One contributing factor is that the LSTM and GRU models can only take the temporal correlation of the intra-series into account, but are powerless to capturing the spatial dependencies of the inter-series, thereby making them suitable solely for short-term prediction. As anticipated, DA-RNN and HRHN models with the attention mechanism demonstrated better performance. For the t+1 step prediction, the STSA model performed comparably well as the DA-RNN and HRHN models with corresponding R^2 values of 0.912, 0.886, and 0.885, respectively. However, as the prediction horizon extended, the STSA model comfortably beat these two models by a significant margin. Concretely, the R^2 values of the STSA model for t+2 to t+4 step prediction were 0.853, 0.815, and 0.654, respectively. While those for DA-RNN were 0.765, 0.622 and 0.328 and for HRHN were 0.719, 0.667 and 0.578 for corresponding prediction steps. Based on statistical analysis results, it can be concluded that the proposed model outperformed both DA-RNN and HRHN models as also supported by observed trends in RMSE and MAE values depicted in Fig. 8. These results demonstrated that the proposed model could effectively extract the dynamic spatio-temporal dependencies among industrial multivariate time series, as evidenced by the less negative impact of increasing the prediction horizon.

To facilitate a more intuitive comparison of t+1 step prediction performances across different models, the prediction results and actual values were recorded in Fig. 9. Scatter plots of ethylene (C_2H_4) yield and propylene (C_3H_6) yield are shown in Fig. 9a and Fig. 9c. On one hand, the ANN model clearly demonstrated a tendency to overestimate the yields of C_2H_4 and C_3H_6. Both LSTM and GRU models displayed a preference for overestimating the yield of C_2H_4 while underestimating that of C_3H_6. On the other hand, the scatter points provided by DA-RNN, HRHN, and STSA models were more tightly and evenly distributed along the diagonal line. Furthermore, the dynamic trends of C_2H_4 yield and C_3H_6 yield in DA-RNN, HRHN, and STSA models are depicted in Fig. 9b and Fig. 9d. Despite that all three models could track the ground truth curve well by capturing the overall trend changes and most of the mutation information, it was still observed that the STSA model exhibited a slightly lower deviation compared to the other two models.

The comparisons between the predicted yields of C_2H_4 and C_3H_6 by the STSA model at t+2, t+3, and t+4 step predictions and their actual values are presented in Fig. 10. Evidently, despite an increasing deviation as the prediction horizon extended, the prediction results consistently aligned with the trends of the actual values, indicating that temporal patterns were accurately captured. In other words, it is feasible

for the STSA model to predict the yields of C_2H_4 and C_3H_6 in advance within a range of 2–8 h, providing operators with sufficient time to adjust the process conditions to optimize the process.

5. Conclusions

In this paper, a deep learning model based on the self-attention mechanism for the prediction of light olefins yields in the industrial-scale MTO process is proposed. This prediction model takes into account various influencing factors, including operational conditions, catalyst properties, and feedstocks, et al. Additionally, Bayesian optimization is adopted to determine the optimum hyperparameters of the model to further enhance the model performance. Experimental results have confirmed the effectiveness of the proposed model in capturing the spatiotemporal interactions among multiple process variables, handling multiple input variables, and predicting multiple output variables. Meanwhile, it can accurately predict the dynamic trends of light olefins yields in advance within 2–8 h. On one hand, the prediction outcomes can offer scientific guidance for intelligent production, such as process monitoring and optimization, for the industrial MTO process. On the other hand, while the primary focus of this study is the prediction of light olefins yields in the MTO process, it provides and validates a novel concept for enhancing prediction performance by capturing the dynamic spatiotemporal dependencies among process variables, demonstrating its potential applicability to other industrial processes. In future work, we will deploy this model onto an end-to-end Industrial Internet Platform to effectively address the specific demands of real-world industrial production.

Declaration of Competing Interest

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

Acknowledgements

We thank the financial support from the National Natural Science Foundation of China (Grant No. 21991093 and 22308348), DICP I202135, and the Energy Science and Technology Revolution Project (E2010412).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.aichem.2024.100067.

References

[1] I. Amghizar, L.A. Vandewalle, K.M. Van Geem, G.B. Marin, New trends in olefin production, Engineering 3 (2017) 171–178.
[2] J. Zhou, M. Gao, J. Zhang, W. Liu, Z. Tao, L. Hua, X. Zhaochao, Y. Mao, Z. Liu, Directed transforming of coke to active intermediates in methanol-to-olefins catalyst to boost light olefins selectivity, Nat. Commun. 12 (2021).
[3] M. Ye, P. Tian, Z. Liu, DMTO: a sustainable methanol-to-olefins technology, Engineering 7 (2021) 17–21.
[4] Z. Chen, X. Luan, F. Liu, Deep learning near-infrared quality prediction based on multi-level dynamic feature, Vib. Spectrosc. 123 (2022) 103450.
[5] D. Xu, X. Xiao, J. Liu, S. Sui, Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning, Reliab. Eng. Syst. Saf. 229 (2023) 108886.
[6] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemom. Intell. Lab. 171 (2017) 16–25.
[7] C.M. Y, H.J. C, C.C. S, Customer short term load forecasting by using ARIMA transfer function model, Proceedings 1995 International Conference on Energy Management and Power Delivery EMPD '95, 1995, 317-322.
[8] G.E.P. Box, D.A. Pierce, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models, publications, Am. Stat. Assoc. 65 (1970) 1509–1526.
[9] G. Jan, G. De, H. Rob J, 25 years of time series forecasting, Int. J. Forecast. 22 (2006) 443–473.
[10] G. Dorffner, Neural networks for time series processing, Neural Netw. World 6 (1996) 447–468.
[11] G. Zhang, B.E. Patuwo, M. Y. Hu, Forecasting with artificial neural networks:: the state of the art, Int. J. Forecast. 14 (1998) 35–62.
[12] B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, Philos. T. R. Soc. A 379 (2021) 20200209.
[13] S. Kumar, L. Hussain, S. Banarjee, M. Reza, Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster, 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), 2018, 1-4.
[14] W. Shao, Z. Ge, Z. Song, Quality variable prediction for chemical processes based on semisupervised Dirichlet process mixture of Gaussians, Chem. Eng. Sci. 193 (2019) 394–410.
[15] J.Q. Wang, Y. Du, J. Wang, LSTM based long-term energy consumption prediction with periodicity, Energy 197 (2020) 117197.
[16] J.T. Connor, R.D. Martin, L.E. Atlas, Recurrent neural networks and robust time series prediction, IEEE T. Neur. Netw. 5 (1994) 240–254.
[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[18] K. Cho, Bv Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014, 1724-1734.
[19] J. Hu, W. Zheng, Multistage attention network for multivariate time series prediction, Neurocomputing 383 (2020) 122–137.
[20] Y. Wang, Y. Zhang, Z. Wu, H. Li, P.D. Christofides, Operational trend prediction and classification for chemical processes: a novel convolutional neural network method based on symbolic hierarchical clustering, Chem. Eng. Sci. 225 (2020) 115796.
[21] Y.-j Wang, Y.M. Ren, H.-g Li, Symbolic multivariable hierarchical clustering based convolutional neural networks with applications in industrial process operating trend predictions, Ind. Eng. Chem. Res. 59 (2020) 15133–15145.
[22] Q. Yao, D. Song, H. Chen, C. Wei, G.W. Cottrell, A. Dual-Stage, Attention-based recurrent neural network for time series prediction, Twenty-Sixth Int. Jt. Conf. Artif. Intell. (2017).
[23] M.M. Aliabadi, H. Emami, M. Dong, Y. Huang, Attention-based recurrent neural network for multistep-ahead prediction of process performance, Comput. Chem. Eng. 140 (2020) 106931.
[24] J. Li, B. Yang, H. Li, Y. Wang, C. Qi, Y. Liu, DTDR–ALSTM: extracting dynamic time-delays to reconstruct multivariate data for improving attention-based LSTM industrial time series prediction models, Knowl. -Based Syst. 211 (2021) 106508.
[25] Y. Yang, Q. Xiong, C. Wu, Q. Zou, Y. Yu, H. Yi, M. Gao, A study on water quality prediction by a hybrid CNN-LSTM model with attention mechanism, Environ. Sci. Pollut. Res. 28 (2021) 55129–55139.
[26] Y. Tao, L. Ma, W. Zhang, J. Liu, W. Liu, Q. Du, Hierarchical attention-based recurrent highway networks for time series prediction, arXiv preprint arXiv: 1806.00685 (2018).
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems, 2017, 5998-6008.
[28] X. Fu, F. Gao, J. Wu, X. Wei, F. Duan, Spatiotemporal attention networks for wind power forecasting. 2019 International Conference on Data Mining Workshops (ICDMW), IEEE, 2019, pp. 149–154.
[29] S. Huang, D. Wang, X. Wu, A. Tang, DSANet: dual self-attention network for multivariate time series forecasting, Proc. 28th ACM Int. Conf. Inf. Knowl. Manag. (2019) 2129–2132.
[30] N. Wu, B. Green, X. Ben, S. O'Banion, Deep transformer models for time series forecasting: The influenza prevalence case, arXiv preprint arXiv:2001.08317 (2020).
[31] X. Bi, J. Zhao, A novel orthogonal self-attentive variational autoencoder method for interpretable chemical process fault detection and identification, Process Saf. Environ. 156 (2021) 581–597.
[32] A. Riboni, N. Ghioldi, A. Candelieri, M. Borrotti, Bayesian optimization and deep learning for steering wheel angle prediction, Sci. Rep. 12 (2022) 1–12.
[33] B. Jiang, H. Gong, H. Qin, M. Zhu, Attention-LSTM architecture combined with Bayesian hyperparameter optimization for indoor temperature prediction, Build. Environ. 224 (2022) 109536.
[34] E. Aksan, P. Cao, M. Kaufmann, O. Hilliges, Attention, please: A spatio-temporal transformer for 3d human motion prediction, arXiv preprint arXiv:2004.08692 2 (2020) 5.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2016) 770–778.
[36] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv: 1607.06450 (2016).
[37] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Free-form image inpainting with gated convolution, Proc. IEEE/CVF Int. Conf. Comput. Vis. (2019) 4471–4480.
[38] Q. Ye, M. Chu, M. Grethler, Upper limb motion recognition using gated convolution neural network via multi-channel sEMG, 2021 IEEE Int. Conf. Power Electron., Comput. Appl. (ICPECA) (2021) 397–402.
[39] H. Sak, A. Senior, F. Beaufays, Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, arXiv preprint arXiv:1402.1128 (2014).
[40] G. Luo, A review of automatic selection methods for machine learning algorithms and hyper-parameter values, Netw. Model. Anal. Health 5 (2016) 1–16.
[41] P.C. Bhat, H.B. Prosper, S. Sekmen, C. Stewart, Optimizing event selection with the random grid search, Comput. Phys. Commun. 228 (2018) 245–257.

[42] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012).
[43] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, Adv. Neural Inf. Process. Syst. 25 (2012).
[44] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv preprint arXiv:1412.6980 (2014).
[45] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520–525.
[46] J. González, Z.J.A. Dai, GPyOpt: a Bayesian optimization framework in Python, (2016).
[47] X. He, S. Shi, X. Geng, L. Xu, Dynamic Co-Attention Networks for multi-horizon forecasting in multivariate time series, Future Gener. Comp. Sy. 135 (2022) 72–84.
[48] J.B. Zhou, X. Li, D.P. Liu, F. Wang, T. Zhang, M. Ye, A hybrid spatial-temporal deep learning prediction model of industrial methanol-to-olefins process, Front. Chem. SCI. Eng. 18 (2024).