



# Multiscale convolutional neural-based transformer network for time series prediction

Zhixing Wang<sup>1</sup> · Yepeng Guan<sup>1,2,3</sup>

Received: 13 August 2023 / Revised: 19 September 2023 / Accepted: 2 October 2023 / Published online: 25 October 2023  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

Time series prediction is tough resulting from the lack of multiple time-scale dependencies and the correlation among input concomitant variables. A novel method has been developed for time series prediction by leveraging a multiscale convolutional neural-based transformer network (MCTNet). It is composed of multiscale extraction (ME) and multidimensional fusion (MF) frameworks. The original ME has been designed to mine different time-scale dependencies. It contains a multiscale convolutional feature extractor and a temporal attention-based representator, following a transformer encoder layer for high-dimensional encoding representation. In order to use the correlation among variables sufficiently, a novel MF framework has been designed to capture the relationship among inputs by utilizing a spatial attention-based highway mechanism. The linear elements of the input sequence are effectively preserved in MF, which helps MCTNet make more efficient predictions. Experimental results show that MCTNet has excellent performance for time series prediction in comparison with some state-of-the-art approaches on challenging datasets.

**Keywords** Time series prediction · Transformer · Multiscale extraction · Multidimensional fusion

## 1 Introduction

Time series prediction plays a vital role in numerous domains, including meteorology [1–3], finance [4, 5], engineering [6, 7], and industry [8–10]. It can not only use a large amount of historical data to make a reasonable analysis of the past system state space in the fields mentioned above, but also make a crucial and effective prediction for future series. However, the actual prediction tasks are often accompanied by the challenges of long-time series and multivariate collaboration [11, 12]. For example, in meteorological prediction tasks, pollutant concentration shows a trend of long-term

accumulation in space with time. At the same time, pollutant concentration is often affected by many factors, such as season, weather, and factory emissions [13].

A large number of data-driven models have been proposed to mine the regularities and patterns of the series. Autoregressive series models have been utilized for time series prediction, including auto regression (AR) [14] and ARIMA [15]. However, this kind of model [14, 15] cannot be applied to predict future trends with multiple variables. With the rapid development of deep learning, an increasing number of neural network models have been proposed to predict time series. Recurrent neural network (RNN) has been used to perform time series prediction [16, 17]. However, RNN encounters significant challenges with gradient vanishing and exploding. Besides, since the model does not contain long-term memory, it is difficult to achieve significant results on long-term time series. Some models [18–21] have been developed to address the issue of an unstable gradient. A gated unit has been introduced in [18–21] to constrain the loss of long-term memory. It makes the gradient more stable. In addition, naive LSTM [19] added a forget gate to reinforce the importance of short-term memory. However, the long-term memory loss problem still exists in the models [18–21]. With the continuous growth of the sequence, the prediction effect of the model is gradually

✉ Yepeng Guan  
Ypguan@Shu.Edu.Cn

<sup>1</sup> School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

<sup>2</sup> Key Laboratory of Advanced Display and System Application, Ministry of Education, Shanghai 200072, China

<sup>3</sup> Key Laboratory of Silicate Cultural Relics Conservation, Shanghai University, Ministry of Education, Shanghai 200444, China

becoming worse [21]. An encoder–decoder architecture has been designed to predict time series in [22]. It alleviates long-term dependency loss and enables the prediction of arbitrary step sizes through an autoregressive architecture. MS-LSTM [23] adds an attention mechanism to ensure the effectiveness of the prediction results. Att-LSTM [24] superimposes the attention mechanism to get global correlation for improving the performance of the model in long-term prediction. However, it is impossible to extend the spatial distance of dependency extraction to infinite lengths in [22–24] due to their causal architectures.

With the development of deep learning, more and more transformer-based [25] deep learning methods have been proposed for time series prediction, such as Transformer [26] and ConvTrans [27]. However, transformer-based methods [26, 27] only take atomic time-scale data as input without different fine-grained feature fusions. They cannot make accurate predictions without rich contextual information. At the same time, the multidimensional input variables are embedded into the feature representation by a simple linear transformation [26, 27], which makes them impossible to mine the correlation between variables.

A novel time series prediction method has been developed in the multiscale convolutional neural-based transformer network (MCTNet). It is composed of multiscale extraction (ME) and multidimensional fusion (MF) frameworks. The ME framework contains both a multiscale convolutional feature extractor and a temporal attention-based representator. It can be used to mine different time-scale dependencies. In order to use the correlation among variables sufficiently, the MF framework has been designed to capture the relationship among inputs in a spatial attention-based highway mechanism. The linear elements of the input sequence are effectively preserved in MF. Experimental results demonstrate that the MCTNet has distinguished performance for time series prediction in comparison with some state-of-the-art methods. Some of the main contributions are summarized as follows: Firstly, a ME framework has been proposed to mine the context dependency over different time lengths. It contains both a multiscale convolutional feature extractor and a temporal attention-based representator. It can be applied to extract context dependence for different timescales. Secondly, a MF framework has been developed to capture the relationship among inputs by utilizing a spatial attention-based highway mechanism. It can be used to learn patterns among different variables. The linear elements of the input sequence are effectively preserved in MF. Finally, and importantly, MCTNet can learn the sequence representations at different timescales and adaptively fuse representations based on each contribution level. Experiments highlight that MCTNet has outstanding performance for time series prediction based on comparisons with some state-of-the-art methods on challenging datasets.

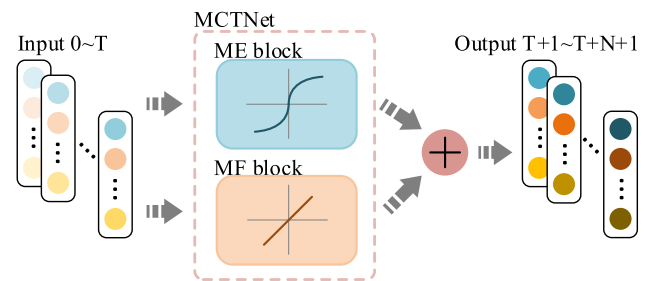


Fig. 1 A workflow of MCTNet for time series prediction

The rest of this paper is as follows: MCTNet-based time series prediction is described in Sect. 2. Experimental analyses and discussions are introduced in Sect. 3 and followed by conclusions in Sect. 4.

## 2 MCTNet-based time series prediction

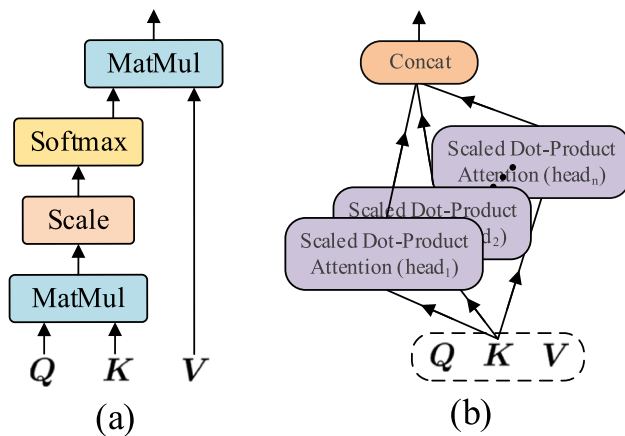
In terms of common sense, deep learning models contain a large number of nonlinear activation functions. It makes them able to deal with sophisticated tasks. However, the large superposition of nonlinear transformations results in models that often struggle to capture linear dependencies in time series. MCTNet has been designed with the idea of parallel architecture with both nonlinear and linear extractors, which makes the different dependencies not discarded by the model as shown in Fig. 1. It provides a brief MCTNet architecture, including both ME and MF in parallel. MCTNet profits from highly parallelized computations of the transformer encoder [25] with much lower run-time overhead, which is unlike the autoregressive generation method [21] with turgid time complexity.

Since ME has adopted the transformer encoder [25] to capture the information in different time steps globally by the self-attention mechanism, the problem of long-term dependencies ambiguity can be solved. MF has been used to preserve linear dependencies from the time series data, which is different from the ME block used to catch nonlinear dependencies from the data. A highway network in MF has been applied to add linear elements from input time series data for the final prediction. A spatial attention mechanism has been added to enhance the relationship among different variables. MCTNet can efficiently perform time series prediction on different timescales by combining ME and MF together.

The multi-step prediction result of MCTNet can be defined as:

$$\mathbf{Y} = \mathbf{L}_o + \mathbf{G}_o + \mathbf{H}_o \quad (1)$$

where  $\mathbf{Y} \in \mathbf{R}^{N \times F}$  is the output of the MCTNet,  $N$  stands for the length of the predict output sequence, and  $F$  stands for



**Fig. 2** **a** Scaled dot-product attention, **b** Multi-head attention

the dimension of the predict output sequence. The sum of  $L_o \in \mathbf{R}^{N \times F}$  and  $G_o \in \mathbf{R}^{N \times F}$  represents the output of the block, and  $H_o$  is the output of MF block.

## 2.1 Multiscale extraction mechanism

A novel feature extraction structure has been designed to mine the nonlinearity in multiscale time series data. It combines a transformer encoder [25] with a multiscale feature fusion block. At first, the ME has been used to extract different ranges of dependencies from the input sequence. The transformer encoder [25] has been selected to further process some high-dimensional features.

### 2.1.1 Transformer

The canonical transformer model utilized an encoder–decoder framework as seq2seq model [28]. The encoder is made up of encoding layers that process the input recursively, while the decoder is made up of decoding layers that process the encoder’s output recursively. Both the encoder and decoder process input tensors layer by layer.

A transformer-based natural language processing model has been proposed inspired by BERT [29], that only uses the encoder to learn the representations from a bulk of natural language datasets. We chose the transformer’s encoder as the backbone of MCTNet for time series dependency extraction. As a result, the computational cost is reduced, and the results of multiple time step predictions can be obtained by only one inference.

The self-attention mechanism serves as the encoder’s central component. Each self-attention block is composed of multiple scaled dot-product attention mechanism, as illustrated in Fig. 2a [25].

However, when encoding information at the current position by single scaled dot-product attention, the model will

excessively focus on its own position and ignore other positions’ information. Therefore, the model with multi-head attention jointly attends to information from several representation subspaces at different locations, as shown in Fig. 2b. Meanwhile, the multi-head self-attention mechanism supports parallel computing.

A multi-head attention process is defined as follows:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^o \quad (2)$$

where  $\mathbf{Q}, \mathbf{K}, \mathbf{V}$  denote query, key, and value matrix, respectively. Each of these matrices represents a high-dimensional input embedding.  $\mathbf{W}^o \in \mathbf{R}^{d \times d}$  is the self-attention output matrix used as a linear transformation after concatenating the individual heads. And  $d$  represents the embedding dimension of the input sequence.

The attention score of each head is defined as:

$$\text{head}_i = \text{Attention}(\mathbf{Q} \mathbf{W}_i^q, \mathbf{K} \mathbf{W}_i^k, \mathbf{V} \mathbf{W}_i^v) \quad (3)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d}}\right) \mathbf{V} \quad (4)$$

where  $\mathbf{W}_i^q \in \mathbf{R}^{d \times d}$ ,  $\mathbf{W}_i^k \in \mathbf{R}^{d \times d}$ ,  $\mathbf{W}_i^v \in \mathbf{R}^{d \times d}$  are the parameter matrices that can be learned to map  $\mathbf{Q}, \mathbf{K}, \mathbf{V}$  into the high-dimensional representation space, respectively. And the square root of  $d$  is used to linearly scale to ensure the stable propagation of the gradient.

The softmax nonlinear function is defined as:

$$\text{softmax}(x_m) = \frac{e^{x_m}}{\sum_n e^{x_n}}, \quad (5)$$

Due to the over-the-horizon advantage of transformer [25], correlation links for all time steps can be established without the restriction of distance. However, it also makes itself unable to learn causal temporal trends. Therefore, the positional embedding is added to capture temporal information for the model.

### 2.1.2 Local and global dependency extraction

Time series context dependency at different scales is typically presented in distinct time series datasets. A novel multiscale sequence dependency extraction strategy has been proposed to improve the generalization performance of the model with different dependency scales. The structure of ME is shown in Fig. 3a.

The ME module consists of a local dependency extraction convolution kernel and a global dependency preadaptive extraction convolution kernel. Local convolution kernel and global preadaptive convolution kernel are used to capture

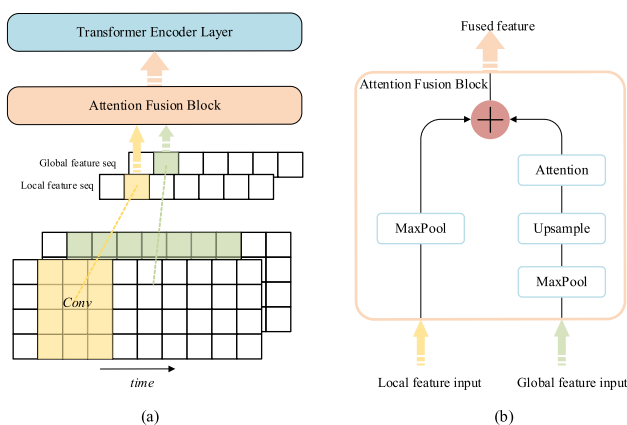


Fig. 3 a The structure of ME, b Local and global feature attention fusion

short-term and long-term dependency, respectively. The size of each convolutional filter is not identical. Their parameter weights are not shared. The results of these two kernels will be fused by the attention fusion shown in Fig. 3b.

The local feature extractor uses convolutional kernel of moderate size and short stride, which are sufficient to collect comprehensive time series data to mine dense context dependencies. On the contrary, the global feature preadaptive extractor has a wide receptive field, which takes advantage of the sparse convolutional kernel to acquire latent long-term dependencies. The filter size and slide stride of the global feature preadaptive extractor are more than twice those of the local feature extractor. The input series are encoded as high-dimensional global features after being distilled by the global feature preadaptive extractor. Then, maxpool is used to keep the local and global feature lengths consistent. In order to enhance the strong correlation and attenuate the weak correlation within different periods, the global feature preadaptive extractor utilizes a temporal attention mechanism to constrain feature expression. After the local extraction outputs are superimposed on the global extraction outputs, the ME block finally obtains dependency at different timescales.

The output of local convolution kernel is defined as follows:

$$L_c = W_l \cdot X + b_l, \tag{6}$$

where  $L_c \in \mathbf{R}^{d^*f\_len}$  is the output of local convolution kernel,  $X \in \mathbf{R}^{d^*f\_len}$ ,  $f\_len$  is the temporal length of the input sequence,  $W_l$  is a parameter matrix of local convolutional kernel, and  $b_l$  is a bias vector.

The output of local feature extractor process is described as follows:

$$L_o = \text{maxpool}(L_c), \tag{7}$$

where  $L_o \in \mathbf{R}^{d^*(f\_len/2)}$  is local output, and the maxpool operation makes the  $L_o$  become half of  $L_c$  in the temporal dimension.

The global feature preadaptive extractor process is described as follows:

$$G_c = W_g \cdot X + b_g, \tag{8}$$

$$G'_o = \text{maxpool}(G_c), \tag{9}$$

where  $G_c \in \mathbf{R}^{d^*(f\_len/4)}$  is the output of global preadaptive convolution kernel,  $G'_o \in \mathbf{R}^{d^*f\_len}$  is defined as the transitory globally output,  $W_g$  is a parameter matrix of global convolutional filter, and  $b_g$  is a bias.

The temporal attention mechanism in Fig. 3b is described as follows:

$$\alpha = \tanh(\text{upsample}(G'_o)), \tag{10}$$

$$S_t = \text{softmax}(\alpha \cdot W_a + b_a), \tag{11}$$

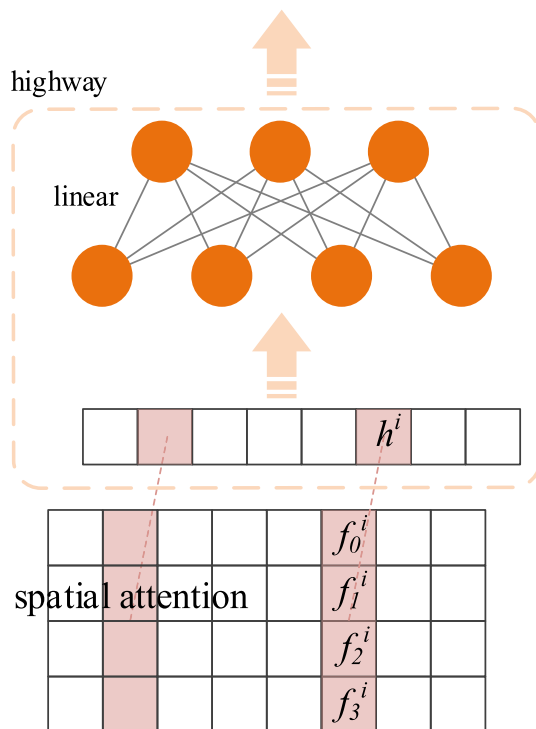
$$G_o = G'_o \odot S_t, \tag{12}$$

where the matrix multiplication result from two matrices  $W_o$  and  $X$  is not identical in shape to the first one, so it is upsampled to ensure that both have the same dimension,  $\alpha$  is the transitory output.  $S_t$  stands for attention scores at different time steps.  $W_a$  is a learnable parameter matrix,  $b_a$  is preadaptive output, which is the result of Hadamard product  $G'_o$  with  $S_t$ .

The results obtained by the above process are inputted into the transformer encoder [25] layer to further mine the nonlinear correlation in the time series for the final prediction.

### 2.2 Multidimensional feature fusion mechanism

While ME has advantages in extracting nonlinear dependencies in sequences, it ignores some insignificant linear information. MF mechanism has been proposed inspired by [30, 31], which combines highway mechanism [31] with spatial attention to enhance the representation of linear features. Similar to the temporal attention mechanism of the ME component, the calculation is the same. However, while the former adaptively weights different time steps in the temporal dimension, the latter adaptively weights the importance of input multivariate covariates in the feature dimension. The weighted high-dimensional vectors are compressed into a single dimension by adding them. To preserve the linear element of the representations to a great extent, only linear layers without nonlinear activation functions are used to transform the representations. The output of MF is as shown in Fig. 4.



**Fig. 4** The structure of MF

Take Fig. 4 as an example, the feature dimension of the input sequence is 4. It means that there are four separated feature sequences as input. Define the input vector at the  $i$ th time step as  $\mathbf{F}^i = [f_0^i, f_1^i, f_2^i, f_3^i]$ . The spatial attention process is described as follows:

$$\mathbf{s} = \tanh(\mathbf{F}_i), \quad (13)$$

$$\mathbf{A}^i = \text{softmax}(\mathbf{s} \cdot \mathbf{W}_f + \mathbf{b}_f), \quad (14)$$

$$h^i = \mathbf{F}^i \cdot \mathbf{A}^i, \quad (15)$$

where  $\mathbf{s}$  is the transitory output,  $\mathbf{A}^i \in \mathbf{R}^{4 \times 1}$  stands for the attention score of  $\mathbf{F}^i$ ,  $\mathbf{W}_f$  is a learnable parameter matrix,  $\mathbf{b}_f$  is an attention bias vector, and  $h^i$  is the output representation at  $i$ th time step.

### 3 Experimental analyses and discusses

In order to verify the effectiveness of the MCTNet, experiments have been done on some challenging time series datasets with different dependency scales and covariate dimensions. The experiment is conducted in a server with Ubuntu 20.04 LTS operating system, an Intel(R) i7-7700 CPU, a Nvidia(R) Geforce RTX 4070ti GPU, and PyTorch 1.11.0.

### 3.1 Datasets

**Electricity Transformer Temperature [12] (ETT)** contains 1 h-level and 15 min-level electrical transformer data. The ETT [12] dataset contains seven attributes with six power load features and an oil temperature as the prediction target. The variation trend of electrical load often shows a certain recent dependency, it has a strong correlation with recent historical data. Electrical data tests the ability of the model to extract short-term dependencies.

**Beijing Multi-Site Air-Quality Dataset [32] (AQI)** contains hourly air pollution data with 18 attributes and 420,768 time steps. It reflects seasonal trends to some extent. The variation tendency of air quality has an obvious periodicity, and the change of season will affect the weather and air quality to a large extent. At the same time, the temperature and humidity show significant periodicity in the short term on the scale of days. Besides, the proportion of missing values in the AQI dataset [32] is as high as 25.67%. Such predictions with strong long short-term correlations are a test for the model.

**Grottoes Physics Properties (Grottoes)** is another dataset that is significantly different from the above ones [12, 32]. It is collected from the Yungang Grottoes in China, UNESCO World Heritage site. The Grottoes dataset contains six attributes including mass, rock stress, chromatic aberration, ultrasonic conduction velocity, hardness, and magnetic susceptibility with 14,400 time steps. The changes in the physical properties of the Grottoes are accumulated over a long period of time, which is observably different from the previous two datasets [12, 32]. The predicted values strongly correlate with extremely long-term historical data. It requires the model to capture extremely long-time dependencies.

### 3.2 Metrics and parameters discusses

In order to get a fair comparison with some state-of-the-art, the first 128 time steps of each sequence at the mentioned above datasets are used to forecast the next 16 time steps. Mean square error (MSE), mean absolute error (MAE), and Pearson's correlation coefficient (CORR) are selected to evaluate the models. A stronger prediction performance is indicated by a higher CORR as well as lower both MSE and MAE values.

$$\text{MSE} = \frac{1}{n} \sum_{n}^{i=1} (y_i - \hat{y}_i)^2, \quad (16)$$

$$\text{MAE} = \frac{1}{n} \sum_{n}^{i=1} |y_i - \hat{y}_i|, \quad (17)$$

$$\text{CORR} = \frac{\sum_{n}^{i=1} (y_i - \bar{Y})(\hat{y}_i - \bar{\hat{Y}})}{\sqrt{\sum_{n}^{i=1} (y_i - \bar{Y})^2} \sqrt{\sum_{n}^{i=1} (\hat{y}_i - \bar{\hat{Y}})^2}}, \quad (18)$$

where  $y_i$  is the output value of the model,  $\hat{y}_i$  is the target value of time step  $i$ , and  $n$  is the quantity of samples in the sequence. In  $F$ ,  $\widehat{Y}$  and  $\widetilde{Y}$  stand for the mean of the predicted sequence and the mean of the target sequence.

The proportional segmentation method is applied to generate the train dataset and test dataset, to ensure their causality. The first 80% of the time series is used as the train dataset, and the last 20% is used as the test dataset to evaluate the performance of different models. The learning rate and batch size are set to 0.001 and 512, respectively. The model has undergone 5000 training iterations.

### 3.3 Prediction performance analyses in different feature dimensions

Low embedding dimension has superior generalization performance for reasoning the test dataset though, it will lead to poor performance for learning nonlinear patterns from complex massive data. Conversely, both generalization and learning performances exhibit an opposite trend as the dimension increases. In order to get a reasonable embedding dimension for transformer encoder layer  $d$  in (1),  $d$  changes from 64 to 384 at interval 64. Some experimental results for different  $d$ s are given in Figs. 5, 6, 7, respectively.

One can find from Figs. 5, 6, 7 that the prediction performances are the best at the chosen datasets [12, 32], and the Grottoes dataset when the embedding dimension is 128 as a whole. The model does not achieve optimal performance when  $d$  is 64. When  $d$  increases to 128, the prediction performance on each dataset tends to be optimal. As the value of  $d$  increases, the model overfits on each dataset, and the prediction performance deteriorates sharply. It can be seen that the performance level of the same indicator on different datasets is not completely consistent, even under the condition of keeping the embedding dimension constant. The reason is that the data distribution and feature correlation of different datasets are inconsistent, which leads to uncertainty in performance. The embedding dimension  $d$  in (1) is set to 128 and kept the subsequent experiments.

### 3.4 Transformer self-attention analyses

To investigate the contribution of different locations in the sequence contribute to the final prediction, the heatmaps of the self-attention scores for different time steps are shown in Fig. 8.

The horizontal and vertical axes represent keys and queries at different time steps in Fig. 8, respectively. The shade of color represents the attention score, that is, the contribution of different time steps to the final prediction. Locus with lighter colors represents higher contributions, and vice versa.

It shows the self-attention scores on the ETT [12], AQI [32], and Grottoes datasets from left to right, respectively.

The highlights are mainly distributed in the interval 112–128 for ETT [12] dataset. It indicates that the prediction of its trend depends on recent data. However, the model is not interested in the short-term data for AQI [32] dataset. On the contrary, the model prefers the relatively long-term historical data, and its brightness distribution shows a certain period of superposition. The contribution degree of different time steps increases with time for the Grottoes dataset with long-time dependencies. It indicates that the weight of each time step requires to be balanced in the model.

Among them, the prediction of MCTNet for ETT [12] dataset is strongly correlated with the recent historical data, while the contribution of the distant historical data is weak. The MF compensates for the historical data over a period of time in the prediction output. This enables the model to learn the trend of data change in the recent period. The model is more dependent on distant historical data for AQI [32] dataset. The importance degree also reflects a certain periodicity from Fig. 8. Multiscale convolution filters and attention fusion in ME enable MCTNet to mine the correlation of its different periods. It improves the prediction's effectiveness. For the Grottoes dataset with a slow trend, the scores of different time steps also exhibit a smooth change. The self-attention mechanism in MCTNet captures the dependencies of long-term with adaptive weighting for different time steps.

### 3.5 Ablation study

To further test the performance of MCTNet, ablation experiments have been done on the ETT [12] and AQI [32] datasets, respectively. Some experimental results are shown in Table 1. The optimal results in Table 1 are highlighted in boldface.

One can find from Table 1 that our proposed model and mechanism including ME-Conv, ME-Att, MF, and MCTNet help to improve the prediction performance, and MCTNet has the best prediction performance. Some reasons are as follows: ME preadaptive feature extractor is mainly composed of a multiscale convolution filter and an attention fusion block. ME degrades to a naive linear encoder after removing the convolution filters, namely ME-Conv. ME degenerates to a simple local information extractor after removing multiscale convolution filter and temporal attention-based fusion, namely ME-Att. MF is composed of a spatial attention-based highway network. The introduction of the MF settles the problem of missing linear elements and strengthens the extraction of short-term information. Both ME and MF leverage multiscale and multidimensional information to make more precise predictions. To verify that maxpool operation in the attention fusion block has optimal performance, maxpool has been replaced by avgpool and power-average pool for ablation study, corresponding to '(w/) AvgPool' and '(w/)

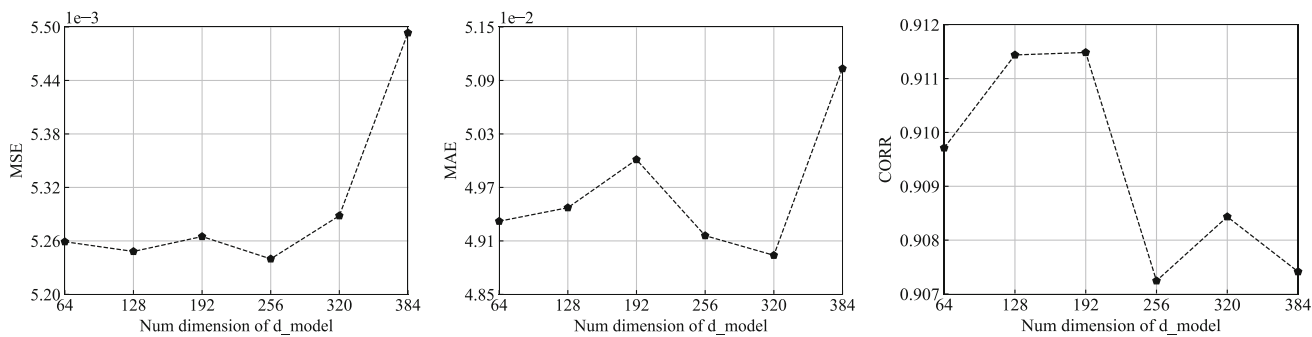


Fig. 5 Prediction performances in different  $d_s$  at the ETT dataset [12]

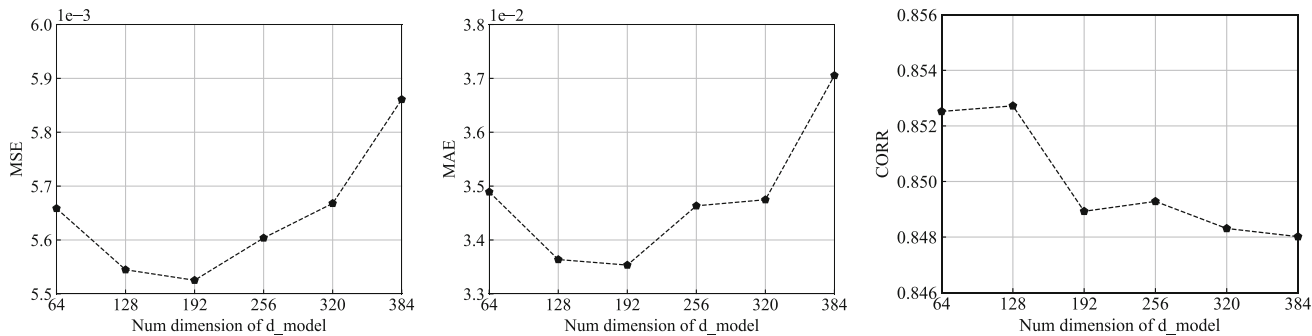


Fig. 6 Prediction performances in different  $d_s$  at the AQI dataset [32]

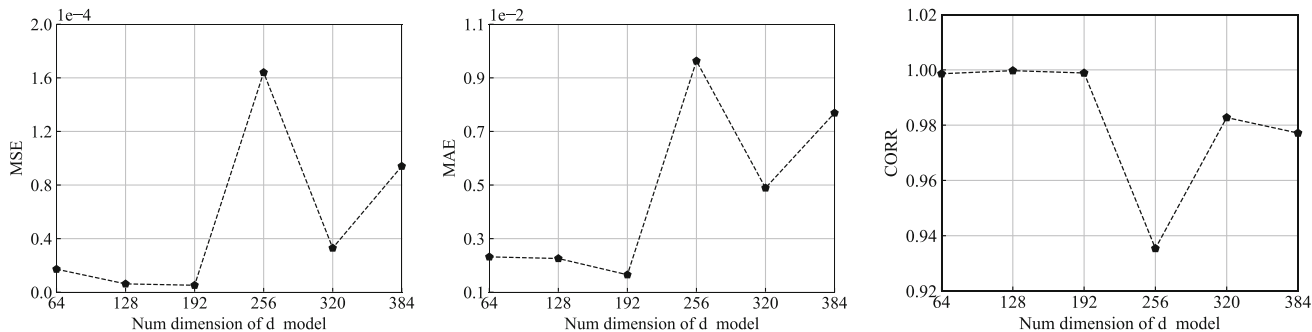


Fig. 7 Prediction performances in different  $d_s$  at the Grottoes dataset

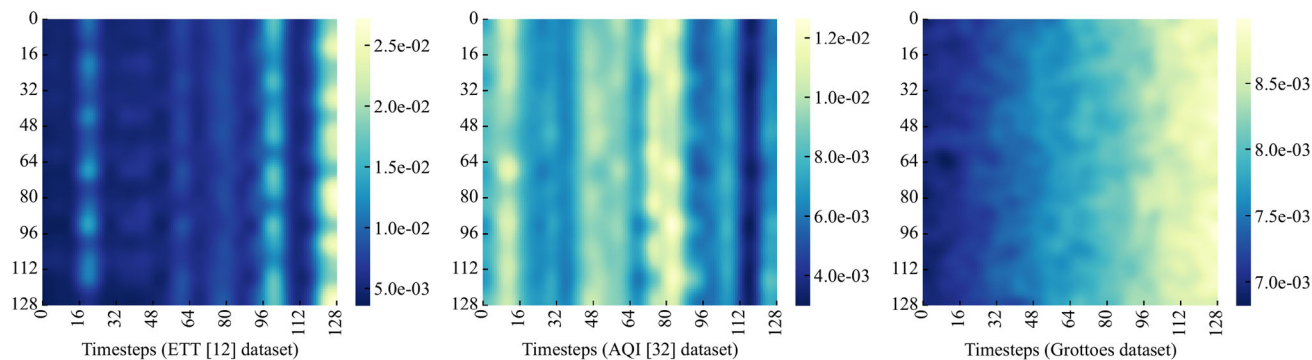


Fig. 8 Self-attention score heatmaps at disparate datasets

**Table 1** Ablation experimental results with different models at different datasets

Method	ETT [12]			AQI [32]		
	MSE	MAE	CORR	MSE	MAE	CORR
<b>MCTNet</b>	<b>0.374</b>	<b>0.458</b>	<b>0.715</b>	<b>0.346</b>	<b>0.319</b>	<b>0.832</b>
(w/o) ME-Conv	0.418	0.493	0.711	0.466	0.397	0.807
(w/o) ME-Att	0.392	0.472	0.709	0.428	0.371	0.788
(w/o) MF	0.436	0.501	0.702	0.401	0.352	0.806
(w/) AvgPool	0.384	0.469	0.711	0.359	0.328	0.817
(w/) PAPool	0.396	0.472	0.706	0.377	0.339	0.812

PAPool' in Table 1, respectively. Experimental results show that the performance of MCTNet is worse than that of max-pool when avgpool is used. As a result, every portion of the suggested model contributes in a specific way to overall performance.

### 3.6 Comparisons with some state-of-the-art methods

To further evaluate the prediction performance for MCTNet, some state-of-the-art methods have been chosen for a fair comparison across different datasets including LSTM [33], Att-BLSTM [34], Transformer [35], and ConvTrans [27]. To get a fair comparison, all the corresponding parameters used are the authors' recommended ones for each method. The experiment has been done with six different prediction lengths of data with an input time series length of 128 at ETT [12], AQI [32], and the Grottoes datasets. Some prediction performances at different lengths are shown in Figs. 9, 10, and 11, respectively.

It can be found that MCTNet has the best performance among the investigated methods [27, 33–35] from Figs. 9, 10, 11, respectively. MCTNet has the lowest MSE, MAE, and the highest CORR in different lengths as a whole. To further demonstrate the MCTNet has the best prediction performance, we calculate the average MSE, MAE, and CORR for six different prediction lengths from Figs. 9, 10, 11. Some comparative results are given in Table 2. The optimal results in Table 2 are highlighted in boldface. One can find from Table 2 that MCTNet has the best prediction performance among the investigated approaches [27, 33–35]. Some reasons are as follows: Sequential dependency extraction is achieved by a causal approach in LSTM [33] and Att-BLSTM [34]. The pattern learned at the current time step must depend on the previous historical time steps, which makes the prediction error steadily accumulate. As the length of the sequence increases, the results predicted by LSTM [33] and Att-BLSTM [34] are more and more deviated from the actual target. It can be shown in Figs. 9, 10, 11, respectively. Transformer [35] and ConvTrans [27] can ignore the distance information in space and overcome the problem of error

accumulation. Transformer [35] and ConvTrans [27] have superior prediction performance for different time-dependent tasks. However, Transformer [35] and ConvTrans [27] cannot mine efficient representations from sequences with different timescales and dimensions. MCTNet takes advantage of information at different timescales, which can efficiently predict outcomes by accounting for covariate relationships.

To verify the robustness of the MCTNet, we conducted a variety of experiments with different inputs and different output lengths. The results are shown in Table 3. Among them, the values of the length index: '16 ≥ 8,' '32 ≥ 16,' '64 ≥ 32' represent the input length of 16, 32, 64 and the output length of 8, 16, 32, respectively. The results show that MCTNet can maintain great prediction performance even when the length of input and output of time series data changes. It is proved that the model has optimal robustness.

Figure 12 shows the prediction curves of multiple methods [27, 34, 35] on the ETT dataset. The average deviation of MCTNet in the whole prediction time window is smaller than that of other models. Profit from the over-the-horizon ability of the self-attention architecture, MCTNet can still maintain stable prediction results with the growth of the prediction time step. The experimental results from Tables 2 and 3 further show that MCTNet outperforms some state-of-the-art methods by comparisons on different challenging datasets.

## 4 Conclusions

A novel method has been developed for time series prediction in MCTNet. It is composed of ME and MF frameworks. The ME framework contains both a multiscale convolutional feature extractor and a temporal attention-based representator. It can be used to mine different time-scale dependencies. The MF component simultaneously mines the correlation between different covariates to improve the model's capacity for linear expression. The relationship among inputs can be captured by utilizing a spatial attention-based highway mechanism in MF. Experimental results demonstrate that MCTNet has outstanding performance for time series prediction in comparison with some state-of-the-art methods at challenging datasets.



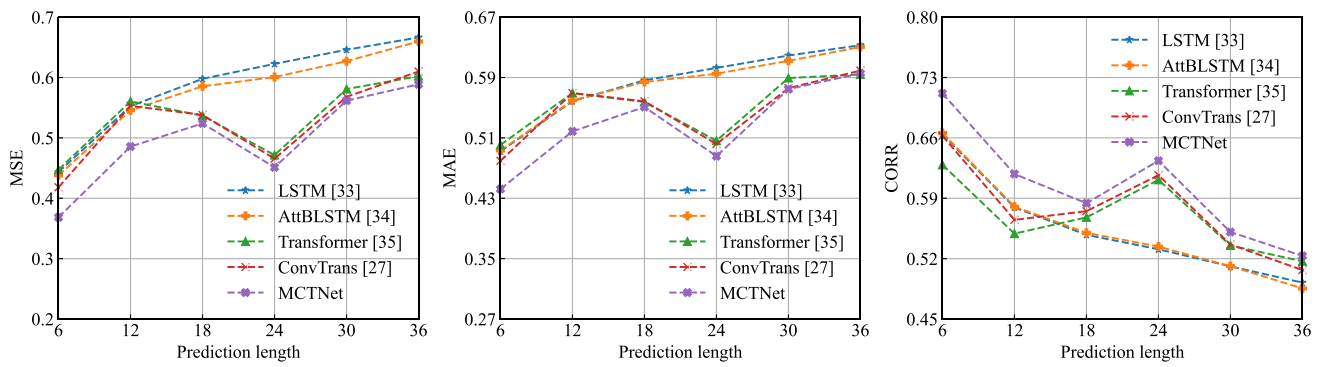


Fig. 9 Prediction performance in different lengths at the ETT dataset [12]

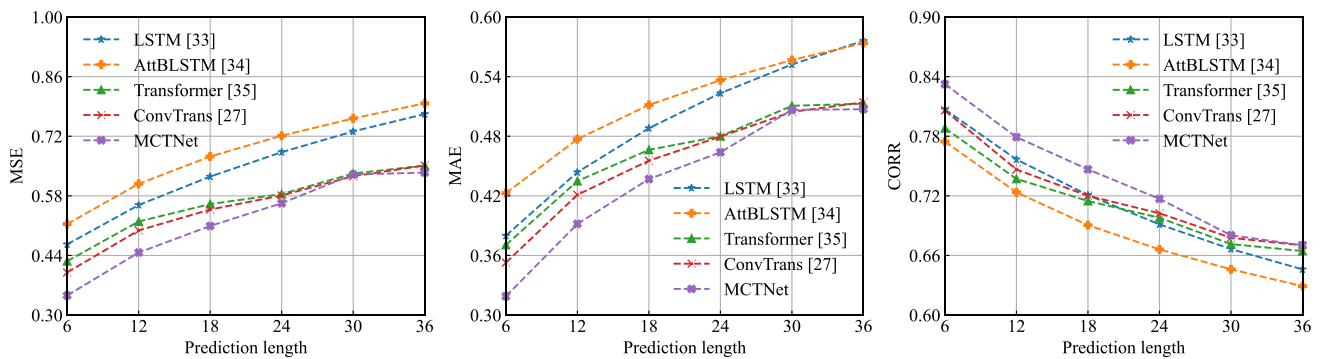


Fig. 10 Prediction performance in different lengths at the AQI dataset [32]

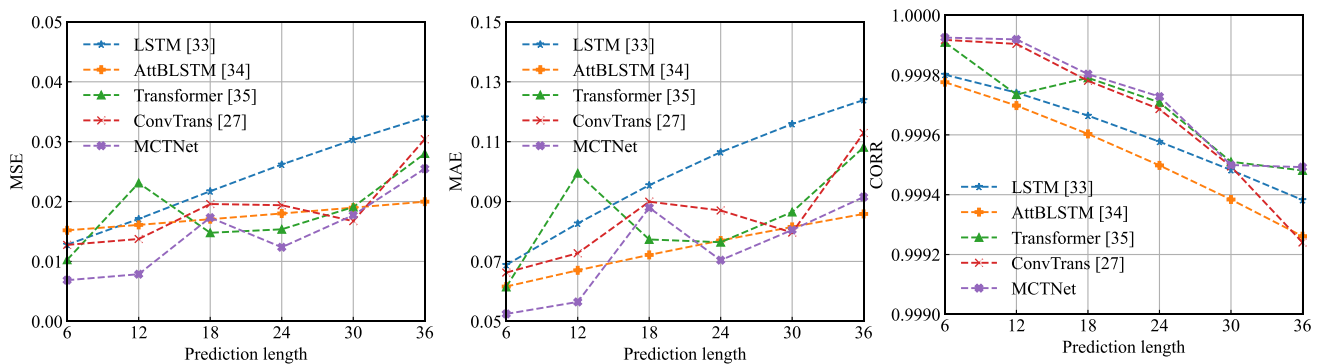


Fig. 11 Prediction performance in different lengths at the Grottoes dataset

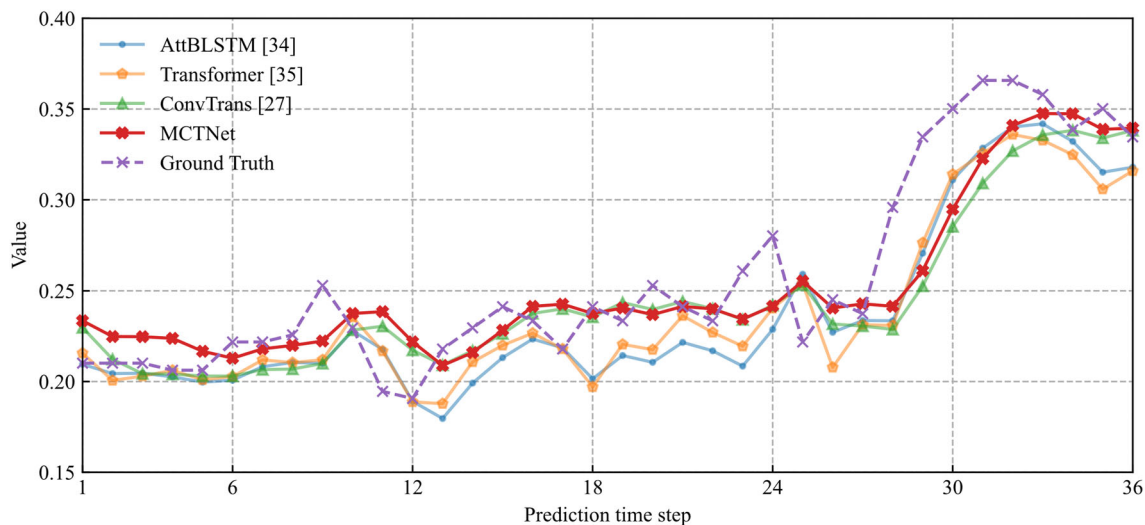
Table 2 Some comparative results in different methods

Method	ETT [12]			AQI [32]			Grottoes ( $\times 10^{-1}$ )		
	MSE	MAE	CORR	MSE	MAE	CORR	MSE	MAE	CORR
LSTM [33]	0.588	0.582	0.554	0.639	0.494	0.715	0.236	0.988	9.996
AttBLSTM [34]	0.576	0.579	0.554	0.679	0.513	0.688	0.175	0.742	9.995
Transformer [35]	0.533	0.553	0.568	0.562	0.463	0.712	0.184	0.848	9.996
ConvTrans [27]	0.525	0.547	0.577	0.551	0.455	0.720	0.187	0.847	9.996
<b>MCTNet (ours)</b>	<b>0.496</b>	<b>0.528</b>	<b>0.604</b>	<b>0.522</b>	<b>0.438</b>	<b>0.738</b>	<b>0.146</b>	<b>0.732</b>	<b>9.997</b>

**Table 3** Some comparative results with different methods in multiple sequence lengths

Length	Method	ETT [12]			AQI [32]			Grottoes ( $\times 10^{-1}$ )		
		MSE	MAE	CORR	MSE	MAE	CORR	MSE	MAE	CORR
$16 \geq 8$	LSTM [33]	0.446	0.491	0.662	0.468	0.382	0.807	0.128	0.692	9.998
	AttBLSTM [34]	0.439	0.488	0.664	0.516	0.425	0.769	0.154	0.634	9.997
	Transformer [35]	0.452	0.502	0.626	0.426	0.373	0.788	0.107	0.628	<b>9.999</b>
	ConvTrans [27]	0.421	0.483	0.673	0.408	0.362	0.806	0.131	0.661	<b>9.999</b>
	<b>MCTNet (ours)</b>	<b>0.374</b>	<b>0.445</b>	<b>0.706</b>	<b>0.358</b>	<b>0.322</b>	<b>0.829</b>	<b>0.072</b>	<b>0.537</b>	<b>9.999</b>
$32 \geq 16$	LSTM [33]	0.646	0.623	0.511	0.627	0.493	0.752	0.219	0.953	9.996
	AttBLSTM [34]	0.629	0.615	0.511	0.676	0.517	0.718	0.173	<b>0.733</b>	9.996
	Transformer [35]	0.581	0.592	0.535	0.563	0.472	0.733	<b>0.156</b>	0.776	9.997
	ConvTrans [27]	0.572	<b>0.575</b>	0.534	0.556	0.461	0.743	0.194	0.899	9.997
	<b>MCTNet (ours)</b>	<b>0.567</b>	0.578	<b>0.551</b>	<b>0.519</b>	<b>0.442</b>	<b>0.774</b>	0.178	0.883	<b>9.998</b>
$64 \geq 32$	LSTM [33]	0.657	0.628	0.511	0.742	0.562	0.663	0.298	1.117	9.991
	AttBLSTM [34]	0.648	0.621	0.511	0.770	0.569	0.642	0.208	0.816	9.993
	Transformer [35]	0.598	0.597	0.532	0.651	0.524	0.672	0.192	0.868	<b>9.995</b>
	ConvTrans [27]	0.585	0.586	0.535	<b>0.644</b>	<b>0.518</b>	0.675	0.177	0.796	9.994
	<b>MCTNet (ours)</b>	<b>0.579</b>	<b>0.583</b>	<b>0.547</b>	0.648	0.521	<b>0.679</b>	<b>0.183</b>	<b>0.808</b>	9.994

Bold values are represent the optimal predict results, at different “length” experimental parameter settings

**Fig. 12** Predict curves of multiple methods

In future, we will expand the covariate prediction dimension so that we can handle time series data with a large number of input dimensions. At the same time, the computational structure is optimized to ensure low computational overhead even in the case of sharp data growth.

**Author contributions** ZW presented the innovation of paper, designed and carried out the experiments, and analyzed the result of the experiments. YG drafted the work or revised it critically for important intellectual content. All authors reviewed the manuscript.

**Funding** This work is supported in part by National Key R&D program of China (Grant No. 2020YFC1523004).

**Data availability** The ETT and AQI datasets used in this paper are publicly available. The ETT and AQI datasets can be acquired from the following links. All data used in this paper, including images and codes, are available by contacting the corresponding author by reasonable request. **ETT:** <https://opendatalab.com/ETT> **AQI:** <https://www.kaggle.com/datasets/fedesoriano/air-quality-data-set>

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This paper does not contain any studies involving humans or animals. This declaration is not applicable.

## References

- Zouaidia, K., Rais, S., Ghanemi, S.: Weather forecasting based on hybrid decomposition methods and adaptive deep learning strategy. *Neural Comput. Appl.* **35**(1), 11109–11124 (2023)
- Mosavi, A., Ozturk, P., Chau, K.: Flood prediction using machine learning models: Literature review. *Water* **10**(11), 1536–1576 (2018)
- Liu, R., Zhou, H., Li, D., et al.: Evaluation of artificial precipitation enhancement using UNET-GRU algorithm for rainfall estimation [J]. *Water* **15**(8), 1585–1602 (2023)
- Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **270**(2), 654–669 (2018)
- Kim, Y., Won, H.: Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **103**(1), 25–37 (2018)
- Polson, G., Sokolov, O.: Deep learning for short-term traffic flow prediction [J]. *Transport. Res. Part C: Emerg. Technol.* **79**(1), 1–17 (2017)
- Yuan, Q., Shen, H., Li, T., et al.: Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sens. Environ.* **241**(1), 111716–111736 (2020)
- Yuan, X., Li, L., Yang, C., et al.: Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. *IEEE Trans. Industr. Electron.* **68**(5), 4404–4414 (2020)
- Yuan, X., Li, L., Yang, C., et al.: Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Trans. Industr. Inf.* **16**(5), 3168–3176 (2019)
- Liu, L., Hsaio, H., Yao, Tu., et al.: Time series classification with multivariate convolutional neural network. *IEEE Trans. Ind. Electr.* **66**(6), 4788–4797 (2018)
- Lacasa, L., Nicosia, V., Latora, V.: Network structure of multivariate time series. *Sci. Rep.* **5**(1), 1–9 (2015)
- Zhou, H., Zhang, S., Peng, J., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: AAAI Conference on Artificial Intelligence, pp. 11106–11115 (2021)
- Das, M., Ghosh, K.: Data-driven approaches for meteorological time series prediction: a comparative study of the state-of-the-art computational intelligence techniques. *Patt. Recogn. Lett.* **105**(1), 155–164 (2018)
- Johannesson, P., Podgórski, K., Rychilk, I., et al.: AR: time series with autoregressive gamma variance for road topography modeling. *Probab. Eng. Mech.* **43**(1), 106–116 (2016)
- Schaffer, L., Dobbins, A., Pearson, S.: Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC Med. Res. Methodol.* **21**(1), 1–12 (2021)
- Li, Q., Lin, C.: A new approach for chaotic time series prediction using recurrent neural network. *Math. Probl. Eng.* **2016**(1), 1–9 (2016)
- Chen, X., Chen, X., She, J., et al.: A hybrid time series prediction model based on recurrent neural network and double joint linear–nonlinear extreme learning network for prediction of carbon efficiency in iron ore sintering process. *Neurocomputing* **249**(1), 128–139 (2017)
- Zheng, W., Chen, G.: An accurate GRU-based power time-series prediction approach with selective state updating and stochastic optimization. *IEEE Trans. Cybernet.* **52**(12), 13902–13914 (2021)
- Karevan, Z., Suykens, A.: Transductive LSTM for time-series prediction: an application to weather forecasting. *Neural Netw.* **125**(1), 1–9 (2020)
- Dey, R., Salem, M.: Gate-variants of gated recurrent unit (GRU) neural networks. In: International Midwest Symposium on Circuits and Systems, pp. 1597–1600 (2017)
- Yu, Y., Si, X., Hu, C., et al.: A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **31**(7), 1235–1270 (2019)
- Hwang, S., Jeon, G., Jeong, J., et al.: A novel time series based Seq2Seq model for temperature prediction in firing furnace process. *Procedia Comput. Sci.* **155**(1), 19–26 (2019)
- Wang, X., Cai, Z., Wen, Z., et al.: Long time series deep forecasting with multiscale feature extraction and Seq2seq attention mechanism. *Neural Process. Lett.* **54**(4), 3443–3466 (2022)
- Chen, H., Zhang, X.: Path planning for intelligent vehicle collision avoidance of dynamic pedestrian using Att-LSTM, MSFM, and MPC at unsignalized crosswalk. *IEEE Trans. Industr. Electron.* **69**(4), 4285–4295 (2021)
- Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 1–11 (2017)
- Zerveas, G., Jayaraman, S., Patel, D., et al.: A transformer-based framework for multivariate time series representation learning. In: Knowledge Discovery & Data Mining, pp. 2114–2124 (2021)
- Li, S., Jin, X., Yao, X., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Advances in Neural Information Processing Systems, pp. 1–14 (2019)
- Li, Z., Cai, J., He, S., et al.: Seq2seq dependency parsing. In: Computational Linguistics, pp. 3203–3214 (2018)
- Sun, F., Liu, J., Wu, J., et al.: BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Information and Knowledge Management, pp. 1441–1450 (2019)
- Geng, Z., Chen, Z., Meng, Q., et al.: Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes. *IEEE Trans. Industr. Inf.* **18**(3), 1521–1529 (2021)
- Lai, G., Chang, C., Yang, Y., et al.: Modeling long-and short-term temporal patterns with deep neural networks. In: Research & Development in Information Retrieval, pp. 95–104 (2018)
- Vito, S., Massera, E., Piga, M., et al.: On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sens. Actuators, B Chem.* **129**(2), 750–757 (2008)
- Afrin, T., Nita, Y.: A long short-term memory-based correlated traffic data prediction framework. *Knowl.-Based Syst.* **237**(1), 775–786 (2022)
- Zhang, J., Hao, K., Tang, X., et al.: A multi-feature fusion model for Chinese relation extraction with entity sense. *Knowl.-Based Syst.* **206**(1), 106348–106358 (2020)
- Tang, B., Matteson, S.: Probabilistic transformer for time series analysis. In: Advances in Neural Information Processing Systems, pp. 23592–23608 (2021)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.