Time-Series Forecasting Based on Multi-Layer Attention Architecture

Na Wang[1,2*] and Xianglian Zhao[2]
[1] Department of Accounting and Audit, Nanjing Audit University Jinshen College, Nanjing 210046, China
[2] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[e-mail: wna1982@163.com]
*Corresponding author: Na Wang

Received June 18, 2022; revised August 14, 2023; accepted October 9, 2023; published January 31, 2024

Abstract

Time-series forecasting is extensively used in the actual world. Recent research has shown that Transformers with a self-attention mechanism at their core exhibit better performance when dealing with such problems. However, most of the existing Transformer models used for time series prediction use the traditional encoder-decoder architecture, which is complex and leads to low model processing efficiency, thus limiting the ability to mine deep time dependencies by increasing model depth. Secondly, the secondary computational complexity of the self-attention mechanism also increases computational overhead and reduces processing efficiency. To address these issues, the paper designs an efficient multi-layer attention-based time-series forecasting model. This model has the following characteristics: (i) It abandons the traditional encoder-decoder based Transformer architecture and constructs a time series prediction model based on multi-layer attention mechanism, improving the model's ability to mine deep time dependencies. (ii) A cross attention module based on cross attention mechanism was designed to enhance information exchange between historical and predictive sequences. (iii) Applying a recently proposed sparse attention mechanism to our model reduces computational overhead and improves processing efficiency. Experiments on multiple datasets have shown that our model can significantly increase the performance of current advanced Transformer methods in time series forecasting, including LogTrans, Reformer, and Informer.

Keywords: Time-Series Forecasting, Attention, Cross-attention, Deep neural network

1. Introduction

Time-series forecasting has been extensively used in stock forecasting, transportation, et al. In these real-world uses, an urgent problem to be solved is to increase the forecast time, which is conducive to long-term planning and decision-making. With the arrival of big data era, the time-series forecasting model begins to face the scenario of predicting a longer time span. Therefore, for each rolling window, the model should also be able to process more past information. Because the specific structure of classical time-series forecasting algorithms such as ARIMA [1,2] and SSM [3] needs to be manually selected after considering various relevant factors, its prediction ability is insufficient and cannot meet the needs of long-term prediction.

The model based on deep learning is a good candidate model to solve the above problems, especially transformer [4-7] model. Compared with CNN [8-10] or RNN [11-13], the self-attention mechanism of transformer helps the model to treat any length of time series equally, which enables the transformer to better handle long-term time sequence and capture long-term dependencies. However, existing transformer based time series prediction algorithms have the following problems: 1) The encoder-decoder structure of the transformer makes the model more complex and inefficient, which limits the ability to construct deep temporal information dependencies by stacking multiple layers of attention. 2) In existing algorithms, the information interaction between historical and predicted sequences is poor, which affects the predictive performance of the model. 3) The self-attention mechanism of the transformer brings the calculation expense and memory usages of the model to increase twice with the input length, so the issue of computational complexity further limits the algorithm's expansion in model depth.

To address the above problems, a novel time-series forecasting algorithm is designed to simplify traditional models and achieve breakthroughs in constructing deep temporal dependencies, as shown in **Fig. 1**. The model abandons the traditional encoder-decoder structure of transformer, it is only composed of multi-layer attention modules. The head and tail of the model are composed of sparse self-attention modules, and the middle part is composed of history-prediction cross-attention modules. The main work of this article is introduced as follows.

(1) The complex encoder-decoder structure in traditional Transformer based time series prediction models is abandoned. Using only multi-layer attention modules to construct a time series prediction model makes the model structure more concise and improves the depth of the model, thereby improving the algorithm's ability to mine deep time dependencies.

(2) A cross-attention module is designed to optimize the information of history sequences and prediction sequences separately, and realize the information interaction between history sequences and prediction sequences through the cross-attention mechanism. It improves the information exchange capabilities of history sequences and prediction sequences.

(3) Apply sparse attention mechanism to the algorithm to decrease the computation complexity of the model. Inspired by ProbeSpare attention mechanism proposed in Informer, the self-attention and cross attention modules in our model are replaced with ProbeSpare attention modules, reducing the computational complexity of long sequence self-attention from $O(L^2)$ to $O(L\log L)$ and improving the efficiency of the entire model.

The test results indicates that the model raised in this article has obtained good results on main public datasets, which exceeds the traditional time-series forecasting methods and the recently proposed time-series forecasting method based on transformer.

2. Related work

2.1 Traditional time-series forecasting methods

Many time-series forecasting models have been well developed because of its important role, and time-series forecasting algorithms begin with classical tools [14, 15]. ARIMA [1, 2] uses the difference methods to transform non-stationary process into stationary process in time series forecasting. The stationarity of time-series data is an important prerequisite for constructing ARIMA model. In addition, the recursive neural network model (RNN) is used to model the time dependence of time-series [16-19]. RNN is a kind of feedback neural network, which is windy used to deal with sequence data. But it cannot learn too long sequence features. DeepAR [20] integrates autoregressive model and RNN to simulate the probability distribution of the coming sequences. LSTNet [21] brings in convolutional neural networks (CNNs) with recursive skip connections to catch short-term and long-term time dependencies. Refs [22-24] introduces temporal attention to seek the long-term time pattern of forecasting. LSTM [25] is a special form of RNN, which can avoid the gradient disappearance problem and learn long-term information. Relying on the traditional self-attention mechanism, different attention mechanism modules are proposed in the recent years [41-44]. PSP learning framework designs a novel Pyramid Polymerizing Attention (PPA) mechanism, which is able to supplement the body, part and joint level semantic information [41]. SDS-CL designs a new Spatiotemporal-decoupling Intra-Inter Attention (SIIA) that intended to capture spatiotemporal specific information separately [42]. In addition, many studies based on time convolution network (TCN) [26-29] try to use causal convolution to model time causality. TCN is a method that introduces the idea of convolutional neural network into time-series data prediction, that is, convolutional network is used to process time-series forecasting. Compared with LSTM, the main advantage of TCN is that its training and processing speed is much faster than LSTM when the accuracy of timing prediction tasks is similar or even higher than LSTM. This is because TCN is based on the idea of image parallelism and the confluence of convolution neural core, which can integrate a large amount of bottom information in a small processing unit. It can enhance the exactitude and efficiency of TCN in processing a large number of multi-dimensional data and longtime time-series data.

2.2 Transformer model in time-series forecasting

Recently, the Transformer [30, 31] model has shown strong ability to process sequence information and is widely used in tasks such as natural language processing [32, 33], audio signal processing [34]. Some scholars have also adopted this model in the images [35, 36]. Therefore, some researchers try to use transformer-based models to predict time series data. This method uses self-attention mechanism to mine useful information association relationships in time series data, including univariate and multivariate time series information. Transformer contains two parts: encoder and decoder. The former takes the historical sequence as input, and the decoder part takes the splicing of the historical sequence and the sequence to be predicted as input. The latter passes information from the historical sequence to the decoder. In this way, the model can "focus" on the most valuable information in the historical sequence before making a prediction. Decoders use masked self-attention to prevent information leakage. However, the self-attention mechanism in Transformer has a large drawback in dealing with long-term time series prediction because of the computational complexity of $O(L^2)$. Some work [37, 38] proposed different attention models to alleviate this problem. Recently, the ProbSparse attention mechanism based on KL dispersion was proposed in Informer [7], which implements the computational complexity of $O(L\log L)$. As can be seen,

above algorithms are based on the classical architecture of Transformer, attempt to change the self-attention mechanism to a sparse form. It still adopts an encoder-decoder structure. The reason why this structure is complex is that the encoder and decoder are each composed of self-attention layers, which is equivalent to two sets of parallel attention modules executing calculations simultaneously, resulting in a decrease in computational efficiency. Our algorithm abandons this inherent structure and only uses a more concise multi-layer attention structure to build a deep attention-based model.

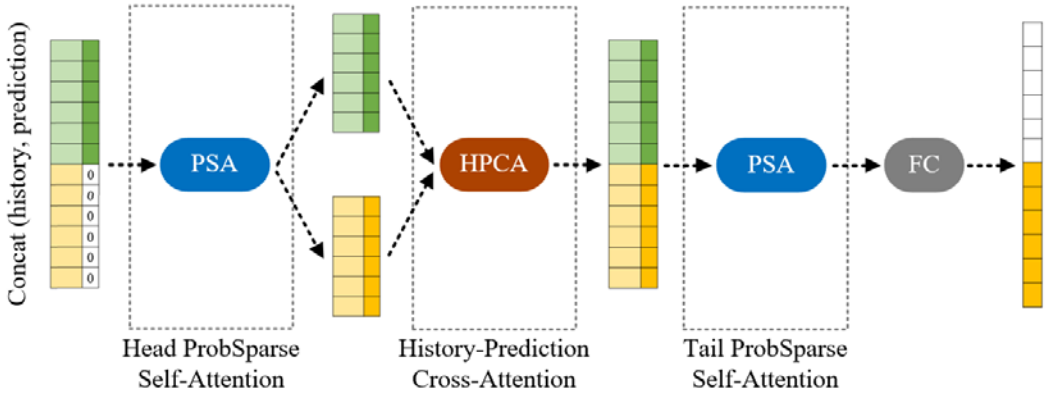

Fig. 1. Our time-series forecasting network architecture

3. Network architecture

Our network architecture is composed of head-tail ProbSparse Self-Attention (PSA) layer and History-Prediction Cross-Attention (HPCA) layer. The structure diagram is shown in **Fig. 1**. In the figure, green represents the historical sequence and orange represents the predicted sequence. Firstly, the model accepts long sequence input, which is composed of history sequences and sequences to be predicted, in which the prediction sequences are filled with zeros. The sequences are sent to the PSA layer to learn the dependencies between sequences, and then the output is sent to the HPCA layer, which optimizes the history sequences and prediction sequences independently, realizes the cross information interaction between the two sequences, and then splices the two sequences. The spliced sequences are sent to the PSA again for further overall optimization. Finally, the ultimate forecasting results are achieved through a fully connected layer. Through the modeling of the deep dependency between sequences, the accurate prediction of future data is realized.

3.1 Head-tail probsparse self-attention layer

The original self-attention is based on tuple input, i.e. Query (Q), Key (K) and Value (V). Given Q, K and V, the attention function is scaled dot-product attention, which is defined in equation (1).

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{1}$$

where d_k is the dimension of K.

Expanding the self-attention mechanism to multiple heads for considering different attention distributions, it allows the model to focus on multiple aspects of information. The multi-head attention mechanism is described in equation (2). More comprehensive explanation can be

found in Ref. [30].

$$MultiHead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = Concat(\boldsymbol{H}_1,\dots,\boldsymbol{H}_{n_h})\boldsymbol{W}^O \tag{2}$$

$$\boldsymbol{H}_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \tag{3}$$

where $\in \mathbb{R}^{d_m\times d_k}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d_m\times d_k}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d_m\times d_v}$, $\boldsymbol{W}^O \in \mathbb{R}^{n_h d_v\times d_m}$ is a parameter matrix.

This self-attention mechanism has high computational overhead and high memory consumption, which limits the time-series forecasting ability of transformer model.

Informer breaks the limitation that each multi-headed self-attention in Sparse transformer [39], LogSparse transformer [6] and Longformer [40] is processed with the same strategy, and proposes a ProbSparse self-attention mechanism, which cuts down the computational complexity of self-attention from $O(L^2)$ to $O(L\log L)$, thus improving the processing efficiency.

The ProbSparse attention mechanism is applied to all attention layers of the model in this paper. The self-attention layer receives the long sequences spliced by history sequences and prediction sequences, and outputs the long sequences with the same dimension. Specifically, the following vectors are input into the attention layer.

$$\boldsymbol{X}_i^t = Concat(\boldsymbol{X}_p^t,\boldsymbol{X}_0^t) \in \mathbb{R}^{(L_P+L_y)\times d_m} \tag{4}$$

where $\boldsymbol{X}_p^t \in \mathbb{R}^{L_P\times d_m}$ is the history sequence; and $\boldsymbol{X}_0^t \in \mathbb{R}^{L_y\times d_m}$ is the prediction sequence, which is filled with 0. Taking the prediction data of 168 time points (the experimental part is 7-day data) as an example, the first 4 days of the known sequences are taken as the history sequences, and the spliced sequence is $\boldsymbol{X}_i^t = \{\boldsymbol{X}_{4d},\boldsymbol{X_0}\}$. $\boldsymbol{X_0}$ contains the timestamp of the target sequences. Then, this model predicts the output through a forward process, rather than the 'dynamic decoding' in the traditional encoder-decoder architecture.

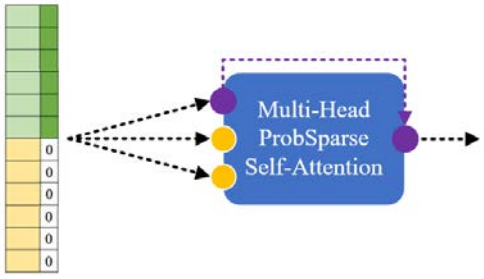

Fig. 2. PSA structure diagram

The tail attention layer obtains the terminal output through the fully connected layer, and the scale of the output is determined by making the univariate or multivariate predictions.

The structural diagram of ProbSparse Self-Attention layer (PSA) is shown in **Fig. 2**. The purple dotted arrow indicates the residual structure. The purple dot indicates the matrix Q, and the orange dot indicates the matrix K and V.

Under the action of self-attention mechanism, the history sequences and prediction sequences are regarded as a whole to construct the dependence between different time points. At the same time, under the front multi-head attention mechanism, the prediction sequences can also learn some information from the history sequences, which is conducive to the subsequent processing. In this paper, we set the multi-head ProbSparse Self-Attention stack two layers to mine the deeper time dependence.

3.2 History-prediction sequence cross-attention layer

The history-prediction sequence cross-attention layer is located in the middle of the head-tail ProbSparse self-attention layer. The history sequences and prediction sequences are separated and optimized separately, and the two are associated by using the cross-attention module. By highlighting their independence, the information interaction ability of history sequences and prediction sequences is improved, and then the forecasting effects of the algorithm is increased.

Specifically, the output of the ProbSparse self-attention layer in the head of algorithm is used as the input of this layer. At this time, the prediction sequences have learned some information from the history sequences. First, the two parts of the long sequences (history sequences and prediction sequences) is separated, as shown in equation (5):

$$\boldsymbol{X}_p^t, \boldsymbol{X}_f^t = Split\left(\boldsymbol{X}_i^t\right) \in \mathbb{R}^{L_P \times d_m}, \mathbb{R}^{L_f \times d_m} \tag{5}$$

where $\boldsymbol{X}_p^t \in \mathbb{R}^{L_P \times d_m}$ is the history sequence and $\boldsymbol{X}_f^t \in \mathbb{R}^{L_f \times d_m}$ is the prediction sequence. The above two vectors are input into the multi-head ProbSparse self-attention module respectively.

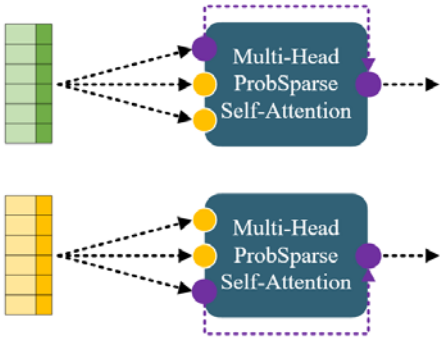

Fig. 3. Schematic diagram of HPCA self-attention stage

The multi-head ProbSparse self-attention module adaptively integrates the information from different time points of history sequences and prediction sequences in the form of residual and multi-head ProbSparse self-attention. Through the self-attention operation of the history sequences and the prediction sequences respectively, the information association within the two sequences can be established, and the heterogeneity of the information of the two sequences can be highlighted, which is convenient to realize the interaction of high-value information in the history sequences and the prediction sequences in the subsequent cross-attention module. Because the attention mechanism does not have the ability to distinguish the relative position relationship between different time points, position location is introduced to encode the input $\boldsymbol{X}_p^t$ and $\boldsymbol{X}_f^t$, and a sinusoidal function is adopted to produce time location code. In the end, the mechanism of multi-head self-attention module can be summarized.

$$\boldsymbol{X}_{SA} = \boldsymbol{X} + MultiHead(\boldsymbol{X} + \boldsymbol{P}_x, \boldsymbol{X} + \boldsymbol{P}_x, \boldsymbol{X}) \tag{6}$$

where $\boldsymbol{P}_x \in \mathbb{R}^{L_p \times d_m}$, $\mathbb{R}^{L_f \times d_m}$ represents time position code, $\boldsymbol{X}_{SA} \in \mathbb{R}^{L_p \times d_m}$, $\mathbb{R}^{L_f \times d_m}$ is the output of the module. The output is fed into the multi-head ProbSparse cross-attention module. The schematic diagram of the self-attention stage of the history-prediction HPCA is shown in **Fig. 3**. The purple dotted arrow indicates the residual structure. The purple dot indicates the

matrix Q, and the orange dot indicates the matrix K and V.

The multi-head ProbSparse cross-attention module also adopts the form of residual, and uses the multi-head ProbSparse cross-attention to fuse the feature vectors of two input to realize the information interaction between the history sequences and the prediction sequences. Similar to the multi-head ProbSparse self-attention module, time location code is also used in the multi-head ProbSparse cross-attention module. Furthermore, to improve the fitting performance of the model, an FFN module is used. The module is a fully connected feedforward network, it contains two linear transformations with a relu activation function ReLU in the middle, i.e.

$$FFN(\boldsymbol{X}) = \max(0, \boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_3 \tag{7}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ indicate the weight matrix and the basis vector respectively. Subscripts indicate different layers. So the action mechanism of multi-head ProbSparse cross-attention module can be summarized.

$$\begin{aligned} \boldsymbol{X}_{CA} &= \boldsymbol{X}'_{CA} + FFN(\boldsymbol{X}'_{CA}), \\ \boldsymbol{X}'_{CA} &= \boldsymbol{X}_q + MultiHead(\boldsymbol{X}_q + \boldsymbol{P}_q, \boldsymbol{X}_{kv} + \boldsymbol{P}_{kv}, \boldsymbol{X}_{kv}) \end{aligned} \tag{8}$$

where $\boldsymbol{X}_q$ is the input and $\boldsymbol{P}_q$ is the corresponding spatial location code of $\boldsymbol{X}_q$. $\boldsymbol{X}_{kv}$ is an input from another branch, $\boldsymbol{P}_{kv}$ is the time coding of $\boldsymbol{X}_{kv}$. $\boldsymbol{X}_{CA}$ is the output of the multi-head ProbSparse cross-attention module. According to equation (8), the cross-attention module computes the attention map based on the multiple scaled point products between $\boldsymbol{X}_{kv}$ and $\boldsymbol{X}_q$, then reweights $\boldsymbol{X}_{kv}$ based on the attention map and puts it into $\boldsymbol{X}_q$ to strengthen the expression performance of time-series information. The schematic diagram of the cross-attention stage of the history-prediction HPCA is shown in **Fig. 4**. The purple dotted arrow indicate the residual structure. The purple dot indicates the matrix Q, and the orange dot indicates the matrix K and V.

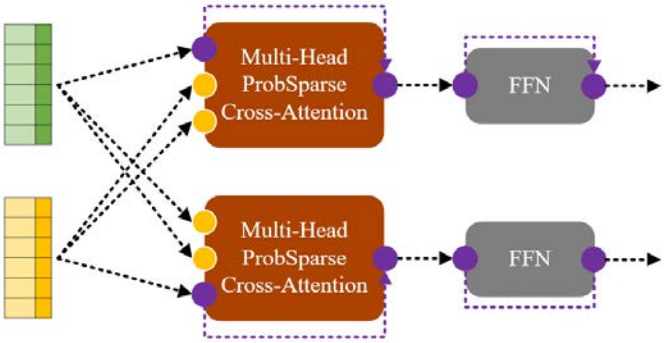

Fig. 4. Schematic diagram of HPCA cross-attention stage

Under the action of the cross-attention layer of history-prediction sequences, the history sequences and prediction sequences are separated, the independence of the two is highlighted, the two are optimized separately, and the correlation of the two sequences is learned through the cross-attention mechanism, it enhances the information interaction ability of history sequences and prediction sequences. In this paper, the history-prediction sequence cross-attention module is stacked two layers to obtain a deep time-series forecasting algorithm based on attention, it greatly improves the ability of information interaction.

4. Experimental results and performance analysis

The same datasets (ETT, ECL and Weather) as Informer [7] is used to do experiments in this paper. A detailed description of the dataset can be found in reference [7]. This paper selects three Transformer-based time-series forecasting methods as comparison, including Reformer, LogTrans and Informer. Compared to various high-performance time-series forecasting algorithms such as ARIMA, Prophet and LSTM, these Transformer-based algorithms have demonstrated their advanced performance through experiments. Therefore, it is meaningful to deeply study the extent to which our algorithm can improve Transformer-based time-series forecasting methods.

4.1 Experimental details

Through several groups of experiments, it is confirmed that our algorithm can improve the prediction effect and efficiency of transformer or like transformer model in time-series forecasting. The loss function is MSE. All methods use Adam optimizer, whose learning rate starts from 1e-4 and decays twice per epoch. The entered history sequence length is set to 96. The total number of epoch is 8, which should be stopped early appropriately. The batchsize is set to 32. Each individual experiment is repeated five times and the mean results are used. All the experiments are conducted on a single NVIDIA GTX 3090 24GB GPU. More experimental descriptions will be presented in the following parts. MSE loss function is selected in the prediction of the target sequence.

4.2 Analysis of experimental results

The performance of these four time series prediction algorithms is evaluated under univariate and multivariate conditions respectively, in order to understand the accuracy improvement of this method over Transformer-based methods. ETTh1, ETTh2, ETTm1, Weather and ECL datasets are used for testing. To explore the performance with various granularities, the length of the forecasting series are set to {24, 48, 168, 336} for ETTh1, ETTh2 and Weather datasets, {24, 48, 96, 288} for ETTm1 dataset and {48, 168, 336, 720} for ECL dataset. The indicators used to compare the accuracy of different algorithms are: $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$ and $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$, n represents the size of the forecasting window. The experiment in this part is conducted on five datasets, ETTh1, ETTh2, ETTm1, Weather and ECL, and is divided into univariate prediction and multivariate prediction. If necessary, these two indexes will also be used to estimate the prediction effect of subsequent experiments. When our model is tested, the training time on the ETTh1 dataset is 1.5 hours, and the testing time is 8.65 seconds, 9.22 seconds, 9.64 seconds and 9.98 seconds for 24, 48, 168 and 336 prediction length, respectively. The specific experimental results are list in **Table 1 ~ 2**.

It can be shown from **Table 1 ~ 2** that the effect of our algorithm is better than Transformer-based algorithms in all the univariate and multivariable cases, which shows that our proposed architecture does improve the prediction ability and performance of like transformer algorithm in time-series forecasting problems. Especially in the case of multivariable prediction, the proposed architecture is significantly improved compared with other algorithms, which shows that the deep attention-based architecture constructed in this paper has strong ability to mine the time dependence between multivariable and has higher prediction accuracy compared with the encoder-decoder based Transformer architecture. It is helpful to meet the actual demand for time-series prediction performance (especially multivariable time-series prediction).

Table 1. Univariate forecasting results on ETTh1, ETTh2, ETTm1, Weather and ECL

Methods		Ours		Informer		LogTrans		Reformer	
Prediction length		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	**0.083**	**0.228**	0.110	0.271	0.115	0.283	0.234	0.413
	48	**0.129**	**0.283**	0.221	0.405	0.230	0.414	0.347	0.531
	168	**0.243**	**0.412**	0.311	0.487	0.335	0.516	1.650	1.332
	336	**0.273**	**0.435**	0.354	0.520	0.362	0.531	1.992	1.257
ETTh2	24	**0.077**	**0.209**	0.097	0.241	0.106	0.256	0.267	0.438
	48	**0.142**	**0.291**	0.169	0.327	0.183	0.361	0.472	0.558
	168	**0.221**	**0.374**	0.256	0.410	0.270	0.443	1.053	0.900
	336	**0.246**	**0.396**	0.281	0.428	0.285	0.448	1.686	1.239
ETTm1	24	**0.025**	**0.126**	0.034	0.150	0.073	0.221	0.100	0.242
	48	**0.056**	**0.180**	0.097	0.258	0.107	0.278	0.304	0.465
	96	**0.258**	**0.443**	0.330	0.524	0.322	0.531	0.997	0.858
	288	**0.493**	**0.622**	0.639	0.744	0.647	0.763	1.187	1.392
Weather	24	**0.100**	**0.232**	0.131	0.275	0.152	0.305	0.243	0.426
	48	**0.146**	**0.282**	0.249	0.404	0.284	0.449	0.401	0.505
	168	**0.354**	**0.474**	0.452	0.560	0.500	0.604	0.709	0.709
	336	**0.365**	**0.468**	0.474	0.559	0.565	0.646	1.919	1.222
ECL	48	**0.204**	**0.332**	0.268	0.394	0.313	0.469	1.023	0.939
	168	**0.365**	**0.446**	0.625	0.639	0.625	0.668	2.042	1.894
	336	**0.651**	**0.630**	0.831	0.743	0.832	0.775	3.825	2.456
	720	**0.663**	**0.641**	0.861	0.767	0.878	0.813	5.238	4.526
Counting		40		0		0		0	

Table 2. Multivariate forecasting results on ETTh1, ETTh2, ETTm1, Weather and ECL

Methods		Ours		Informer		LogTrans		Reformer	
Prediction length		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	**0.522**	**0.524**	0.637	0.582	0.746	0.637	1.051	0.787
	48	**0.568**	**0.550**	0.645	0.606	0.726	0.738	1.273	0.887
	168	**0.661**	**0.605**	0.986	0.789	1.057	0.883	1.879	1.175
	336	**0.860**	**0.715**	1.137	0.854	1.371	0.933	2.126	1.261
ETTh2	24	**0.933**	**0.766**	1.885	1.102	1.993	1.187	2.696	2.050
	48	**1.444**	**0.937**	2.063	1.156	2.412	1.189	2.477	1.890
	168	**6.600**	**2.025**	6.919	2.294	7.500	2.460	8.090	2.625
	336	**4.769**	**1.747**	5.101	1.929	6.253	2.352	6.406	2.277
ETTm1	24	**0.292**	**0.352**	0.357	0.391	0.456	0.435	0.768	0.634
	48	**0.410**	**0.442**	0.465	0.488	0.481	0.568	1.065	0.761
	96	**0.481**	**0.494**	0.718	0.644	0.810	0.827	1.476	0.976
	288	**0.805**	**0.644**	1.064	0.769	1.472	1.294	1.828	1.078
Weather	24	**0.303**	**0.364**	0.370	0.404	0.473	0.503	0.695	0.609
	48	**0.328**	**0.404**	0.372	0.445	0.404	0.483	0.707	0.652
	168	**0.435**	**0.456**	0.644	0.595	0.767	0.700	1.358	0.883
	336	**0.535**	**0.507**	0.708	0.607	0.759	0.657	1.938	1.150
ECL	48	**0.311**	**0.375**	0.380	0.417	0.386	0.441	1.489	1.043
	168	**0.306**	**0.373**	0.347	0.411	0.349	0.421	1.469	1.047
	336	**0.271**	**0.347**	0.404	0.452	0.393	0.458	1.649	1.140
	720	**0.309**	**0.363**	0.409	0.433	0.412	0.445	2.018	1.153
Counting		40		0		0		0	

4.3 Ablation Experiment Analysis

To evaluate the effect and contribution of the two important components of the proposed algorithm (head-tail ProbSparse self-attention layer and history-prediction sequences cross-attention layer) in time-series forecasting, two sets of ablation experiments are carried out on the ETTh1 dataset based on MSE and MAE indexes.

Table 3. Ablation experiments of head-tail ProbSparse self-attention layer

Methods		Ours(N=2)		N=1		N=0	
Prediction length		MSE	MAE	MSE	MAE	MSE	MAE
Univariate	24	**0.083**	**0.228**	0.086	0.232	0.088	0.233
	48	**0.129**	**0.283**	0.156	0.317	0.147	0.305
	168	**0.243**	**0.412**	0.266	0.436	0.283	0.442
	336	**0.273**	**0.435**	0.343	0.497	0.308	0.470
Multivariate	24	0.522	0.524	0.494	0.498	**0.478**	**0.491**
	48	0.568	0.550	0.544	0.533	**0.518**	**0.521**
	168	**0.661**	**0.605**	0.685	0.624	0.715	0.639
	336	0.860	**0.715**	**0.855**	0.716	0.903	0.731
Counting		11		1		4	

Table 4. Ablation experiments of history-prediction cross-attention layer

Methods		Ours(N=2)		N=1		N=0	
Prediction length		MSE	MAE	MSE	MAE	MSE	MAE
Univariate	24	**0.083**	**0.228**	0.086	0.236	0.096	0.247
	48	**0.129**	**0.283**	0.175	0.340	0.211	0.377
	168	0.243	0.412	0.265	0.426	**0.234**	**0.400**
	336	0.273	0.435	**0.267**	**0.433**	0.271	0.439
Multivariate	24	**0.522**	**0.524**	0.550	0.527	0.654	0.593
	48	**0.568**	**0.550**	0.596	0.558	0.675	0.613
	168	**0.661**	**0.605**	0.662	**0.605**	0.803	0.667
	336	**0.860**	**0.715**	0.916	0.742	0.965	0.752
Counting		12		3		2	

In the first group of experiments, the number of head-tail ProbSparse self-attention layers is reduced to one layer, univariate and multivariable prediction are carried out on the ETTh1 dataset, and then the head-tail ProbSparse self-attention layer is completely removed for the same experiment. The role of head-tail ProbSparse self-attention layer in this algorithm is verified by comparing the experimental results. The experimental results are list in **Table 3**.

In the second group of experiments, the number of head-tail ProbSparse self-attention layers is kept unchanged, the number of cross-attention layers of history-prediction sequences is reduced to one layer, the univariate and multivariable prediction on ETTh1 dataset are carried out, and then the of history-prediction sequences cross-attention layer is completely removed for the same experiment. The comparison experimental results verify the role of history-prediction sequence cross-attention layer in this algorithm. The experimental results are shown in **Table 4.**

It can be seen from **Table 3** that in the case of multivariate prediction and short prediction sequences, ProbSparse self-attention layer may bring negative effects. The reason is that the

periodic relationship of short sequences is not obvious. For the prediction of short sequences, whether the algorithm can catch the subtle and short-term dependence between sequences is very important. ProbSparse self-attention mechanism tends to focus on the periodic information of the sequence, which will ignore or even destroy the subtle local information to a certain extent, and in the case of multivariable, the periodic dependence between variables becomes worse, so this disadvantage of ProbSparse self-attention mechanism will be further amplified. The ProbSparse self-attention layer destroys the local fine and short-term dependence information beneficial to the model prediction, it will negatively affect the forecasting accuracy of the algorithm. However, in most other cases, the algorithm of stacking two layers of head-tail ProbSparse self-attention layers can achieve good results.

As can be seen from **Table 4**, the algorithm of stacking two layers of history-prediction sequences cross-attention layer can achieve better results in most cases, and only in a few cases achieve suboptimal results, but the gap is not significant. It can be seen that in addition to in the case of univariate and long series prediction it will bring a small loss of accuracy, in most other cases, the history-prediction sequence cross-attention layer designed in this paper can bring a significant performance improvement. This is because the proposed history-prediction sequence cross-attention layer can enhance the information interaction between history sequences and prediction sequences, simultaneously increase the depth of the algorithm, and make the model more capable of capturing time-dependent relationships.

Through the analysis of the ablation experiment results, it can be seen that each module constructed in this model can increase the effect of time-series forecasting to some extent, and make a certain contribution in time-series forecasting. The integrated algorithm structure architecture is simple and clear. By means of multi-level self-attention layer connection and information interactive fusion layer based on cross attention, multi-layer attention architecture can give full play to its potential in time-series forecasting task. The experiments results confirm that our algorithm can achieve higher prediction effect in most cases, it further demonstrates the robustness and effectiveness of the algorithm and its modules designed in this paper.

5. Conclusion

A novel time-series forecasting algorithm based on multi-layer attention architecture is proposed and tested on multiple public datasets in this paper, it has achieved excellent prediction performance. Our algorithm abandons the classical encoder-decoder architecture of Transformer and only uses multi-layer feedforward structure to construct the deep attention-based model, which improves the ability of mining deep time dependencies. At the same time, the head-tail ProbSparse self-attention layer and history-prediction sequence cross-attention layer are designed to enhance the ability of information interaction and mining deep information relevance. Experiments results confirm that our model can get better forecasting performance on five datasets. In the follow-up research, we will explore the limitations of ProbSparse attention mechanism, explore more effective methods of time-series relationship modeling based on attention, and further increase the effect of the algorithm.

Acknowledgments

This work is supported by University Philosophy and Social Science Research Project of Jiangsu province, 2019SJA2034; Key Projects of Special Topics for Financial Development of High-quality Social Science Application Research Project of Jiangsu province,17SCA-06.

References

[1] G. E. P. Box, G. M. Jenkins, J. F. MacGregor, "Some recent advances in forecasting and control," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 23, no. 2, pp. 158-179, February 1974. Article (CrossRef Link).

[2] G. E. P. Box, and G. M. Jenkins, "Time series analysis: forecasting and control," *Journal of Time*, vol. 31, no. 3, March 2010. Article (CrossRef Link).

[3] J. Durbin, and S. J. Koopman, "Time series analysis by state space methods," *Oxford university press*, July 2012. Article (CrossRef Link).

[4] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, January 2019. Article (CrossRef Link).

[5] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748-1764, April 2021. Article (CrossRef Link).

[6] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. of NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, no. 471, pp. 5243-5253, December 2019. Article (CrossRef Link).

[7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: beyond efficient transformer for long sequence time-Series forecasting," *arXiv preprint arXiv:2012.07436*, December 2019. Article (CrossRef Link).

[8] S. Bai, J. Z. Kolter, and V. Koltun, "Convolutional sequence modeling revisited," in *Proc. of ICLR 2018 Conference*, February 2018. Article (CrossRef Link).

[9] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, "Wavenet: a generative model for raw audio," *arXiv preprint arXiv:1609.03499*, September 2016. Article (CrossRef Link).

[10] D. Stoller, M. Tian, S. Ewert, and S. Dixon, "Seq-U-Net: a one-dimensional causal U-Net for efficient sequence modelling," *arXiv preprint arXiv:1911.06393*, November 2019. Article (CrossRef Link).

[11] G. K. Lai, W. C. Chang, Y. M. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95-104, June, 2018. Article (CrossRef Link).

[12] R. Yu, S. Zheng, A. Anandkumar, and Y. Yue, "Long-term forecasting using tensor-train rnns," *arXiv preprint arXiv: 1711.00073*, October 2017. Article (CrossRef Link).

[13] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181-1191, March 2020. Article (CrossRef Link).

[14] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16-18, pp. 2861-2869, October 2007. Article (CrossRef Link).

[15] Renyi. Chen and M. Tao, "Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps," *arXiv preprint arXiv: 2103.05632*, March 2021. Article (CrossRef Link).

[16] R. Wen, K. Torkkola, B. Narayanaswamy, "A multi-horizon quantile recurrent forecaster," *arXiv preprint arXiv: 1711.11053*, November 2017. Article (CrossRef Link).

[17] S. S. Rangapuram, M. Seeger, J. Gasthaus, "Deep state space models for time series forecasting," in *Proc. of NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7796–7805, December 2018. Article (CrossRef Link).

[18] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell. "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv: 1704.02971*, August 2017. Article (CrossRef Link).

[19] D. C. Maddix, Y. Wang, and A. Smola, "Deep factors with gaussian processes for forecasting," *arXiv preprint arXiv:1812.00098*, November 2018. Article (CrossRef Link).
[20] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, Vol. 36, no. 3, pp. 1181-1191, July–September 2020. Article (CrossRef Link).
[21] G. Lai, W. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95-104, April 2018. Article (CrossRef Link).
[22] Q. Yao, D. Song, H. Chen, C. Wei, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 2627-2633, August 2017. Article (CrossRef Link).
[23] S. Y. Shih, F. K. Sun, and H. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, pp. 1424-1441, June 2019. Article (CrossRef Link).
[24] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. of the Association for the Advancement of Artificial Intelligence*, February 2018. Article (CrossRef Link).
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, November 1997. Article (CrossRef Link).
[26] A. V. D. Oord, S. Dieleman, H. Zen, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv: 1609.03499*, September 2016. Article (CrossRef Link).
[27] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, March 2017. Article (CrossRef Link).
[28] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, March 2018. Article (CrossRef Link).
[29] R. Sen, H. F. Yu, and I. S. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," *arXiv preprint arXiv: 1905.03806*, May 2019. Article (CrossRef Link).
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of the 31st International Conference on Neural Information Processing Systems*, pp. 6000-6010, December 2017. Article (CrossRef Link).
[31] S. Wu, X. Xiao, Q. Ding, "Adversarial sparse transformer for time series forecasting," in *Proc. of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, February 2020. Article (CrossRef Link).
[32] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv: 1810.04805*, October 2018. Article (CrossRef Link).
[33] T. Brown, B. Mann, N. Ryder, "Language models are few-shot learners," *arXiv preprint arXiv: 2005.14165*, May 2020. Article (CrossRef Link).
[34] C. Z. Anna Huang, A. Vaswani, J. Uszkoreit, "Music transformer," *arXiv preprint arXiv: 1809.04281*, September 2018. Article (CrossRef Link).
[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. of the ICLR 2021*, June 2021. Article (CrossRef Link).
[36] Z. Liu, Y. Lin, Y. Cao, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the ICCV 2021*, August 2021. Article (CrossRef Link).
[37] S. Li, X. Jin, Y. Xuan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. of the 33rd International Conference on Neural Information Processing Systems*, pp. 5243-5253, December 2019. Article (CrossRef Link).
[38] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. of the ICLR 2020*, January 2020. Article (CrossRef Link).

[39] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating Long Sequences with Sparse Transformers," *arXiv preprint arXiv: 1904.10509*, April 2019. Article (CrossRef Link).
[40] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," *arXiv preprint arXiv: 2004.05150*, April 2020. Article (CrossRef Link).
[41] B. Xu, X. Shu, "Spatiotemporal Decouple-and-Squeeze Contrastive Learning for Semi-Supervised Skeleton-based Action Recognition," *arXiv preprint arXiv: 2302.02316*, February 2023. Article (CrossRef Link).
[42] B. Xu, X. Shu, "Pyramid Self-attention Polymerization Learning for Semi-supervised Skeleton-based Action Recognition," *arXiv preprint arXiv: 2302.02327*, February 2023. Article (CrossRef Link).
[43] X. Shu, B. Xu, L. Zhang, and J. Tang, "Multi-Granularity Anchor-Contrastive Representation Learning for Semi-supervised Skeleton-based Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7559-7576, June 2023. Article (CrossRef Link).
[44] B. Xu, X. Shu, and Y. Song, "X-invariant Contrastive Augmentation and Representation Learning for Semi-Supervised Skeleton-Based Action Recognition," *IEEE Transactions on Image Processing*, vol. 31, no. 5, pp. 3852-3867, May 2022. Article (CrossRef Link).

Na Wang received the B.S. degree from the Shandong University of Technology, in 2004, and the M.S. degree from the Nanjing University of Aeronautics and Astronautics, in 2008, where she is currently pursuing the Ph.D. degree with the College of Economics and Management, Nanjing University of Aeronautics and Astronautics. She is an Associate Professor with the Nanjing Audit University Jinshen College. Her current research interests include machine learning and time series prediction.

Xianglian Zhao received the Ph.D. degree from Nanjing University of Science and Technology, in 2005. Since 2005, she has been with Nanjing University of Aeronautics and Astronautics, Nanjing, China, where she is currently a professor in the College of Economics and Management. Her current research interests include machine learning and time series prediction.