HMSformer: Hierarchical Multi-Scale Transformer for Multivariate Long-Term Series Forecasting

Xinyu Li, Yunqi Cai, Hao Xu, Xinyu Sun, Zhiheng Yang, Hong Lu, Xin Wang†, Jin Zhao
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University

***Abstract*—Multi-scale analysis, a classical and crucial methodology, is extensively employed in multivariate long-term time series forecasting. However, the existing methods struggle to model arbitrary scales, thus limiting their ability to delve into complex patterns, which in turn limits the forecasting precision. Therefore, we propose the HMSformer, a multi-scale Transformer that includes a shifted stacked embedding mechanism and a unified hierarchical framework, solving this problem in terms of the in-depth and comprehensiveness of multi-scale analysis. The mechanism conducts multi-scale modeling by sampling at different positions and sizes. This sampling reflects the arbitrariness of scales and allows for a thorough scan of any scale, supporting a deeper analysis of temporal relationships. Additionally, it converts input series into structured embeddings optimized for Transformer input, allowing the self-attention mechanism to effectively learn intricate cross-scale relationships. Moreover, the unified hierarchical framework integrates multi-scale modeling across different levels of abstraction by unifying global structures, local segments, and fine details into a consistent representation. Together, these two innovations ensure that HMSformer achieves a more comprehensive and in-depth analysis of temporal relationships. HMSformer demonstrates its effectiveness with a significant improvement, achieving an average mean squared error (MSE) that is 9.925% lower than the baseline across nine authoritative datasets. Source code is available at: https://github.com/lxy-PhD2022/HMSformer**

***Index Terms*—Time series forecasting, Multivariate long-term forecasting, Multi-Scale analysis**

I. INTRODUCTION

Time series is widely employed in multimedia, such as audio. Time series forecasting is essential for predicting future trends using past data. Its journey began in the 1970s with statistical methods [1]–[4] and machine learning methods [5]–[7]. It could be classified according to single or multiple variables and short-term or long-term predictions. Among them, the challenge of multivariate long-term prediction is particularly demanding and highly significant. This is because multiple variables interact, making the prediction of long-term variations more complex.

In recent years, powerful deep learning models have been extensively utilized, starting with Recurrent Neural Networks (RNNs) [8], [9]. Subsequently, models based on Convolutional Neural Network (CNN) [10], linear layer, and Transformer [11], were introduced to overcome the issue of error accumulation in long-term prediction task. Among them, methods such as Informer [12], Autoformer [13], DLinear [14], and PatchTST [15] have effectively improved the accuracy of multivariate long-term prediction. However, the analysis of temporal relationships by these methods is relatively simplistic, which limits further improvements in accuracy. Due to the effectiveness of multi-scale methods in temporal relationship analysis, many corresponding methods have been proposed but they still have shortcomings. Specifically, FEDformer [16] and TimesNet [17] potentially overlook non-periodic scale insights. Pyraformer [18] and TimeMixer [19] only model global scales and lack local ones. MSGNet [20] analyzes the relationship only between variables.

To improve prediction accuracy, there is an urgent need to develop methods that could comprehensively consider both macro trends and micro fluctuations, ensuring a comprehensive and in-depth analysis of the multi-scale characteristics in multivariate time series. Such a more in-depth approach could reveal richer layers of information, leading to higher accuracy in multivariate long-term prediction.

Thus, we propose a hierarchical multi-scale Transformer (HMSformer). Its hierarchical structure refers to the level of multivariate time series, while scale refers to the length of the sequence. HMSformer conducts multi-scale modeling through the shifted stacked embedding mechanism. This mechanism successively shifts the input sequence according to the stride and obtains the corresponding new sequence, and then stacks these sequences in the channel dimension.

At this time, each time step in the channel dimension is a local sequence of different positions and sizes, and they could be sampled at intervals and could represent a longer-term sequence range with a limited size. In this way, these local sequences could form a multi-scale representation that simultaneously has short-term fluctuations and long-term trends. Since different positions and sizes could represent the arbitrariness of scales, this means the mechanism could thoroughly and systematically scan any scale of the sequence. Therefore, the multi-scale analysis of HMSformer is more in-depth than existing methods. After that, HMSformer embeds multi-scale sequences into different tokens and fully learns the correlation of various details and trends through the self-attention to extract the complex patterns of time series.

Moreover, we design a unified hierarchical framework to extend the mechanism for comprehensively analyzing among sequences, between subsequences, and within subsequences level. The key operation of this framework is to treat all levels as consistent units. This enables us to systematically

†represents corresponding author, whose email is xinw@fudan.edu.cn. This work is partially supported by the National Natural Science Foundation of China (Project No. 62472101) and Shanghai Municipal Science and Technology Commission (Project No. 22dz1204900).

explore the overall structure of multivariate time series from a hierarchical perspective. By doing this, HMSformer could ensure seamless integration of information at different levels, providing a more comprehensive multi-scale analysis than existing methods. Combining these two innovations, HMSformer effectively addresses the limitations of existing methods in comprehensively and deeply analyzing temporal relationships.

Our main contributions are as follows.

- We design the shifted stacked embedding mechanism for HMSformer. It performs multi-scale modeling through sampling at different positions and sizes, which represent the arbitrariness of scales, enabling a thorough scan of any scale. Thereby, it could overcome the limitations of existing methods in analyzing temporal relationships with insufficient depth.
- We propose the unified hierarchical framework to apply the mechanism across sequences, between subsequences, and within subsequences. It overcomes the limitations of existing methods in the comprehensive analysis of temporal relationships.
- HMSformer achieves an average MSE that is 9.925% lower than the baseline across nine authoritative datasets, indicating its superior performance and effectiveness in multivariate long-term series forecasting.

II. RELATED WORK

At the beginning, Informer [12] introduced the Transformer which is good at long sequence modeling and proposed a more efficient ProbSparse self-attention. By means of one-time prediction, it surpasses the long-term prediction accuracy of RNN-based models. Autoformer [13] calculated the similarity within time series and then performs autocorrelation fusion on the most similar parts and makes predictions. DLinear [14] demonstrated that the linear layer can effectively capture temporal relationships, and its prediction accuracy surpasses that of previous Transformer-based models. For this reason, PatchTST [15] explored a more efficient application way of Transformer and proposed patch-level self-attention with channel-independent strategy.

In multi-scale analysis methods, TimesNet [17] obtained 2D sequences of different scales through different period divisions, then applied CV models to extract features and make predictions. This may lose aperiodic information. FEDformer [16] converted the input sequence into frequencies and then randomly selected a fixed number of components from high-frequency and low-frequency components for scale learning. However, the number of scales it utilizes is insufficient.

Pyraformer [18] repeated convolving time series to obtain multi-scale representations and then applied the self-attention mechanism for modeling and prediction. Relatively, TimeMixer [19] obtained multi-scale representations by performing average pooling on univariate sequences and then used a mixing mechanism to utilize multi-scale features for prediction. They rely on global scaling strategies and lose the diversity of scales. MSGNet [20] calculated the positive and negative correlations of different scale periodic subsequences among variables. However, MSGNet did not systematically and comprehensively analyze multi-scale relationships from multiple dimensions. The common problem of the above methods is that the multi-scale analysis is not in-depth enough, which may affect the prediction accuracy.

III. HMSFORMER

A. Problem Definition

In multivariate time series forecasting, the inputs are multiple time series $\boldsymbol{x}$ and each series x represents the single variate (channel). $\boldsymbol{x} \in R^{C \times L}$, where C is variate number and L is look-back window size. For each input series $x = (x_1, \ldots, x_L) \in R^{1 \times L}$ with single variate, the goal is to forecast T future values $y = (x_{L+1}, \ldots, x_{L+T}) \in R^{1 \times T}$. Therefore, the prediction result is $\boldsymbol{y} \in R^{C \times T}$.

B. Workflow Overview

We first apply instance normalization [21] to the input, normalizing each sample point to have zero mean and unit standard deviation. This effectively mitigates the impact of non-stationary sequences. We show HMSformer's workflow in Fig. 1(a) and describe its process at the caption. Formally, the workflow could be expressed as follows.

$$\boldsymbol{x}_{\text{uni}} = UHF(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x}$ and UHF are the input and unified hierarchical framework, respectively. Let $\boldsymbol{x}_{\text{uni}}$ represents the set of different levels unit sequences. Each level unit sequence will independently undergo the following processing until they are concatenated together.

$$\boldsymbol{x}_{\text{shi}} = Shift(\boldsymbol{x}_{\text{uni}}), \boldsymbol{x}_{\text{sta}} = Stack(\boldsymbol{x}_{\text{shi}}), \boldsymbol{x}_{\text{tok}} = Embed(\boldsymbol{x}_{\text{sta}}), \tag{2}$$

where $\boldsymbol{x}_{\text{uni}}$, $\boldsymbol{x}_{\text{shi}}$, $\boldsymbol{x}_{\text{sta}}$, $\boldsymbol{x}_{\text{tok}}$, $Shift$, $Stack$, and $Embed$ represent the input unit sequence, shifted sequence, stacked sequence, embedded multi-scale token, shifting operation, stacking operation, and embedding operation, respectively.

$$\boldsymbol{Q}_{\text{h}} = (\boldsymbol{x}_{\text{tok}})^T \boldsymbol{W}_{\text{h}}^{\text{Q}}, \boldsymbol{K}_{\text{h}} = (\boldsymbol{x}_{\text{tok}})^T \boldsymbol{W}_{\text{h}}^{\text{K}}, \boldsymbol{V}_{\text{h}} = (\boldsymbol{x}_{\text{tok}})^T \boldsymbol{W}_{\text{h}}^{\text{V}}, \tag{3}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent query, key, and value, respectively. $\boldsymbol{W}_{\text{h}}^{\text{Q}}$, $\boldsymbol{W}_{\text{h}}^{\text{K}}$, and $\boldsymbol{W}_{\text{h}}^{\text{V}}$ is projection matric in multi-head attention space, respectively. h represents the h-th head. $\boldsymbol{W}_{\text{h}}^{\text{Q}}, \boldsymbol{W}_{\text{h}}^{\text{K}}, \boldsymbol{W}_{\text{h}}^{\text{V}} \in R^{D \times d_k}$, where d_k is projection dimension.

$$\boldsymbol{O}_{\text{h}} = Softmax\left(\frac{\boldsymbol{Q}_{\text{h}} \boldsymbol{K}_{\text{h}}^T}{\sqrt{d_k}}\right) \boldsymbol{V}_{\text{h}}, \tag{4}$$

where $\boldsymbol{O}_{\text{h}}$ and $Softmax$ represent the output of self-attention and the softmax operation, respectively.

$$\boldsymbol{x}_{\text{att}} = LN(FFN(LN(\boldsymbol{Res})) + LN(\boldsymbol{Res})), \tag{5}$$

where $\boldsymbol{Res} = \boldsymbol{O}_{\text{h}} + \boldsymbol{x}_{\text{tok}}$ and it is a residue connection. $\boldsymbol{x}_{\text{att}}$, LN, and FFN represent the output of Transformer encoder, layer normalization, and feed-forward network, respectively.

$$\boldsymbol{x}_{\text{g}} = Flatten(\boldsymbol{x}_{\text{att}}), \boldsymbol{x}_{\text{enc}} = Concat(\boldsymbol{x}_{\text{g}}, \boldsymbol{x}_{\text{l}}, \boldsymbol{x}_{\text{d}}), \tag{6}$$

where $\boldsymbol{x}_{\text{g}}$ and $Flatten$ represent the global level multi-scale features and flatten operation, respectively. Similarly, we could obtain the

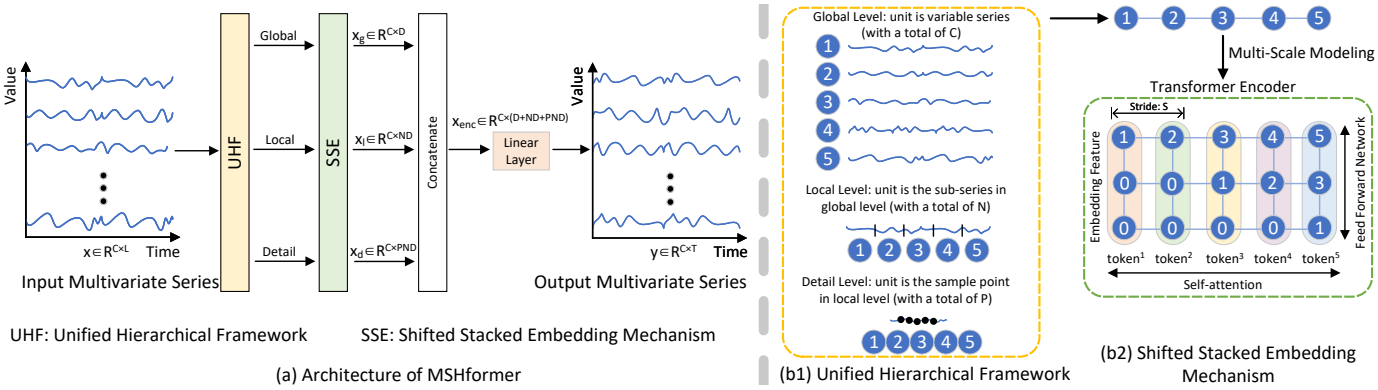

Fig. 1. (a) shows the workflow of HMSformer. It first converts the input into unit sequences of different levels through the unified hierarchical framework (shown in (b1)). Then, it applies the shifted stacked embedding mechanism (shown in (b2)) to conduct multi-scale analysis on them respectively. After that, it concatenates the results of all levels and passes them into a linear layer to make predictions.

local level and detail level multi-scale features, namely $\boldsymbol{x}_{\mathrm{l}}$ and $\boldsymbol{x}_{\mathrm{d}}$. In additon, $\boldsymbol{x}_{\mathrm{enc}}$ and $Concat$ represent the output feature of the entire encoding stage and concatenation operation, respectively.

$$\boldsymbol{y} = Linear(\boldsymbol{x}_{\mathrm{enc}}), \tag{7}$$

where $\boldsymbol{y}$ and $Linear$ represent the predicted multivariate time series and linear layer, respectively. $\boldsymbol{y} \in R^{C \times T}$ and T is prediction length.

C. Unified Hierarchical Framework

The role of this framework is to provide a more comprehensive and systematic perspective for multi-scale modeling and address the limitations of existing methods in analyzing temporal relationships comprehensively. The main temporal relationship levels of multivariate time series are among variables, among sub-sequences, and among sample points.

In the framework, we define the basic units for multi-scale modeling, shown in Fig. 1(b1). We consider the overall structure as the global level, where the unit is a single variable sequence. Its multi-scale modeling could learn the highest global relationship in the multivariate sequence, namely the variable relationship. We then refine the global structure into local segments, which serve as the local level. Here, the units are sub-series within each variable sequence. We divide each variable sequence into N sub-series and each variable conducts independent multi-scale modeling while sharing parameters. Similarly, the detail level focuses on the fine details within the local segments, with the units being sample points within the sub-series. The sub-series length P is the units number and each sub-series conducts independent multi-scale modeling while sharing parameters.

In this way, hierarchical approach could progressively refine modeling from global to detailed levels, enabling comprehensive and systematic learning. The basic units of global, local, and detail level could be represented as $\boldsymbol{u}_{\mathrm{glo}} \in R^{1 \times L}$, $\boldsymbol{u}_{\mathrm{loc}} \in R^{1 \times N}$, and $\boldsymbol{u}_{\mathrm{det}} \in R^{1 \times P}$, where the symbol meaning is explained above. We could treat each unit as a compressed point containing a sequence, where the first dimension is 1 and the second dimension corresponds to the sequence length. This allows us to unify the multi-scale modeling process by treating each unit as a sample point along the first dimension. In this way, we obtain unit sequences of the global, local, and detail level and feed them separately into the shifted stacked embedding mechanism for multi-scale modeling.

D. Shifted Stacked Embedding Mechanism

This mechanism's primary objective is to accurately modele multi-scale representations with different positions and sizes for unit sequences. Different positions and sizes represent the arbitrariness of scales. Therefore, this mechanism could thoroughly model multi-scale representations. By doing so, it addresses the deficiency of existing methods that often struggle to analyze temporal relationships with the necessary depth. For each hierarchical level, we apply this mechanism separately for multi-scale modeling. As shown in Fig. 1(b2), the input is a unit sequence from the unified hierarchical framework, which serves as the fundamental building block for our analysis. We then determine the shift stride S, which is a critical parameter dictating how the unit sequence will be transformed. The choice of S depends on the nature of the data and the desired level of detail in the multi-scale analysis. A smaller S will result in a more fine-grained exploration of the sequence, while a larger S could capture broader trends.

In shifted stacked embedding mechanism, the first step of multi-scale modeling is the shifting operation. The unit sequence is shifted forward along the time axis by S. This means that we are moving the window of observation over the time series data. As we shift, the parts of the sequence that are no longer covered are filled with zeros. This zero-padding could store multi-scale information while maintaining the sequence structure. For example, if we shift the unit sequence by stride $S = 3$, the values for the first 3 units in the new shifted sequence will be zero, while the subsequent values will be the actual values read from the original sequence, but now in a new relative position.

The new sequence obtained after shifting and zero-padding is then stacked along the channel dimension. This stacking operation creates a multi-dimensional structure that begins to capture the scales with different positions and sizes within the unit sequence. We repeat this entire process of shifting, zero-padding, and stacking until the entire new sequence consists solely of zeros. This repetition enables us to build up a full multi-scale representation.

After shifting and stacking, the next step is embedding. For the stacked unit sequence, the channel information at each time step is embedded as a scale, shown as the $token^1$ ... $token^5$ in Fig. 1(b2). Together, all embeddings construct the multi-scale representation. It could be observed that each scale contains sequence information of different positions and sizes. For instance, the first scale ($token^1$) and the second scale ($token^2$) represent different position, and the third scale ($token^3$) and the fourth scale ($token^4$) reflect the same relationship. Meanwhile, the first scale ($token^1$), the third scale ($token^3$), and the fifth scale ($token^5$) represent different size. Similarly, the second scale ($token^2$), the fourth scale ($token^4$), and the fifth scale ($token^5$) also relate to different size. In addition, we could perform interval sampling according to the setting of stride S, not just simply using the original time series segments, so as to store longer-term trend features in a limited size. Therefore, the

TABLE I
STATISTICS OF BENCHMARK DATASETS.

Datasets	Weather [22]	Traffic [13]	Electricity [13]	Exchange [23]	ILI [13]	ETTh1 [12]	ETTh2 [12]	ETTm1 [12]	ETTm2 [12]
Variates	21	862	321	8	7	7	7	7	7
Timesteps	52696	17544	26304	7,588	966	17420	17420	69680	69680
Granularity	10min	1hour	1hour	1day	1week	1hour	1hour	5min	5min

shifted stacked embedding mechanism could scan arbitrary scales of different positions and sizes, achieving thorough and in-depth multi-scale modeling.

After obtaining multi-scale representation, we further learn the correlation between scales. We decompress the basic units to restore the multi-channel unit sequence to the form of a multivariate time series. To achieve multi-scale learning, we first convert each scale into a high-dimensional feature and regard it as a token through linear projection. The dimension of embedding features is D. Next, we feed these tokens into the Transformer encoder. The self-attention mechanism within the encoder computes the relationships between tokens, thereby modeling the relationships between the multi-scale sequences. The feed-forward network linearly maps and enhances the embedded features within each token, extracting the internal features of the multi-scale sequences. Herein, the shifted stacked embedding mechanism, with its ingenious design, contributes significantly by adopting an efficient multi-scale structure for the Transformer and seamlessly integrating the two. This allows the HMSformer to effectively capture and analyze dependencies across various scales at both global and local levels. Central to this ability is the intrinsic connectivity of its multi-scale sequences and the precise mapping of these connections via the self-attention mechanism. By capturing and processing information across multiple scales, HMSformer extracts key temporal trends. It also observes how these trends interact across different time scales. For example, it examines how short-term fluctuations influence long-term trends and how local incidents impact overall data patterns. This cross-scale learning approach significantly enhances the flexibility and accuracy of the forecasting model. Additionally, through the feed-forward network, the HMSformer enhances its sensitivity to subtle differences in scale, thereby achieving higher accuracy and reliability in multivariate forecasting.

The output of Transformer encoder is then reshaped to the dimension similar to $\boldsymbol{x} \in R^{C \times L}$. Then, the outputs of multi-levels are concatenated along the time dimension. Finally, we fuse multi-levels features through a linear layer and output the prediction sequence by learning temporal features.

E. Loss Function

We adopt the MSE loss to constrain the predictions to be close to the ground truth. MSE loss is widely used in long-term series prediction task. Its formalized representation is as follows.

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}} \frac{1}{C} \sum_{c=1}^{C} \left\| \boldsymbol{y}_{L+1:L+T}^{(c)} - \boldsymbol{x}_{L+1:L+T}^{(c)} \right\|_2^2 \tag{8}$$

where $\boldsymbol{x}_{L+1:L+T}^{(c)}$, $\boldsymbol{y}_{L+1:L+T}^{(c)}$, c, and C represent the future T time steps $(x_{L+1}, \ldots, x_{L+T})$ in training set, predicted T time steps, c-th variate, and the number of variates, respectively. $c \in [1, C]$.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: Following the baselines below, we use 9 authoritative public datasets [12] [13] [22] [23] listed in TABLE I for evaluation. They are collected from six scenarios in the real world and could comprehensively verify the effectiveness of the methods.

2) Baselines: Our baselines are numerous high-impact SOTA methods of multivariate long-term forecasting, including (1) Linear-based methods: FreTS [24], DLinear [14]; (2) Transformer-based methods: MSGNet [20], PatchTST [15], FEDformer [16], Pyraformer [18], Autoformer [13], Informer [12]; and (3) CNN-based methods: TimesNet [17].

3) Implementation Details: HMSformer is implemented using PyTorch on three 3090 GPUs. We utilize the Adam optimizer with a learning rate set at 0.0001. Due to space limitations, please review the complete hyper-parameters and details in our source codes.

B. Quantitative Experiment

Following baselines [20] [24] [17], the prediction window T is set within $\{96, 192, 336, 720\}$, while the look-back window size L is set at 36 for ILI and 96 for other datasets. We calculate the MSE and mean absolute error (MAE) and show the results in TABLE II. By surpassing the limitations of existing methods in analyzing temporal relationships both in-depth and comprehensively, HMSformer outperforms SOTA methods overall with a notable advantage. Specifically, the average MSE of HMSformer (0.481) on all datasets is 9.925% lower than that of the best baseline PatchTST [15] (0.534). It could be found that on larger datasets such as Traffic and Electricity, HMSformer has a greater leading margin. For example, on Traffic, the average MSE of HMSformer is 0.458, which is much lower than 0.518 of the second place FreTS [24]. This shows that HMSformer is good at analyzing complex time series relationships and verifies the effectiveness of its shifted stacked embedding mechanism and unified hierarchical framework.

We conduct an experiment of "MSE variation with increasing look-back window sizes" to verify the long-term prediction ability of HMSformer. As shown in Fig. 2, regardless of the shortest (left) or longest (right) prediction length, the MSE decreases significantly as the look-back window size expands and increases slightly at the end. This consistent trend indicates that the HMSformer effectively leverages longer input sequences to enhance its long-term forecasting accuracy. Specifically, the usable limit length is between 192 and 336. Exceeding this limit brings in irrelevant information and reduces accuracy.

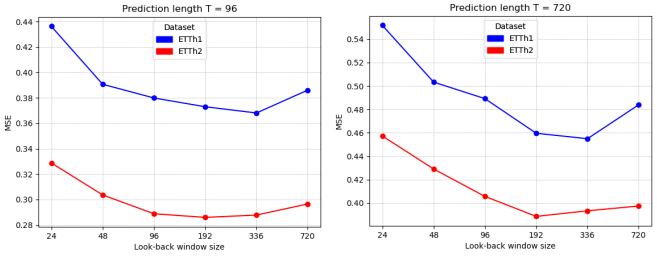

Fig. 2. MSE variation of the increasing look-back window.

C. Ablation Study

We design ablation study to verify the respective effects of the shifted stacked embedding mechanism (SSE) and the unified hierarchical framework (UHF) of HMSformer. Since the multiple levels of UHF have an influence on each other as a whole, we could not ablate each level individually, so we perform ablation on it as

TABLE II

MULTIVARIATE LONG-TERM FORECASTING RESULTS. THE BEST RESULTS ARE IN BOLD AND THE SECOND BEST ARE UNDERLINED.

Models		HMSformer		MSGNet [20]		FreTS [24]		PatchTST [15]		TimesNet [17]		DLinear [14]		FEDformer [16]		Pyraformer [18]		Autoformer [13]		Informer [12]	
Venue		-		AAAI 24		NeurIPS 23		ICLR 23		ICLR 23		AAAI 23		ICML 22		ICLR 22		NeurIPS 21		AAAI 21	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	**0.434**	**0.287**	0.609	0.349	0.486	0.308	0.526	0.347	0.593	0.321	0.650	0.396	0.587	0.366	0.867	0.468	0.613	0.388	0.719	0.391
	192	**0.443**	**0.284**	0.636	0.371	0.506	0.322	0.522	0.332	0.617	0.336	0.598	0.370	0.604	0.373	0.869	0.467	0.616	0.382	0.696	0.379
	336	**0.460**	**0.293**	0.671	0.393	0.517	0.316	0.517	0.334	0.629	0.336	0.605	0.373	0.621	0.383	0.881	0.469	0.622	0.337	0.777	0.420
	720	**0.493**	**0.310**	0.727	0.422	0.561	0.336	0.552	0.352	0.640	0.350	0.645	0.394	0.626	0.382	0.896	0.473	0.660	0.408	0.864	0.472
Electricity	96	**0.147**	**0.245**	0.165	0.274	0.164	0.258	0.190	0.296	0.168	0.272	0.197	0.282	0.193	0.308	0.386	0.449	0.201	0.317	0.274	0.368
	192	**0.167**	**0.263**	0.184	0.292	0.177	0.270	0.199	0.304	0.184	0.289	0.196	0.285	0.201	0.315	0.378	0.443	0.222	0.334	0.296	0.386
	336	**0.188**	**0.284**	0.195	0.302	0.197	0.290	0.217	0.319	0.198	0.300	0.209	0.301	0.214	0.329	0.376	0.443	0.231	0.338	0.300	0.394
	720	**0.209**	**0.302**	0.231	0.332	0.238	0.327	0.258	0.352	0.220	0.320	0.245	0.333	0.246	0.355	0.376	0.445	0.254	0.361	0.373	0.439
Weather	96	**0.160**	**0.208**	0.163	0.212	0.181	0.242	0.186	0.227	0.172	0.220	0.196	0.255	0.217	0.296	0.622	0.556	0.266	0.336	0.300	0.384
	192	**0.207**	**0.251**	0.212	0.254	0.220	0.277	0.234	0.265	0.219	0.261	0.237	0.296	0.276	0.336	0.739	0.624	0.307	0.367	0.598	0.544
	336	**0.264**	**0.293**	0.272	0.299	0.275	0.325	0.284	0.301	0.280	0.306	0.283	0.335	0.339	0.380	1.004	0.753	0.359	0.395	0.578	0.523
	720	0.344	**0.346**	0.350	0.348	**0.340**	0.377	0.356	0.349	0.365	0.359	0.345	0.381	0.403	0.428	1.420	0.934	0.419	0.428	1.059	0.741
Exchange	96	**0.080**	**0.197**	0.102	0.230	0.087	0.216	0.081	0.198	0.107	0.234	0.088	0.219	0.148	0.278	1.748	1.105	0.197	0.323	0.847	0.752
	192	**0.169**	**0.292**	0.195	0.317	0.203	0.334	0.171	0.293	0.226	0.344	0.176	0.315	0.271	0.380	1.874	1.151	0.300	0.369	1.204	0.895
	336	0.318	**0.407**	0.359	0.436	0.393	0.473	0.319	**0.407**	0.367	0.448	**0.313**	0.427	0.460	0.500	1.943	1.172	0.509	0.524	1.672	1.036
	720	0.849	0.694	0.940	0.738	1.261	0.832	**0.836**	**0.688**	0.964	0.746	0.839	0.695	1.195	0.841	2.085	1.206	1.447	0.941	2.478	1.310
ILI	24	**1.411**	**0.769**	3.993	1.411	1.952	0.936	1.800	0.855	2.317	0.934	2.398	1.040	3.228	1.260	7.394	2.012	3.483	1.287	5.764	1.677
	36	**1.599**	**0.819**	3.728	1.310	2.187	1.008	1.658	0.823	1.972	0.920	2.646	1.088	2.679	1.080	7.551	2.031	3.103	1.148	4.755	1.467
	48	2.080	0.888	3.379	1.259	2.474	1.057	**2.003**	**0.879**	2.238	0.940	2.614	1.086	2.622	1.078	7.662	2.057	2.669	1.085	4.763	1.469
	60	**1.375**	**0.769**	4.055	1.430	2.784	1.126	1.917	0.908	2.027	0.928	2.804	1.146	2.857	1.157	7.931	2.100	2.770	1.125	5.264	1.564
ETTh1	96	0.387	**0.399**	0.390	0.411	0.394	0.409	0.460	0.447	0.384	0.402	0.386	0.400	**0.376**	0.419	0.664	0.612	0.449	0.459	0.865	0.713
	192	0.438	**0.427**	0.442	0.442	0.448	0.441	0.512	0.477	0.436	0.429	0.437	0.432	**0.420**	0.448	0.790	0.681	0.500	0.482	1.008	0.792
	336	0.481	**0.446**	0.480	0.468	0.507	0.481	0.546	0.496	0.491	0.469	0.481	0.459	**0.459**	0.465	0.891	0.738	0.521	0.496	1.107	0.809
	720	**0.491**	**0.473**	0.494	0.488	0.551	0.527	0.544	0.517	0.521	0.500	0.519	0.516	0.506	0.507	0.963	0.782	0.514	0.512	1.181	0.865
ETTh2	96	**0.291**	**0.338**	0.328	0.371	0.340	0.390	0.308	0.355	0.340	0.374	0.333	0.387	0.358	0.397	0.645	0.597	0.346	0.388	3.755	1.525
	192	**0.368**	**0.385**	0.402	0.414	0.453	0.461	0.393	0.405	0.402	0.414	0.477	0.476	0.429	0.439	0.788	0.683	0.456	0.452	5.602	1.931
	336	**0.367**	**0.398**	0.435	0.443	0.477	0.482	0.427	0.436	0.452	0.452	0.594	0.541	0.496	0.487	0.907	0.747	0.482	0.486	4.721	1.835
	720	**0.409**	**0.432**	0.417	0.441	0.754	0.625	0.436	0.450	0.462	0.468	0.831	0.657	0.463	0.474	0.963	0.783	0.515	0.511	3.647	1.625
ETTm1	96	0.332	0.368	**0.319**	**0.366**	0.346	0.387	0.352	0.374	0.338	0.375	0.345	0.372	0.379	0.419	0.543	0.510	0.505	0.475	0.672	0.571
	192	**0.365**	0.388	0.376	0.397	0.412	0.441	0.390	0.393	0.374	**0.387**	0.380	0.389	0.426	0.441	0.557	0.537	0.553	0.496	0.795	0.669
	336	**0.400**	0.412	0.417	0.422	0.435	0.445	0.421	0.414	0.410	**0.411**	0.413	0.413	0.445	0.459	0.754	0.655	0.621	0.537	1.212	0.871
	720	0.475	0.453	0.481	0.458	0.553	0.534	**0.462**	**0.449**	0.478	0.450	0.474	0.453	0.543	0.490	0.908	0.724	0.671	0.561	1.166	0.823
ETTm2	96	**0.174**	**0.259**	0.177	0.262	0.186	0.278	0.183	0.270	0.187	0.267	0.193	0.292	0.203	0.287	0.453	0.507	0.255	0.339	0.365	0.453
	192	**0.240**	**0.304**	0.247	0.307	0.257	0.325	0.255	0.314	0.249	0.309	0.248	0.362	0.269	0.328	0.730	0.673	0.281	0.340	0.533	0.563
	336	**0.299**	**0.341**	0.312	0.346	0.320	0.364	0.309	0.347	0.321	0.351	0.369	0.427	0.325	0.366	1.201	0.845	0.339	0.372	1.363	0.887
	720	**0.392**	**0.396**	0.414	0.403	0.518	0.497	0.412	0.404	0.408	0.403	0.554	0.522	0.421	0.415	3.625	1.451	0.433	0.432	3.379	1.338

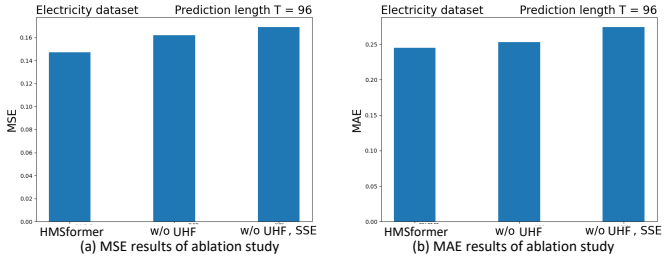

Fig. 3. Intuitive ablation study results. UHF and SSE represent the unified hierarchical framework and shifted stacked embedding mechanism, respectively.

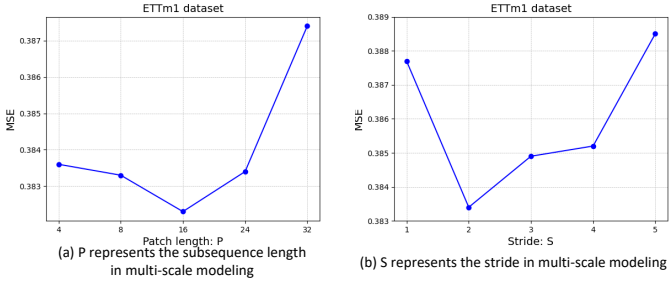

Fig. 4. Hyper-parameter ablation study results. A compromise parameter value is optimal.

TABLE III

DETAILED ABLATION STUDY RESULTS. THE BEST RESULTS ARE IN BOLD AND THE SECOND BEST ARE UNDERLINED.

Models		HMSformer		w/o UHF		w/o (UHF, SSE)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	**0.174**	**0.259**	0.179	0.261	0.216	0.296
	192	**0.240**	0.304	0.242	**0.302**	0.279	0.334
	336	**0.299**	**0.341**	0.307	0.343	0.339	0.370
	720	**0.392**	**0.396**	0.400	0.396	0.436	0.420
	AVG	**0.276**	**0.325**	0.282	0.326	0.317	0.355
Exchange	96	**0.080**	**0.197**	0.084	0.204	0.105	0.229
	192	**0.169**	**0.292**	0.171	0.294	0.192	0.314
	336	**0.318**	**0.407**	0.327	0.414	0.372	0.446
	720	**0.849**	**0.694**	0.849	0.694	0.941	0.737
	AVG	**0.354**	**0.398**	0.358	0.401	0.403	0.431

a whole. In addition, UHF is attached on top of SSE, so we could not ablate SSE alone. We present the intuitive visualization results in Fig. 3 and the detailed results in TABLE III. As shown in the MSE and MAE results in Fig. 3, when removing UHF (w/o UHF), both MSE and MAE increase, with MSE increasing more. This indicates that UHF is very effective and contributes more to reducing MSE. When further removing SSE (w/o UHF, SSE), both MSE and MAE increase further, with MAE increasing more. This indicates that SSE is also very effective and contributes more to reducing MAE. The results in TABLE III confirm these conclusions. The HMSformer with both UHF and SSE has the lowest overall error. The next is the one without UHF (w/o UHF). The largest error is that of the one without both UHF and SSE (w/o UHF, SSE). This also proves the effectiveness of both.

To explore the effects of parameters P and S in multi-scale modeling, we conducted a hyper-parameter ablation study on the ETTm1 dataset and calculated the average results across all prediction lengths. As shown in Fig. 4(a), given the input length of 96, we reasonably set P to range from a smaller value of 4 to a larger value of 32. The MSE decreases progressively with an increase in P, reaching its lowest at 16, before rapidly increasing. This is because subsequence lengths that are too short or too long respectively lead to too short unit sequence for multi-scale modeling at the detail and

local levels, while a moderate P is optimal as it achieves a balance.

Due to the influence of P on the range of S, we reasonably set S from 1 to 5. Fig. 4(b) displays a similar phenomenon, where a very small S hinders modeling of scales at different positions, and a very large S hinders modeling of scales of different sizes. Therefore, a moderate S is optimal as it balances both aspects.

TABLE IV
OVERHEAD COMPARISON OF THE TRAINING TIME (S/EPOCH), PARAMETER NUMBER (M), AND GPU MEMORY UTILIZATION (GB). THE BEST RESULTS ARE IN BOLD AND THE SECOND BEST ARE UNDERLINED.

Models		HMSformer			MSGNet			TimesNet		
Venue		-			AAAI 24			ICLR 23		
Metric		Time	Param	GPU	Time	Param	GPU	Time	Param	GPU
Exchange	96	**4.3**	**0.2**	**1.8**	11.3	1.9	2.7	9.2	4.7	3.5
	192	**4.5**	**0.4**	**1.8**	14.4	3.1	3.3	10.8	4.7	3.4
	336	**4.2**	**0.2**	**1.8**	14.3	3.1	3.3	9.5	1.2	2.4
	720	11.0	**0.4**	**1.6**	13.6	3.2	2.0	**9.9**	1.3	2.0
	AVG	**6.0**	**0.3**	**1.8**	13.4	2.8	2.8	9.9	3.0	2.8
ETTh1	96	**11.4**	**0.3**	2.4	11.7	0.9	**2.1**	19.2	0.6	3.0
	192	**11.2**	**0.5**	2.4	11.6	1.0	**2.1**	18.8	0.6	2.3
	336	**11.1**	0.7	2.4	18.2	1.9	2.6	19.1	**0.6**	**2.1**
	720	**10.3**	1.5	2.4	10.9	1.0	**2.1**	20.1	**0.7**	3.0
	AVG	**11.0**	0.7	2.4	13.1	1.2	**2.2**	19.3	**0.6**	2.6
ETTh2	96	**11.3**	**0.3**	**2.4**	24.3	3.0	3.1	15.3	1.2	3.3
	192	**11.5**	**0.5**	**2.4**	24.7	3.0	3.1	13.7	1.2	2.8
	336	**10.8**	**0.7**	2.4	24.2	3.1	3.1	14.8	1.2	**2.3**
	720	**10.5**	**1.5**	**2.4**	23.0	3.1	3.1	14.9	1.3	3.7
	AVG	**11.0**	**0.7**	**2.4**	24.0	3.1	3.1	14.7	1.2	3.0

D. Resource Overhead Analysis

TABLE IV shows a detailed comparison of the time-space overheads between HMSformer and SOTA multi-scale methods (MSGNet [20], TimesNet [17]), considering training time per epoch, parameter count, and GPU memory utilization. Since GPU memory utilization is greatly affected by batch size, we unify the batch size of all methods for a fair comparison. The results highlight HMSformer's superior computational efficiency, with an absolute advantage in overall metrics.

Notably, HMSformer achieves significantly lower training times, even several times smaller than the comparison methods. For example, on the Exchange dataset with a prediction length of 96, HMSformer takes 4.3 s/epoch. In comparison, MSGNet takes 11.3 s/epoch and TimesNet takes 9.2 s/epoch. This remarkable efficiency extends to its parameter count, where HMSformer maintains a much smaller model size, such as 0.3 M parameters for the Exchange dataset with an average prediction length, compared to 2.8 M for MSGNet and 3.0 M for TimesNet, ensuring reduced computational complexity without compromising performance.

Additionally, HMSformer shows low GPU memory utilization, such as 2.4 GB on the ETTh2 dataset with an average prediction length, significantly less than MSGNet (3.1 GB) and TimesNet (3.0 GB). These findings underline HMSformer's efficiency and scalability. Combined with the high performance of HMSformer, it is a suitable choice for multivariate long-term series forecasting in various practical scenarios.

V. CONCLUSION

In this paper, we propose HMSformer, a hierarchical and multi-scale Transformer designed to thoroughly capture the long-term multi-scale characteristics of multivariate time series. By the novel shifted stacked embedding mechanism, HMSformer captures scale information at varying positions and sizes. This enables effective modeling of arbitrary scales and addresses the limitations of existing methods in deeply analyzing temporal relationships. Furthermore, HMSformer extends this mechanism to multiple levels of temporal relationships through a unified hierarchical framework. Consequently, it could comprehensively analyze temporal relationships, overcoming the lack of completeness in analysis of existing methods. HMSformer achieves an average MSE that is 9.925% lower than the baseline across nine authoritative datasets, showcasing its superior performance and robustness in multivariate long-term forecasting.

REFERENCES

[1] Tim Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[2] Robert F Engte, "Autoregressive conditional heteroscedasticity with estimates of the variance of uk inflations," *Econometrica*, vol. 59, pp. 987–1007, 1982.

[3] Lon-Mu Liu, "Identification of seasonal arima models using a filtering method," *Communications in Statistics-Theory and Methods*, vol. 18, no. 6, pp. 2279–2288, 1989.

[4] Keith Ord and Sam Lowe, "Automatic forecasting," *The American Statistician*, vol. 50, no. 1, pp. 88–94, 1996.

[5] Wei-Chiang Hong, "Hybrid evolutionary algorithms in a svr-based electric load forecasting model," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 7-8, pp. 409–417, 2009.

[6] Rob J Hyndman and Yeasmin Khandakar, "Automatic time series forecasting: the forecast package for r," *Journal of statistical software*, vol. 27, pp. 1–22, 2008.

[7] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder, *Forecasting with exponential smoothing: the state space approach*, Springer Science & Business Media, 2008.

[8] Alex Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, pp. 132306, 2020.

[9] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, pp. 11106–11115.

[13] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22419–22430, 2021.

[14] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, pp. 11121–11128.

[15] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[16] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27268–27286.

[17] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[18] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International conference on learning representations*, 2021.

[19] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," *arXiv preprint arXiv:2405.14616*, 2024.

[20] Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu, "Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, vol. 38, pp. 11141–11149.

[21] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.

[22] "Weather," https://www.bgc-jena.mpg.de/wetter/.

[23] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95–104.

[24] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu, "Frequency-domain mlps are more effective learners in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, 2024.