Received 19 March 2025, accepted 14 April 2025, date of publication 21 April 2025, date of current version 2 May 2025.
Digital Object Identifier 10.1109/ACCESS.2025.3563070

RESEARCH ARTICLE

DiffTST: Diff Transformer for Multivariate Time Series Forecast

SONG YANG, WENYONG HAN, YAPING WAN, TAO ZHU, (Senior Member, IEEE), ZHIMING LIU, (Member, IEEE), AND SHUANGJIAN LI
School of Computer Science, University of South China, Hengyang 421001, China
Corresponding author: Yaping Wan (512828758@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62006110, in part by the Natural Science Foundation of Hunan Province under Grant 2024JJ7428 and Grant 2023JJ30518, and in part by the Scientific Research Project of Hunan Provincial Department of Education under Grant 24B0385.

ABSTRACT Deep learning models employing the Transformer architecture have demonstrated exceptional performance in the field of multivariate time series forecasting research. However, these models often incorporate irrelevant or weakly relevant information during the processing of time series, leading to noise. This phenomenon diverts the attention mechanism from crucial features within the time series, thereby impacting the overall forecasting performance. To mitigate this issue, our study introduces DiffTST, which employs a Differential Transformer to enhance the model's focus on relevant context within the time series, thereby mitigating the influence of noise on forecasting accuracy. The model utilizes independent channels to process time series data, ensuring that each input token contains information from a single channel exclusively. Furthermore, each channel is segmented into multiple patches to facilitate the extraction of local information. Subsequently, the Differential Transformer module is employed to process the sequence features of these patches, alleviating the tendency of Transformer-based models to allocate excessive attention to irrelevant sequence information. Ultimately, the forecast outcomes are derived through a Multi-Layer Perceptron. Our findings indicate that DiffTST achieves higher or comparable long-term forecasting accuracy compared to the current state-of-the-art Transformer-based models. On the main datasets (Weather, Traffic, Electricity), our method reduces MSE by 0.008, 0.087, and 0.023 and MAE by 0.004, 0.069, and 0.025 compared to PatchTST.

INDEX TERMS Time series forecasting, differential transformer, deep learning, independent channels.

I. INTRODUCTION

Multivariate time series (MTS) forecasting [1] has emerged as a significant research area in the fields of statistics and machine learning, finding applications across various domains including economics and finance, energy scheduling and management, climate forecasting and disease spread modeling, and transportation planning. The proliferation of data collection and storage technologies has enabled the generation of a vast amount of time series data, serving as a robust foundation for MTS forecasting. In real-world scenarios, numerous phenomena are influenced by multiple interconnected factors, such as the interplay between economic indicators or the relationship between temperature and humidity in climate change. MTS forecasting seeks to identify the dynamic relationships between the historical values of multiple variables, thereby enhancing forecast accuracy and reliability.

The advancement of deep learning and improvements in computing performance have led to the proposal of numerous convolutional neural network (CNN), recurrent neural network (RNN), and Transformer-based models for time series forecasting. However, each of these models has limitations in sequence modeling. CNNs are primarily focused on local features and struggle to capture long-term dependencies. RNNs are capable of handling long-term dependencies

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

and dynamically updating based on the current input and historical hidden states, but they may encounter issues such as gradient vanishing or exploding during training. Transformers have been successful in natural language processing (NLP), computer vision (CV), and speech processing due to their attention mechanism, which has a strong ability to capture relationships and sequence correlations. In recent years, several Transformer-based models have been proposed for multivariate time series forecasting and have shown great potential. A substantial number of Transformer-based models have emerged, achieving significant breakthroughs in modeling long multivariate time series. Despite these advancements, existing Transformer-based models often overemphasize irrelevant context while neglecting important temporal information due to the self-attention mechanism. This suggests that there are limitations to the application of Transformers in MTS forecasting tasks.

Recent research [2] indicates that the Differential Transformer (Diff Transformer) offers significant advantages over the traditional Transformer model in practical applications, including better long-context modeling, enhanced key information retrieval, improved context learning, and reduced activation outliers. The Diff Transformer introduces a differential attention mechanism to mitigate attention noise through differential denoising, thereby encouraging the model to concentrate on crucial information. Motivated by these findings, we adopted the Diff Transformer model for time series forecasting tasks and achieved promising results. Initially, we reversibly instantiated the time series data and divided each channel into patches, with each patch encapsulating both channel and time series information. Subsequently, we applied position embeddings to each patch, transforming them into linear vectors and feeding them into the Diff Transformer blocks, followed by a forecast phase using a Multi-Layer Perceptron. In long sequence forecast tasks, the Diff Transformer block enhances the model's capability to extract key contextual information, offering new insights for constructing an effective and efficient time series infrastructure.

The main contributions of this paper can be summarized as follows:

- We propose a novel sequence modeling architecture, named DiffTST, which represents a pioneering approach to mitigating contextual noise in time series forecast tasks by employing a differential attention mechanism.
- The DiffTST framework employs an independent channel approach and patch operations to process time series data, modeling the global context and position embeddings of time series patches through the Diff Transformer.
- Experimental evaluations on four widely adopted benchmark datasets demonstrate that DiffTST exhibits superior recognition performance compared to alternative frameworks.

The remainder of this paper is structured as follows: Section II reviews related work, Section III provides a comprehensive description of the adopted model architecture and its components, and Section IV presents experimental results evaluating the DiffTST model on four public datasets while visualizing its prediction performance. Finally, the concluding section discusses the research findings and outlines future work.

II. RELATED WORK

A. MULTIVARIATE TIME SERIES FORECAST METHODS

Machine learning methods [3] such as support vector machine (SVM) and random forest (RF) are widely used in multivariate time series forecast tasks. SVM usually handles complex nonlinear relationships by mapping time series data into high-dimensional space, while RF method uses ensemble learning method to improve the forecast ability of the model through voting of multiple decision trees. Traditional linear models are often difficult to fully capture the nonlinear relationships in multivariate time series data. In addition, deep learning has the characteristics of autonomy, multi-layer and diversified feature extraction [4], which provides an effective and feasible solution to time series problems. In recent years, more and more studies have used deep learning methods as the basic model of multivariate time series. Time series models based on deep learning can be roughly divided into RNN [5], CNN [6], MLP [7] and Transformer [8] based methods. The fixed convolution kernel size of CNN may hinder their ability to capture long-distance dependencies when processing long sequences. RNN-based methods have weak memory capacity, which limits long-term forecast ability when the sequence length increases. Model forecasting performance is the measure of the probability of success [9]. In recent years, multi-layer projection (MLP) has also been introduced into time series forecast, which has achieved good performance in both forecast performance and efficiency. The introduction of the Transformer model has greatly promoted the modeling research of multivariate time series. Transformer processes sequence data through the self-attention mechanism and can capture long-distance dependencies. Recent research works such as Informer [10], iTransformer [11], etc. have proposed the Transformer architecture specifically for time series data, which improves the model's ability in long-term dependency and efficient computing.

B. LONG-TERM TIME SERIES FORECAST BASED ON TRANSFORMER

In recent years, many Transformer-based models have been proposed for MTS forecast and have shown great potential. Models such as LogTrans [12], Informer [10], Reformer [13], Autoformer [14], PatchTST [15], FEDformer [16], MCformer [17], Pyraformer [18], FPPformer [19], Pathformer [20], Fredformer [21], SAMformer [22], Crossformer [23], iTransformer [11], DeformableTST [24], Timexer [25] and ElasTST [26] have emerged, which have made significant breakthroughs in multivariate long time series modeling, making them ideal for time series forecasting modeling tasks. However, Transformer-based models face limitations when processing very long time series data. The main obstacles are the quadratic time complexity of self-attention calculation and the tendency of the model to assign only a small part of the attention score to the correct answer, while paying too much attention to irrelevant context, resulting in attention noise, which limits the amount of useful information that can be extracted from each time point. To overcome these shortcomings, a large number of studies are exploring more effective attention variants, but most of them are at the expense of the effective characteristics of attention. The differential attention mechanism, as the basic architecture of large language models (LLMs), is proposed to eliminate attention noise with differential denoising, which divides the query vector and the key vector into two groups and calculates two independent softmax attention maps. Then, the difference between the two maps is regarded as the attention score. The differential attention mechanism eliminates attention noise and encourages the model to focus on key information. It is better than Transformer in key information retrieval and context learning, so it is worth trying as a time series modeling infrastructure. In this article, we also verify the effectiveness of using DIFF Transformer as the basic model for time series forcast.

III. METHODS

A. PROBLEM DEFINITION

In multivariate time series forecasting, the task goal is to simultaneously predict the future values of C time series $\mathbf{X}_{T+1:T+t} \in \mathbb{R}^{t\times C}$ given a set of historical values $\mathbf{X}_{1:T} \in \mathbb{R}^{T\times C}$, where T represents the number of time steps in the historical data, t represents the number of time steps in the future to be predicted, and C represents the number of channels. Our goal is to predict the future values of C variables in the next t time steps. We obtain the input data $\mathbf{X}_{T-L:T} \in \mathbb{R}^{C\times L}$, which represents the observations of a look-back window, where L represents the size of the look-back window and T represents the initial position of the prediction window. We use channel independence and patch division to process the observation values of the look-back window, and then use a Differential Transformer decoder to improve the attention between related patch blocks and eliminate attention noise, thereby improving the robustness of the model. Finally, we obtain the predicted values $\mathbf{X}_{T+1:T+t}$ through multi-layer linear mapping. The overall architecture of the model is shown in Fig. 1.

B. PATCHING AND CHANNEL INDEPENDENCE

Before the observations $X_{T-L:T}$ are fed into the channel-independent Patching module, Reversible instance Normalization (RevIN) [27] is applied to the data of each channel to address the issue of uneven temporal distribution between training and test data. RevIN employed in DiffTST is executed independently for each channel, with the means and standard deviations computed exclusively from historical data (i.e., the input look-back window $X_{T-L:T}$), excluding any future predictive data. This design ensures that no future information is leaked during prediction, adhering to the causality requirements of time series forecasting. For each channel $\mathbf{x}^i = \left[x_1^i, x_2^i, \cdots, x_L^i\right]$ (where L denotes the look-back window size), the mean $Mean(\mathbf{x^i}) = \frac{1}{L}\sum_{j=1}^{L} x_j^i$ and standard deviation

$$\sqrt{Var(\mathbf{x^i})} = \sqrt{\frac{1}{L}\sum_{j=1}^{L}(x_j^i - Mean(\mathbf{x^i}))^2}$$

are calculated. Subsequently, the normalization formula is:

$$RevIN(\mathbf{x}^i) = \left\{ \gamma_i \frac{\mathbf{x}^i - Mean(\mathbf{x}^i)}{\sqrt{Var(\mathbf{x}^i) + \varepsilon}} \right\}, i = 1, 2, \cdots, M \quad (1)$$

where ε is a small constant (typically set to 10^{-5}) to prevent division by zero. Upon completion of the prediction, the predicted values are reverted to their original scale through a reverse operation utilizing the same means and standard deviations. This methodology ensures consistent statistical properties for the input data of each channel during both training and inference phases, while precluding information conflation across channels or time points. A single channel is represented as $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_t^i]$, where the mean and standard deviation are calculated for each instance x_t^i. After the prediction results are obtained, these non-stationary information components are incorporated back into the prediction value.

We adopt a channel-independent (CI) strategy to flatten data from C channels. The input data $X_{T-L:T}$ is a $C \times L$ matrix, where C represents the number of channels and L denotes the look-back window size. The Flatten operation unfolds this two-dimensional matrix into a one-dimensional vector X_F of length LC, following a row-major order. Specifically, for C univariate time series $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^C$(each of length L), Flatten concatenates them sequentially into a long vector $X_F = \left[x_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_L^1, x_1^2, \mathbf{x}_2^2, \ldots, \mathbf{x}_L^2, \ldots, \mathbf{x}_1^C, x_2^C, \ldots, \mathbf{x}_L^C\right]$. This operation effectively transforms C parallel time series into a single, extended sequence of length LC. However, we do not directly input X_F as a monolithic entity; rather, adhering to a channel-independent strategy, we re-partition it into C independent univariate sequences, each retaining a length of L. The function of Flatten is to provide a standardized format for subsequent channel-independent processing, while preserving the independence of each channel during actual modeling. Our channel-independent strategy treats the C channels as C distinct time series samples, rather than a unified multivariate sequence. In practice, X_F is logically segmented back into C subsequences $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^C$ (each corresponding to an original channel), and patching is performed independently on each subsequence. This design obviates direct cross-channel dependency modeling, instead capturing potential inter-channel relationships indirectly through the subsequent differential Transformer module. This approach allows the model to focus on the local temporal dynamics of each channel, while maintaining computational efficiency and

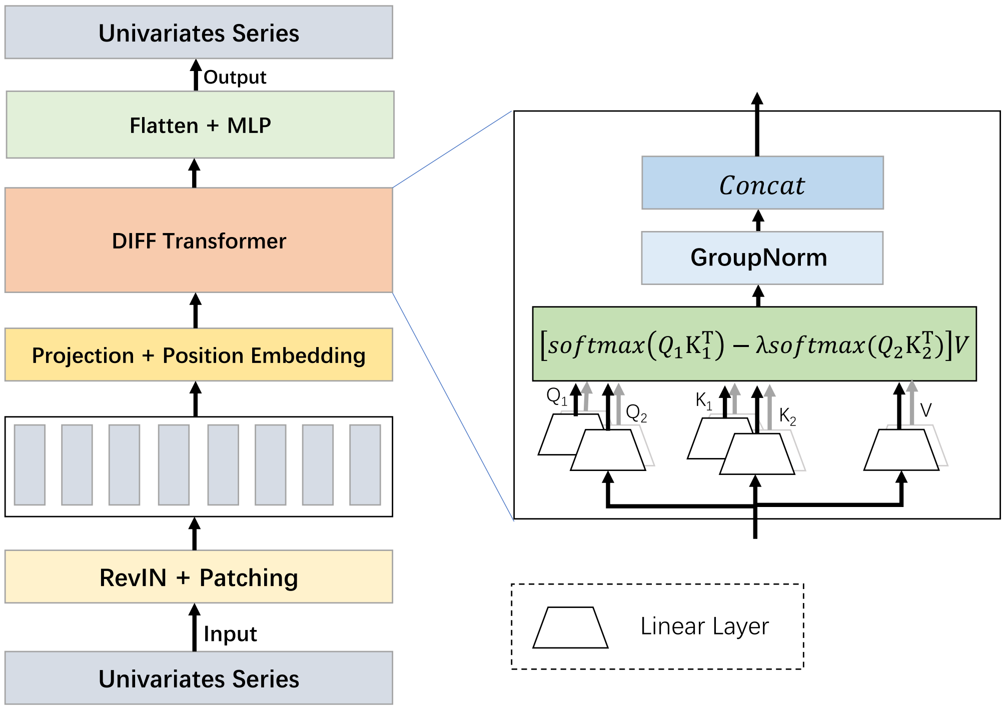


FIGURE 1. Illustrates the DiffTST architecture, where multivariate time series data is decomposed into independent channels and segmented into patches for local feature extraction. This design, combined with the Differential Transformer Decoder, enhances the model's ability to capture key temporal dependencies while reducing attention noise, culminating in accurate predictions via a linear mapping layer.

high adaptability to multivariate data. Each channel sequence $\mathbf{x}^i$ is partitioned into multiple patches. The patch length is p, the stride is S, and the number of patches $N = \lfloor (L-p)/S \rfloor + 2$. A single-layer multilayer perceptron is used to project each patch, this can be expressed as follows:

$$\mathcal{P}^i = \text{Projection}(\text{Patch}(\mathbf{x}^i)) \in \mathbb{R}^{P \times N} \tag{2}$$

where P is the projected dimension. This patching aggregation effectively captures local information within the sequences, while the projection integrates dependencies between adjacent time steps. Finally, these patch sequences are input into the differential Transformer for further modeling of global context and temporal dependencies, thereby generating the predictive outputs.

C. DIFF TRANSFORMER DECODER

Inspired by the DIFF Transformer [2], we adopt its architecture for temporal modeling, which consists of L Diff Transformer layers stacked together. We input the patch sequence into the embedding layer to obtain the embedding vector $X^0 = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{N \times d_{\text{model}}}$, where d_{model} represents the hidden dimension of the model. This vector is then contextualized to obtain $X^l = \text{Decoder}(X^{l-1}), l \in [1, L]$. Each layer consists of two modules: a differential attention module followed by a feed-forward network module. Differential attention is used to replace the traditional attention mechanism, which maps query, key, and value vectors to the output. We use the query and key vectors to calculate the attention score, and then calculate the weighted sum of the value vector, using a pair of softmax functions to eliminate the noise of the attention score. Given the input $X \in \mathbb{R}^{N \times d_{\text{model}}}$, we first project them into queries, keys, and values $Q_1, Q_2, K_1, K_2 \in \mathbb{R}^{N \times d}, V \in \mathbb{R}^{N \times 2d}$. Then the differential attention operator DiffAttn$(\cdot)$ computes the output in the following way:

$$[Q_1; Q_2] = XW^Q, \quad [K_1; K_2] = XW^K, \quad V = XW^V$$
$$\text{DiffAttn}(X) = \left(\text{softmax}\left(\frac{Q_1 K_1^T}{\sqrt{d}}\right) - \lambda\, \text{softmax}\left(\frac{Q_2 K_2^T}{\sqrt{d}}\right)\right) V \tag{3}$$

Where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times 2d}$ are parameters, and λ is a learnable scalar. In order to learn the dynamics synchronously, the scalar λ is reparameterized as:

$$\lambda = \exp(\lambda_{\mathbf{q_1}} \cdot \lambda_{\mathbf{k_1}}) - \exp(\lambda_{\mathbf{q_2}} \cdot \lambda_{\mathbf{k_2}}) + \lambda_{\text{init}} \tag{4}$$

$\lambda_{\mathbf{q_1}}, \lambda_{\mathbf{k_1}}, \lambda_{\mathbf{q_2}}, \lambda_{\mathbf{k_2}} \in \mathbb{R}^d$ are learnable vectors and $\lambda_{\text{init}} \in (0, 1)$ is a constant used to initialize λ.

A multi-head mechanism is used in the differential transformer. Let h denote the number of attention heads. We use different projection matrices $W_i^Q, W_i^K, W_i^V, i \in [1, h]$ for the heads. The scalar λ is shared between heads within the same layer. Then, the head output is normalized and projected to the final result as follows:

$$\text{head}_i = \text{DiffAttn}(X; W_i^Q, W_i^K, W_i^V, \lambda)$$
$$\overline{\text{head}_i} = (1 - \lambda_{\text{init}}) \cdot \text{LN}(\text{head}_i)$$
$$\text{MultiHead}(X) = \text{Concat}(\overline{\text{head}_1}, \ldots, \overline{\text{head}_h}) W^O \tag{5}$$

where λ_{init} is the constant scalar in Equation (4), $W^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ is a learnable projection matrix. $LN(\cdot)$ uses RMSNorm [28] for each head and $Concat(\cdot)$ concatenates the heads together along the channel dimension. We use a fixed multiplier $(1-\lambda_{init})$ as the scale of $LN(\cdot)$ to align the gradients with the transformer.

The overall architecture stacks L layers, each of which contains a multi-head differential attention module and a feedforward network module. We describe the differential Transformer layer as follows:

$$Y^l = \text{MultiHead}(\text{LN}(X^l)) + X^l$$
$$X^{l+1} = \text{SwiGLU}(\text{LN}(Y^l)) + Y^l \tag{6}$$

Among them, $LN(\cdot)$ is RMSNorm, $\text{SwiGLU}(X) = (\text{swish}(XW^G) \odot XW_1)W_2$, and $W^G, W_1 \in \mathbb{R}^{d_{\text{model}} \times \frac{8}{3} d_{\text{model}}}$, $W_2 \in \mathbb{R}^{\frac{8}{3} d_{\text{model}} \times d_{\text{model}}}$ is a learnable matrix.

D. LOSS FUNCTION

We chose MSE (Mean Squared Error) and MAE (Mean Absolute Error) losses to evaluate the disparity between the model's forecastings and the actual values. MSE measures the model performance by calculating the average of the squared differences between the predicted and actual values:

$$\text{MSE} = \frac{1}{h} \sum_{i=1}^{h} (y_i - \hat{y}_i)^2 \tag{7}$$

MAE assesses the model performance by computing the average of the absolute differences between the predicted and actual values:

$$\text{MAE} = \frac{1}{h} \sum_{i=1}^{h} |y_i - \hat{y}_i| \tag{8}$$

IV. EXPERIMENTS

In this section, we provide a comprehensive overview of our experiments and utilize cutting-edge research to evaluate our proposed model. First, we introduce the datasets and baseline models employed in our study. Subsequently, we conduct experiments to assess the effectiveness of our approach.

Our approach is implemented using PyTorch and is trained on an RTX 4090 GPU, with a batch size set to 128. We employ the Adam optimizer, maintaining a learning rate of 0.0001. The fully connected dropout rate is configured at 0.05, the default patch length is set to 16, and the stride is 8. The model utilizes 8 attention heads and consists of 7 decoder layers, and it is trained for 100 epochs.

To facilitate reproducibility and further research, the source code for DiffTST, including data preprocessing, model implementation, and training scripts, is provided as supplementary material and is available at https://github.com/YS-mker/DiffTST.

A. DATASET

We evaluate our model on five mature benchmark datasets for long-term time series forecasting, including Electricity, Traffic, Weather [14], Solar-Energy [29]. These datasets are commonly used as benchmarks for multivariate time series forecasting, with detailed information about each dataset provided in Table 1.

Weather: The weather dataset was collected at approximately 1, 600 locations across the United States between 2010 and 2013, with a sampling frequency of one record every ten minutes. This dataset contains 21 channels.

Solar-Energy: The Solar-Energy dataset documents the solar power generation of a photovoltaic power station in Alabama in 2006, with readings captured every 10 minutes. Data from a total of 137 channels were collected.

Electricity: The Electricity dataset captures the hourly electricity consumption (measured in kilowatt-hours) of 321 customers from 2012 to 2014.

Traffic: The Traffic dataset encompasses road occupancy data recorded by sensors on San Francisco Bay area freeways from 2015 to 2016. Readings are logged on an hourly basis, ranging from 0 to 1. A total of 862 sensor channels are included.

B. BASELINES

In the field of time series forecast, Transformer-based deep learning models have achieved remarkable results, surpassing traditional methods in many tasks. To evaluate the performance of our proposed approach, we carefully selected a set of state-of-the-art (SOTA) multivariate time series forecasting models. From the Transformer-based models, we selected the most representative models, including InFormer [10], AutoFormer [14], FEDFormer [16], CrossFormer [23], and PatchTST [15], given their recent promising results in time series forecasting tasks. Furthermore, recognizing the recent promising results obtained by MLP-based models, we included the most prominent representatives, DLinear [7], and TiDE [30]. Additionally, acknowledging the unique advantages of CNN-based models in extracting multivariate features, we included TimesNet [31] in our evaluation.

C. MAIN RESULT

As shown in the Table 2, our proposed DiffTST model achieved the best results across all datasets. In all experimental comparisons, we obtained 10 first places and 6 second places in MSE, and 13 first places and 3 second places in MAE. This indicates that our method outperforms all compared methods. It should be noted that we have conducted extensive comparisons with Transformer-based state-of-the-art (SOTA) models such as PatchTST, CrossFormer, FEDformer and Autoformer. On four datasets, our model outperforms the above models, which fully demonstrates the effectiveness of DiffTST based on DIFF Transformer in the time series prediction task. Moreover, compared with Tide and Dlinear which are based on MLP, as well as TimesNet which is based on CNN, DiffTST still has an edge over these models.

Figure 2 illustrates the prediction performance of the DiffTST model on the electricity and traffic flow datasets.

TABLE 1. Detailed dataset descriptions. *Dim* denotes the variate number of each dataset. *Dataset Size* denotes the total number of time points in (Train, Validation, Test) split respectively. *Prediction Length* denotes the future time points to be predicted and four prediction settings are included in each dataset. *Frequency* denotes the sampling interval of time points.

Dataset	Dim	Prediction Length	Dataset Size	Frequency	Information
Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10min	Weather
ECL	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Hourly	Electricity
Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Hourly	Transportation
Solar-Energy	137	{96, 192, 336, 720}	(36601, 5161, 10417)	10min	Energy

TABLE 2. Full results on the multivariate forecasting task. We used a look-back window of length 96 for all datasets, and we used forecasting windows $h \in \{96, 192, 336, 720\}$. The best results are highlighted in **bold**, and the second-best results are underlined.

Models		DiffTST (Ours)		TiDE [30] (2023)		PatchTST [15] (2023)		TimesNet [31] (2023)		CrossFormer [23] (2023)		Dlinear [7] (2023)		FEDFormer [16] (2022)		AutoFormer [14] (2021)		InFormer [10] (2021)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.175	**0.215**	0.202	0.261	0.177	0.218	0.172	0.220	**0.158**	0.230	0.196	0.255	0.217	0.296	0.266	0.336	0.300	0.384
	192	0.222	**0.257**	0.242	0.298	0.225	0.259	0.219	0.261	**0.206**	0.277	0.237	0.296	0.276	0.336	0.307	0.367	0.598	0.544
	336	**0.277**	**0.297**	0.287	0.335	0.278	0.297	0.280	0.306	0.272	0.335	0.283	0.335	0.339	0.380	0.359	0.395	0.578	0.523
	720	**0.328**	**0.339**	0.351	0.386	0.354	0.348	0.365	0.359	0.398	0.418	0.345	0.381	0.403	0.428	0.419	0.428	1.059	0.741
	AVG	**0.251**	**0.277**	0.271	0.320	0.259	0.281	0.259	0.287	0.259	0.315	0.265	0.317	0.309	0.360	0.338	0.382	0.634	0.548
Traffic	96	**0.445**	**0.283**	0.805	0.493	0.544	0.359	0.593	0.321	0.522	0.290	0.650	0.396	0.587	0.366	0.613	0.388	0.719	0.391
	192	**0.454**	**0.285**	0.756	0.474	0.540	0.354	0.617	0.336	0.530	0.293	0.598	0.370	0.604	0.373	0.616	0.382	0.696	0.379
	336	**0.469**	**0.292**	0.762	0.477	0.551	0.358	0.629	0.336	0.558	0.305	0.605	0.373	0.621	0.383	0.622	0.337	0.777	0.420
	720	**0.504**	**0.313**	0.719	0.449	0.586	0.375	0.640	0.350	0.589	0.328	0.645	0.394	0.626	0.382	0.660	0.408	0.864	0.472
	AVG	**0.468**	**0.293**	0.761	0.473	0.555	0.362	0.620	0.336	0.550	0.304	0.625	0.383	0.610	0.376	0.628	0.379	0.764	0.416
Electricity	96	**0.168**	**0.254**	0.237	0.329	0.195	0.285	0.168	0.272	0.219	0.314	0.197	0.282	0.193	0.308	0.201	0.317	0.274	0.368
	192	**0.176**	**0.263**	0.236	0.330	0.199	0.289	0.184	0.289	0.231	0.322	0.196	0.285	0.201	0.315	0.222	0.334	0.296	0.386
	336	**0.191**	**0.280**	0.249	0.344	0.215	0.305	0.198	0.300	0.246	0.337	0.209	0.301	0.214	0.329	0.231	0.338	0.300	0.394
	720	0.237	0.318	0.284	0.373	0.256	0.337	**0.220**	**0.320**	0.280	0.363	0.245	0.333	0.246	0.355	0.254	0.361	0.373	0.439
	AVG	**0.193**	**0.279**	0.252	0.344	0.216	0.304	**0.193**	0.295	0.244	0.334	0.212	0.300	0.214	0.327	0.227	0.338	0.311	0.397
Solar-Energy	96	**0.226**	**0.267**	0.312	0.399	0.234	0.286	0.250	0.292	0.310	0.331	0.290	0.378	0.242	0.342	0.884	0.711	0.236	0.259
	192	0.253	0.286	0.339	0.416	0.267	0.310	0.296	0.318	0.734	0.725	0.320	0.398	0.285	0.380	0.834	0.692	**0.217**	**0.269**
	336	0.270	0.298	0.368	0.430	0.290	0.315	0.319	0.330	0.750	0.735	0.353	0.415	0.282	0.376	0.941	0.723	**0.249**	**0.283**
	720	0.271	**0.301**	0.370	0.425	0.289	0.317	0.338	0.337	0.769	0.765	0.356	0.413	0.357	0.427	0.882	0.717	**0.241**	0.317
	AVG	0.255	0.287	0.347	0.418	0.270	0.307	0.301	0.319	0.641	0.639	0.330	0.401	0.292	0.381	0.885	0.711	**0.236**	**0.282**

For various time series forecasting tasks, three forecast window lengths 96, 192, *and* 336 were established, allowing for a comprehensive evaluation of the model's forecasting capabilities. The figure demonstrates that the predicted values from DiffTST *orangecurve* strongly align with the true values *bluecurve* in terms of both trend and amplitude, particularly in segments exhibiting strong periodicity, where the prediction effect is notably significant. In the context of traffic flow forecasting, as the prediction window length increases, DiffTST effectively captures the primary fluctuation trend of the time series while maintaining high prediction accuracy even with longer time windows (96_336). This indicates that the model exhibits enhanced robustness in long-sequence predictions. Furthermore, in predicting the power datasets, DiffTST also showed exceptional performance, confirming the model's capacity to capture short-term dynamic changes. Although an increase in the prediction window *e.g.*, *Electricity*_96_336 leads to a slight increase in the local deviation of the prediction curve, the overall trend continues to accurately reflect the actual data.

The experimental results indicate that DiffTST demonstrates strong generalization ability when processing datasets with varying characteristics. Its outstanding performance in both short-term predictions and long-term trend capture further validates its efficacy in the domain of time series forecasting.

V. DISCUSSION AND FUTURE WORK

The experimental results presented in this paper demonstrate that the differential attention mechanism effectively mitigates

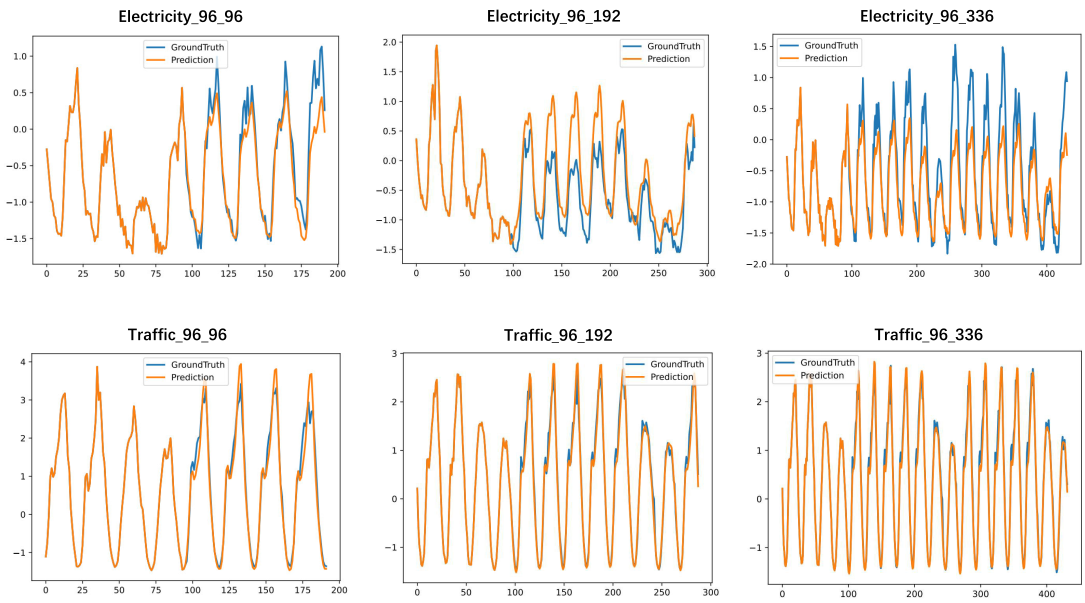


FIGURE 2. DiffTST presents the prediction performance on the Electricity and Traffic datasets with look-back window 96 and forecast horizons of 96, 192, and 336. The orange curve (predictions) closely aligns with the blue curve (actual values), demonstrating the model's robustness in capturing both short-term fluctuations and long-term trends across diverse time series.

the issue of excessive focus on irrelevant contextual information within traditional self-attention mechanisms. However, the implementation of the differential mechanism introduces additional parameters and computational complexity. The model's performance on extremely large-scale datasets warrants further investigation. While DiffTST excels in multivariate time series prediction tasks, its generalization capabilities in univariate prediction and hybrid tasks—such as classification and regression—remain to be thoroughly explored. Furthermore, the influence of varying prediction window lengths on model performance calls for a more comprehensive analysis.

The primary limitation of the current model is the computational complexity associated with Transformers. Future work will focus on enhancing the model's computational efficiency by incorporating sparse attention mechanisms, low-rank decomposition, or approximate calculation methods, enabling it to adapt to real-time prediction tasks. Moreover, the "black-box" nature of Transformer-based models limits their applicability in high-risk scenarios. Therefore, developing an interpretability framework based on DiffTST is essential to help users gain a clearer understanding of the model's decision-making process by visualizing differential attention weights and local feature contributions. Additionally, DiffTST could be extended into a multi-task learning framework to simultaneously perform time series prediction and anomaly detection tasks, thereby meeting diverse industrial needs.

VI. CONCLUSION

This paper presents a multi-variable time series prediction model, DiffTST which is based on a differential Transformer architecture designed to address the noise issue inherent in traditional Transformer models when processing time series data. DiffTST employs a differential attention mechanism that significantly mitigates the interference of irrelevant information while enhancing the model's ability to capture key features. Additionally, the model incorporates independent channel processing and local patch division strategies to further optimize time series modeling capabilities. DiffTST provides a novel perspective for time series prediction modeling by utilizing differential attention to enhance the focus on key temporal features. Experimental results demonstrate that DiffTST outperforms existing representative models across multiple public benchmark datasets, particularly in long sequence prediction tasks. This work aims to establish a foundational step by illustrating that differential attention can reduce attention noise and enhance prediction robustness. We hope that these findings will contribute to the broader field and inspire future hybrid methodologies.

While the DiffTST model demonstrates strong performance in multivariate time series forecasting, it has certain limitations. The differential Transformer architecture introduces additional parameters and computational complexity, which may pose challenges in resource-constrained environments or real-time industrial applications requiring ultra-low latency. Additionally, our experiments primarily focus on multivariate forecasting, leaving the model's generalization to univariate tasks or hybrid scenarios insufficiently explored, potentially limiting its applicability in certain contexts. As a Transformer-based model, DiffTST remains inherently opaque. Although the differential attention mechanism helps reduce irrelevant information, the model's internal computations are not easily interpretable. Future work could integrate visualization techniques (e.g., attention map analysis) or interpretability methods (e.g., SHAP analysis) to enhance model transparency.

ACKNOWLEDGMENT

(Song Yang and Wenyong Han contributed equally to this article.) This research was conducted at the School of Computer Science, University of South China. The authors wish to express their sincere gratitude to each collaborator for their invaluable guidance, ongoing evaluation, and steadfast support throughout the research process. Additionally, they employed the Big Model tool to enhance the quality of the manuscript by refining grammar and adopting a more technical approach to academic writing.

REFERENCES

[1] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 6555–6565.

[2] T. Ye, L. Dong, Y. Xia, Y. Sun, Y. Zhu, G. Huang, and F. Wei, "Differential transformer," 2024, *arXiv:2410.05258*.

[3] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, "A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks," *Future Internet*, vol. 15, no. 8, p. 255, Jul. 2023.

[4] Z. Gao, W. Dang, X. Wang, X. Hong, L. Hou, K. Ma, and M. Perc, "Complex networks and deep learning for EEG signal analysis," *Cogn. Neurodynamics*, vol. 15, no. 3, pp. 369–388, Jun. 2021.

[5] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020.

[6] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–7.

[7] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 9, pp. 11121–11128.

[8] 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. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[9] R. K. Guntu, P. K. Yeditha, M. Rathinasamy, M. Perc, N. Marwan, J. Kurths, and A. Agarwal, "Wavelet entropy-based evaluation of intrinsic predictability of time series," *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 3, Mar. 2020, Art. no. 033117.

[10] 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," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 12, pp. 11106–11115.

[11] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "ITransformer: Inverted transformers are effective for time series forecasting," 2023, *arXiv:2310.06625*.

[12] X. Nie, X. Zhou, Z. Li, L. Wang, X. Lin, and T. Tong, "LogTrans: Providing efficient local–global fusion with transformer and CNN parallel network for biomedical image segmentation," in *Proc. IEEE 24th Int. Conf. High Perform. Comput. Commun., 8th Int. Conf. Data Sci. Syst., 20th Int. Conf. Smart City, 8th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Dec. 2022, pp. 769–776.

[13] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020, *arXiv:2001.04451*.

[14] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 22419–22430.

[15] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. Int. Conf. Learn. Represent.*, Kigali, Rwanda, May 2023.

[16] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 27268–27286.

[17] W. Han, T. Zhu, L. Chen, H. Ning, Y. Luo, and Y. Wan, "MCformer: Multivariate time series forecasting with mixed-channels transformer," *IEEE Internet Things J.*, vol. 11, no. 17, pp. 28320–28329, Sep. 2024.

[18] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–8.

[19] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2022.

[20] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, "Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting," 2024, *arXiv:2402.05956*.

[21] X. Piao, Z. Chen, T. Murayama, Y. Matsubara, and Y. Sakurai, "Fredformer: Frequency debiased transformer for time series forecasting," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 2400–2410.

[22] R. Ilbert, A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko, "Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 20924–20954.

[23] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–14.

[24] D. Luo and X. Wang, "DeformableTST: Transformer for time series forecasting without over-reliance on patching," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 1–18.

[25] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu, Y. Qiu, J. Wang, and M. Long, "TimeXer: Empowering transformers for time series forecasting with exogenous variables," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 1–17.

[26] J. Zhang, S. Zheng, X. Wen, X. Zhou, J. Bian, and J. Li, "ElasTST: Towards robust varied-horizon forecasting with elastic time-series transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 1–6.

[27] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[28] B. Zhang and R. Sennrich, "Root mean square layer normalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

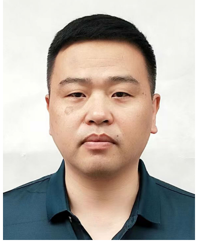
[29] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 95–104.

[30] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term forecasting with TiDE: Time-series dense encoder," 2023, *arXiv:2304.08424*.

[31] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, 2023.

SONG YANG received the bachelor's degree from the School of Information Science and Engineering, Hunan Institute of Technology, in 2018. He is currently pursuing the master's degree with the School of Computer Science, University of South China. His current research interests include deep learning and time series analysis.

WENYONG HAN received the B.S. degree from the School of Computer Science, University of South China, in 2020, where he is currently pursuing the master's degree. His current research interests include neural networks and time series analysis.



YAPING WAN received the B.S. and Ph.D. degrees from Huazhong University of Science and Technology (HUST), in 2004 and 2009, respectively. He is currently a Professor and the Dean with the School of Computer, University of South China, and the International Cooperation Research Center for Medical Big Data of Hunan Province. He has authored several books and over 40 papers in journals and at international conferences/workshops. His current research interests include intelligent nuclear security, big data analysis and causal inference, high-reliability computing, and security evaluation. He has been the Workshop Chairperson at the 16th IEEE International Conference on Big Data Science and Engineering, in 2022, and the Session Chairperson of Asian Conference on Artificial Intelligence Technology, in 2021 and 2022.

TAO ZHU (Senior Member, IEEE) received the B.E. degree from Central South University, Changsha, China, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2009 and 2015, respectively. He is currently an Associate Professor with the University of South China, Hengyang, China. He is the Principal Investigator of several projects funded by the National Natural Science Foundation of China and Science Foundation of Hunan Province. His research interests include the IoT, pervasive computing, assisted living, and evolutionary computation. He is also the Chair of the IEEE CIS Smart World Technical Committee Task Force on "User-Centred Smart Systems."

ZHIMING LIU (Member, IEEE) is currently a Professor and a Master's Supervisor with the University of South China. He is also a representative and a Senior Member of ACM and CCF and the Director of the Computer Education Professional Committee of Hunan Higher Education Society of China.

SHUANGJIAN LI received the B.E. degree from Jilin University of Finance and Economics, in 2022. He is currently pursuing the M.S. degree with the School of Computer Science, University of South China. His research interests include intelligent perception and pattern recognition.