UnetTSF: A Better Performance Linear Complexity Time Series Prediction Model

1st Chu Li
University of Science and Technology of China
Hefei, China
lichuzm@mail.ustc.edu.cn

2nd Bingjia Xiao
Institute of Plasma Physics, Hefei Institutes of Physical Science
Chinese Academy of Sciences
Hefei, China
bjxiao@ipp.ac.cn

3rd Qingping Yuan
Institute of Plasma Physics, Hefei Institutes of Physical Science
Chinese Academy of Sciences
Hefei, China
qpyuan@ipp.ac.cn

Abstract—Recently, Transformer-base models have made significant progress in the field of time series prediction which have achieved good results and become baseline models beyond Dlinear. The paper proposes an U-Net time series prediction model (UnetTSF) with linear complexity, which adopts the U-Net architecture.We are the first to use FPN technology to extract features from time series data, replacing the method of decomposing time series data into trend and seasonal terms, while designing a fusion structure suitable for time series data. After testing on 8 open-source datasets, compared to the best linear model DLiner, Out of 32 testing projects, 31 achieved the best results,The average decrease in mse is 10.1%, while the average decrease in mae is 9.1%. Compared with the complex transformer-base PatchTST, UnetTSF obtained 9 optimal results for mse and 15 optimal results for mae in 32 testing projects.Code is available at https://github.com/lichuustc/UnetTSF.

I. INTRODUCTION

As is well known,time series prediction has always been one of the hot areas of deep learning research. time series prediction has important applications in transportation, energy, weather, finance, and other fields [23] [24]. In actual production and life, there is a lot of demand for long time series prediction. To meet the needs of production and life, researchers have launched various machine learning algorithms, and many excellent deep learning algorithms have emerged. From the traditional machine learning algorithm ARMA [33], GBRT to recursive neural networks, causal time networks, etc.

Transformer [3] is currently one of the most successful sequence modeling architectures in the field of machine learning, exhibiting unparalleled performance in various applications such as natural language processing (NLP), speech recognition, and computer vision. In recent years, transformer-base time series prediction models have also emerged in large numbers and achieved good results, such as PatchTST [26], ETSformer, Autoformer [5], FEDformer [6], etc, However, Transformer-base models generally have the drawbacks of having multiple model parameters, high computational complexity, and long

inference time. Therefore, Researchers have proposed a linear complexity model (DLlinear [1]). The DLinear [1] decomposes data into seasonal and trend terms, predicts them separately and adds them up. DLlinear [1] is a model composed of two fully connected layers, with extremely low model parameters and computational complexity. However, its prediction performance significantly surpasses many complex models such as Autoformer [5], FEDformer [6], and lightTST [11], making it an important baseline model in the field of time series prediction.

Preprocessing temporal data is one of the methods to improve the predictive ability of models, among which data decomposition and normalization are the most commonly used methods. The earliest origin of data decomposition is the ARMA [33], which decomposes time series data into four parts: trend factor (T), cycle factor (C), seasonal factor (S), and random factor (I), which is more in line with human perception of data. Decomposing the data into trend and seasonal terms is a simplification of these four parts. As shown in Figure 1,Traditional machine learning is based on raw data, extracting features from the raw data and predicting future data, such as GBRT, Informar [4], Linear [1] and other models [34] [29]. There is another data decomposition strategy: to decompose time series data into trend and seasonal items. The PatchTST [26] and DLlinear [1] decompose time series data into trend and seasonal terms, extract and predict the two separately, and add them up to output the prediction results. Autoformer [5] and FEDformer [6] use feature fusion models to fuse trend and seasonal features, and use the fused features to predict the future. After verification by many scientific researchers, binary decomposition is a very suitable method for deep learning. However, the data binary decomposition method has two problems:

- The seasonal and trend characteristics of data items are fundamentally related. Simply and roughly decomposing the data into trend and seasonal items will cause the
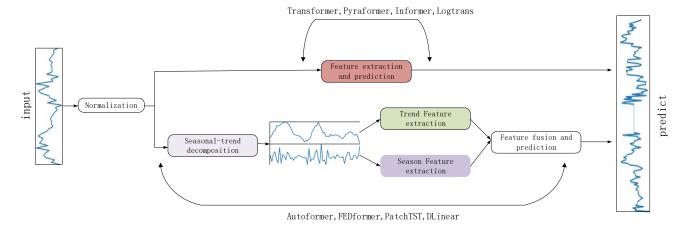
Fig. 1. The pipeline of existing Model TSF solutions.

seasonal items to lose their trend characteristics, while the trend characteristics will also lose their seasonal characteristics
- The trend and season terms enter their respective feature extraction modules for feature extraction and prediction, followed by prediction feature fusion. Currently, feature fusion mainly uses simple addition to output prediction results, and there is no correlation feature between the two for processing

After DLiner [1], transformer-base time series prediction models have once again become mainstream. However, these models suffer from high training resource consumption and slow inference speed. In order to solve these problems, this paper proposes a U-Net time series prediction model (UnetTSF). The main three contributions of this paper are as follows:

- **Time series FPN method**: This article proposes an FPN [12] method for describing time series data, using pooling functions to perform multi-layer operations on the data to extract trend information at different depths and jointly form a data group to replace the original data. Compared to the commonly used binary decomposition method in the field of temporal data, the FPN [12] method uses small pooling kernels for data processing, which has lower computational complexity and is more effective in extracting temporal shallow and deep features. Shallow features include seasonal and trend terms, and as depth increases, seasonal features are gradually removed, retaining more trend features.
- **UnetTSF**:For the first time, we introduce the U-Net structure into the field of time series prediction. We combine the multi-level prediction characteristics of the U-Net network with time series data sets to predict data in the same level dimension. Through multi-step fusion, we integrate higher depth trend features into lower level features. The gradual fusion approach can better utilize the depth features of trend items, avoiding the influence of a large proportion of depth features on prediction results.

- The experimental results show that in terms of model parameter count and computational complexity. In the 32 tests of multivariate time series prediction, compared to DLiner [1], UnetTSF achieved 31 optima in both mae and mse, with an average decrease of 10.1% in mse and 9.1% in mae. Compared with PatchTST [26], UnetTSF achieved 9 optima in terms of mse and 15 optima in terms of mae.

Below, we will provide a detailed introduction to our method and demonstrate its effectiveness through extensive experiments.

II. RELATED WORK

Time series linear models: time series linear models refer to models with a complexity of $O(L)$, mainly including ARMA, NLinear, DLlinear, etc. The DLlinear [1] model and NLlinear are currently the fundamental models in the long-term time series forecasting task. Linear is the normalization of input data into a fully connected network layer. NLlinear model uses the nearest value of x_L as the basis and $x - x_L$ to construct a new historical data $x^{'}$, enter a layer of fully connected output prediction result $y^{'}$, and add the predicted result $y^{'}$ to x_L to output the predicted data y. The DLlinear [1] model divides time series data into seasonal-term x_{season} and trend-term x_{trend} through data decomposition, and predicts seasonal term y_{season} and trend term y_{trend} using fully connected layers respectively. The two are added together to obtain the predicted data y. The complexity of a linear model is $O(L)$.

Transformer models:With the breakthrough progress of Transformer in many fields such as NLP [10] and imaging, a large number of researchers have also applied Transformer models to long-term time series prediction. For example, LogTrans [9] uses convolutional self attention layers and LogSparse design to capture local information and reduce spatial complexity. Informer [4] proposed a ProbSparse self attention extraction technique that can effectively extract the most important features. Autoformer [5] draws inspiration
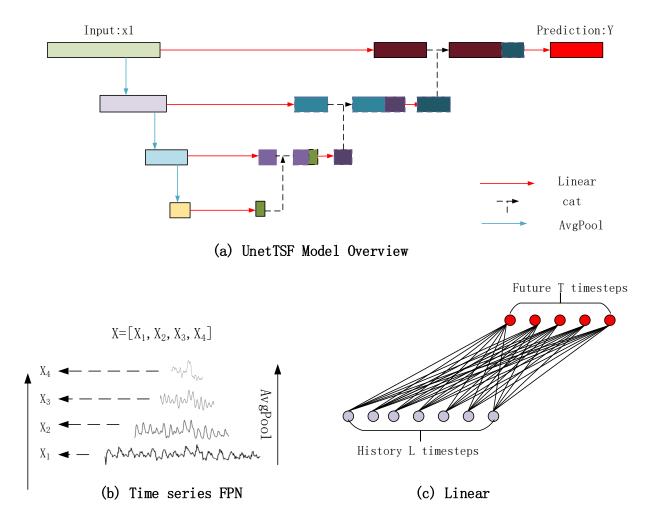
(a) UnetTSF Model Overview

Linear

cat

AvgPool

$X=[X_1, X_2, X_3, X_4]$

X_4

X_3

X_2

X_1

AvgPool

(b) Time series FPN

Future T timesteps

History L timesteps

(c) Linear

Fig. 2. UnetTSF architecture.(a)The overall inference process of UnetTSF model.The model consists of downsampling and fully connected components, with linear complexity.(b)Using avgpool to complete FPN sampling of temporal data.(c)Illustration of the basic linear model.

from traditional time series analysis methods to decompose time series data into trend and seasonal terms, and proposes the concept of autocorrelation. FEDformer [6] uses Fourier enhanced structures to achieve linear complexity. Pyraformer proposed a pyramid attention module with inter scale and intra scale connections, which also achieved linear complexity. PatchTST [26] applies patch technology to shorten sequence length, significantly reduce model complexity, and enhance local features of the representation sequence.

Time series decomposition: A time typically consists of long-term trends, seasonal fluctuations, cyclical fluctuations, and irregular fluctuations. Long term trend refers to a trend or state in which a phenomenon continues to develop and change over a long period of time. Seasonal fluctuations are regular changes in the development level of phenomena caused by seasonal changes. Cyclic fluctuations refer to periodic and continuous changes that do not follow strict rules during a certain period of time. Irregular fluctuations refer to the impact of numerous accidental factors on a time series. Autoformer [5] first applies seasonal trend decomposition,

which is time series analysis to make the raw data more predictable. Specifically, they input the trend period components extracted from the sequence. The difference between the original sequence and trend components is considered seasonal. Based on Autoformer's decomposition scheme, FEDformer [6] proposes an expert strategy for mixed extraction of trend components. By moving average kernels with various kernel sizes, PatchTST [26] and DLinear [1] both use seasonal trend binary decomposition methods to separately predict seasons and trend items, and then add them up to obtain the prediction results

U-Net:The U-Net [27] architecture, also known as ResNet in the field of image segmentation, is a well-known network architecture in the field of medical imaging. It performs well on small training datasets. The overall structure of U-Net [27] is concise, stable, and efficient, and the strong compatibility of Encoder Decoder structure allows U-Net [27] to seamlessly integrate with new generation models such as Transformer in both segmentation and generation fields. The compression characteristics of the Encoder module in U-Net [27]. As the

initial application of the Encoder module, the input image is downsampled to extract high-dimensional features that are much smaller than the original image, which is equivalent to performing compression operations. The Decoder module in U-Net utilizes multi-layer gradual fusion operations to effectively denoise and preserve effective features.

III. PROPOSED METHOD

The time series prediction problem is to predict future data with a length of T given historical data with an input length of L.Input historical data $X = \{x_1, x_2, x_3, ..., x_L\}$ with a look-back window of fixed length L , the model outputs predicted data $x = \{x_{L+1}, x_{L+2}, ..., x_{L+T}\}$, where x_i represents a vector dimension of C at time $t = i$, and C represents the number of channels in the input dataset.We designed UnetTSF using the U-Net [27] architecture and specifically designed time series FPN [12] and multi-step fusion modules suitable for temporal data, as shown in Figure 2

UnetTSF model:UnetTSF consists of fully connected layers and pooling layers. The left side of the model mainly consists of time series FPN, and the pooling function is used to form the descriptive feature $X = \{X_1, X_2, X_3, ..., X_{stage}\}$ of the input data. The stage represents the number of layers of the Unet network, and the right side of the model is the fusion module. The fully connected layer is used to fuse the upper layer features with the local layer features to output the final feature of the current layer, while the feature length remains unchanged.

Time series FPN:data decomposition is often used by time series prediction models to extract features from time series data. Generally, the data is divided into seasonal, periodic, trend, and fluctuation terms. Autoformer and DLiner both use large-scale adaptive smoothing kernels to extract trend terms from the original data. Subtracting the trend term from the original data results in seasonal terms, which can cause certain feature loss. Therefore, we adopt a multi-level extraction approach, For example, setting the data to be divided into 4 layers($stage = 4$) , using the avgpool Extract trend features with a configuration of $kernel\_szie = 3, stride = 2$ and $padding = 0$, set the original input data as x, and after passing through the FPN module, form four levels of input data:$X = [x_1, x_2, x_3, x_4]$.

$$x_1 = x$$

$$x_2 = AvgPool(x_1)$$

$$x_3 = AvgPool(x_2)$$

$$x_4 = AvgPool(x_3)$$

$$\text{len}(\text{x}_i) = \left\lfloor \frac{x_{i-1} + 2 \times padding - \ker nel\_size}{stride} + 1 \right\rfloor$$

As shown in (b) of Figure 2, the FPN structure of temporal data can effectively extract trend features. The trend information in the top layer of the pyramid is more concentrated than in the bottom layer, and the seasonal features in the bottom layer of the pyramid are more abundant.

Multi stage fusion module: Through the temporal data FPN module, a multi-scale temporal feature X is formed. To fully utilize these features, multiple fully connected predictions are used to obtain $Y = [y_1, y_2, y_3, y_4]$. The length calculation method of y_i is the same as x, and the same pooling operation is used to calculate the length at each level as X. The fusion module adopts y_i and y_{i-1} is spliced,, and then a fully connected layer is used to output $y_i^{'}$, The length of $y_i^{'}$ and y_i is the same.

$$len(y1) = len(y) = T$$

$$y_{i-1}^{'} = Linear(cat(y_i^{'}, y_{i-1}))$$

$$\text{len}(\text{y}_i) = \left\lfloor \frac{y_{i-1} + 2 \times padding - \ker nel\_size}{stride} + 1 \right\rfloor$$

IV. EXPERIMENTS

Dataset: We evaluate the performance of our proposed UnetTSF on eight widely used real-world datasets, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), transportation, electricity, weather, and ILI.These datasets have been extensively utilized for benchmarking and publicly available on [5]. We would like to emphasize several large datasets: weather, transportation, ILI, and electricity. They have more time series, and compared to other smaller datasets, the results will be more stable and less susceptible to overfitting. Univariate time series prediction testing will be conducted on the ETT dataset, while multivariate time series prediction testing will be conducted on 8 datasets. Table 1 summarizes the statistical data of these datasets.The statistics of those datasets are summarized In Table I.

Evaluation metric: Following previous works, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) [13] as the core metrics to compare performance.

Compared SOTA methods: We choose the SOTA Transformer-based models:PatchTST [26], FEDformer [6], Autoformer [5], Informer [4].At the same time, we choose the best linear model DLiner [1] as the baseline model.All of the models are following the same experimental setup with prediction length $T \in [24, 36, 48, 60]$ for ILI dataset and $T \in [96, 192, 336, 720]$ for other datasets as in the original papers. We get baseline results of models from PachTST [26] and DLinear [1] with the default look-back window $L \leq 720$.The result of patchTST [26] is taken from the optimal results of patchTST/64 and patchTST/42.

Platform:UnetTSF was trained/tested on a single Nvidia RTX A4000 16GB GPU.

V. RESULTS AND ANALYSIS

Univarite Time-series Forecasting:Table III summarize the univariate evaluation results of all the methods on ETT datasets, which is the univariate series that we are trying to forecast. We cite the baseline results from [26] and [1].Among the 16 tests on the ETT dataset, overall, UnetTSF achieved 13 optima in terms of mse and 11 optima in terms of mae. Compared with DLiner, UnetTSF achieved a maximum reduction of 54% in mse, an average decrease of 11.0%, a maximum reduction of 35.0% in mae, and an average decrease of 5.5%. Compared with

Datasets	ETTh1	ETTh2	ETTm1	ETTm2	traffic	Electricity	Weather	ILI
Variates	7	7	7	7	862	321	211	7
Timessteps	17420	17420	69680	69680	17544	26304	52696	966
Granularity	1hour	1hour	5min	5min	1hour	1hour	10min	11week

TABLE II
MULTIVARIATE LONG-TERM FORECASTING RESULTS. WE USE PREDICTION LENGTHS $T \in \{24, 36, 48, 60\}$ FOR ILI DATASET AND $T \in \{96, 192, 336, 720\}$ FOR THE OTHERS. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Models		UnetTSF		PatchTST		DLinear		FEDformer		Autoformer		Informer		Pyraformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.145**	0.210	0.147	**0.198**	0.176	0.237	0.238	0.314	0.249	0.329	0.354	0.405	0.896	0.556
	192	**0.187**	**0.234**	0.190	0.240	0.220	0.282	0.275	0.329	0.325	0.370	0.419	0.434	0.622	0.624
	336	**0.238**	**0.274**	0.242	0.282	0.265	0.319	0.339	0.377	0.351	0.391	0.583	0.543	0.739	0.753
	720	**0.304**	**0.325**	0.304	0.328	0.323	0.362	0.389	0.409	0.415	0.426	0.916	0.705	1.004	0.934
Traffic	96	**0.395**	0.263	0.360	**0.249**	0.410	0.282	0.576	0.359	0.597	0.371	0.733	0.410	2.085	0.468
	192	0.406	0.277	**0.379**	**0.256**	0.423	0.287	0.610	0.380	0.607	0.382	0.777	0.435	0.867	0.467
	336	0.422	0.285	**0.392**	**0.264**	0.436	0.296	0.608	0.375	0.623	0.387	0.776	0.434	0.869	0.469
	720	0.443	0.299	**0.432**	**0.286**	0.466	0.315	0.621	0.375	0.639	0.395	0.827	0.466	0.881	0.473
Electricity	96	0.132	0.229	**0.129**	**0.222**	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449
	192	0.146	0.244	**0.141**	**0.241**	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417	0.386	0.443
	336	**0.162**	0.262	0.163	**0.259**	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422	0.378	0.443
	720	0.200	0.297	**0.197**	**0.290**	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427	0.376	0.445
ILI	24	1.696	0.789	**1.281**	**0.704**	2.215	1.081	2.624	1.095	2.906	1.182	4.657	1.449	1.420	2.012
	36	1.693	0.811	**1.251**	**0.752**	1.963	0.963	2.516	1.021	2.585	1.038	4.650	1.463	7.394	2.031
	48	1.867	0.881	**1.673**	**0.854**	2.130	1.024	2.505	1.041	3.024	1.145	5.004	1.542	7.551	2.057
	60	**1.421**	**0.747**	1.526	0.795	2.368	1.096	2.742	1.122	2.761	1.114	5.071	1.543	7.662	2.100
ETTh1	96	**0.368**	**0.394**	0.370	0.400	0.375	0.399	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612
	192	0.406	0.417	0.413	0.431	**0.405**	**0.416**	0.423	0.446	0.456	0.457	1.007	0.786	0.790	0.681
	336	**0.408**	**0.425**	0.422	0.440	0.439	0.443	0.444	0.462	0.486	0.487	1.038	0.784	0.891	0.738
	720	0.458	0.462	**0.447**	**0.468**	0.472	0.490	0.469	0.492	0.515	0.517	1.144	0.857	0.963	0.782
ETTh2	96	0.279	0.333	**0.274**	0.336	0.289	0.353	0.332	0.374	0.332	0.368	1.549	0.952	0.645	0.597
	192	0.343	0.395	**0.339**	**0.379**	0.383	0.418	0.407	0.446	0.426	0.434	3.792	1.542	0.788	0.683
	336	0.379	0.423	**0.331**	**0.380**	0.448	0.465	0.400	0.447	0.477	0.479	4.215	1.642	0.907	0.747
	720	0.446	0.464	**0.379**	**0.422**	0.605	0.551	0.412	0.469	0.453	0.490	3.656	1.619	0.963	0.783
ETTm1	96	**0.287**	**0.336**	0.290	0.342	0.299	0.343	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510
	192	0.330	**0.359**	**0.328**	0.365	0.335	0.365	0.365	0.415	0.514	0.495	0.725	0.619	0.557	0.537
	336	0.368	**0.380**	**0.361**	0.393	0.369	0.386	0.392	0.425	0.510	0.492	1.005	0.741	0.754	0.655
	720	0.425	**0.413**	**0.416**	0.419	0.425	0.421	0.446	0.458	0.527	0.493	1.133	0.845	0.908	0.724
ETTm2	96	0.163	0.250	**0.162**	0.254	0.167	0.260	0.180	0.271	0.205	0.293	0.355	0.462	0.435	0.507
	192	**0.216**	**0.287**	0.216	0.293	0.224	0.303	0.252	0.318	0.278	0.336	0.595	0.586	0.730	0.673
	336	0.271	0.324	**0.269**	0.329	0.281	0.342	0.324	0.364	0.343	0.379	1.270	0.871	1.201	0.845
	720	0.360	0.389	**0.350**	**0.380**	0.397	0.421	0.410	0.420	0.414	0.419	3.001	1.267	3.625	1.451

PatchTST, UnetTSF has 7 leading items and 6 identical items in the mse indicator, with only 3 items worse than PatchTST. In the mae indicator, UnetTSF has 10 leading items, 1 identical item, and 4 items worse than PatchTST

Multivariate Time-series Forecasting:Table II summarize the multivarite evaluation results of all the methods on eight datasets. We cite the baseline results from ($Zengetal.$, 2022).Overall, the UnetTSF model performed very well in testing. Among the 32 tests on 8 datasets, 10 achieved the best results in terms of mse and 15 achieved the best results in terms of mae.Compared to DLinear, the average decrease in mse was 10.1% and the decrease in mae was 9.1%. On the most periodic dataset weather, we achieved all optimal results, with UnetTSF and patchTST showing a decrease of 1.0% in

mse and 0.4% in mae, and a decrease of 12.6% and 15.1% in mae compared to DLinear. On the traffic and electricity datasets, UnetTSF showed a decrease of about 4% in mse compared to DLiner. On the ILI dataset, UnetTSF achieved the best results in predicting a length of 60. Compared with PatchTST, mse decreased by 7.4% and mae decreased by 6.4%. Compared with DLinear, mse decreased by 66.6% and mae decreased by 46.7%. Among the 16 tests on the ETT dataset, UnetTSF achieved 11 optima in mae and mse achieved 4 optima,Especially on the test item with a predicted length of 192 on ETTh1, UnetTSF showed a 3.4% decrease in mse and a 3.5% decrease in mae compared to PatchTST.

model efficiency:Table IV The parameter quantity and computational complexity are one of the important evaluation

TABLE III

Models		UnetTSF		PatchTST		DLinear		FEDformer		Autoformer		Informer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.055**	**0.178**	**0.055**	0.179	0.056	0.180	0.079	0.215	0.071	0.206	0.193	0.377	0.283	0.468
	192	**0.071**	0.205	**0.071**	0.205	**0.071**	**0.204**	0.104	0.245	0.114	0.262	0.217	0.395	0.234	0.409
	336	0.078	0.221	**0.076**	**0.220**	0.098	0.244	0.119	0.270	0.107	0.258	0.202	0.381	0.386	0.546
	720	**0.087**	0.233	**0.087**	**0.232**	0.189	0.359	0.142	0.299	0.126	0.283	0.183	0.355	0.475	0.629
ETTh2	96	**0.124**	**0.271**	0.129	0.282	0.131	0.279	0.128	0.271	0.153	0.306	0.213	0.373	0.217	0.379
	192	**0.165**	**0.320**	0.168	0.328	0.176	0.329	0.185	0.330	0.204	0.351	0.227	0.387	0.281	0.429
	336	0.184	0.348	**0.171**	**0.336**	0.209	0.367	0.231	0.378	0.246	0.389	0.242	0.401	0.293	0.437
	720	0.227	0.382	**0.223**	**0.380**	0.276	0.426	0.278	0.420	0.268	0.409	0.291	0.439	0.218	0.387
ETTm1	96	**0.026**	**0.121**	**0.026**	**0.121**	0.028	0.123	0.033	0.140	0.056	0.183	0.109	0.277	0.049	0.171
	192	**0.039**	**0.149**	**0.039**	0.150	0.045	0.156	0.058	0.186	0.081	0.216	0.151	0.310	0.157	0.317
	336	**0.051**	**0.171**	0.053	0.173	0.061	0.182	0.084	0.231	0.076	0.218	0.427	0.591	0.289	0.459
	720	**0.071**	**0.203**	0.073	0.206	0.080	0.210	0.102	0.250	0.110	0.267	0.438	0.586	0.430	0.579
ETTm2	96	0.063	**0.181**	0.065	0.186	**0.063**	0.183	0.067	0.198	0.065	0.189	0.088	0.225	0.075	0.208
	192	**0.090**	**0.224**	0.094	0.231	0.092	0.227	0.102	0.245	0.118	0.256	0.132	0.283	0.129	0.275
	336	**0.116**	**0.256**	0.120	0.265	0.119	0.261	0.130	0.279	0.154	0.305	0.180	0.336	0.154	0.302
	720	**0.170**	**0.318**	0.171	0.322	0.175	0.320	0.178	0.325	0.182	0.335	0.300	0.435	0.160	0.321
Count		13	11	9	4	2	1	0	0	0	0	0	0	0	0

TABLE IV

Method	MACs	Parameter	Memory
UnetTSF	13.56M	0.42M	20.39M
DLinear	14.52M	0.45M	20.55M
PatchTST	164.97M	0.46M	20.72M
Autoformer	90517.61M	10.54M	119.39M
Informer	79438.08M	11.33M	126.22M

indicators of the model.We used a fixed length $L = 336$ review window, a predicted future length $T = 96$, and a $batch = 32$ configuration on the ETTh2 dataset to evaluate the parameter count, computational complexity, and GPU memory usage of all model inference stages.In Tab IV we compare the average practical efficiencies with 5 runs.

- In terms of computational complexity, DLiner and UnetTSF have significant advantages. PatchTST uses patch technology to significantly reduce computational complexity, but it is still 12.1 times that of UnetTSF. UnetTSF and DLiner both use fully connected and pooling layers, while UnetTSF uses a small pooling kernel to reduce computational complexity by 6.6%.
- In terms of model parameter quantity and inference memory usage, UnetTSF has the same magnitude as DLline and PatchTST. Combining the results of univariate and multivariate time series prediction, UnetTSF is more suitable for use in limited resource scenarios than DLline and PatchTST.

VI. CONCLUSION

This paper proposes a long time series prediction model UnetTSF with linear complexity. UnetTSF has two innovative points: the first proposes an FPN description structure for time series data, which becomes the third choice for raw data description and binary decomposition methods for time series data. The second is to introduce the Unet network structure into the field of time series prediction, and design and implement an Unet network suitable for time series prediction. The experimental results show that UnetTSF has superior performance compared to DLiner and PatchTST, and UnetTSF with linear complexity is more suitable for practical production and life.

REFERENCES

[1] Ailing Zeng, Muxi Chen, Lei Zhang and Qiang Xu, "Are Transformers Effective for Time Series Forecasting?", arXiv preprint arXiv:2205.13504, 2022.

[2] Jake Grigsby, Zhe Wang and Yanjun Qi, "Long-Range Transformers for Dynamic Spatiotemporal Forecasting",arXiv preprint arXiv:2109.12218,2021.

[3] Vaswani Ashish , Shazeer Noam , Parmar, Niki Uszkoreit, Jakob Jones, Llion Gomez, Aidan N Kaiser, Lukasz Polosukhin and Illia, "Attention is All you Need", NeurIPS,2017.

[4] Haoyi Zhou, Shanghang Zhang , Jieqi Peng , Shuai Zhang , Jianxin Li , Hui Xiong and Wancai Zhang, Informer: Beyond Efficient Transformer for Long Sequence Time-Series Fo recasting, The Thirty-Fifth AAAI Conference on Artificial Intelligence,2021.

[5] 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,2021.

[6] ZhouTian Ma, Ziqing Wen, Qingsong Wang, Xue Sun, Liang Jin and Rong, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting,Proc. 39th International Conference on Machine Learning,2022.

[7] Liu, Shizhan Yu, Hang Liao, Cong Li, Jianguo Lin, Weiyao Liu, Alex X Dustdar and Schahram, Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting,International Conference on Learning Representations,2022.

[8] Cirstea Razvan-Gabriel , Guo Chenjuan , Yang Bin , Kieu Tung , Dong Xuanyi and Pan Shirui, Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting,Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22,2022.

[9] Li Shiyang , Jin Xiaoyong , Xuan Yao , Zhou Xiyou , Chen Wenhu , Wang Yu-Xiang and Yan Xifeng,Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting, Advances in Neural Information Processing Systems,https://proceedings.neurips.cc/paper/2019/file/6775a0635c302542da2c32aa19d86be0-Paper.pdf,2019.

[10] Jacob Devlin , Ming-Wei Chang , Kenton Lee and Kristina Toutanova,"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding",2018,http://arxiv.org/abs/1810.04805.

[11] JTianping Zhang , Yizhuo Zhang , Wei Cao , Jiang Bian , Xiaohan Yi , Shun Zheng and Jian Li,"Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures",2022,https://arxiv.org/abs/2207.01186.

[12] TY Lin , P Dollár , R Girshick , K He , B Hariharan and S Belongie,"Feature Pyramid Networks for Object Detection",2016,https://arxiv.org/abs/1612.03144.

[13] Kaiming He , Xinlei Chen , Saining Xie , Yanghao Li , Piotr Dollár , Ross B. Girshick,"Masked Autoencoders Are Scalable Vision Learners",CoRR,2021,https://arxiv.org/abs/2111.06377,

[14] Hangbo Bao , Li Dong , Songhao Piao and Furu Wei, BERT: Pre-Training of Image Transformers,International Conference on Learning Representations,2022,https://openreview.net/forum?id=p-BhZSz59o4.

[15] Alexey Dosovitskiy, Lucas Beyer , Alexander Kolesnikov , Dirk Weissenborn , Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, International Conference on Learning Representations,2021,https://openreview.net/forum?id=YicbFdNTTy.

[16] Cao Defu , Wang Yujing , Duan Juanyong , Zhang Ce , Zhu Xia , Huang Congrui, Tong Yunhai , Xu Bixiong, Bai Jing and Tong Jie, "Spectral temporal graph neural network for multivariate time-series forecasting",Advances in neural information processing systems,2020.

[17] Chen Yuzhou , Segovia Ignacio and Gel Yulia R,"Z-GCNETs: time zigzags at graph convolutional networks for time series forecasting",International Conference on Machine Learning,2021.

[18] Zerveas George , Jayaraman Srideepika , Patel Dhaval, Bhamidipaty Anuradha and Eickhoff Carsten,"A transformer-based framework for multivariate time series representation learning",Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining,2021.

[19] Yue Zhihan, Wang Yujing, Duan Juanyong, Yang Tianmeng, Huang Congrui, Tong Yunhai and Xu Bixiong,"Ts2vec: Towards universal representation of time series", Proceedings of the AAAI Conference on Artificial Intelligence,2022.

[20] Zheng Yi, Liu Qi, Chen Enhong, Ge Yong and Zhao J Leon,"Time series classification using multi-channels deep convolutional neural networks",International conference on web-age information management,2014.

[21] Ulyanov Dmitry , Vedaldi Andrea and Lempitsky Victor , "Instance normalization: The missing ingredient for fast stylization",arXiv preprint arXiv:1607.08022,2016.

[22] 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",International Conference on Learning Representations,2022,https://openreview.net/forum?id=cGDAkQo1C0p.

[23] Bryan Lim and Stefan Zohren, "Time-series forecasting with deep learning: a survey","Phil. Trans. R. Soc. A",2021.

[24] Torres, José F , Hadjout Dalil, Sebaa Abderrazak, Martínez-Álvarez Francisco and Troncoso Alicia,"Deep learning for time series forecasting: a survey", "Big Data" 2021.

[25] Wu Haixu, Hu Tengge , Liu Yong, Zhou Hang, Wang Jianmin and Long Mingsheng,"TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis", "ICLR",2023.

[26] Nie Yuqi, Nguyen Nam H, Sinthong Phanwadee and Kalagnanam Jayant,"A Time Series is Worth 64 Words: Long-term Forecasting with Transformers",ICLR,2023.

[27] Ronneberger Olaf , Fischer Philipp and Brox Thomas,"U-net: Convolutional networks for biomedical image segmentation",Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference,2015.

[28] Lai Guokun, Chang Wei-Cheng, Yang Yiming and Liu Hanxiao,"Modeling long-and short-term temporal patterns with deep neural networks",SIGIR,2018.

[29] Zhao Zheng, Chen Weihai, Wu Xingming, Chen Peter CY and Liu Jingmeng,"LSTM network: a deep learning approach for short-term traffic forecast",IET Intelligent Transport Systems",2017".

[30] Yong Liu, Haixu Wu, Jianmin Wang and Mingsheng Long,"Non-stationary Transformers: Rethinking the Stationarity in Time Series Forecasting",NeurIPS,2022.

[31] Bommasani Rishi, Hudson Drew A, Adeli Ehsan, Altman Russ, Arora Simran, von Arx Sydney, Bernstein Michael S, Bohg Jeannette, Bosselut Antoine, Brunskill Emma and others,"On the opportunities and risks of foundation models",arXiv preprint arXiv:2108.07258",2021.

[32] G Woo, C Liu, D Sahoo, A Kumar and S Hoi, "xponential Smoothing Transformers for Time-series Forecasting","http://aixpaper.com/view/etsformer_exponential_smoothing_transformers_for_times

[33] G. E. P. Box and Gwilym M. Jenkins, "Time series analysis, forecasting and control",1970.

[34] Salinas David, Flunkert Valentin, Gasthaus Jan and Januschowski Tim, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks",International Journal of Forecasting,2020.