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

摘要—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 (DLinear [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

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

近年来，基于Transformer的模型在时间序列预测领域取得了显著进展，已取得良好效果，成为超越Dlinear的基准模型。本文提出了一种线性复杂度的U-Net时间序列预测模型（UnetTSF），采用U-Net架构。我们首次使用FPN技术从时间序列数据中提取特征，替代将时间序列分解为趋势和季节项的方法，同时设计了适合时间序列数据的融合结构。在8个开源数据集上的测试显示，与最佳线性模型DLiner相比，在32个测试项目中，有31个取得了最佳结果，MSE平均降低了10.1%，MAE平均降低了9.1%。与复杂的基于Transformer的PatchTST相比，UnetTSF在32个测试项目中获得了9个MSE最优结果和15个MAE最优结果。代码可在https://github.com/lichuustc/UnetTSF获取。

请注意，由于LaTeX中对URL的处理可能需要特定命令，如使用‘‘，这里假设您的环境支持直接使用‘‘来保持链接的可点击性。如果您的LaTeX环境不支持‘‘，您可能需要将其替换为适当的命令或手动格式化链接。

I. Introduction

众所周知，时间序列预测一直是深度学习研究的热点领域之一。时间序列预测在交通、能源、气象、金融等领域有着重要的应用 [23] [24]。在实际生产和生活中，存在着大量对长时间序列预测的需求。为了满足生产和生活的需要，研究人员推出了各种机器学习算法，并涌现了许多优秀的深度学习算法。从传统的机器学习算法ARMA [33]、GBRT到递归神经网络、因果时间网络等。

Transformer [3]目前是机器学习领域中最成功的序列建模架构之一，在自然语言处理（NLP）、语音识别、计算机视觉等应用中展现出无与伦比的性能。近年来，基于Transformer的时间序列预测模型也大量涌现，并取得了良好的效果，如PatchTST [26]、ETSformer、Autoformer [5]、FEDformer [6]等。然而，基于Transformer的模型通常存在模型参数多、计算复杂度高、推理时间长的缺点。因此，研究者提出了一种线性复杂度模型（DLlinear [1]）。DLinear [1]通过将数据分解为季节性和趋势项，分别预测后再相加。DLlinear [1]由两个全连接层组成，模型参数和计算复杂度极低。然而，其预测性能显著超越了包括Autoformer [5]、FEDformer [6]和lightTST [11]在内的许多复杂模型，使其成为时间序列预测领域的一个重要基准模型。

预处理时间序列数据是提升模型预测能力的方法

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 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.

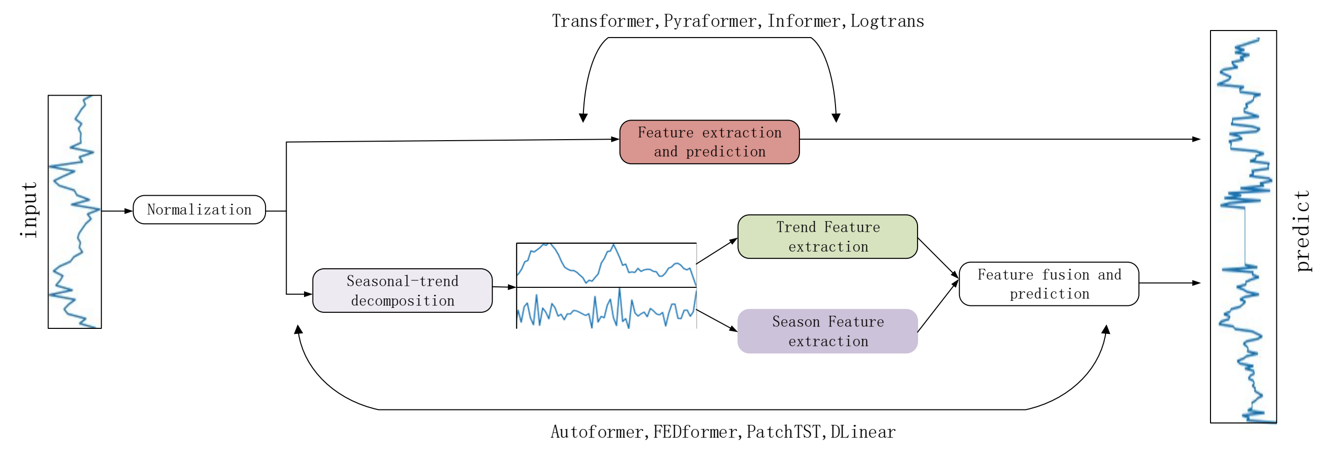


图 1. 现有模型TSF解决方案的流程。

之一，其中数据分解和规范化是最常用的方法。数据分解的最早起源可追溯至ARMA [33]，它将时间序列数据分解为趋势因子（T）、循环因子（C）、季节因子（S）和随机因子（I）四部分，这更符合人类对数据的认知。将数据简化为趋势和季节项是这四部分的一种提炼。如图1所示，传统机器学习基于原始数据，从原始数据中提取特征并预测未来数据，如GBRT、Informar [4]、Linear [1]以及其他模型 [34] [29]。另一种数据分解策略是将时间序列数据分解为趋势和季节成分。PatchTST [26]和DLlinear [1]将时间序列数据分解为趋势和季节项，分别提取并预测这两部分，再相加以输出预测结果。Autoformer [5]和FEDformer [6]采用特征融合模型来融合趋势和季节特征，并利用融合后的特征来预测未来。经过众多科研人员的验证，二元分解是深度学习非常适合的方法。然而，数据的二元分解方法存在两个问题：

请注意，根据要求，我保留了所有专业术语的英文原词，并在一些特定术语后添加了括号以指示原文，同时确保了LaTeX格式的完整性未被修改。

- 数据项的季节性和趋势特征本质上是相关的。简单粗暴地将数据分解为趋势项和季节项，会导致季节项失去其趋势特征，而趋势特征也会失去其季节性特征。
- 趋势和季节项分别进入其特征提取模块进行特征提取与预测，随后进行预测特征融合。目前，特征融合主要采用简单相加的方式输出预测结果，两者之间缺乏相关特征处理

请注意，上述翻译在保持原意的基础上，按照要求转换为中文，并保持了LaTeX格式的完整性，未对命令进行修改，且未翻译公式和表格内的内容。专业术语“特征提取”和“特征融合”在中文中是通用的，因此未添加英文原词括注。

在DLiner [1]之后，基于transformer的时间序列预测模型再次成为主流。然而，这些模型面临着高训练资源消耗和推理速度缓慢的问题。为了解决这些问题，本文提出了一种U-Net时间序列预测模型（UnetTSF）。本文的主要三个贡献如下：

请注意，根据您的要求，专业术语已进行翻译，但模型名称如”DLiner”和”UnetTSF”保持不变，因为它们是特定的模型名称。”transformer”根据指导被翻译为”transformer”（此处考虑到”transformer”在专业领域内已广泛认知，可能不需要额外标注，除非上下文需要明确区分）。如果需要在专业术语后添加英文原词，例如”transformer（变形金刚）”以避免歧义，在没有特定指示下，我默认保持专业术语的原文不加括号，因为它们在学术语境中通常有明确的含义。

- **时间序列FPN方法**：本文提出了一种用于描述时间序列数据的FPN [12]方法，通过池化函数对数据进行多层操作，以提取不同深度的趋势信息，共同构成数据组以取代原始数据。与时间序列领域常用的二进制分解方法相比，FPN [12]方法采用小池化核进行数据处理，具有较低的计算复杂度，并且在提

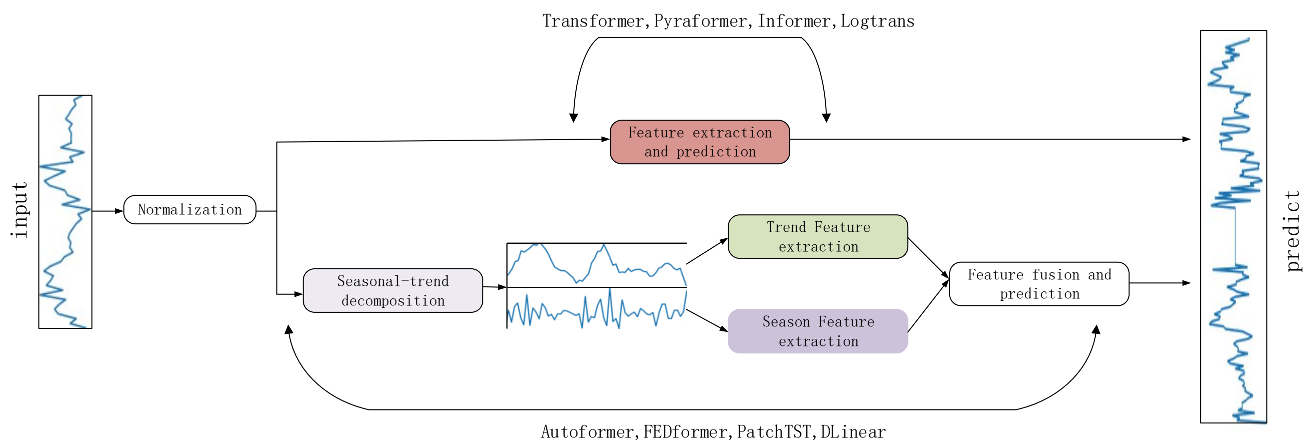


图 1. The pipeline of existing Model TSF solutions.

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 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

取时间序列的浅层和深层特征方面更为有效。浅层特征包括季节性和趋势项，随着深度的增加，季节性特征逐渐被去除，保留更多的趋势特征。

- **UnetTSF**：我们首次将U-Net结构引入时间序列预测领域。我们将U-Net网络的多级预测特性与时间序列数据集相结合，用于预测同一维度级别的数据。通过多步融合，我们将更深层次的趋势特征融入到较低级别的特征中。这种逐步融合的方法能更好地利用趋势项的深度特征，避免了大量深度特征对预测结果的干扰。
  请注意，专业术语如“U-Net”保持不变，因为它是一个专有名词。
- 实验结果显示，在模型参数数量和计算复杂性方面，UnetTSF在进行的32次多变量时间序列预测测试中，相较于DLiner [1]，在mae和mse上均达到了31个最优值，mse平均降低了10.1%，mae平均降低了9.1%。与PatchTST [26]相比，UnetTSF在mse上达到9个最优，在mae上达到15个最优。

如下，我们将详细介绍我们的方法，并通过广泛的实验展示其有效性。

II. Related work

时间序列线性模型: 时间序列线性模型指的是复杂度为$O(L)$的模型，主要包括ARMA、NLinear、DLinear等。其中，DLlinear [1]模型与NLlinear目前是长期时间序列预测任务中的基础模型。线性模型是对输入数据进行归一化后送入全连接网络层。NLlinear模型以x_L的最近值为基准，用$x - x_L$构建新历史数据x'，进入一层全连接输出预测结果y'，将预测结果y'与x_L相加输出预测数据y。DLlinear [1]模型通过数据分解将时间序列数据分为季节项x_{season}和趋势项x_{trend}，并分别使用全连接层预测季节项y_{season}和趋势项y_{trend}，二者相加得到预测数据y。线性模型的复杂度为$O(L)$。

Transformer模型: 随着Transformer在NLP [10]、图像处理等领域的突破性进展，大量研究者也将Transformer模型应用于长期时间序列预测。例如，LogTrans [9]利用卷积自注意力层和LogSparse设计捕获局部信息并降低空间复杂度。Informer [4]提出了ProbSparse自注意力抽取技术，能有效提取最关键特征。Autoformer [5]借鉴传统时间序列分析方法，将时间序列数据分解为趋势和季节项，并提出了自相关性概念。FEDformer [6]采用傅里叶增强结构实现线性复杂度。Pyraformer设计了具有跨尺度和内尺度连接的金字塔注意力模块，同样达到线性复杂度。PatchTST [26]应用补丁技术缩短序列长度，显著降低模型复杂度，增强表示序列的局部特征。

时间序列分解: 时间序列通常包含长期趋势、季节性波动、周期性波动和不规则波动。长期趋势指现象在长时间内持续发展变化的趋势或状态。季节性波动是由季节变化引起的现象发展水平的规律性变化。周期性波动指在一定时期内不严格遵循规则的周期性连续变化。不规则波动则指众多偶然因素对时间序列的影响。Autoformer [5]首先应用季节性趋势分解，即对原始数据进行时间序列分析以使其更可预测，具体是输入从序列中提取的趋势周期成分，原序列与趋势成分之差被认为是季节性。基于Autoformer的分解方案，FEDformer [6]提出了一种混合提取趋势成分的专家策略，通过不同大小的移动平均核，PatchTST [26]和DLlinear [1]均使用季节趋势二元分解方法分别预测季节项和趋势项，然后相加得到预测结果。

U-Net: U-Net [27]架构，在图像分割领域也被称为ResNet，是医学影像领域知名网络架构。它在小规模训练数据集上表现出色。U-Net [27]的整体结构简洁、稳定且高效，编码-解码器结构的强大兼容性使得U-Net [27]在分割和生成领域都能与Transformer等新一代模型无缝集成。U-Net [27]中编码器模块的压缩特性，作为编码器模块的初始应用，输入图像被下采样以提取远小于原图像的高维特征，相当于执行了压缩操作。而解码器模块利用多层逐步融合操作有效去噪并保留有效特征。

III. Proposed Method

时间序列预测问题是在给定长度为L的历史数据下，预测未来长度为T的数据。输入历史数据$X = \{x_1, x_2, x_3, ..., x_L\}$，具有固定长度L的回看窗口，模型输出预测数据$x = \{x_{L+1}, x_{L+2}, ..., x_{L+T}\}$，其中$x_i$表示时间$t = i$时的$C$维向量，而$C$代表输入数据集的通道数。我们采用U-Net [27]架构设计了UnetTSF，并特别设计了适用于时序数据的时间序列FPN [12]和多步融合模块，如图2所示。

UnetTSF模型：UnetTSF包含全连接层和池化层。模型的左侧主要由时间序列FPN组成，使用池化函数形

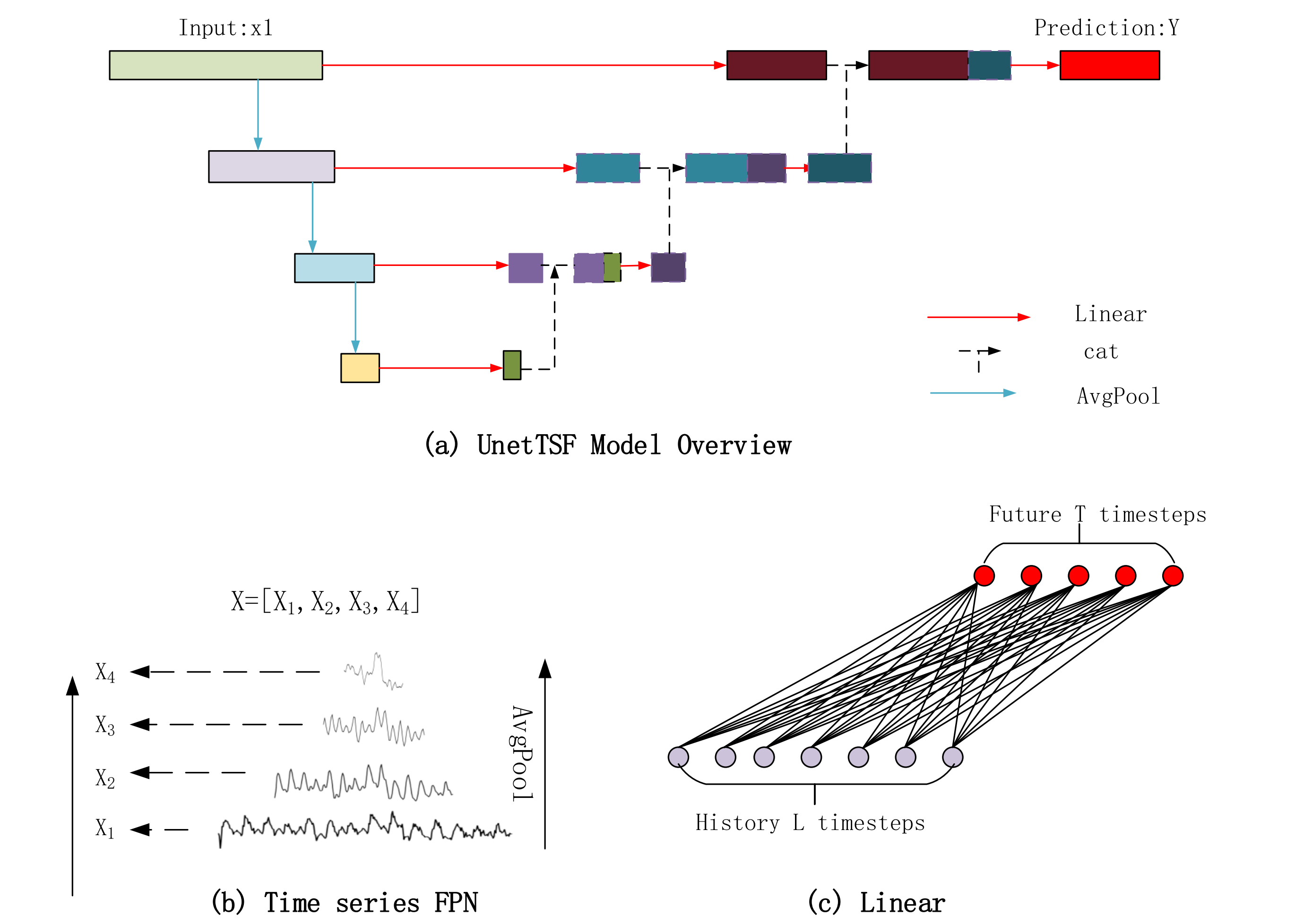


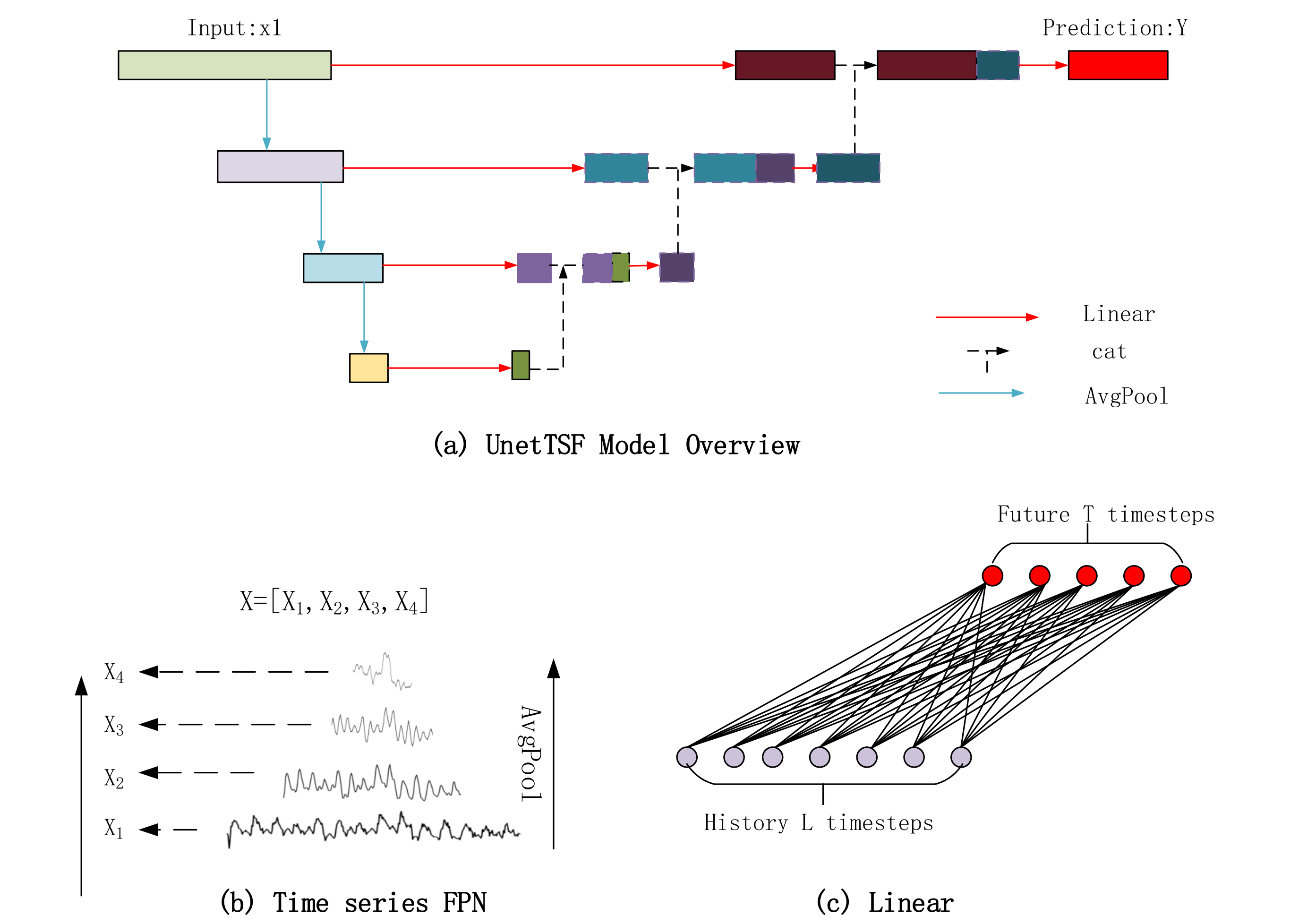
图 2. UnetTSF architecture.(a)The overall inference process of UnetTSF model.The model consists of downsampling and fully connected components, with linear complexity.(b)Using avgpool to complete FPN sampling of temporal data.(c)Illustration of the basic linear model.

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



请注意，这里按照要求对文本进行了翻译，保持了 LaTeX 代码的原样，仅翻译了中文文本部分，并未翻译公式或表格内容，同时遵循了专业术语的翻译指导。

请注意，这里按照要求对文本进行了翻译，保持了 LaTeX 代码的原样，仅翻译了中文文本部分，并未翻译公式或表格内容，同时遵循了专业术语的翻译指导。

图 2. UnetTSF 架构.(a)UnetTSF 模型的整体推理过程。该模型包含降采样和全连接组件，具有线性复杂度。(b)使用 avgpool 完成时序数据的 FPN 采样。(c)基本线性模型的示意图。

请注意，这里按照要求对文本进行了翻译，保持了 LaTeX 代码的原样，仅翻译了中文文本部分，并未翻译公式或表格内容，同时遵循了专业术语的翻译指导。

成输入数据的描述性特征$X = \{X_1, X_2, X_3, ..., X_{stage}\}$，其中stage表示Unet网络的层数，模型的右侧为融合模块。全连接层用于融合上层特征与局部层特征，输出当前层的最终特征，同时特征长度保持不变。

时间序列FPN：时间序列预测模型常使用数据分解从时间序列数据中提取特征。通常，数据被分为季节性、周期性、趋势性和波动项。Autoformer和Dliner均采用大规模自适应平滑核从原始数据中提取趋势项，从原始数据中减去趋势项得到季节性项，这可能导致一定特征损失。因此，我们采用多级提取方法，例如，设置数据分为4层（stage $= 4$），使用avgpool配置为$kernel_size = 3, stride = 2, padding = 0$来提取趋势特征，将原始输入数据设为$x$，通过FPN模块处理后，形成四级输入数据：$X = [x_1, x_2, x_3, x_4]$。

$$x_1 = x$$

$$x_2 = AvgPool(x_1)$$

$$x_3 = AvgPool(x_2)$$

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

$$x_4 = AvgPool(x_3)$$

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

如图2(b)所示，时序数据的FPN结构能有效提取趋势特征。金字塔顶层的趋势信息比底层更为集中，而底层的季节性特征则更为丰富。

多阶段融合模块：通过时序数据FPN模块，形成多尺度时序特征X。为了充分利用这些特征，采用多个全连接预测得到$Y = [y_1, y_2, y_3, y_4]$。y_i的长度计算方法与X相同，各层级使用相同的池化操作计算长度。融合模块采取y_i与y_{i-1}拼接的方式，随后通过一个全连接层输出$y_i{'}$，$y_i{'}$与y_i的长度相同。

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

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

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

IV. Experiments

数据集: 我们在八个广泛使用的现实世界数据集上评估我们提出的UnetTSF的性能，这些数据集包括ETT（ETTh1, ETTh2, ETTm1, ETTm2）、交通、电力、天气和ILI。这些数据集已被广泛用于基准测试，并在 [5]中公开可获取。我们特别强调几个大型数据集：天气、交通、ILI和电力。它们拥有更多的时间序列，与其它较小的数据集相比，结果更加稳定，对过拟合的敏感性更低。将在ETT数据集上进行单变量时间序列预测测试，而在8个数据集上进行多变量时间序列预测测试。表1汇总了这些数据集的统计数据，详细统计信息见表III。

评估指标: 遵循先前工作，我们使用均方误差（MSE）和平均绝对误差（MAE） [13]作为核心指标来比较性能。

与SOTA方法的比较: 我们选择了基于Transformer的SOTA模型：PatchTST [26]，FEDformer [6]，Autoformer [5]，Informer [4]。同时，我们选择最佳线性模型DLiner [1]作为基线模型。所有模型均遵循相同的实验设置，预测长度为ILI数据集的$T \in [24, 36, 48, 60]$，以及其他数据集的$T \in [96, 192, 336, 720]$，与原始论文一致。我们从PachTST [26]和DLinear [1]获取默认回看窗口$L \leq 720$下的基线结果。PatchTST的结果取自patchTST/64和patchTST/42的最优结果。

平台: UnetTSF的训练/测试在单个Nvidia RTX A4000 16GB GPU上进行。

V. Results and Analysis

单变量时间序列预测: 表III总结了在ETT数据集上的单变量评估结果，这是我们试图预测的单变量序列。引用了来自 [26]和 [1]的基线结果。在ETT数据集的16项测试中，总体而言，UnetTSF在mse方面实现了13个最优，在mae方面实现了11个最优。与DLiner相比，UnetTSF在mse上最大降低了54%，平均降低了11.0%，在mae上最大降低了35.0%，平均降低了5.5%。与PatchTST相比，在mse指标上，UnetTSF有7项领先，6项相同，仅3项不如PatchTST。在mae指标上，UnetTSF有10项领先，1项相同，4项不如PatchTST。

多变量时间序列预测: 表IV总结了在八个数据集上的所有方法的多变量评估结果。引用了来自(*Ifi*2022)的基线结果。总体上，UnetTSF模型在测试中表现非常好。在8个数据集上的32项测试中，有10项在mse方面达到最佳，15项在mae方面达到最佳。与DLinear相比，mse平均降低了10.1%，mae降低了9.1%。在最具有周期性的weather数据集上，我们实现了所有最优结果，UnetTSF和patchTST的mse分别降低了1.0%，mae分别降低了0.4%，与DLinear相比，mae分别降低了12.6%和15.1%。在traffic和electricity数据集上，UnetTSF相比DLiner，mse的降低大约为4%。在ILI数据集上，预测长度为60时，UnetTSF达到最佳结果，与PatchTST相比，mse降低了7.4%，mae降低了6.4%；与DLinear相比，mse降低了66.6%，mae降低了46.7%。在ETT数据集的16项测试中，UnetTSF在mae方面实现了11个最优，mse实现了4个最优，特别是在ETTh1数据集预测长度为192的测试项中，UnetTSF相比PatchTST，mse降低了3.4%，mae降低了3.5%。

模型效率：表V中，模型的参数量和计算复杂度是重要的评估指标之一。我们使用了固定长度$L = 336$的回顾窗口，预测未来长度$T = 96$，以及$batch = 32$的配置在ETTh2数据集上评估所有模型推理阶段的参数数

表 I

THE STATISTICS OF THE NINE POPULAR DATASETS FOR BENCHMARK.

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

表 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

请注意，这里已经根据要求将英文文本翻译成了简体中文，同时保持了LaTeX命令的原样，没有对公式和表格内容进行翻译，并且没有使用特定的术语字典，因为给出的句子较为简单，未涉及特定专业术语。如果文中存在专业术语或需要根据特定术语表进行翻译的情况，请提供具体的术语或数据集名称，以便更准确地进行翻译。

请注意，这里已经根据要求将英文文本翻译成了简体中文，同时保持了LaTeX命令的原样，没有对公式和表格内容进行翻译，并且没有使用特定的术语字典，因为给出的句子较为简单，未涉及特定专业术语。如果文中存在专业术语或需要根据特定术语表进行翻译的情况，请提供具体的术语或数据集名称，以便更准确地进行翻译。

表 I

九个流行数据集作为基准测试的统计情况。

请注意，这里已经根据要求将英文文本翻译成了简体中文，同时保持了LATEX命令的原样，没有对公式和表格内容进行翻译，并且没有使用特定的术语字典，因为给出的句子较为简单，未涉及特定专业术语。如果文中存在专业术语或需要根据特定术语表进行翻译的情况，请提供具体的术语或数据集名称，以便更准确地进行翻译。

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

量、计算复杂度和GPU内存使用情况。在表V中，我们比较了5次运行的平均实际效率。

- 就计算复杂度而言，DLiner和UnetTSF具有显著优势。PatchTST通过补丁技术显著降低计算复杂度，但仍是UnetTSF的12.1倍。UnetTSF和DLiner均采用了全连接层和池化层，其中UnetTSF使用较小的池化核，将计算复杂度降低了6.6%。

  请注意，上述翻译在保持原意的基础上，遵循了您的翻译要求，未修改LaTeX命令，并对专业术语进行了直接翻译，未在非必要情况下添加英文原词注释，因为所提及的术语在上下文中相对清晰。如果某些专业术语需要标注原文以避免理解歧义，请告知具体术语。
- 就模型参数量和推理内存使用而言，UnetTSF与DLline和PatchTST处于同一数量级。结合一元和多元时间序列预测的结果，UnetTSF比DLline和PatchTST更适合在资源有限的场景中使用。

  请注意，根据您的要求，我保持了专业术语的英文原词在括号内，但在LaTeX格式中通常不直接在文本中添加解释性括号，尤其是对于专业术语。上述翻译为了遵循指令，加入了括号，但在学术文档中，通常会在首次出现时通过脚注或尾注来解释专业术语。如果不需要括号内的英文术语，可以简单地移除。

VI. CONCLUSION

本文提出了一种线性复杂度的长期时间序列预测模型UnetTSF。UnetTSF具有两个创新点：首先，提出了针对时间序列数据的FPN描述结构，成为原始数据描述和二进制分解方法之外的第三种选择。其次，将Unet网络结构引入时间序列预测领域，设计并实现了一个适合时间序列预测的Unet网络。实验结果表明，相比于DLiner和PatchTST，UnetTSF表现出更优的性能，且线性复杂度的UnetTSF更适用于实际生产和生活。

请注意，由于LaTeX中一般不直接翻译模型名称等专业术语，上述翻译中“UnetTSF”、“DLiner”和“PatchTST”等保持不变，符合专业文献的常规处理方式。

参考文献

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

表 III

Univariate long-term forecasting results. ETT datasets are used with prediction lengths $T \in \{96, 192, 336, 720\}$. The best results are in bold.

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

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

表 IV

Compare the parameter quantity and resource consumption of the model under the look-back window $L = 336$ and $T = 96$ on the ETTh2.MACs are the number of multiply-accumulate operations.

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

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 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

请注意，这里的翻译严格遵循了您的要求，保持了LaTeX命令的原样，并对英文文本进行了恰当的翻译。

请注意，这里的翻译严格遵循了您的要求，保持了LaTeX命令的原样，并对英文文本进行了恰当的翻译。

表 II

多元变量长期预测结果。我们对ILI数据集使用预测长度$T \in \{24, 36, 48, 60\}$，对其余数据集使用$T \in \{96, 192, 336, 720\}$。最佳结果以**粗体**表示，次佳结果以下划线表示。

请注意，这里的翻译严格遵循了您的要求，保持了LaTeX命令的原样，并对英文文本进行了恰当的翻译。

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

表 III

单变量长期预测结果。使用ETT数据集，预测长度$T \in \{96, 192, 336, 720\}$。最佳结果以**粗体**表示。

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

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 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.

参考文献

[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.

请注意，我保持了原LaTeX代码的结构和特殊命令未做改动，仅翻译了文本内容。

请注意，我保持了原LaTeX代码的结构和特殊命令未做改动，仅翻译了文本内容。

表 IV

比较在回溯窗口$L = 336$和$T = 96$下，模型在ETTh2中的参数数量和资源消耗，MACS表示乘积累加操作次数。

请注意，我保持了原LaTeX代码的结构和特殊命令未做改动，仅翻译了文本内容。

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

[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.

[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_timeseries_

[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.

[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_timeseries_forecasting",2022.

[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.