

融合全局和变量间相关性的多变量时间序列预测研究

王宏博¹

WANG Hongbo

摘要

时间序列在现实生活中有着广泛的应用，通过时间序列预测模型可以有效预测未来的变化趋势，为决策提供依据。在多变量时间序列预测研究中，已有诸多模型被提出，但现有方法存在一些问题，如对多变量时间序列中的全局相关性利用不足、未充分考虑时间序列本身的变量信息等。针对这些问题，文章设计了两种不同的序列嵌入方式，通过 Transformer 架构分别对时间维度上的时间标记和通道维度上的变量标记进行特征提取，并将这些特征进行融合，实现对时间序列的精准预测。通过在 3 个真实时间序列数据集上对该模型进行实验评估，结果表明，模型在预测准确度上优于其他基线模型，相比于最优的基线模型 PatchTST，在 3 个数据集上的 MSE 平均降低了约 17.3%，MAE 平均降低了约 12.8%。

关键词

多变量时间序列预测；全局相关性；协变量；注意力机制

doi: 10.3969/j.issn.1672-9528.2025.05.013

0 引言

时间序列是指按照时间顺序排列的一组数据点，通常用于表示随时间变化的某种现象或变量的变化过程。在多变量时间序列中，每个时间戳包含多个变量且每个变量都随着时间变化。

时间序列预测 (time series forecasting) 已成为各研究领域的重点，广泛应用于传感器网络监测^[1]、金融与能源管理^[2]、交通流量预测^[3]和气象预测与监控^[4]等领域。在这些应用

场景中，目标时间序列通常指希望预测的主要变量或指标，即整个时间序列预测模型的核心。近年来，多变量时间序列预测 (multivariate time series forecasting, MTSF) 为长期规划提供了较多帮助，在上述场景下的应用显著增加。例如，由清华大学软件学院机器学习实验室与国家气象中心合作设计的基于自研深度自相关网络 (Autoformer^[5]) 的实时场馆气象预报系统，有效支撑了 2022 年北京冬奥会的气象监控和预测工作。

在时间序列预测领域，传统的统计学方法已被广泛应用于各类实际问题中。早期的研究主要集中于自回归积分滑动平均模型 (ARIMA)^[6] 和指数平滑 (Exponential Smoothing)^[7]

1. 贵州财经大学信息学院 贵州贵阳 550025

[15] 林凯滨, 陈云华, 钟金煜, 等. 基于时空邻域关联去噪时间面的事件数据表示 [J]. 计算机应用研究, 2024, 41(2): 3716-3721.

[16] FANG H C, WU J J, LI L D, et al. AEDNet: asynchronous event denoising with spatial-temporal correlation among irregular data[C]//Proceedings of the 30th ACM International Conference on Multimedia (MM'22). New York: ACM, 2022: 1427-1435.

[17] 徐凤. 图像去噪算法及噪声估计研究 [D]. 重庆: 重庆大学, 2020.

[18] MUEGGLER E, REBECQ H, GALLEGRO G, et al. The event camera dataset and simulator: event-based data for pose estimation, visual odometry, and SLAM[J]. International

journal of robotics research, 2017, 36(2): 142-149.

[19] FENG Y, LÜ H Y, LIU H L, et al. Event density based denoising method for dynamic vision sensor[J/OL]. Applied sciences, 2020[2024-10-06]. <https://doi.org/10.3390/app10062024>.

【作者简介】

苗好田 (1990—), 男, 辽宁沈阳人, 博士, 副教授、教师, 研究方向: 机器视觉, email:mht@sylu.edu.cn.

李凯 (2000—), 男, 辽宁锦州人, 硕士, 研究方向: 图像处理技术, email:loyalkid00@163.com.

(收稿日期: 2025-03-07 修回日期: 2025-05-07)

等经典方法, 这些方法在数据呈现出稳定趋势和季节性变化, 能够提供较为准确的预测。但这类方法会受模型假设、参数选择和表达能力低等限制, 且泛化能力差, 对时序中的部分非线性关系进行建模时存在一定局限性。

随着计算能力和深度学习技术的飞速发展, 深度学习方法逐渐成为时间序列预测的一个重要方向。与传统的统计学方法不同, 深度学习模型能够通过多层网络结构自动提取数据中的非线性特征, 并学习更为复杂的模式, 因此在处理具有高度非线性关系和复杂时变特征的时间序列数据时, 展现出更强的学习能力和预测精度。

基于RNN(recurrent neural network)和CNN(convolutional neural network)的方法不仅在预测性能上优于传统统计学模型, 且具有更强的鲁棒性。然而, 这些模型在MTSF任务中, 尤其是在处理全局相关性时, 仍然存在一定的局限性。以LSTNet^[8]为例, 通过结合RNN、CNN和注意力机制来捕获时序中的长短期依赖关系, 试图提升模型在复杂时序数据上的表现。但由于LSTNet主要聚焦于局部特征的提取, 在全局相关性建模方面存在一定的不足, 尤其是在需要捕捉多个变量之间长时间跨度的全局依赖时, 因此在MTSF任务中的预测能力有限。此外, TCN^[9]通过膨胀因果卷积的方式来扩大感受野, 能够更有效地整合时序中的局部信息, 使其在短期预测任务中具备一定的优势。但TCN虽然在局部建模上有显著表现, 但其扩张卷积的局部感受野大小有限, 限制了其在处理长时间跨度的全局依赖关系时的能力。为捕捉全局依赖, TCN需要引入更多的卷积层, 这无疑增加了模型的复杂性, 并且提高了训练的难度。而过深的层数会导致模型在学习全局信息时面临过拟合的风险。

在此背景下, Transformer^[10]作为一种在自然语言处理和计算机视觉领域取得重大突破的模型, 近年来被广泛应用于MTSF任务中。Transformer一般采用编码器-解码器架构, 其注意力机制使它在长时间序列中的不同时间点拥有一定的交互能力, 从而能够捕获时序的长期依赖性。LogTrans^[11]提出了卷积自注意力机制并在自注意力模型中引入了稀疏偏置; Reformer^[12]将注意力模块替换为局部敏感哈希来降低其复杂度; Informer^[13]提出了稀疏注意力机制和注意力的蒸馏操作来降低其计算复杂度及内存; Autoformer通过将时间序列分解和将自注意力机制替换成自相关机制在MTSF上取得了更好的效果; FEDformer^[14]利用傅里叶变换来分解序列和小波变换在频域中应用注意力操作, 通过随机选择固定大小的频率子集实现了线性复杂度。Transformer通常采用编码器-解码器架构, 并通过其自注意力机制使得模型能够在不同时间点之间建立交互, 从而有效捕捉时序中的长期依赖关系。然而, 该机制虽然能够在时间维度上建立联系, 却往往忽视了变量间的多重依赖。

在MTSF中, 除预测目标时间序列本身, 还需要综合考虑其他可能影响该目标序列的多个输入变量。例如, 在交通流量预测中, 目标时间序列可能是未来的交通流量, 而与之相关的协变量可能包括天气、节假日、时间段等因素。因此, 理解目标时间序列及其与其他变量之间的关系是构建高效预测模型的关键。协变量与目标时间序列变量之间可能存在一定的关联或依赖关系, 帮助提升模型的预测能力。通过考虑协变量, 模型能够更好地捕捉到目标变量的变化规律。尽管已有大量关于时间序列预测的研究, 但现有MTSF方法往往存在以下问题:

(1) 未能同时考虑协变量和时间序列本身的信息, 即不同变量之间的复杂相互依赖性。

(2) 忽略了全局信息的影响, 即全局下的时序相关性。

为解决这些问题, 提出了一种新的基于Transformer的框架, 针对MTSF任务的复杂性进行优化。在模型中, 不仅考虑到时间维度上的相关性, 还进一步关注通道维度——即变量之间的相关性。具体而言, 将每个时间戳的多个变量嵌入为时间标记, 并通过注意力机制对这些时间标记进行处理, 从而有效捕捉全局时间依赖性。同时, 将每个时间序列嵌入为变量标记, 更多地关注序列中各变量之间的多元相关性, 并通过前馈神经网络进一步进行序列编码。通过这种设计, 模型在捕捉全局依赖的同时, 也能够处理复杂的变量间依赖关系, 从而提高了MTSF任务的预测性能。

1 本文模型介绍

大多数基于Transformer的预测模型通常将同一时间的多个变量视为时间标记, 并按照这一套流程进行预测任务。然而, 越来越多的研究将计算机视觉中的补丁技术应用于时间序列预测, 因此发现关于数值模态的方法对于学习注意力图指导性的意义较小。同时基于多层感知机(multilayer perceptron, MLP)方法的成功, 也对采用重型编码器-解码器Transformer来生成标记的必要性提出了质疑。对此, 提出的模型如图1所示, 专注于多变量序列的表示学习和自适应相关。

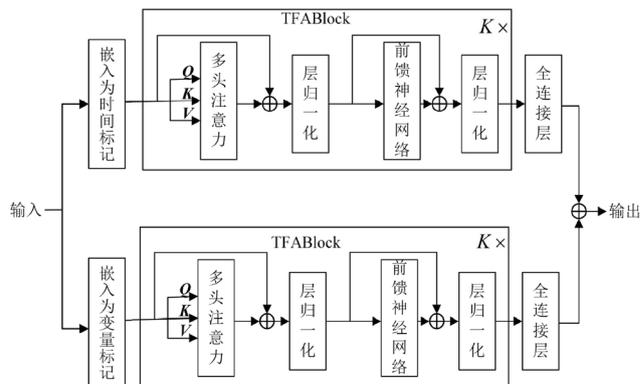


图1 模型框架图

其模型主要包括以下几个模块：

(1) **Embedding**：不同时间戳上的原始序列和不同变量上的原始序列被独立地嵌入标记。

(2) **TFABlock**：自注意力被应用于具有增强的可解释性的嵌入时间和变量标记，从而揭示多变量相关性；其次通过共享前馈网络提取每个时间和变量系列表示。最后采用层归一化方法减少变量间的差异。同时将序列的预测过程可视化，如图 2 所示。图 2 为不同序列嵌入方式之间的比较，与基于 Transformer 嵌入方式不同，不仅将每个时间步长嵌入到时间标记中，还将整个序列独立地嵌入到变量标记中，这样可以通过注意力机制来描述多变量地全局依赖性和变量之间的相关性，并且通过前馈网络对序列表示进行编码预测。

1.1 问题定义

在多变量时间序列预测中，给定序列历史观测值

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{T1} & \cdots & x_{TN} \end{pmatrix} \in \mathbb{R}^{T \times N}$$

包含 T 个时间步， N 个变量，目标是

预测未来 L 个时间步值 $Y = \{x_{T+1}, x_{T+2}, \dots, x_{T+L}\} \in \mathbb{R}^{L \times N}$ 。为操作更为严谨，将 X_t^n 表示为第 t 个时间步，第 n 个变量的时间值， $X_{1:N} = \{X_{1:T,1}, X_{1:T,2}, \dots, X_{1:T,N}\}$ ， $X_{1:N,j}$ 表示第 j 列的值， $X^{1:T} = \{X_{1,1:N}, X_{2,1:N}, \dots, X_{T,1:N}\}$ ， $X_{1,1:T}$ 表示第 i 行的值。

1.2 Embedding 模块

简单来说，不仅在时间维度上嵌入为时间标记，还在通道上嵌入变量标记，后应用自注意力的相互作用，并单独处理的前馈神经网络的系列表示。对于将每个时间戳的多个变量嵌入为时间标记，用公式表示为：

$$\begin{cases} X_s = \text{Embedding}(X^{1:T}) \\ X_v = \text{Embedding}(X_{1:N}) \end{cases} \quad (1)$$

对于时间维度嵌入表示： $\mathbb{R}^N \rightarrow \mathbb{R}^D$ ，对于通道维度嵌入表示： $\mathbb{R}^T \rightarrow \mathbb{R}^D$ ，其中 $X_s \in \mathbb{R}^{T \times D}$ ， $X_v \in \mathbb{R}^{D \times N}$ ， D 表示隐藏层维度。

1.3 TFABlock 模块

在这个模块中，设置了一个由归一化、前馈神经网络和自注意机制组成的 K 个堆栈，其公式为：

$$\begin{cases} H_s^{l+1} = \text{TFABlock}(H_s^l), l = 0, \dots, K-1 \\ H_v^{l+1} = \text{TFABlock}(H_v^l), l = 0, \dots, K-1 \end{cases} \quad (2)$$

式中： $H_s^{l+1} \in \mathbb{R}^{T \times D}$ 包含 T 个时间维度上的 D 个嵌入标记； $H_v^{l+1} \in \mathbb{R}^{D \times N}$ 包含 N 个通道维度上的 D 个嵌入标记。

1.3.1 层归一化

层归一化^[15]最初是为提高深度网络的收敛性和训练稳定性而提出的。在典型的基于 Transformer 的预测中，该模块对同一时间步的变量表示进行归一化，逐渐使每个变量彼此无法区分。一旦收集的时间点未按时间对齐，该操作还会引入非因果或延迟过程之间的交互噪声。在文中模型中，归一化应用于单个变量的序列表示，可有效解决非平稳问题。此外，由于所有序列作为变量标记都被归一化为正态分布，因此可以减少由不一致的测量引起的差异。相反，在此前的架构中，时间步骤的不同标记被标准化，从而导致序列过于平滑。

1.3.2 前馈神经网络

Transformer 利用前馈神经网络作为对时间节点表示进行编码的基本构建。在传统 Transformer 中，形成节点标记的相同时间戳的多个变量可能位置有误，过于局部化而无法揭示用于预测的足够信息。在此基础上，前馈网络同样也在变量节点的通道上起作用。前馈神经网络由几个主要部分组成：一是输入层：接收外部输入数据。每个节点（神经元）表示

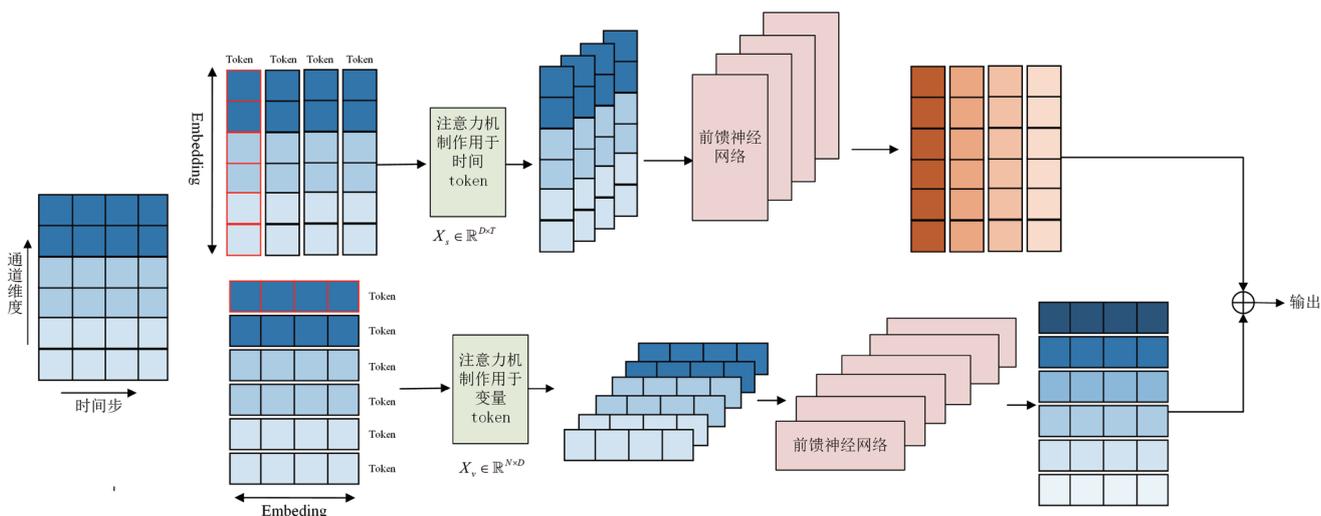


图 2 序列的预测过程图

输入数据中的一个特征。二是隐藏层：位于输入层和输出层之间，包含若干个神经元。神经元通过激活函数对接收到的输入进行变换。前馈网络可以有多个隐藏层，通过层层非线性变换提取数据的高级特征。三是输出层：根据输入数据和网络的权重参数，输出最终的预测结果。在回归任务中，输出层通常只有一个节点，用于预测一个连续值。简单来说，将多个线性层作用于时间维度和通道维度上，确保模型能捕捉到更多有用的序列信息。

1.3.3 自注意力机制

自注意力机制是一种广泛用于深度学习中的技术，尤其在自然语言处理和计算机视觉等任务中取得了显著成效。自注意力机制的核心思想是，通过计算输入数据之间的相似性来为每个数据元素分配一个“注意力权重”，从而让模型能够关注到输入序列中不同位置的相关信息。这一机制的关键在于能够捕捉到数据中远距离依赖的关系，而不仅仅是局部信息。在自注意力机制中，每个输入元素（如一个词语、一张图像的像素等）都会与其他输入元素进行关联。通过计算相似性，自注意力机制赋予每个输入元素一个权重，表示与其他元素的相关性。这样，模型可以“关注”输入序列中的某些部分，而忽略其他部分。多头自注意力基于缩放点积注意力机制实现。缩放点积注意力的输入由查询（ Q ）、键（ K ）和值（ V ）组成，这些向量是通过学习得到的参数矩阵与输入数据的线性组合：

$$Q = XW_Q, K = XW_K, V = XW_V \quad (3)$$

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (4)$$

式中： X 为输入序列； W_Q 、 W_K 、 W_V 为训练中的可学习权重矩阵。分别代表查询、键和值； d 为特征向量的维度。最后通常使用多头注意力（Multi-Head Attention）来增强模型的代表能力。具体来说，将查询、键和值分别线性变换为多个子空间，并计算多个不同的注意力分数，最后将这些结果拼接起来，经过一次线性变换得到最终输出。

$$\begin{cases} \text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V) \\ \text{MultiHead}(Q, K, V) = \text{concat}(\text{head}_1, \dots, \text{head}_h)W_O \end{cases} \quad (5)$$

式中： W_i^Q 、 W_i^K 、 $W_i^V \in \mathbb{R}^{d \times d_h}$ 表示第 i 个注意力头 Q 、 K 、 V 的权重矩阵； $W_O \in \mathbb{R}^{h \times d_h \times d}$ 表示多头自注意力权重矩阵； d_h 是每个注意力头的输出特征维度。

1.3.4 输出

最后利用全连接层将经注意力模块的序列进行输出，目的是对输出的序列分别在时间和变量维度上做出调整。

$$Y = \text{Concat}(\text{FC}(H_s^{l+1}), \text{FC}(H_v^{l+1})) \quad (6)$$

2 实验分析

2.1 数据集和实验设置

本文基于三个真实数据集评估文中所提模型。

用电量数据集 Electricity：该数据集记录了某一区域 2012—2014 年 321 位用户的用电量数据，采样频率为 15 min，每位用户约有 14 万个采样点，单位为 kWh。因同一区域内，多位用户间具有相邻性，所以对应产生的用电量序列间具有相关性，部分工作采用此数据集进行多变量序列预测模型的训练以及评估。

太阳能数据集 Solar-Energy：数据集记录了阿拉巴马州 2006 年在 137 个光伏电站采集的太阳能发电量，采样频率为 10 min，共有 36 601 个时间戳。这些电站之间通常存在一定的空间邻近性，而每个地点的发电量与其他地点的发电量之间可能会表现出相关性，因此同样被广泛使用。

交通数据集（performance measurement system, PeMS）：加利福尼亚州高速公路和主要城市道路的交通流量数据，这些数据每隔 5 min 收集一次，15 701 个时间戳，每个时间戳包含 358 个变量。

对于 Electricity 和 Solar-Energy 数据集，将序列回溯窗口长度设置为 96，并且预测长度为 {96, 192, 336, 720}；对于 PEMS 数据集，将回溯窗口设置为 12，序列预测长度为 {12, 24, 36, 48}。同时划分训练集验证集和测试机的比例为 7:1:2。所有实验均在 PyTorch 中实现，并在单个 NVIDIA 4090 24 GB GPU 上进行。利用 ADAM 优化器，将初始学习率设置为 { 10^{-3} , 5×10^{-4} , 10^{-4} }，L2 损失用于模型优化。批量大小统一设置为 32，训练轮数设置为 10。在对模型中设置 Transformer 块的数量 K 为 {2, 3, 4}。隐藏维度 D 的取值为 {256, 512}。

2.2 基线模型

将本文提出的模型与各种基线进行比较，比较的基线模型如下：

PatchTST^[16]：该模型将时间序列分割成子序列级补丁，这些补丁用作 Transformer 的输入标记块，提取局部语义信息，并采用通道独立策略。

Crossformer^[17]：一个基于 Transformer 的模型，利用跨维依赖性进行多变量时间序列预测。

FEDformer^[1]：一种基于傅立叶和小波形式的频率增强分解的 Transformer 模型，在频域中实现了具有低秩近似的稀疏注意力，线性计算复杂度和内存成本均为 $\mathcal{O}(L)$ 。

Autoformer：一种采用具有自相关机制的深度节趋势输入序列分解架构的 Transformer 模型，从输入序列中提取季节性和趋势成分，并将序列分解块嵌入为内部运算符。

Informer: 一种基于稀疏注意力机制的 Transformer, 复杂度达到 $\mathcal{O}(L \log L)$, 有效编码变量之间的依赖关系, 并引入一种具有 DMS 预测策略的新型架构。

2.3 模型性能评估

为评估本文所提出方法的有效性, 将提出的模型与基线模型进行比较, 采用 MSE 和 MAE 来作为评估模型的指标。如表 1 所示, 相比于基线模型 DLinear, 本文模型在所有数据集上的 MSE 平均降低了约 17.3%, MAE 平均降低了约 12.8%; 在 Electricity 数据集上的 MSE 降低了约 15.6%, MAE 降低了约 5.8%; 在 Solar-Energy 数据集上的 MSE 降低了约 10.1%, MAE 降低了约 16.3%; 在 PEMS 数据集上的 MSE 降低了约 30.0%, MAE 降低了约 20.9%。结果验证了模型的有效性和优越性。

2.4 消融实验

为进一步验证本文提出的基于 Transformer 框架的有效性, 通过设计消融实验, 逐步去除模型中的关键组件, 以评估不同部分对模型性能的贡献。并在电力、太阳能和交通流量 3 个数据集上进行消融实验。如表 2 所示, TFANet I 表示去除变量间的多元相关性建模, 即使用传统的时间序列编码模式。TFANet II 表示去除时间维度上的嵌入, 仅考虑变量之间的关系。TFANet III 表示将前馈神经网络替换为简单的线性层, 评估其对模型表现的影响。

从实验结果可以看出, 去除时间标记嵌入和变量标记嵌入会导致模型预测准确率显著下降。这表明, 模型中对时间和变量相关性的双重建模是提升预测性能的关键因素。此外, 去除前馈神经网络后, 模型在全局依赖建模上

表 1 多变量序列预测任务结果

| Methods | | TFANet | | PatchTST | | FEDformer | | Autoformer | | Informer | | Crossformer | |
|--------------|-----|--------------|--------------|----------|-------|-----------|-------|------------|-------|--------------|--------------|-------------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Electricity | 96 | 0.158 | 0.260 | 0.195 | 0.285 | 0.193 | 0.308 | 0.201 | 0.317 | 0.274 | 0.368 | 0.219 | 0.314 |
| | 192 | 0.164 | 0.273 | 0.199 | 0.289 | 0.201 | 0.315 | 0.222 | 0.334 | 0.296 | 0.386 | 0.231 | 0.322 |
| | 336 | 0.188 | 0.289 | 0.215 | 0.305 | 0.214 | 0.329 | 0.231 | 0.338 | 0.300 | 0.394 | 0.246 | 0.337 |
| | 720 | 0.220 | 0.323 | 0.256 | 0.337 | 0.246 | 0.355 | 0.254 | 0.361 | 0.373 | 0.439 | 0.280 | 0.363 |
| Solar-Energy | 96 | 0.211 | 0.236 | 0.234 | 0.286 | 0.242 | 0.342 | 0.884 | 0.711 | 0.236 | 0.259 | 0.310 | 0.331 |
| | 192 | 0.246 | 0.272 | 0.267 | 0.310 | 0.285 | 0.380 | 0.834 | 0.692 | 0.217 | 0.269 | 0.734 | 0.725 |
| | 336 | 0.255 | 0.286 | 0.290 | 0.315 | 0.282 | 0.376 | 0.941 | 0.723 | 0.249 | 0.283 | 0.750 | 0.735 |
| | 720 | 0.249 | 0.278 | 0.289 | 0.317 | 0.357 | 0.427 | 0.882 | 0.717 | 0.241 | 0.317 | 0.769 | 0.765 |
| PEMS | 12 | 0.087 | 0.180 | 0.099 | 0.216 | 0.126 | 0.251 | 0.272 | 0.385 | 0.126 | 0.233 | 0.090 | 0.203 |
| | 24 | 0.113 | 0.210 | 0.142 | 0.259 | 0.149 | 0.275 | 0.334 | 0.440 | 0.139 | 0.250 | 0.121 | 0.240 |
| | 48 | 0.130 | 0.245 | 0.211 | 0.319 | 0.227 | 0.348 | 1.032 | 0.782 | 0.186 | 0.289 | 0.202 | 0.317 |
| | 96 | 0.174 | 0.293 | 0.269 | 0.370 | 0.348 | 0.434 | 1.031 | 0.796 | 0.233 | 0.323 | 0.262 | 0.367 |

表 2 模型在电力, 太阳能和交通流量数据集上的消融结果

| Methods | | TFANet | | TFANet I | | TFANet II | | TFANet III | |
|--------------|-----|--------------|--------------|----------|-------|--------------|--------------|--------------|--------------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Electricity | 96 | 0.158 | 0.260 | 0.162 | 0.265 | 0.160 | 0.263 | 0.161 | 0.266 |
| | 192 | 0.164 | 0.273 | 0.173 | 0.273 | 0.172 | 0.271 | 0.182 | 0.282 |
| | 336 | 0.188 | 0.289 | 0.194 | 0.310 | 0.199 | 0.294 | 0.193 | 0.305 |
| | 720 | 0.220 | 0.323 | 0.240 | 0.333 | 0.214 | 0.330 | 0.246 | 0.343 |
| Solar-Energy | 96 | 0.211 | 0.236 | 0.223 | 0.238 | 0.226 | 0.240 | 0.208 | 0.232 |
| | 192 | 0.246 | 0.272 | 0.250 | 0.302 | 0.257 | 0.283 | 0.251 | 0.271 |
| | 336 | 0.255 | 0.286 | 0.271 | 0.311 | 0.262 | 0.295 | 0.277 | 0.303 |
| | 720 | 0.249 | 0.278 | 0.282 | 0.315 | 0.271 | 0.297 | 0.301 | 0.317 |
| PEMS | 12 | 0.087 | 0.180 | 0.107 | 0.192 | 0.083 | 0.180 | 0.094 | 0.195 |
| | 24 | 0.113 | 0.210 | 0.120 | 0.220 | 0.112 | 0.223 | 0.206 | 0.244 |
| | 48 | 0.130 | 0.245 | 0.144 | 0.287 | 0.146 | 0.265 | 0.159 | 0.291 |
| | 96 | 0.174 | 0.293 | 0.189 | 0.345 | 0.179 | 0.311 | 0.230 | 0.356 |

的能力大大下降,尤其在序列预测长度较长任务中的表现明显下降,这进一步证明了前馈神经网络在处理全局信息时的重要性。

3 结论

本文研究了多变量多步时间序列预测问题,并针对现有方法的不足,提出了一种新的多变量时间序列预测模型。该模型采用基于自注意力机制的编码器,用于提取时间序列的时序特征,同时融合历史信息 and 不同协变量的信息。为更好地建模时间序列数据,设计了两种不同的序列嵌入方式,通过 Transformer 架构分别对时间维度上的时间标记和通道维度上的变量标记进行特征提取。通过实验评估了所提出方法的性能,并与多个基准模型进行了比较。实验结果表明,所提模型相比于基准模型能够提供更为精确的预测结果。

参考文献:

- [1] PAPANITRIOU S, YU P. Optimal multi-scale patterns in time series streams[C]//Proceedings of the 2006 ACM SIGMOD international conference on Management of data. New York: ACM, 2006: 647-658.
- [2] ZHU Y Y, SASHA D. Statstream: statistical monitoring of thousands of data streams in real time[C]//VLDB'02: Proceedings of the 28th International Conference on Very Large Databases. HK:Morgan Kaufmann, 2002: 358-369.
- [3] CHEN C, PETTY K, SKABARDONIS A, et al. Freeway performance measurement system: mining loop detector data[J]. Transportation research record, 2001, 1748(1): 96-102.
- [4] ANGRYK R A, MARTENS P C, AYDIN B, et al. Multivariate time series dataset for space weather data analytics[J]. Scientific data, 2020, 7(1): 227.
- [5] WU H X, XU J H, WANG J M, et al. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting[C]//NIPS'21: Proceedings of the 35th International Conference on Neural Information Processing Systems. New York: ACM, 2021, 34:22419-22430.
- [6] DURBIN J, KOOPMAN S J. Time series analysis by state space methods[EB/OL].(2013-12-17)[2024-06-12].https://doi.org/10.1093/acprof:oso/9780199641178.001.0001.
- [7] HYNDMAN R, KOEHLER A, ORD K, et al. Forecasting with exponential smoothing: the state space approach[EB/OL].(2009-07-02)[2024-09-13].https://robjhyndman.com/expsmooth/.
- [8] LAI G K, CHANG W C, YANG Y M, et al. Modeling long-and short-term temporal patterns with deep neural networks[C]//SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. New York: ACM, 2018: 95-104.
- [9] BAI S J, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[EB/OL]. (2018-04-09)[2024-02-13].https://doi.org/10.48550/arXiv.1803.01271.
- [10] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 6000-6010.
- [11] LI S Y, JIN X Y, XUAN Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, 2019: 5243-5253.
- [12] KITAEV N, KAISER L, LEVSKAYA A. Reformer: the efficient transformer[EB/OL]. (2020-02-18)[2024-10-12].https://doi.org/10.48550/arXiv.2001.04451.
- [13] ZHOU H Y, ZHANG S H, PENG J Q, et al. Informer: beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2021: 11106-11115.
- [14] ZHOU T, MA Z Q, WEN Q S, et al. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting[EB/OL].(2022-06-16)[2024-05-25].https://doi.org/10.48550/arXiv.2201.12740.
- [15] BA J L, KIROS J R, HINTON G E. Layer normalization[EB/OL]. (2016-07-21)[2024-05-23].https://doi.org/10.48550/arXiv.1607.06450.
- [16] NIE Y Q, NGUYEN N H, SINHTONG P, et al. A time series is worth 64 words: long-term forecasting with transformers[EB/OL]. (2023-03-05)[2024-09-11].https://doi.org/10.48550/arXiv.2211.14730.
- [17] ZHANG Y H, YAN J C. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting[EB/OL].(2023-03-02)[2024-09-12].https://openreview.net/forum?id=vSVLM2j9eie.

【作者简介】

王宏博(2000—),男,江西吉安人,硕士研究生,研究方向:计算机应用技术。

(收稿日期:2025-02-06 修回日期:2025-05-08)