DOI:10.11918/202411025

特征选择的高效时间序列预测模型

沈 瑜[1],李江柽[1],梁 栋[2,3],王若暄[1],吴红刚[2,3],
马殷军[4],严 源[2,3],马煜堃[1],刘佳英[1],闫佳荣[1]

(1.兰州交通大学 电子与信息工程学院,兰州 730070;2.中铁西北科学研究院有限公司,兰州 730046;
3.中国中铁科学研究院集团有限公司,成都 610031;4.中国铁路青藏集团有限公司,西宁 810007)

摘 要:为解决时间序列预测方法普遍存在计算复杂度高、推理速度慢以及关键信息容易丢失的问题,提出一种特征选择的高效时间序列预测模型(Targetformer)。该模型采用 Transformer 仅编码器结构,并引入倒置输入编码块,将输入时间序列转换为独立的 Token;提出特征选择性注意力机制,从映射后的时序 Token 中提取关键特征;最后利用投影层将提取的特征映射为最终的预测结果。在 6 个数据集上的实验结果表明:相较于最优基线模型,Targetformer 的均方误差和平均绝对误差分别降低 5.7% 和 2.0%,均取得最优的预测性能,验证了其有效性和实用性。该模型为时间序列预测领域提供一种高效且实用的解决方案。

关键词:时间序列预测;Targetformer;特征选择性注意力机制;倒置输入编码块;Transformer 仅编码器结构
中图分类号:TP183 **文献标志码**:A **文章编号**:0367-6234(2026)09-0173-10

Efficient time series forecasting model for feature selection

Shen Yu[1], Li Jiangcheng[1], Liang Dong[2,3], Wang Ruoxuan[1], Wu Honggang[2,3],
Ma Yinjun[4], Yan Yuan[2,3], Ma Yukun[1], Liu Jiaying[1], Yan Jiarong[1]

(1. School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China;
2. Northwest Research Institute Co., Ltd. of C. R. E. C., Lanzhou 730046, China; 3. China Railway Academy Group Co., Ltd., Chengdu 610031, China; 4. China Railway Qinghai-Tibet Group Co., Ltd., Xining 810007, China)

Abstract: In order to solve the problems of high computational complexity, slow inference speed and easy loss of key information prevalent in time series prediction methods, an efficient time series prediction model with feature selection (Targetformer) is proposed. The model adopts the Transformer encoder-only structure and introduces an inverted embedding to convert the input time series into independent Token; A feature-selective attention mechanism is proposed to extract the key features from the mapped time-series Tokens. Finally, a projection layer is used to map the extracted features into the final prediction results. Experimental results on six datasets show that, compared with the best baseline model, Targetformer reduces mean squared error and mean absolute error by 5.7% and 2.0%, respectively, and achieves the best prediction performance in terms of both metrics, verifying its effectiveness and practical applicability. The model provides an efficient and practical solution in the field of time series prediction.

Keywords: time series forecasting; Targetformer; feature-selective attention mechanism; inverted embedding; Transformer encoder-only structure

时间序列预测是时间序列分析领域的重要研究方向,通过对未来的预测,为决策提供科学依据[1]。时间序列预测在能源消耗、经济规划、天气预报、疾病传播以及突发灾害预测等领域有着广泛的应用。在这些现实应用中,迫切的需求是提高预测结果的准确性,这对长期规划和预警有着重要意义。

传统的基于统计学习的时间序列预测方法通过捕捉时间序列中的线性关系进行预测,最常用的是自回归移动平均模型(autoregressive integrated moving average model, ARIMA)。但 ARIMA 要求数据具有平稳性,这导致其在处理复杂的非线性关系时表现欠佳[2]。此外,传统模型通常依赖大量的手工特

收稿日期:2024-11-14;**录用日期**:2024-12-30;**网络首发日期**:2025-03-21
网络首发地址:https://link.cnki.net/urlid/23.1235.T.20250321.1430.002
基金项目:甘肃省技术创新引导计划-企业研发机构能力建设专项资助(23CXJA0011);中国中铁股份有限公司科技研究开发计划(2022-重大专项-07)
作者简介:沈 瑜(1982—),女,教授,博士生导师
通信作者:李江柽,15351428752@163.com

征,难以适应现代数据驱动的应用场景。

机器学习技术的突破推动了非线性预测模型在时间序列预测领域的广泛应用,如支持向量机(support vector machines,SVM)、随机森林(random forest,RF)和梯度提升机(gradient boosting machine,GBM)等。这些方法通过对数据进行特征提取和训练,能够捕捉到更复杂的模式和非线性关系。然而,这些方法在处理大规模时序数据时,训练时间较长,且对参数选择较为敏感,在捕捉长时依赖关系和复杂时间序列模式方面仍存在局限性。

随着计算能力的提升和数据量的增加,深度学习方法在时间序列预测领域展现出强大的能力。卷积神经网络(convolutional neural network,CNN)和循环神经网络(recurrent neural network,RNN)是其中的代表性方法。CNN 以其独特的卷积与池化机制,在捕捉局部时间特征方面展现出优势,代表性的方法有时间卷积网络(temporal convolutional network,TCN)、WaveNet[3] 及最新的现代时间卷积网络(modern temporal convolutional network,ModernTCN)[4]等。RNN 通过递归结构尝试建模序列的长期依赖,然而,在训练过程中容易受到梯度消失与爆炸问题的影响,限制了其在长序列预测中的表现。尽管长短时记忆网络(long short-term memory,LSTM)、门控循环神经网络(gated recurrent unit,GRU)[5] 及长短期时间序列网络(long-and short-term time-series network,LSTNet)[6] 等变体在一定程度上缓解了这些问题,但 RNN 固有的模型劣势依然存在。此外,RNN 多为串行计算,训练过程中对内存消耗极大。针对这些问题,Salinas 等[7] 提出了深度自回归模型(deep autoregressive model,DeepAR),该模型利用序列数据的历史信息,采用自回归模型进行预测,能够捕捉时间序列中的潜在模式。Oreshkin 等[8] 提出了一种用于时间序列的神经基展开分析框架(neural basis expansion analysis for time series,N-BEATS),该框架采用全局和分解的预测策略,通过堆叠块的方式捕捉时间序列的各种特征,展现出了较好的预测能力。上述模型一定程度上克服了 RNN 的局限性,但仍需要进一步研究与优化。

近年来,Transformer[9] 模型在自然语言处理和计算机视觉领域得到了广泛应用,并逐渐扩展到时间序列预测任务中。Transformer 通过自注意力机制,可以有效捕捉时间序列中的全局时间依赖关系。与 RNN 不同,Transformer 不依赖序列的顺序处理,因此可以实现更高的并行计算效率。然而,Transformer 模型存在参数量大、计算资源消耗高的问题,需要大量的数据和计算资源来训练。特别是在长序列预测任务中,因为其序列长度较大,且 Transformer 本身二次计算复杂度的问题,导致性能下降。在基于 Transformer 的时间序列预测模型中,Zhou 等[10] 提出了 Informer 模型,该模型采用特殊的稀疏性采样注意力机制的方法缓解 Transformer 二次计算复杂度的问题。Kitaev 等[11] 提出了 Reformer 模型,该模型利用局部敏感散列算法缓解 Transformer 二次计算复杂度的问题。Wu 等[12] 提出了 Autoformer 模型和 Zhou 等[13] 提出了 FEDformer 模型,两者更注重序列本身存在的时间依赖关系,将序列分解思想应用于模型中,以更加准确地提取历史序列之间的相关性。为了克服传统 Transformer 中 Token 构建方式的局限性,Zhang 等[14] 提出了 Crossformer 模型、Nie 等[15] 提出了 PatchTST 模型,两者将 Token 划分为 Patch(数据块)的思想,但此类方式生成的 Patch Token 存在信息流失、粒度选择困难等缺陷。Liu 等[16] 提出了 iTransformer 模型,该模型通过将注意力机制和前馈网络应用于倒置维度,提升了预测性能。Ashok 等[17] 通过引入简化的注意力机制,提高了对时序间非线性依赖关系的捕捉能力。Rasul 等[18] 提出了一种将向量量化与注意力机制相结合的时间序列预测模型,该模型减少了注意力矩阵的复杂度。Dai 等[19] 通过将时间序列的周期性成分与非周期性成分进行分离和独立建模,提高了模型对复杂长期序列的预测精度。Chen 等[20] 提出了 Pathformer 模型,该模型通过多尺度特征提取和自适应路径选择,增强了对不同时间尺度特征的捕捉能力。以上基于 Transformer 的时间序列预测模型虽然在一定程度上降低了计算复杂度,但不可避免地丢失一些关键信息而导致预测性能下降,难以达到预测性能与计算效率的平衡。

针对时间序列预测模型中存在的计算复杂度高、关键信息容易丢失的问题,本文提出特征选择的高效时间序列预测模型(Targetformer)。Targetformer 采用 Transformer 仅编码器结构,引入倒置输入编码块增强特征保持能力,设计特征选择性注意力机制(feature-selective attention mechanism,FSAM)优化计算路径。期望为时间序列预测领域提供一种新的解决方案,以更好地平衡预测性能与计算效率。

1 Targetformer 模型框架

Targetformer 基于 Transformer 仅编码器架构构建,包含倒置输入编码块(inverted embedding,IE)、特征选择 Transformer 块(Targetformer block)和投影层(projection)3 部分。其中 Targetformer 块可根据实际需求堆叠,堆叠数用 N 表示,查询向量、键向量和值向量分别用 $\boldsymbol{q}$、$\boldsymbol{k}$、$\boldsymbol{v}$ 表示,模型结构如图 1 所示。

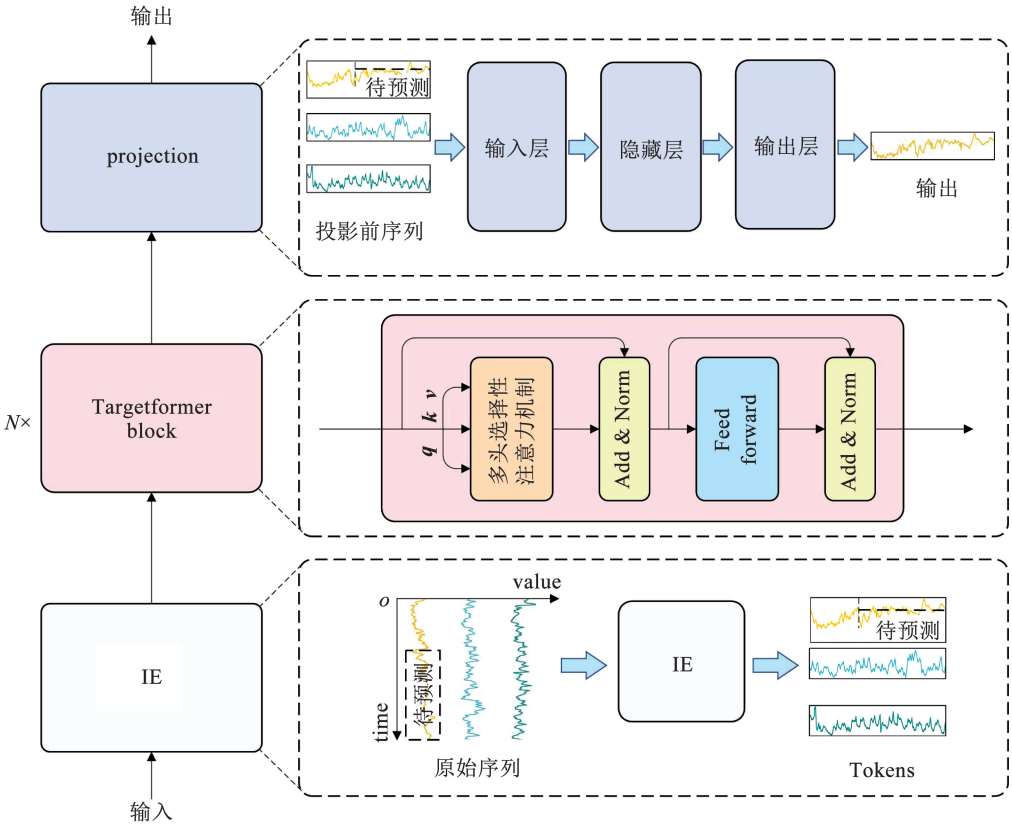


图 1　Targetformer 模型结构

Fig. 1　Targetformer model structure

1.1　Transformer 仅编码器结构

Transformer 仅编码器结构如图 2 所示。在原始 Transformer 结构的基础上舍弃解码器(decoder)部分，并加入了投影层(projection)，使得模型更加轻量化，并且能够更好地理解序列输入的上下文信息，从而提高时间序列预测的准确性。

1.2　倒置输入编码块

在传统的时序预测方法中，所有变量在同一时刻的时间点通常被表示为一个 Token。这种方法虽然能捕捉变量之间的时刻关系，但在处理复杂的时序数据时，无法充分利用变量的全局信息，并且由于其过小的感受野，限制了注意力机制建模 Token 之间的关系。另外，来自不同变量的时间点被映射到 Token 后，原本独立的变量被杂糅为多维特征，使得模型无法捕捉变量间的联系。传统时序预测中的 Token 构建方式为

$$\boldsymbol{B}_t = C([\beta_{1,t},\beta_{2,t},\cdots,\beta_{n,t}]) \tag{1}$$

式中：$\boldsymbol{B}_t$ 为时刻 t 的 Token；n 为变量的数量；$\beta_{n,t}$ 为第 n 个变量在时刻 t 的值；$C(\cdot)$ 为 Concat 函数，用于将序列连接形成一个新的序列。

为克服传统 Token 构建方式的局限性，本文采用倒置输入编码块的方法，将整条序列直接映射为 Token，强化模型的全局信息捕捉能力。本文 Token 的构建方式与其他构建方式的对比如图 3 所示。其中，图 3(a)为原始的 Transformer 以时间点为 Token 的构建方式，图 3(b)为以 Patch 为 Token 的构建方式，图 3(c)为本文以变量整条序列为 Token 的构建方式。

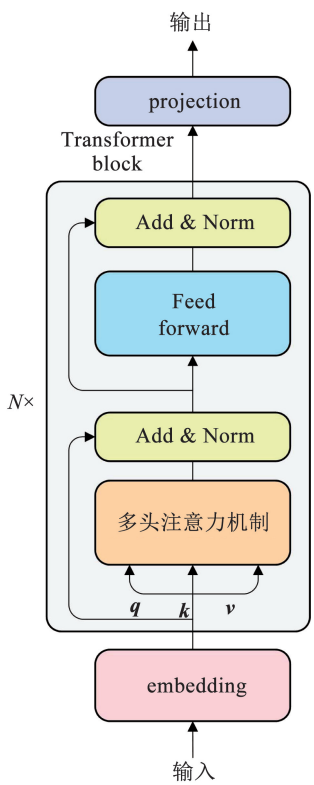


图 2　Transformer 仅编码器结构

Fig. 2　Transformer encoder-only structure

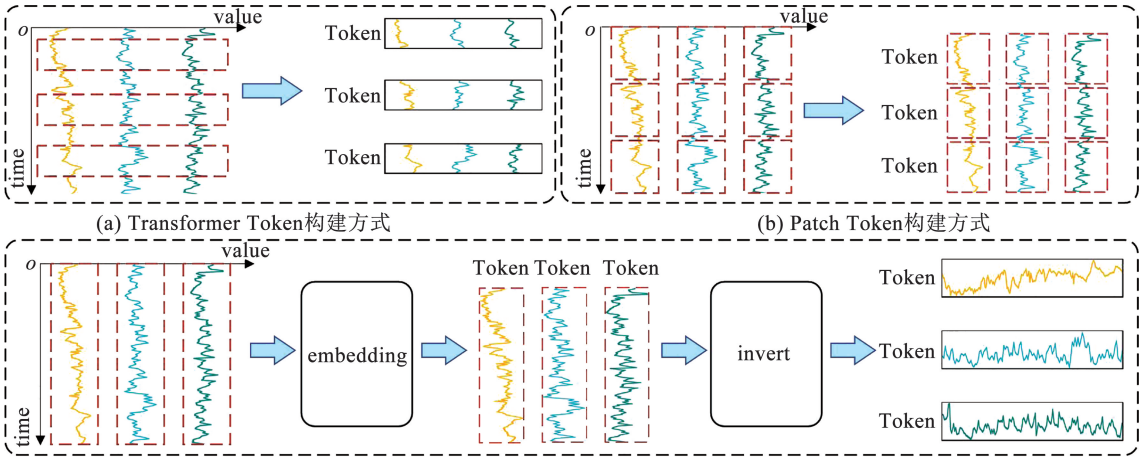


图 3 Token 构建方式对比

Fig. 3 Comparison of Token construction methods

倒置输入编码块的核心思想是将每个变量的整条时间序列独立地映射为一个 Token。具体而言，给定一个包含多个变量的时序数据集，首先将每个变量的时间序列提取出来，然后对于每个变量，将其整个时间序列映射为一个 Token，从而在模型的输入层形成一个由变量时序 Token 组成的序列。倒置输入编码块的 Token 构建方式为

$$\boldsymbol{K}_i = f(\boldsymbol{S}_i) = f([s_{i,1}, s_{i,2}, \cdots, s_{i,m}]) \tag{2}$$

式中：$\boldsymbol{K}_i$ 为第 i 个变量的 Token，$\boldsymbol{S}_i$ 为第 i 个变量的时间序列，m 为时间步的数量，$f(\cdot)$ 为映射函数。

1.3 特征选择性注意力机制

为解决传统的 Transformer 模型在处理大规模时序数据时面临计算复杂度高的问题，常用的方法是采用稀疏注意力机制，即选择注意力得分较高的前若干分数进行预测，如此不仅使得模型更加复杂，并且容易忽略序列之间的关键信息。为此，本文提出 FSAM，如图 4 所示。

此机制基于前期的转置输入，核心在于仅计算待预测目标特征与其他特征之间的注意力分数，更聚焦于对预测目标有实质性贡献的注意力分数，使得模型更准确地捕捉时间序列数据中的关键信息，舍弃冗余信息，从而提升预测精度。该机制的特征选择过程能够根据预测任务的改变而动态适应，使得模型能够适应多样化的应用场景，并且不会增加额外的参数，不改变 Transformer 模型的基本结构，可以应用于其他基于 Transformer 的时间序列预测模型中。

FSAM 的实现细节如下：假设一时间序列 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$，每个时间步 $\boldsymbol{x}_t$ 包含 D 维特征，需要预测的特征为第 D' 维特征，即 D' 为特定的待预测特征索引，记 $\boldsymbol{y}_t = \boldsymbol{x}_t^{D'}$。在计算过程中，首先为待预测特征计算查询向量 $\boldsymbol{q}_t$，同时为所有特征计算键向量 $\boldsymbol{k}_t^d$ 和值向量 $\boldsymbol{v}_t^d$（$d = 1, 2, \cdots, D$），其中 d 为特征维度。查询向量、键向量和值向量的计算式分别为：

$$\boldsymbol{q}_t = \boldsymbol{W}_{\mathrm{q}} \boldsymbol{y}_t \tag{3}$$

$$\boldsymbol{k}_t^d = \boldsymbol{W}_{\mathrm{k}} \boldsymbol{x}_t^d \tag{4}$$

$$\boldsymbol{v}_t^d = \boldsymbol{W}_{\mathrm{v}} \boldsymbol{x}_t^d \tag{5}$$

式中 $\boldsymbol{W}_{\mathrm{q}}$、$\boldsymbol{W}_{\mathrm{k}}$、$\boldsymbol{W}_{\mathrm{v}}$ 为可学习的权重矩阵。

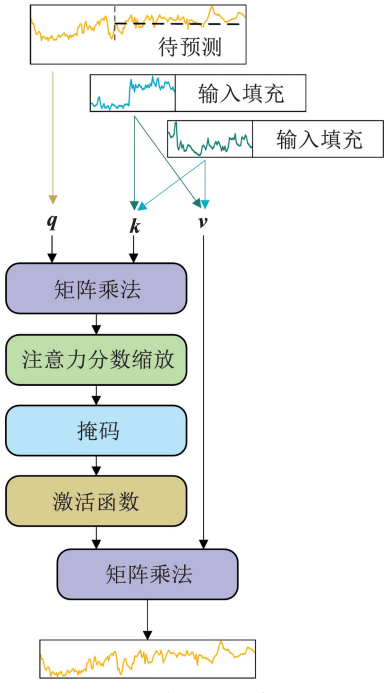


图 4 特征选择性注意力机制

Fig. 4 Feature-selective attention mechanism

得到需要的查询向量、键向量和值向量之后，仅计算待预测特征 $\boldsymbol{y}_t$ 对所有特征的注意力分数，忽略其他特征间的注意力分数，以此得到注意力分数 $\alpha_t^{D'}$ 的计算式。

$$\alpha_t^{D'} = \frac{\exp\left(\frac{\boldsymbol{q}_t \cdot \boldsymbol{k}_t^d}{\sqrt{d_k}}\right)}{\sum_{d=1}^{D}\exp\left(\frac{\boldsymbol{q}_t \cdot \boldsymbol{k}_t^d}{\sqrt{d_k}}\right)}, D' = 1,2,\cdots,D \quad (6)$$

式中 d_k 为键向量的维度。

得到注意力分数后，计算加权后的值向量之和，作为注意力层的输出。

$$\boldsymbol{A}_t = \sum_{d=1}^{D}\alpha_t^d \boldsymbol{v}_t^d \quad (7)$$

在 FSAM 中，实际只关心与待预测特征 $\boldsymbol{y}_t$ 相关的输出，其他注意力分数 $\alpha_t^d (d \neq D')$ 在计算时被显式地忽略，以符合特征选择性的原则。因此实际计算式为

$$\boldsymbol{T}_t = a_t^{D'}\boldsymbol{v}_t^{D'} \quad (8)$$

式中：$a_t^{D'}$ 为待预测特征与其他特征之间计算得到的注意力分数，$\boldsymbol{v}_t^{D'}$ 为与 $a_t^{D'}$ 维度相同的值向量。

上述介绍的是 FSAM 的单头版本，在实际代码中使用的是多头版本。假设使用了 λ 个头，则多头版本的实现过程为：

$$\boldsymbol{T}_{\text{Multi}}(i) = \boldsymbol{W}_{\text{output}} \times C(h_1,h_2,\cdots,h_\lambda) \quad (9)$$

$$h_\lambda = \boldsymbol{T}_\lambda(\boldsymbol{q}_t\boldsymbol{W}_\lambda^{\text{q}}, \boldsymbol{k}_t^d\boldsymbol{W}_\lambda^{\text{k}}, \boldsymbol{v}_t^d\boldsymbol{W}_\lambda^{\text{v}}) \quad (10)$$

式中：h_λ 为第 λ 个注意力头的输出，$\boldsymbol{W}_{\text{output}}$ 为投影权重矩阵，$\boldsymbol{W}_\lambda^{\text{q}}$、$\boldsymbol{W}_\lambda^{\text{k}}$、$\boldsymbol{W}_\lambda^{\text{v}}$ 分别为第 λ 个注意力头的查询、键和值投影矩阵。

将 $\boldsymbol{T}_t$ 通过一个投影层得到最终的预测输出

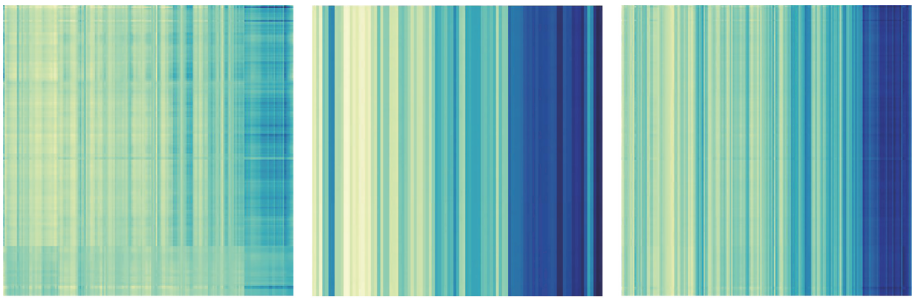
$$\hat{\boldsymbol{y}}_t = P(\boldsymbol{T}_t) \quad (11)$$

式中 P 为投影操作。

如果 $\boldsymbol{T}_t$ 的维度与 $\hat{\boldsymbol{y}}_t$ 的维度相匹配，可以直接采用 $\boldsymbol{T}_t$ 得到最终的预测输出

$$\hat{\boldsymbol{y}}_t = \boldsymbol{T}_t \quad (12)$$

图 5 为将 Transformer 经过倒置后的模型在 ETTh1 数据集上的注意力得分，分别为第 1、2、3 个注意力头各自的第 1 层（{head1、head2、head3} @ layer1）的可视化结果。



(a) headl@layerl　(b) head2@layerl　(c) head3@layerl

图 5　注意力得分可视化实例

Fig. 5　Attention score visualization example

由图 5 可以观察到，在计算注意力得分过程中，仅有少数注意力值对预测结果产生了显著影响，这些得分主要集中于待预测目标特征与其他特征之间的交互区域。该发现表明了注意力机制在实际计算中的稀疏性特征，同时验证了本文提出的特征选择性注意力机制的理论有效性，进一步确认了其在设计层面的合理性与有效性。

2　实验结果及分析

2.1　数据集

采用 6 个数据集验证模型的有效性，其中包含变压器油温数据集（ETTh1）[10]、天气预报数据集（Weather）[12]、金融汇率数据集（Exchange）[6]、疾病数据集（ILI）[19]、交通数据集（Traffic）[21] 5 个基准数据集与滑坡灾害数据集（Landslide）。

鉴于数据集中普遍存在缺失值、异常值及噪声问题，对所有数据集采取数据预处理措施以提升数据质量。具体而言，针对缺失值问题，采用插值策略进行填补，以确保数据集的完整性和连续性；对于异常值的识别与剔除，结合统计检验方法与领域专业知识，有效剔除对分析结果产生不利影响的极端值；此外，为减少数据中的随机波动与噪声干扰，应用滚动平均值技术对数据进行平滑处理；最后，通过归一化方法将所有特征变量转换至统一量纲。

本文将所有数据集按时间顺序分为训练集、验证集和测试集，其中 ETTh1 数据集中三者的比例为 6∶2∶2，其他数据集中三者的比例均为 7∶1∶2。

2.2　评价指标

采用时间序列预测任务中通用的均方误差（mean square error，E_{ms}）和平均绝对误差（mean absolute error，E_{ma}）两个指标进行性能评价。

$$E_{\text{ms}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (13)$$

$$E_{\text{ma}} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (14)$$

式中：y_i 为观测值，$\hat{y}_i$ 为预测值，n 为样本数量。

2.3 实验平台及参数设置

实验硬件环境中,操作系统为 Windows 10,CPU 为 Intel(R) Core(TM) i7-9700K,GPU 为 NVIDIA GeForce RTX 2080 Ti;软件环境为 Python 3.8.18、PyTorch 1.12.1 + cu113、CUDA 11.3。训练过程中,优化器采用 Adam,初始学习率为 0.000 1,损失函数为均方误差损失,迭代次数为 20,具有适当的早停策略,批量大小设置为 32,特征选择 Transformer 块的数量设置为 $N=2$。

2.4 对比实验

为验证模型的有效性,挑选 10 个主流的预测模型分别在不同的数据集上进行对比。具体分为 4 类:传统预测方法有 ARIMA;基于 RNN 的方法有 GRU 和 LSTNet;基于 CNN 的方法有 TCN 和 ModernTCN;基于 Transformer 的方法有 iTransformer、Informer、Crossformer、Pathformer 和 Flowformer。

在实验中,统一将 ILI 数据集的预测长度设置为 $S\in\{24,36,48,60\}$,回溯长度为 $L\in\{24,36\}$。除 ILI 和 Landslide 数据集外,其他数据集预测长度为 $S\in\{96,192,336,720\}$,回溯长度为 $L=96$,对比实验默认为多变量预测单变量。表 1 为本文模型与传统预测方法、基于 RNN 和 CNN 方法的对比结果。

表 1 本文模型与其他方法比较

Tab. 1 Comparison between the presented model and the mainstream algorithms

数据集	预测长度	Targetformer		ModernTCN		TCN		LSTNet		GRU		ARIMA	
		E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}
ETTh1	96	**0.110**	**0.249**	0.221	0.449	0.416	0.313	0.512	0.648	0.431	0.553	0.409	0.564
	192	**0.122**	**0.271**	0.248	0.499	0.533	0.470	0.546	0.872	0.534	0.613	0.491	0.624
	336	**0.120**	**0.272**	0.395	0.503	0.649	0.698	0.661	0.813	0.548	0.614	0.434	0.795
	720	**0.152**	**0.313**	0.437	0.680	0.665	0.660	0.701	0.964	0.512	0.658	0.467	0.765
Weather	96	**0.001**	**0.026**	0.004	0.041	0.055	0.287	0.067	0.389	0.310	0.423	0.073	0.080
	192	**0.001**	**0.029**	0.002	0.034	0.064	0.335	0.034	0.426	0.317	0.480	0.073	0.109
	336	**0.001**	**0.030**	0.003	0.051	0.069	0.392	0.077	0.436	0.346	0.517	0.086	0.074
	720	**0.002**	**0.036**	0.003	0.057	0.078	0.564	0.081	0.537	0.420	0.509	0.087	0.089
Exchange	96	**0.114**	**0.251**	0.311	0.464	0.942	0.913	1.089	1.194	1.216	1.312	1.246	1.672
	192	**0.217**	**0.339**	0.223	0.552	1.887	1.798	1.189	1.201	1.323	1.403	1.488	1.491
	336	**0.517**	**0.570**	0.548	0.649	1.717	1.727	1.188	1.278	1.475	1.535	1.595	1.610
	720	**0.759**	**0.694**	0.896	0.798	1.773	1.825	1.279	1.295	1.502	1.681	1.512	1.693
Traffic	96	**0.171**	**0.256**	0.328	0.380	0.773	0.857	0.681	0.829	0.348	0.554	0.553	0.981
	192	**0.150**	**0.237**	0.304	0.412	0.649	0.914	0.711	0.840	0.250	0.627	0.487	1.092
	336	**0.142**	**0.232**	0.347	0.514	0.718	0.994	0.747	0.934	0.482	0.847	0.648	1.402
	720	**0.151**	**0.243**	0.447	0.586	0.799	1.163	0.792	1.202	0.563	0.893	0.788	1.400
ILI	24	**1.031**	**0.661**	1.448	0.798	3.447	1.476	2.026	1.249	1.707	1.246	3.775	1.774
	36	**0.706**	**0.649**	1.650	0.908	3.287	1.648	2.358	1.268	1.845	1.250	3.801	1.967
	48	**0.694**	**0.637**	1.703	1.073	3.453	1.852	2.017	1.387	1.808	1.259	4.940	2.380
	60	**0.670**	**0.647**	1.888	1.198	3.560	1.999	2.441	1.420	1.906	1.331	4.895	2.408

由表 1 可知,与传统预测方法、基于 RNN 和 CNN 的方法相比,本文模型在 E_{ms} 和 E_{ma} 两个性能指标上均取得最优结果。从具体数据来看,本文模型相较于传统预测方法的 E_{ms} 和 E_{ma} 下降幅度均超过两倍,且相比 RNN 和 CNN 方法的改进同样具有优势,说明本文模型在复杂的时间序列预测任务中能够有效地捕获数据的时间依赖性和复杂非线性关系。

在此基础上,为进一步验证本文模型的有效性和相对性能优势,选取近年来基于 Transformer 的多种主流时间序列预测模型进行对比实验,这些模型代表了当前时间序列预测研究的最新进展,将其纳入对比范围,以便从更广泛的技术维度评估本文模型的实际表现。对比结果见表 2。

表 2　本文模型与基于 Transformer 模型比较

Tab. 2　Comparison of the model in this paper with other Transformer-based models

数据集	预测长度	Targetformer		iTransformer		Flowformer		Pathformer		Informer		Crossformer	
		E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}
ETTh1	96	**0.110**	**0.249**	**0.110**	0.254	0.273	0.426	0.268	0.367	0.137	0.306	0.291	0.394
	192	**0.122**	**0.271**	0.123	0.272	0.212	0.370	0.197	0.319	0.190	0.352	0.309	0.426
	336	**0.120**	**0.272**	0.122	0.273	0.185	0.347	0.205	0.317	0.208	0.353	0.315	0.595
	720	**0.152**	0.313	0.153	**0.311**	0.572	0.628	0.597	0.621	0.881	0.839	0.691	0.664
Weather	96	**0.001**	**0.026**	**0.001**	0.027	0.003	0.048	0.003	0.036	0.004	0.500	0.003	0.039
	192	**0.001**	**0.029**	0.002	**0.029**	0.002	0.032	0.004	0.042	0.002	0.036	0.012	0.089
	336	**0.001**	**0.030**	0.002	0.031	0.199	0.040	0.312	0.118	0.014	0.048	0.005	0.054
	720	**0.002**	**0.036**	**0.002**	0.037	0.015	0.030	0.020	0.036	0.185	0.102	0.007	0.053
Exchange	96	**0.114**	**0.251**	0.121	0.259	0.792	0.731	0.661	0.644	1.416	1.041	1.164	1.122
	192	**0.217**	**0.339**	0.220	0.352	1.927	1.168	0.729	0.501	1.210	0.970	1.577	1.145
	336	**0.517**	**0.570**	0.634	0.624	0.917	0.727	0.784	0.638	1.790	1.136	1.834	1.265
	720	**0.759**	**0.694**	1.199	0.849	1.297	1.026	0.812	0.799	1.107	0.860	1.312	0.993
Traffic	96	**0.171**	**0.256**	0.172	0.258	0.301	0.373	0.478	0.279	0.383	0.432	0.293	0.401
	192	**0.150**	**0.237**	**0.150**	0.238	0.381	0.422	0.362	0.492	0.619	0.586	0.387	0.412
	336	**0.142**	**0.232**	0.143	0.237	0.326	0.391	0.522	0.305	0.625	0.596	0.389	0.398
	720	**0.151**	**0.243**	0.152	0.244	0.326	0.388	0.532	0.322	0.939	0.756	0.408	0.402
ILI	24	**1.031**	**0.661**	1.040	0.669	3.114	1.543	2.103	1.339	4.268	1.801	2.402	1.423
	36	**0.706**	0.649	0.731	**0.648**	3.693	1.692	2.440	1.419	4.717	1.922	2.603	1.596
	48	0.694	**0.637**	**0.682**	0.666	3.566	1.663	2.674	1.604	4.592	1.910	2.904	1.684
	60	**0.670**	**0.647**	0.678	0.656	4.101	1.804	3.004	1.524	4.932	1.996	2.586	1.608

由表 2 可知，本文模型在不同规模的数据集以及多样化的预测长度设定下，均展现出了卓越的性能，普遍达到甚至超越了当前最优模型的基准。具体而言，在小型数据集范畴内，以 ILI 数据集为例，本文模型在不同预测时段上相较于最优模型均实现了更优的预测效果。在预测长度为 36 时，E_{ms} 相较于最优模型降低了 3.4%；当预测长度延长至 48 时，E_{ma} 减少了 4.4%。说明即便是在数据量有限的情况下，本文模型依然能够保持较高的预测精度。在中大型数据集中，以 Exchange 数据集为例，本文模型在所有考察的预测长度上均优于现有的最佳模型。具体而言，在预测长度为 96 时，E_{ms} 和 E_{ma} 分别下降了 5.8% 和 3.1%；预测长度增加至 336 时，E_{ms} 和 E_{ma} 的降幅扩大至 18.5% 和 8.7%；而当预测长度达到 720 时，E_{ms} 和 E_{ma} 分别下降了 36.7% 和 18.3%。说明本文模型不仅在短期预测中具备较强的适应性，而且在处理大规模数据集和长序列预测任务时展现出优越的泛化能力和鲁棒性。

为更全面直观地评估本文模型在时序预测任务上的表现，将 ETTh1 与 Traffic 数据集上所取得的预测结果与真实值进行可视化对比，预测值曲线与真实值曲线的贴合度越高，表明模型的预测效果越好。输入长度和预测长度为均 96 的可视化结果如图 6 和图 7 所示。

由图 6 和图 7 可以看出，本文模型的预测值与真实值贴合得更好，特别是在快速变化的区域（如突变点），对于时间序列中的快速变化点或复杂模式，预测值能够更加准确地跟随真实值变化，比其他模型更准确地预测时间序列的变化趋势。

在滑坡灾害预测数据集上，将预测长度设置为 $S \in \{48, 96, 192, 336\}$，回溯长度设置为 $L \in \{48, 96\}$，本文模型与其他模型的对比结果见表 3。

由表 3 可知，本文模型在所有评估的预测长度上均取得了最优结果。具体而言，相较于其他模型，本文模型的 E_{ms} 和 E_{ma} 表现最优。当预测长度为 48 时，本文模型的 E_{ms}、E_{ma} 相较于最优方法分别降低了 37.4%、34.4%；当预测长度为 336 时，E_{ms}、E_{ma} 的降低幅度分别达到 41.0%、45.3%。此外，与基于 Transformer 架构的模型对比，本文模型同样取得了最优结果。当预测长度为 48 时，本文模型的 E_{ms}、

E_{ma}相较于基于 Transformer 模型的最优实验基准分别下降了 3.4%、7.1%；当预测长度为 336 时，E_{ms}、E_{ma}的降低幅度分别为 2.8%、1.9%。上述结果表明，在面对复杂且高度动态的滑坡灾害预测任务时，本文模型能够有效捕捉数据中的关键信息，提供更为精确和可靠的预测结果。

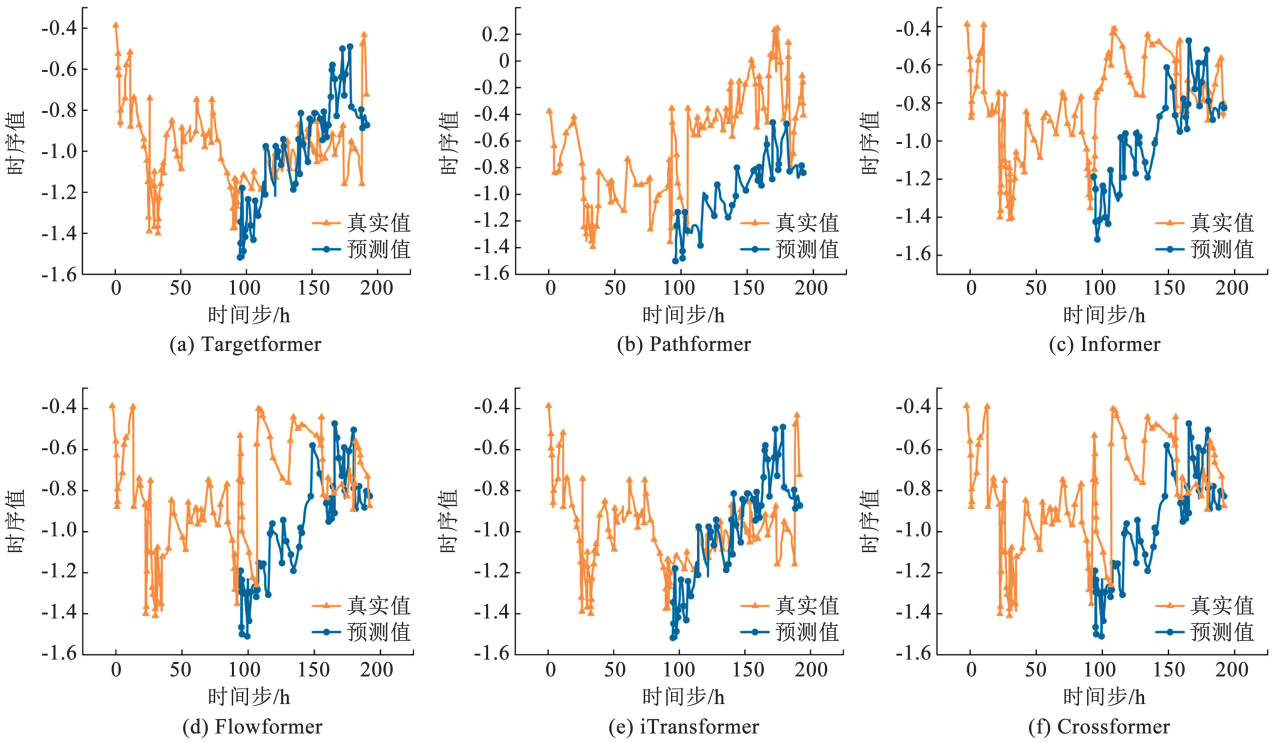


图 6 ETTh1 数据集预测可视化效果

Fig. 6 Visualization results of ETTh1 dataset prediction

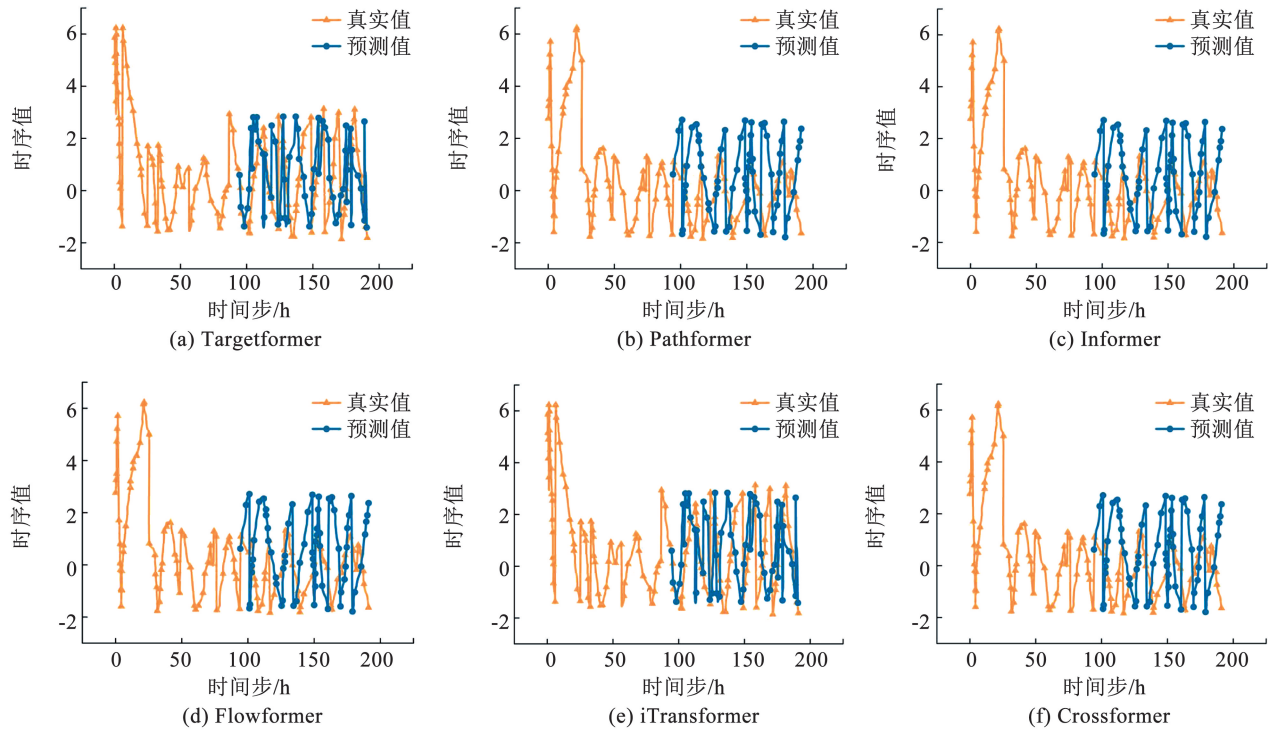


图 7 Traffic 数据集预测可视化效果

Fig. 7 Visualization results of Traffic dataset prediction

表 3　滑坡灾害预测数据集对比结果

Tab. 3　Comparison results of landslide prediction dataset

预测长度	Targetformer		ModernTCN		TCN		LSTNet		GRU	
	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}
48	**1.003**	**0.561**	1.098	0.768	1.602	0.855	1.825	0.918	1.803	0.912
96	**1.005**	**0.526**	1.189	0.814	1.755	0.876	1.847	0.937	1.814	1.105
192	**1.015**	**0.561**	1.276	0.901	1.806	0.904	1.984	1.001	1.956	1.036
336	**1.018**	**0.570**	1.313	0.978	1.792	0.990	2.020	1.065	1.877	1.124

预测长度	ARIMA		iTransformer		Flowformer		Pathformer		Informer		Crossformer	
	E_{ms}	E_{ms}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}
48	1.675	1.675	1.104	0.662	1.039	0.606	1.036	0.618	1.049	0.604	1.037	0.604
96	1.711	1.711	1.054	0.626	1.028	0.569	1.028	0.581	1.039	0.570	1.031	0.612
192	1.660	1.660	1.049	0.611	1.028	0.577	1.019	0.570	1.031	0.581	1.029	0.598
336	1.726	1.726	1.058	0.591	1.048	0.599	1.039	0.681	1.053	0.600	1.046	0.595

为进一步全面评估本文模型的性能，从计算复杂度的角度进行对比分析。为确保对比的公平性，在 3 个不同规模的数据集 ETTh1、Weather、Traffic 上选取不同的预测长度，并在模型大小和参数量相同的情况下进行实验，对比可视化效果如图 8 所示。由图 8 可以看出，与 iTransformer 相比，本文模型均取得了最低的计算量，进一步验证了本文模型的高效性。

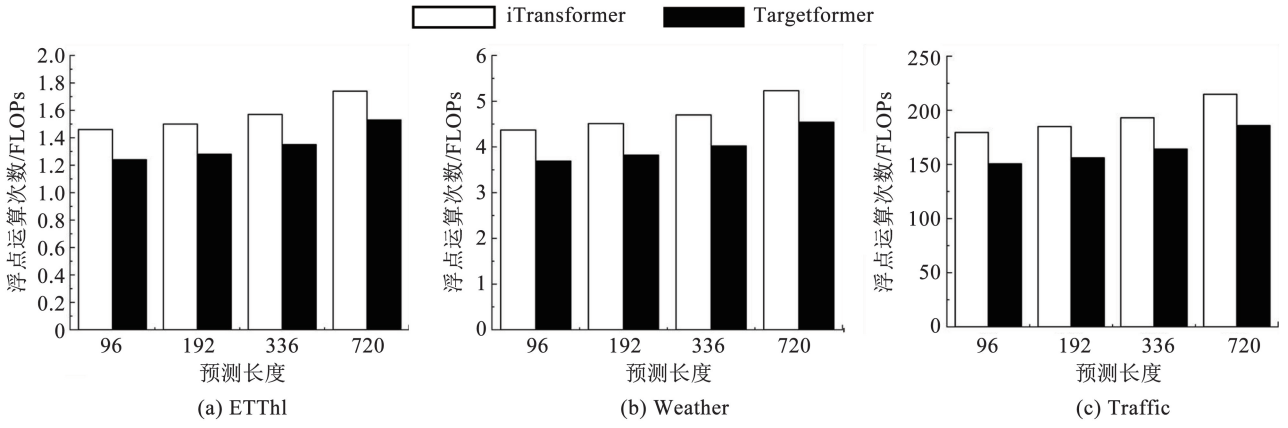


图 8　本文模型与 iTransformer 计算量对比

Fig. 8　Comparison of the computational cost of the model in this paper with iTransformer

2.5　消融实验

为了更深入地对本文模块的有效性进行分析，在 ETTh1、Weather、Landslide 数据集上，对 IE 与 FSAM 进行消融实验。实验预测长度与回溯长度均设为 96，默认设置为多变量预测单变量，结果见表 4。

表 4　消融实验结果

Tab. 4　Results of ablation experiment

数据集	模块增量	Transformer		Flowformer		Pathformer		Informer		Crossformer	
		E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}	E_{ms}	E_{ma}
ETTh1	original	0.253	0.404	0.273	0.426	0.268	0.367	0.137	0.306	0.291	0.394
	+ IE	0.113	0.254	0.242	0.457	0.121	0.266	0.124	0.268	0.128	0.278
	+ FSAM	**0.111**	**0.248**	**0.238**	**0.393**	**0.115**	**0.254**	**0.114**	**0.259**	**0.117**	**0.262**
Weather	original	0.002	0.033	0.003	0.048	0.003	0.036	0.004	0.500	0.003	0.039
	+ IE	**0.001**	0.027	**0.002**	0.038	**0.001**	0.028	**0.001**	0.300	0.003	0.036
	+ FSAM	**0.001**	**0.026**	**0.002**	**0.034**	**0.001**	**0.027**	**0.001**	**0.271**	**0.002**	**0.030**
Landslide	original	1.177	0.764	1.139	0.706	1.086	0.688	1.249	0.904	1.138	0.704
	+ IE	1.054	0.626	1.052	0.621	1.075	0.667	1.152	0.739	1.053	0.620
	+ FSAM	**1.004**	**0.525**	**1.029**	**0.588**	**1.054**	**0.626**	**1.054**	**0.625**	**1.006**	**0.528**

由表 4 可知，加入 IE 后，原有基于 Transformer 模型的效果均得到了提升。在 ETTh1 数据集上，相较于原 Transformer 模型，E_{ms}、E_{ma} 分别下降了 55.3%、37.1%。这得益于倒置输入编码块增强了模型处理复杂数据特征的能力，使得模型能捕捉到变量在整个时间范围内的动态变化，增强特征表示，从而提升

了预测精度。进一步加入 FSAM 后,各个模型的效果均得到了进一步提升。在 ETTh1 数据集上,与加入倒置输入编码块之后的模型相比,E_{ms}、E_{ma}进一步下降了 1.8%、2.4%,并且由于 FSAM 仅计算待预测目标特征与其他特征之间的注意力分数,使得模型的收敛速度显著提高,在提升模型效率的同时,也降低了计算成本,从而验证了模块在提升模型效能与效率方面的双重有效性及实用性。

3 结 论

1)本文设计一种特征选择的高效时间序列预测模型 Targetformer,采用 Transformer 仅编码器结构和倒置输入编码块与特征选择性注意力机制,提升了时间序列预测的准确性,降低了计算复杂度,增强了模型的整体性能。

2)提出特征选择性注意力机制,该机制仅关注目标特征的注意力分数,通过精细化的特征选择提升了预测精度。此机制的提出,不仅深化了对时间序列数据内在特性的理解,还揭示了时间序列数据中潜在的特征关联模式,为后续的模型优化与算法改进奠定了理论基础。

3)在 5 个基准数据集上的实验结果表明,Targetformer 的 E_{ms}和 E_{ma}较最优模型分别降低 13.8% 和 10.0%,验证了所提模型的有效性和实用性。

4)在滑坡灾害预测数据集上的实验结果表明,相较于最优模型,Targetformer 的 E_{ms}和 E_{ma}分别降低 39.2% 和 39.9%,此结果对于突发滑坡灾害的提前预警与长期规划具有重要的现实意义。

参考文献

[1]李兆玺, 刘红岩. 融合全局和序列特征的多变量时间序列预测方法[J]. 计算机学报, 2023, 46(1): 70
Li Zhaoxi, Liu Hongyan. Combining global and sequential patterns for multivariate time series forecasting[J]. Chinese Journal of Computers, 2023, 46(1): 70. DOI:10.11897/SP.J.1016.2023.00070

[2]薛自杰, 卢昱妃, 宁芊, 等. 基于时空特征融合的网络流量预测模型[J]. 哈尔滨工业大学学报, 2023, 55(5): 30
Xue Zijie, Lu Yufei, Ning Qian, et al. Network flow prediction based on spatial-temporal features fusion[J]. Journal of Harbin Institute of Technology, 2023, 55(5): 30. DOI: 10.11918/202203059

[3]Rueda F D, Suárez J D. Short-term load forecasting using encoder-decoder wavenet: application to the french grid[J]. Energies, 2021, 14(9): 2524. DOI:10.3390/en14092524

[4]Luo Donghao, Wang Xue. ModernTCN: a modern pure convolution structure for general time series analysis[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[5]Dey R, Salem F M. Gate-variants of gated recurrent unit (GRU) neural networks[C]//60th International Midwest Symposium on Circuits and Systems. Boston: IEEE, 2017: 1558. DOI: 10.1109/MWSCAS.2017.8053243

[6]Lai Guokun, Chang Weicheng, Yang Yiming, et al. Modeling long- and short-term temporal patterns with deep neural networks[C]//Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. Ann Arbor: ACM, 2018: 95. DOI:10.1145/3209978.3210006

[7]Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: probabilistic forecasting with autoregressive recurrent networks[J]. International Journal of Forecasting, 2020, 36(3): 1181. DOI: 10.1016/j.ijforecast.2019.07.001

[8]Oreshkin B N, Dudek G, Peka P, et al. N-BEATS neural network for mid-term electricity load forecasting[J]. Applied Energy, 2021, 293: 116918. DOI:10.1016/j.apenergy.2021.116918

[9]Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//31st International Conference on Neural Information Processing Systems. Long Beach: Neural Information Processing Systems Foundation, 2017: 6000

[10]Zhou Haoyi, Zhang Shanghang, Peng Jieqi, et al. Informer: beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the 35th AAAI Conference on Artificial Intelligence. Vancouver: AAAI, 2021: 11106. DOI: 10.1609/aaai.v35i12.17325

[11]Kitaev N, Kaiser L, Levskaya A. Reformer: the efficient transformer[C]//8th International Conference on Learning Representations. Addis Ababa: ICLR, 2020

[12]Wu Haixu, Xu Jiehui, Wang Jianmin, et al. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting[C]//35th Conference on Neural Information Processing Systems. Vancouver: Neural Information Processing Systems Foundation, 2021: 22419

[13]Zhou Tian, Ma Ziqing, Wen Qingsong, et al. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting[C]//Proceedings of the 39th International Conference on Machine Learning, ICML 2022. Baltimore: ML Research Press, 2022: 27268

[14]Zhang Yunhao, Yan Junchi. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting[C]//Proceedings of 11th International Conference on Learning Representations, ICLR 2023. Kigali: ICLR, 2023

[15]Nie Yuqi, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: long-term forecasting with transformers[C]//Proceedings of 11th International Conference on Learning Representations, ICLR 2023. Kigali: ICLR, 2023

[16]Liu Yong, Hu Tengge, Zhang Haoran, et al. ITransformer: inverted transformers are effective for time series forecasting[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[17]Ashok A, Marcotte É, Zantedeschi V, et al. TACTiS-2: better, faster, simpler attentional copulas for multivariate time series[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[18]Rasul K, Bennett A, Vicente P, et al. VQ-TR: vector quantized attention for time series forecasting[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[19]Dai Tao, Wu Beiliang, Liu Peiyuan, et al. Periodicity decoupling framework for long-term series forecasting[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[20]Chen Peng, Zhang Yingying, Cheng Yunyao, et al. Pathformer: multi-scale transformers with adaptive pathways for time series forecasting[C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024

[21]Li Shiyang, Jin Xiaoyong, Yao Xuan, 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: Curran Associates, 2019: 5243