

D-LINet:融合双线性层与双向归一化的时间序列预测框架

耿海军, 李东鑫

引用本文

耿海军, 李东鑫. D-LINet:融合双线性层与双向归一化的时间序列预测框架[J]. 计算机科学, 2026, 53(2): 170-179.

GENG Haijun, LI Dongxin. D-LINet: Time Series Forecasting Framework Integrating Dual-linear Layers and Dual Normalization [J]. Computer Science, 2026, 53(2): 170-179.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[融合多尺度特征和注意力机制的时间序列预测模型](#)

Time Series Forecasting Model Integrating Multi-scale Features and Attention Mechanism

计算机科学, 2026, 53(2): 180-186. <https://doi.org/10.11896/jsjcx.250100113>

[层次时间序列预测方法与应用综述](#)

Comprehensive Review of Hierarchical Time Series Forecasting Methods and Applications

计算机科学, 2025, 52(11A): 241000139-7. <https://doi.org/10.11896/jsjcx.241000139>

[基于动态图学习与注意力机制的多变量时间序列预测](#)

Multivariate Time Series Prediction Based on Dynamic Graph Learning and Attention Mechanism

计算机科学, 2025, 52(6A): 240700047-8. <https://doi.org/10.11896/jsjcx.240700047>

[基于多粒度多尺度深度时空模型的长期序列预测方法](#)

Long-term Series Forecasting Method Based on Multi-granularity Multi-scale Deep Spatio-Temporal Modeling

计算机科学, 2025, 52(2): 91-98. <https://doi.org/10.11896/jsjcx.240400127>

[基于季节分解的混合神经网络的时间序列预测](#)

Time Series Prediction of Hybrid Neural Networks Based on Seasonal Decomposition

计算机科学, 2024, 51(11A): 231200008-7. <https://doi.org/10.11896/jsjcx.231200008>

D-LINet: 融合双线性层与双向归一化的时间序列预测框架

耿海军 李东鑫

山西大学自动化与软件学院 太原 030006

摘要 时间序列预测在能源管理、交通流量和气象分析等多个实际场景中具有重要应用价值。然而,时间序列数据中存在的分布漂移(Distribution Shift)与长程依赖(Long-term Dependency)仍限制了传统方法与现有深度学习模型在长期预测中的表现。为此,提出了一种名为 D-LINet(Dual-Normalization and Linear Integration Network)的创新模型。该模型结合了 Dish-TS(Distribution Shift in Time Series Forecasting)框架的分布归一化能力与线性映射的高效性,并采用双向归一化与双线性层的设计,有效缓解输入与输出空间的分布偏移,增强了对周期性与趋势性特征的捕捉能力。在多个真实数据集上对 D-LINet 的预测性能进行了全面评估。结果显示,在短期与长期预测中,D-LINet 的均方误差和平均绝对误差均显著优于主流模型(如 Transformer, Informer, Autoformer 和 DLinear)。此外,实验还探讨了输入窗口长度及先验知识的引入对预测性能的影响,为后续模型优化提供了重要指导。该研究针对复杂分布漂移问题提出了新的解决思路,并有助于提升时间序列预测的精度与稳健性。

关键词: 时间序列预测; 分布漂移; 双向归一化; 线性映射; 周期性与趋势性建模

中图分类号 TP393

D-LINet: Time Series Forecasting Framework Integrating Dual-linear Layers and Dual Normalization

GENG Haijun and LI Dongxin

School of Automation and Software Engineering, Shanxi University, Taiyuan 030006, China

Abstract Time series forecasting plays a crucial role in various real-world applications such as energy management, traffic flow forecasting, and meteorological analysis. However, the presence of distribution shift and long-term dependency in time series data continues to limit the performance of both traditional methods and existing deep learning models in long-range forecasting. To address these challenges, this paper proposes an innovative model named D-LINet. The proposed model integrates the distribution normalization capability of the Dish-TS framework with the efficiency of linear mappings. By employing dual-direction normalization and dual-linear-layer designs, it effectively mitigates distribution shifts in both input and output spaces, while significantly enhancing the capture of periodic and trend-related features. A comprehensive evaluation of D-LINet on multiple real-world datasets demonstrates that, for both short- and long-term forecasting, D-LINet consistently achieves lower MSE and MAE compared to mainstream models such as Transformer, Informer, Autoformer and DLinear. In addition, experiments investigate the influence of input window length and the incorporation of prior knowledge on forecasting performance, providing valuable insights for subsequent model optimization. Overall, this study offers a novel solution to address complex distribution shifts, contributing to improved accuracy and robustness in time series forecasting.

Keywords Time series forecasting(TSF), Distribution shift, Dual normalization, Linear mapping, Periodicity and trend modeling

1 引言

时间序列预测(Time Series Forecasting, TSF)是数据科学中一项应用广泛的长期研究任务,覆盖金融投资、能源管理^[1]、交通流量估计^[2]、天气状况估计^[3]等多个领域。随着应用场景的复杂性和数据量的增长,传统统计模型(如 ARIMA^[4]和 SARIMA^[5])逐渐暴露出局限性,同时也出现了更复杂的混合设计模型。这些统计模型在短期预测中表现良好。

然而,在长期预测和大规模数据集上,尤其当数据呈现非平稳性和分布漂移时,它们的性能往往受限。

近年来,深度学习方法逐渐被应用于时间序列预测任务,尤其是循环神经网络(RNN)^[6-7]和卷积神经网络(CNN)^[8-9]等模型。这些模型凭借强大的时序特征学习能力,在容量和鲁棒性上超越了传统统计方法,但在长期预测和复杂分布变化方面仍面临挑战。针对这一问题,近年来基于 Transformer^[10]的方法因在捕捉时间序列长期相关性上的优势而备受

到稿日期:2025-01-22 返修日期:2025-03-29

基金项目:国家自然科学基金(62472267)

This work was supported by the National Natural Science Foundation of China(62472267).

通信作者:耿海军(genghaijun@sxu.edu.cn)

关注,其在自然语言处理^[11]、语音识别^[12]和计算机视觉^[13-14]等领域均展现出无与伦比的性能。

Transformer通过自注意力机制高效捕捉长程依赖,能够建模全局的时序关系,但其对顺序信息的敏感度有限,可能导致时间信息的丢失(Temporal Information Loss)。尽管位置编码技术(Positional Encoding)在一定程度上缓解了这一问题,但时序数据通常缺乏语义,关键在于捕捉连续数据点之间的时间变化,而非仅仅依赖固定位置的关系。为此,Informer^[15]通过优化注意力计算和内存消耗,显著提升了模型在长时间序列上的推理速度与预测精度;Autoformer^[16]则通过引入自动相关机制替代传统的注意力机制,专门针对时序数据的周期性设计,更好地捕捉周期性和趋势性变化,进一步增强了预测性能。这些经典模型理论基础扎实,实现简洁,且在长序列预测和处理复杂分布变化时表现优越,因此被广泛选作时序预测任务中的基准模型。然而,近期的LTSF-Linear系列研究^[17]表明,这些基于Transformer的方法或许并没有此前认定的那样有效,并且其预测结果主要依赖一次性预测,而非自回归生成。相反,仅使用单个线性层的LTSF-Linear在性能上出人意料地超越了现有复杂架构。

从传统的统计方法^[18]到基于深度学习的TSF模型^[19-20],这些方法已在众多领域取得卓越成果。然而,时间序列数据的非平稳性(Non-stationarity)仍未得到充分解决。其中,时间序列分布会随时间变化,这种现象可视为时间序列中的分布漂移^[21]。时间序列预测仍因分布随时间变化而受到分布漂移影响,导致模型泛化能力不足,极大地阻碍了TSF模型的性能提升。近期,Dish-TS框架提出了一种数据归一化方法^[22],通过捕捉输入空间(Lookback Window)和输出空间(Horizon Window)分布的变化,提升了模型应对分布变化的鲁棒性。它引入了Intra-space Shift和Inter-space Shift两种分布变化类型:前者表示输入空间内的数据分布随时间变化,后者反映输入与输出空间之间的分布差异。通过有效建模这两种变化,Dish-TS不仅提高了对时序变化的适应性,也显著增强了长期预测任务中的性能。在此背景下,如何有效应对分布漂移与长程依赖,已成为时间序列预测领域的核心挑战。为此,本文提出了D-LINet,它结合了Dish-TS框架的分布归一化能力与线性映射的高效性,进行长期时间序列预测。

1)缓解分布漂移:通过BACKCONET和HORICONET分别对输入和输出的分布特性进行建模,并采用归一化与反归一化的双向处理,有效提升模型应对分布漂移的鲁棒性。

2)捕捉周期性与趋势性特征:在线性映射部分,D-LINet采用双线性层(Dual Linear)设计,通过线性变换高效捕捉周期性特征。同时,结合分布归一化进一步增强对时间序列趋势特性的建模,从而兼顾简洁性与高效性。

3)强化反归一化过程:通过在HORICONET中引入“先验知识”(滑动窗口和目标序列的均值)作为辅助监督信号,显著增强模型对长时间跨度预测任务的支持能力,进一步增强模型对长时间跨度预测任务的适应能力,并优化了反归一化效果。

D-LINet在设计中融合了线性映射的高效性与双向归一

化的鲁棒性,尤其是双线性层的引入,不仅在计算上简洁高效,也为时间序列预测中的分布漂移问题带来了更稳健的解决方案。本文的主要贡献如下:

1)提出了一种基于Dish-TS策略的新方法D-LINet,通过结合输入与输出空间的分布特性,并引入双线性层结构高效捕捉时间序列的周期性与趋势特征,同时在反归一化过程中融入先验知识,提升了模型应对分布漂移的鲁棒性以及长时间跨度预测的性能。

2)在模型设计上,通过双向归一化与双线性层结构,解决了时序预测中的分布漂移问题,并显著增强了对周期性和趋势性特征的捕捉能力。该方法有效地突破了传统模型在处理复杂时序数据时的局限性,特别是在长时间跨度预测任务中展示了优越的性能。

3)通过实验进一步分析了先验知识的引入与回溯窗口长度对模型性能的影响。结果显示,先验知识能够有效促进HORICONET的学习,尤其在长时间跨度预测中能迅速降低预测误差;同时,合理增加回溯窗口长度有助于提升模型预测性能,为时间序列预测提供关键设计指导。

2 问题定义

2.1 时间序列预测问题定义

时间序列预测问题定义:时间序列由按时间顺序排列的观测值组成,形式化地表示为 $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T \in \mathbb{R}^T$,其中 \mathbf{x}_t 表示时间 t 时刻的观测值。针对多变量时间序列,记 $\mathbf{X} = \{\mathbf{x}_t^{(c)}\}_{t=1}^T \in \mathbb{R}^{T \times c}$,其中 c 表示变量的维度或通道数。时间序列预测旨在根据历史观测序列 $\mathbf{X}_{\text{past}} = \{\mathbf{x}_{t-L+1}, \dots, \mathbf{x}_t\} \in \mathbb{R}^{L \times c}$ (长度为 L 的回溯窗口)预测未来 H 个时间步长的观测值 $\mathbf{X}_{\text{future}} = \{\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+H}\} \in \mathbb{R}^{H \times c}$ 。预测问题可归结为学习一个映射函数 \mathcal{F} :

$$\mathcal{F}: \mathbb{R}^{L \times c} \rightarrow \mathbb{R}^{H \times c} \quad (1)$$

其中, \mathcal{F} 代表一个模型,该模型通过历史窗口 \mathbf{X}_{past} 捕捉时间序列的动态特性,并预测未来的值。

预测任务可分为单变量预测、多变量预测,以及短期预测与长期预测。

1)单变量预测(Univariate Forecasting)

在单变量情况下,时间序列仅包含一个变量,即 $c=1$,目标为预测该单一变量的未来值。

$$\mathbf{x}_{t+1:t+H} = \mathcal{F}(\mathbf{x}_{t-L+1:t}) \quad (2)$$

2)多变量预测(Multivariate Forecasting)

多变量预测涉及多个变量之间的复杂交互关系,其中 $c>1$,目标是同时预测所有变量的未来值。

$$\mathbf{X}_{t+1:t+H} = \mathcal{F}(\mathbf{X}_{t-L+1:t})$$

3)短期预测与长期预测(Short-term vs Long-term Forecasting)

短期预测($H \ll L$)主要关注局部模式,而长期预测($H \gg L$)则须建模更复杂的趋势性和季节性特征。

2.2 分布漂移问题

时间序列预测的一个核心挑战在于数据的分布漂移。在Dish-TS^[22]中,将时间序列预测中的分布漂移划分为以下两种主要类型。

1)输入空间内漂移(Intra-space Shift):输入空间(即回溯

窗口, Lookback Window) 内部的时间序列分布随时间发生变化。对于任意时间步 $u \neq v$, 分布漂移可以形式化为:

$$|d(\mathbf{x}_{\text{input}}^{(i)}(u), \mathbf{x}_{\text{input}}^{(i)}(v))| > \delta \quad (3)$$

其中, δ 是一个小的阈值, d 是一个距离函数(如 KL 散度), $\mathbf{x}_{\text{input}}^{(i)}$ 表示回溯窗口的分布。

2) 输入空间与输出空间之间的漂移(Inter-space Shift): 输入空间(Lookbacks)与输出空间(Horizons)之间的分布存在差异。其形式化定义如下:

$$|d(\mathbf{x}_{\text{input}}^{(i)}(u), \mathbf{x}_{\text{output}}^{(i)}(u))| > \delta \quad (4)$$

其中, $\mathbf{x}_{\text{output}}^{(i)}$ 表示预测窗口的分布。

综上, 本文主要针对提升时序预测精度和缓解分布漂移问题进行探讨, 后续研究在下文展开。

3 D-LINet 框架

在时间序列预测中, 分布漂移是一个关键问题, 可以分为两种形式: 输入序列的分布漂移, 输入与输出之间的分布

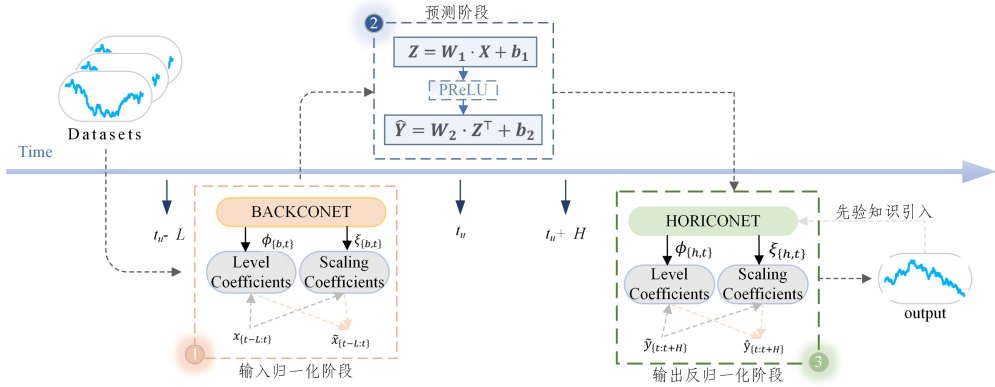


图1 D-LINet 框架图

Fig. 1 Framework diagram of D-LINet

3.1 输入归一化阶段

在时间序列预测中, 对输入序列进行归一化以消除分布漂移的影响, 并将其映射到稳定分布的特征空间, 这对后续模型的学习至关重要。然而, 时间序列往往存在非平稳性问题, 例如日均电力消耗会随月份波动, 若直接使用原始序列, 可能导致模型学习效率低下。此外, 传统归一化方法多仅针对全局均值和方差, 难以同时捕捉序列的局部特征(如短期波动), 从而限制了其实际应用表现。而 BACKCONET 的设计可以解决这一问题。

BACKCONET 是一个轻量级神经网络, 接受输入序列 $\mathbf{x}_{(t-L:t)}$, 生成两个关键的归一化系数。

1) 全局水平系数(Level Coefficients) $\phi_{(b,t)}$: 表示序列的整体均值, 反映全局趋势。

2) 波动幅度系数(Scaling Coefficients) $\xi_{(b,t)}$: 表示序列的标准差, 捕捉短期波动。

这些系数由 BACKCONET 模型生成。使用 $\phi_{(b,t)}$ 和 $\xi_{(b,t)}$ 对输入序列进行归一化:

$$\tilde{\mathbf{x}}_{(t-L:t)} = \frac{\mathbf{x}_{(t-L:t)} - \phi_{(b,t)}}{\xi_{(b,t)}} \quad (5)$$

输出:

$$\phi_{(b,t)}, \xi_{(b,t)} = \text{BACKCONET}(\mathbf{x}_{(t-L:t)}) \quad (6)$$

借助全局水平与波动幅度这两项关键归一化系数进行输

偏差。前者受外部环境或季节性变化影响, 时间序列的分布特性(如均值和方差)在不同时间段发生变化, 例如电力消耗在不同月份的峰值存在差异; 后者指预测目标序列与输入序列之间可能存在显著的分布差异, 例如历史数据主要表现出趋势性, 而目标数据则包含突发性的变化。这些分布偏移在长时间跨度预测任务(如预测跨度为 336 或 720 时)中尤为明显。现有的许多时间序列预测模型未能有效应对这种分布漂移问题。一些模型默认假设输入与目标序列分布一致, 或未对分布特性进行充分建模, 从而在分布漂移的场景下适应性不足。

为应对上述挑战, D-LINet 引入了一个包含 4 个关键阶段的流程: 输入归一化, 预测, 输出反归一化, 以及辅助监督。D-LINet 的整体框架如图 1 所示。这种设计结合了 Dish-TS 框架的分布归一化能力和线性映射模型的简洁有效性, 通过双向归一化(Dual Normalization)来缓解分布漂移问题, 提高了模型的预测性能。

入归一化后, 序列的均值被调整为 0, 标准差被调整为 1, 从而使其分布更加稳定, 有利于后续预测模型的学习。其次, D-LINet 模型在设计上足够灵活, 不仅能应对全局线性分布漂移, 也能有效捕捉时间序列的局部动态特征, 从而提升其在复杂环境下的预测能力。

3.2 预测阶段

本阶段主要利用归一化后的输入序列, 通过高效的线性预测模型来捕捉时间序列中的周期性和趋势性特征, 并输出归一化预测结果。现有的预测方法仍面临若干挑战。首先, 复杂的时间序列预测模型(如 Transformer 或 CNN)通常需要精细的参数调整, 且预测效果不一定优于简单的线性模型^[17], 从而削弱了其在实际应用中的优势。其次, 对于强周期性时间序列(如电力消耗或天气数据), 复杂模型在捕捉周期特征方面往往不如简单线性模型, 后者更能有效学习周期模式。因此, 在处理周期性与趋势性特征时, 采用高效的线性预测模型可能更为合适。

为应对时间序列预测中分布漂移与长期依赖建模的挑战, 本文提出了一种基于双线性层的新方法。通过两次线性变换, 双线性层可将输入空间映射到特征空间, 再从特征空间映射到输出空间, 从而高效捕捉时间序列中的周期性与趋势性特征。具体而言, 双线性层的计算过程如下。

1) 第一层: 输入到特征空间

$$\mathbf{Z} = \text{PreLU}(\mathbf{W}_1 \cdot \mathbf{X} + \mathbf{b}_1) \quad (7)$$

其中, $\mathbf{X} \in \mathbb{R}^{L \times d_{in}}$ 为输入序列, $\mathbf{W}_1 \in \mathbb{R}^{d_{model} \times d_{in}}$, $\mathbf{Z} \in \mathbb{R}^{L \times d_{model}}$ 为特征表示。

2) 第二层: 特征到输出空间

$$\hat{\mathbf{Y}} = \mathbf{W}_2 \cdot \mathbf{Z}^T + \mathbf{b}_2 \quad (8)$$

其中, $\mathbf{W}_2 \in \mathbb{R}^{H \times d_{model}}$, $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times d_{out}}$ 为最终输出。

3) 完整公式

双线性层可形式化为:

$$\hat{\mathbf{Y}} = \mathbf{W}_2 \cdot \text{PreLU}(\mathbf{W}_1 \cdot \mathbf{X} + \mathbf{b}_1)^T + \mathbf{b}_2 \quad (9)$$

在该模型中, \mathbf{W} 为权重矩阵, 用于捕捉输入和输出序列之间的线性关系; \mathbf{b} 为偏置项, 用于调整输出值。其主要优势体现在以下几个方面。首先, 在周期性与趋势性建模中, 第一层映射捕捉周期特征, 第二层映射生成全局趋势。其次, 模型具有较强的鲁棒性, 借助特征空间映射提升了对分布漂移的适应能力。再者, 与复杂的深度学习框架相比, 双线性层在计算效率方面更具优势。具体而言, Linear 的线性性质使其能够有效捕捉时间序列中的周期性特征。例如, 电力负荷的日周期或周周期可通过线性映射直接学习。对于长时间跨度预测, Linear 也能在较少参数的条件下生成准确结果, 从而降低复杂模型可能出现的过拟合风险。在预测阶段, 模型输出归一化后的预测结果 $\tilde{\mathbf{y}}_{(t,t+H)}$, 其具有稳定的分布特性, 便于后续的反归一化处理。本文发现 Linear 对数据质量和分析结果存在潜在的负面影响因素, 因此采取了一系列数据预处理措施以提高数据的准确性。

3.3 输出反归一化阶段

将归一化的预测结果 $\tilde{\mathbf{y}}_{(t,t+H)}$ 转换回目标序列的原始分布空间, 确保预测结果能够准确反映实际的时间序列特征。然而, 该过程存在两个主要问题: 首先, 分布漂移的反作用, 如果预测结果没有恢复到目标序列的真实分布, 可能导致最终预测值存在偏差; 其次, 输出分布的推断难度较大, 特别是在长时间跨度预测中, 输出序列的分布特性通常需要从输入序列中推断。HORICONET 模块可以较好的解决这一问题。

HORICONET 是一个神经网络模块, 接受输入序列 $\mathbf{x}_{(t-L,t)}$, 预测输出序列的分布系数:

$$\boldsymbol{\phi}_{(h,t)}, \boldsymbol{\xi}_{(h,t)} = \text{HORICONET}(\mathbf{x}_{(t-L,t)}) \quad (10)$$

其中, $\boldsymbol{\phi}_{(h,t)}$ 为预测序列的全局水平系数, $\boldsymbol{\xi}_{(h,t)}$ 为预测序列的波动幅度系数。

利用 $\boldsymbol{\phi}_{(h,t)}$ 和 $\boldsymbol{\xi}_{(h,t)}$ 这两个系数, 可以通过反归一化公式将归一化的预测结果恢复到目标序列的原始分布空间:

$$\hat{\mathbf{y}}_{(t,t+H)} = \boldsymbol{\xi}_{(h,t)} * \tilde{\mathbf{y}}_{(t,t+H)} + \boldsymbol{\phi}_{(h,t)} \quad (11)$$

反归一化后, 预测结果恢复到目标序列的原始分布空间, 从而确保最终输出值与实际时间序列的一致性。

3.4 辅助监督阶段(先验知识引入)

在 Dish-TS 的设计中, HORICONET 的目标是估计输出序列的分布系数 ($\boldsymbol{\phi}_h, \boldsymbol{\xi}_h$), 而输出序列的分布通常是未知的。为了让 HORICONET 在长时间跨度的预测中表现得更稳定, 可以将滑动窗口的均值和目标序列的均值结合在一起, 形成一个更鲁棒的加权目标, 用于指导 HORICONET 学习分布系数。

3.4.1 滑动窗口与目标序列的均值组合

目标序列的均值 μ_{future} 虽然在预测阶段是未知的, 但在训练阶段是可用的, 因此可以将它作为监督信号。滑动窗口的均值 μ_{slide} 可以在预测阶段获得, 且可被视为目标序列均值的近似估计。虽然滑动窗口的均值和目标序列的均值不同, 但在某些情况(如趋势平稳的时间序列中)下这两者之间有很强的相关性。组合这两种均值的加权目标, 可以在训练阶段使用未来的监督信息, 在预测阶段使用滑动窗口的估计, 从而增强 HORICONET 的鲁棒性和泛化能力。

在训练阶段, 将滑动窗口的均值 μ_{slide} 和目标序列的均值 μ_{future} 组合成一个加权目标来约束 HORICONET 输出的水平系数 $\boldsymbol{\phi}_h$ 。在预测阶段, 使用滑动窗口的均值作为对未来均值的估计。以此提供一个稳定的参考, 帮助 HORICONET 生成更为准确的分布系数, 增强预测结果的稳定性。

滑动窗口的均值(历史特征) μ_{slide} :

$$\mu_{\text{slide}} = \frac{1}{L} \sum_{i=1}^L \mathbf{x}_{t-L+i} \quad (12)$$

其中, L 表示滑动窗口的长度, \mathbf{x}_{t-L+i} 表示输入序列 $\mathbf{x}_{t-L:t}$ 中的第 i 个数据点。

目标序列的均值(未来特征) μ_{future} :

$$\mu_{\text{future}} = \frac{1}{H} \sum_{j=1}^H \mathbf{x}_{t+j} \quad (13)$$

其中, H 表示目标序列的长度, \mathbf{x}_{t+j} 表示目标序列 $\mathbf{x}_{t+1:t+H}$ 中的第 j 个数据点。

输入序列 $\mathbf{x}_{t-L:t}$ 的长度为 L , 目标序列 $\mathbf{x}_{t+1:t+H}$ 的长度为 H 。滑动窗口 $\mathbf{x}_{t-L:t}$ 中的一个子序列, 通常是包含一小段连续时间点的窗口, 例如窗口长度为 24。

分布系数: HORICONET 生成的系数包括水平系数 $\boldsymbol{\phi}_h$ 和波动系数 $\boldsymbol{\xi}_h$ 。

$$\boldsymbol{\phi}_h, \boldsymbol{\xi}_h = \text{HORICONET}(\mathbf{x}_{t-L:t}) \quad (14)$$

3.4.2 先验损失函数

为了优化 HORICONET 生成的分布系数 $\boldsymbol{\phi}_h$, 需要设计一个多目标损失函数。这部分的关键在于, 如何设计滑动窗口的均值和目标序列的均值的组合。其目标是让 HORICONET 预测的水平系数 $\boldsymbol{\phi}_h$ 同时接近目标序列的均值 μ_{future} 和滑动窗口的均值 μ_{slide} 。

先验损失函数的定义如下:

$$L_{\text{prior}} = \alpha_1 (\mu_{\text{future}} - \boldsymbol{\phi}_h)^2 + \alpha_2 (\mu_{\text{slide}} - \boldsymbol{\phi}_h)^2 \quad (15)$$

其中, μ_{future} 是目标序列的均值(训练阶段已知, 预测阶段未知); μ_{slide} 是滑动窗口的均值(训练和预测阶段都可用); $\boldsymbol{\phi}_h$ 是 HORICONET 预测的目标分布的水平系数; α_1 是控制目标序列的均值项的权重, α_2 是控制滑动窗口的均值项的权重, 可通过实验调整(如 $\alpha_1 = 0.5, \alpha_2 = 0.5$)。

在 D-LINet 中, 除了 HORICONET 的先验损失 L_{prior} , 还需要考虑整个 D-LINet 的预测损失(MSE), 因此最终的总损失函数为:

$$L_{\text{total}} = L_{\text{mse}} + \beta L_{\text{prior}} \quad (16)$$

其中, L_{mse} 是模型的常规预测误差(衡量模型预测的目标序列与真实目标序列之间的误差); L_{prior} 是对 HORICONET 的先验监督损失函数, 用于约束 HORICONET 生成的分布系数; β 是超参数, 控制先验损失在总损失中的权重。

3.4.3 训练和预测阶段的操作

训练阶段:使用滑动窗口和目标序列的真实值,计算 μ_{slide} 和 μ_{future} 。通过 HORICONET 预测出分布水平系数 ϕ_h ,使用损失函数 L_{total} 进行反向传播,更新 HORICONET 的参数。

预测阶段:在预测阶段,目标序列的均值 μ_{future} 是未知的,无法用作监督信号。使用滑动窗口的均值 μ_{slide} 作为对 ϕ_h 的估计。HORICONET 根据滑动窗口的输入,生成预测的分布系数 ϕ_h 和 ξ_h 。这里的 ϕ_h 由 HORICONET 预测,而 μ_{slide} 是对 ϕ_h 的一个近似估计。

关键超参数的选择: α_1 和 α_2 后续设置为 0.25:0.75,0.5:0.5 或 0.75:0.25; β 用于控制先验损失 L_{prior} 的影响大小,后续实验设置为 0,0.3 或 0.6,具体的权重值需要在实验中通过调节得到。算法 1 给出了 D-LINet 的训练测试伪代码。首先,通过 BACKCONET 对输入数据进行归一化,以缓解输入空间的分布漂移。其次,数据被分批处理,并通过模型计算损失(包括均方误差和先验损失)(第 1-18 行)。在每个 epoch 结束时,使用验证集进行评估,并通过早停机机制避免过拟合(第 19-24 行)。训练完成后,模型在测试集上进行推理,并使用 HORICONET 对预测结果进行反归一化,最终计算评估指标(如 MAE,MSE)(第 25-41 行)。

算法 1 D-LINet 训练测试伪代码

Input: Training dataset $\{\mathbf{X}_{train}, \mathbf{Y}_{train}\}$, D-LINet model, optimizer (e.g., Adam), hyperparameters (learning rate lr, batch size, number of epochs, etc.)

1. Input Normalization;
2. Normalize input sequence using BACKCONET to alleviate intraspace shift
3. for epoch=1 to num_epochs do
4. Shuffle dataset $\{\mathbf{X}_{train}, \mathbf{Y}_{train}\}$ and split into batches
5. for batch=1 to num_batches do
6. Get current batch data $\mathbf{X}_{batch}, \mathbf{Y}_{batch}$
7. Forward pass;
8. Normalize input using BACKCONET: $\mathbf{X}_{batch} \leftarrow \text{BACKCONET}(\mathbf{X}_{batch})$
9. Use normalized input in D-LINet model: $\mathbf{Y}_{pred} = \text{DLINet}(\mathbf{X}_{batch})$
10. Calculate loss;
11. MSE loss: $L_{MSE} = \text{Loss}(\mathbf{Y}_{pred}, \mathbf{Y}_{batch})$
12. Prior loss: $L_{prior} = \text{Priorloss}(\mathbf{Y}_{batch})$
13. Total loss: $L_{total} = L_{MSE} + \beta \times L_{prior}$
14. Backward pass;
15. Compute gradients of the loss function with respect to model parameters
16. Update model parameters using optimizer (e.g., Adam)
17. Record the loss for each batch
18. Output current epoch loss: Epoch_epoch, L_{total} , L_{MSE} , L_{prior}
19. Validation;
20. Perform forward pass and loss computation on validation set
21. Early Stopping;
22. Check if early stopping condition is met based on validation loss
23. if Early stopping condition met then
24. Stop training and return best model
25. Prediction;
26. After training, normalize input using BACKCONET
27. Make forecast: $\mathbf{Y}_{pred} = \text{DLINet}(\mathbf{X}_{test})$

28. Output Denormalization;

29. Apply HORICONET to denormalize predictions to the original scale

30. $\mathbf{Y}_{denorm} = \text{HORICONET}(\mathbf{Y}_{pred})$

31. Testing Phase;

32. Load the best model obtained from early stopping

33. Set model to evaluation mode

34. Initialize lists to store predictions and ground truth: preds, trues

35. for batch in test_loader do

36. Get batch: batch_x, batch_y, dec_inp=get_init_batch(batch)

37. Perform forward pass to get forecast: forecast=model(batch_x, batch_y)

38. Convert forecast and true values to numpy arrays for evaluation

39. Store predictions and true values in lists: preds, trues

40. Calculate metrics: MAE, MSE, RMSE, using preds and trues

41. Output evaluation metrics: MAE, MSE, RMSE

3.4.4 D-LINet 框架的优势

1)多源信息的融合:滑动窗口的均值与目标序列的均值分别来自历史数据和未来数据,通过将二者结合,可更全面地捕捉时间序列特征,从而为 HORICONET 提供更准确的监督信号。

2)鲁棒的分布系数估计:即使测试阶段无法获取未来的真实信息,滑动窗口的均值仍可作为分布系数的稳定近似,保证模型在实际预测中的稳健性和鲁棒性,使其在长时间跨度预测任务中表现更佳。

3)灵活的损失函数:损失函数中的权重 α_1 和 α_2 可根据不同的数据分布特性进行动态调整,增强了模型在不同数据场景下的适应能力。这种灵活性,使得模型能够在各种不同的预测任务中表现出更强的泛化能力。

4 实验及分析

本文选取以下 7 个公开的真实世界数据集用于时序预测任务的实验验证。数据集涵盖了多个领域,包括电力消耗、电力变压器温度、气象指标特征和交通流量数据,具有多样性和挑战性。

1)Electricity 数据集

Electricity 数据集^[23]包含了 321 个客户的电力消耗数据(以千瓦时为单位)。该数据集的时间粒度为每小时,总计包含 26304 个时间步长(Time Steps)。其在能源消耗预测中被广泛应用,已成为时序预测任务的重要基准。

2)ETT 数据集

ETT(Electricity Transformer Temperature)数据集^[24]记录了中国两个地区在两年内采集的变压器温度和相关电力负载数据,其通过捕捉电力设备的运行状态,为短期电力负载预测和工业过程优化提供了理想的实验数据。实验中使用了以下两种时间粒度版本。

(1)ETTh1/ETTh2:时间粒度为每小时,总计包含 17420 个时间步长。

(2)ETTh1/ETTh2:时间粒度为每 15 分钟,总计包含 69680 个时间步长。

3)Weather 数据集

Weather 数据集^[25]记录了 2020 年期间美国近 1600 个地点的以每 10 分钟为粒度的气象数据,总计包含 52696 个时间

步长。该数据集包括 21 个气象指标,如空气湿度、气温等。由于具有多变量的特性,Weather 数据集为复杂的多维时间序列预测任务提供了丰富的数据支持。

4)Traffic 数据集

Traffic 数据集包含了来自美国加州交通部的 862 个传感器在 2016 年至 2018 年间每小时的数据记录,其中详细记录了旧金山湾区高速公路上各传感器所测量的道路占用率。道路占用率的数值范围为 0~1,0 表示道路完全空闲,而 1 表示道路完全拥堵,这个数值范围反映了道路的实际使用情况。在数据划分方面,本文遵循以下策略。

1)对于 ETTh1/h2 和 ETTm1/m2 数据集,按照 6:2:2 (训练集:验证集:测试集)进行基于月份的划分。

2)对于 Electricity,Weather 和 Traffic 数据集,按照比例 7:1:2(训练集:验证集:测试集)进行划分。

上述数据集的多样性和高分辨率时间粒度,能够有效验证所提方法在不同行业应用场景中的性能与鲁棒性。为评估

时间序列预测的性能,本文采用均方误差(MSE)和平均绝对误差(MAE)^[10,15-16]作为主要的评估指标,来量化预测值与真实值之间的偏差大小。所有实验均基于原始数据进行,未进行额外的数据归一化或缩放处理。为确保不同数据集之间的评估指标具有一致的可比性,避免量纲差异带来的偏差,报告的评估指标经过适当的缩放处理,以提升结果的可读性。具体地,在模型对比任务中,电力数据集(Electricity)的 MSE 和 MAE 分别缩放 1×10^{-7} 和 1×10^{-3} ,天气数据集(Weather)缩放 1×10^{-4} 和 1×10^{-2} ,ETT 数据集(包括多个电力变压器温度数据集)缩放 1×10^{-1} 和 1×10^{-1} ,交通流量数据集(Traffic)缩放 1×10^3 和 1×10^2 。在 Lookback Window 逐步增加的实验中,电力数据集的 MSE 和 MAE 分别缩放 1×10^{-7} 和 1×10^{-2} ,天气数据集缩放 1×10^{-3} 和 1×10^{-1} ,ETT 数据集缩放 1×10^{-1} 和 1×10^{-1} ,交通流量数据集(Traffic)缩放 1×10^3 和 1×10^2 。

详细设置如表 1 所列。

表 1 实验指标缩放系数
Table 1 Scaling coefficient of experimental indicator

Dataset	Electricity		ETTh1		ETTm1		ETTh2		ETTm2		Weather		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Baseline	1×10^{-7}	1×10^{-3}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-4}	1×10^{-2}	1×10^3	1×10^2
Lookback-window	1×10^{-7}	1×10^{-2}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-3}	1×10^{-1}	1×10^3	1×10^2

训练过程采用 L2 损失函数,并使用 Adam 优化器^[26-27],学习率范围设置为 $[1 \times 10^{-4}, 1 \times 10^{-3}]$,每个实验重复 3 次,并报告平均性能以提高结果的稳健性。Lookback 窗口与 Horizon 窗口的长度一致,从 36 逐步延长至 420,以评估不同时间跨度的预测性能。模型训练过程中使用早停策略,设定为 20 个 epoch,以防止过拟合并提高训练效率。本文使用 Dish-TS 方法,结合 4 个高效的竞争基线作为对比实验,包括 Informer,Autoformer,Transformer 和 DLinear,并在相同的配置、超参数与训练策略下进行实验,以确保结果的公平性和可比性。实验基于 PyTorch 框架^[28]实现,并在 NVIDIA V100 32GB GPU 上运行。

4.1 实际数据集上的对比性能分析

在实验结果中(见表 2,其中最佳结果用加粗表示,次优结果用下划线标注),D-LINet 模型在多个数据集(如 ETTh1,ETTm1,Weather)及不同预测窗口(36,96,192,336,420)下均表现出显著优势,尤其在短时间跨度(如 96 和 192 步)预测中,MSE 和 MAE 均优于其他主流模型。例如,在 ETTm1 数据集上,D-LINet 在 96 步预测时的 MSE 和 MAE 分别为 0.790 和 0.143,而 Autoformer 分别为 1.014 和 0.176,Informer 为 1.028 和 0.173,Transformer 为 1.067 和 0.179,Dlinear 为 0.808 和 0.147,这充分体现了 D-LINet 在捕捉短期趋势与波动特征方面的优势。双向归一化通过分别对输入和输出空间的分布进行建模与归一化,有效保持了数据分布的一致性,减少了分布变化带来的影响,从而提高了模型在短期预测中的精度。

此外,在长时间跨度预测(如 336 和 420 步)中,D-LINet 的误差增长更为平缓。例如,在 Weather 数据集上,D-LINet

在 336 步预测中的 MSE 和 MAE 分别为 0.181 和 0.147,而 Autoformer 则高达 0.388 和 0.221,凸显了 D-LINet 在应对长时间依赖和分布漂移方面的稳定性与鲁棒性。双线性层设计通过线性变换高效地捕捉了周期性和趋势性特征,增强了模型对时间序列长期趋势变化的适应性,减少了长期预测中的误差波动。

相比之下,其他模型尽管在部分任务中表现良好,但在长时间跨度预测中误差波动更大,特别是在多变量复杂数据集(如 Weather 和 Electricity)上暴露出对时间序列特征适应能力的不足。例如,在 Electricity 数据集上,D-LINet 在 96 步预测时的 MSE 为 0.481,但在 336 步预测时,其 MSE 上升至 0.966,虽仍低于 Informer 的 0.684 和 Informer 的 1.313,但表明仍须进一步改进在特定场景下的长时间依赖建模。总体而言,D-LINet 在多个数据集和预测任务下展现了出色且稳定的 MSE 与 MAE 表现,尤其是在多变量时间序列的短期与长期预测场景中均具备较高的可靠性与适用性。双向归一化和双线性层的设计不仅提升了模型在应对分布漂移和周期性特征捕捉方面的能力,也为时间序列预测任务的模型设计与优化提供了新的基准与研究方向。具体实验结果如表 2 所列。表 3 列出了 7 个实际数据集的统计信息。

4.2 回视步长对时间序列预测性能的影响分析

为探究输入回看窗口大小对 D-LINet 的影响,选取 $L \in \{36, 48, 72, 96, 144, 168, 192, 336, 420, 720\}$ 进行预测实验($T=720$)。图 2 展示了各数据集上的 MSE 和 MAE 结果,整体趋势与既有研究相似。实验表明,回看步长对时间序列预测性能影响显著,不同步长会导致预测精度出现明显差异。随着回看步长的增大,大多数数据集(如 Electricity,Weather)的

MSE 和 MAE 明显下降,尤其在步长从 36 增至 96 时,性能提升最为明显。例如,在 Weather 数据集中,MSE 从 2.5 迅速降至 1.25,表明更多历史信息有助于捕捉数据中的周期性与趋势

势性特征。然而,在 ETTh2 等数据集中,当步长超过 336 时,可能引入过多历史噪声,导致预测性能的边际效应减弱甚至反弹,说明通过增加回视步长所实现的性能提升是有上限的。

表 2 时间序列预测结果
Table 2 Time series forecasting results

Method	Metric	D-LINet		Autoformer		Informer		Transformer		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	36	0.891	0.154	1.259	0.189	1.148	0.185	1.017	0.173	0.914	0.156
	96	0.967	0.172	1.386	0.206	1.353	0.209	1.421	0.217	1.019	0.174
	192	1.077	0.187	1.883	0.255	1.435	0.219	1.799	0.252	1.565	0.284
	336	1.175	0.209	2.129	0.275	1.481	0.224	2.38	0.271	1.496	0.211
	420	1.186	0.214	2.506	0.299	2.592	0.299	2.621	0.288	1.189	0.296
ETTm1	36	1.163	0.154	1.379	0.195	1.168	0.171	1.377	0.188	1.504	0.195
	96	0.790	0.143	1.014	0.176	1.028	0.173	1.067	0.179	0.808	0.147
	192	0.973	0.163	1.657	0.228	1.235	0.191	1.257	0.189	1.012	0.163
	336	1.042	0.178	1.468	0.216	1.545	0.215	1.371	0.201	1.094	0.176
	420	1.244	0.205	2.519	0.277	1.669	0.223	1.373	0.218	1.146	0.225
ETTh2	36	1.138	0.212	1.297	0.234	1.445	0.246	1.466	0.247	1.351	0.236
	96	1.486	0.249	1.813	0.279	2.193	0.305	2.078	0.296	1.510	0.253
	192	1.786	0.284	1.774	0.282	2.112	0.312	2.061	0.311	1.827	0.287
	336	1.824	0.293	2.192	0.318	2.261	0.336	2.172	0.315	1.952	0.319
	420	1.703	0.285	3.353	0.403	2.394	0.343	2.447	0.343	2.446	0.354
ETTh2	36	0.928	0.193	0.956	0.199	0.887	0.186	0.831	0.183	1.083	0.211
	96	0.995	0.198	1.144	0.219	1.153	0.215	1.27	0.233	1.039	0.215
	192	1.317	0.226	2.243	0.306	1.528	0.253	1.351	0.232	1.322	0.233
	336	1.579	0.249	2.046	0.302	1.817	0.282	1.833	0.278	1.541	0.256
	420	2.161	0.294	2.956	0.355	2.076	0.304	2.023	0.295	1.761	0.296
Weather	36	0.409	0.154	0.443	0.187	0.365	0.148	0.433	0.154	0.47	0.318
	96	0.253	0.146	0.474	0.252	0.247	0.155	0.328	0.183	0.286	0.176
	192	0.187	0.143	0.389	0.226	0.192	0.133	0.271	0.181	0.22	0.169
	336	0.181	0.147	0.388	0.221	0.217	0.155	0.203	0.151	0.206	0.164
	420	0.189	0.154	0.381	0.223	0.264	0.178	0.213	0.171	0.203	0.167
Electricity	36	0.315	0.216	0.532	0.272	0.622	0.297	7.285	0.954	0.338	0.225
	96	0.481	0.231	0.684	0.294	1.313	0.386	7.554	0.971	0.527	0.260
	192	0.659	0.259	3.291	0.857	1.117	0.356	7.795	0.979	0.717	0.285
	336	0.966	0.309	7.785	1.181	3.103	0.499	7.828	0.981	0.895	0.329
	420	0.912	0.325	7.413	0.977	1.441	0.404	8.763	0.973	0.955	0.338
Traffic	36	1.24	1.758	0.91	1.671	1.04	1.671	2.77	3.424	1.40	1.908
	96	0.95	1.492	1.11	1.79	1.24	1.829	1.22	1.739	1.27	1.769
	192	0.84	1.355	2.33	3.267	1.4	1.988	1.15	1.681	0.92	1.415
	336	0.83	1.36	1.03	1.691	1.2	1.858	2.9	3.513	0.88	1.377
	420	0.87	1.343	1.09	1.714	2.78	3.387	1.15	1.712	0.82	1.396

表 3 时间序列预测中所有数据集的统计信息

Table 3 Statistical information of all datasets for time series forecasting

Dataset	ETTh1/h2	ETTm1/m2	Weather	Electricity	Traffic
# Channel	7	7	21	321	862
Timesteps	17420	69680	52696	26304	17544
Granularity	1 hour	15 minutes	10 minutes	1 hour	1 hour
Data Partition	6:2:2(month)			7:2:1	

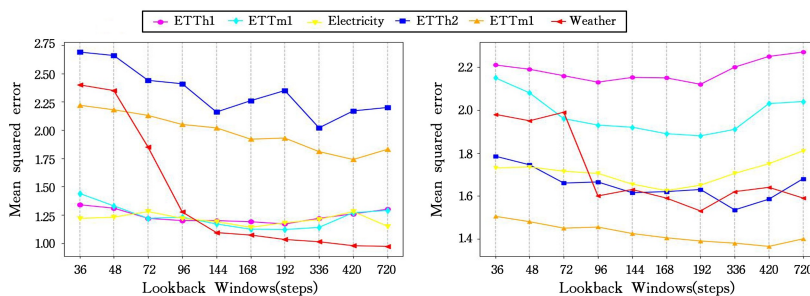


图 2 固定预测窗口 $T=720$ 下不同回视步长对多数据集预测性能的结果(MSE 与 MAE 分析)

Fig. 2 Results of the impact of different lookback window sizes on multi-dataset forecasting performance with a fixed prediction horizon of $T=720$ (MSE and MAE analysis)

进一步分析显示,周期与趋势较明显的数据集(如 Weather, Electricity)更依赖较长回看步长,而短期动态特征较多的数据集(如 ETTh2)则在中等回看步长(144/336步)下表现最佳。这表明,自适应选择回看步长对于提升预测性能至关重要。实际应用中,应结合不同数据集特性选取合适的回看窗口长度,在捕获足够历史信息与避免噪声干扰之间取得平衡。未来可探索动态调整回看步长的策略,以更好地应对时间序列的复杂性与分布变化,从而提升模型的泛化性能。

4.3 约束参数和漂移优化对模型性能的影响

在实验中,通过调整参数 α_1 和 α_2 分别研究了均值 μ_{future} 和方差 σ_{future} 对时间序列预测性能的影响。具体实验结果如图3所示。实验结果表明, α_1 和 α_2 的合理配置对模型性能具有重要作用。具体而言,适度的均值约束($\alpha_1=0.5$ 或 $\alpha_1=0.75$)和方差约束($\alpha_2=0.25$ 或 $\alpha_2=0.5$)能够有效提升模型的长期预测能力,尤其在回视步长较长(如336到720)的情况下,周期性和趋势性突出的数据集(如ETTm2和ETTh1)表现尤为优异。例如,在ETTm2数据集中, $\alpha_1=0.75$ 和 $\alpha_2=0.25$ 时,MSE最低,表明适度的均值 μ_{future} 和方差 σ_{future} 的约束能够帮助模型更好地捕捉目标分布的动态特性。相反,当 $\alpha_1=0$ 或 $\alpha_2=0$ 时,模型忽略 μ_{future} 和 σ_{future} 的约束,MSE显著升高,尤其在长步长条件(如336步以上)下,模型对长期趋势和分布动态的建模能力大幅减弱。此外,尽管在短回视步长(如36到96)下, $\alpha_1=1$ 或 $\alpha_2=1$ 能取得较好的效果,但在长回视步长下,过度依赖均值 μ_{future} 或方差 σ_{future} 的绝对约束会削弱对局部动态特征的敏感性,导致预测精度下降。这表明,均值 μ_{future} 和方差 σ_{future} 的约束需要在短期动态与长期趋势之间实现平衡,以确保模型预测性能的稳定性和可靠性。

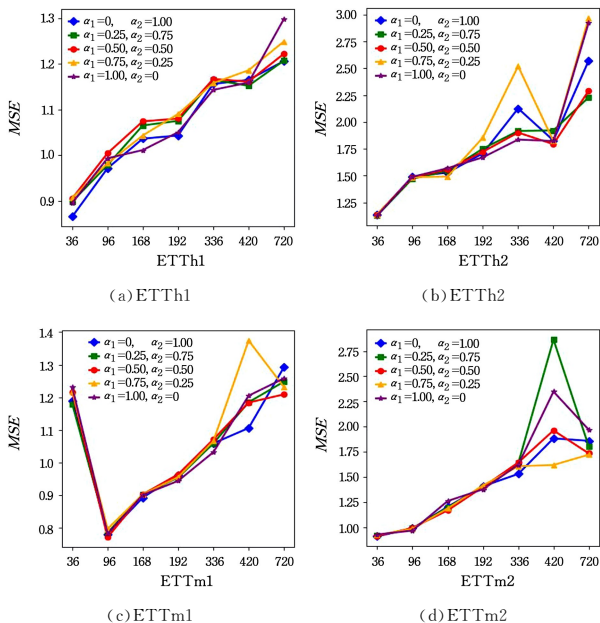


图3 不同 α_1 和 α_2 取值对时间序列预测性能的影响

$$(\alpha_1, \alpha_2 = \{0, 0.25, 0.5, 0.75, 1\})$$

Fig. 3 Impact of different α_1 and α_2 values on time series forecasting performance ($\alpha_1, \alpha_2 = \{0, 0.25, 0.5, 0.75, 1\}$)

实验进一步分析了分布漂移优化权重参数 β 对模型性能

的影响,发现适度的分布漂移优化对长期预测至关重要。实验结果如图4所示。当 $\beta=0$ 时,模型完全忽略输入序列分布漂移,这在ETTm2和ETTh2等分布漂移特性突出的数据集上表现较差,MSE在长回视步长(如336到720)下大幅上升。而当 $\beta=1$ 时,模型过度依赖输入分布漂移的变化,尽管在短步长(如36到96)下能够捕捉到部分变化特征,但在长步长(如420到720)下,目标分布 $(\mu_{\text{future}}, \sigma_{\text{future}})$ 的学习能力下降会导致性能退化。在所有实验条件下, $\beta=0.3$ 是相对最优的选择,能够有效平衡输入序列 $(\mu_{\text{slide}}, \sigma_{\text{slide}})$ 的分布漂移和目标分布 $(\mu_{\text{future}}, \sigma_{\text{future}})$ 的动态特征,较好地提高了模型在分布漂移突出数据集(如ETTm2)中的长期预测效果。以上实验结果表明,在时间序列预测任务中,适度的分布漂移权重优化和均值、方差约束的合理配合,能够显著提高模型的预测性能,尤其是在长回视步长下的复杂时间序列中表现出强大的泛化能力。这为模型设计提供了重要启示,即通过动态调整 α_1 , α_2 和 β 参数,能够更好地适应不同时间序列的特性,为高效的长期预测提供理论支持和实践依据。具体实验结果如图4所示。

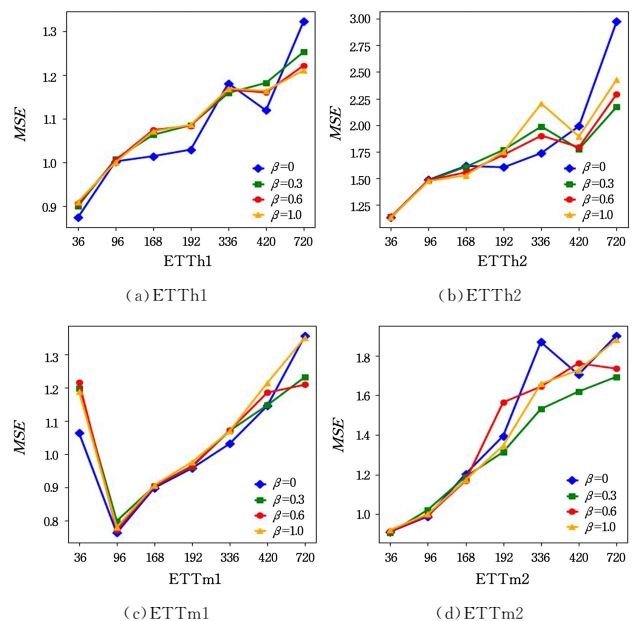


图4 不同 β 值对时间序列预测性能的影响($\beta = \{0, 0.3, 0.6, 1\}$)

Fig. 4 Impact of different β values on time series forecasting performance ($\beta = \{0, 0.3, 0.6, 1\}$)

4.4 D-LINet 模型性能评估

本文实验评估了D-LINet模型在计算成本(MACs)、内存使用、参数数量和推理时间等方面的实际效率。实验在Weather数据集上进行,回视步长和预测步长均为 $T=420$ 。对比模型包括Transformer模型和其他变种的模型,如Informer, Autoformer。

如表4所列,D-LINet模型在计算效率上表现突出,仅需需要 5.80×10^8 MACs和 4.3×10^5 参数,推理时间为0.2 ms,内存消耗为1630 MiB,因此其在实际部署中成为了一个高效的选择。相比之下,其他模型如Transformer和Autoformer需要更多的计算资源和更长的推理时间,其中Transformer需要 2.84×10^{11} MACs和28 ms的推理时间,Autoformer则需要 3.31×10^8 MACs和32.4 ms的推理时间。

表4 回视步长和预测步长均为 $T=420$ 的模型性能评估实验Table 4 Model performance evaluation with lookback and prediction step length of $T=420$

Method	MACs	Parameter	Time/ms	Memory/MiB
D-LINet	5.80×10^8	4.300×10^5	0.2	1630
Transformer	2.84×10^{11}	1.059×10^7	28.0	1975
Informer	2.59×10^8	1.138×10^7	28.6	2091
Autoformer	3.31×10^8	1.061×10^7	32.4	2050

结束语 本文提出的 D-LINet 框架针对时间序列预测中分布漂移与长程依赖的核心挑战,通过双向归一化策略与双线性层设计,构建了一个能够有效应对输入与输出分布漂移的模型,大幅提升了对时间序列周期性和趋势性特征的捕捉能力,显著提升了模型在复杂场景下的预测能力。在 7 个真实数据集上的实验结果表明,D-LINet 在短期与长期预测任务中均展现出优异的预测精度,尤其在捕捉周期与趋势特征方面表现突出。实验还深入探讨了输入窗口长度与先验知识对预测性能的影响。结果表明,合理选择输入窗口长度和融合先验知识(如滑动窗口与目标序列均值)能够较好地提升模型的预测精度和反归一化效果,为时间序列预测模型的优化设计提供重要指导。

D-LINet 尽管展现了卓越的预测性能,但仍存在改进空间。未来研究可进一步探索动态窗口调整策略,以适应各类数据集特性;同时结合多模态数据及更复杂的先验知识,以增强时间序列预测的普适性。此外,在特定场景中进一步优化模型以应对更复杂的分布漂移模式,亦是值得深入研究的方向。综上所述,本文通过理论分析与实验验证,为复杂时间序列预测任务中的模型设计与优化奠定了坚实的理论基础与实践依据。

参 考 文 献

- [1] KHAN Z A, HUSSAIN T, ULLAH A, et al. Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid cnn with a lstm-ae based framework [J]. *Sensors*, 2020, 20: 1399.
- [2] LI Y, GAO Y, YAO Z X, et al. Intelligent traffic flow prediction for data scarcity scenarios [J]. *Journal of Software*, 2024, 35(4): 7018-7030.
- [3] ANGRYK R A, MARTENS P C, AYDIN B, et al. Multivariate time series dataset for space weather data analytics [J]. *Scientific Data*, 2020, 7: 227.
- [4] CHEN W, YANG Y Y, LIU J Y, et al. A combination model based on sequential general variational mode decomposition method for time series prediction [J]. *arXiv:2406.03157*, 2024.
- [5] XU S J, CHAN H. Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2019, 122: 169-180.
- [6] JIA Y, LIN Y F, YU J, et al. PGN: The RNN's new successor is effective for long-range time series forecasting [J]. *arXiv:2409.17703*, 2024.
- [7] HOU H W, YU F R, RWKV T S. Beyond traditional recurrent neural network for time series tasks [J]. *arXiv:2401.09093*, 2024.
- [8] HU A, WANG D K, DAI Y, et al. TimeCNN: Refining cross-variable interaction on time point for time series forecasting [J]. *arXiv:2410.04853*, 2024.
- [9] SHI J, MYANA R, STEBLIANKIN V, et al. Explainable parallel rnn with novel feature representation for time series forecasting [C] // *International Workshop on Advanced Analytics and Learning on Temporal Data*. IEEE Computer Society, 2023: 56-75.
- [10] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C] // *Advances in Neural Information Processing Systems*. IEEE Computer Society, 2017: 5998-6008.
- [11] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [J]. *arXiv:1810.04805*, 2018.
- [12] DONG L, XU S, XU B. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition [C] // *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Computer Society, 2018: 5884-5888.
- [13] LIU Z, LIN Y, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows [C] // *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, 2021: 10012-10022.
- [14] ZENG A, JU X, YANG L, et al. DeciWatch: A simple baseline for 10x efficient 2D and 3D pose estimation [C] // *17th European Conference on Computer Vision (ECCV)*. IEEE Computer Society, 2022: 607-624.
- [15] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting [C] // *Proceedings of the AAAI Conference on Artificial Intelligence*. IEEE Computer Society, 2021: 11106-11115.
- [16] XU J, WANG J, LONG M, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting [C] // *Advances in Neural Information Processing Systems*. IEEE Computer Society, 2021: 22419-22430.
- [17] ZENG A, CHEN M H, ZHANG L, et al. Are transformers effective for time series forecasting? [C] // *Proceedings of the AAAI Conference on Artificial Intelligence*. IEEE Computer Society, 2023: 11121-11128.
- [18] WANG R, ZHANG Y C, WANG W D, et al. Algorithm of mixed traffic scheduling among data centers based on prediction [J]. *Journal of Computer Research and Development*, 2021, 58(6): 1307-1317.
- [19] SALINAS D, FLUNKERT V, GASTHAUS J, et al. DeepAR: Probabilistic forecasting with autoregressive recurrent networks [J]. *International Journal of Forecasting*, 2020, 36(3): 1181-1191.
- [20] RANGAPURAM S S, SEEGER M W, GASTHAUS J, et al. Deep state space models for time series forecasting [J]. *Ad-*

vances in Neural Information Processing Systems, 2018, 31: 7785-7794.

[21] WEN Y M, LIU S, MIAO Y Q, et al. Survey on semi-supervised classification of data streams with concept drifts [J]. Journal of Software, 2022, 33(4): 1287-1314.

[22] FAN W, WANG P, WANG D, et al. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting [C] // Proceedings of the AAAI Conference on Artificial Intelligence. IEEE Computer Society, 2023: 7522-7529.

[23] Local climatological data [DB/OL]. <https://www.nccl.noaa.gov/data/local-climatological-data/>.

[24] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting [C] // Proceedings of the AAAI Conference on Artificial Intelligence. IEEE Computer Society, 2021: 11106-11115.

[25] Electricity load diagrams 2011 - 2014 [DB/OL]. <https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014>.

[26] LOSHCHELOV I, HUTTER F. Decoupled weight decay regu-

larization [C] // 7th International Conference on Learning Representations. IEEE Computer Society, 2019: 1-11.

[27] DU S D, LI T R, YANG Y, et al. A sequence-to-sequence spatial-temporal attention learning model for urban traffic flow prediction [J]. Journal of Computer Research and Development, 2020, 57(8): 1715-1728.

[28] PASZKE A, GROSS S, MASSA F, et al. Pytorch: An imperative style, high-performance deep learning library [J]. Advances in Neural Information Processing Systems, 2019, 32: 8024-8035.



GENG Haijun, born in 1983, Ph.D, professor, is a member of CCF (No. 84802M). His main research interests include network security and network architecture.

(责任编辑:柯颖)

CCF 将于 4 月在北京举行青少年科学教育交流活动暨首次 CCF LMCC 青少年组全国活动

CCF 于 2025 年 8 月推出 CCF 大模型能力认证 (Large Model Competence Certification, 简称 LMCC), 并于 2025 年 11 月进行首次认证。

为给青少年提供更多交流机会, CCF 将于 2026 年 4 月在中国人民大学附属中学举行 CCF 青少年科学教育交流活动暨首次 CCF LMCC 青少年组全国活动, 2025 LMCC 青少年组认证等级评定者、计算机方面的优秀中学生及其指导教师可以优先报名本次活动。

活动期间将进行上机测试, 并分别在中国人民大学、清华大学、北京大学三所高校举行讲座交流和参观等丰富活动。

作为 LMCC 系列活动之一, CCF 青少年科学教育交流活动暨 LMCC 青少年组全国活动旨在增进中学生之间以及中学与高校间的交流, 培养对人工智能和大模型感兴趣的青少年并普及大模型基本素养。

本次活动的报名通知预计将于近期在 LMCC 官网发布, 具体报名时间、报名要求及相关信息以报名通知公示信息为准, 请持续关注。

据 CCF 微信公众号