中国科学 : 信息科学 2026 年 第 56 卷 第 2 期: 345–361
SCIENTIA SINICA Informationis
论文

面向富信息时序预测的元数据变换器网络

董家祥[1,2†], 吴海旭[1,2*†], 王雨轩[1,2†], 张力[1,2], 王建民[1,2], 龙明盛[1,2*]

1. 清华大学软件学院, 北京 100084
2. 北京信息科学与技术国家研究中心, 北京 100084
* 通信作者. E-mail: wuhx23@mails.tsinghua.edu.cn, mingsheng@tsinghua.edu.cn
† 同等贡献

收稿日期: 2024–12–31; 修回日期: 2024–12–31; 接受日期: 2025–07–21; 网络出版日期: 2026–01–28

国家自然科学基金 (批准号: U2342217, 62022050) 资助项目

摘要 时间序列预测在金融分析、能源规划等领域中具有重要的应用价值. 以往研究主要聚焦于时间序列自身模态, 致力于捕捉其数值变化特征. 然而, 本文注意到, 除了数值模态外, 描述领域、变量和任务的元数据能够有效区分不同的预测场景, 对时间序列的准确预测十分重要. 基于这一观察, 本文提出了一种元数据引导的时序变换器网络 MetaTST, 能够有效地将特定场景下的多层级元数据融入深度模型, 实现富信息支撑的时间序列预测. 具体地, MetaTST 首先通过预定义的解析模板, 将非结构化元数据信息标准化为自然语言形式的元数据描述, 然后利用大语言模型将其编码为元数据令牌, 以作为原有数值模态的补充; 进一步, MetaTST 采用 Transformer 编码器将时间序列数据与元数据令牌深度融合, 借助丰富的元数据信息扩展时序表征, 让模型能够自适应感知不同的预测场景与任务, 从而实现更准确的时序预测. 在实验中, MetaTST 在时间序列短期和长期预测基准数据集上取得了一致最优的预测性能, 特别是在处理大规模、多样化的时间序列预测任务时表现出色.

关键词 时间序列预测, 元数据, 深度学习, 大语言模型, Transformer

1 引言

时间序列预测在能源、交通、气象、金融等诸多领域的应用需求日益增长 [1~4], 受到了学界和业界的广泛关注. 此前的相关研究主要通过深度模型学习时间序列在历史观测中的变化特征, 从而实现未来预测 [5~8]. 虽然前人的工作在捕捉复杂的时序变化模式方面取得了一系列进展, 但是本文注意到, 仅仅依赖历史观测的数值信息往往无法对时间序列的未来变化趋势进行准确预测. 以经典的交通流预测场景为例, 不同的十字路口在早高峰时往往呈现出相似的交通流量变化, 但由于附近公司及商家的下班和闭店时间不同, 导致未来变化趋势存在明显差异. 尽管历史观测中可能存在能够捕捉上述场景差异的细微线索, 但这要求预测模型能够综合考虑较长的时间跨度并准确识别隐藏在历史观测中的潜在模式差异, 对模型的建模能力和计算效率带来了极大的挑战.

为了更加直接有效地为预测模型提供有价值的上下文场景信息, 我们注意到, 构建时间序列的元数据, 对提升时序预测任务的性能具有重要价值. 以文本或其他数据格式记载的元数据可以刻画时间序列预测任务的上下

引用格式: 董家祥, 吴海旭, 王雨轩, 等. 面向富信息时序预测的元数据变换器网络. 中国科学: 信息科学, 2026, 56: 345–361, doi: 10.1360/SSI-2024-0397
Dong J X, Wu H X, Wang Y X, et al. Metadata-informed transformers for informative time series forecasting. Sci Sin Inform, 2026, 56: 345–361, doi: 10.1360/SSI-2024-0397

© 2026《中国科学》杂志社 www.scichina.com infocn.scichina.com
https://www.sciengine.com/doi/10.1360/SSI-2024-0397

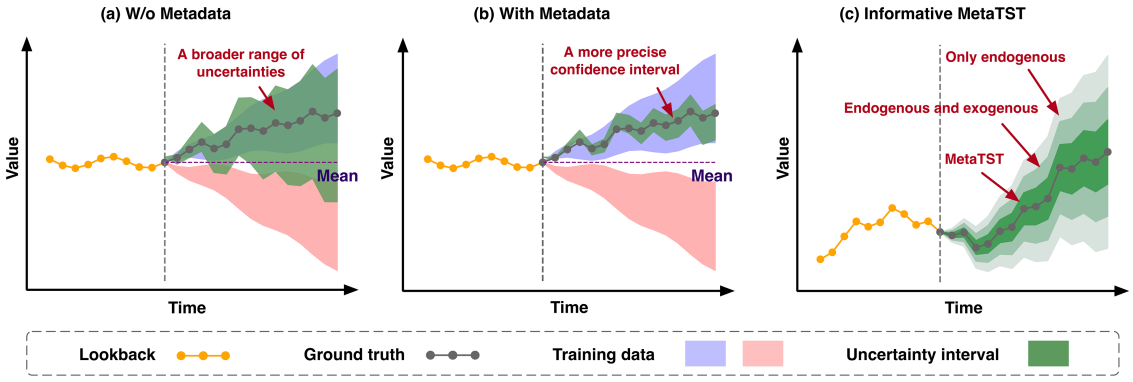


图 1　(网络版彩图) 不同时间序列预测范式的概念性对比. (a) 无元数据的经典时间序列预测范式; (b) 元数据引导的时间序列预测范式; (c) MetaTST 利用上下文特定的元数据以及协变量, 实现高确定性的富信息时间序列预测.

Figure 1　(Color online) A conceptual illustration for different time series forecasting paradigms. (a) Canonical time series forecasting without metadata; (b) metadata-informed time series forecasting; (c) MetaTST utilizes context-specific metadata to achieve highly certain informative forecasts.

文信息, 并能够有效区分不同的时序预测场景, 从而显著提升模型在多样化时序预测任务中的适应能力. 值得注意的是, 元数据通常是非结构化的, 它包含了来自多种异构视角下文本化的相关描述. 因此, 尽管元数据具有引导时间序列预测的潜在优势, 但是 "如何合理、高效地将其融入时间序列的预测任务中辅助时间序列预测" 仍然未被充分探索.

从时间序列预测任务的问题定义出发, 并综合考虑实际预测场景及数据获取难度后, 我们总结出影响时间序列准确预测的 3 个关键要素: (1) 准确捕捉目标时间序列的内在时间变化, 对应内生变量 (endogenous series); (2) 充分理解影响目标序列的外部因素, 对应外生变量 (exogenous series); (3) 合理引入与预测场景相关的上下文描述信息, 对应元数据 (metadata). 此前研究仅关注模型的架构设计以学习时序数据内生变量中的依赖关系 [6,9], 只有少部分工作强调了外生变量的关键性 [10,11]. 如图 1 所示, 缺乏描述预测场景的元数据会使模型需要考虑更广泛的未来可能性, 阻碍模型输出更确定性的结果. 具体而言, 回顾上述的交通流预测问题, 交通流量变化不仅依赖历史流量观测, 还与假期、管控政策甚至传感器位置等因素密切相关, 当缺乏对预测场景的描述时, 如果两个场景历史观测相似但未来变化大相径庭时, 会造成模型优化目标的混乱, 导致预测结果的不确定性范围增加.

受上述观点启发, 本文提出了一种基于变换器网络 (Transformer) 的时序预测模型 MetaTST, 以实现元数据引导下的富信息时间序列预测. 为解决元数据的非结构化特性, MetaTST 设计了包括数据集、任务和样本 3 个层次的自然语言元数据解析模板, 将元数据整理为标准化文本格式. 此外, 不同于已有的以大语言模型为主体的时间序列分析方法 (LLM4TS) [12,13], 这类工作试图通过微调大语言模型对齐时间序列和自然语言模态, 从而利用大语言模型强大的推理建模能力进行时间序列预测. MetaTST 将大语言模型作为独立的文本元数据编码器, 以保持其原有的自然语言理解能力. 进一步地, MetaTST 基于 Transformer 将大语言模型生成的多层次元数据令牌、目标时间序列的片段级内生时序令牌、相关时间序列的序列级外生时序令牌进行深度融合, 实现富信息时间序列预测. 特别地, MetaTST 能够通过学习上下文特定的时序变化模式来处理大规模、多样化的预测任务, 这也为利用大规模数据训练时间序列基础模型提供了技术支撑. 凭借丰富的元数据信息, MetaTST 在时间序列的短期和长期预测任务中实现了一致最先进的预测性能, 测试涵盖了单一数据集的独立训练和多数据集的联合训练场景. 综上所述, 本文主要贡献可以概括为以下 3 个方面.

- 全面思考并探索了影响时间序列预测的关键因素, 提出元数据引导的时序变换器网络 MetaTST, 将多层次元数据信息引入深度模型以增强时序表征, 实现更准确、可靠的富信息时间序列预测.
- 不同于以往微调大语言模型进行时序分析的方法, MetaTST 提出将固定参数的完整大语言模型作为独立的元数据编码器, 能够充分保持并发挥大语言模型强大的文本理解能力, 同时更好地捕捉多样化场景中上下文特

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 1　相关工作对比. “TS” 是时间序列的缩写, “–” 表示该方法没有使用大语言模型.

Table 1　Related work comparison. “TS” is short for time series. “–” refers to without LLMs.

Method	MetaTST	TimeLLM [13]	GPT4TS [12]	TimeXer [25]	iTransformer [7]	PatchTST [6]	DLinear [43]
Input modality	TS + language	TS + language	TS + language	TS	TS	TS	TS
LLMs usage	Metadata encoder	Main backbone	Main backbone	–	–	–	–

定的时序变化模式.

• MetaTST 在 12 个真实的时间序列短期和长期预测基准数据集中显著超越了 10 个先进的基线方法, 测试涵盖了单一数据集独立训练和多数据集联合训练场景, 充分验证了所提出模型的有效性.

2　相关工作

2.1　原生时间序列模型

近年来, 深度模型在时间序列领域的应用引起了广泛关注 [2]. 已有的研究基于不同的深度模型架构设计原生的时间序列模型以解决时序预测问题, 其中包括卷积神经网络 (convolutional neural network, CNN) [14~16] 及循环神经网络 (recurrent neural network, RNN) [17~20] 等. 然而, 这些模型常受到有限感受野的限制, 难以捕捉时间维度上的长期依赖关系. 此外, 部分基于多层感知机 (multilayer perceptron, MLP) 设计的时间序列模型 [21~23] 虽然展现出良好的预测效果和推理效率, 但是受限于模型容量不足, 在处理复杂预测场景时表现欠佳. 作为基础模型架构的里程碑, Transformer 模型也被广泛应用于时间序列预测任务, 以充分建模时间维度上的长期依赖并挖掘数据内部复杂的时序变化 [24]. 其中, PatchTST [6] 基于分块 (patch) 技术, 提出了对时间片段之间的时间依赖进行建模的方法, 显著提高了预测效果. iTransformer [7] 将时间序列包含的多条相关序列分别编码为单独的序列级令牌, 并利用注意力机制捕捉序列间的相关性, 颠覆了注意力机制与前馈网络的传统职责. 进一步地, TimeXer [25] 通过利用不同层次的序列表征以同时捕捉时间和变量维度的关联关系, 取得了更准确的预测效果. 除此之外, 受大语言模型成功训练与应用的启发, 基于大规模时序数据训练的时间序列大模型也逐渐受到关注 [26~30]. 已有研究工作 [26,31] 通过构建跨领域的大规模时间序列数据集来训练原生的时间序列模型, 旨在使模型能够学习到更加丰富、多样的时序变化模式. 然而, 这些方法通常仅关注时间序列自身模态的建模, 忽视了描述预测场景的元数据在时序预测任务中的重要作用, 这极大地限制了模型对不同领域数据中模式变化差异的感知能力. 本文提出的元数据引导下的富信息时间序列预测范式能够有效地解决这一问题.

2.2　基于大语言模型的时间序列模型

随着大语言模型领域的迅速发展 [32~37], 越来越多的研究方法开始利用大语言模型进行时间序列分析 [38,39]. 这类研究面临的核心挑战在于如何有效地将大语言模型应用于时间序列分析任务. 其中, 一类经典方法以预训练的大语言模型为主体, 通过使用时间序列数据对大语言模型参数进行微调, 使其具备时间序列分析的能力. 例如, GPT4TS [12] 基于预训练的 GPT-2 模型 [35], 通过微调大语言模型的位置嵌入和归一化层, 使其能够适配多种时间序列分析任务. 此外, LLM4TS [40] 提出了一种两阶段的微调策略, 包括时间序列对齐与预测微调, 以使大语言模型更好地适配时间序列数据. 不同于上述方法, 另一类研究则在保持大语言模型参数冻结的前提下, 充分利用其强大的表示学习和推理能力. 例如, TimeLLM [13] 通过将时间序列重构为文本形式, 实现时间序列数据与自然语言模态的对齐; AutoTimes [41] 学习了一个新的时间序列到自然语言模态的映射层, 将时间序列嵌入到语言模型的潜在语义空间中. 尽管相关研究已经取得了一定进展, 但是文献 [42] 通过一系列分析实验发现, 现有方法尚未充分挖掘大语言模型在时间序列任务中的潜力, 导致其预测效果有限且计算代价高昂. 如表 1 [6,7,12,13,25,43] 所示, 与以往将大语言模型作为主体的工作不同, 本文提出的 MetaTST 将大语言模型作为元数据编码器, 专注于处理文本格式的元数据, 从而充分发挥大语言模型在自然语言理解方面的优势.

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

2.3　时间序列元数据

元数据是指描述其他数据信息的数据, 被定义为 “描述数据的数据”, 其通过结构化形式记录数据的特征、上下文和语义信息, 为其他数据提供解释与说明 [44, 45]. 时间序列的元数据应用广泛, 在时序数据库领域, 元数据通常用于记录数据源详情和统计摘要等信息 [46, 47], 其对于设计高效的数据组织形式、提升查询效率至关重要. 除数据管理以外, 可描述的元数据还可以刻画时间序列分析任务的上下文语义信息, 使模型能够更全面、深入地理解不同的时序分析场景. 本文聚焦于时间序列预测任务, 创新性地提出了一种涵盖数据集、任务和样本的多层次元数据描述框架, 不仅为时序预测任务提供了丰富的上下文信息, 还能有效增强模型对不同预测场景的适应能力, 从而有效提升预测准确性.

3　方法

如前文所述, 为了解决现有时间序列预测研究对元数据信息考虑不足带来的建模挑战, 本文提出富信息时间序列预测范式, 并设计天然实现该预测范式的元数据时序变换器网络 MetaTST. 与现有方法不同, MetaTST 不仅关注时间序列自身的模态信息, 还将描述特定预测场景上下文的关键元数据信息融入时间序列预测过程中, 从而增强深度模型在多变的预测场景下对不同的预测任务的理解能力. 在方法设计上, MetaTST 精心设计了一种基于大语言模型的元数据嵌入机制, 能够生成多层次的元数据令牌. 这些多层次元数据令牌与不同粒度的内生和外生序列令牌共同输入 Transformer 编码器, 协同强化模型学习到的内生序列表征, 从而实现富信息时间序列预测.

3.1　富信息时间序列预测

本文首先提出了富信息时间序列预测范式, 其核心在于构建丰富的信息输入, 以提高对内生序列未来值的预测准确性. 具体来说, MetaTST 在充分考虑实际应用场景需求的基础上, 精心筛选影响时间序列预测的关键因素, 以构建富信息嵌入, 包括内生序列的历史观测值 $\mathbf{x}_{\mathrm{en}} \in \mathbb{R}^{T_{\mathrm{en}}}$, 多个与内生序列相关的外生序列 $\mathbf{x}_{\mathrm{ex}} = \{\mathbf{x}_{\mathrm{ex},1}, \mathbf{x}_{\mathrm{ex},2}, \ldots, \mathbf{x}_{\mathrm{ex},C}\} \in \mathbb{R}^{T_{\mathrm{ex}} \times C}$ 和元数据 $\mathbf{x}_{\mathrm{meta}}$. 其中, T_{en} 和 T_{ex} 分别表示内生、外生序列的历史输入长度, C 表示外生序列的数量. 与标准的时间序列预测形式不同, 本文提出的富信息时间序列预测的目标是优化深度模型, 通过整合多种不同类型的输入, 准确地预测内生序列 $\mathbf{y}_{\mathrm{en}} \in \mathbb{R}^{S}$ 未来的 S 个时间步, 具体如下:

$$\arg\min_{\theta} \|\mathbf{y}_{\mathrm{en}} - \mathcal{F}_{\theta}\big(\mathbf{x}_{\mathrm{en}}, \mathbf{x}_{\mathrm{ex}}, \mathbf{x}_{\mathrm{meta}}\big)\|_2^2, \tag{1}$$

其中, $\mathcal{F}_{\theta}(\cdot)$ 表示参数为 θ 的时间序列预测模型. 值得注意的是, 本文所采用的元数据是关于预测场景上下文的描述信息, 包括数据集、任务描述以及变量含义等. 鉴于此类元数据本身已经包含在预测任务的定义之中, 因此在实际应用中获取成本极低. 这使得本文提出的富信息预测范式能够相对简单地从标准设置中扩展实现, 无需投入大量资源去收集或标注额外的数据.

3.2　元数据嵌入

鉴于元数据的非结构化性质, 本文设计了一种多层次的元数据解析方法, 采用精心设计的自然语言模板对元数据进行结构化处理, 并进一步利用大语言模型作为元数据编码器, 充分发挥其在预训练阶段所学习的丰富先验知识和强大的语义理解能力, 从多个层面全面理解时间序列数据和预测场景.

多层次元数据解析. 如图 2 所示, MetaTST 设计了 3 种不同类型的元数据, 以从 3 个不同的角度全面描述预测上下文信息. (1) 数据集层级: 提供有关数据集的基本属性, 如数据采集领域、采样频率与周期等, 为模型赋予与预测场景相关的外部先验知识; (2) 任务层级: 引入对任务的基本描述, 如任务关注的目标、模型的输入和输出长度等, 增强模型对特定预测行为的理解; (3) 样本层级: 揭示时间序列样本的动态统计信息, 如样本采集的开始时间、样本均值和标准差等, 使模型能够显示考虑样本之间的细微差异. 进一步地, 为了有效应用多种类型的元数据信息, MetaTST 详细设计了一个元数据解析模块 MetaParser(·), 它利用预定义的语言模板将原始的元

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

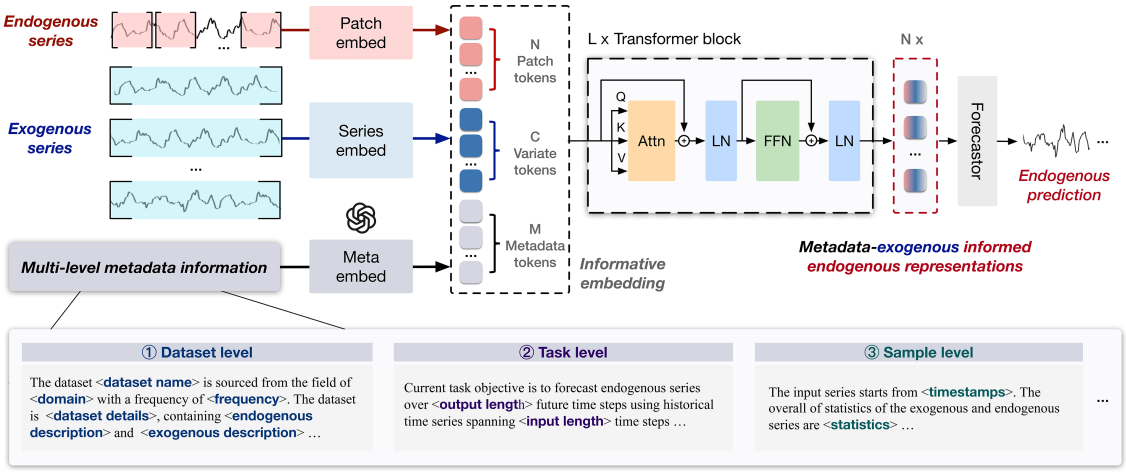


图 2　(网络版彩图) MetaTST 的整体设计. 该模型整合了内生序列、外生序列以及上下文特定的文本形式元数据, 旨在基于 Transformer 实现元数据引导的富信息时间序列预测.

Figure 2　(Color online) Overall design of MetaTST, which integrates endogenous series, exogenous series, and context-specific textual metadata to enable informative time series forecasting with Transformers.

数据信息形式化为多个不同层次的自然语言形式的元数据描述. 各个层次的元数据描述均能够从不同的角度为理解预测任务提供多元的信息视角, 进而强化模型对特定预测上下文的理解. 多层次元数据解析过程可以形式化为

$$\{\hat{\mathbf{x}}_{\text{meta},k}\}_{k=1}^{M} = \text{MetaParser}\left(\mathbf{x}_{\text{meta}}\right), \tag{2}$$

其中, M 是用于控制元数据层次的超参数. 在 MetaTST 中, 我们使用了数据集、任务以及样本三类元数据构建富信息时间序列预测, 因此该超参数默认设置为 3.

基于大语言模型的元数据编码. 为了充分利用多层次的元数据, MetaTST 使用大语言模型作为元数据编码器 LLMEncoder(·) 以全面理解元数据中蕴含的与预测场景相关的语义信息. 值得注意的是, MetaTST 所选择的元数据编码器并不受限于大语言模型的模型架构, 既可以选择自回归大语言模型, 如 LLaMA-3-8B[1)], GPT-2 [2], 也可以是自编码大语言模型, 如 T5 [34] 和 BERT [33] 等. 如前文所述,MetaTST 为三类元数据分别构建了自然语言模板, 并为每类元数据创建了对应的描述性段落, 以生成多样的元数据描述. 不同层次的元数据描述输入大语言模型后, 可以获得一系列对应不同元数据类型的词级令牌序列 (word-level token sequence). 随后, 我们按照其所属的元数据类型, 将多个词级令牌序列转化为全局令牌 (global-level token), 并将其融入时间序列的预测任务中.

如图 3 所示, 本文探索了 3 种不同的词级到全局级令牌的信息聚合方法: (a) 使用特殊的全局令牌, 该方法通过初始化特殊的全局令牌, 并基于自注意力机制汇聚不同词级令牌信息到全局令牌, 如 BERT 中提出的 [CLS] 令牌 [33]; (b) 使用平均池化计算所有词级令牌的平均值表示为全局令牌; (c) 基于交叉注意力 (cross-attention) 的路由机制, 遵循文献 [25, 48] 中提出的信息路由方法, 定义固定数量的随机初始化令牌, 并基于交叉注意力机制来汇聚词级令牌的信息到固定数量的全局令牌中. 通过实验效果对比, 我们发现 (b) 平均池化方法在多数情况下表现出更优的预测效果 (见后文实验). 因此, 我们选择平均池化 AvgPooling(·) 作为 MetaTST 的默认设置. 此外, 本文采用一个包含两个线性层 (MLP) 及中间包含激活函数 (ReLU) 的模态对齐模块 ModalAlign(·) 以确保大语言模型与原生时间序列模型在模态和潜在语义空间上的对齐, 具体形式如下所示:

$$\begin{aligned}\{\widetilde{\mathbf{h}}_{\text{meta},k}\}_{k=1}^{M} &= \text{AvgPooling}\left(\text{LLMEncoder}\left(\{\hat{\mathbf{x}}_{\text{meta},k}\}_{k=1}^{M}\right)\right),\\ \{\mathbf{h}_{\text{meta},k}\}_{k=1}^{M} &= \text{ModalAlign}\left(\{\widetilde{\mathbf{h}}_{\text{meta},k}\}_{k=1}^{M}\right).\end{aligned} \tag{3}$$

1) https://llama.meta.com/llama3.

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

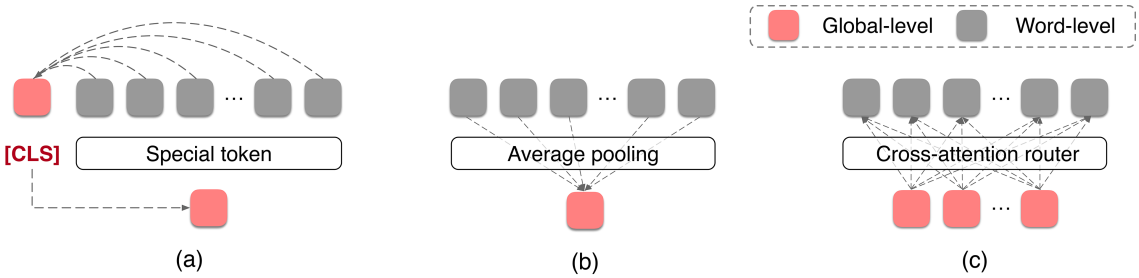


图 3　(网络版彩图) 不同的词级令牌序列到全局令牌的信息聚合方法对比, 包括 (a) 全局令牌, (b) 平均池化和 (c) 交叉注意力 3 种实现方法.

Figure 3　(Color online) Different aggregating methods to transform word-level token sequences to global-level, including three different implementations: (a) special token, (b) average pooling, and (c) cross-attention.

本文整体的元数据编码过程可以总结为 $\{\mathbf{h}_{\text{meta},k}\}_{k=1}^{M} = \text{MetaEmbed}\left(\{\hat{\mathbf{x}}_{\text{meta},k}\}_{k=1}^{M}\right)$.

3.3　MetaTST

进一步地, 本文基于 Transformer 构建了元数据引导的时间序列预测模型 MetaTST. MetaTST 利用时间序列的内生序列、外生序列和元数据构建富信息嵌入, 并基于 Transformer 编码器强化内生序列表征, 从而提升时间序列的预测性能, 实现富信息时间序列预测.

富信息嵌入. 遵循文献 [6, 49] 中使用的时间序列分块建模方法, 对于内生序列 $\mathbf{x}_{\text{en}}$, MetaTST 将其分割成 $N = \lfloor \frac{T_{\text{en}}}{P} \rfloor$ 个不重叠的时间片段, 其中 P 为片段长度. 对于第 i 个时间片段, MetaTST 通过一个可训练的线性时间片段嵌入层 $\text{PatchEmbed}(\cdot) : \mathbb{R}^{P} \to \mathbb{R}^{D}$ 将其映射为 D 维的内生片段令牌 $\mathbf{h}_{\text{en},i}$. 对于内生序列相关的 C 条外生序列 $\{\mathbf{x}_{\text{ex},j}\}_{j=1}^{C}$, MetaTST 采用序列级线性嵌入层 $\text{SeriesEmbed}(\cdot) : \mathbb{R}^{T_{\text{ex}}} \to \mathbb{R}^{D}$, 将第 j 个相关的外生序列 $\mathbf{x}_{\text{ex},j}$ 整体映射为单独的 D 维外生序列令牌 $\mathbf{h}_{\text{ex},j}$. 上述设计可以突出内生序列的精细时间片段建模和外生序列的序列相关性建模, 同时避免外生序列间可能存在的采样频率不匹配问题 [25]. 以上内生与外生序列嵌入过程可以形式化为如下所示:

$$\begin{aligned} \{\mathbf{h}_{\text{en},i}\}_{i=1}^{N} &= \text{PatchEmbed}\left(\mathbf{x}_{\text{en}}\right), \\ \{\mathbf{h}_{\text{ex},j}\}_{j=1}^{C} &= \text{SeriesEmbed}\left(\{\mathbf{x}_{\text{ex},j}\}_{j=1}^{C}\right). \end{aligned} \tag{4}$$

除上述不同类型的序列嵌入外, 元数据令牌已经按照式 (3) 中的形式与时间序列模态对齐. 因此, MetaTST 直接结合 3 种类型的令牌构建富信息嵌入 $\mathbf{h}^0$, 其中包含 N 个片段级内生令牌、C 个序列级外生令牌和 M 个元数据令牌, 具体形式如下所示:

$$\mathbf{h}^0 = \text{Concat}\left(\{\mathbf{h}_{\text{en},i}\}_{i=1}^{N}, \{\mathbf{h}_{\text{ex},j}\}_{j=1}^{C}, \{\mathbf{h}_{\text{meta},k}\}_{k=1}^{M}\right). \tag{5}$$

富信息预测. 为了实现 3 种令牌之间充分的信息融合, MetaTST 采用具有 L 层的 Transformer 编码器进行深度表征学习. Transformer 编码器中蕴含的强大的自注意力机制能够逐步且有选择性地将不同层次的元数据信息与内生序列相关的外生序列信息汇聚至 N 个时间片段级的内生令牌中, 从而强化内生序列表示 $\mathbf{h}_{\text{en}}^{L}$, 以实现准确的富信息预测. 其具体表示如下:

$$\begin{aligned} \mathbf{h}^{l+1} &= \text{TransformerBlock}\left(\mathbf{h}^{l}\right),\ l \in \{1, \ldots, L\}, \\ \hat{\mathbf{y}}_{\text{en}} &= \text{Forecastor}\left(\mathbf{h}_{\text{en}}^{L}\right), \end{aligned} \tag{6}$$

其中, $\text{Forecastor}(\cdot)$ 为一个简单的线性回归层, 用于根据强化后的内生序列表示 $\mathbf{h}_{\text{en}}^{L}$ 预测内生序列未来 S_{en} 个时间步的值 $\hat{\mathbf{y}}_{\text{en}}$. 最后, MetaTST 基于预测值与真实值之间的 L2 损失函数进行监督训练.

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 2　数据集描述.

Table 2　Dataset descriptions.

Dataset	Dimension	Exogenous series	Endogenous series	Frequency	Domain
Electricity	321	Electricity consumption	Electricity consumption	1 h	Electricity
Weather	21	Climate feature	CO_2 concentration	10 min	Weather
ETTh1/h2	7	Power load feature	Oil temperature	1 h	Electricity
ETTm1/m2	7	Power load feature	Oil temperature	15 min	Electricity
Traffic	862	Road occupancy rates	Road occupancy rates	1 h	Transportation
NP	3	Grid load, wind power	Nord pool electricity price	1 h	Electricity
PJM	2	System load, SyZonal COMED load	Pennsylvania-New Jersey-Maryland electricity price	1 h	Electricity
BE	3	Generation, system load	Belgium's electricity price	1 h	Electricity
FR	3	Generation, system load	France's electricity price	1 h	Electricity
DE	3	Ampirion zonal load, wind power	German's electricity price	1 h	Electricity

表 3　多数据集联合训练实验中采用的统一超参数配置.

Table 3　Unified hyperparameter values for all baselines in different benchmarks of joint-training setting.

Tasks	L	d_model	d_ff	n_heads	patch-size	patch-stride	learning-rate	batch-size	training-epochs
Short-term	3	256	2048	8	24	24	$1e-4$	32	10
Long-term	3	256	2048	8	12	12	$1e-4$	32	10

4　实验

本文时间序列长期和短期预测这两类经典的任务上进行了充分的实验, 以评估本文所提出的 MetaTST 方法的有效性. 此外, 本文还构建了典型的单数据集独立训练和更具挑战性的多数据集联合训练两种实验设置, 以全面评估不同模型在复杂预测场景下的表现.

基准数据集. 在实验部分, 本文选取了 12 个时间序列分析领域广泛使用的基准数据集, 构建包含外生序列的时间序列预测任务, 以充分验证不同方法的有效性. 对于短期预测任务, 本文采用了 EPF 基准数据集 [50] 进行效果验证, 该数据集包含来自 5 个电力市场的电价预测数据. 对于长期预测任务, 我们遵循文献 [25] 中使用外生序列进行预测的实验设置, 在来自不同领域的 7 个公认基准数据集上评估了不同时序模型的预测性能, 其中多变量数据的最后一维被指定为内生序列, 其他维度被视为外生序列, 更多数据集详情如表 2 所示.

基线方法. 本文选用了 10 个先进的时间序列预测模型与 MetaTST 进行了全面的性能比较. 这些基线模型包括 2 个基于大语言模型的时间序列模型 (LLM4TS): GPT4TS [12] 和 TimeLLM [13], 以及 8 个主流的原生时间序列模型: Autoformer [1], Crossformer [48], DLinear [43], TimesNet [16], PatchTST [6], iTransformer [7], Timer [27] 和 TimeXer [25]. 所有基线方法都是在近 4 年提出的.

单数据集独立训练和多数据集联合训练. 已有的研究通常关注单数据集独立训练的结果 [1,6], 即模型的训练集仅包含来自单一领域的单个数据集. 随着大语言模型的迅速发展, 时间序列基础模型的研究也受到了广泛关注, 发展其处理大规模、多样化预测场景的能力已成为一项不可或缺的要求. 因此, 本文进一步构建多数据集联合训练场景, 以评估 MetaTST 在复杂训练场景下的训练效率和预测性能. 与单数据集独立训练相比, 多数据集联合训练要求模型能够同时处理多样化的训练数据, 并在变化的数据分布、不一致的变量数量以及不同的训练场景下充分学习具有显著差异的时序变化, 从而实现良好的泛化性能, 其训练难度更具挑战性. 值得注意的是, 由于不同的数据集包含不同的变量数, 导致并非所有基线方法都能同时处理不同的数据集. 因此, 在多数据集联合训练实验中, 本文仅选择部分先进的基线方法, 如 PatchTST [6], iTransformer [7] 和 TimeXer [25] 进行效果对比.

评价指标. 本文采用时间序列预测任务中最常用的均方误差 (mean squared error, MSE) 和平均绝对误差

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 4　单数据集独立训练下的时间序列短期预测结果. 最优效果被加粗标识.

Table 4　Short-term forecasting results under single-dataset individual training. The best results are in bold.

Datasets		NP		PJM		BE		FR		DE		Avg.	
Models		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TS native	Autoformer [1]	0.402	0.398	0.168	0.267	0.500	0.333	0.519	0.295	0.674	0.544	0.453	0.367
	DLinear [43]	0.309	0.321	0.108	0.215	0.463	0.431	0.429	0.260	0.520	0.463	0.366	0.338
	TimesNet [16]	0.250	0.289	0.097	0.195	0.419	0.288	0.431	0.234	0.502	0.446	0.340	0.290
	Crossformer [48]	0.240	0.285	0.101	0.199	0.420	0.290	0.434	0.208	0.461	0.432	0.331	0.283
	PatchTST [6]	0.267	0.284	0.106	0.209	0.400	0.262	0.411	0.220	0.574	0.498	0.352	0.295
	iTransformer [7]	0.265	0.300	0.097	0.197	0.394	0.270	0.439	0.233	0.479	0.443	0.335	0.289
	Timer [27]	0.275	0.294	0.095	0.193	0.380	0.254	0.437	0.211	0.469	0.432	0.331	0.277
	TimeXer [25]	**0.236**	0.268	0.093	0.192	0.379	**0.243**	0.385	0.208	0.440	0.415	0.307	0.265
LLM4TS	GPT4TS [12]	0.282	0.302	0.109	0.219	0.421	0.281	0.395	0.220	0.513	0.459	0.344	0.296
	TimeLLM [13]	0.330	0.330	0.134	0.248	0.448	0.290	0.455	0.253	0.542	0.472	0.382	0.319
MetaTST		0.239	**0.267**	**0.089**	**0.188**	**0.364**	0.244	**0.384**	**0.210**	**0.423**	**0.409**	**0.300**	**0.264**

(mean absolute error, MAE) 作为模型预测准确性的评估指标. 这两个指标分别通过计算预测值与真实值之间的平方误差均值和绝对误差均值来衡量模型的预测性能, 其公式如下:

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}||\mathbf{y}_{\mathrm{en},i}-\widehat{\mathbf{y}}_{\mathrm{en},i}||_2^2,\ \mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}|\mathbf{y}_{\mathrm{en},i}-\widehat{\mathbf{y}}_{\mathrm{en},i}|, \tag{7}$$

其中, $\mathbf{y}_{\mathrm{en},i}$ 和 $\widehat{\mathbf{y}}_{\mathrm{en},i}$ 分别表示内生序列 $\mathbf{x}_{\mathrm{en},i}$ 的未来真实值和未来预测值, n 为样本数量.

实现细节. 为了确保公平比较, 本文涉及的所有实验均使用 PyTorch [51] 框架实现, 在单个 NVIDIA A100 40 GB GPU 上运行, 采用 ADAM 优化器 [52], 初始学习率设置为 $1\mathrm{e}-4$, 并使用 L2 损失函数进行模型优化. 在单数据集独立训练设置下, 本文依照文献 [6, 23] 中使用的实验策略, 在不同的基准数据集中搜索所有基线方法的模型超参数配置, 搜索的超参数及其搜索范围如下: Transformer 编码器的层数 $L\in\{1,2,3\}$, 隐藏层维度 d_model $\in\{128,256,512\}$, 前馈层维度 d_ff $\in\{512,1024,2048\}$, 注意力头的数量 n_heads $\in\{4,8,16\}$, 训练小批次大小 $\in\{16,32,64,128\}$, 模型神经元丢弃 (dropout) 比例 $\in\{0,0.1,0.2,0.3\}$. 另外, 所有实验中均采用了 RevIN [53] 技术, 用于缓解时序数据的分布偏移问题, 且所有对比基线均基于 TSLib [2] 复现.

由于进行了参数调优, 以上方式会使得不同的方法在处理不同基准数据集时, 模型的大小不一致, 这与统一训练一个大规模时序模型处理多种不同任务的联合训练设置相矛盾. 因此, 在多数据集联合训练实验中, 本文统一设置模型的超参数, 以确保所有基线具有可比较的参数大小, 并在不同任务中保持一致. 对于多数据集联合训练, 对于不同的方法, 本文在所有数据集上训练一个统一模型, 分别微调到不同的数据集上进行效果验证. 统一的模型配置参数如表 3 所示.

4.1　单数据集独立训练

在传统的单数据集独立训练实验设置下, 短期预测的输入和输出长度参照文献 [11] 设置为输入 168 个历史时间步预测未来的 24 个时间步. 对于时间序列的长期预测, 输入长度设被置为 96, 输出结果为 4 种涵盖不同跨度预测长度 $\{96,192,336,720\}$ 的结果平均值. Avg. 表示该模型在全部基准数据集上的平均预测性能指标, 用于评估不同模型的整体预测性能.

短期预测. 如表 4 [1,6,7,12,13,16,25,27,43,48] 所示, MetaTST 在大部分短期预测任务上始终展现出最先进的预测性能. 与以大型语言模型为主题设计的时间序列预测方法 GPT4TS [12] 和 TimeLLM [13] 相比, MetaTST 的平均均方误差 (MSE) 分别降低了 12.8% 和 21.5%. 值得注意的是, 当前最优的原生时间序列预测模型 TimeXer [25] 在 NP 和 FR 数据集上与 MetaTST 的预测效果相近, 这是因为这些数据集中不同的序列间呈现出高度的相关

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 5　单数据集独立训练下的时间序列长期预测结果.

Table 5　Long-term forecasting results under single-dataset individual training.

Datasets		ETTh1		ETTh2		ETTm1		ETTm2		Weather		Traffic		ECL		Avg.	
Models		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TS native	Autoformer [1]	0.130	0.282	0.242	0.386	0.085	0.230	0.154	0.305	0.006	0.060	0.302	0.353	0.495	0.528	0.202	0.306
	DLinear [43]	0.116	0.259	0.224	0.369	0.066	0.188	0.126	0.263	0.006	0.066	0.323	0.404	0.393	0.457	0.179	0.287
	TimesNet [16]	0.076	0.215	0.210	0.369	0.054	0.175	0.129	0.271	0.097	0.115	0.171	0.264	0.410	0.476	0.164	0.269
	Crossformer [48]	0.285	0.447	1.027	0.873	0.411	0.548	0.976	0.769	0.005	0.055	0.182	0.268	0.344	0.412	0.461	0.482
	PatchTST [6]	0.078	0.215	0.192	0.345	0.053	0.173	0.120	0.258	**0.002**	0.031	0.173	0.253	0.394	0.446	0.145	0.246
	iTransformer [7]	0.075	0.211	0.199	0.352	0.053	0.175	0.127	0.267	**0.002**	0.031	0.161	0.246	0.365	0.442	0.140	0.246
	Timer [27]	0.081	0.220	0.186	0.344	0.053	0.173	0.139	0.280	**0.002**	0.034	0.340	0.409	0.364	0.425	0.166	0.269
	TimeXer [25]	0.073	0.209	0.189	0.342	0.052	0.171	0.120	0.258	**0.002**	0.031	0.156	0.234	0.327	0.408	0.132	0.236
LLM4TS	GPT4TS [12]	0.077	0.214	0.189	0.341	0.052	0.171	0.120	0.256	**0.002**	0.031	0.185	0.286	0.362	0.429	0.141	0.247
	TimeLLM [13]	0.077	0.215	0.199	0.352	0.053	0.173	0.122	0.261	0.003	0.036	0.186	0.271	0.365	0.413	0.144	0.246
MetaTST		**0.069**	**0.203**	**0.182**	**0.335**	**0.051**	**0.170**	**0.118**	**0.254**	**0.002**	**0.029**	**0.146**	**0.227**	**0.308**	**0.402**	**0.125**	**0.231**

表 6　多数据集联合训练下的时间序列短期预测结果. 符号 ↑ 和 ↓ 分别表示联合训练带来了正面或负面效果.

Table 6　Short-term forecasting results under multi-dataset joint training. Symbols ↑ and ↓ indicate the positive and negative effects brought by joint training, respectively.

Models		NP		PJM		BE		FR		DE		Avg.	
Scenarios		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Individual	PatchTST [6]	0.263	0.278	0.095	0.206	0.399	0.259	0.415	0.222	0.462	0.429	0.327	0.279
	iTransformer [7]	0.415	0.391	0.163	0.273	0.560	0.368	0.530	0.298	0.656	0.538	0.465	0.374
	TimeXer [25]	0.275	0.289	0.090	0.217	0.408	0.259	0.424	0.225	0.465	0.432	0.332	0.284
	MetaTST	0.244	0.273	0.101	0.200	0.377	0.246	0.405	0.220	0.446	0.419	0.315	0.272
Joint	PatchTST [6]	0.256↑	0.273↑	0.088↑	0.190↑	0.342↑	0.240↑	0.360↑	0.194↑	0.466↓	0.430↓	0.302↑	0.265↑
	iTransformer [7]	0.376↑	0.377↑	0.154↑	0.260↑	0.516↑	0.337↑	0.531↓	0.298	0.661↓	0.550↓	0.448↑	0.364↑
	TimeXer [25]	0.262↑	0.276↑	**0.085**↑	**0.181**↑	0.358↑	0.242↑	0.384↑	0.196↑	0.464↑	0.430↑	0.311↑	0.265↑
	MetaTST	**0.234**↑	**0.263**↑	0.087↑	0.186↑	**0.318**↑	**0.234**↑	**0.329**↑	**0.193**↑	**0.435**↑	**0.415**↑	**0.281**↑	**0.258**↑
Our promotion (%)		4.1	3.7	13.9	7.0	15.6	4.9	18.8	12.3	2.5	1.0	10.8	4.9

表 7　多数据集联合训练下的时间序列长期预测结果.

Table 7　Long-term forecasting under multi-dataset joint training.

Models		ETTh1		ETTh2		ETTm1		ETTm2		Weather		Traffic		ECL		Avg.	
Scenarios		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Individual	PatchTST [6]	0.077	0.214	0.201	0.353	0.054	0.173	**0.120**	**0.257**	0.002	0.031	0.191	0.283	0.382	0.443	0.147	0.250
	iTransformer [7]	0.079	0.217	0.204	0.357	0.055	0.179	0.131	0.275	0.002	0.032	0.270	0.365	0.444	0.495	0.169	0.274
	TimeXer [25]	0.078	0.216	0.197	0.353	0.055	0.174	0.121	0.258	0.002	0.031	0.190	0.281	0.368	0.438	0.144	0.250
	MetaTST	0.078	0.215	0.202	0.353	0.054	0.174	0.130	0.267	**0.002**	**0.030**	0.180	0.267	0.343	0.422	0.141	0.247
Joint	PatchTST [6]	0.078↓	0.216↓	0.203↓	0.351↑	**0.052**↑	**0.171**↑	0.127↓	0.263↓	0.002	0.031↓	0.218↓	0.302↓	0.366↑	0.437↑	0.149↓	0.253↓
	iTransformer [7]	0.083↓	0.224↓	0.223↓	0.376↓	0.055	0.179	0.139↓	0.286↓	0.002	0.032	0.541↓	0.562↓	0.599↓	0.588↓	0.235↓	0.321↓
	TimeXer [25]	0.078	0.215↑	0.199↓	0.350↓	0.053↑	0.173↑	0.128↓	0.266↓	**0.002**	**0.030**↑	0.198↓	0.288↓	0.390↓	0.452↓	0.150↓	0.253↓
	MetaTST	**0.077**↑	**0.213**↑	**0.196**↑	**0.349**↑	**0.052**↑	**0.171**↑	0.124↑	0.262↑	**0.002**	**0.030**	**0.171**↑	**0.261**↑	**0.332**↑	**0.414**↑	**0.136**↑	**0.243**↑
Our promotion (%)		1.3	0.9	3.0	1.1	3.7	1.7	4.6	1.9	–	–	5.0	2.3	3.2	1.9	3.5	1.6

性, 时序变化比较单一, 所以在单数据集独立训练的场景下, 仅包含序列信息自身就能够取得较高的预测准确性. 尽管如此, 结合元数据建模的 MetaTST 在所有数据集上的整体平均预测性能仍优于所有基线方法, 凸显了元数据建模的通用性和有效性.

长期预测. 本文在表 5 [1, 6, 7, 12, 13, 16, 25, 27, 43, 48] 所示的基准数据集上评估 MetaTST 的长期预测效果. 与所有基线方法相比, MetaTST 在包含 4 种预测长度的综合评价指标上实现了一致的最优效果. 与 TimeXer [25] 相比, MetaTST 降低了 4.9% 的平均均方误差; 与基于大型语言模型的时间序列预测基线方法 GPT4TS [12] 相比, 降低了 11.3% 的平均均方误差. 这表明 MetaTST 能够有效地从自然语言形式的元数据中捕获有价值的额外语义信息, 进行富信息时间序列预测, 进一步提高时间序列预测的准确性.

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

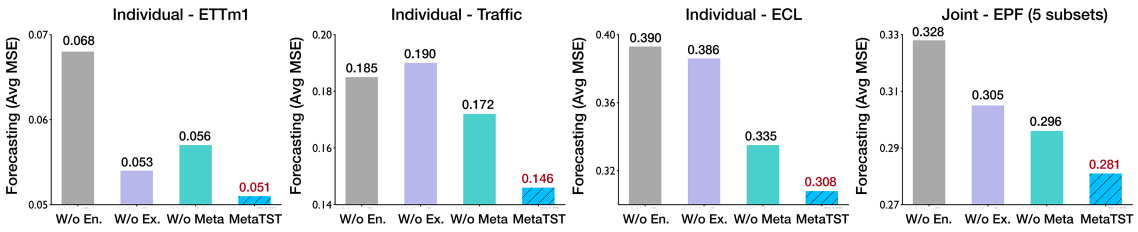


图 4　(网络版彩图) MetaTST 的消融研究, 涉及富信息嵌入的不同输入信息类型, 涵盖了长、短期预测任务中的单数据集独立训练 (individual-) 和多数据集联合训练 (joint-) 策略.

Figure 4　(Color online) Ablation studies of MetaTST with various types informative forecasting, covering individual and joint training strategy in long- and short-term forecasting tasks.

4.2　多数据集联合训练

在经典单数据集独立训练的基础上, 本文进一步探索了一种多数据集联合训练的模型训练方法, 通过联合来自不同领域的多个数据集中的数据来统一训练时间序列模型. 不同数据集的混合训练不仅提高了数据规模, 还提供了更为多样化的时序信息, 并引入了更复杂的时序变化模式, 这为时序模型处理复杂多样的预测场景带来了巨大挑战. 需要注意的是, 正如本文在实现细节中所述, 为了训练一个统一的时间序列预测模型以处理不同的时序预测任务, 我们需要对模型参数进行统一. 由于未进行超参调优, 这导致表 6 和 7 [6,7,25] 中部分统一参数下基于单数据集训练的结果未能超过表 4 和 5 中的结果. 然而, 联合训练相对于独立训练的相对误差降低量 $(1-\frac{\text{联合训练误差}}{\text{独立训练误差}})$ 可以作为评估模型泛化能力的有效指标.

短期预测. 本文所选择的 5 个短期预测数据集均采集自电价预测领域, 具有相似的数据特点和一致的外生序列数量. 因此, 本文通过混合 5 个短期预测数据集的全部数据来训练一个统一的预测模型, 并直接在每个数据集上评估其零样本电价预测性能. 结果如表 6 所示, 我们观察到来自相似领域的多数据集联合训练显著提升了包含 MetaTST 在内的多个模型的性能. 此外, 在这一实验设置下, MetaTST 优于所有基线模型, 实现了卓越的统一建模能力, 甚至超过了表 4 中所示的非统一参数的模型超参数调优后的预测结果. 此结果表明, 元数据的引入具有显著优势, 它能够通过在预测过程中显示融入并理解特定的上下文语义信息, 大幅提升 MetaTST 对相似领域不同数据集间可共享的时序变化模式的理解, 从而增强模型在多样化预测场景中的适应性与泛化性.

长期预测. 由于长期预测数据集来自不同的领域, 且具有不一致的外生序列数目、不匹配的采样频率与周期和截然不同的序列含义, 即使完成联合训练之后也很难直接进行零样本泛化. 因此, 本文参考文献 [54] 中的设计, 基于 7 个长期预测数据集的全部数据混合训练一个统一的模型后, 随后采用线性探测 (linear probing) 方法分别微调到不同的数据集, 以验证模型学习可泛化时序表征的能力. 如表 7 所示, 所对比的基线方法的线性探测结果均不如单数据集全量数据训练的结果, 这是由于不同领域的数据集之间的变化差异会对模型的统一训练提出巨大挑战, 特别是在不同的数据集之间呈现出相互矛盾的时间变化模式时. 相比之下, 本文提出的 MetaTST 在元数据的增强下, 即使面对差异化较大的多数据集联合训练场景也能从中受益, 实现整体最先进的预测性能.

4.3　消融实验

预测性能分析. 本文开展了全面的消融实验以验证 MetaTST 中各种设计的有效性, 具体涵盖了富信息嵌入中的内生序列 (En.)、外生序列 (Ex.) 以及元数据 (Meta). 如图 4 中的结果所示, 3 种类型的输入信息均对 MetaTST 最终的预测结果有所助益, 但是内生序列在大部分预测任务中是最关键的影响因素. 这是因为若缺少作为目标序列的内生序列信息, 将会导致关键的历史信息缺失, 进而使预测性能大幅下降. 此外, 在包含大量外生序列的数据集 (如 Traffic 和 ECL) 中, 内生序列与外生序列之间的相关性也起着至关重要的作用, 它为获取准确的预测结果提供了有价值的额外信息. 而在外生序列数量有限的数据集 (如 ETTm1 和 EPF) 中, 结合元数据能够显著提升预测性能.

本文进一步对不同层次的元数据进行更详细的消融实验以验证其有效性, 结果如表 8 所示. 可以发现, 在多

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 8　多数据集联合训练下不同层次元数据的消融实验.

Table 8　Ablations on different levels of metadata under multi-dataset joint training.

Design	NP		PJM		BE		FR		DE		Avg.	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Only dataset-level metadata	0.239	0.266	0.089	0.190	0.324	0.239	0.341	0.190	0.448	0.423	0.288	0.262
Only task-level metadata	0.239	0.266	0.090	0.190	0.333	0.240	0.356	0.193	0.460	0.420	0.296	0.262
Only sample-level metadata	0.291	0.313	0.105	0.209	0.406	0.279	0.432	0.237	0.524	0.473	0.352	0.302
Multi-levels metadata (ours)	**0.234**	**0.263**	**0.087**	**0.186**	**0.318**	**0.234**	**0.329**	0.193	**0.435**	**0.415**	**0.281**	**0.258**

表 9　多数据集联合训练下修改元数据模板的消融实验.

Table 9　Ablations on different metadata templates under multi-dataset joint training.

Design	NP		PJM		BE		FR		DE		Avg.	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Remove auxiliary words	0.331	0.250	0.473	0.441	0.353	0.209	0.258	0.284	0.093	0.193	0.302	0.275
Shuffle sentences	**0.315**	**0.232**	0.438	0.416	0.331	**0.193**	0.236	0.264	**0.086**	0.187	**0.281**	**0.258**
Original template (ours)	0.318	0.234	**0.435**	**0.415**	**0.329**	**0.193**	**0.234**	**0.263**	0.087	**0.186**	**0.281**	**0.258**

表 10　预测确定性的分析实验.

Table 10　Analysis of predictive certainty.

Quantile loss	Endogenous	Endogenous and exogenous	Endogenous, exogenous and metadata
Q90	0.05207	0.05086	0.05103
Q10	0.04022	0.03963	0.04032
Interval of difference	0.01185	0.01123	**0.01071**

数据集联合训练设置中, 数据集级别的元数据提供了更具区分性的上下文信息, 进而对最终的预测效果起到了关键作用. 其他类型的元数据则贡献了额外的有价值信息, 进一步提升了模型整体的预测性能. 这也验证了我们设计的 3 种元数据对最终效果的正向作用.

元数据模板分析. 为探究修改元数据模板对模型预测性能的影响, 我们设计了分析实验, 重点对比以下两种模板修改方案的效果: (1) 打乱元数据模板中的句子顺序 (shuffle sentences); (2) 仅保留关键元数据, 移除模板中用于构建通顺自然语言描述的其他辅助性词语 (remove cohesive words). 实验结果如表 9 所示, 作为以时间序列为主导的预测模型, MetaTST 对元数据模板中句子顺序打乱这类细微变化并不敏感, 因此未表现出明显的预测性能差异. 但是, 当使用的元数据仅包含关键信息但缺乏辅助性词语衔接时, 模型的预测性能会出现明显下降. 这一结果表明, 大语言模型需要以符合自然语言习惯的流畅元数据描述作为提示词 (prompt), 才能充分发挥其语言建模优势.

预测确定性分析. 为了验证不同层级元数据信息的效果, 我们使用分位数损失函数 (quantile loss) 对 MetaTST 进行了预测确定性的定量分析. 具体地, 我们将分位数损失函数的参数分别设定为 $\tau = 0.9$ (Q90) 和 $\tau = 0.1$ (Q10), 这两个设定可以分别获得模型预测的分布中数值大小处于 90% 和 10% 位置处的结果. 在复杂的多数据集联合训练场景下, 我们依次在 MetaTST 中引入内生序列、外生序列以及元数据信息使用对应的分位数损失函数进行训练, 并计算所得预测模型在电价预测 FR 测试集上 Q90 和 Q10 之间的平均数值差异, 进而评估模型预测的确定性. 通常较小的 Q90 和 Q10 差距意味着更高的预测确定性. 结果如表 10 所示, 我们发现随着逐步引入外生序列和元数据, 模型的预测确定性逐步提升. 这为图 1 中所展示的概念性说明提供了明确的实验支持.

4.4　深入理解基于大语言模型的元数据编码器

为什么选择大语言模型编码元数据?　基于大语言模型的元数据编码 (LLM encoding) 提供了一种灵活的方

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

表 11　基于 LLMs 的元数据编码与独热编码和可学习编码的结果对比.

Table 11　Compared LLMs-based metadata encoding with one-hot and learnable encoding.

Design	NP		PJM		BE		FR		DE		Avg.	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
One-hot encoding	0.244	0.268	0.089	0.189	0.323	0.236	0.337	0.195	0.447	0.419	0.288	0.261
Learnable encoding	0.242	0.268	0.090	0.190	0.328	0.237	0.345	0.199	0.458	0.422	0.293	0.263
LLM encoding (ours)	**0.234**	**0.263**	**0.087**	**0.186**	**0.318**	**0.234**	**0.329**	**0.193**	**0.435**	**0.415**	**0.281**	**0.258**

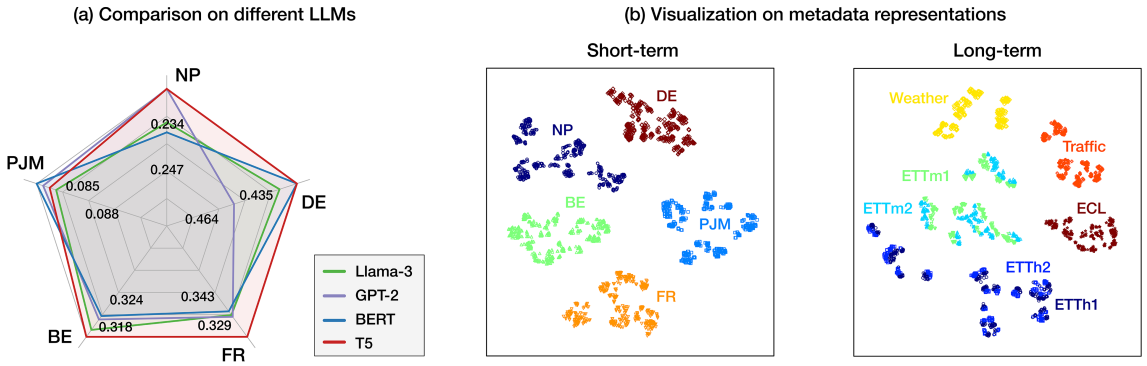


图 5　(网络版彩图) (a) 不同大语言模型作为元数据编码器的性能比较; (b) 联合训练设置下, MetaTST 在短期 (左) 和长期 (右) 预测任务下学习到的元数据表征可视化.

Figure 5　(Color online) (a) Performance comparison on different LLMs as the metadata encoder and (b) representation visualization of metadata on short-term (left) and long-term (right) joint training settings.

式来编码特定上下文的文本级元数据, 使其能够高度适应多样化的时间序列分析场景. 与常见的独热编码 (one-hot encoding) 和可学习编码 (learnable encoding) 不同, 基于大语言模型的元数据嵌入能够更灵活地编码有价值的先验知识, 并且可以充分利用大语言模型强大的语义表示能力, 从而实现比传统方法更为准确的预测. 表 11 展示了数据集级别的独热编码、可学习编码以及 MetaTST 所提出的基于大语言模型的元数据编码之间的预测性能对比.

Encoder 或 Decoder 架构下的大语言模型效果. 我们也分析了不同架构和规模的大语言模型在作为 MetaTST 的元数据编码器的效果. 如图 5(a) 所示, 采用不同的大语言模型, MetaTST 均能够取得良好的最终预测效果, 进一步凸显了其设计方法的通用性和鲁棒性. 值得注意的是, 相比于基于解码器 (decoder) 架构的大语言模型 (如 LLaMA-3, GPT-2 [2]), MetaTST 更倾向于使用基于编码器 (encoder) 训练的大语言模型 (如 BERT [33] 和 T5 [34]) 来编码元数据. 这可能是因为基于自编码形式训练的大语言模型在提取文本表征方面表现更佳, 而基于解码器训练的自回归式大语言模型则更适合直接应用于序列生成任务. 这一发现也为后续大语言模型在时序领域的应用提供了一些启发.

大语言模型驱动的元数据表征学习. 如图 5(b) 所示, 本文使用 t-SNE [55] 对不同预测任务设计的元数据进行了可视化表示. 具体来说, 我们对元数据在联合训练设置下短期和长期任务的测试集表示分布进行了定性分析. 结果表明, 不同数据集的元数据表示具有明显的可区分性, 这表明模型成功学习到了不同数据集的领域特性. 此外, 我们观察到来自相似数据集 (如 ETTh1 和 ETTh2) 的元数据表示相比于差异性更大的数据集 (如 ETTh1, Traffic, Weather) 在表征空间中具有更明显的聚类关系. 这可以归因于合理的元数据设计和大语言模型强大的表示学习能力, 导致更为相似的元数据 (如相同的领域、相似频率与周期等) 被构建后, 能够引入有价值的额外先验知识.

词级令牌序列聚合. 在基于大语言模型的元数据编码器的设计中, 我们探索了多种不同的表示聚合方法, 以将不同层次的元数据令牌有效地融入时间序列预测过程中, 包括图 3 中描述的特殊令牌 (special token)、平均池

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

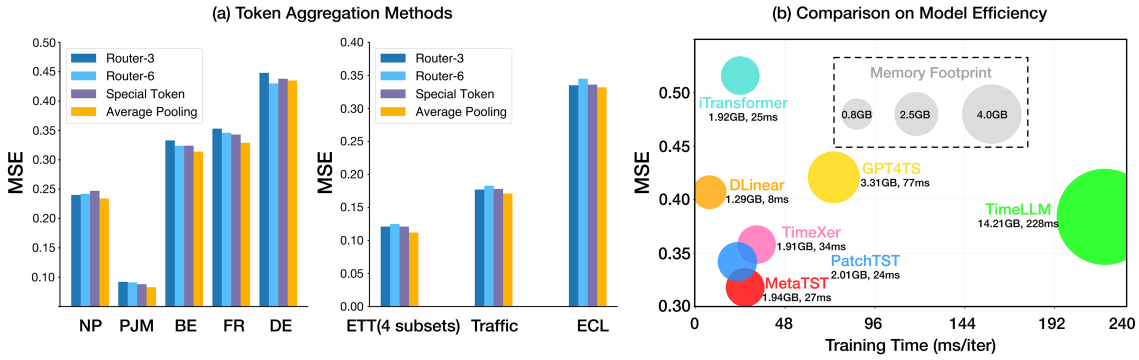


图 6　(网络版彩图) (a) 元数据词级令牌序列聚合方法有效性对比; (b) 不同模型效率比较.

Figure 6　(Color online) Comparison on (a) token aggregation methods and (b) model efficiency.

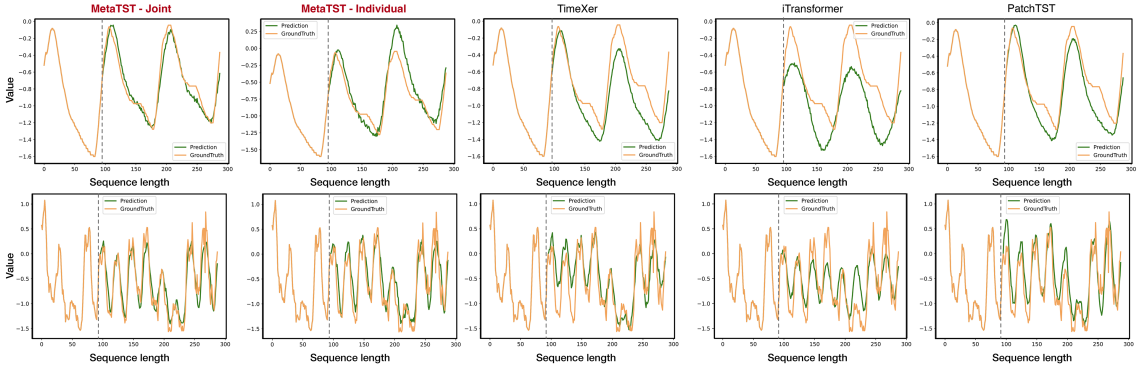


图 7　(网络版彩图) 时间序列预测案例对比, 上半部分来自于 **ETTm2** 数据集, 下半部分来自于 **ECL** 数据集.

Figure 7　(Color online) Case comparison. The upper and bottom are from ETTm2 and ECL, respectively.

化 (average pooling) 及路由机制 (router) 等. 对于路由机制, 我们尝试设置了不同数量的路由令牌进行效果对比, 数量分别为 $\{3, 6\}$. 预测均方误差 (MSE) 结果如图 6(a) 所示, 在长、短期时序预测任务中, 简单直接的平均池化方法表现更加出色. 因此, MetaTST 采用平均池化作为默认的元数据表示聚合方法, 以高效地将词级别元数据令牌转换为全局级元数据表示.

模型效率分析. 本文对 MetaTST 与其他先进的基线模型进行了全面的效率对比. 具体而言, 对于原生的时间序列模型和我们提出的 MetaTST, 我们采用多数据集联合训练设置, 并统一模型超参数进行训练. 对于基于大语言模型的时间序列分析模型, 我们使用 6 层的 GPT-2 作为模型的主干网络进行效率对比. 最终的效率分析结果如图 6(b) 所示, 在相同的模型配置下, MetaTST 在综合效率和性能指标上优于其他基线模型. 尽管部分基于大语言模型的时间序列分析模型引入了复杂的微调方法, 但其训练和推理成本仍然不可忽视. 相比之下, MetaTST 的新颖之处在于将完整的、冻结参数的大语言模型作为独立的元数据编码器使用, 这不仅降低了计算成本, 还达到了更优的预测效果.

4.5　案例分析

模型预测案例. 为了直观地对比不同模型的预测效果, 本文对在输入 96 个历史时间步、预测未来 192 个时间步的实验设置下, 不同模型在 ETTm2 和 ECL 数据集上的长期预测表现进行了可视化对比. 由于不同数据集中外生序列的数量和类型不一致, 我们仅可视化了不同模型对内生序列的预测表现. 如图 7 所示, 无论是单数据集独立训练还是多数据集联合训练, MetaTST 的预测准确性都明显优于其他先进的时间序列预测模型. 特别是在多数据集联合训练下, 得益于更大规模且更多样化的训练数据以及元数据对不同预测场景动态变化的敏锐感知, MetaTST 达到了最优的预测效果.

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

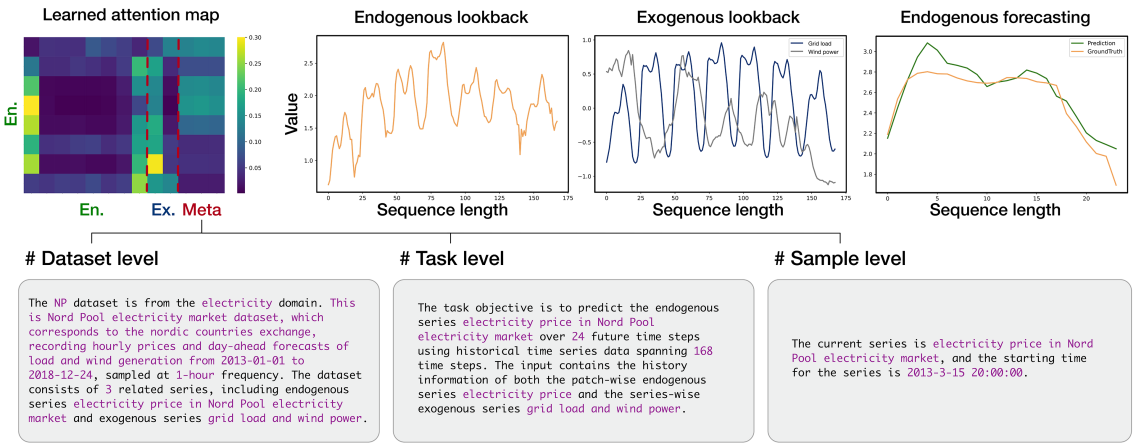


图 8　(网络版彩图) 对 NP 数据集中内生序列 (En.)、外生序列 (Ex.)、元数据 (Meta) 以及 MetaTST 中学习到的注意力分布图进行可视化. 注意力分布图通过对所有注意力头的注意力矩阵进行平均计算得出.

Figure 8　(Color online) Visualization of raw endogenous series (En.), exogenous series (Ex.), language-based metadata (Meta) from the NP dataset, and the learned attention maps in MetaTST. Attention map is calculated by averaging the attention matrices over all the heads and across all the layers.

富信息时间序列预测案例. 我们对 MetaTST 中学习到的片段级内生序列、序列级外生序列以及多层次的元数据所构成的富信息嵌入间的注意力分布图进行了可视化分析. 如图 8 所示, 3 种不同类型的嵌入信息在注意力分布图中呈现出不同的重要性, 这表明不同类型的元信息对预测结果的贡献程度各不相同. 受益于 Transformer 的注意力机制, MetaTST 能够有效利用不同类型的输入信息进行信息选择和表征融合, 为不同的片段级内生序列表征分配具有区分性的注意力, 以强化预测性能.

5　结论与展望

本文提出了一种名为 MetaTST 的元数据引导的富信息时间序列预测模型. 该模型利用预设定的解析模板, 将非结构化的元数据转化为自然语言形式的元数据描述, 并使用大语言模型作为元数据编码器来提取多层次的预测场景信息, 最终通过 Transformer 融合片段级编码的内生序列、序列级编码的外生序列以及多层次的元信息, 实现富信息时间序列预测. MetaTST 在短期和长期时间序列预测任务中均优于其他基线方法. 特别地, 在更具挑战的大规模、多数据集联合训练场景下, MetaTST 表现出强大的场景适应与泛化能力, 为大规模数据训练的时间序列基础模型提供了研究基础. 在未来的研究工作中, 可以深入挖掘时序数据背后的物理、经济等内在机理知识, 并以元数据嵌入的形式与深度模型进行融合, 强化时间序列的表征学习, 以进一步提升时序预测任务的准确性和泛化能力.

参考文献

1　Wu H, Xu J, Wang J, et al. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 22419–22430
2　Wang Y, Wu H, Dong J, et al. Deep time series models: a comprehensive survey and benchmark. 2024. ArXiv:2407.13278
3　Li Z X, Liu H Y. Combining global and sequential patterns for multivariate time series forecasting. Chinese J Comput, 2023, 46: 70–84 [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. 计算机学报, 2023, 46: 70–84]
4　Zhao H K, Wu L K, Li Z, et al. Predicting the dynamics in internet finance based on deep neural network structure. J Comput Res Develop, 2019, 56: 1621–1631 [赵洪科, 吴李康, 李徵, 等. 基于深度神经网络结构的互联网金融市场动态预测. 计算机研究与发展, 2019, 56: 1621–1631]
5　Salinas D, Flunkert V, Gasthaus J, et al. DeepAR: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast, 2020, 36: 1181–1191

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

6 Nie Y, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: long-term forecasting with transformers. In: Proceedings of International Conference on Learning Representations, 2023
7 Liu Y, Hu T, Zhang H, et al. Itransformer: inverted transformers are effective for time series forecasting. In: Proceedings of International Conference on Learning Representations, 2023
8 Dong J, Wu H, Wang Y, et al. TimeSiam: a pre-training framework for Siamese time-series modeling. In: Proceedings of International Conference on Machine Learning, 2024. 11412–11436
9 Zhou H, Zhang S, Peng J, et al. Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 11106–11115
10 Lim B, Arık S Ö, Loeff N, et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast, 2021, 37: 1748–1764
11 Olivares K G, Challu C, Marcjasz G, et al. Neural basis expansion analysis with exogenous variables: forecasting electricity prices with NBEATSx. Int J Forecast, 2023, 39: 884–900
12 Zhou T, Niu P, Sun L, et al. One fits all: power general time series analysis by pretrained LM. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 43322–43355
13 Jin M, Wang S, Ma L, et al. Time-LLM: time series forecasting by reprogramming large language models. In: Proceedings of International Conference on Learning Representations, 2024
14 Zhou F Y, Jin L P, Dong J. Review of convolutional neural network. Chinese J Comput, 2017, 40: 1229–1251 [周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. 计算机学报, 2017, 40: 1229–1251]
15 Liu M, Zeng A, Chen M, et al. Scinet: time series modeling and forecasting with sample convolution and interaction. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 5816–5828
16 Wu H, Hu T, Liu Y, et al. Timesnet: temporal 2D-variation modeling for general time series analysis. In: Proceedings of International Conference on Learning Representations, 2023
17 Liu J W, Wang Y F, Luo X L. Research and development on deep memory network. Chinese J Comput, 2021, 44: 1549–1589 [刘建伟, 王园方, 罗雄麟. 深度记忆网络研究进展. 计算机学报, 2021, 44: 1549–1589]
18 Zhao Z, Chen W, Wu X, et al. LSTM network: a deep learning approach for short-term traffic forecast. IET Intell Trans Sys, 2017, 11: 68–75
19 Lai G, Chang W C, Yang Y, et al. Modeling long-and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018. 95–104
20 Wang Y, Smola A, Maddix D, et al. Deep factors for forecasting. In: Proceedings of International Conference on Machine Learning, 2019. 6607–6617
21 Oreshkin B N, Carpov D, Chapados N, et al. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In: Proceedings of International Conference on Learning Representations, 2020
22 Das A, Kong W, Leach A, et al. Long-term forecasting with tide: time-series dense encoder. 2023. ArXiv:2304.08424
23 Wang S, Wu H, Shi X, et al. Timemixer: decomposable multiscale mixing for time series forecasting. In: Proceedings of International Conference on Learning Representations, 2024
24 Wen Q, Zhou T, Zhang C, et al. Transformers in time series: a survey. 2022. ArXiv:2202.07125
25 Wang Y, Wu H, Dong J, et al. Timexer: empowering transformers for time series forecasting with exogenous variables. In: Proceedings of Advances in Neural Information Processing Systems, 2024
26 Das A, Kong W, Sen R, et al. A decoder-only foundation model for time-series forecasting. In: Proceedings of International Conference on Machine Learning, 2024. 10148–10167
27 Liu Y, Zhang H, Li C, et al. Timer: generative pre-trained transformers are large time series models. In: Proceedings of International Conference on Machine Learning, 2024. 32369–32399
28 Cao D, Jia F, Arik S O, et al. Tempo: prompt-based generative pre-trained transformer for time series forecasting. In: Proceedings of International Conference on Learning Representations, 2024
29 Woo G, Liu C, Kumar A, et al. Unified training of universal time series forecasting transformers. In: Proceedings of International Conference on Machine Learning, 2024. 53140–53164
30 Gao S, Koker T, Queen O, et al. UniTS: a unified multi-task time series model. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 140589–140631
31 Liu X, Hu J, Li Y, et al. Unitime: a language-empowered unified model for cross-domain time series forecasting. In: Proceedings of the ACM on Web Conference, 2024. 4095–4106
32 Chen X T, Ye J J, Zu C, et al. Robustness of GPT large language models on natural language processing tasks. J Comput Res Develop, 2024, 61: 1128–1142 [陈炫婷, 叶俊杰, 祖璨, 等. GPT 系列大语言模型在自然语言处理任务中的鲁棒性. 计算机研究与发展, 2024, 61: 1128–1142]
33 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In:

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019. 4171–4186
34 Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res, 2020, 21: 1–67
35 Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. OpenAI blog, 2019, 1: 9
36 Gao T, Fisch A, Chen D. Making pre-trained language models better few-shot learners. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2020. 3816–3830
37 Touvron H, Lavril T, Izacard G, et al. LLaMA: open and efficient foundation language models. 2023. ArXiv:2302.13971
38 Jin M, Wen Q, Liang Y, et al. Large models for time series and spatio-temporal data: a survey and outlook. 2023. ArXiv:2310.10196
39 Jia F, Wang K, Zheng Y, et al. GPT4MTS: prompt-based large language model for multimodal time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 23343–23351
40 Chang C, Peng W C, Chen T F. LLM4TS: two-stage fine-tuning for time-series forecasting with pre-trained LLMs. 2023. ArXiv:2308.08469
41 Liu Y, Qin G, Huang X, et al. Autotimes: autoregressive time series forecasters via large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 122154–122184
42 Tan M, Merrill M A, Gupta V, et al. Are language models actually useful for time series forecasting? In: Proceedings of Advances in Neural Information Processing Systems, 2024. 60162–60191
43 Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting? In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2023. 11121–11128
44 Riley J. Understanding Metadata. Washington DC: National Information Standards Organization, 2017. 7–10
45 Gonzalez-Perez C, Gonzalez-Perez C. Metainformation. In: Information Modelling for Archaeology and Anthropology: Software Engineering Principles for Cultural Heritage. Berlin: Speinger, 2018. 181–189
46 Li T A, Huang X D, Wang J M, et al. The design of Apache IoTDB distributed framework. Sci Sin Inform, 2020, 50: 621–636 [李天安, 黄向东, 王建民, 等. Apache IoTDB 的分布式框架设计. 中国科学: 信息科学, 2020, 50: 621–636]
47 Liu X Y, Song S X, Huang X D, et al. Short time series group compression and merging method in Apache TsFile. J Softw, 2024, 36: 941–961 [刘星宇, 宋韶旭, 黄向东, 等. Apache TsFile 中的短时间序列分组压缩及合并方法. 软件学报, 2024, 36: 941–961]
48 Zhang Y, Yan J. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: Proceedings of International Conference on Learning Representations, 2023
49 Dong J, Wu H, Zhang H, et al. SimMTM: a simple pre-training framework for masked time-series modeling. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 29996–30025
50 Lago J, Marcjasz G, de Schutter B, et al. Forecasting day-ahead electricity prices: a review of state-of-the-art algorithms, best practices and an open-access benchmark. Appl Energy, 2021, 293: 116983
51 Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library. In: Proceedings of Advances in Neural Information Processing Systems, 2019
52 Diederik P K. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980
53 Kim T, Kim J, Tae Y, et al. Reversible instance normalization for accurate time-series forecasting against distribution shift. In: Proceedings of International Conference on Learning Representations, 2021
54 Goswami M, Szafer K, Choudhry A, et al. MOMENT: a family of open time-series foundation models. In: Proceedings of International Conference on Machine Learning, 2024. 16115–16152
55 van der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res, 2008, 9: 2579–2605

https://www.sciengine.com/doi/10.1360/SSI-2024-0397

Metadata-informed transformers for informative time series forecasting

Jiaxiang DONG[1,2†], Haixu WU[1,2*†], Yuxuan WANG[1,2†], Li ZHANG[1,2],
Jianmin WANG[1,2] & Mingsheng LONG[1,2*]

1. *School of Software, Tsinghua University, Beijing 100084, China*
2. *Beijing National Research Center for Information Science and Technology, Beijing 100084, China*
* Corresponding author. E-mail: wuhx23@mails.tsinghua.edu.cn, mingsheng@tsinghua.edu.cn
† Equal contribution

Abstract Time series forecasting is widely employed in real-world applications such as financial analysis and energy planning. Previous studies have mainly focused on the time series modality to capture the intricate variations and dependencies inherent in time series. Apart from numerical time series data, metadata (e.g., dataset and variate descriptions) convey valuable information, which is essential for forecasting; this information can be used to identify application scenarios and provide more interpretable knowledge than digit sequences. Based on this observation, we propose a metadata-informed time series Transformer (MetaTST), which incorporates multiple levels of context-specific metadata into Transformer forecasting models to enable informative time series forecasting. To tackle the unstructured nature of metadata, MetaTST formalizes them into natural languages bypre-designed templates and leverages large language models (LLMs) to encode these texts into metadata tokens as a supplement to classic series tokens, resulting in an informative embedding. Furthermore, a Transformer encoder is employed to communicate series and metadata tokens, extending series representations using metadata information to improve forecasting accuracy. Additionally, the proposed method allows the model to adaptively learn context-specific patterns across various scenarios; this is particularly effective in handling large-scale, diverse-scenario forecasting tasks. The results show that when applied to widely acknowledged short- and long-term forecasting benchmarks, MetaTST is superior to advanced time series models and LLM-based methods; additionally, it is suitable for both individual single- and joint-training multidataset settings.

Keywords time series forecasting, metadata, deep learning, large language model, Transformer

https://www.sciengine.com/doi/10.1360/SSI-2024-0397