
SparseTSF 分析报告

高景珩^{*1} 丛方昊^{*1} 章昊^{*1}

Abstract

本报告通过对 SparseTSF 论文的阅读与分析，深入探讨了该模型在计算资源受限条件下，实现高效长时间序列预测的解决方案。论文提出的 SparseTSF 模型通过减少参数数量，并有效利用数据的周期性和趋势性特征，显著降低了模型复杂度。实验结果表明，SparseTSF 在多个数据集上表现优异，不仅在预测精度上与先进模型相当，还极大地减少了计算资源的需求。该模型的创新性在于其轻量化设计，为未来的长时间序列预测研究提供了新思路，并在资源受限环境下展示了广阔的应用前景。

1. 小组分工

- **组长：高景珩** 负责详细分析论文中提出的 SparseTSF 模型及其实验设计。
- **组员：丛方昊** 主要负责深入阅读和总结论文的问题定义和研究动机部分。
- **组员：章昊** 负责总结论文，提出问题，以及文档的整合和排版。

2. 论文所解决问题的定义

论文旨在构建一个可以通过给定的数据集对未来数据进行预测的模型，尤其是在数据具有周期性，规模较小时（ < 1000 ），能尽可能优化运行时间。

当数据分布在较长时间跨度上时，模型需要综合分析数据的趋势、周期性和相互依赖性。同时，模型还需选取合适的参数，以减轻“离群点”和“数据缺失”等问题对预测精度的影响。此外，论文还考

虑了计算资源有限的实际情况，提出如何在计算资源受限的条件下，通过较少的参数搭建一个轻量级的长时间序列预测模型，尤其针对具有良好周期性和趋势性的时间序列数据进行合理的预测。这正是本论文所要解决的问题。

3. 论文研究动机的分析

其一，生产生活的需求。时间序列预测在实际生产生活中过于重要，诸多行动需要通过历史数据先进行未来趋势的预测后再考虑执行。

其二，长时间序列预测模型的缺口较大。之前最主要的模型依靠于 Transformer 模型。然而，一方面，Transformer 的自注意力机制在某种程度上会损失信息，特征之间的关系会损失，尤其是当时间跨度较长时，这一缺点会被明显放大，模型的误差积累效应比较显著；另一方面，Transformer 动辄参数百万，而实际的计算资源是有限的，这反过来限制 Transformer 的可行性。我们需要一个更合理的模型构建方式来解决长时间序列预测问题。

其三，潜力巨大的可优化空间。长时间序列预测完善空间很大。实际生活中，常常会出现某些节假日效应、突发网络事件、多周期的趋势合成等等导致结果偏差。长时间序列预测的不确定性很高，变量优化空间很大，是一个可以长期研究的项目。

4. 论文提出方法的分析

首先需要明确的是，并非所有的时间序列都可以进行预测，更不要说是长时间序列预测。本论文的方法的适用对象是可以呈现出周期性和趋势性的原始序列。

论文中提出了一种稀疏预测技术，它关注到并充分

利用研究对象的周期性，并采样分解为子序列，从而将任务由对原始序列的预测转变为对各个周期内的子序列的预测。这为摆脱长时间段的复杂性、减轻周期性的影响、集中算力攻克趋势预测提供了可能。

为了降低参数数量，文中采用单个线性层进行建模。这在事实上是可行的，但结构简单带来的脆弱性是一定的。这里需要原始数据的强相关性和严格的标准把控，大幅度的减小了可应用范围。

具体方法：对给定周期 w 的序列，数据集长度为 L ，分为 w 个长为 $\frac{L}{w}$ 的子序列，用一个**共享参数**的模型预测，即将 $n \times w$ 的矩阵转置为 $w \times n$ ，在 $n \times m$ 的线性层作用之后变为 $w \times m$ 的矩阵，最后转为 $m \times w$ 的矩阵。预测之后，得到 w 组长为 $\frac{H}{w}$ 的子序列，拼接回长为 H 的完整预测序列。

由于数据的使用量少，存在**信息丢失**的问题是必然的，规模变小后，随之而来的就是异常点的放大。理论上，频率变小，噪声就会放大，而且实际的例子中噪声未必是白噪声，因此这个问题是无法避免的。尽管如此，论文仍然提供了优化的方式——引入**滑动聚合方法**，来让问题不那么致命。也就是采用规范化的策略，在采样之前用一个一维卷积实现每个周期数据的加权平均，再进行预测。这样一来，就包括了多个数据，并且减轻了异常值的影响。

这些过程组合起来，就是**跨周期的稀疏滑动预测**。

5. 论文贡献的分析

其一，思想贡献。本论文为长时间序列预测提供了一个新的视角——对原始序列进行分解和变换，从而摆脱长时间的周期性的影响，减少模型复杂度，集中有限算力重点攻坚趋势预测，这种思路对后续长时间序列预测模型建构的发展会起到一定程度的帮助。

其二，技术贡献。本论文搭建的模型将所需参数限制在 1000 以内，在大幅度降低计算资源需求的同时（居于首位），保持了位列前茅的正确率。这在工程实现上，极大降低了开销。

其三，应用贡献。SparseTSF 模型特别适用于计算资源有限的环境，如物联网设备和边缘计算场景。通过使用较少的参数下实现高效的长期预测，论文为实际应用提供了可行的解决方案，推动了高效预测模型在资源受限条件下的应用。此外，模型的轻量化设计也为实时预测系统的开发提供了有价值的参考。

6. 论文实验设计的分析

6.1. 目的与核心思路

- **全面评估 SparseTSF 模型：**测试在长短期记忆网络（LTSF）任务中的性能表现，涵盖预测准确性、效率优势以及稀疏技术的有效性等多个方面。通过与其他主流模型对比，凸显 SparseTSF 模型的优势和特点，为其在实际应用中的可行性和适用性提供依据。
- **核心思路：**以多个具有日常周期性的主流 LTSF 数据集为基础，对 SparseTSF 模型及其他相关模型进行实验。通过比较不同模型在这些数据集上的各项性能指标，深入分析 SparseTSF 模型的性能表现及其稀疏技术的作用机制。同时，考虑模型的效率指标以及泛化能力，以全面评估模型的综合性能。

6.2. 实验具体过程

6.2.1. 实验设置

- **数据集** 论文作者精心挑选了 ETTh1&ETTh2、Electricity、Traffic 这四个主流的 LTSF 数据集进行实验，这些数据集涵盖了不同的领域和数据特征，具有广泛的代表性。ETTh1&ETTh2 数据集可能来自能源领域的相关监测数据，Electricity 数据集直接与电力消耗相关，Traffic 数据集则聚焦于交通流量信息。这种多样化的数据集选择有助于全面评估 SparseTSF 模型在不同实际场景中的性能表现，使研究结果更具说服力和通用性。同时，这些数据集都具有日常周期性的特点，这与 SparseTSF 模型利用数据周期性的核心理念紧密相关，能够更好地

验证模型在处理此类数据时的优势和有效性。

- **基线模型** 将 SparseTSF 模型与众多先进和代表性的基线模型进行比较是一种非常严谨和全面的实验设计方法。涵盖了基于 Transformer 架构的多种模型以及其他主流架构的模型，包括 Informer、Autoformer、Pyraformer、FEDformer 等，还特别选取了轻量级模型 DLinear 和 FITS 进行对比。这种广泛的对比能够清晰地展示 SparseTSF 模型在不同模型架构中的位置和优势。通过与各种模型在不同数据集和预测任务上的性能比较，可以更准确地评估 SparseTSF 模型的创新性和实用性。例如，与基于 Transformer 架构的模型对比，可以看出 SparseTSF 在处理长短期记忆网络任务时是否具有独特的优势；与轻量级模型对比，则能突出其在模型轻量化方面的突破。
- **评估指标** 对模型的效率评估从多个维度展开，包括可训练参数数量、MACs（乘加运算次数）、最大内存使用和单轮训练时间等指标，这种全面的评估方式能够深入了解模型的计算复杂度和资源需求。
- **实验公平性** 首先进行了模型的训练和测试，在不同数据集上使用相同的实验设置（如优化器、学习率调整策略等）对所有模型进行训练和测试，以保证实验的公平性。这种公平的实验设置确保了实验结果可靠性。

6.2.2. 实验结果

- **预测性能对比** 对比了 SparseTSF 模型与基线模型在预测性能（MSE）方面的差异，结果表明 SparseTSF 模型在所有场景中都排名前二，在参数规模显著更小的情况下达到或接近先进水平。这一结果充分证明了 SparseTSF 模型在预测性能方面的优势，即使在与优秀的模型相比时，也能够保持出色的表现。
- **效率优势分析** 通过比较可训练参数数量、MACs、最大内存使用和单轮训练时间等指标，发现 SparseTSF 模型在这些方面都显著优于

其他主流模型，与现有的轻量级模型相当。这表明 SparseTSF 模型不仅在预测性能上表现出色，而且在计算效率方面也具有明显的优势，能够在计算资源有限的设备上高效运行。

- **消融研究分析** 消融研究是深入理解模型各个组成部分贡献的重要手段。通过将稀疏技术与不同的基础模型（如线性模型、Transformer 模型和 GRU 模型）相结合，并分析其性能提升效果，能够清晰地揭示稀疏技术的有效性。例如，看到稀疏技术对 Transformer 模型性能提升高达 21.4%，这充分证明了该技术在不同的模型架构中的通用性和重要性。对超参数 w 的研究也很有意义，它表明了模型对数据周期性的敏感性以及合理设置超参数的重要性。此外，对模型泛化能力的研究展示了 SparseTSF 模型在不同数据集上的适应性，这对于模型在实际应用中的推广至关重要，因为实际数据往往具有多样性和复杂性，良好的泛化能力能够保证模型在不同场景下的性能稳定性。

7. 论文阅读过程中遇到的问题及解决方法

- **语言问题** 部分内容不理解含义，英语专业名词较晦涩难懂。
解决方法：使用腾讯翻译、知云文献进行论文的翻译工作以辅助阅读。
- **背景知识缺乏** 在阅读时发现缺少必要的背景知识，例如与研究相关的基本理论（预测的原理，滑动聚合等）或相关概念（采样，消融研究等）。
解决方法：回顾相关的基础知识，参考教科书或网络课程进行学习，以及与 AI 名词解释。
- **整体梳理困难** 论文结构比较庞大，某些部分之间的逻辑联系模糊。
解决方法：在阅读过程中，做笔记并总结各部分的核心思想，使用思维导图工具整理论文的结构，帮助理清各部分之间的关系。
- **对结果理解不足** 论文中结果的解释部分难以理解，无法准确把握研究的贡献，模棱两可。
解决方法：重新审视结果部分，结合图表与数

据进行分析，同时查阅相关理论，形成自己的理解。另外，我们也积极进行讨论，集思广益。

8. 论文阅读过程中遇到的具体问题分析

8.1. 为什么 SparseTSF 可以用较少的参数预测较准确的结果？

8.1.1. 独特的预测技术

- **交叉周期稀疏预测技术** SparseTSF 模型采用交叉周期稀疏预测技术，将原始序列按固定周期下采样为子序列。例如，对于具有明显日周期的数据，按日周期下采样。这样做可以聚焦于跨周期的趋势预测，减少了需要处理的数据量和模型的复杂度。在这个过程中，模型不需要像一些传统方法那样对整个原始序列进行复杂的处理来捕捉特征，而是通过对相对简单的子序列进行操作，从而可以用较少的参数来实现有效的预测。
- **简化的任务形式** 通过下采样和将预测任务转化为交叉周期趋势预测任务，模型可以更清晰地分离数据的周期性和趋势。模型能够将主要精力放在预测趋势变化上，而周期性特征可以通过下采样和子序列的处理方式更高效地被提取和利用，不需要大量的参数来同时处理周期性和趋势这两个复杂的因素。

8.1.2. 有效的特征提取策略

- **对周期性的利用** 很多时间序列数据本身具有内在的周期性，如交通流量的日周期和周周期、电力消耗的日周期等。SparseTSF 模型基于这种周期性进行下采样，能够准确地提取出周期性特征。例如，在实验中可以看到下采样后的子序列能够保留趋势特征，同时周期性在原始序列和下采样序列中呈现出可利用的规律，使得模型可以用较少的参数来描述这些特征，而不是像一些模型需要大量参数去学习周期性模式。
- **线性层的应用** SparseTSF 模型在框架内使用单一线性层来建模 LTSF 任务。线性层可以在一

定程度上捕捉数据的线性关系，对于时间序列数据中存在的趋势等线性特征有较好的拟合能力。它不像一些复杂的神经网络结构需要大量的参数来构建多层的非线性映射，而是通过简单有效的线性映射来处理数据，从而减少了参数数量，同时也能达到较好的预测效果。

8.1.3. 合理的辅助策略

- **滑动聚合操作** 在进行稀疏预测之前，对原始序列进行滑动聚合操作。这个操作可以让每个聚合的数据点包含其周围周期内其他点的信息。一方面解决了下采样过程中只利用每个周期一个数据点进行预测导致的信息丢失问题，另一方面也通过加权平均的方式缓解了异常值的影响。这使得模型可以更稳定地利用数据进行预测，不需要过多的参数来处理数据的不确定性和异常情况。
- **实例归一化** 实例归一化策略通过在模型输入前减去序列的均值，在输出后加上均值的方式，帮助模型应对数据分布在训练和测试集之间的变化。这种归一化方式可以使模型更稳定地学习数据的特征，不需要大量的参数来适应不同的数据分布情况，从而提高了模型的性能和预测准确性。

8.2. 如何让 SparseTSF 模型具有更好的实用性？

8.2.1. 模型结构改进

- **处理超长周期数据** 对于超长周期数据，目前模型可能因稀疏参数连接过于稀疏而性能不佳。可以考虑设计一种自适应的下采样策略，根据周期的长度动态调整下采样的方式和参数。例如，当周期非常长时，不是简单地按照固定的大周期下采样，可以尝试先按较小的子周期下采样，然后再逐步整合这些子周期的信息，这样可以避免参数连接过于稀疏的问题，提高模型对超长周期数据的适用性。
- **处理多周期数据** 针对多周期数据，模型目前只能处理一个主要周期。可以引入多尺度分析的

方法，同时对多个周期进行建模。比如，对于既有日周期又有周周期的数据，可以设计不同的分支或模块分别处理不同的周期，然后将这些不同周期的处理结果进行融合，使模型能够充分捕捉多周期数据的特征，提高对这类数据的适用性。

8.2.2. 超参数优化

- **周期参数 w 的自适应选择** 超参数 w 目前需要手动设置，且对模型性能有重要影响。可以研究一种自适应确定 w 的方法，根据数据的特征自动选择合适的周期值。例如，可以通过分析数据的自相关函数等统计特征，找到数据中最显著的周期，然后将其作为 w 的值。这样可以使模型在不同数据集上更自动地适应其周期性特征，提高适用性。
- **其他超参数的调整** 除了 w 之外，模型可能还有其他超参数可以进一步优化。例如，滑动聚合操作中的核大小等参数，可以根据数据的噪声水平和周期性复杂程度进行调整。如果数据噪声较大且周期性复杂，可以适当增大核大小，以更好地整合信息和处理异常值，从而提高模型对不同情况的适用性。

8.2.3. 数据预处理增强

- **异常值处理** 进一步完善数据预处理过程中的异常值处理方法。除了滑动聚合操作中的加权平均方式外，可以结合其他异常值检测和处理技术。比如，使用基于统计的异常值检测方法，先识别出明显的异常值，然后采用合适的替换或修正策略，使数据更加干净，提高模型对包含异常值情况的适用性。
- **数据增强技术** 可以应用数据增强技术来增加数据的多样性，提高模型的泛化能力。例如，对于时间序列数据，可以采用时间序列数据增强的方法，如添加噪声、进行时间平移、缩放等操作，使模型能够学习到更多不同情况的数据特征，从而更好地适应实际应用中的各种情况。

8.2.4. 融合其他技术

- **与其他模型结构融合** 考虑将 SparseTSF 模型与其他有效的时序预测模型结构进行融合。比如，可以将其与基于 Transformer 的一些模型结构进行融合，取各自的优势。Transformer 结构在处理长序列依赖关系上有优势，融合后可以提高 SparseTSF 模型对长序列和复杂依赖关系情况的适用性。
- **结合领域知识** 根据具体的应用领域，结合领域知识对模型进行改进。例如，在能源消耗预测领域，可以结合能源消耗的物理规律和相关因素（如天气、季节等）对模型进行调整。通过将领域知识融入模型，可以使模型更好地适应特定领域的各种情况。

9. 使用的 AI 工具

9.1. 辅助翻译

- **使用工具:** ChatGPT, DeepL
- **使用场景:** 由于论文为英文撰写，而我们组员的语言水平对理解复杂的学术语言存在一定障碍，尤其是长句和专业术语难以理解。因此，我们使用了 ChatGPT、DeepL 对论文的关键部分进行翻译，帮助我们准确理解文中的内容。
- **效果:** 这些翻译工具为我们提供了清晰的中文释义，帮助我们扫清了阅读过程中存在的语言障碍，尤其是在分析复杂的句子结构时起到了非常大的帮助。

9.2. 术语解释

- **使用工具:** ChatGPT
- **使用场景:** 在阅读过程中，遇到了许多尚不熟悉的专业术语，如“消融技术（Ablation Study）”、“下采样（Downsample）”、“上采样（Upsample）”等。为了解这些术语的具体含义及其在 SparseTSF 中的应用，我们通过 ChatGPT 进行查询与解释。

- 效果: ChatGPT 不仅对专业术语做了详细解释, 还为我们提供了实际的使用场景和示例代码, 帮助我们在实践中理解这些概念, 并将它们与论文中的方法结合起来。

9.3. 交流思考

- 使用工具: ChatGPT
- 使用场景: 在论文方法部分 (如跨周期稀疏预测、线性变换、滑动聚合等), 我们对具体的技术原理和数学公式理解不够透彻。通过 ChatGPT 的帮助, 我们对这些技术方法的原理进行了细化解读, 并获得了与之相关的例子和类比, 帮助我们更深入地理解论文提出的模型构建思路。
- 效果: ChatGPT 能够快速响应我们提出的问题, 并提供详细的解释, 使我们对论文中的关键技术和模型架构有了更加深入的理解, 也提高了我们分析报告的完成效率。

太棒了! 谢谢你。我突然又有个问题, 论文中提到的“采样”具体是什么意思?

不客气! 很高兴能帮到你。关于你提到的“采样”问题, 这是时间序列处理中的一个核心概念, 我们来详细看看。

什么是“采样”?

“采样 (Sampling)”是指从一组连续数据中按一定规则挑选出一部分数据点, 通常是为了降低数据的分辨率、减少计算量或者捕捉主要趋势。在时间序列预测中, 采样的目的是在保持数据主要特征的前提下, 简化计算过程。

采样的两种形式:

1. 下采样 (Downsampling):
 - 减少数据点, 从原始数据中按照固定的时间间隔提取少量的数据点。这可以帮助减少计算复杂性, 同时保留主要的周期性或趋势特征。
 - 举例: 如果你有一组每秒钟的数据, 但你只关心每分钟的平均值, 你可以对数据进行下采样。比如每60秒取一个数据点。这样, 你就从每秒的数据变成了每分钟一个数据点。
2. 上采样 (Upsampling):
 - 增加数据点, 通过插值或者其他方法在原有数据之间填充新的数据点。上采样常用于提高数据的分辨率或者补充缺失的数据。
 - 举例: 如果你有一组每小时的温度数据, 但你想得到每10分钟的温度变化, 可以通过插值的方法在两个小时温度之间进行推算, 从而生成更高分辨率的数据。

Figure 2. 术语解释

方法部分翻译 (Methodology)

3.1. 预备知识

“长期时间序列预测 (LTSF)”任务涉及使用之前观测到的多变量时间序列 (MTS) 数据来预测未来一段时间的值。公式化表达为 $\hat{x}_{t+1:t+H} = f(x_{t-L+1:t})$, 其中 $x_{t-L+1:t} \in \mathbb{R}^{L \times C}$, $\hat{x}_{t+1:t+H} \in \mathbb{R}^{H \times C}$. 在这个公式中, L 代表历史观测窗口的长度, C 是不同特征或通道的数量, H 是预测时间跨度的长度。LTSF 的主要目标是延长预测时间跨度 H , 因为这能在实际应用中提供更丰富和先进的指导。然而, 延长的预测时间跨度 H 会增加模型的复杂性, 从而导致主流模型中的参数大幅增加。为了解决这一挑战, 我们的研究致力于开发既轻量化又稳健有效的模型。

3.2. SparseTSF

由于需要预测的数据通常展示出先验的周期性 (例如电力消耗和交通流量通常有固定的日周期), 我们提出了“跨周期稀疏预测 (Cross-Period Sparse Forecasting)”技术, 以增强长序列依赖的提取能力, 同时减少模型的参数规模。利用一个单层线性层来处理 LTSF 任务, 这构成了我们的 SparseTSF 模型, 如图 2 所示。

跨周期稀疏预测

假设时间序列 $x_{t-L+1:t}^{(i)}$ 已知周期为 w , 第一步是将原始序列下采样为 w 个子序列, 每个子序列的长度为 $n = \lfloor \frac{L}{w} \rfloor$ 。然后将具有共享参数的模型应用于这些子序列进行预测。预测完成后, w 个子序列中的每个子序列的长度为 $m = \lfloor \frac{H}{w} \rfloor$, 最终再将这些子序列上采样为完整的预测序列, 长度为 H 。

为什么不能是 w , 而是要除2取整再乘2+1呢?

这是一个很好的问题, 涉及到卷积核的设计和其在时间序列数据中的有效性。我们来逐步解释为什么在 SparseTSF 模型中选择卷积核大小使用了公式 $2 \times \lfloor \frac{w}{2} \rfloor + 1$ 。

1. 卷积核的对称性

卷积核通常是对称的, 特别是在处理时间序列和图像数据时。使用对称卷积核有助于保持输入数据的中心特征。例如, 假设我们希望每个卷积操作都能“关注”中心点周围的数据。

2. 中心点的选择

当我们将卷积核大小设置为 w 时, 卷积核会覆盖整个周期。这会导致以下问题:

- 没有中心点: 当卷积核大小是偶数时, 没有明确的中心数据点 (例如, 如果 $w = 24$, 那么卷积核的范围是1到24, 中心无法定义)。这可能导致在卷积计算时难以确保卷积操作对所有输入数据的贡献是均匀的。
- 处理的点不均衡: 如果直接用 w , 在卷积操作时会丢失一部分重要的上下文信息, 特别是在处理边界条件时。

Figure 3. 交流思考

Figure 1. 辅助翻译