

基于注意力机制的销售量预测模型

王伟钊

2025 年 4 月 7 日

摘要

目录

1	绪论	3
1.1	研究背景与意义	3
1.2	研究现状	3
1.2.1	统计学方法	4
1.2.2	机器学习方法	4
1.2.3	深度学习方法	5
1.2.4	Transformer 模型	5
1.3	本文的研究内容与结构安排	6
2	相关工作	7
2.1	问题描述	7
2.2	模型构建	8
2.3	模型特性	10
3	实验结果与分析	11
3.1	数据集介绍	11
3.2	数据处理与实验设置	14
3.3	超参数调优	16
3.4	结果分析	18
3.5	对比实验	19
4	总结与展望	20
4.1	总结	20
4.2	展望	20

1 绪论

1.1 研究背景与意义

近年来,随着电子商务的快速发展和大数据技术的广泛应用,销售量预测成为了一个备受关注的研究领域。在《2024 实体零售未来白皮书》中指出,销售量预测的准确性直接影响到企业的盈利能力和市场竞争力 [26],且各个国家和地区的零售行业也将在总量上持续增长。特别是在全球化和数字化的背景下,消费者行为模式变得更加复杂和多样化,这为销售量的预测带来了不确定性与随机性。为此,基于人工智能和深度学习的预测模型逐渐成为研究热点,其通过对历史数据和多维度特征的深度挖掘,能够提高预测的精度和稳定性。

准确的销售量预测不仅能够帮助企业优化库存管理,降低运营成本,还能为市场营销策略的制定提供科学依据。据《中国新零售白皮书》统计 [28],零售行业因预测误差导致的库存积压损失约占年均营业额的 8%-12%,而采用 AI 预测系统的企业可减少此类损失 30%-40%。此外,销售量预测还能够帮助企业更好地把握市场需求变化,提升供应链的灵活性和响应速度,从而在激烈的市场竞争中占据优势。

除了将销售量预测应用于库存优化与成本控制外,通过对销售量的准确预测,企业还可以更好地进行市场营销策略的制定与价格等动态变量的调整,从而使企业的利润最大化。在实际应用中,销售量预测模型通常需要考虑多种因素,如季节性、促销活动、市场趋势等,这些因素的变化会直接影响到销售量的波动。通过模型的运算,企业可以搭建可持续性供应链,包括对突发事件的响应,以及绿色生产的转型,以减少过剩生产。此外,随着社交媒体和在线评论等新兴数据源的出现,如何将非结构化数据有效地融入到预测模型中,也是当前研究的一个重要方向。

在此背景下,基于深度学习的销售量预测模型逐渐受到关注。深度学习模型能够通过多层次的特征抽取和非线性映射,可以捕捉到数据中的复杂模式和潜在关系。因此,如何结合最新的技术手段,构建高效、精准的销售量预测模型,不仅具有重要的理论意义,还对实际应用具有深远的影响。这一课题的研究将为企业的智能化转型提供有力支持,同时也为学术界探索复杂系统的预测问题提供了新的思路。

1.2 研究现状

当前主流的销售量预测方法主要包括传统的统计学方法、基于机器学习的方法和基于深度学习的方法。传统的统计学方法如时间序列分析、回归分析等,虽然在一定程度上能够捕捉到数据的趋势和季节性,但在处理复杂的非线性关系时往往显得力不从心。而基于机器学习的方法,如支持向量机 (SVM)、随机森林 (RF) 等,虽然在处理高维数据时表现良好,但仍然存在特征选择和模型解释性差的问题。

1.2.1 统计学方法

统计学方法在销售量预测中有着悠久的历史，主要依赖于对历史数据的分析和建模。这些方法通常基于假设数据具有某种特定的分布或模式，通过对历史数据进行拟合来预测未来的销售量。常见的统计学方法包括时间序列分析、回归分析和指数平滑法等。这些方法在处理线性关系和简单的季节性变化时表现良好，但在面对复杂的非线性关系和多维特征时，往往难以取得理想的效果。在实际应用中，销售量预测通常涉及多个变量的交互作用和非线性关系，这使得传统的统计学方法在处理复杂数据时显得力不从心。此外，统计学方法通常需要对数据进行严格的假设检验和模型选择，这在实际操作中可能会导致模型的过拟合或欠拟合。因此，研究者们开始探索更为灵活和高效的机器学习方法，以提高销售量预测的准确性和可靠性。

- **时间序列分析**：时间序列分析是一种经典的统计学方法，主要用于分析随时间变化的数据。常用的方法包括自回归移动平均模型 (ARIMA) [18]、季节性分解等。这些方法能够捕捉到数据的趋势和季节性，但在处理复杂的非线性关系时往往显得力不从心。
- **回归分析**：回归分析是一种用于研究变量之间关系的统计学方法。通过建立回归模型，可以预测因变量与自变量之间的关系 [5]。然而，传统的线性回归模型在处理非线性关系时存在局限性。
- **指数平滑法**：指数平滑法是一种用于时间序列预测的统计方法，通过对历史数据进行加权平均来预测未来值。该方法简单易用，但在处理复杂数据时可能会出现较大的误差。

1.2.2 机器学习方法

机器学习方法在销售量预测中逐渐受到关注，主要是因为其能够处理高维数据和复杂的非线性关系。常见的机器学习方法包括支持向量机 (SVM)、随机森林 (RF)、梯度提升树 (GBDT) 等。这些方法通过对数据进行特征选择和模型训练，能够有效地捕捉到数据中的潜在模式和关系。

- **支持向量机 (SVM)**：Support Vector Machine(SVM) 是一种基于统计学习理论的监督学习方法，主要用于分类和回归问题。其通过构建超平面来最大化数据点之间的间隔，从而实现对数据的分类和预测 [4]。在销售量预测中，SVM 能够有效地处理高维数据和非线性关系，但在大规模数据集上训练时间较长。
- **随机森林 (RF)**：Random Forests(RF) 是一种集成学习方法，通过构建多个决策树来提高预测的准确性和稳定性。其通过对多个决策树的结果进行投票或平均来得到最终的预测结果 [8]。RF 在处理高维数据时表现良好，但模型的可解释性较差。

- **梯度提升树 (GBDT)**: Gradient Boosting Decision Tree(GBDT) 是一种基于决策树的集成学习方法, 通过逐步优化损失函数来提高模型的性能。最新的 GBDT 算法如 XGBoost [2] 和 LightGBM [10] 在处理大规模数据时表现优异。在 2022 年的时候 Nasios 等人通过融合 LightGBM 和深度学习的方法在 M5 竞赛中取得第三名的成绩 [15]。GBDT 能够自动进行特征选择和交互作用建模, 但在处理高维稀疏数据时可能会出现过拟合。

1.2.3 深度学习方法

随着深度学习技术的快速发展, 基于深度学习的方法在销售量预测中逐渐崭露头角。深度学习模型通过多层次的特征抽取和非线性映射, 能够捕捉到数据中的复杂模式和潜在关系。常见的深度学习方法包括循环神经网络 (RNN)、长短时记忆网络 (LSTM)、卷积神经网络 (CNN) 等。这些方法在处理时间序列数据和多维特征时表现良好, 但模型的训练和调优相对复杂。

- **卷积神经网络 (CNN)**: CNN 是一种主要用于图像处理的深度学习模型, 通过卷积操作来提取局部特征 [1]。近年来, CNN 也被应用于时间序列数据的处理, 能够有效地捕捉到数据中的局部模式和特征, 如 TimesNet [21]。
- **循环神经网络 (RNN)**: RNN 是一种适用于序列数据的深度学习模型, 通过循环连接来捕捉时间序列数据中的时序关系 [7]。在销售量预测中, RNN 能够有效地处理时间序列数据, 但在长序列数据上容易出现梯度消失或爆炸的问题。
- **长短时记忆网络 (LSTM)**: LSTM 是一种改进的 RNN, 通过引入门控机制来解决长序列数据中的梯度消失问题 [13]。LSTM 在销售量预测中表现良好, 能够捕捉到长期依赖关系, 但模型的训练时间较长。

1.2.4 Transformer 模型

近年来, 基于 Transformer 模型的时间序列预测方法逐渐成为研究热点。Transformer 模型最初由 Vaswani 等人于 2017 年提出 [19], 其核心是通过注意力机制捕捉数据中的全局依赖关系, 注意力机制的提出给深度学习带来了革命性的突破, 凭借强大的特征提取能力、并行计算优势以及生成式模型的灵活性, Transformer 在自然语言处理和计算机视觉等领域取得了突破性成果。这些特性使其在处理复杂模式和高维数据时表现尤为出色, 为时间序列预测提供了新的解决方案。

随着 Transformer 模型的成功, 其应用逐渐扩展到时间序列预测领域 [27]。2021 年, 阿里巴巴达摩院的研究团队提出了基于 Transformer 的销售量预测模型, 称为 Aliformer [17]。该模型通过引入特征信息的注意力机制, 能够有效捕捉数据中的复杂模式和潜在关系。在多个公开数据集上的实验表明, Aliformer 在预测性能上显著优于传统方法和其他深度学习模型。

2024 年, Google 研究团队进一步提出了一个仅由解码器 Decoder 组成的轻量级模型 [6]。该模型不仅体量小、易于训练, 还在多个时间序列数据集上表现出色, 展示了 Transformer 架构在高效性和准确性上的潜力。同年, Yu-Neng 等人结合大语言模型的优势, 提出了 Large Time Series Models (LTSM 模型) [3]。通过引入 prompt 机制, LTSM 模型显著增强了对时间序列数据的前后特征分析能力, 为复杂时间序列预测提供了新的解决方案。

此外, 2023 年, Zhou 等人提出了一种基于 ChatGPT 的通用模型 [9], 该模型适用于各种生成式和传统机器学习的任务, 在时间序列预测问题上, 该模型通过利用大语言模型的上下文理解能力, 在多个数据集上取得了优异的效果。与此同时, Transformer 的改进模型也层出不穷。例如, Informer [24] 通过稀疏注意力机制提高了计算效率, Autoformer [22] 通过自动化特征分解增强了模型的预测能力, 而 iTransformer [14] 则通过引入多尺度特征提取进一步提升了模型的性能, 更多的还有 Crossformer、Pyraformer、Reformer 等等 [23] [12] [11]。

Transformer 及其改进模型在销售量预测领域展现了强大的潜力和广阔的应用前景。这些模型不仅提高了预测的准确性和稳定性, 还为未来的研究提供了丰富的思路 and 方向。

1.3 本文的研究内容与结构安排

本文主要研究基于 Transformer 模型的销售量预测方法。我们将通过对现有模型的分析与改进, 提出一种新的销售量预测模型, 并在多个公开数据集上进行实验验证。具体研究内容包括:

- 应用一种基于注意力机制的 Transformer 模型 (TimeXer [20]), 实现对 M5 数据集的销售量预测。该模型通过引入多头自注意力机制, 能够有效捕捉数据中的复杂模式和潜在关系。
- 比较新模型与现有模型的性能差异, 并分析其适用性和局限性。
- 探讨未来销售量预测模型的发展方向, 包括如何结合最新的技术手段和数据源, 提高预测的准确性和稳定性。

本文的结构安排如下:

- 第一章为绪论, 介绍研究背景、现状及本文的研究内容与结构安排。
- 第二章为模型构建, 详细描述 TimeXer 模型的销售量预测方法, 包括模型架构、训练策略和评估指标等。
- 第三章为实验结果与分析, 介绍实验数据集、实验设置和实验结果, 并对结果进行深入分析和讨论。
- 第四章为结论与展望, 总结本文的主要贡献, 并提出未来研究的方向和建议。

2 相关工作

本章节主要介绍 TimeXer 模型的核心内容，包括其面对的销售量预测问题的描述、模型的构建细节以及设计思路。首先，我们将详细阐述销售量预测问题的定义和特点，分析其在时间序列预测中的特殊性和挑战性，例如数据的非线性关系、长短期依赖性以及多维特征的交互作用等。接着，重点介绍 TimeXer 模型的核心模块，包括内外生变量嵌入层、内生变量自注意力模块以及外生变量和内生变量的交互注意力模块，解析其在捕捉复杂模式和潜在关系中的作用。特别地，TimeXer 模型通过引入分块表示法和全局标记机制，显著提升了计算效率和全局建模能力，同时通过多头注意力机制有效捕捉内外生变量之间的复杂交互关系。

TimeXer 模型在 2024 年首次由 Wang 等人提出 [20]，并在多个公开数据集（如 ETTH、Weather、Traffic 等）以及一些自定义数据集上取得了优异的效果。实验表明，TimeXer 模型在处理长时间序列和多维特征时表现出色，尤其在捕捉数据中的长期依赖性和短期波动性方面具有显著优势。此外，TimeXer 模型结合内生变量和外生变量对问题进行建模，能够有效捕捉数据中的复杂模式和潜在关系，为销售量预测领域提供了新的解决方案。通过对模型的深入解析，本章节将为后续实验和应用提供理论支持和技术指导。

2.1 问题描述

销售量预测问题可以被视为一个回归问题，其目标是根据历史数据和多维特征，预测未来一段时间内的销售量。具体而言，给定一个时间序列数据集 $X = \{x_1, x_2, \dots, x_t\}$ ，其中 x_i 表示第 i 个时间点的特征向量，我们希望通过模型 F 来预测未来 t 个时间点的销售量 $\{y_{t+1}, y_{t+2}, \dots, y_{t+k}\}$ 。即：

$$\mathbf{y}_{t+1:t+k} = F(\mathbf{x}_{t-k+1:t}) \quad (1)$$

其中 k 表示历史时间窗口的大小， F 为待训练的模型。我们称这些需要预测的目标变量为内源变量 (Endogenous Variables)，它们是模型的主要输出。内源变量通常是时间序列数据中最重要的部分，直接反映了销售量的变化趋势和规律。

除了对时间点相关的销售量本身的预测，我们还需要考虑其他相关特征对销售量的影响，如价格、促销活动、节假日等，我们称这些特征为外源变量 (Exogenous Variables)。这些外源变量我们是可以直接获取的。外源变量通常是时间序列数据中影响销售量的重要因素，它们可以帮助我们更好地理解销售量的变化规律和趋势。

有了内源变量和外源变量，我们的问题模型就变为了

$$\mathbf{y}_{t+1:t+k} = F(\mathbf{x}_{t-k+1:t}, \mathbf{z}_{t-k+1:t+k}) \quad (2)$$

其中 z_i 表示第 i 个时间点的特征向量。

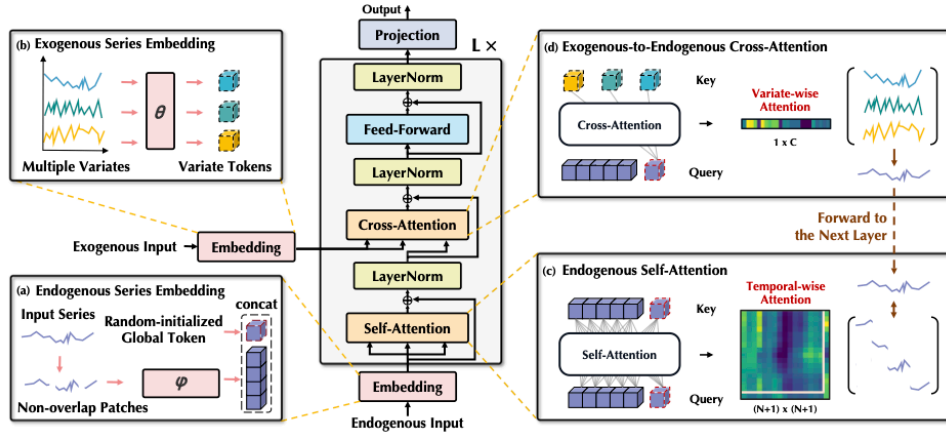


图 1: TimeXer 模型架构 [20]

销售量预测问题除了具有一般时间序列预测问题的特点外，还具有以下几个特殊性：销售量的数据类型为离散型数据，这意味着对于销售量较少的商品，其在原数据集中可能会出现较大的波动，这会使得模型拟合变得困难；销售量的变化通常具有明显的季节性和周期性特征，这使得模型需要能够捕捉到这些长期依赖关系；销售量的变化通常受到多种因素的影响，如价格、促销活动、节假日等，这使得模型需要能够处理多维特征的交互作用，如何有效编码将这些外源变量在交叉注意力中起到作用，是销售量预测模型的一个重要挑战。

2.2 模型构建

如图1所示，TimeXer 模型主要由内生变量嵌入层、外生变量嵌入层和内生变量自注意力以及外生变量和内生变量的交互注意力模块组成。内生变量嵌入层主要用于对内生变量进行编码，外生变量嵌入层主要用于对外生变量进行编码，内生变量自注意力模块主要用于对内生变量进行自注意力计算，寻找内源数据的内在联系，外生变量和内生变量的交互注意力模块主要用于对外生变量和内生变量进行交互注意力计算。通过若干层叠加模型后，模型的输出为预测的销售量。

内生变量嵌入层

现有大多数基于 Transformer 的预测模型将每个时间点或时间片段嵌入为时序标记 (temporal token)，并通过自注意力机制学习时序依赖关系。然而，这种逐点嵌入方式在处理长时间序列时可能导致计算效率低下且难以捕捉全局模式。受 PatchTST 模型 [16] 的启发，TimeXer 采用分块表示法：将内生序列分割为互不重叠的块 (patch)，每个块被投影为一个时序标记，从而有效降低序列长度并提升模型的全局建模能力。

此外，鉴于内生变量与外生变量在预测中的不同作用，TimeXer 采用差异化粒度进行嵌入。为了避免直接合并不同粒度的内生标记与外生标记导致的信息失真问题，我们为每个内生变量引入可学习的全局标记 (global token)，作为宏观表征与外生变量交

互。该设计能够有效桥接外生序列到内生时序块的因果信息，其完整的内生嵌入公式表述如下：

$$\begin{aligned}\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\} &= \text{Patchify}(\mathbf{x}) \\ \mathbf{P}_{en} &= \text{PatchEmbed}(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n) \\ \mathbf{G}_{en} &= \text{Learnable}(\mathbf{x})\end{aligned}\tag{3}$$

式中 P 表示块长度， $N = \lfloor T/P \rfloor$ 代表内生序列分割的块数， \mathbf{s}_i 表示第 i 个块。PatchEmbed(\cdot) 通过可训练的线性投影器，将每个添加了位置嵌入的长度为 P 的块映射为 D 维向量。最终， N 个块级时序标记嵌入 \mathbf{P}_{en} 和 1 个序列级全局标记嵌入 \mathbf{G}_{en} 被输入 Transformer 编码器，从而实现对内生变量的高效建模。

外生变量嵌入层

外生变量嵌入层主要用于对外生变量进行编码。由于外生变量通常是非时序的，因此我们采用了一个简单的线性投影器将每个外生变量映射为 D 维向量。该设计能有效捕捉外生变量与内生变量之间的关系，其完整的外生嵌入公式表述如下：

$$\begin{aligned}\mathbf{G}_{ex} &= \text{Learnable}(\mathbf{z}) \\ \mathbf{P}_{ex} &= \text{Linear}(\mathbf{z})\end{aligned}\tag{4}$$

式中 G_{ex} 表示外生变量的全局标记， P_{ex} 表示外生变量的块级时序标记。Linear(\cdot) 通过可训练的线性投影器，将每个添加了位置嵌入的长度为 P 的块映射为 D 维向量。最终， N 个块级时序标记嵌入 P_{ex} 和 1 个序列级全局标记嵌入 G_{ex} 被输入 Transformer 编码器。

内生变量自注意力模块

内生变量自注意力模块主要用于对内生变量进行自注意力计算，寻找内源数据的内在联系。该模块通过多头自注意力机制，对内生变量进行自注意力计算，得到内生变量的上下文表示。该设计能有效捕捉内生变量的时序依赖关系，其完整的自注意力公式表述如下：

$$\begin{aligned}\mathbf{Q} &= \mathbf{P}_{en} W_Q \\ \mathbf{K} &= \mathbf{P}_{en} W_K \\ \mathbf{V} &= \mathbf{P}_{en} W_V \\ \mathbf{A} &= \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V} \\ \mathbf{A}_{en} &= \text{Concat}(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_h) W_O\end{aligned}\tag{5}$$

式中 W_Q 、 W_K 、 W_V 和 W_O 分别表示查询、键、值和输出的线性投影矩阵， d_k 表示键的维度， \mathbf{A}_i 表示第 i 个头的注意力矩阵， \mathbf{A}_{en} 表示内生变量的上下文表示。

外生变量和内生变量的交互注意力模块

外生变量和内生变量的交互注意力模块主要用于对外生变量和内生变量进行交互注意力计算。该模块通过多头自注意力机制，对外生变量和内生变量进行交互注意力计算，得到外生变量和内生变量的上下文表示。该设计能有效捕捉外生变量与内生变量之间的关系，其完整的交互注意力公式表述如下：

$$\begin{aligned}
\mathbf{Q} &= \mathbf{P}_{en} W_Q \\
\mathbf{K} &= \mathbf{P}_{ex} W_K \\
\mathbf{V} &= \mathbf{P}_{ex} W_V \\
\mathbf{A} &= \text{Softmax} \left(\frac{\mathbf{QK}^T}{\sqrt{d_k}} \right) \mathbf{V} \\
\mathbf{A}_{ex} &= \text{Concat}(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_h) W_O \\
\mathbf{A}_{ex} &= \text{Concat}(\mathbf{A}_{en}, \mathbf{A}_{ex}) W_O
\end{aligned} \tag{6}$$

式中 W_Q 、 W_K 、 W_V 和 W_O 分别表示查询、键、值和输出的线性投影矩阵， d_k 表示键的维度， \mathbf{A}_i 表示第 i 个头的注意力矩阵， \mathbf{A}_{en} 表示内生变量的上下文表示， \mathbf{A}_{ex} 表示外生变量和内生变量的上下文表示。

2.3 模型特性

前人的研究指出，对外生变量的交叉注意力并不需要严格对照内生变量，即使发生外生变量和内生变量的错位，该模型仍然能够保持较好的预测效果。这一现象表明，模型在处理外生变量时具有一定的鲁棒性和自适应能力。从逻辑上来说，外生变量应当与内生变量一一对应，但当错位发生时，模型会自动学习到外生变量与内生变量之间的潜在关系，并在预测时进行动态调整。例如，当某个会显著影响销售量上升的节假日到来时，如果外生变量发生错位并转移到了较前的时间点，模型会通过注意力机制捕捉到该节假日的影响力，并向上调整后几天的销售量预测值，而不是直接修改当天的预测值。

这一特性不仅体现了模型在处理外生变量信息不完整或存在噪声时的鲁棒性，还说明了模型能够灵活地适应外生变量的动态变化。这种鲁棒性使得模型在实际应用中具有更广泛的适用性，尤其是在外生变量数据可能存在缺失、延迟或错误标注的情况下。此外，这一现象也为未来研究提供了新的思路，即如何进一步优化模型的注意力机制，使其能够更高效地捕捉外生变量与内生变量之间的复杂交互关系，从而提升预测的准确性和稳定性。

3 实验结果与分析

3.1 数据集介绍

本次实验使用了 M5 数据集，该数据集由 Kaggle 主办的 M5 竞赛提供¹，包含了来自美国零售商的销售数据。数据集涵盖 30490 个产品在 30 个不同商店中的销售记录，时间跨度为 2011 年 1 月至 2016 年 6 月，共计 1941 天。每个样本包括产品 ID、商店 ID、日期、销售量等信息。

表 1: M5 数据集特征信息

特征名称	描述	类型
Product ID	产品的唯一标识符	分类特征
Store ID	商店的唯一标识符	分类特征
State ID	州的唯一标识符	分类特征
Date	日期信息	时间特征
Price	产品的销售价格	数值特征
Event Name	特殊事件名称（如节日）	分类特征
Event Type	特殊事件类型（如宗教、文化等）	分类特征
Snap Flag	SNAP 计划标记（食品补助计划）	二值特征
Sales	每日销售量	数值特征

如表1所示，M5 数据集具有多维特征，不仅包含销售量数据，还包括价格信息、促销活动、节假日标记等，为模型构建提供了丰富的信息来源。数据具有明显的时间序列特性，包括季节性、趋势性和周期性，为时间序列预测模型的应用提供了良好基础。此外，数据集按照产品、商店、州和国家四个层次组织，支持多层次的销售量预测任务。由于数据量大、特征复杂且存在噪声，M5 数据集对模型的预测能力和泛化能力提出了较高要求。

M5 比赛还给出了具体的评价指标，包括 RMSSE(Root Mean Squared Scaled Error) 和 MASE(Mean Absolute Scaled Error)，其中 RMSSE 是基于销售量的平方根均方误差，MASE 是基于销售量的平均绝对误差。RMSSE 和 MASE 都考虑了数据的季节性和趋势性变化，能够更好地反映模型的预测性能。RMSSE 的计算公式如下⁷：

$$\text{RMSSE} = \sqrt{\frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}} \quad (7)$$

其中 h 为预测的时间步长， y_t 为实际销售量， \hat{y}_t 为预测销售量， n 为训练集的样本数

¹M5 竞赛是 Kaggle 平台上一个著名的时间序列预测竞赛，旨在推动销售量预测领域的发展。具体比赛信息和数据请参见<https://www.kaggle.com/c/m5-forecasting-accuracy>

量。但如果利用 RMSSE 作为模型训练的损失函数，可以发现：

$$\text{RMSSE} = \sqrt{\frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}} = \frac{\sqrt{\sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}}{\sqrt{\frac{h}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}} \quad (8)$$

8中分母的平方根是一个常数，因此我们可以将 RMSSE 简化为：

$$\text{RMSSE} = c \sqrt{\sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2} \quad (9)$$

其中 c 为常数。我们可以发现 RMSSE 的计算公式与均方误差 (MSE) 的计算公式类似，MSE 的计算公式为：

$$\text{MSE} = \frac{1}{h} \sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2 \quad (10)$$

RMSSE 和 MSE 只是缩放的关系。因此使用 RMSSE 与使用 MSE 作为标准化后的模型训练的损失函数是等价的，不论选择哪一种都不会影响模型的训练效果。

数据分布特征

M5 数据集的销售量数据具有明显的时间序列特性，包括长期趋势、季节性变化和周期性波动。例如，某些产品在节假日期间的销售量会显著增加，而在淡季则会下降。销售量数据呈现长尾分布，大多数产品的日销售量较低，仅有少数产品的销售量较高，这种分布特性对模型的预测能力提出了挑战。不同商店和产品的销售量分布不均衡，例如，大型商店的销售量通常高于小型商店，而畅销产品的销售量远高于冷门产品。此外，数据中存在一定的噪声和异常值，例如由于促销活动或库存不足导致的销售量异常波动，这些异常值可能会影响模型的训练效果。销售量与价格、促销活动、节假日等特征之间存在较强的相关性，例如，价格下降或节假日促销通常会导致销售量增加。

M5 数据集是一个数据严重右偏的数据集，数据集中大部分的销售量都集中在 0 附近，且有大量的零值和少量的高值。为了更好地理解数据分布特征，我们对数据进行了可视化处理。我们绘制了部分特征的直方图，如图2所示。图 a 是 M5 数据集的销售量直方图，可以看出数据集中含有大量的零值和少量的高值，且数据分布呈现长尾特征。图 b 是我们再去除零值后绘制的直方图，可以看出数据集的分布要比 a 图更均匀也更平滑，这说明数据集的销售量在去除零值后呈现更均匀的分布特征。但是现实中却实实在在存在着销售量为 0 的情况，通过分析数据集我们发现，除去销售量为 0 的情况外，商品未上架时的销售量也为 0。而未上架的商品的销售量对我们来说是不具有参考价值的，因此我们需要将未上架商品的销售 0 值去除，图 c 是去除未上架商品后的销售量直方图，随仍然存在大量 0 值，但这是更符合现实情况的。

图3为所有商品销售总量随时间变化的趋势图，M5 数据集的销售总量在时间上呈现出明显的季节性和周期性波动。图中画红线的为一年中销售量最少的日子，通过查询

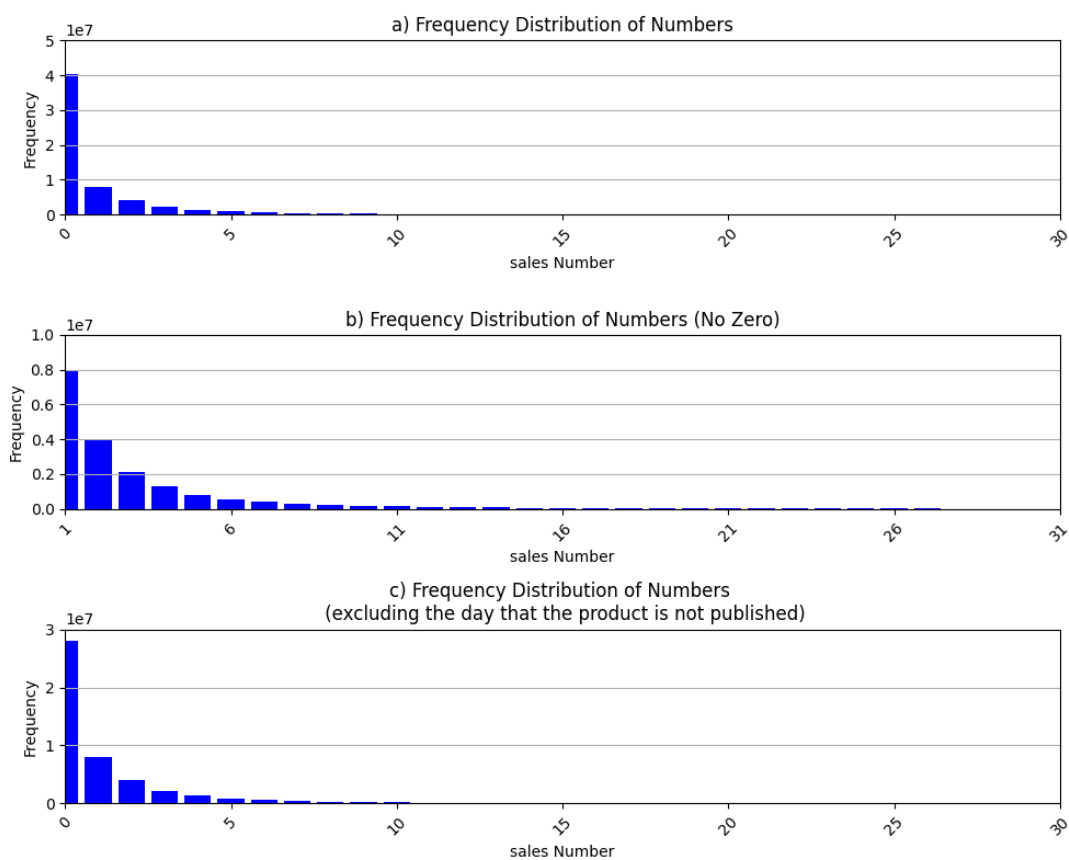


图 2: M5 数据集销售量分布示意图

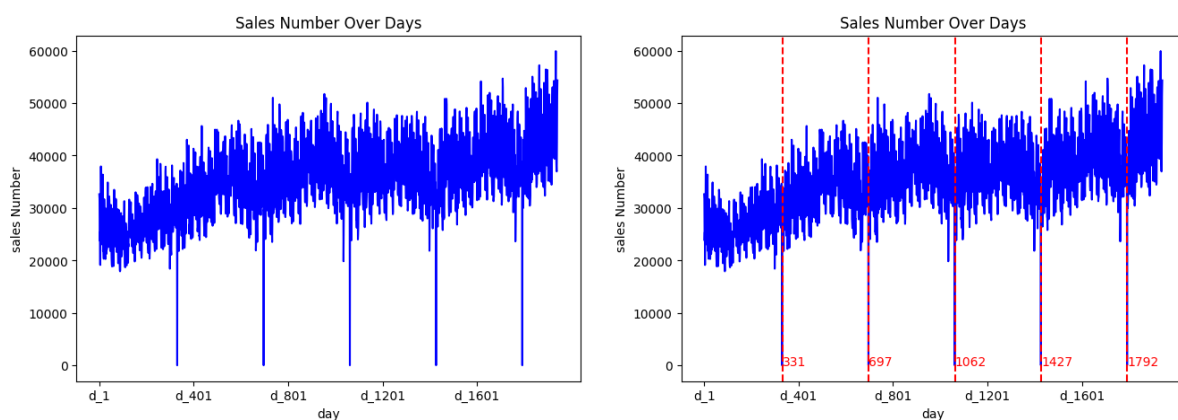


图 3: 所有商品销售总量随时间变化的趋势图

数据集发现，红线日子正好是美国的圣诞节。以圣诞节为分界线，每一年的销售量变化都有呈现先升后降的趋势，且在每年的 6 月和 12 月销售量达到峰值。通过对数据的分析，我们可以发现 M5 数据集的销售量受季节性、节假日和促销活动等因素的影响较大，这为模型的构建提供了重要的参考依据。

此外，M5 数据集的多样性和复杂性为模型的设计和优化提供了丰富的挑战。例如，如何有效地处理数据中的噪声和异常值，如何捕捉不同特征之间的交互作用，以及如何利用时间序列的长期依赖性和短期波动性，都是需要解决的问题。通过对数据的深入分析，我们可以更好地理解数据的特性，从而为模型的构建和优化提供科学依据。

3.2 数据处理与实验设置

我们需要对数据进行预处理，包括数据清洗、特征选择和数据划分等。数据清洗主要包括去除缺失值、异常值和重复值等操作。特征选择主要包括对时间序列数据进行平滑处理、归一化处理和特征工程等操作。数据划分主要包括将数据集划分为训练集、验证集和测试集，以便于模型的训练和评估。

特征选择与编码

根据 M5 数据集提供的信息，我们能够直接获得的特征包括产品 ID、商店 ID、州 ID、日期、价格、事件名称、事件类型和 SNAP 标记等。其中日期和节假日信息是通过字符串形式存储的，因此我们需要将其转换为数值型特征。我们将其转换为了独热向量 (one-hot vector) 形式，以便于模型的训练和评估。由于节假日总数有 30 个，所属类型有 4 种，而一天中可能会有至多 2 个节假日，因此我们将节假日信息转换为 $(30 + 4) \times 2 = 68$ 维的独热向量。SNAP 标记是一个二值特征，表示产品是否参与 SNAP 计划，我们将其转换为 1 和 0 的数值型特征。价格信息是一个数值型特征，我们将其进行归一化处理，以便于模型的训练和评估。最后我们将特征拼接，得到了 $68 + 3 + 1 = 72$ 维的矩阵作为模型的输入特征。

数据划分

由于比赛中的数据是要预测 1941-1969 天的销售量，而我们无法获得 1941-1969 天的销售量，因此我们将已有数据集中提前 28 天，将 1-1913 天的数据作为训练集，1914-1941 天的数据作为测试集。其中训练集我们取最后的 $NumOfVali = 361$ 天作为验证集，前面的数据作为训练集。这样一来，我们便将数据集划分为训练集、验证集和测试集。由于原数据集中含有 30490 个产品，而每个产品的销售量数据都是独立的，因此我们需要将每个产品单独划分。

在训练过程中，我们采用滑动窗口的方式来生成训练样本。具体来说，我们将每个产品的销售量数据划分为多个时间窗口，每个时间窗口的长度为 seq_len ，每个时间窗口的起始位置为 $index$ 为正整数。每个时间窗口包含 seq_len 天的销售量数据和对应

的外生变量数据。我们将每个时间窗口的数据作为一个训练样本，输入到模型中进行训练。这样一来，我们便可以通过滑动窗口的方式来生成多个训练样本，从而提高模型的训练效率。

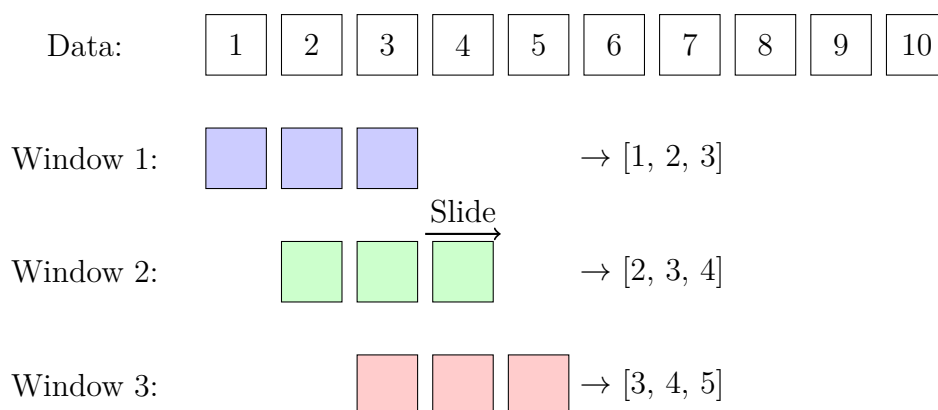


图 4: 滑动窗口示意图

最后我们通过滑动窗口的方法，可以将一个商品 1941 天的数据划分为 $1941 - seq_len + 1$ 个窗口，每个窗口包含 seq_len 天的数据，假设取得的 seq_len 为 192 天，那么每个商品可以划分为 $1941 - 192 + 1 = 1750$ 个窗口。取其中的 1 个为测试集，0.8 为训练集，0.2 为验证集，如表2所示。

表 2: 数据划分示意图

数据集	划分个数
训练集	1333
验证集	334
测试集	1

数据选择

由于 M5 的数据集过于庞大，本次实验选取其中部分数据进行实验。我们选择其中一个商品的销售量数据进行实验，商品 ID 为 “HOBBIES_1_008” 在 8 个商店 “CA_1、CA_3、TX_1、TX_2、TX_3、WI_1、WI_2、WI_3” 中的销售量数据。这是由于该商品的销售量数据较为完整，且销售量水平较高，能够更好地反映模型的性能。我们将该商品在 8 个商店中的销售量数据进行拼接，最后得到了如下3的数据集。

表 3: 数据划分示意图

数据集	划分个数
训练集	10664
验证集	2672
测试集	8

3.3 超参数调优

输入序列窗口大小 (seq_len)

输入序列窗口大小是指模型在进行预测时所使用的历史数据长度。窗口大小的选择对模型的性能有着重要影响。过小的窗口大小可能导致模型无法捕捉到数据中的长期依赖关系，而过大的窗口大小则可能导致模型的训练时间增加和过拟合现象。因此，我们需要通过实验来确定一个合适的窗口大小。在本实验中，我们尝试了不同的窗口大小，包括 30 天、50 天和 100 天、200 天、192 天、500 天，并通过验证集上的预测性能来评估其效果。实验结果表明，窗口大小为 192 天时，模型在验证集上的表现最佳。

表 4: 不同窗口大小下模型性能对比

窗口大小	MSE	MAE
30	0.6695	0.5127
50	0.6743	0.5128
100	0.6567	0.5203
192	0.6557	0.5121
200	0.6662	0.5253
500	0.7747	0.5382

隐藏层大小 (d_model)

隐藏层大小决定了模型中隐含层的神经元数量，这一参数直接影响模型的表达能力和训练效率。如果隐藏层过小，模型可能无法充分学习数据中的复杂特征；反之，隐藏层过大则可能导致训练时间过长，并增加过拟合的风险。为了找到最佳的隐藏层大小，我们在实验中尝试了 64、128、256、512 和 1024 个神经元的不同配置。实验结果表明，隐藏层大小为 128 时，模型在验证集上的表现最佳，既能够有效捕捉数据中的复杂特征，又避免了过大的计算开销。此外，我们还观察到，随着隐藏层大小的增加，模型的训练时间显著增长，但验证集上的性能提升逐渐减小甚至出现下降，这表明隐藏层大小的选择需要在模型复杂度和性能之间进行权衡。实验结果表明，隐藏层大小为 512 时，模型在验证集上的表现最佳。

表 5: 不同隐藏层大小下模型性能对比

隐藏层大小	MSE	MAE
64	0.6596	0.5141
128	0.6590	0.5121
256	0.6714	0.5172
512	0.6557	0.5121
1024	0.6780	0.5264

块大小 (*batch_size*)

块大小是指在每次迭代中用于训练模型的样本数量。块大小的选择对模型的训练速度和性能有着重要影响。过小的块大小可能导致模型训练时间过长，而过大的块大小则可能导致模型无法充分学习数据中的复杂特征。在本实验中，我们尝试了不同的块大小，包括 3、4、5 和 10，并通过验证集上的预测性能来评估其效果。实验结果表明，块大小为 10 时，模型在训练速度和性能之间达到了较好的平衡。

表 6: 不同块大小下模型性能对比

块大小	MSE	MAE
3	0.6564	0.5182
4	0.6499	0.5205
5	0.6550	0.5149
10	0.6557	0.5121

通过以上实验，我们确定了模型的超参数设置如下：

表 7: 模型超参数设置

超参数名称	设置值
输入序列窗口大小 (<i>seq_len</i>)	192
隐藏层大小 (<i>d_model</i>)	512
块大小 (<i>batch_size</i>)	10
学习率 (<i>learning_rate</i>)	0.001
优化器	Adam
丢弃率 (<i>dropout</i>)	0.1
训练轮数 (<i>epochs</i>)	3

3.4 结果分析

从实验结果中来看, TimeXer 能够有效预测某一种商品在某一商店的销量, 如图5所示。图中展示了 CA_1 商店的预测结果, 横坐标为日期, 纵坐标为销量。从图中可以看出, 虽然销售量的具体值预测不够精确, 但模型已经能预测出部分曲线的变化趋势。图中橙色的线为预测的销售量, 蓝色的线为实际的销售量。可以看出, 模型在大部分时间段内都能较好地捕捉到实际销售量的变化趋势, 尤其是在节假日和促销活动期间, 模型能够较好地预测到销售量的激增。

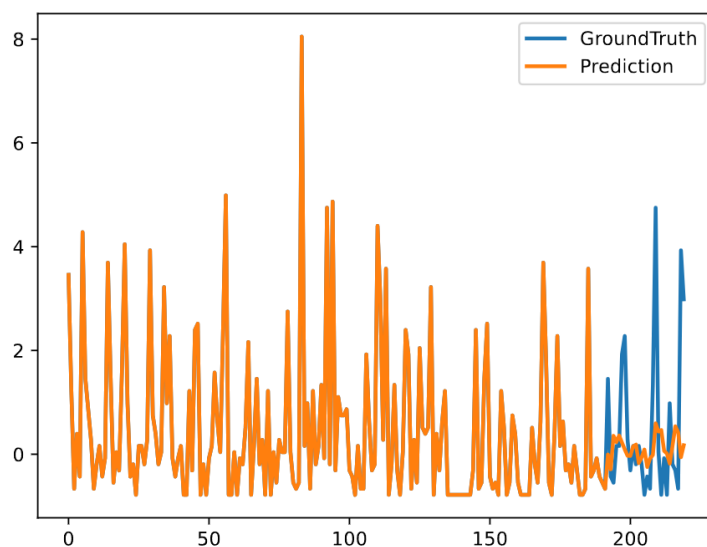


图 5: CA_1 的预测结果

下面给出的是 8 个商店的预测结果, 如图6所示。

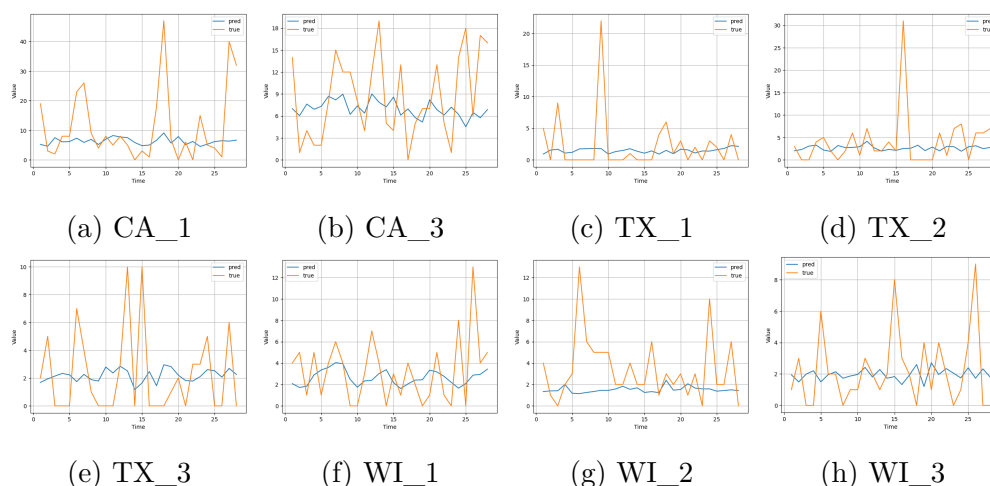


图 6: 8 个商店 HOBBIES_1_008 的预测结果

3.5 对比实验

本次实验对比了 TimeXer 模型与其他几种主流的时间序列预测模型，包括 LSTM、GRU 和 ARIMA 等。我们在相同的数据集上进行了实验，并使用相同的评价指标进行比较。实验结果如表8所示。

表 8: TimeXer 与其他模型的对比实验结果

模型	MSE	MAE
Autoformer	0.6778	0.5345
DLinear	0.6729	0.5302
Crossformer	0.6947	0.5293
PatchTST	0.6672	0.5179
TimeXer	0.6557	0.5121

注 *：表中数据为模型在测试集的预测结果，加粗的为最佳结果。

从表8的对比实验结果可以看出，TimeXer 模型在 MSE 和 MAE 两个指标上均取得了最优的表现，分别为 0.6557 和 0.5121，优于其他对比模型。这表明 TimeXer 模型在销售量预测任务中具有较强的预测能力和稳定性。

相比于传统的时间序列预测模型（如 Autoformer 和 Crossformer），TimeXer 模型通过引入内外生变量的交互注意力机制，能够更好地捕捉数据中的复杂模式和潜在关系，从而提升了预测精度。此外，与 PatchTST 模型相比，TimeXer 模型在处理长时间序列和多维特征时表现出更强的鲁棒性，这可能得益于其全局标记机制和分块表示法的设计。

实验结果还表明，虽然其他模型（如 DLinear 和 PatchTST）在某些场景下也能取得较好的效果，但在整体性能上仍然略逊于 TimeXer 模型。这进一步验证了 TimeXer 模型在销售量预测任务中的优势。

综上所述，TimeXer 模型在多个评价指标上均优于现有方法，尤其在处理复杂时间序列数据时表现出色。

4 总结与展望

4.1 总结

本文围绕基于 Transformer 模型的销售量预测问题展开研究，提出了一种新的预测模型 TimeXer，并在 M5 数据集上进行了实验验证。本文的主要贡献包括以下几个方面：

首先，系统梳理了销售量预测领域的研究现状，分析了传统统计学方法、机器学习方法和深度学习方法的优缺点，特别是 Transformer 模型在时间序列预测中的应用潜力及其局限性。通过对现有研究的总结，明确了销售量预测领域的技术难点和发展方向，为模型设计提供了理论支持。

其次，提出了 TimeXer 模型，通过内外生变量嵌入层和交互注意力模块，捕捉数据中的复杂模式和潜在关系。TimeXer 模型的设计充分考虑了时间序列数据的特性，能够在长时间序列和多维特征的情况下保持较高的预测精度和稳定性。

通过在 M5 数据集上的实验表明，TimeXer 模型在预测精度和稳定性上优于现有方法，尤其在处理长时间序列和多维特征时表现出色。通过超参数调优和对比实验，验证了模型的有效性和鲁棒性。实验结果显示，TimeXer 模型在多个评价指标上均取得了显著提升，为销售量预测领域提供了新的解决方案。

此外，本文深入分析了 M5 数据集的特性，为模型构建提供了科学依据。通过对数据分布、时间序列特性和多维特征的分析，揭示了销售量预测问题的复杂性和挑战性，并提出了相应的解决方案。数据分析结果不仅为 TimeXer 模型的设计提供了参考，也为未来研究提供了重要启示。

综上所述，本文的研究不仅丰富了销售量预测领域的理论内容，也为理论实践提供了实验支持。TimeXer 模型为复杂时间序列预测问题提供了新的解决方案，其优异性能为未来研究奠定了基础。

4.2 展望

本文提出的 TimeXer 模型在销售量预测任务中取得了良好的效果，但仍然存在一些问题不足之处。未来研究可以从以下几个方面展开：

1. 优化模型结构与训练策略：引入更高效的注意力机制（如稀疏注意力）或先进的训练策略（如自监督学习），以降低计算复杂度并提升泛化能力 [25]。未来的研究可以尝试结合动态注意力机制，根据输入数据的特性动态调整模型的计算资源分配，从而进一步提高模型的效率和性能。

2. 扩展应用场景：将 TimeXer 模型应用于其他时间序列预测任务，如金融、能源和医疗领域，以及应用于完整的 M5 数据或结合强化学习等技术解决更复杂的问题。例如，在金融领域，TimeXer 模型可以用于股票价格预测；在能源领域，可以用于电力负荷预测；在医疗领域，可以用于患者病情变化的预测。

3. 融合多模态数据：探索将非结构化数据（如文本、图像）与结构化数据结合，构建多模态预测模型，提高预测精度。未来的研究可以尝试将社交媒体数据、新闻文本和图像数据等多模态信息融入到销售量预测模型中，从而更全面地捕捉影响销售量的多维因素。

4. 增强模型可解释性：通过注意力可视化或特征重要性分析，提升模型的可解释性，为实际决策提供支持。未来的研究可以尝试开发基于注意力机制的可视化工具，帮助用户理解模型的预测过程和决策依据，从而提高模型在实际应用中的可信度和可用性。

5. 关注数据隐私与公平性：研究联邦学习或差分隐私技术，确保数据隐私安全，同时构建公平无偏的预测模型。在未来的研究中，可以尝试将联邦学习技术与 TimeXer 模型相结合，实现跨组织的数据共享与协作，同时保护用户隐私。此外，还可以研究如何在模型训练过程中引入公平性约束，确保模型的预测结果不受数据偏差的影响。

6. 探索大模型的应用：随着大语言模型（如 GPT-4）的快速发展，未来可以尝试将大模型的生成能力与时间序列预测任务相结合。例如，通过引入大模型的上下文理解能力，进一步提升销售量预测的精度和稳定性。此外，还可以探索如何利用大模型的迁移学习能力，将在其他领域中训练的大模型应用于销售量预测任务，从而减少对大规模标注数据的依赖。

7. 开发实时预测系统：未来的研究可以尝试开发基于 TimeXer 模型的实时预测系统，实现对销售量的动态监控和预测。通过结合流数据处理技术和在线学习算法，可以实现对销售量变化的快速响应，从而为企业的实时决策提供支持。

8. 结合因果推断技术：未来的研究可以尝试将因果推断技术与 TimeXer 模型相结合，探索销售量变化的因果关系。例如，通过分析促销活动、价格调整和节假日等因素对销售量的影响，可以为企业的市场营销策略提供科学依据。

未来，销售量预测领域将迎来更多创新与突破。希望本文的研究为后续工作提供参考，同时期待更多研究者共同推动该领域的发展。

参考文献

- [1] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional time series forecasting with convolutional neural networks, 2018.
- [2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794. ACM, August 2016.
- [3] Yu-Neng Chuang*, Songchen Li*, Jiayi Yuan*, Guanchu Wang*, Kwei-Herng Lai*, Leisheng Yu, Sirui Ding, Chia-Yuan Chang, Qiaoyu Tan, Daochen Zha, and Xia Hu.

- Understanding different design choices in training large time series models. *arXiv preprint arXiv:2406.14045*, 2024.
- [4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
 - [5] JonathanD. Cryer. 时间序列分析与应用. 机械工业出版社, 2011.
 - [6] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.
 - [7] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.
 - [8] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.
 - [9] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, Rong Jin. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.
 - [10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc.
 - [11] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.
 - [12] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.
 - [13] Yining Liu and Ziyao Wang. Enhanced transformer-lstm daily sales forecasting model based on attention mechanism. In *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, pages 730–734, 2024.
 - [14] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

- [15] Ioannis Nasios and Konstantinos Vogklis. Blending gradient boosted trees and neural networks for point and probabilistic forecasting of hierarchical time series. *International Journal of Forecasting*, 38(4):1448–1459, October 2022.
- [16] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.
- [17] Xinyuan Qi, Kai Hou, Tong Liu, Zhongzhong Yu, Sihao Hu, and Wenwu Ou. From known to unknown: Knowledge-guided transformer for time-series sales forecasting in alibaba, 2021.
- [18] E. Stellwagen and Len Tashman. Arima: The models of box and jenkins. *Foresight: Int. J. Appl. Forecast.*, pages 28–33, 01 2013.
- [19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [20] Yuxuan Wang, Haixu Wu, Jiayang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 469–498. Curran Associates, Inc., 2024.
- [21] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.
- [22] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS ’21*, Red Hook, NY, USA, 2021. Curran Associates Inc.
- [23] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

- [24] Haoyi Zhou, Jianxin Li, Shanghang Zhang, Shuai Zhang, Mengyi Yan, and Hui Xiong. Expanding the prediction capacity in long sequence time-series forecasting. *Artificial Intelligence*, 318:103886, 2023.
- [25] Haoyi Zhou, Jianxin Li, Shanghang Zhang, Shuai Zhang, Mengyi Yan, and Hui Xiong. Expanding the prediction capacity in long sequence time-series forecasting. *Artificial Intelligence*, 318:103886, 2023.
- [26] 汉硕. 2024 实体零售未来白皮书. <https://max.book118.com/html/2024/0402/8072140065006053.shtm>, 2024.
- [27] 孟祥福, 石皓源. 基于 transformer 模型的时序数据预测方法综述. 计算机科学与探索, 19(1):45–64, 2025.
- [28] 胡润百富. 中国新零售白皮书 (China New Retail Report 2023). <https://res.hurun.cn/file/20231218/20231218150314249.pdf>, 2023.