arXiv:2301.02332v1 [cs.LG] 5 Jan 2023

DANLIP：用于局部可解释概率预测的深度自回归网络

Ozan Ozyegen[a]、Juyoung Wang[b]、Mucahit Cevik[a]

[a]加拿大多伦多都会大学机械与工业工程系[b]加拿大多伦多大学机械与工业工程系

摘要

尽管基于神经网络的时间序列预测方法性能出色，但解释其预测结果这一固有难题限制了它们在某些应用领域的适用性。由于难以识别此类黑箱方法的输入与输出之间的因果关系，它们很少被应用于法律和医疗等领域，而在这些领域，结果的可靠性和可解释性至关重要。本文提出 DANLIP，这是一种新颖的、基于深度学习且本质上可解释的概率时间序列预测架构。我们使用多个数据集和性能指标开展实验，并通过实证表明，我们的模型不仅具有可解释性，其性能也可与最先进的概率时间序列预测方法相媲美。此外，我们还证明，对所关注随机过程的参数进行解释，能够为多个应用领域提供有益的见解。

关键词： 可解释性、时间序列、深度学习、预测

1. 引言

可解释机器学习是统计学习的一个子领域，在过去几年中受到了广泛关注。尽管统计学习技术取得了显著成果，但此类方法在某些决策关键场景中仍未得到充分采用，例如法律和医疗领域中经常出现的场景 [15]。这主要是因为在此类情况下，决策者通常需要回答这样一个问题“机器学习模型为什么会做出这样的决策？”，正如 Molnar [16]所述。

根据 Doshi-Velez 和 Kim 的观点 [4], ，机器学习语境下的可解释性被定义为“向人类解释或以人类能够理解的方式呈现的能力”。按照这一标准，基于朴素神经网络的学习方法由于其黑箱性质而无法实现可解释性。具体而言，网络的神经元激活状态很可能无法提供人类层面的理解，使人了解神经网络如何做出预测。

线性回归常被用作传统的预测算法，也可视为可解释统计学习方法的典型示例。该方法允许用户将预测结果表示为协变量的线性组合。也就是说，模型假设响应变量表示为

$$y = \sum_{f \in \mathcal{F}} \gamma_f x_f + \varepsilon \tag{1}$$

其中， y 是目标变量， $\mathcal{F}$ 是特征集合， $\gamma_f \in \mathbb{R}$ 是所有$f \in \mathcal{F}$对应的线性模型系数， ε 是高斯噪声。标准化线性回归被视为一种可解释的预测方法，因为每个特征 $f \in \mathcal{F}$ 对应的线性组合系数γ_f 都可以视为该特征重要性的度量。

尽管线性回归已得到广泛应用，但在实践中，这种方法并非总是适用。其函数形式限制本质上限制了模型的表达能力，因为许多复杂函数无法表示为特征的线性组合。因此，为了获得更好的性能，通常不可避免地需要使用复杂的学习方法。然而，此类方法（例如深度神经网络）未必具有可解释性。

在许多领域中，确保模型的可解释性极其重要 [15]。即使在一些并不十分重视可解释性的领域，可解释模型也能大力帮助决策者做出更明智的决策。例如，对于一家大型零售公司而言，了解对旗舰产品销售额影响最大的关键因素，可能有助于制定更优的生产、库存管理和营销政策。

可解释机器学习领域主要存在两大分支 [16]。事后 方法包括应用各种技术，在模型训练完成后对模型进行分析，例如 LIME [22] 和 SHAP [12]。另一方面，内在 方法则致力于构建本质上具有可解释性的模型。这通常是

通过限制机器学习模型本身的复杂度来实现的，例如 RETAIN [3]。

一般而言，SHAP 等事后方法速度较慢，在大规模机器学习问题中效率不高。这些方法通常需要进行大量实验来计算可解释性分数。实际上，SHAP 方法的运行时间复杂度会随数据集中特征数量呈指数级增长 [16]。相比之下，内在可解释方法通常运行速度很快。研究表明，所提出方法的性能可以与不可解释方法相媲美 [3]。因此，我们专注于开发一种使用神经网络的内在可解释时间序列预测方法。由于内在方法通常训练速度更快，该框架更适合利用大量训练数据开发预测算法。

只有少数研究探索了可解释的基于深度神经网络的时间序列预测方法。在这些研究中，Temporal Fusion Transformer（TFT） [11] 与我们的工作密切相关，因为其架构在进行时间序列预测的同时也实现了可解释性。然而，TFT 与我们的工作有本质区别，因为它只计算特征的重要性分数，而不计算贡献分数。与重要性分数不同，贡献分数使用户能够量化特征对预测的正面和负面影响。我们将本研究的贡献总结如下：

- 我们提出了 DANLIP，这是一种基于深度学习、可解释的参数化概率时间序列预测模型。
- 我们证明，该模型在保持较高可解释性的同时，能够达到与最先进的概率时间序列预测方法相当的性能。我们使用三个真实世界数据集进行了广泛的数值研究，以展示所提出方法的性能。

- 我们通过实证展示，如何利用拟合到模型中的随机过程参数来解释模型的预测结果，从而获得重要的实践见解。

本文其余部分结构如下：第2节简要回顾相关文献，重点介绍 RETAIN [3] 和 DeepAR [23]模型，这两种方法与概率预测密切相关。第3节介绍我们提出的可解释时间序列预测模型。接下来，

我们将在本节中描述模型架构，并展示如何从均值和方差两个方面解读预测结果。第4节通过多个数据集上的多种性能指标，实证展示模型的性能。此外，我们还将讨论如何利用真实世界数据集，通过解读概率预测的均值和方差来获得有用的见解。最后，第5节总结研究发现并探讨未来工作。

2. 背景

本节首先简要回顾文献中最相关的研究，并概述时间序列预测的基础知识。随后，我们详细回顾 Salinas 等人 [23] 和 Choi等人 [3]的研究，为我们自己的方法论奠定基础。

2.1. 相关工作

时间序列预测领域有大量文献。该领域早期研究主要关注统计模型，例如移动平均 (MA)、自回归移动平均 (ARMA) 和自回归差分移动平均 (ARIMA) [2]。决策树及其集成模型，即梯度提升树和随机森林，也常用于时间序列预测，因为它们具有可解释的结构，并能够捕捉数据集中各种特征之间的非线性关系 [5, 14]。

虽然以往的大多数研究都侧重点预测，但近期有几项研究提出了旨在实现高质量概率预测的方法论。概率预测通常指围绕点预测的置信区间，并由上下限界定。标准方法（例如 ARIMA 和指数平滑）可以通过目标预测分布的闭式表达式或模拟生成概率预测 [2]。

近年来，随着可用数据越来越多，许多研究开始关注基于深度学习的时间序列预测方法。例如，Rangapuram等人 [21] 和 Salinas 等人 [23] 近期提出了用于概率预测的深度学习模型，可直接预测用于指定概率预测的概率分布参数（如均值和方差）。对于包含大量时间序列的数据集（例如，数量达到

数百或数千条）的数据集而言，这些方法的表现显著优于标准方法。尽管这些深度学习预测方法取得了成功，但它们也有一个重要缺点：架构高度复杂且属于黑箱架构，缺乏可解释性。我们建议读者参阅 Parmezan 等人撰写的近期综述文章， [20]以详细了解时间序列预测的统计模型和机器学习模型。此外，近期的预测竞赛也为表现最佳的时间序列预测方法提供了宝贵见解 [1, 13, 14]。

近年来，可解释人工智能也受到了广泛关注。事后可解释性方法已被用于解释时间序列模型的决策。Mujkanovic [17] 使用 SHAP 解释时间序列分类器，而 Ozyegen 等人 [19] 评估了三种事后可解释性方法（包括 SHAP），以解释时间序列预测模型。另一方面，许多时间序列预测方法在模型开发过程中考虑了可解释性因素 [3, 8]。虽然内在可解释方法通常会以牺牲性能为代价来提高可解释性，但这种权衡程度会因模型和预测任务而异。已有研究表明，一些内在可解释模型的性能与具有黑箱性质的模型相近 [10, 11]。我们的研究与以往关于可解释时间序列预测的工作有着根本区别：所提出的架构能够辨别哪些特征会对预测结果产生正面或负面影响。

2.2. 预备知识

在我们的分析中，主要采用标准的神经网络方法。为方便起见，我们另外引入以下记号：

- $[N] = \{1, 2, \ldots, N\}$ 对于任意 $N \in \mathbb{N}$。
- 给定 $A \in \mathbb{R}^{m \times n}$, $A[i,:]$ 表示矩阵 A 的第$_{\text{th}}$行， $A[:, j]$表示矩阵 A 的第 j_{th} 列。对于一维和多维张量，记号可类似地扩展。
- 给定 $A \in \mathbb{R}^{m \times n}$, $A[i_1 : i_2, :]$ 表示 $(A_{ij})_{i \in \{i_1, \ldots, i_2 - 1\}, j \in [n]}$。对于 $A[:, j_1 : j_2]$ 和 $A[i_1 : i_2, j_1 : j_2]$，定义类似。对于一维和多维张量，记号可类似地扩展。
- $\mathbb{I}_{\text{条件}}$是一个指示函数：当下标所示的条件满足时取值为 1，否则取值为 0。

- 对于任意的 $a \in \mathbb{R}^A$ 和 $b \in \mathbb{R}^B$，$[a, b]$ 表示两个向量的拼接，即 $[a, b] = [a_1, \cdots, a_A, b_1, \cdots, b_B]$。

遵循 Melis 和 Jaakkola 提出的自解释神经网络模型 [15]，，该模型与Choi等人提出的 RETAIN架构类似。[3]，我们通过限制神经网络输出的形式，确保预测方法具有可解释性：

$$y = \sum_{f \in \mathcal{F}} \gamma^{\Theta}(x)_f x_f. \tag{2}$$

方程（2）与方程（1）高度相似，但与方程（1）不同的是，与 x_f 相乘的项现在是输入的函数，$\gamma^{\Theta}: \ D \to \mathbb{R}^{|\mathcal{F}|}$，其中 D 表示输入数据的域。我们用 $\gamma^{\Theta}(\cdot)$ 来表示函数，并用 Θ表示神经网络参数的集合。

*2.3. DeepAR*架构综述

我们在数值研究中将 DeepAR [23]，视为一种最先进的概率参数化时间序列预测方法，并将其作为基线。DeepAR是一种基于RNN的参数化概率时间序列预测架构，旨在求解对数似然最大化问题。图 1 直观地概述了DeepAR架构。

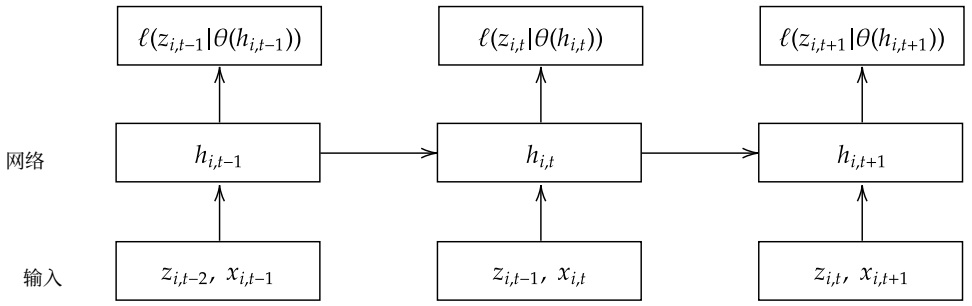


图 1：DeepAR 的简化架构

*2.4. RETAIN*架构综述

RETAIN 是一种由 Choi等人 [3]开发的基于神经网络的可解释二元分类方法。该架构将形式限制为

方程（2）所示的预测形式，并将特征 f 对预测的贡献定义为 $\gamma(x)_f \cdot x_f$。该架构使用两个循环神经网络。将嵌入输入向量与循环神经网络输出相乘后求和，并对结果应用 Sigmoid 函数，以预测二元变量 y，即感兴趣的变量。图 2展示了 RETAIN架构的简化示意图。

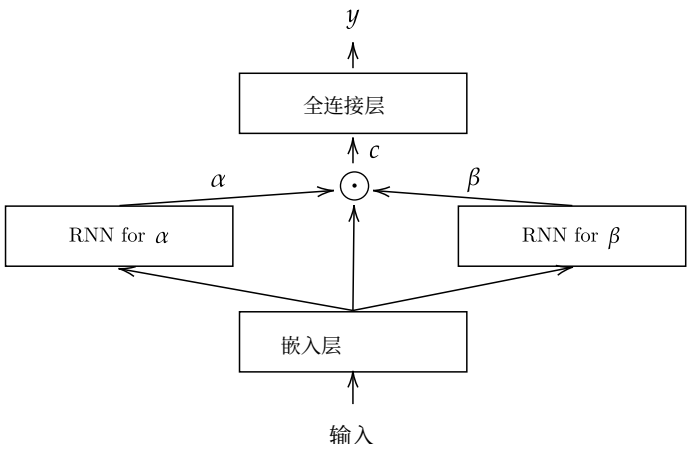


图 2：RETAIN 的简化架构

更正式地说，给定输入数据 $(x_t)_{t\in[T]}$，其中 $x_t \in \mathbb{R}^p$ 对所有 $t \in [T]$ 均成立，p 表示特征数量，RETAIN架构涉及以下一组数学运算：

$$v_i = W_{\text{emb}} x_i \tag{3}$$

$$g_i, \ldots, g_1 = \text{RNN}_\alpha(v_i, \ldots, v_1) \tag{4}$$

$$e_j = w_\alpha^\top g_j + b_\alpha \qquad \text{for } j \in [i] \tag{5}$$

$$\alpha_1, \ldots, \alpha_T = \text{Softmax}(e_1, \ldots, e_i) \tag{6}$$

$$h_i, \ldots, h_1 = \text{RNN}_\beta(v_i, \ldots, v_1) \tag{7}$$

$$\beta_j = \tanh(w_\beta^\top h_j + b_\beta) \qquad \text{for } j \in [i] \tag{8}$$

$$c_t = \sum_{j\in[i]} \alpha_j \beta_j \odot v_j \tag{9}$$

$$\mathbb{P}(y_i | (x_t)_{t\in[i]}) = \text{Softmax}(W c_i + b) \tag{10}$$

其中 α 计算每个时间步的注意力分数， $\odot$ 对应 Hadamard乘积算子。

设 r 为特征的嵌入维度。为了定义输出的特征级贡献，Choi等人 [3] 指出，方程（10）可展开如下：

$$\mathbb{P}(y_i|(x_t)_{t\in[i]}) = \text{Softmax}\left(\sum_{j\in[i]}\sum_{k\in[r]} x_{j,k}\alpha_j W\left(\beta_j \odot W_{\text{emb}}[:,k]\right) + b\right) \tag{11}$$

然后，根据方程（11），特征 k 在时间步 j对 y_i 的预测所作的贡献可表示为

$$\omega(y_i; x_{j,k}) = \alpha_j W(\beta_j \odot W_{\text{emb}})[:,k]x_{j,j} \tag{12}$$

请注意，量 ω（y_i；$x_{j,\ k}$）的定义类似于方程（ γ^{Θ}（x）中的分量；该分量可用于解释为何 RETAIN 是一种可解释模型。

2.5. 概率参数化时间序列预测

一般而言，时间序列预测方法可分为点预测和概率预测。点预测旨在直接估计目标时间序列后续各步的取值。另一方面，概率预测则量化目标时间序列取特定值的概率。具体而言，设 t 为时间步数， $I \in \mathbb{N}$ 为时间序列的数量。设 $(x_{ij},\ z_{i,\ j-1})_{t\in[I],\ j\in\{t-h,\cdots,t-1\}}$ 满足 $(x_{ij},\ z_{i,\ j-1}) \in \mathbb{R}^{|\mathcal{F}|+1}$ 为时间序列数据集的集合，其中 h 为历史窗口大小。我们将 $(x_{ij})_{ij}$ 称为协变量，将 $(z_{ij})_{ij}$ 称为目标序列。此外，假设对于所有 $i \in [I]$，均已给定 $(x_{ij})_{t\in[t,T]\cap\mathbb{N}}$，其中 $t + F > t$ 满足 $F \in \mathbb{N}$。那么，概率预测的目标是估计由下式给出的量

$$\mathbb{P}((z_{i,j})_{i\in[I],t\in[t,\ldots,t+F]\cap\mathbb{N}}|(z_{i,j})_{i\in[I],j\in\{t-h,\cdots,t-1\}}, (x_{ij})_{i\in[I],j\in\{t-h,\cdots,t+F\}}) \tag{13}$$

其中， F 为预测时间步。

估计公式（13）中所示数量主要有两种方法：

- 参数化方法通过假设 $(z_{i,j})_{i\in[I],j\in\{t+1,t+F\}\cap\mathbb{N}}$ 是来自某个联合$z_{i,j})_{i\in[I],\ j\in\{t-h,\cdots,t-1\}}$ 的随机过程来进行估计

密度 $\pi(\varsigma)$，其中已知 π 而未知 ς。作为参数化时间序列预测方法的一个示例，我们请读者参阅 Salinas 等人 [23]的研究。

- 非参数方法直接估计方程 (13)，而不对 π作任何假设。有关此类非参数方法论的更多详情，请参阅 Koenker 和 Bassett Jr [9], 以及 Lim 等人 [11]的研究。

本文仅关注参数化情形。提出可解释的参数化时间序列预测方法的主要优势在于，它能够对随机过程的参数进行解释。例如，通过解释高斯过程的参数，我们可以同时解释概率预测的均值和标准差。

3. DANLIP架构

接下来，我们将详细介绍 DANLIP 的模型规格和架构，并讨论该模型如何实现可解释性。

3.1. 模型描述

DANLIP 旨在通过逐时间步参数预测来执行联合密度参数估计。具体而言，我们将随机过程限定为高斯过程，因为高斯过程是一种直接以均值和标准差为输入的概率分布。不过，我们注意到， DANLIP架构可以进行适当修改，以处理其他类型的随机过程。例如，按照 Salinas 等人 [23], 提出的步骤，可以轻松实现此类修改。为便于阅读，我们在下文的数学方程中省略时间序列索引 i 。

首先，令 M 表示连续特征的数量， N 表示类别特征的数量。我们指出，输入数据可以分解为一个向量，其中包含目标序列值、连续特征和离散特征。更具体地说，输入数据可表示为公式（14）所示，即对于所有 $j \in \{t-h, \cdots, t-1\}$，其中 h 表示历史窗口大小，带协变量的时间序列可写为

$$[z_j, (x_{jf}^{\mathrm{Cont}})_{f=1}^{M}, (x_{jf}^{\mathrm{Cat}})_{f=1}^{N}] \tag{14}$$

其中， z_j 是时间步 j处的目标序列值，$(x_{jf}^{\text{Cont}})_{f=1}^{M}$ 表示向量 x的连续分量向量，而$(x_{jf}^{\text{Cat}})_{f=1}^{N}$ 表示类别分量。为方便记号，我们作如下假设：

- 不失一般性，我们假设默认采用这样的索引顺序。
- 我们将 $\mathcal{F}_{\text{Cat}}$ 定义为方程（14）中所示向量的类别特征索引集合。类似地，我们将连续特征定义为 $\mathcal{F}_{\text{Cont}}$ 。
- 我们假设 $x_{\text{Cat}_{jf}} \in \mathbb{N}$，即类别以自然数表示。我们假设（$x_{\text{Cat}itf}$）$_{it}$ 中的值属于从 0 开始、没有间隔的有序自然数子集，因此类别特征的值可以进行独热编码。

该 DANLIP 定义为三个重要部分的拼接：特征处理层、编码层和解码层。在图3中，我们展示了 DANLIP的整个神经网络架构的简化概览。

我们从描述特征处理层开始模型说明。对于每个时间步 $j \in \{t-h, t-1\}$,

$$e_f^{\text{Cat}} = \text{Embedding}(x_{jf}^{\text{Cat}}) \qquad \forall f \in \mathcal{F}_{\text{Cat}} \tag{15}$$

$$v_j = (z_j, x_{j,1}^{\text{Cont}} \cdots x_{j,M}^{\text{Cont}}, e_{j,1} \cdots e_{j,N}) \tag{16}$$

其中，对于所有 $f \in \mathcal{F}_{\text{Cat}}$， $e_{jf}^{\text{Cat}} \in \mathbb{R}^{\xi_f}$， ξ_f 表示类别特征 $f \in \mathcal{F}_{\text{Cat}}$的嵌入维度。使用上述记号，我们将编码层的输入定义为

$$v = [v_{t-h}, \cdots, v_{t-1}] \tag{17}$$

接下来，我们可以将编码层定义如下：

$$g_{t-h}, \ldots, g_{t-1} = \text{RNN}_\alpha(v_{t_h}, \ldots, v_{t_1}) \tag{18}$$

$$q_j = w_\alpha^\top g_j + b_\alpha \qquad \forall j \in \{t-h, \cdots, t-1\} \tag{19}$$

$$\alpha_{t-h}, \ldots, \alpha_{t-1} = \text{Softmax}(q_{t-h}, \ldots, q_{t-1}) \tag{20}$$

$$h_{t-h}, \ldots, h_{t-1} = \text{RNN}_\beta(v_{t-h}, \ldots, v_{t-1}) \tag{21}$$

$$\beta_j = \tanh(w_\beta^\top h_j + b_\beta) \qquad \forall j \in \{t-h, \cdots, t-1\} \tag{22}$$

$$c_j = \alpha_j \beta_j \odot v_j \qquad \forall j \in \{t-h, \cdots, t-1\} \tag{23}$$

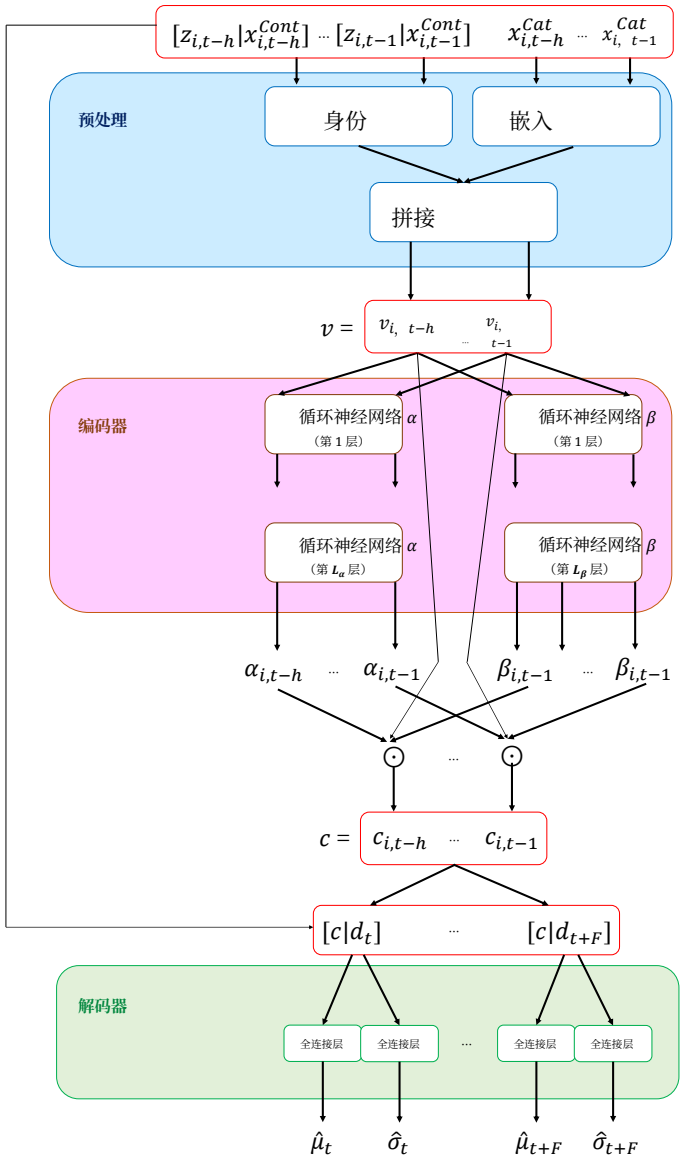


图3：DANLIP模型架构

其中，$\text{RNN}_\alpha(\cdot)$ 和 RNN_β 都可以是单向或双向的。

编码层的输出为

$$c = (c_{t-h}, \cdots, c_{t-1}) \tag{24}$$

可据此将解码层的输入定义为

$$d_j = [z_{j-1}, x_{j,1}^{\text{Cont}} \cdots x_{j,M}^{\text{Cont}}, e_{j,1} \cdots e_{j,N}] \qquad \forall j \in \{t, \cdots, t+F\} \tag{25}$$

$$d = (d_t, \cdots, d_{t+F}) \tag{26}$$

使用向量序列 $(c_j)_{j\in\{t-h,\cdots,t-1\}}$，我们最终可以在解码层中使用以下公式预测均值和标准差：

$$\mu_j = W_{\mu,j}^{\top} \cdot [c|d_j] \qquad \forall j \in \{t, \cdots, t+F\} \tag{27}$$

$$\sigma_j = \text{Softplus}(W_{\sigma,j}^{\top} \cdot [c|d_j]) \qquad \forall j \in \{t, \cdots, t+F\} \tag{28}$$

其中 $W_{\mu,\ j}$，$W_{\sigma,\ j} \in \mathbb{R}^{1\times N\times\sum_{f\in\mathcal{F}} d_f}$。我们注意到，$\mu_{it}$ 和 σ_{it} 之后都会用于定义输出条件单变量高斯分布的参数。

给定神经网络的输出，我们对神经网络权重进行优化，以求解以下似然最大化问题：

$$\max_{\Theta} \prod_{i\in I} \prod_{j=t}^{t+F} \ell_{\mathsf{UVN}}(z_{i,j}|\mu_{ij}(\Theta), \sigma_{ij}(\Theta)) \tag{29}$$

此处，我们改为求解30式所描述的对数似然最大化问题，即

$$\max_{\Theta} \sum_{i\in I} \sum_{j=t}^{t+F} \log(\ell_{\mathsf{UVN}}(z_{i,j}|\mu_{ij}(\Theta), \sigma_{ij}(\Theta))) \tag{30}$$

我们恢复已消去的索引 i，以描述时间序列索引。我们用$\ell_{\mathsf{UVN}}(\cdot|\cdot)$表示单变量高斯似然，其中 Θ 是根据网络输入进行优化的所有网络参数组成的向量。

在训练 DANLIP 期间，解码层的输入定义为每个 $j \in \{t-h, \cdots, t-1\}$ 对应的真实目标序列值 z_j。然而，在预测阶段，无法获取 $j \in \{t, \cdots, t+F\}$ 对应的未来信息 z_j。为解决这一问题，我们执行自回归预测，并将解码层输入向量 d_j 替换为 d_j。具体而言，

$$d_t = [z_{t-1}, x_{t,1}^{\text{Cont}} \cdots x_{t,M}^{\text{Cont}}, e_{t,1} \cdots e_{t,N}] \tag{31}$$

$$\hat{d}_j = [\hat{z}_{j-1}, x_{j,1}^{\text{Cont}} \cdots x_{j,M}^{\text{Cont}}, e_{j,1} \cdots e_{j,N}] \qquad \forall j \in \{t+1, \cdots, t+F\} \tag{32}$$

$$\hat{d} = (d_t, \hat{d}_{t+1}, \cdots, \hat{d}_{t+F}) \tag{33}$$

其中 $\hat{z}_{j-1}$ 是从 $\pi(\hat{\mu}_{j-1}, \hat{\sigma}_{j-1})$ 中抽取的样本，表示所有 $j \in \{t+1, \cdots, t+F\}$ 对应的预测目标序列值。在预测阶段，DANLIP 执行序列采样，如图4所示。

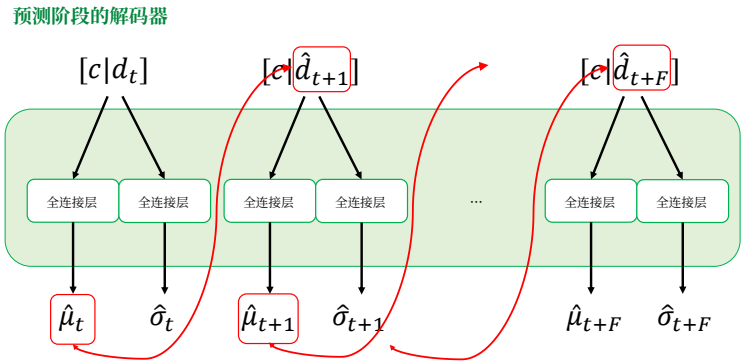


图4：预测阶段 DANLIP 架构的解码器

3.2. 模型解释

我们可以评估在 DANLIP 进行预测时，输入特征对模型输出的贡献有多大。为简洁起见，令 $\phi = 1 + |\mathcal{F}^{\text{Cont}}| + \sum_{r=0}^{f} \xi_r$ 表示单个时间步上下文向量的长度。我们可以将特征 f 在时间步 s 对 μ_j 预测的贡献定义如下：

$$\omega_\mu(f, s, j) = \alpha_s W_{\mu,j}[(s - (t-h)) \cdot \phi] \cdot (\beta_s[0] \cdot z_s), \qquad f \notin \mathcal{F}^{\text{Cont}} \cup \mathcal{F}^{\text{Cat}} \tag{34}$$

$$\omega_\mu(f, s, j) = \alpha_s W_{\mu,j}[(s - (t-h)) \cdot \phi + f] \cdot (\beta_s[f] \cdot x_{sf}), \qquad f \in \mathcal{F}^{\text{Cont}} \tag{35}$$

$$\omega_\mu(f, s, j) = \alpha_s W_{\mu,j}\Big[(s - (t-h)) \cdot \phi + (\phi - \xi_f) : (s - (t-h)) \cdot \phi + \phi\Big] \cdot \Big(\beta_s\Big[(s - (t-h)) \cdot \phi + (\phi - \xi_f) : (s - (t-h)) \cdot \phi + \phi\Big] \odot W_f^{\text{Cat,Emb}}[:, x_{sf}]\Big), \qquad f \in \mathcal{F}^{\text{Cat}} \tag{36}$$

同样， σ_j 预测的贡献定义为

$$\omega_\sigma(f,s,j) = \alpha_s W_{\sigma,j}[(s-(t-h))\cdot\phi]\cdot(\beta_s[0]\cdot z_s), \quad f \notin \mathcal{F}^{\mathrm{Cont}} \cup \mathcal{F}^{\mathrm{Cat}} \tag{37}$$

$$\omega_\sigma(f,s,j) = \alpha_s W_{\sigma,j}[(s-(t-h))\cdot\phi + f]\cdot(\beta_s[f]\cdot x_{sf}), \quad f \in \mathcal{F}^{\mathrm{Cont}} \tag{38}$$

$$\omega_\sigma(f,s,j) = \alpha_s W_{\sigma,j}\Big[(s-(t-h))\cdot\phi + (\phi-\xi_f) : (s-(t-h))\cdot\phi + \phi\Big] \cdot \Big(\beta_s\Big[(s-(t-h))\cdot\phi + (\phi-\xi_f) : (s-(t-h))\cdot\phi + \phi\Big] \odot W_f^{\mathrm{Cat,Emb}}[:, x_{sf}]\Big), \quad f \in \mathcal{F}^{\mathrm{Cat}} \tag{39}$$

我们注意到，只要所有连续特征都位于类别特征之前，且目标序列位于所有连续特征之前，所提供的公式就是有效的。若不作此假设，则必须对贡献分数的公式作适当修改。当 $\omega(f、\ s、\ j) > 0$ 时，特征 f 在时间步 j 的贡献为正。这意味着所提供的分量会使预测值增大。另一方面，负值意味着贡献分数为负。如果解码器中全连接层的激活函数不具有单调递增性质，就无法作出这样的解释。还需注意，特征对标准差的贡献是非线性的，这是由 Softplus 运算符的存在造成的。对贡献分数 $|\omega(f、\ s、\ j)|$取范数后，我们可以得到特征的重要性，而非贡献分数。随后，可以对这些特征在数据集的各预测步和样本上取均值，从而得到输入特征在整个数据集上的平均重要性，进而计算特征的全局层面贡献分数。

4. 数值研究

本节将介绍一系列详细实验的结果，以评估所提出模型DANLIP的预测性能和可解释性。首先，我们介绍实验设置，并详细说明所考察的预测模型、超参数和数据集。随后，我们比较

多个模型在常用点预测和概率预测评估指标上的预测性能。接着，我们重点介绍展示 DANLIP可解释性方面的结果。具体而言，我们首先展示模型生成的解释，然后讨论这些解释与底层预测任务的关系。

4.1. 实验设置

我们考虑了三个各具特色的数据集，它们在规模以及观测到的季节性、趋势等方面有所不同。我们采用滑动窗口法构造数据集，并将各数据集中每个时间序列最后两个预测时域分别划分为验证集和测试集。我们使用树结构 Parzen 估计器（TPE）算法，对所有模型进行详细的超参数调优。超参数调优的参数范围如表 1所示。对于 DANLIP 和 DeepAR，我们尝试了不同数量的循环神经网络层、循环神经网络单元类型、隐藏单元、丢弃率和学习率。对于多层感知机模型，我们尝试了类似范围的隐藏层、学习率和丢弃率。对于隐藏单元，我们采用了更大的范围，因为与基于循环神经网络的网络相比，全连接神经网络通常需要使用更多隐藏单元才能达到相近的性能。对于 GBR 模型，我们测试了三个重要参数：集成中的树的数量、每棵树的叶节点数（即 最大深度）以及分裂内部节点所需的最少样本数（即 最小分裂样本数）。

表 1：超参数调优搜索空间。

模型	**搜索空间**
DANLIP/ DeepAR	# 隐藏单元: [16, 128], 丢弃率: [0, 0.5], 单元类型：长短期记忆网络或门控循环单元, # 循环神经网络层： [1, 8], 学习率： [1e-4, 1e-1] (对数均匀分布)
MLP	# 隐藏单元： [50, 500], 丢弃率： [0, 0。5], # 隐藏层： [1, 8], 学习率： [1e-4, 1e-1] (对数均匀分布)
GBR	树的数量： [10, 200], 最大深度： [2, 5], 最小分裂样本数： [2, 15]

我们运行 TPE算法进行 100 次试验，以验证集上的归一化偏差指标为优化目标。神经网络模型采

用 PyTorch 实现，GBR 模型使用 Scikit-learn 实现，并使用 Optuna 库进行超参数调优。所有实验均在一台配备 RTX2070 Super 8GB GPU 和 128GB 内存、运行 Debian Linux 操作系统的计算节点上进行。

我们采用两组性能指标来评估点预测和概率预测的预测性能。为衡量点预测的质量，我们使用归一化均方根误差（NRMSE）和归一化偏差（ND），与 Salinas 等人 [23]的方法类似。对于真实值（y）和预测值（$\hat{y}$），可按如下方式计算 NRMSE 和 ND：

$$\mathrm{NRMSE}(y,\hat{y}) = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}}{\frac{1}{N}\sum_{i=1}^{N}|y_i|}, \qquad \mathrm{ND}(y,\hat{y}) = \frac{\sum_{i=1}^{N}|\hat{y}_i - y_i|}{\sum_{i=1}^{N}|y_i|} \tag{40}$$

为评估模型的概率预测能力，我们采用 ρ-风险，也称为分位数损失。$\rho-$风险可按如下方式计算：

$$\rho_\alpha(y,\hat{y}^\alpha) = \frac{\sum_{i=1}^{N}\max\{\alpha(y_i - \hat{y_i}^\alpha),(1-\alpha)(\hat{y_i}^\alpha - y_i)\}}{\sum_{i=1}^{N}|y_i|} \tag{41}$$

其中，$\hat{y}^\alpha$ 表示分位数水平为 α 时的预测值。请注意，ρ-风险的值越低越好。最后，我们使用 Friedman 检验（一种非参数检验）来比较多个数据集上的预测模型 [6]。为进行比较，我们将每个组的结果设为在所测试数据集上得到的平均误差。然后，我们应用 Friedman 检验，以确定各组均值之间是否存在统计显著差异。

我们在实验中考虑了三个数据集，即 Electricity、Rossmann 和沃尔玛数据集，下面将对它们进行简要介绍。

- *Electricity*：Electricity 数据集包含 370 个家庭的每小时电力消耗记录。该数据集已在以往研究中被广泛用于性能基准测试[11, 23]。在我们的实验中，除了电力消耗时间序列外，我们还使用了一些额外的协变量，例如一天中的小时、一周中的日期、一个月中的周次和月份。输入窗口大小和预测时域分别设为 168 和 12。

- *Rossmann*：Rossmann 销售数据集于 2015 年作为 Kaggle 竞赛的一部分发布，包含丰富的特征和大量每日销售历史记录。该数据集由每日销售额

记录组成，涵盖多家 Rossmann 商店，并包含各种协变量。在可用特征中，我们使用销售额、门店索引、商店营业指示变量、促销指示变量、州假日指示变量、学校假日指示变量、星期、月份和一个月中的周次。输入窗口大小和预测时域分别设为 30 和 12。

- 沃尔玛：沃尔玛门店销售预测数据集包含 45 家商店中 77 个部门的每周销售额，该数据集于 2014年作为 Kaggle 竞赛的一部分发布。数据集包含多种特征，例如温度、降价促销信息以及各类经济指标 （如失业率、燃油价格和居民消费价格指数）。输入窗口大小和预测时域分别设为 30 和 6。

4.2. 模型性能结果

表 2汇总了模型性能的统计数据。这些结果是在针对验证集上的每个数据集-模型组合进行严格微调后得出的。对汇总结果（即合并三个数据集的所有结果）进行 Friedman 检验后，得到 NRMSE、ND、 p 和 $\rho_{0.75}$ 的 $\rho_{0.90}$值分别为 0.12、0.08、0.12 和 0.12，这表明p值 >0.05 时， DANLIP 与 DeepAR 模型之间不存在统计显著差异。同样， DANLIP与 DeepAR 的平均性能值相近，预测性能相似。 DANLIP 在 Rossmann 和 Electricity 数据集上的表现略好，而 DeepAR 在沃尔玛数据集上的表现略好。

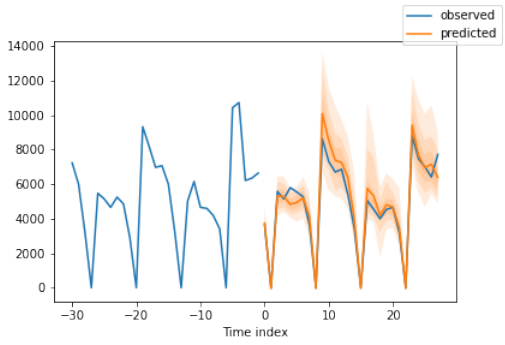
基线模型多层感知机和梯度提升回归的复杂度显著低于 DeepAR 和 DANLIP。如各种预测性能指标所示，在所有数据集中，多层感知机在受测模型中的预测性能最差。对于 Rossmann 数据集，当数据集呈现出显著的季节性时，梯度提升回归的表现优于 DANLIP和 DeepAR。对于电力数据集，梯度提升回归的 NRMSE 指标排名第二，ND 指标排名第一，但两项指标的差距都很小。对于沃尔玛数据集，按 NRMSE 指标衡量，梯度提升回归的排名仅次于 DeepAR 和 DANLIP，位居第三；按 ND 指标衡量则排名第二。梯度提升回归的高性能可归因于我们训练该模型的方式。具体而言，对于梯度提升回归模型，我们针对数据集中的每条时间序列分别训练一个模型；而对于其他模型，我们训练一个单一的

表 2：模型性能的汇总统计。报告了测试集上 10 次随机种子运行的均值和标准差。

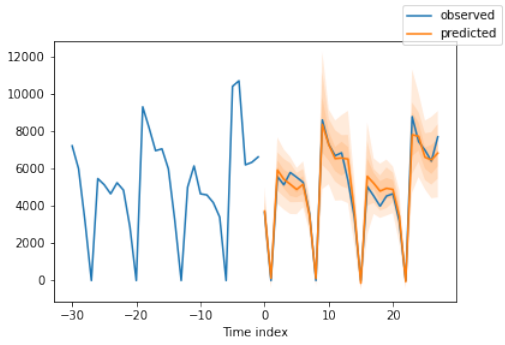
数据集	模型	归一化均方根误差		ND		$\rho_{0.75}$		$\rho_{0.90}$	
		均值	标准差	均值	标准差	均值	标准差	均值	标准差
电力	DeepAR	0.255	0.017	0.077	0.005	0.133	0.026	0.250	0.041
	GBR	0.223	0.002	0.066	0.000	0.116	0.000	0.229	0.002
	MLP	0.359	0.043	0.120	0.010	0.238	0.019	0.451	0.035
	DANLIP	0.221	0.020	0.070	0.004	0.130	0.017	0.249	0.027
Rossmann	DeepAR	0.177	0.027	0.119	0.020	0.241	0.070	0.463	0.121
	GBR	0.148	0.000	0.102	0.000	0.195	0.000	0.483	0.001
	MLP	0.179	0.005	0.122	0.003	0.281	0.013	0.596	0.032
	DANLIP	0.153	0.005	0.105	0.004	0.214	0.027	0.407	0.056
沃尔玛	DeepAR	0.152	0.013	0.076	0.007	0.136	0.026	0.248	0.043
	GBR	0.198	0.005	0.093	0.001	0.164	0.007	0.363	0.025
	MLP	0.228	0.023	0.120	0.013	0.250	0.037	0.465	0.062
	DANLIP	0.186	0.010	0.095	0.006	0.151	0.032	0.307	0.061

模型来处理数据集中的所有时间序列。梯度提升回归采用的这种训练方法在计算上开销更大。然而，当相似时间序列的可用性无法提升预测效果时，这种方法可以显著改善预测性能。我们的初步分析表明，为每个数据集训练一个单一的梯度提升回归模型（即与其他三个模型类似），会导致各数据集的预测性能显著下降，因此我们采用了上述方法。我们注意到，这些数据集的上述结果与先前研究报告的预测性能指标基本一致（例如，参见 [8, 19, 23]）。

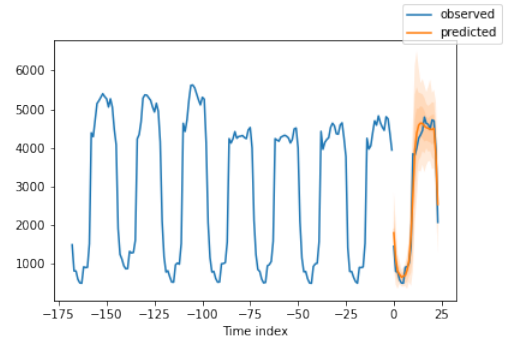
图 5展示了 DeepAR 和 DANLIP 在三个数据集上的预测可视化结果。我们用蓝线表示真实值，用橙线表示各模型预测均值。我们将 75%、90% 和 98% 的预测区间显示为阴影区域。Rossmann 、电力和沃尔玛数据集的预测时域分别为 28 日、24 小时和 6 周。总体而言，我们观察到所有模型都能生成捕捉数据集总体趋势的预测结果，而且这两个模型的概率预测非常相似。



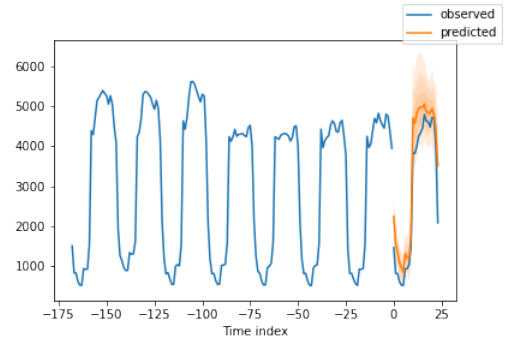
(a) Rossmann：DeepAR



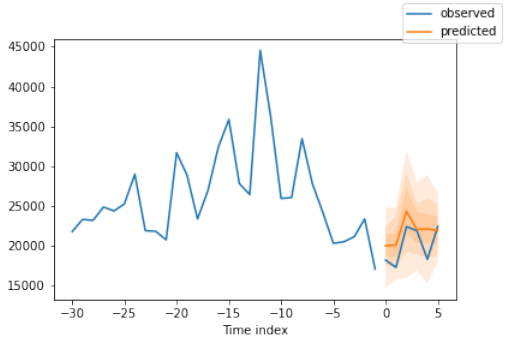
(b) Rossmann：DANLIP



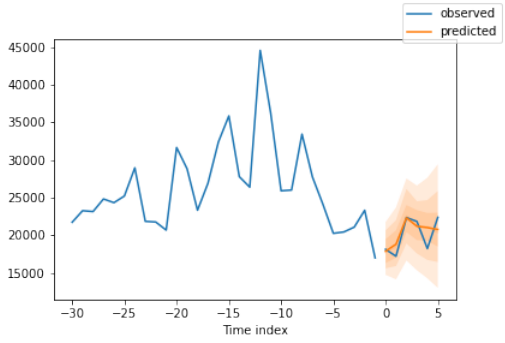
(c) 电力：DeepAR



(d) 电力：DANLIP



(e) 沃尔玛：DeepAR



(f) 沃尔玛：DANLIP

图 5：DeepAR 和 DANLIP 在随机时间序列样本上的预测结果可视化。每幅图展示一个预测窗口的预测。点预测以橙色线条表示，75%、90% 和 98% 分位数预测以橙色区域表示。每个模型生成的预测结果都能捕捉给定样本的趋势。

4.3. 模型可解释性结果

接下来，我们展示 DANLIP 的可解释性结果。我们以 Rossmann 数据集作为代表案例进行分析，因为它包含大量有趣的特征，且这些特征与目标变量（例如促销）具有较强的相关性。对于 Electricity 数据集，我们发现目标值（即电力负荷）对预测结果的影响最大，而某些时间协变量（如“月中第几周”）有助于降低预测结果的方差。对于沃尔玛数据集，我们同样发现目标值（即每周销售额）对预测结果的影响最大。沃尔玛和 Electricity 数据集的可解释性结果见附录（参见 附录 A.1）。

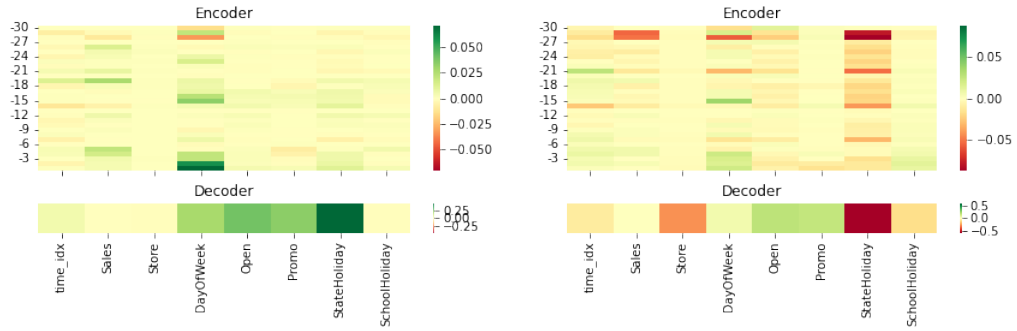
我们通过分析从模型权重中获得的解释，讨论 DANLIP 的可解释性。按照相关研究 [7, 18], ，我们将解释可视化为特征贡献热图。我们使用方法论中描述的贡献分数公式，计算每个输入特征对预测的贡献。请注意，对于每个样本、输入特征和预测时域，我们都会分别计算贡献分数。这样便能实现局部可解释性，有助于理解各特征如何影响单个预测。我们还可以对不同预测时域和样本的这些分数进行汇总，以实现全局可解释性，从而了解哪些特征对模型很重要。

图 6 展示了模型对单个样本计算出的贡献分数。我们选取 Rossmann 测试集中的第一个样本，并使用其第 9 个预测时域生成这些可视化结果。该样本和预测时域清晰地展现了模型在特定事件下的表现。该样本的预测日期是星期一，商店营业，正在进行促销活动，且没有州假日或学校假日。

图 6a显示，“星期”“营业”“促销”和“州假日”对预测结果的正向贡献最大。这符合直觉，因为周末没有销售额，而星期一的销售额理应更高。商店营业、促销活动和州假日都对预测结果有正向贡献。观察编码器的贡献时，我们发现编码器特征对预测结果的影响可以忽略不计。不过，我们还注意到，与其他编码器特征相比，“星期”特征近期的取值以及某些季节性取值对预测结果的影响更大。

图 6b 展示了各特征对预测方差的贡献。总体而言，我们观察到“StateHoliday”、“Store”和“SchoolHoli-

day”解码器特征具有显著的负贡献，而“Open”和“Promo”对预测具有正贡献。商店和节假日特征的负值表明，它们会降低预测结果的方差。有趣的是，“StateHoliday”对方差的负贡献最大。由于本例中没有与节假日对应的日期，这一解释可能表明州假日会导致更高的不确定性（方差），这需要进一步研究。另一方面，我们观察到，当商店营业且存在促销时，“Open”和“Promo”具有正贡献。这是预期之中的，因为与商店关闭且没有销售额的日期相比，商店营业时销售额预测中的不确定性更高。该解释还表明，促销的存在对预测具有类似影响，并会导致更高的方差。

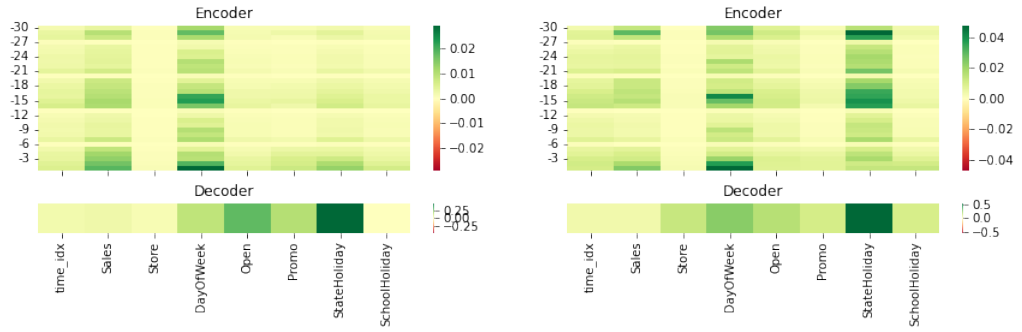


(a) μ贡献分数　　(b) σ 贡献分数

图 6：DANLIP 对 Rossmann 测试集中第一个样本和第9个预测时域计算得到的贡献分数。图 (a) 显示了对 μ 的贡献，图 (b) 显示了对 σ 的贡献。

在某些情况下，计算特征对模型的整体重要性可能比计算它们对某个特定预测的重要性更有用。这些特征重要性值可用于多种用途，包括特征选择。我们对贡献分数执行以下步骤，以确定整体重要性。首先，获取数据集中每个样本的贡献分数。所得数组的形状为：#$forecasting\ horizon \times \#samples \times \#timesteps \times \#features$。随后，我们对该数组取绝对值，因为我们关注的是特征的整体重要性，而非它们的贡献。最后，对各预测时域求平均，然后

再对样本求平均，从而得到输入特征的重要性分数。我们在图中展示了这些数组的可视化结果，分别对应 μ 和 σ ，见图7a 和图7b。这些图展示了特征对模型的整体重要性。因此，显示的所有数值均为正数，并以绿色呈现。查看图 μ 中的重要性分数7a，可以发现，样本贡献中相同的四个解码器特征（“星期”、“营业”、“促销”和“州假日”）对模型也很重要。此外，“销售额”和“星期”特征的多个编码器时间步也很重要，这表明模型会针对数据集中的不同样本关注这些特征的不同时间步。图 σ 7b所示的整体重要性分数与样本贡献分数呈现出相似的模式。相同的解码器特征，即“州假日”、“商店”、“营业”、“促销”和“学校假日”，对模型都很重要。此外，解码器中的“星期”特征对于预测目标值也很重要。在解码器侧，“销售额”、“星期”和“州假日”的某些过去值都对方差预测很重要。



(a) μ平均重要性　　(b) σ 平均重要性

图 7：DANLIP 对所有样本计算得到的贡献分数，并对样本取平均以获得全局重要性分数。图 (a) 显示了 Rossmann 数据集中 μ 的平均重要性，(b) 显示了 σ 的平均重要性。

5. 结论与讨论

在本文中，我们提出了一种新颖的深度概率型、内在可解释的时间序列预测方法 DANLIP，该方法旨在预测

底层高斯过程的均值和标准差。我们展示了如何将序数协变量和类别协变量适当地纳入 DANLIP。随后，我们介绍了利用 DANLIP网络参数计算序数特征和离散特征贡献的方法。我们还将 DANLIP 与各种基线方法进行了比较，结果表明，我们的方法在预测性能方面与最先进的概率型时间序列预测方法 DeepAR 具有竞争力。我们还分析了针对均值和标准差生成的贡献分数。我们展示了如何利用贡献分数分析单个样本的模型预测结果，并找出这些特征对模型的整体重要性。

一个值得进一步探索的方向是利用贡献分数进行特征选择。由于贡献分数直接根据模型权重计算，与其他方法相比，它们在特征选择方面可能表现更佳。其次，可以在其他数据集上测试所提出的架构。尽管对于已测试的数据集， DANLIP的表现与DeepAR相近，但该架构在应用于其他数据集和问题时，可能需要进一步调优。最后，可以分析正则化在寻找最有用解释方面所发挥的作用。L1正则化和Dropout等方法可用于调整分配给相关特征的贡献。根据具体用例，可以选择特定的正则化超参数，以获得最有用的解释。

致谢

本研究部分由LG科学园区资助。

声明与陈述

作者未报告任何潜在的利益冲突。

参考文献

[1] C. S. Bojer 和 J. P. Meldgaard。Kaggle 预测竞赛：一个被忽视的学习机会。《国际预测学杂志》，37(2):587–603，2021年。[2] G. E. Box、G. M. Jenkins、G. C. Reinsel 和 G. M. Ljung。《时间序列分析：预测与控制》。约翰·威立父子出版公司，2015年。

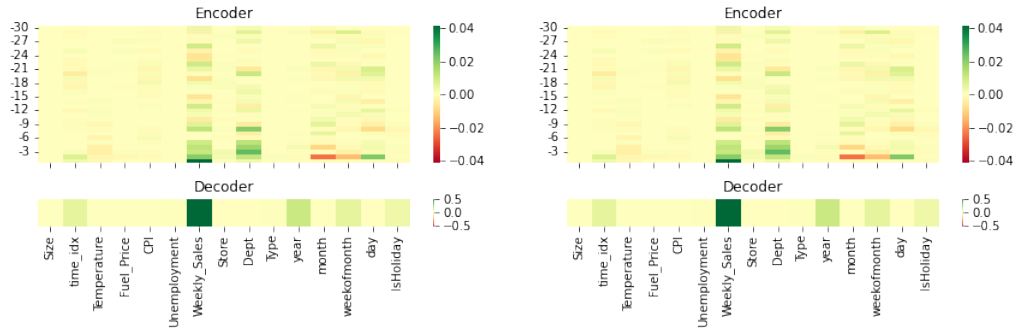
[3] E. Choi、M. T. Bahadori、J. Sun、J. Kulas、A. Schuetz 和 W. Stewart。RETAIN：一种利用逆时序注意力机制的医疗保健可解释预测模型。载于神经信息处理系统进展，第3504–3512页，2016年。[4] F. Doshi-Velez 和 B. Kim。迈向可解释机器学习的严谨科学。*arXiv* 预印本 *arXiv:1702.08608*，2017年。[5]A. Galicia、R. Talavera-Llames、A. Troncoso、I. Koprinska 和 ´F. Mart´ınez-Alvarez。基于集成学习的大数据时间序列多步预测。*Knowledge-Based Systems*，163:830–841，2019年。[6] S. Garc´ıa、A. Fern´andez、J. Luengo 和 F. Herrera。计算智能与数据挖掘实验设计中用于多重比较的高级非参数检验：功效的实验分析。《信息科学》，180(10):2044–2064，2010年。[7] T. Guo、T. Lin 和 N. Antulov-Fantulin。探索适用于多变量数据的可解释LSTM神经网络。载于国际机器学习会议，第2494–2504页。PMLR，2019年。[8] I. Ilic、B. G¨org¨ul¨u、M. Cevik 和 M. G. Baydo˘gan。用于时间序列预测的可解释提升线性回归。*Pattern Recognition*，第108144页，2021年。[9] R. Koenker 和 G. Bassett Jr.。基于回归分位数的异方差性稳健检验。*Econometrica*：计量经济学会杂志，第43–61页，1982年。[10] L. Li、J. Yan、X. Yang 和 Y. Jin。学习用于概率时间序列预测的可解释深度状态空间模型。*arXiv* 预印本 *arXiv:2102.00397*，2021年。.[11] B. Lim、S. O. Arık、N. Loeff 和 T. Pfister。用于可解释多时域时间序列预测的时间融合变换器。《国际预测学杂志》，2021年。[12] S. Lundberg 和 S.-I. Lee。解释模型预测的统一方法。*arXiv* 预印本 *arXiv:1705.07874*，2017年。[13] S. Makridakis、E. Spiliotis 和 V. Assimakopoulos。M4竞赛：10万条时间序列和61种预测方法。《国际预测学杂志》，36(1):54–74，2020年。[14] S. Makridakis、E. Spiliotis 和 V. Assimakopoulos。M5准确性竞赛：结果、发现与结论。《国际预测学杂志》，2020年。

[15] D. A. Melis 和 T. Jaakkola。《借助自解释神经网络实现稳健的可解释性》。载于神经信息处理系统进展，第 7775–7784 页，2018年。[16] C. Molnar。《可解释机器学习》。Lulu.com，2020年。[17] F. Mujkanovic。《解释任意时间序列分类器的预测结果》。硕士论文，哈索·普拉特纳研究所，2019年7月。[18] B. Norgeot、D. Lituiev、B. S. Glicksberg 和 A. J. Butte。《慢性病门诊护理中深度多变量纵向患者结局预测系统的时间聚合与模型解释》。《style id='9'>arXiv 预印本 arXiv:1811.12589，2018年。[19]O. Ozyegen、I. Ilic 和 M. Cevik。《多变量时间序列预测局部解释方法的评估》。《style id='12'>arXiv 预印本 arXiv:2009.09092，2020年。[20] A. R. S. Parmezan、V. M. Souza 和 G. E. Batista。《时间序列预测中统计模型与机器学习模型的评估：确定各模型的最新水平及其最佳应用条件》。《style id='15'>《信息科学》，484：302–337，2019年。[21] S. S. Rangapuram、M. W. Seeger、J. Gasthaus、L. Stella、Y. Wang 和 T. Januschowski。《用于时间序列预测的深度状态空间模型》。载于神经信息处理系统进展，第 7785–7794 页，2018年。[22] M. T. Ribeiro、S. Singh 和 C. Guestrin。《“我为什么要相信你？”——解释任意分类器的预测结果》。载于第 *22* 届 *ACM SIGKDD* 国际知识发现与数据挖掘会议论文集，第 1135–1144 页，2016年。[23] D. Salinas、V. Flunkert、J. Gasthaus 和 T. Januschowski。《DeepAR：基于自回归循环网络的概率预测》。《style id='24'>《国际预测期刊》，36(3)：1181–1191，2020年。

附录 A.

附录 *A.1.* 关于 *DANLIP*可解释性的补充结果

图 A.1 展示了模型针对单个样本得出的贡献分数，而图 A.2展示了各特征对模型的总体重要性。总体而言，我们发现，目标特征（每周销售额）的影响最大，尤其是在解码器步以及编码器较近的时间步上。仔细观察贡献分数，我们还发现，“部门”、“年份”、“月份”、“月份中的周”和“日”这些特征的多个时间步都会对预测结果产生影响。然而，查看图 A.2中的重要性分数后，我们发现，这些特征对预测的整体影响对于 μ 而言并不显著，但对于预测的σ则显著。

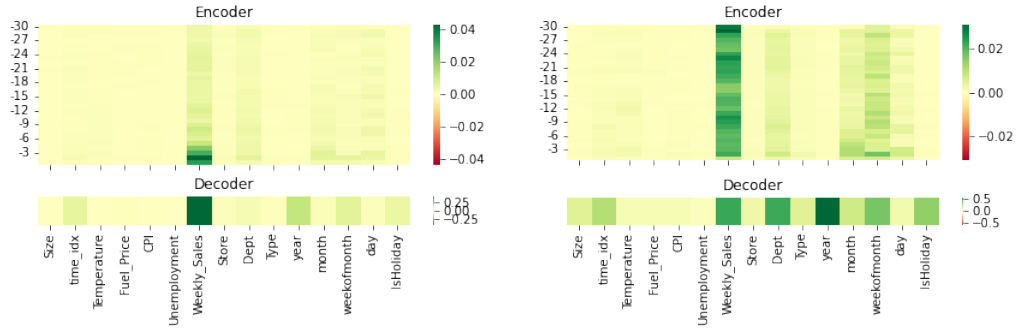


(a) μ贡献　　(b) σ 贡献

图 A.1：DANLIP 对沃尔玛测试集第一个样本和第 4 个预测时域计算得到的贡献分数。图 (a) 显示对 μ 的贡献，(b) 显示对 σ 的贡献。

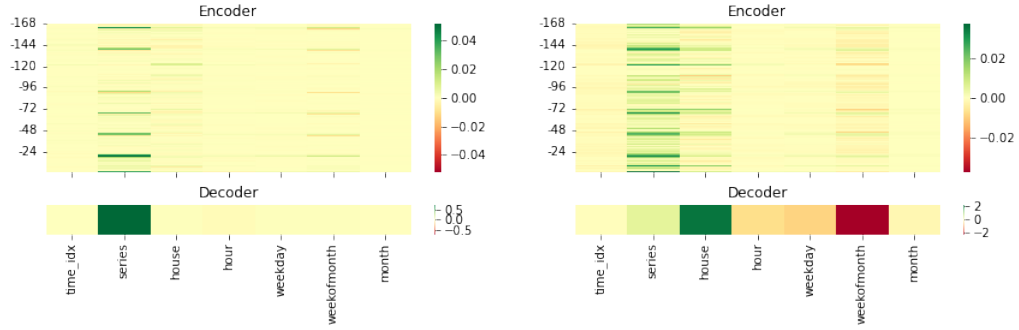
图 A.3展示了模型针对单个样本得到的贡献分数，图 A.4展示了模型中特征的整体重要性。与沃尔玛数据集类似，我们观察到，目标特征（序列）的影响最为显著，尤其是在解码器时间步上。比较贡献分数与整体重要性分数后，我们发现，对于不同的样本，模型倾向于关注“序列”特征不同的过去编码器时间步。这与沃尔玛数据集不同，在沃尔玛数据集中，模型更关注编码器中较近的时间步。这表明，对于沃尔玛数据集，较近的时间步是

信息量最大的，而对于电力数据集，目标特征的季节性取值（$t-24$、$t-48\cdots$）信息量最大。查看 σ 图A.3b中的贡献分数，我们发现，“房屋”特征表示电力负荷信息采集自哪栋房屋，该特征会导致预测方差增加。另一方面，时间协变量（“小时”、星期”和“月份中的周”）的解码器时间步会降低预测方差。



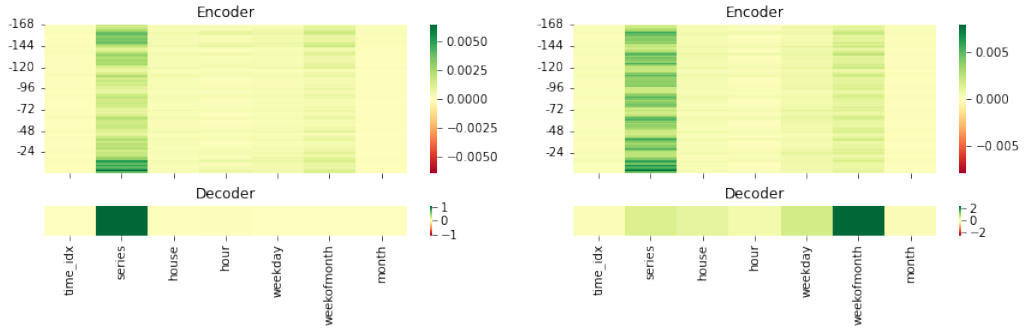
(a) μ平均重要性　　(b) σ 平均重要性

图 A.2：DANLIP 对所有样本计算的贡献分数，并对样本取平均以获得全局重要性分数。图 (a) 显示对 μ 的平均重要性，(b) 显示沃尔玛数据集中对 σ 的平均重要性。



(a) μ贡献　　(b) σ 贡献

图 A.3：DANLIP 针对电力测试集的第一个样本和第 4 个预测时域计算的贡献分数。图 (a) 显示对 μ 的贡献，(b) 显示对 σ的贡献。



(a) μ平均重要性　　(b) σ 平均重要性

图 A.4：DANLIP 对所有样本计算的贡献分数，并对样本取平均以获得全局重要性分数。图 (a) 展示了 Electricity 数据集中 μ 的平均重要性，图 (b) 展示了 σ 的平均重要性。