arXiv:2304.03877v1 [stat.ML] 8 Apr 2023

OFTER：用于时间序列预测的在线流程

预印本

Nikolas Michael[†*]、Mihai Cucuringu[†‡⊙§] 和 Sam Howison[‡]

[†]牛津大学统计系，英国牛津 St Giles' 街 24–29 号，邮编 OX1 3LB
[‡]牛津大学数学研究所，Andrew Wiles 大楼，Woodstock 路，英国牛津，邮编 OX2 6GG[§] 牛津大学牛津-曼恩定量金融研究所，英国牛津[⊙] 艾伦·图灵研究所，John Dodson House，Euston 路 96 号，英国伦敦，邮编 NW1 2DB

2023年4月

摘要

我们介绍了 OFTER，这是一种专为中等规模多元时间序列设计的时间序列预测流程。OFTER 将 k 近邻和广义回归神经网络这两种非参数模型与降维组件结合使用。为避开维数灾难，我们采用基于最大相关系数修正版的加权范数。我们提出的流程专为在线任务设计，输出具有可解释性，并能够胜过若干最先进的基线方法。该算法的计算效率、在线特性，以及在低信噪比环境下运行的能力，使 OFTER 成为金融多元时间序列问题的理想方法，例如每日股票预测。我们的研究表明，尽管深度学习模型在时间序列预测方面前景广阔，但谨慎整合主流工具的传统方法仍是极具竞争力的替代方案，并兼具可扩展性和可解释性。

关键词 多元时间序列预测；非参数回归；降维；最大相关；秩一更新；股票价格与成交量预测。

目录

1. 引言.2 2. 预备知识与相关工作.3 2.1 k 近邻与广义回归神经网络.4 2.2 降维.4 2.3 最大相关.5 3. 方法论.6 3.1 嵌入的在线更新.7 3.2 缩放因子的选择.8 3.3 克服 k 近邻和广义回归神经网络的缺点.10 3.4 算法概述.11 4. 实证结果.11 4.1 基线模型.12

*通讯作者。电子邮箱：nikolas.michael@kellogg.ox.ac.uk

4.2 合成数据集.134.3 真实数据集.134.4 数值结果.155. 特征重要性与异常值检测的应用.175.1 特征重要性.175.2 异常值检测.196. 结论与未来工作.21

A. 一种用于求解世俗方程的凸化方法.24B. 基线方法简述.24C. 稳健性分析.26

1 引言

时间序列是具有时间结构的数据集，广泛存在于从物理学到社会科学的大多数学科中。在预测任务中，可以利用这种时间结构来提升性能，同时避免训练阶段出现前视偏差。对于单变量时间序列，ARIMA（自回归积分移动平均）和ARCH（自回归条件异方差）等参数化线性时间序列模型，在大多数关注的应用中都取得了相当好的效果。尽管这些算法可以扩展到多元时间序列场景，但近几十年来计算能力的迅速提升，使机器学习技术成为传统时间序列方法的一种颇具吸引力的替代方案。较早的一项不同机器学习模型比较研究是 [1], , 作者由此得出结论：方法的选择对性能有显著影响。

深度学习模型一直是多元时间序列预测的热门选择，包括长短期记忆网络（LSTM） [2], 双向 LSTM [3] 和门控循环单元（GRU） [4] 。近年来，人们提出了这类深度学习模型的多种扩展，例如 DeepAR [5] 和时间融合变换器 [6],, 它们通常在各个领域都达到了最先进的性能。然而，这些方法往往使用数千个可训练参数，在观测数量有限时难以训练，因而导致样本外预测性能下降。尽管深度学习取得了这些进展，基于简单回归的模型仍被广泛使用且备受青睐。对于中低频金融任务而言，情况尤其如此，例如预测日收益率、成交量和已实现波动率；这部分是由于金融市场复杂、难以预测且动态不断变化。比如，普通最小二乘法（OLS）因其简单、具有闭式解且无需选择超参数，至今仍广泛用于预测和解释性统计分析。其正则化变体，如最小绝对收缩和选择算子（LASSO）和弹性网络，是金融领域应用最广泛的回归模型之一，因为它们只需要少量超参数。

另一方面，近期的多项研究表明，将传统时间序列模型与机器学习模型相结合具有优势，例如 [7, 8, 9, 10, 11]。本文考虑一种最简单的混合模型：以单变量 ARIMA 模型的结果为基础，并通过对 ARIMA 模型的残差采用一种新颖的回归方法来增强模型。

非参数模型与降维。 机器学习领域相对忽视了两种简单模型，即 k 近邻（kNN）和广义回归神经网络（GRNN）。它们都是简单、非参数且快速的方法，只需一个超参数。这主要是因为维数灾难会影响这两种技术。GRNN 主要用于工程应用，例如 [12] 用于标志识别、 [13] 用于饲喂模式预测，以及 [14] 用于预测土壤的桩承载力。 [15]对其在动力系统中的不同应用进行了综述。在这项工作中，我们旨在通过以下两种技术克服这两种方法所面临的维数灾难。首先，我们不采用 kNN 和 GRNN 使用的欧几里得范数，而是考虑一种由特征的预测潜力驱动的阈值加权范数。其次，在使用这些知名模型之前，我们加入降维环节，这是迄今为止应对维数灾难最常见的方法。

降维是无监督机器学习的主要范式之一。过去几十年间，非线性降维方法取得了开创性进展，例如局部线性

嵌入（LLE） [16], ISOMAP [17], HessianLLE [18], 拉普拉斯特征映射 [19], 和扩散映射 [20]。近年来，深度学习模型也越来越多地被用作降维手段，例如自编码器这一类模型 [21], ，它们可视为 PCA 的非线性扩展。先进行降维再采用 k近邻技术，已被证明能显著提升性能 [22]。偏最小二乘是一种常用方法，可在嵌入阶段将目标值的信息纳入初始特征。最近， [23] 通过根据每个因子相对于目标值的预测斜率进行缩放，提升了 PCA 的性能。

在金融应用中的用途。 时间序列预测任务中的降维应用示例包括[24], ，其使用截断扩散映射为降阶动力学模型的慢动力学构建全局流形；以及 [25] ，其将流形学习技术与多种监督学习模型结合，为外汇市场构建盈利策略。与此类似， [26] 提出了一种混合方法，将 Midas-RV 模型与 PCA 相结合，以更准确地预测中国市场原油期货的波动率。本文考虑的一项金融应用是预测每日成交量。准确预测次日成交量对从业者，尤其是做市商而言具有重要意义。此前的研究包括 [27], ，其展示了如何利用搜索引擎数据预测未来交易量。与我们的工作更为相近的是 [28], ，其使用支持向量机预测每日成交量百分比，并取得了良好的效果；这一思路与 [29] 相似，后者在使用自回归模型消除残差后，将支持向量机与长短期记忆网络相结合。

主要贡献：OFTER——一种新颖的集成流程。我们研究单步时间序列预测任务，并提出一种方法，将 ARIMA 模型、降维和简单的非参数方法整合到单一流程中，称为基于时间嵌入回归的在线预测（OFTER）。为缓解高维度带来的问题并增强降维组件，我们基于最大相关的概念，对嵌入特征进行特征选择。我们将 OFTER 应用于金融任务，通过基于 10 只交易所交易基金（ETF）的多种特征构建嵌入，预测次日成交量和收益。通过对合成数据和真实数据进行大量数值模拟，我们证明，面对中等规模数据集时，OFTER 是其他主流机器学习模型的有力替代方案，其表现甚至优于成熟的、针对时间序列设计的深度学习模型。具体而言，在中等规模数据集的低信噪比环境下，OFTER 仍能有效工作，同时保持较高的计算效率。

此外，我们提出的方法论还具有以下优势。

- 一旦完成嵌入，算法便可轻松进行样本外扩展，并纳入新到达的时间观测；因此，OFTER 自然可以被视为一种在线算法，能够在在线环境中高效运行。
- 我们方法的某些配置属于无监督学习设置，因为嵌入与目标值无关。换言之，同一嵌入有可能用于预测不同的目标变量，因此 OFTER 也适用于扩展至多目标设置。
- 我们的方法还可用于变点检测和异常检测任务，这些可视为自然产生的副产品。在嵌入中，先前或新到达的观测若没有任何局部邻近点，可能表明测量值极端，或时间序列中出现了变点。

论文结构。 本文其余部分的结构如下。第 2 节概述所提出流程中使用的一些基本概念，包括 k近邻和广义回归神经网络算法、主成分分析以及最大相关。第 3节介绍我们提出的方法论，以及其主要成分和如何将其整合为一种算法。第 4 节分析 OFTER 在合成实验和真实世界数据中的表现。我们还在第 5节讨论 OFTER 的其他性质。最后，我们在第 6节作出总结。

2 Preliminaries and related works

考虑一个由矩阵 $X = (x_1, \ldots, x_T) \in \mathbb{R}^{T\times d}$ 表示的 d 维时间序列，其中各行表示沿 T时间步的观测值，我们称之为特征矩阵。在本文余下部分中，假设我们关注的是对单个单变量目标变量进行预测，我们将其记为 $y = (y_1, \ldots, y_T) \in \mathbb{R}^T$，并称之为目标变量。此外，用 $x' \in \mathbb{R}^d$ 和 $y' \in \mathbb{R}$分别表示与新观测相关的特征值和目标变量值。我们用 e_d 表示长度为 d的全 1 向量。在矩阵索引中，我们通常用 $X_{i,\ j}$表示 X 的第 (i,j) 个元素，用 $X_{t_1:t_2}$ 表示 X 中从第 t_1行到第 t_2 行的子矩阵，即 $(x_{t_1}, \ldots, x_{t_2})$；用 $X_{t_1:t_2,\ j}$ 表示 $X_{t_1:t_2}$的第 j^{th} 列。

2.1 k 近邻与广义回归神经网络

k 近邻（kNN）是一种非参数监督学习算法，最早可追溯到 Fix 和 Hodges 于 1951 [30], 的研究，现已发展成为最受欢迎的机器学习技术之一。其核心思想是，特征空间中的数值越相似，目标变量的数值也越相似。将$x_{(1)}$，...，$x_{(T)}$ 记为时间序列各项的一种重新排列，使得

$$||x_{(1)} - x'|| \leq \cdots \leq ||x_{(T)} - x'||, \tag{1}$$

其中，$||.||$ 表示欧几里得范数。新条目 x 的 k 近邻集合定义如下：

$$\text{NN}_k(x') = \{\, i \in \{1, \ldots, T\} : ||x_{(i)} - x'|| \leq ||x_{(k)} - x'||\}. \tag{2}$$

基于 $\text{NN}_k(x')$，y' 的预测值计算如下：

$$\hat{y} = \frac{1}{k} \sum_{i \in \text{NN}_k(x')} y_i. \tag{3}$$

一种密切相关的方法是广义回归神经网络（GRNN），该方法在 [31] 中提出。预测y'时，GRNN 会为之前时间序列中的每个条目赋予一个权重，该权重由它们与 x'的距离决定。具体而言，将 i^{th} 个观测的权重定义为 w_i，如下所示：

$$w_i = \frac{\exp\left(\frac{-||x_i - x'||^2}{h}\right)}{\sum_{j=1}^{T} \exp\left(\frac{-||x_j - x'||^2}{h}\right)}, \tag{4}$$

其中 h 是带宽 超参数，需要对其进行优化。新观测的响应预测值 $\hat{y}$ 为 n 训练观测响应的 加权平均，定义如下

$$\hat{y} = \sum_{i=1}^{T} w_i y_i. \tag{5}$$

考虑观测值 $(x_1,\ y_1)$.....，$(x_T,\ y_T)$，将其视为某个 $\mathbb{R}^{d+1}$ 随机向量的实现，并令 f 表示该随机变量相应的真实底层密度。广义回归神经网络的理论依据源自帕尔 zen 估计量 [31],，它们用于估计未知概率密度函数 f，如下所示

$$\hat{f}_h(x', y') = \frac{1}{T} \frac{1}{(\pi h)^{\frac{d+1}{2}}} \sum_{i=1}^{T} \exp\left[-\frac{(x' - x_i)^T (x' - x_i)}{h}\right] \exp\left[-\frac{(y' - y_i)^2}{h}\right]. \tag{6}$$

如果 h 随 T 变化，使得 $\lim_{T\to\infty} h(T) = 0$ 且 $\lim_{T\to\infty} Th^d(T) = 0$，则可证明这些估计量是一致的，即

$$plim_{T\to\infty} \hat{f}_{h(T)}(x', y') = f(x', y'),\ \forall (x', y') \in \mathbb{R}^{d+1}. \tag{7}$$

ˆ 非正式地，[31]认为，如果 $f(x, y) \approx f(x, y)$，那么

$$\hat{y} = \sum_{i=1}^{T} w_i y_i = \int y' \frac{\hat{f}(x, y')}{\int \hat{f}(x, y) dy} dy' \approx \int y' \frac{f(x, y')}{\int f(x, y) dy} dy' = \mathbb{E}\left[y' | x'\right]. \tag{8}$$

提醒读者注意，尽管广义回归神经网络（GRNN）的名称中包含“神经网络”，但它与其他流行变体截然不同。具体而言，除宽度超参数外，无需选择或优化任何临时超参数。此外，它是一种单遍算法，因为权重不会经过优化。因此，与其他神经网络结构不同，它的训练速度快得多、也容易得多，但表达能力不及神经网络。

2.2 降维

k近邻和广义回归神经网络都受到所谓的维数灾难 的影响。维数灾难指的是这样一种现象：对于某些方法，要以一定的精度估计未知函数，所需的数据点数量会随着数据集维度的增加而呈指数级增长。[2] 为缓解这一问题，通常会先对 k近邻或广义回归神经网络进行

[2]举例说明这一问题：设有若干点均匀分布在边长为（$1+\epsilon$）r的 d维超立方体上。在这种情况下，只有（$1+\epsilon$）$^{-d}$ 的数据点位于边长为 r的中心内层超立方体中，其余点则位于外壳层。这表明，数据点之间的距离已不适合用于有意义地描述高维数据集的结构

将数据点从 $\mathbb{R}^d$ 投影到低维空间 $\mathbb{R}^p$ （$p \leq d$），同时在一定程度上保留原始数据集的结构。迄今为止，最受欢迎的此类方法是主成分分析（PCA）：它对原始特征进行线性变换，选择能够最大化投影特征在新降维空间中的方差的变换，同时确保所得特征彼此不相关。

将 $C(X)$ 记为 X 的协方差矩阵，其特征分解记为 $C(X) = U\Lambda U^T$，其中 Λ 是一个对角矩阵，包含 $C(X)$ 的 $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ 个特征值；U 的各列为 $u_1, \ldots, u_d$正交归一特征向量（顺序与对应特征值相同）。主成分得分由 XU 的各列给出，新成分的方差等于相应特征值（成分 i 的方差为 λ_i）。选择前 p 个成分，可保留特征矩阵总方差的 $\sum_{i=1}^{p} \lambda_i / \sum_{i=1}^{d} \lambda_i$。给定要保留的方差比例 $\delta \in (0, 1)$，我们设定

$$p = \min \left\{ i : \frac{\sum_{k=1}^{i} \lambda_k}{\sum_{k=1}^{d} \lambda_k} \geq \delta \right\}. \tag{9}$$

我们指出，也可以采用基于随机矩阵理论的方法，从对应于噪声分量的特征值主体部分中划定谱的信息部分 [32]。注意，在计算 $C(X)$ 之前，我们确保 X 具有满秩，方法是移除 X 中会导致秩亏的列。[3] 应用 PCA 前，我们对 X 进行标准化（每个特征减去其均值，再除以其标准差）。我们将˜变换后的特征记为 $X = XU$。

2.3 最大相关

考虑两个单变量随机变量，分别记为 V_1 和 V_2。粗略地说，V_1 与 V_2 之间的相关性用于衡量它们之间的关联。常见的相关性类型包括皮尔逊相关、斯皮尔曼相关和肯德尔相关。另一种类型是最大相关（例如参见 [33]），其定义如下：

$$\underset{\phi, \psi \in \mathcal{F}}{\operatorname{argmax}} \operatorname{Cor}(\phi(V_1), \psi(V_2)), \tag{10}$$

其中，$\mathcal{F}$ 是一个适当的函数类，Cor 是皮尔逊相关，其定义如下：

$$\operatorname{Cor}(V_1, V_2) = \frac{\mathbb{E}[V_1 V_2] - \mathbb{E}[V_1]\,\mathbb{E}[V_2]}{\sqrt{\mathbb{E}[V_1^2] - \mathbb{E}[V_1]^2}\sqrt{\mathbb{E}[V_2^2] - \mathbb{E}[V_2]^2}}. \tag{11}$$

通过迭代优化函数 ϕ 和 ψ，可以求得最大相关问题的解。我们考虑一个相关但更简单的问题，目标是计算

$$\underset{\phi \in \mathcal{F}}{\operatorname{argmax}} \operatorname{Cor}(\phi(V_1), V_2). \tag{12}$$

考虑式（12）而非式（10）的原因是，在回归场景中，我们只关注对特征进行变换，而非对目标值进行变换。通过求解式（10），可以变换嵌入空间中的特征，使其与目标值的相关性达到最大。

我们考虑在受限函数类 $\mathcal{F}$中求解。[4] 具体而言，我们考虑伯恩斯坦多项式类（例如参见 [34]），其定义如下：

$$b_{\nu,n}(x) = \binom{\nu}{n} x^{\nu}(1-x)^{n-\nu}, \quad \nu = 0, 1, \ldots, n. \tag{13}$$

根据斯通-魏尔斯特拉斯定理，可以证明选择该函数基的合理性。[5] 伯恩斯坦多项式已得到广泛应用，尤其是在贝塞尔曲线 [35], 和 B样条 [36], 领域；它们常见于计算机辅助几何设计中的建模工具开发。

[3]使用QR分解算法，我们可以表示为 $X = QR$，其中 Q 是一个正交矩阵，R 是一个上三角矩阵。然后我们删除任意特征 i，其中$|R_{i,\,i}| < \epsilon$对于某个较小的 $\epsilon > 0$[4]如果选择 $\mathcal{F}$ 为所有函数的类，那么很容易证明上述问题的解为平凡解 $\Phi(V_1) = \frac{V_2 - \mathbb{E}[V_2]}{\text{方差}(V_2)}$。[5]其最常见的表述是：对于区间 $[a, b]$ 上的任意实连续函数 f，以及 $\epsilon > 0$，$\exists$存在一个多项式函数 g，使得 $\forall x \in [a, b]$，我们有 $|f(x) - g(x)| < \epsilon$。

为求解方程（12），我们假设观测到向量 v_1 和 v_2，它们分别包含随机变量 V_1 和 V_2的 N 个观测。用 $\bar{v}_1$ 和 $\bar{v}_2$ 分别表示 v_1 和 v_2 的均值。同样，用 σ_{v_1}和 σ_{v_2}分别表示 v_1和 v_2的标准差。考虑函数基（ϕ_1，.....，ϕ_K），并定义矩阵 Φ，使得 $\Phi_{i,\ j} = \phi_j(v_{1,\ i})$。我们假设 Φ 是中心化的，即 $\Phi^T e_N = \mathbf{0}$。

定义好函数基后，我们考虑函数类

$$\mathcal{F} = \left\{ \sum_{k=1}^{K} c_k \phi_k : c_1, \ldots, c_K \in \mathbb{R} \right\}, \tag{14}$$

使得 $f \in \mathcal{F}$，$f(v_{1,\ i}) = \sum_{k=1}^{K} c_k \phi_k(v_{1,\ i})$，而问题（12）归结为寻找合适的向量 c。

为简化计算，我们还要求变换后的向量经过标准化，即 $c^T \Phi^T e_N = 0$，$\frac{1}{N-1} c^T \Phi^T \Phi c = 1$。因此，我们旨在求解

$$\underset{c \in \mathbb{R}^K,\ c^T \Phi^T e_N = 0,\ c^T \Phi^T \Phi c = N-1}{\operatorname{argmax}} \operatorname{Cor}(\Phi c, v_2). \tag{15}$$

注意

$$\operatorname{Cor}(\Phi c, v_2) = \frac{\operatorname{Cov}(\Phi c, v_2)}{\sigma_{v_2} \sigma_{v_1}} = \frac{\frac{(\Phi c)^T v_2}{N-1} - \left[\frac{(\Phi c)^T e_N}{N}\right]\left[\frac{v_2^T e_N}{N}\right]}{\sigma_{v_2} \sigma_{v_1}}, \tag{16}$$

因此，在满足以下约束条件的情况下（15），我们有

$$\operatorname{Cor}(\Phi c, v_2) = \frac{(\Phi c)^T v_2}{(N-1)\sigma_{v_2}}, \tag{17}$$

通过考虑拉格朗日函数，可以找到其解

$$L = \frac{c^T \Phi^T v_2}{(N-1)\sigma_{v_2}} - \lambda (c^T \Phi^T e_N) - \mu (c^T \Phi^T \Phi c - (N-1)). \tag{18}$$

上述拉格朗日函数的极值点满足

$$\mu = \frac{c^T \Phi^T v_2}{2(N-1)^2 \sigma_{v_2}}, \text{ and} \tag{19}$$

$$\lambda = \frac{\bar{v}_2}{(N-1)\sigma_{v_2}}. \tag{20}$$

代回方程（18），并记 $z = \Phi^T v_2$，$\bar{z} = z - \bar{v}_2 \Phi^T e_N$和 $A = \frac{\Phi^T \Phi}{N-1}$，我们可将问题化简为求解

$$\bar{z} = \left(c^T z\right) A c. \tag{21}$$

设解的形式为 $c = \tau A^{-1} \bar{z}$，其中 $\tau \in \mathbb{R}$，即可得到解

$$\tau = \frac{1}{\sqrt{\bar{z}^T (A^{-1})^T z}}. \tag{22}$$

综上，方程（15）的解为

$$c = \frac{A^{-1}\bar{z}}{\sqrt{\bar{z}^T (A^{-1})^T z}} = \frac{(\Phi^T \Phi)^{-1} \Phi^T \left(v_2 - \bar{v}_2 e_N\right)}{\sqrt{\left(v_2 - \bar{v}_2 e_N\right)^T \Phi \bar{z}^T \left[(\Phi^T \Phi)^{-1}\right]^T \Phi^T v_2}}. \tag{23}$$

我们将目标值 y 与变换后的 x 之间的相关性称为x 和 y的 单侧最大相关性（OSMC），即

$$\operatorname{OSMC}_K(x, y) = \operatorname{Cor}(\Phi c, y), \tag{24}$$

其中，Φ 和 c 取决于 x 以及所用基函数的数量选择 K。

3 方法论

图 1是 OFTER 流程主要成分的示意图，本节余下部分将逐一详细介绍。

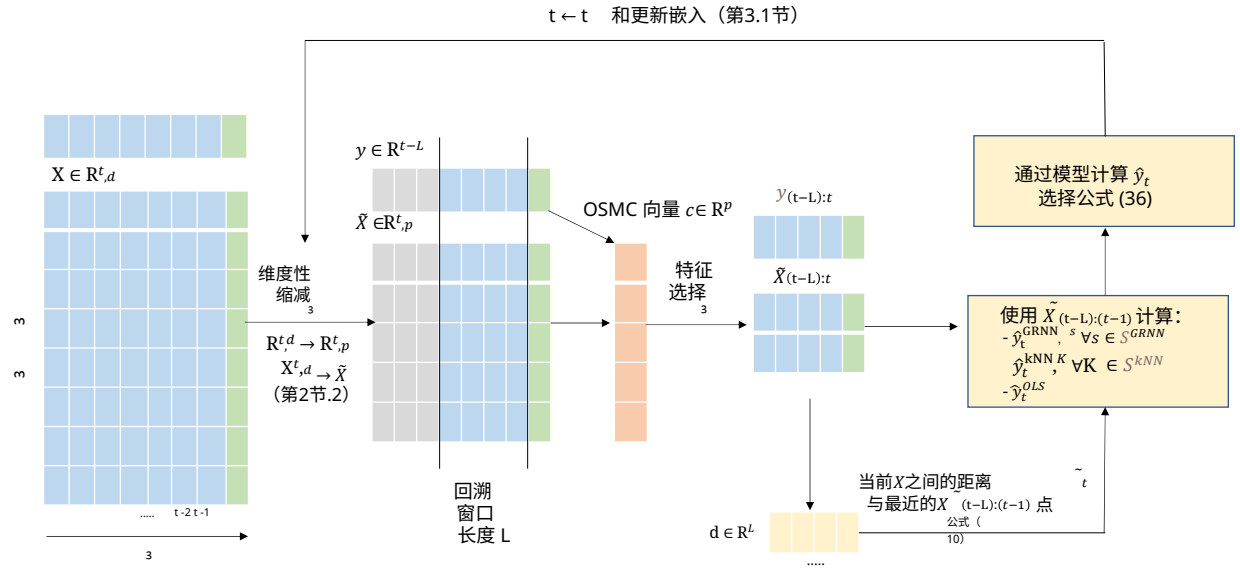


图*1:* *OFTER* 的示意图，其中展示了其迭代部分的主要组成部分 *d.*

3.1 嵌入的在线更新

为使算法能够在线运行，随着新数据到来，嵌入需要随之更新。初始嵌入将基于前 $L_0 < T$ 个数据点计算，之后在每个时间步更新。然而，每次迭代都重新计算嵌入的开销可能相当大（事实上，每一步的复杂度为 $O(nd^2 + d^3)$）。对于 PCA，有一种计算效率较高的方法可以更新 U 和 Λ。具体而言，若 $C(X_t)$ 表示由前 t 个数据点构成的协方差矩阵，则可以计算更新后的协方差矩阵的特征向量

$$C(X_{1:t+1}) = \frac{t-1}{t} C(X_{1:t}) + \frac{1}{t} ||x_{t+1}||^2 \frac{x_{t+1}}{||x_{t+1}||} \frac{x_{t+1}^T}{||x_{t+1}||}, \tag{25}$$

通过使用先前计算得到的 $C(X_{1:t})$ 特征向量（目前我们假设数据已中心化）。[37] 中提出的在线算法正是如此，其在每次迭代中的复杂度为 $O(p^2 d)$（回想一下，p 表示所选的嵌入维度）。

上述问题是一个更一般问题的实例，即特征系统的秩一更新 [38]。再次考虑对称矩阵 A，其特征分解已知，其中 $\lambda_1 \geq \cdots \geq \lambda_d$ 表示其特征值，并令u_i为对应于 λ_i，$\forall i \in \{1, \ldots, d\}$ 的特征向量，使得 $A = U \Lambda U^T$。当前任务是利用已知的 A的特征系统，求出 $A + \rho v v^T$的特征系统。正如 [38], 所示，$A + \rho v v^T$的特征值是世俗方程的根 κ_1，……，κ_d

$$1 + \rho \sum_{i=1}^{d} \frac{z_i^2}{\lambda_i - \kappa} = 0, \text{ where } z = U^T v. \tag{26}$$

计算出更新后的特征值后，每个更新后的特征向量 u_k 可计算为

$$u_k = \frac{U (\Lambda - \kappa_k I)^{-1} z}{||U (\Lambda - \kappa_k I)^{-1} z||}. \tag{27}$$

更进一步，可以通过仅更新前几个特征值来减少计算量。[39], 给出了解的一个良好近似，并展示了如何通过求解来实现这一点。

$$1 + \rho \left(\sum_{i=1}^{p} \frac{z_i^2}{\lambda_i - \kappa} + \frac{1 - \sum_{i=1}^{p} z_i^2}{\mu - \kappa} \right) = 0. \tag{28}$$

该方程的解被证明可以近似前 m 个特征值，误差量级为$O(\max_{m+1 \leq i \leq p} |\lambda_i - \mu|)$。请注意，$\mu$ 是另一个需要选取的参数，其取值应使

$O(\max_{m+1\le i\le p}|\lambda_i-\mu|)$，例如 μ =中位数(λ_{m+1}，.....， λ_p)。参数 p已在算法较早部分通过方程（9）选定，并且在更新步骤中不会改变。尽管已有多种技术可用于求解方程（28），我们采用了文献 [40],所述的凸化方法，并借鉴了 [38]的早期工作。

请注意，最初的特征矩阵是基于初始的 L_0 个条目进行标准化的（即，我们将 $X_{(1:L_0)}$ 标准化）。将 $X_{1:t}$ 的均值向量记为 $\mu_t\in\mathbb{R}^d$，将正确中心化的 $X_{1:t}$ 记为 $\bar{X}_{1:t}\in\mathbb{R}^d$。假设 $X_{(1:t-1)}$ 已通过 μ_{t-1}进行中心化，我们需要基于 μ_t 重新中心化。按照 [37], , 更新形式为 $X_{1:t}=X_{1:t}+e_t(\mu_{t-1}-\mu_t)^T$。事实证明，我们可以通过对协方差矩阵再进行两次秩一更新来调整重新中心化，具体方法是将 $\bar{X}_{1:t}^T\bar{X}_{1:t}$ 表示为以下形式。

$$\begin{aligned}
\bar{X}_{1:t}^T\bar{X}_{1:t} &= X_{1:t}^TX_{1:t}+X_{1:t}^Te_N(\mu_{t-1}-\mu_t)^T+(\mu_{t-1}-\mu_t)e_N^TX_{1:t}+t(\mu_{t-1}-\mu_t)(\mu_{t-1}-\mu_t)^T\\
&= X_{1:t}^TX_{1:t}+\begin{bmatrix}\mu_{t-1}-\mu_t & X_{1:t}^Te_N\end{bmatrix}\begin{bmatrix}t & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}(\mu_{t-1}-\mu_t)^T\\ e_N^TX_{1:t}\end{bmatrix}\\
&= X_{1:t}^TX_{1:t}+\begin{bmatrix}\mu_{t-1}-\mu_t & X_{1:t}^Te_N\end{bmatrix}\begin{bmatrix}\frac{\rho_1}{\sqrt{1+\rho_1^2}} & \frac{\rho_2}{\sqrt{1+\rho_2^2}}\\ \frac{1}{\sqrt{1+\rho_1^2}} & \frac{1}{\sqrt{1+\rho_2^2}}\end{bmatrix}\begin{bmatrix}\rho_1 & 0\\ 0 & \rho_2\end{bmatrix}\begin{bmatrix}\frac{\rho_1}{\sqrt{1+\rho_1^2}} & \frac{\rho_2}{\sqrt{1+\rho_2^2}}\\ \frac{1}{\sqrt{1+\rho_1^2}} & \frac{1}{\sqrt{1+\rho_2^2}}\end{bmatrix}^T\begin{bmatrix}(\mu_{t-1}-\mu_t)^T\\ e_N^TX_{1:t}\end{bmatrix}\\
&= X_{1:t}^TX_{1:t}+\begin{bmatrix}\frac{\rho_1(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_1^2}} & \frac{\rho_2(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_2^2}}\end{bmatrix}\begin{bmatrix}\rho_1 & 0\\ 0 & \rho_2\end{bmatrix}\begin{bmatrix}\frac{\rho_1(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_1^2}}\\ \frac{\rho_2(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_2^2}}\end{bmatrix}\\
&= X_{1:t}^TX_{1:t}+\rho_1b_1b_1^T+\rho_2b_2b_2^T,
\end{aligned}$$

其中，在上述内容中

$$\rho_1=\frac{t-\sqrt{t^2+4}}{2},\tag{29}$$

$$\rho_2=\frac{t+\sqrt{t^2+4}}{2},\tag{30}$$

$$b_1=\frac{\rho_1(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_1^2}},\tag{31}$$

$$b_2=\frac{\rho_2(\mu_{t-1}-\mu_t)+X_{1:t}^Te_N}{\sqrt{1+\rho_2^2}}.\tag{32}$$

3.2 缩放因子的选择

k近邻的超参数 k （在5）和 h （在4）中的超参数对于各自算法的性能都至关重要。注意，当 $h\to\infty$时，$w_i\to 1/T$ （方程（4））对所有 i均成立。另一方面，当 $h\to 0$时，广义回归神经网络等价于1近邻（k近邻，其中 $k=1$）。为了展示缩放因子对预测任务的影响，我们考虑以下玩具模型

$$y_t=0.5+\cos\left(\frac{\pi}{64}t\right)+\epsilon_t,\text{where }\epsilon_t\sim N(0,\sigma^2).\tag{33}$$

在图2 中，我们展示了缩放因子的选择如何影响所得拟合的特性。对于较小的数值h，所得拟合接近于给出等于均值的常数拟合值；而对于较大的缩放因子，所得拟合几乎与观测数据点完全吻合。[6]

为选择合适的数值 h， [31] 提出采用留出法：每次移除一个样本点，计算损失函数并取其均值，以此确定最优参数。更现代的应用则采用K折交叉验证；这些方法也可以针对时间序列数据进行调整（关于不同方法的讨论，见 [41] ）。

为消除任何尺度依赖性，我们将 h 重新参数化为

$$h=\frac{\text{median}(\{||x_j-x'||\}_{j=1}^T)}{s}.\tag{34}$$

[6]请注意，这只是这个特定玩具数据集中的一种模式，并非普遍适用的模式

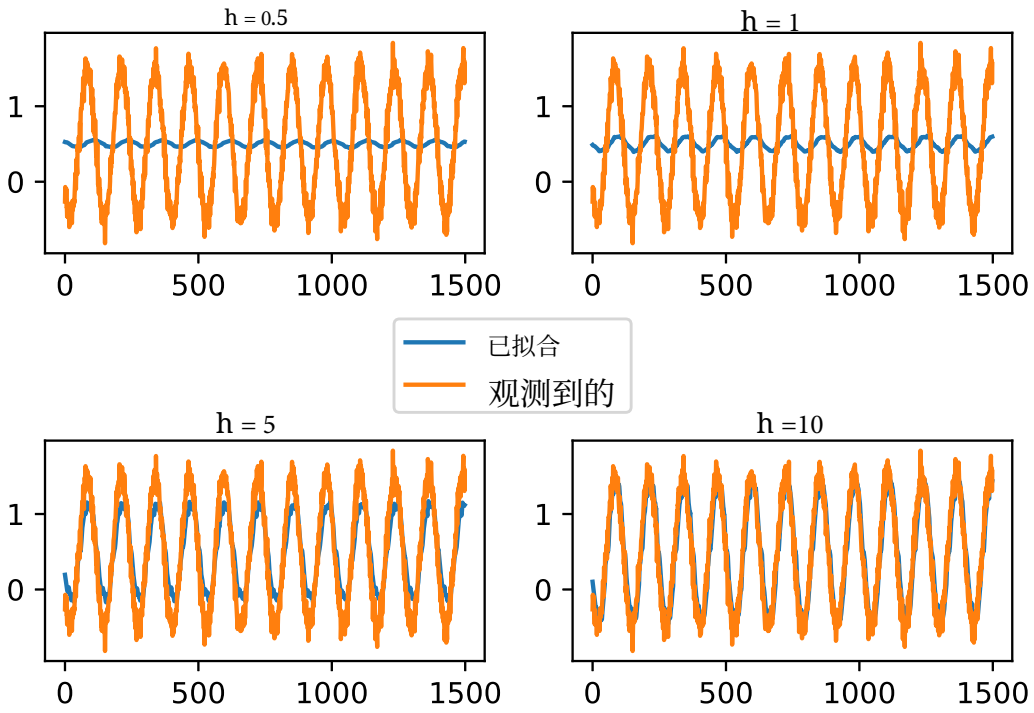


图*2*：使用不同缩放参数的广义回归神经网络（*GRNN*）模型时预测值的可视化(33)。在这类高信噪比问题中，随着 h 变大，*GRNN* 几乎能够完美地预测未来数值。

由于这是一个在线算法，我们可以据此更新缩放参数。我们不从候选集合中选择单个带宽参数，而是使用每个候选带宽拟合不同的模型，并在每个时间点更新所有模型的表现。对于当前时间点的预测值，选择截至上一个时间点表现最佳的模型。[7]

可以通过使用损失函数来选择缩放因子。如果第 t 天对应于缩放参数 s 的拟合值为 $\hat{y}_t^s$，则预测值 $\hat{y}_t$ 的计算方式为

$$\hat{y}_t = \sum_{s=1}^{|S|} \eta_t^s \hat{y}_t^s, \text{ where } \eta_t^s = \begin{cases} 1, & R_t^s = \min(R_t^1, \ldots, R_t^S) \\ 0, & \text{otherwise} \end{cases}. \tag{35}$$

通过 R_t^s，我们定义与参数 s对应的损失，即 $R_t^s = \sum_{i=1}^{t-1}$ 损失$(\hat{y}_t^s, y_t)$。对于一般应用，损失函数可以是均方误差（MSE）或平均绝对误差（MAE）。对于未来收益预测，我们使用负损益作为损失函数（我们在第 4.3 节定义损益）。

同样，对于 k近邻算法，较大的 k 会产生与使用较小 h时相似的预测值。我们不单独使用广义回归神经网络（GRNN）或 k近邻，而是同时使用这两种模型，各自配备超参数候选集，然后按照与（35）类似的方式，根据性能进行选择。除了这两类模型，我们还考虑了标准的普通最小二乘模型。

用 S^{GRNN} 表示广义回归神经网络的超参数候选值，用 S^{kNN} 表示 k近邻的超参数候选集。用 $\hat{y}^{\text{GRNN}}_{,t}{}^{s}$ 表示使用参数 s 的广义回归神经网络得出的预测值，用 $\hat{y}^{\text{kNN}}_{,t}{}^{k}$ 表示使用 k近邻（参数为 k）得出的预测值，

[7] 交替 从另一个角度来看，我们可以取一个 w 每个模型预测值的加权平均，但 t 性能较差。

用 $\hat{}y_t^{\text{OLS}}$ 表示普通最小二乘模型得出的预测值。接着定义

$$\begin{aligned} R_t^{\text{GRNN},s} &= \sum_{i=1}^{t-1} \text{Loss}(\hat{y}_i^{\text{GRNN},s}, y_i). \\ R_t^{\text{kNN},k} &= \sum_{i=1}^{t-1} \text{Loss}(\hat{y}_i^{\text{kNN},k}, y_i). \\ R_t^{\text{OLS}} &= \sum_{i=1}^{t-1} \text{Loss}(\hat{y}_i^{\text{OLS}}, y_i). \end{aligned} \tag{36}$$

根据截至时间点 t的损失，最小损失计算如下：

$$R_t^{\text{optimal}} = \min(\{R_t^{\text{GRNN},s} : s \in S^{\text{GRNN}}\} \cup \{R_t^{\text{kNN},k} : k \in S^{\text{kNN}}\} \cup \{R_t^{\text{OLS}}\}), \tag{37}$$

达到最优性能的模型数量计算如下：

$$N_t^{\text{optimal}} = |\{s : R^{\text{GRNN},sx} = R_t^{\text{optimal}}\}| + |\{k : R^{\text{kNN},k} = R_t^{\text{optimal}}\}| + \mathbb{1}(R_t^{\text{OLS}} = R_t^{\text{optimal}}), \tag{38}$$

其中 $\mathbb{1}(A)$ 是一个指示函数，取值为 1，当条件 A 成立时；若不成立，则 0。随后，我们重新表述式（35），将 k近邻和普通最小二乘模型纳入其中：

$$\hat{y}_t = \sum_{s \in S^{\text{GRNN}}} \eta_t^{\text{GRNN},s} \hat{y}_t^{\text{GRNN},s} + \sum_{k \in S^{\text{kNN}}} \eta_t^{\text{kNN},k} \hat{y}_t^{\text{kNN},k} + \eta_t^{\text{OLS}} \hat{y}_t^{\text{OLS}}, \tag{39}$$

其中

$$\begin{aligned} \eta_t^{\text{GRNN},s} &= \begin{cases} \frac{1}{N_t^{\text{optimal}}}, & R_t^{\text{GRNN},s} = R_t^{\text{optimal}} \\ 0, & \text{otherwise} \end{cases}, \\ \eta_t^{\text{kNN},k} &= \begin{cases} \frac{1}{N_t^{\text{optimal}}}, & R_t^{\text{kNN},k} = R_t^{\text{optimal}} \\ 0, & \text{otherwise} \end{cases}, \\ \eta_t^{\text{OLS}} &= \begin{cases} \frac{1}{N_t^{\text{optimal}}}, & R_t^{\text{OLS}} = R_t^{\text{optimal}} \\ 0, & \text{otherwise} \end{cases}. \end{aligned} \tag{40}$$

3.3 克服k近邻和广义回归神经网络的不足

应用降维方法的一个缺点是，这些方法不考虑要预测的目标值。在转换初始空间、以较少的特征保留较大比例的变异性时，某个原本与目标值高度相关、但与其他特征相关性较弱的特征，最终可能在预测步骤中获得相对较小的权重。为克服这一不足，我们增加了两个步骤。

首先，我们确保原始特征空间中与目标值相关性足够高的所有特征 X 都保留在预测步骤中。这样，我们就能确保在选择最能捕捉变异性的成分时，不会因此舍弃那些本身具有预测能力的原始特征。因此，在计算每个原始特征与目标值之间的 OSMC 后，我们定义以下集合

$$I^{\text{Original}} = \{j : \text{OSMC}(X_{1:L_0,j}, y_{1:L_0}) \geq c_{\text{Original}}\}. \tag{41}$$

~ ~在算法的预处理阶段，我们将 X 替换为 $(X, X_{1:T,I^{\text{Original}}})$，特征数量等于 $p + |I^{\text{Original}}|$。请注意，对于中等规模、信号与误差比偏低的数据集（例如我们在第 4.3 节使用的金融数据集），I_{Original} 通常为空。

其次，嵌入空间中的某些特征应当对邻居选择产生更大的影响。这可以通过使用加权度量而非标准欧几里得范数来实现。具体而言，若 v 表示权重向量，则 X_{t_1} 与 X_{t_2} 之间的加权欧几里得距离为

$$||\tilde{X}_{t_1} - \tilde{X}_{t_2}||_v = \sqrt{\sum_{j=1}^{p+|I^{\text{Original}}|} v_j (\tilde{X}_{t_1,j} - \tilde{X}_{t_2,j})^2}. \tag{42}$$

权重基于每个特征与目标值之间的 OSMC

$$v_j = \frac{c_j^2 \mathbb{1}(|c_j| \geq c_{\min})}{\sum_{l=1}^{p+|I^{\text{Original}}|} c_l^2 \mathbb{1}(|c_l| \geq c_{\min})}, \tag{43}$$

其中 $c_j = \text{OSMC}(\tilde{X}_{1:L_0,\ j},\ y_{1:L_0})\ \ \forall j \in \{1, \ldots, p, p+1, \ldots, |I^{\text{Original}}|\}$。

3.4 算法概述

为确保目标值 y平稳，我们首先以显著性水平 p_{ADF}进行 ADF 检验；如果零假设未被拒绝，则对时间序列进行差分，并重复此过程，直到零假设被拒绝。随后，我们进行自相关函数和偏自相关函数检验，以检查是否存在显著的 ARMA 效应。[8] 如果识别出此类效应，我们将对自回归和移动平均阶数（分别为 p 和 q）进行网格搜索，以找到最优的 ARIMA(p,r,q) 模型，其中 r表示差分的次数。否则，我们直接拟合 ARIMA(0,r,0)。

使用此流程，我们将目标值 y 分解为时间序列和残差分量 t

$$y = \underbrace{y^{\text{TS}}}_{\text{Time series component}} + \underbrace{\epsilon}_{\text{Residual}}. \tag{44}$$

总体而言，该算法包括三个主要步骤：(1) 计算嵌入，(2) 根据 OSMC 选择特征，以及 (3) 基于 k近邻和广义回归神经网络进行预测值计算。完整算法见算法 2。

算法 1： 时间点 t的预测

输入： $\tilde{X}$ (特征矩阵)、 y (目标值)、 L (回看窗口长度)、 v (距离向量)、 S^{kNN}， S^{GRNN}(参数候选集)**O**

输出: $\hat{y}_t$

1: 计算$d_{j,\ t} = ||\quad \tilde{X}_{t,\ I} - X_{j,I}||_v$对于 $j = t-L, \ldots, t-1$

2: **对于** $s \in S^{\text{GRNN}}$ **执行**

3: 设定 $h_s = \frac{\text{median}(\{d_{j,t}\}_{j=t-L}^{t-1})}{s}$

4: 计算 $w^s_{j,t} = \frac{\exp\left(\frac{-d_j^2}{h_s^{\ t}}\right)}{\sum_{m=t-L}^{t-1} \exp\left(\frac{-d_m^2}{h_s^{\ t}}\right)} \forall j \in \{t-L, \ldots, t-1\}$

5: 计算 $\hat{y}^{\text{GRNN},s} = \sum_{j=t-L}^{t-1} w^s_{j,t} y_i$

6: **结束循环**

7: **对于** $k \in S^{\text{kNN}}$ **执行**

8: 查找集合 $\text{NN}_k(\ \tilde{X}_{t,}\ )$ 的 k 个最近邻 $\tilde{X}_t$, 在 $\tilde{X}_{(t-L):(t-1)}$

9: 计算 $\hat{y}^{\text{kNN},\ k} = \frac{1}{k} - \sum_{i \in \text{NN}_k(\ \tilde{X}_t)} y_i$

10: **对每个结束**11: 计算 $\hat{y}^{\text{OLS}} = \beta_0 + \beta^T \tilde{X}_t$

12: 估计拟合值 $\hat{y}_t$ 使用（39）

4 实证结果

我们考虑并比较 OFTER 的四种不同配置，具体如下

- OFTER（不进行降维和特征变换）。这意味着设置 $X = X$，使用皮尔逊相关代替 OSMC，并跳过算法 2 中的第 4、5、6、7 和 17 步。
- OFTER-DR（进行降维但不进行特征变换）。这与算法 2 几乎相同，但我们使用皮尔逊相关代替 OSMC。

[8] 我们采用 0.001 的显著性水平。如果自相关函数或偏自相关函数的置信区间高于或低于零，我们就认为相应的滞后效应显著。

算法 2： OFTER

输入： X (特征矩阵)，y（目标值），L_0 (训练窗口长度)，L (回看窗口长度)，$c_{\min}$，c^{Original}(相关性阈值)，S^{kNN}，S^{GRNN} (参数候选集)，δ (保留方差的比例)，p_{ADF}(ADF 的 p值)，LOSS（损失函数）

输出： $\hat{y}$

1：使用第3.4节所述的流程，确定 y的最优 ARIMA(p,r,q) 模型 2：使用最优模型获取拟合值 y^{TS}。若选择 ARIMA(0,0,0)，则设置 $y^{\text{TS}} = (0, \ldots, 0)$ 3：变换 X，使其成为满列秩（通过 QR 分解）且标准化的矩阵 $X_{1:L_0}$ 4：获取 $X_{1:L_0}$的前 p 个主成分，其中 p 通过公式（9）选取，并设置 $U_p = (u_1, \ldots, u_p)$ 和 $\Lambda_p = (\lambda_1, \ldots, \lambda_p)$ 5：获取 $L_0 \times p$ 矩阵 $\tilde{X} = XU_{1:L_0,1:p}$ 6：设置 $I^{\text{Original}} = \{i : \text{OSMC}(X_{1:L_0,i}, y_{1:L_0}) \geq c^{\text{Original}}\}$ 7：将 X 替换为 $(X, X_{1:T,I^{\text{Original}}})$ 8：使用（36）计算 $R^{\text{GRNN}}_{,L_0}{}^{s} \; \forall s \in S^{\text{GRNN}}, R^{\text{kNN}}_{,L_0}{}^{k} \; \forall k \in S^{\text{kNN}}$ 和 $R^{\text{OLS}}_{L_0}$ 9：使用（40）计算 $\eta^{\text{GRNN}}_{,L_0}{}^{s} \; \forall s \in S^{\text{GRNN}}, \eta^{\text{kNN}}_{,L_0}{}^{k} \; \forall k \in S^{\text{kNN}}$ 和 $\eta^{\text{OLS}}_{L_0}$ 10：对于 $j \leftarrow 1$ 到 $p + |I^{\text{Original}}|$，执行 11：计算 $c_j = \text{OSMC}(X_{1:L_0,j}, y_{1:L_0})$ 12：设置 $v_j = c_j^2 \mathbb{1}(|c_j| \geq c_{\min}) \sum_{l=1}^{p+|I^{\text{Original}}|} c_l^2 \mathbb{1}(|c_l| \geq c_{\min})$ 13：循环结束 14：使用普通最小二乘回归，将 $y_{1:L_0}$ 对 $\tilde{X}_{1:L_0}$进行回归，得到系数向量 β 和截距 $\beta_0$15：对于 $t \leftarrow L_0 + 1$ 到 D，执行 16：使用算法 1 计算 $\hat{y}_t$ 17：使用世俗方程法更新 U_p 和 Λ_p（详见第3.1节）18：如果 t 可被 100 整除，则使用 $X_{(t-L):t}$ 和 $y_{(t-L):t}$重新拟合普通最小二乘模型，得到 β 和 β_0 19：循环结束 20：将 $\hat{y}$ 更新为 $\hat{y} + y^{\text{TS}}_{(L_0+1)} : D$

- OFTER-FT（进行特征变换且不降维）。这意味着设置 $\tilde{X} = X$，并跳过算法2中的步骤 4、5、6、7 和 17。

- OFTER-DR-FT（进行特征变换和降维）。这就是算法2所给出的完整算法。

除非另有说明，我们使用 $p_{\text{ADF}} = 0.05$ $\delta = 0.9$。对于候选集，我们使用 $S^{GRNN} = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$ 和 $S^{kNN} = \{1, 2, 3, 5, 10, 15, 20, 30, 50\}$。在特征方面，我们纳入对应于当前时期、滞后阶数最多为3的特征。最后，我们令 L_0 等于 $\lfloor 0.7T \rfloor$。如附录C中的稳健性分析所示，OFTER 对超参数的不同取值通常具有较强的稳健性。

4.1 基线模型

我们将 OFTER 的表现与几种主流机器学习模型进行比较。我们纳入了线性模型 OLS、PCR、LASSO，也纳入了广泛使用的 SVM 和随机森林。此外，我们还纳入了一些最常用的、针对时间序列量身打造的深度学习模型，例如长短期记忆网络、双向长短期记忆网络和门控循环单元。与 OFTER 不同，这些方法中的大多数并不天然适合在在线环境中高效实现，因此无法更新。为便于比较，我们每隔 100 步重新训练这些模型。对于长短期记忆网络、双向长短期记忆网络和门控循环单元，我们从先前的权重矩阵开始，仅用五个训练轮次重新拟合模型。为确保比较公平，我们没有采用嵌入更新步骤，而是每隔 100 步重新计算 OFTER 的嵌入。附录中概述了这些方法，并介绍了它们在此预测任务中的具体实现细节 B。

4.2 合成数据集

在真实数据集上测试 OFTER 之前，我们先检查它能否在一些简单的低维玩具模型上始终保持良好表现。为此，采用了 [25], 使用的三个五维模型。这些模型较为简单，经过设计以模拟脑电图信号，适合作为此预测任务的数据集。

$$\text{M1:}\begin{cases} y_t^{(1)} = 0.2y_{t-1}^{(1)} - 0.4y_{t-1}^{(2)} + \epsilon_t^{(1)} \\ y_t^{(2)} = -0.5y_{t-1}^{(1)} + 0.15y_{t-1}^{(2)} + \epsilon_t^{(2)} \\ y_t^{(3)} = -0.14y_{t-1}^{(2)} + \epsilon_t^{(3)} \\ y_t^{(4)} = 0.5y_{t-1}^{(1)} - 0.25y_{t-1}^{(2)} + \epsilon_t^{(4)} \\ y_t^{(5)} = 0.15y_{t-1}^{(1)} + \epsilon_t^{(5)} \end{cases} \tag{45}$$

方程（45）中所示的第一个数据集（M1）由一组五个线性方程生成。

$$\text{M2:}\begin{cases} y_t^{(1)} = 3.4y_{t-1}^{(1)}\left(1-\exp\left(-\left(y_{t-1}^{(1)}\right)^2\right)\right)\exp\left(-{y_{t-1}^{(1)}}^2\right) + \epsilon_t^{(1)} \\ y_t^{(2)} = 3.4y_{t-1}^{(2)}\left(1-\exp\left(-\left(y_{t-1}^{(2)}\right)^2\right)\right)\exp\left(-{y_{t-1}^{(2)}}^2\right) + 0.5y_{t-1}^{(1)}y_{t-1}^{(2)} + \epsilon_t^{(2)} \\ y_t^{(3)} = 3.4y_{t-1}^{(3)}\left(1-\exp\left(-\left(y_{t-1}^{(3)}\right)^2\right)\right)\exp\left(-{y_{t-1}^{(3)}}^2\right) + 0.3y_{t-1}^{(2)} + 0.5{y_{t-1}^{(1)}}^2 + \epsilon_t^{(3)} \\ y_t^{(4)} = 0.5y_{t-1}^{(1)} - 0.25y_{t-1}^{(2)} + \epsilon_t^{(4)} \\ y_t^{(5)} = 0.15y_{t-1}^{(1)} + \epsilon_t^{(5)} \end{cases} \tag{46}$$

方程（46）中所示的第二个数据集（M2）由一组五个方程生成，其中一些是非线性的。[25] 这些方程基于一些出现在 [42]中的模型。

$$\text{M3:}\begin{cases} y_t^{(1)} = 0.1y_{t-1}^{(1)} - 0.6y_{t-3}^{(2)} + \epsilon_t^{(1)} \\ y_t^{(2)} = -0.15y_{t-3}^{(1)} + 0.8y_{t-3}^{(2)} + \epsilon_t^{(2)} \\ y_t^{(3)} = -0.45y_{t-3}^{(2)} + \epsilon_t^{(3)} \\ y_t^{(4)} = 0.45y_{t-3}^{(1)} - 0.85y_{t-3}^{(2)} + \epsilon_t^{(4)} \\ y_t^{(5)} = 0.95y_{t-2}^{(1)} + \epsilon_t^{(5)} \end{cases} \tag{47}$$

第三个 数据集（M3）与 M1 相似，但属于更高阶的时间序列，因此 M3 是一个向量自回归模型 .

在上述每种情况下， $\epsilon_t^{(1)}, \ldots, \epsilon_t^{(5)}$ 都是方差为单位方差的不相关正态分布随机变量。$y_1^{(1)}, \ldots, y_1^{(5)}$ 也都等于零。

4.3 真实数据集

金融股票数据集

我们考虑一个用于两种预测应用的金融数据集，即日收益率和成交量。我们考虑由 210 种金融工具构成的样本集合，其中包括 10 只 ETF（SPY、XLF、XLB、XLK、XLV、XLI、XLU、XLY、XLP、XLE）以及 200 只来自金融行业的股票。

设 $P_{i,d}$ 表示资产 i 在第 d 天的收盘价。我们考虑资产 i的收盘到收盘收益率[9]，定义如下

$$R_{i,d}^k = \frac{P_{i,d+k} - P_{i,d}}{P_{i,d}}. \tag{48}$$

作为一种更具代表性的指标，并且扎根于金融应用，我们考虑市场超额收益率，即某一金融工具的收益与标普500指数收益之间的差值。标普500指数是市场收益率的常用代理指标。

$$\tilde{R}_{i,t}^k = \tilde{R}_{i,t}^k - \tilde{R}_{\text{SPY},t}^k. \tag{49}$$

[9]T所考虑的收益已根据拆股和分红进行调整 ends

将金融工具 i 的成交量记为 $V_{i,\ t}$，其定义为在第 t天交易的股票总数。请注意，这一指标是非平稳的，并且对于大多数金融工具而言，通常会随时间推移而增加。为此，我们改为考虑“对数成交量收益”，其定义为

$$\delta_{i,t}^{k} = \frac{\log(V_{i,t+k}) - \log(V_{i,t})}{\log(V_{i,t})}. \tag{50}$$

预测日收益率是金融领域最受欢迎、也最困难的问题之一。由于这项任务的信噪比较低，均方误差和相关性等标准指标被认为并不适用。一种更合适、以经济效益为导向的评估方法，是根据预测值的符号，对标的资产进行模拟交易（其中 $+$ 和 $-$ 分别对应买入和卖出）。由于这仅用于评估，我们将忽略交易成本和流动性约束。

遵循这一假 设策略，我们考虑如下定义的（经超额收益调整的）损益（P&L）指标：

$$\mathrm{P\&L}_t^k = \frac{1}{N}\sum_{i=1}^{N_t} \widetilde{R}_{i,t}^{k} \times \operatorname{sign}(s_{i,t}), \tag{51}$$

上述损益是针对每一天定义的 d，由此得到一个向量 **损益** $=(\text{损益}_1, \text{损益}_2, \ldots, \text{损益}_{T-1})$，其中包含每日损益，用于衡量所考虑的整个时间跨度内模拟策略的结果。衡量策略长期表现的指标是年化夏普比率（以下简称 SR）[43], , 其定义如下：

$$\mathrm{SR} = \frac{\mathrm{mean}(\mathbf{P\&L}) \times \sqrt{252}}{\mathrm{sd}(\mathbf{P\&L})}. \tag{52}$$

最后，我们将每美元交易额对应的损益（PPD）定义为

$$\mathrm{PPD} = \frac{1}{T-1}\sum_{t=1}^{T-1} \mathrm{P\&L}, \tag{53}$$

which 假设每天分配给投资组合的总资本始终为 1 美元 .

我们采用 [44] 的 SR 显著性检验。由于已知收益呈现出比正态分布更厚的尾部，该检验是传统 t检验的扩展，并对收益分布的偏度和峰度进行了调整。

预测值的绝对值越大，自然意味着置信度越高。对于每个交易日，我们根据预测值绝对值的分位数（$\{|s_{i,t}|\}_{i=1}^{N}$）定义五个规模依次递减的五分位投资组合。具体而言，Q1 是包含所有资产的投资组合；Q2 是包含绝对值最高的 80% 资产的投资组合；依此类推，直至 Q5，其包含绝对值最高的 20% 资产。

The 特征矩阵由以下指标组成 cators

- 1各金融工具的 2 日和 3 日收益，即$R_{i,\ t}^{1}$, $R_{i,\ t}^{2}$, $R_{i,\ t}^{3}$
- 1各金融工具的 2 日和 3 日绝对收益率，即$|R_{i,\ t}^{1}|$, $|R_{i,\ t}^{2}|$, $|R_{i,\ t}^{3}|$
- 1各金融工具的 2 日和 3 日对数成交量收益率，即$\delta_{i,\ t}^{1}$, $\delta_{i,\ t}^{2}$, $\delta_{i,\ t}^{3}$
- 三个 Fama-French 日度市场因子
- 49 个不同行业投资组合的日收益率[10]
- VIX 指数的日收益率

我们基于上述所有特征构建嵌入。对于每种金融工具 i 和日期 t，我们将信号 $s_{i,\ t}$ 设为 OFTER 算法的输出。

空气质量数据集

作为一项非金融应用，我们还考虑了来自 UCI 机器学习知识库的北京空气质量数据集。数据由北京市环境监测中心收集，涵盖 12 个监测点的各类空气质量测量值。我们的目标是预测各监测站六种污染物（PM2.5、PM10、03、C0、S02、N02）的浓度（单位为 ug/m^3）。数据集中的其他特征（用作预测变量）包括温度、露点温度、降水量、气压，以及风速和风向。

[10]行业投资组合收益和每日市场因子直接取自 Kenneth R. French 图书馆 [45]。基于市场投资组合，这三个因子包括根据公司规模构建的小市值减大市值（SMB）、根据账面市值比构建的高账面市值比减低账面市值比（HML），以及该投资组合的市场超额收益率。

4.4 数值结果

数据集	T	d	LM	主成分回归	最小绝对收缩和选择算子	随机森林	支持向量机	长短期记忆网络	双向长短期记忆网络	GRU	OFTER	OFTER-DR	OFTER-FT	OFTER-DR-FT
合成	3000	15	0.01	0,02	0.64	33.90	1.81	173.45	202.99	184.81	7.87	9.68	37.52	24.53
金融	5022	154	0.37	0.48	5.38	3,144.02	0.17	442.86	503.78	549.30	17.58	22.25	222.87	153.06
空气质量	1461	228	0.10	0.31	2.23	132.05	1.00	165.48	152.22	184.85	3.25	20.18	157.95	62.64

表*1*： 三种多元时间序列中所有预测量的*12*种方法的平均运行时间。T 表示时间序列的长度，而 d 表示每个数据集的维度。

我们首先讨论 OFTER 的运行时间。在表1中，我们列出了所考察的三个数据集上算法的平均运行时间。可以看到，OFTER 的四种配置都明显快于长短期记忆网络、双向长短期记忆网络和门控循环单元，同时也如预期般明显慢于普通最小二乘法、最小绝对收缩和选择算子、主成分回归和支持向量机。在 OFTER 的各项配置中，OFTER-FT 最慢，因为它会分别转换每个特征，且不会先降低特征空间的维度。OFTER 和 OFTER-DR 的时间复杂度较为接近，后者的计算开销略高。

对于收益预测相关的数值实验，我们采用该领域常用的经济驱动指标，具体讨论见第 4.3 节。对于其余所有实验，我们以（皮尔逊）相关性指标为基础进行性能比较，因为与均方误差及其他类似指标相比，该指标更容易解释。[11]

- **合成数据**

对于这项任务，我们使用 $L = 800$，$C_{\text{min}} = 0.05$ 和 $C_{\text{Original}} = 0.05$。在这项实验中，OFTER 的表现始终良好。在几乎所有特征上，OFTER 的某种配置都位列表现最好的两种算法之中（GRU 通常表现非常出色）。

合成数据集上的结果证实了我们模型的预期。在 M1 上，如图 3 所示，GRU 在五个特征中的四个上似乎优于其他方法。紧随 GRU 之后，OFTER 和 OFTER-DR 在五个特征中的三个上似乎表现最佳。

对于 M2（一个非线性多变量非线性模型），采用特征变换组件的 OFTER 配置在三个方面优于所有其他模型，随机森林除外

五个特征中的（尤其是以非线性方式定义的三个特征），如图 4所示。最后，在采用向量自回归（VAR）模型的 M3 情况下，OFTER 的表现尤为

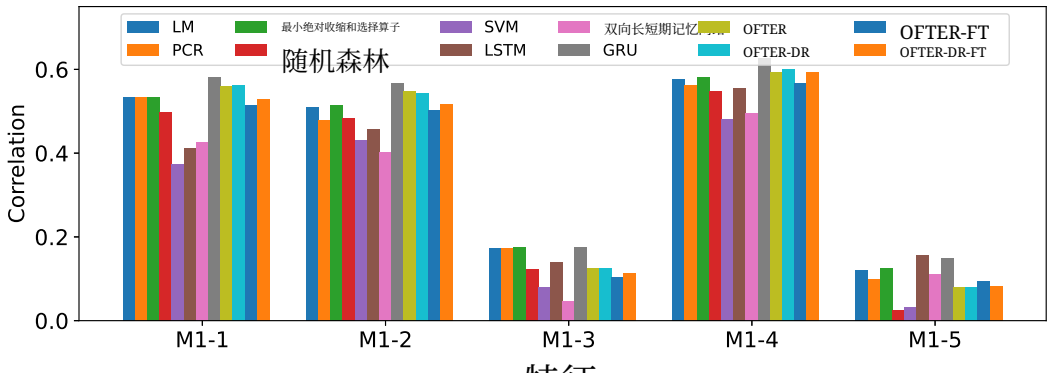


图*3*： 分组柱状图展示了所考察的*12*种方法在*M1*数据集的五个特征上的性能比较，该数据集详见 (45)。比较指标为目标值与预测值之间的相关性。

[11]请注意，如果改用均方误差或平均绝对误差，不同模型的性能比较结果几乎相同。

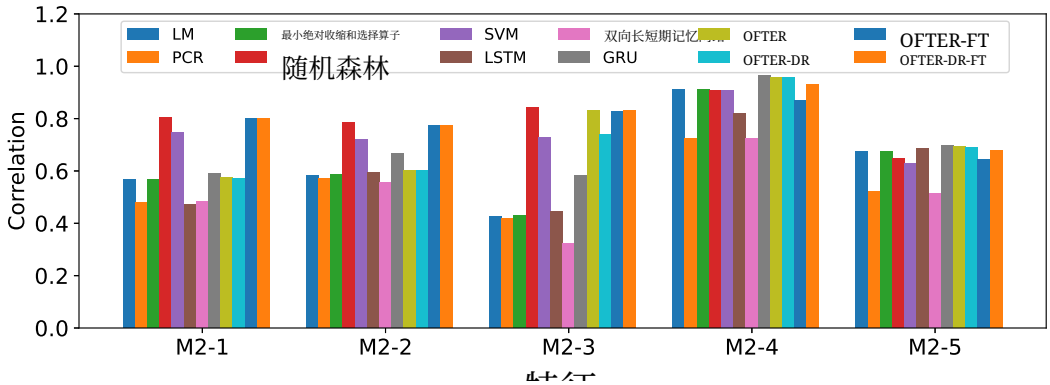


图*4*： 分组柱状图展示了所比较的*12*种方法在*M2*数据集的五个特征上的性能比较，该数据集详见(46)。比较指标为目标值与预测值之间的相关性。

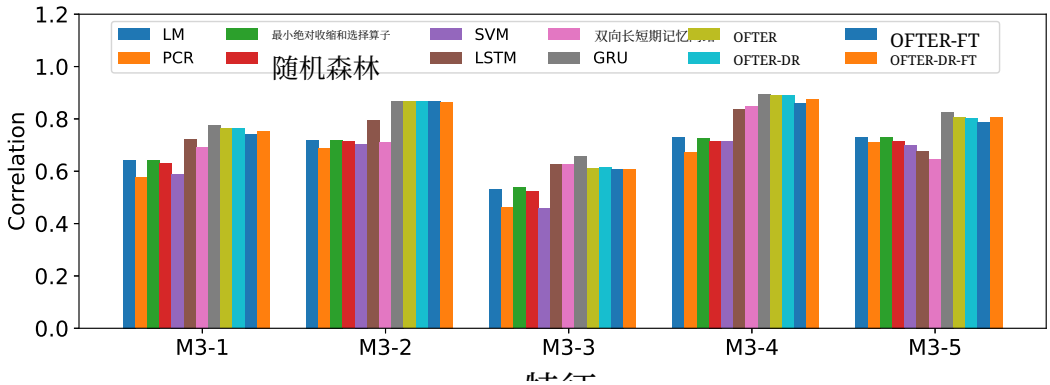


图*5*： 分组柱状图展示了所比较的*12*种方法在*M3*模型的五个特征上的性能比较，该模型详见(47)。比较指标为目标值与预测值之间的相关性。

出色，除 GRU 外，优于所有其他模型，如图 5所示。与 M1 类似，未采用特征变换组件的两种 OFTER 配置表现略好。

- **交易所交易基金成交量**

对于这两项金融任务，我们使用 $L = 800$ 并设置 $C_{\text{min}} = C_{\text{Original}} = 0.01$，因为这类问题通常具有较低的信噪比。对于预测ETF工具的对数增量成交量这一任务（50），OFTER的表现优于所有其他基准方法。如图6所示，OFTER-FT和OFTER-DR两种配置在全部10只ETF上的表现均优于所有基线方法。

• ETF市场超额收益

我们模拟了一种日度策略，该策略基于对 Q1-Q5 投资组合交易十种 ETF工具，并在表 2中汇总结果。四种 OFTER 策略中，有三种的夏普比率具有统计显著性。[12] 作为结果的直观展示，我们还在图 8中绘制了各方法的 Q1策略累计损益，清楚地表明 OFTER-DR 和 OFTER-DR-FT 的表现优于其他方法。除 Q5-SVM策略外，其他基准方法均未产生夏普比率具有统计显著性的策略。在大多数分位数中，OFTER-DR 的表现均为最佳，无论是按夏普比率还是 PPD 衡量。在图 7中，我们绘制了

[12]使用 [44] 的夏普比率显著性检验。该检验基于对 t检验的推广，通过考虑损益分布的峰度和偏度，来处理收益的非正态性。

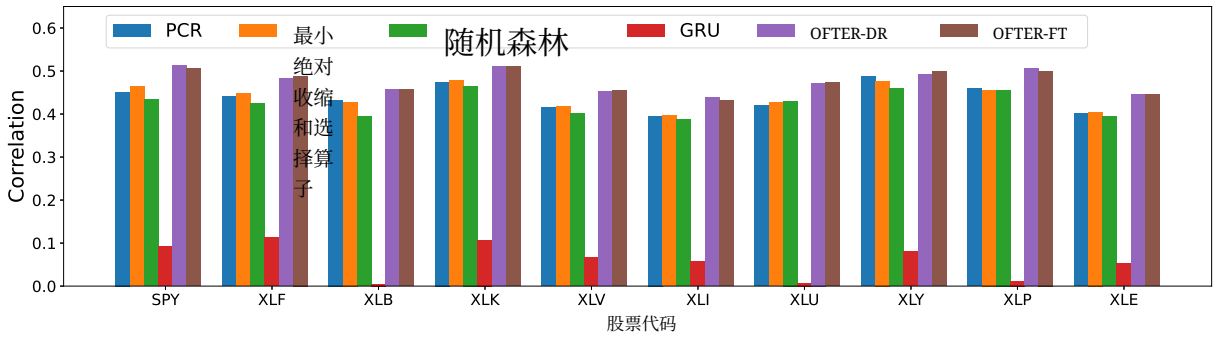


图*6*： 分组条形图展示了所选基线方法和 OFTER 配置中，增量成交量与其预测值之间的相关性。所选方法为各类别中表现最佳的方法。

	Q1			Q2			Q3			Q4			Q5		
	SR	PPD	p值	SR	PPD	p值	SR	PPD	p值	SR	PPD	p值	SR	PPD	p值
LM	-0.745	-1.211	0.967	-0.729	-1.270	0.964	-0.873	-1.783	0.984	-0.740	-1.689	0.965	-0.814	-2.664	0.977
PCR	-0.863	-1.324	0.981	-0.901	-1.523	0.985	-0.841	-1.737	0.979	-0.952	-2.234	0.989	-0.809	-2.728	0.976
最小绝对收缩和选择算子	-0.890	-1.293	0.985	-0.920	-1.411	0.987	-1.052	-1.905	0.995	-1.192	-2.466	0.998	-0.995	-3.163	0.993
随机森林	0.371	0.534	0.182	0.282	0.436	0.246	0.174	0.339	0.335	0.233	0.521	0.285	0.285	1.000	0.243
SVM	-0.370	-0.528	0.807	-0.314	-0.471	0.763	0.309	0.429	0.204	0.613	0.984	0.058	**0.706**	**1.267**	**0.039**
LSTM	0.308	0.433	0.227	0.261	0.396	0.262	0.421	0.771	0.152	0.403	0.818	0.163	0.059	0.176	0.442
双向长短期记忆网络	-0.810	-0.868	0.976	-0.824	-1.017	0.977	-0.388	-0.641	0.828	-0.238	-0.462	0.719	-0.456	-1.377	0.866
GRU	0.424	0.610	0.152	0.286	0.448	0.243	0.148	0.268	0.359	0.144	0.294	0.363	0.129	0.403	0.376
OFTER	0.534	0.811	0.089	0.653	1.082	0.052	0.616	1.203	0.064	0.290	0.646	0.233	0.444	1.463	0.140
OFTER-DR	**1.616**	**2.346**	**0.000**	**1.704**	**2.607**	**0.000**	**1.229**	**2.194**	**0.001**	**1.258**	**2.441**	**0.001**	**1.261**	**3.765**	**0.001**
OFTER-FT	**1.143**	**1.737**	**0.002**	**1.344**	**2.197**	**0.000**	**1.261**	**2.476**	**0.001**	**0.988**	**2.167**	**0.007**	0.491	1.545	0.110
OFTER-DR-FT	**1.468**	**2.125**	**0.000**	**1.532**	**2.409**	**0.000**	**1.562**	**2.795**	**0.000**	**0.884**	**1.760**	**0.015**	**0.826**	**2.371**	**0.022**

表*2*： 本表总结了*12*种模型在*ETF*价格走势预测任务中的表现。*SR*表示年化夏普比率，*PPD*表示每美元利润，*p*值表示通过 *[44]*的单尾夏普比率显著性检验获得的*p*值。*Q1-Q5*对应于第*4.3*节中定义的分位数组合。

OFTER-DR策略的累计损益，清楚地表明该策略组在不同分位数下均持续表现良好。

• 空气污染数据集

最后，对于空气污染任务，我们预测了12个监测站的共6个特征。在图9中，我们展示了一个网格，其中基于相关性呈现一种方法优于另一种方法的案例比例。可以看出，所有四种OFTER配置都优于其他八种基准方法。在这种情况下，OFTER-DR-FT优于OFTER的其他三个版本。

5 特征重要性与异常值检测的应用。

本节将讨论 OFTER 在机器学习领域两项值得关注的任务中的其他应用。具体而言，我们将介绍一个特征重要性组件，以增强 OFTER 的可解释性，并探讨其在异常值检测中的应用。

5.1 特征重要性

在 OFTER流程中，可以采用一种自然的方法来估计每个特征对预测任务的贡献。具体而言，我们旨在找到一个指标，用以衡量各输入特征的重要性。为此，考虑初始特征空间中的两个点 x_1 和 x_2，并将变换变量记为 $x_1' = U^T x_1$ 和 $x_2' = U^T x_2$。将降维空间中的距离（用于计算最近邻）记为 $D' = ||x_1' - x_2'||_v$，并令 $D_j = x_{1,j} - x_{2,j}$ 使得 $||x_1 - x_2||_2 = \sqrt{\sum_j D_j^2}$。

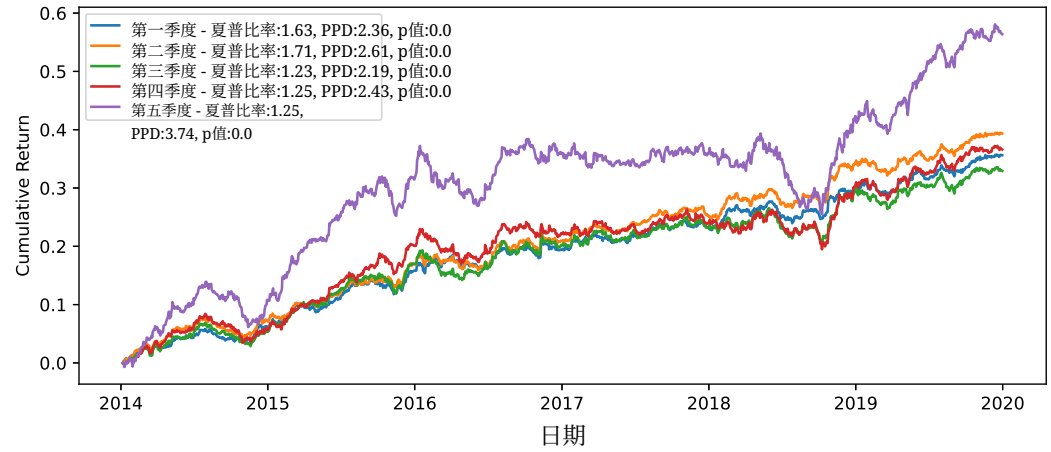


图*7*：*OFTER-DR*生成的每日策略的累计损益图。*SR*表示年化夏普比率，*PPD*表示每美元利润，*p*值表示通过对*[44]*进行单尾夏普比率显著性检验得到的*p*值。*Q1-Q5*对应于第*4.3*节中定义的分位数组合。

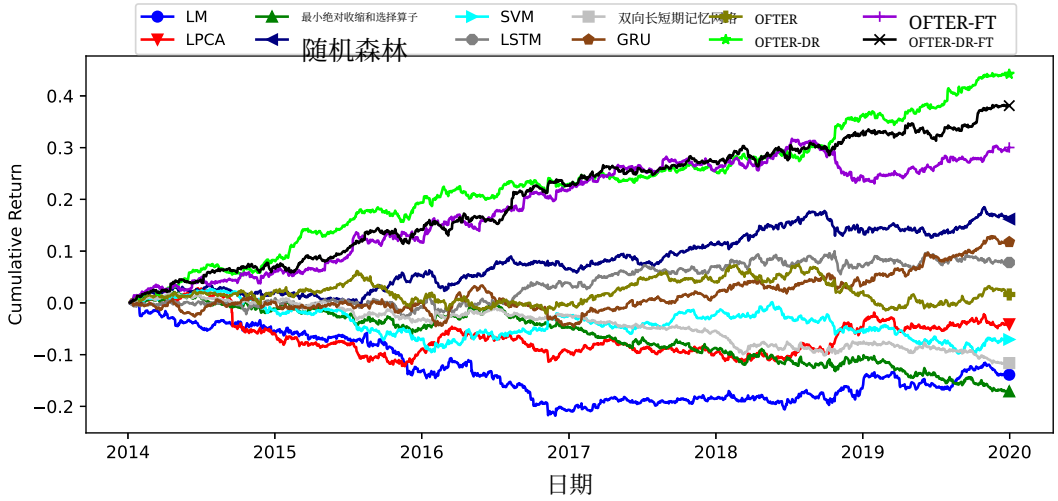


图*8*：*ETF*价格走势预测任务中对应*12*种方法的每日策略累计损益图。*SR*表示年化夏普比率，*PPD*表示每美元利润，*p*值表示通过对 *[44]*进行单尾夏普比率显著性检验得到的*p*值。各方法的绩效指标见表*2*。

衡量某个变量在决定嵌入空间中数据点相对距离时重要性的一种指标，可以建模为距离 D' 对单个距离 d_j 的敏感度，即 $\left|\frac{\partial D'}{\partial D_j}\right|$。对于我们的目标，自然应在满足 $D_1 = \cdots = D_d = 1$的点处求导。我们先写出

$$D'^2 = ||x_1' - x_2'||_v = ||U^T(x_1 - x_2)||_v = \sum_{j=1}^{p} v_j \left(\sum_{k=1}^{d} U_{k,j}(x_1 - x_2)_k\right)^2 = \sum_{j=1}^{p} v_j \left(\sum_{k=1}^{d} U_{k,j} D_k\right)^2. \tag{54}$$

然后我们计算

$$\frac{\partial D'^2}{\partial D_k}\Big|_{d=e_d} = 2\sum_{j=1}^{p} v_j \left(\sum_{k'=1}^{d} U_{k',j} D_{k'}\right) U_{k,j}, \tag{55}$$

其中，使用 $\odot$ 表示阿达马积，可写为

$$\nabla D'^2(e_d) = 2\left|U\ \left[(U^T e_d) \odot\ v\right]\right|. \tag{56}$$

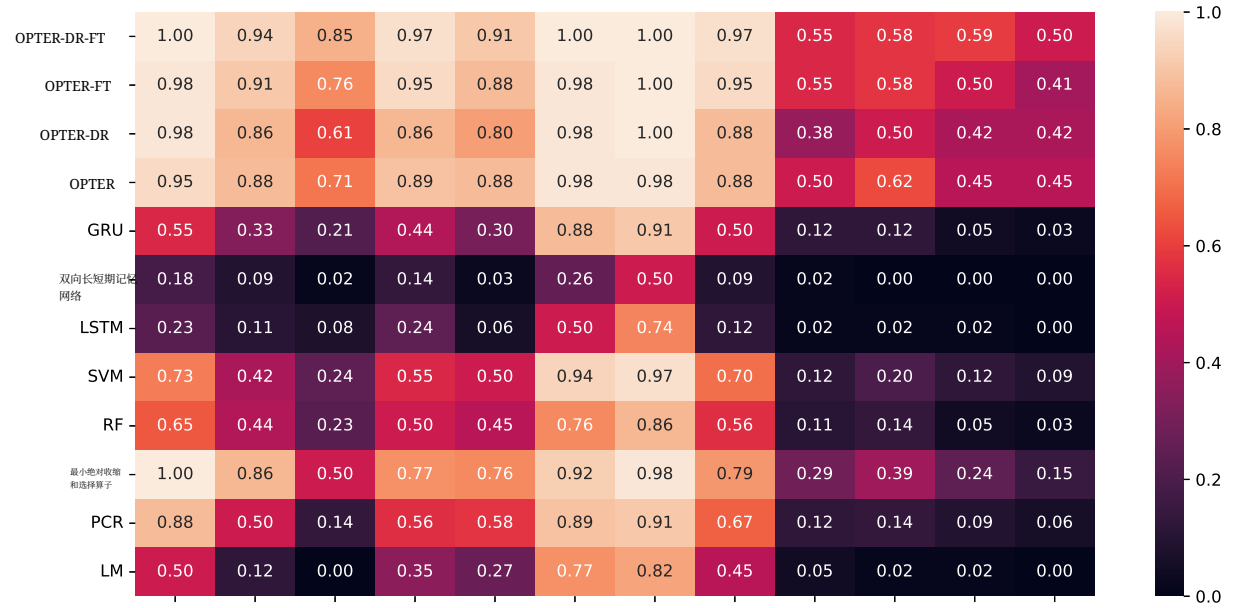


图 ***9***： 此热力图展示了空气污染数据集上不同预测模型之间的比较。相应单元格给出了某种方法（纵轴标签）在相关性方面优于另一种方法（横轴标签）的案例数占比。

记住（41），我们还使用与目标值高度相关的原始特征扩充嵌入空间。考虑向量 v_2 该向量包含在使用原始特征时赋予它的权重，否则为零，即

$$v_{2,j} = \begin{cases} v_{I_k^{\text{Original}}} & j = I_k^{\text{Original}} \\ 0 & j \notin I^{\text{Original}} \end{cases}.$$

由此可得

$$D'^2 = \sum_{j=1}^{p} v_j \left(\sum_{k=1}^{d} U_{k,j} D_k \right)^2 + \sum_{l=1}^{d} v_{2,l} D_l^2, \tag{57}$$

因此，导数可改写为

$$\nabla D'^2(e_d) = 2 \left| U \left[(U^T e_d) \odot v \right] + v_2 \right|. \tag{58}$$

作为示例，我们考虑使用 OFTER-DR、OFTER-FT 和 OFTER-DR-FT 对三个合成数据集（M1-M3）中各数据集的第一个特征进行预测。在图 11 中，我们展示了每个单独特征重要性的网格图。由于我们通过计算 v 方程（43）忽略不重要的特征，OFTER-FT 仅为部分特征赋予非零重要性权重。另一方面，OFTER-DR 为所有原始特征赋予非零权重，因为它们都参与了各主成分的估计。进行降维的 OFTER-DR-FT 也有同样的情况。然而，在大部分可预测性都集中于单个特征（M2.1 - 滞后1）的 M2 中，OFTER-DR-FT 与 OFTER-FT 类似，为大多数特征赋予零重要性权重。因此，对于特征重要性任务，我们建议用户采用 OFTER–FT 或 OFTER-DR-FT。

5.2 异常值检测

OFTER流程的一个附带功能是，它可以通过考察嵌入空间中时间点之间的成对距离，帮助完成异常值检测任务。更具体地说，如果某些点的最近邻在嵌入空间中仍然“很远”，就可以将它们标记为异常值，这与单类支持向量机算法的思路类似 [46]。我们用以下示例来演示这一点。

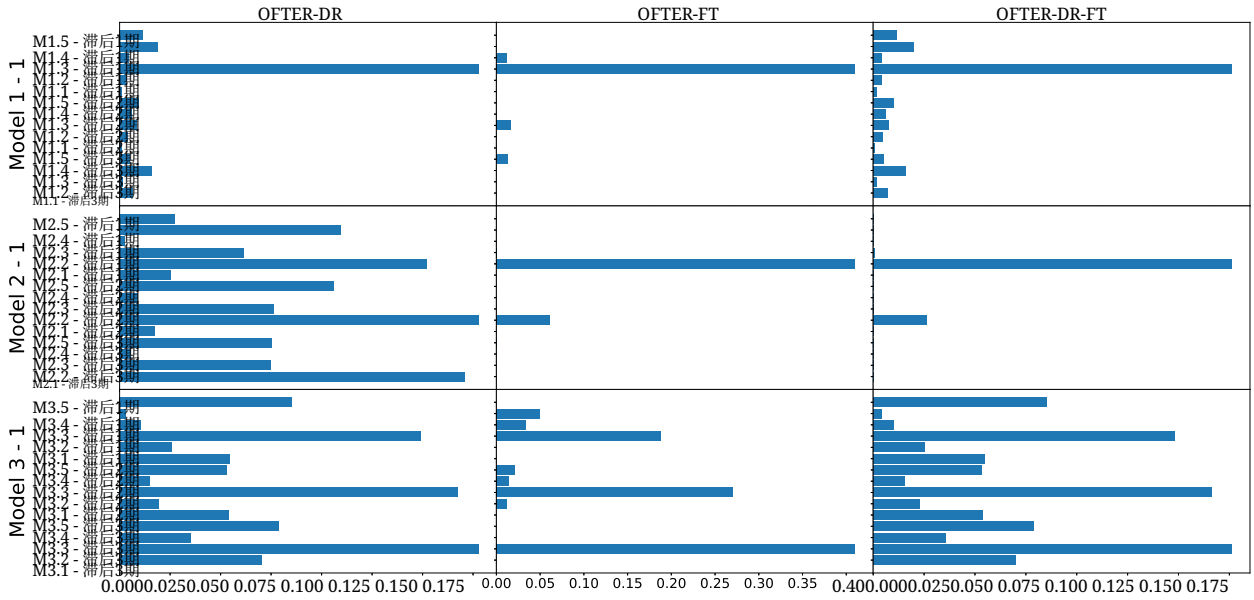


图***10***

图***11***： 在这个 *3×3* 的绘图网格中，我们展示了第 *5.1* 节中得到的特征重要性权重。*M1*、*M2* 和 *M3* 的各自第一个特征分别在第一、第二和第三行中进行预测。每种情况下，都有 *15* 个特征，对应五个特征各自的三个滞后项。此设置针对 OFTER 的三种配置重复进行，分别对应网格的三列。

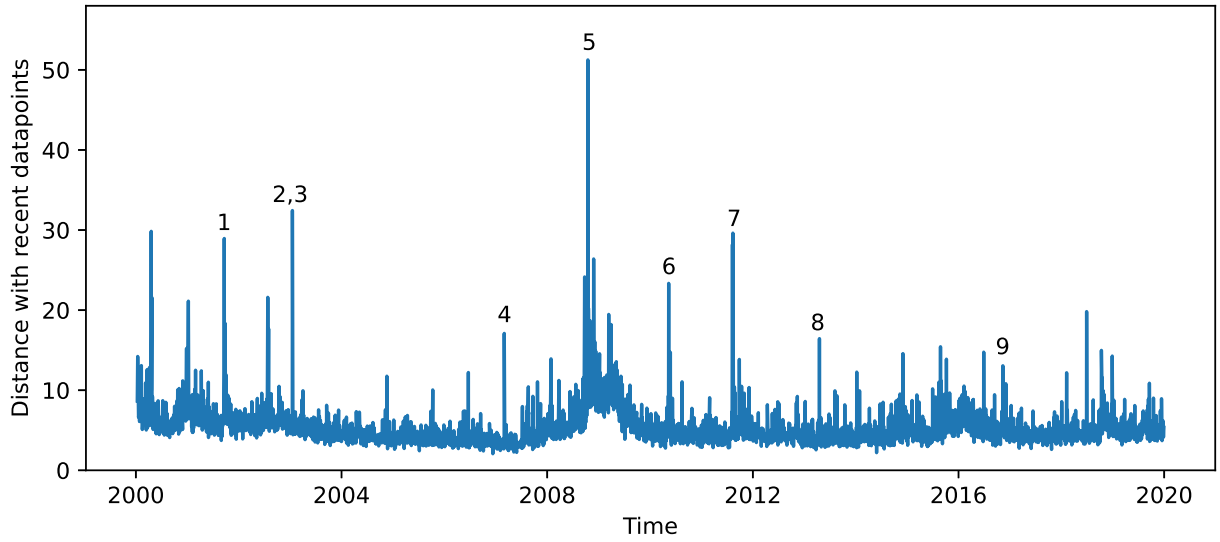


图***12***： 使用建议的异常值检测算法得到的 OFTER-*DR* 的最小距离 d_j^{min}图，其中 $L = 600$和 $\kappa = 5$。所有编号点均被识别为异常值。*1. 2001*年*9*月*18*日 - *9·11*袭击后的余波 *2. 2002*年*7*月*24*日 - *2002*年股市下跌（价格连续下跌 12^{th} 天） *3. 2002*年*7*月*29*日 - *2002*年股市下跌（在（*2*）之后价格大幅上涨后的上涨 3^{rd} 天） *4. 2007*年*2*月*28*日 - 美国和中国股票大幅下跌 *5. 2008*年*10*月*10*日 - 欧洲和亚洲市场崩盘 *6. 2010*年*5*月*7*日 - *2010*年*5*月*6*日闪电崩盘之后 *7. 2011*年*8*月*4*日 - *2011*年*8*月*8*日黑色星期一之前 *8. 2013*年*4*月*15*日 - 标普 *500*指数和道琼斯工业平均指数在*2013*年最大跌幅之一 *9. 2016*年*11*月*8*日 - 美国大选

将当前时间点 t 与其在降维空间中近期历史里的各时间点之间的最小距离定义为

$$d_t^{\min} = \min(\{d_{t-L,t}, \ldots, d_{t-1,t}\}). \tag{59}$$

将 $d_t^{\min}$与以下分布进行比较：$(d_{t-L}^{\min}, \ldots, d_{t-1}^{\min})$ 可作为任意异常值检测算法的输入。[13]记 $q_{t\,1}^{\min}$,和 $q_{t\,3}^{\min}$,分别为 $\{d_{t-L}^{\min}, \ldots, d_{t-1}^{\min}\}$的第一和第三分位数。若对于某个 $\kappa > 0$，时间点 t 满足$d_j^{\min} > q_t^{\min}{}_{,3} + \kappa\left(q_t^{\min}{}_{,3} - q_t^{\min}{}_{,1}\right)$，则将其归类为异常值。

在图 12中，我们绘制了每个交易日与其此前 $L = 600$ 个交易日之间的最小距离，计算时使用了 $\kappa = 5$。从图中可以看出，数据可能存在几种不同的状态：有些时期 $d_{\min}$ 的数值较低，有些时期则较高，例如 2008 年金融危机前后。几乎所有被标记为异常值的点都紧邻金融史上的重要日期，这一事实确实表明，该算法可用于发现异常信号。

最后，我们指出，选择较大的 L 会导致大量异常被标记出来。在实际应用中，如果将这种异常值检测方法用于交易应用，较小的 κ 会更理想，因为可以对异常警告进行定性分析。请注意，上述基于分位数的异常值检测方法只是用于说明 OFTER 潜力的示例算法，因为该流程可以纳入任何其他异常值检测算法中。

6 结论与未来工作

本文介绍了 OFTER，这是一套用于在线多元时间序列预测的完整流程。通过采用 k近邻和广义回归神经网络这两种简单且广为人知、训练复杂度极低的方法，OFTER 在中等规模的数据集上优于多种基线方法。具体而言，大量数值实验表明，在合成数据集上，OFTER 的表现与随机森林和门控循环单元相当；在所考察的真实世界应用中，OFTER 则优于所有其他方法，且运行时间远低于深度学习方法。通过降维以及在比较嵌入空间中的距离时使用加权范数，克服了维数灾难带来的局限。

此外，OFTER还具有其他优势，包括灵活性，能够选择使用或省略其中许多成分。因此，OFTER可以在线无缝集成到异常值检测流程中。最后，OFTER还能自然地为用户提供所有输入变量的特征重要性得分。

未来工作。 这项工作的后续扩展可以包括在算法的模型选择步骤中采用其他机器学习方法，例如使用具有不同正则化参数的LASSO模型。OFTER的一个缺点是难以发现协变量之间的非线性交互效应，尽管对于这种规模的模型和数据集而言，这是一个普遍存在的问题。一种可能的扩展方案是加入预筛选组件，在嵌入步骤之前或之后测试若干潜在的交互项，以扩充特征空间。此外，也值得考察OFTER在其他中等规模数据集上的表现，并评估其他降维算法能否取得优于PCA的性能（在本项目所用的数据集中并非如此）。另一个值得探索的方向是将OFTER改造为用于概率预测。概率模型近年来日益受到关注，因为它们不仅能够提供预测值，还能给出相应的不确定性估计，而这在某些应用中至关重要。最后，我们指出，本文引入的OSMC度量本身也具有独立价值：它既可作为最大相关的替代指标，也可作为其他简单回归方法的一项计算速度快的附加步骤。

致谢

本研究得到英国工程与物理科学研究委员会（EPSRC）项目资助，项目编号为 EP/R513295/1 和 EP/V520202/1。

参考文献

[1] Ahmed NK、Atiya AF、Gayar NE、El-Shishiny H。《时间序列预测机器学习模型的实证比较》。《Econometric Reviews》。2010年；29(5-6)：594-621。参见：https://doi.org/10.1080/07474938.2010.481556。

[13]除了 $d_t^{\min}$，还可以采用平均距离 $\bar{d}_t = \frac{\sum_{j=t-L}^{t-1} d_j}{\cdot\ t}$ ^{t}。

[2] Hochreiter S, Schmidhuber J. 长短期记忆。Neural Comput. 1997年11月;9(8):1735–1780。获取地址：https://doi.org/10.1162/neco.1997.9.8.1735.[3] Cornegruta S, Bakewell R, Withey S, Montana G. 使用双向长短期记忆网络建模放射学语言。载于：健康文本挖掘与信息分析第七届国际研讨会论文集。美国得克萨斯州奥斯汀：计算语言学协会；2016年。第17-27页。获取地址：https://aclanthology.org/W16-6103.[4] Chung J, Gulcehre C, Cho K, Bengio Y. 门控循环神经网络在序列建模中的实证评估。载于：NIPS 2014深度学习研讨会，2014年12月；2014年。.[5] Flunkert V, Salinas D, Gasthaus J. DeepAR：基于自回归循环网络的概率预测。International Journal of Forecasting. 2017年4月;36.[6] Lim B, Arık SO, Loeff N, Pfister T. 用于可解释多步时间序列预测的时间融合变换器。International Journal of Forecasting. 2021年;37(4):1748-64。获取地址：https://www.sciencedirect.com/science/article/pii/S0169207021000637.[7] de O Santos J´unior DS, de Oliveira JFL, de Mattos Neto PSG. 将ARIMA与机器学习模型智能融合以进行时间序列预测。Knowledge-Based Systems. 2019年;175:72-86。获取地址：https://www.sciencedirect.com/science/article/pii/S0950705119301327.[8] Firmino PRA, de Mattos Neto PSG, Ferreira TAE. 修正并组合时间序列预测器。Neural Networks. 2014年;50:1-11。获取地址：https://www.sciencedirect.com/science/article/pii/S0893608013002517.[9] Panigrahi S, Behera HS. 用于时间序列预测的混合ETS–ANN模型。Engineering Applications of Artificial Intelligence. 2017年;66:49-59。获取地址：https://www.sciencedirect.com/science/article/pii/S0952197617301550.[10]Pai PF, Lin CS. 用于股票价格预测的ARIMA与支持向量机混合模型。Omega. 2005年;33(6):497-505。获取地址：https://www.sciencedirect.com/science/article/pii/S0305048304001082.[11]Zhang GP. 使用ARIMA与神经网络混合模型进行时间序列预测。Neurocomputing. 2003年;50:159-75。获取地址：https://www.sciencedirect.com/science/article/pii/S0925231201007020.[12] Zyga K, Price R, Williams B. 用于标志识别的广义回归神经网络。载于：KES'2000：第四届基于知识的智能工程系统及相关技术国际会议论文集（第00TH8516号）。第2卷；2000年。第475-8页，第2卷。[13] Amanda JC, Gary AR, Tami MBB, Joseph PC, Brittney NK. 前馈神经网络和广义回归神经网络在育肥阶段猪采食行为建模中的应用。Biosystems Engineering. 2018年;173:124-33。精密畜牧业基于传感器的监测与管理系统工程进展。获取地址：https://www.sciencedirect.com/science/article/pii/S1537511017304750.[14] Pal M. 使用广义回归神经网络建模桩承载力。载于：印度岩土工程会议，12月；2011年。第15-7页。[15] Al-Mahasneh AJ, Anavatti S, Garratt M, Pratama M. 广义回归神经网络在动态系统中的应用。载于：Asadpour V，编者。数字系统。里耶卡：IntechOpen；2018年。第133-54页。获取地址：https://doi.org/10.5772/intechopen.80258.[16] J Wang JZ, M Xu HW. 使用SMOTE算法和局部线性嵌入对不平衡数据进行分类。Signal Processing，2006年第8届国际会议，第3卷，IEEE。2007年。[17] Tenenbaum JB, Silva V, Langford JC. 一种用于非线性降维的全局几何框架。Science. 2000年;290(5500):2319-23.[18] Donoho DL, Grimes C. Hessian特征映射：用于高维数据的局部线性嵌入技术。Proceedings of the National Academy of Sciences. 2003年;100(10):5591-6.[19] Belkin M, Niyogi P. 用于降维和数据表示的拉普拉斯特征映射。Neural Comput. 2003年6月;15(6):1373-96。获取地址：http://dx.doi.org/10.1162/089976603321780317.[20] Coifman RR, S L. 扩散映射。Applied and Computational Harmonic Analysis. 2006年;21(1):5 30。专刊：扩散映射与小波。获取地址：http://www.sciencedirect.com/science/article/pii/S1063520306000546。

[21] Kramer MA. 使用自联想神经网络进行非线性主成分分析。AIChE期刊. 1991;37(2):233-43. 获取网址：https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic. 690370209.[22] Pulgar FJ, Charte F, Rivera AJ, del Jesus MJ. AEkNN：一种内置降维功能的基于自编码器和k近邻的分类器。《国际计算智能系统杂志》. 2019;12:436-52. 获取网址：https://doi.org/10.2991/ijcis.2018.125905686.[23] Huang D, Jiang F, Li K, Tong G, Zhou G. 尺度化主成分分析：一种新的降维方法。《管理科学》. 2022;68(3):1678-95. 获取网址：https://doi.org/10.1287/mnsc.2021.4020.[24] Chiavazzo E, Gear C, Dsilva C, Rabin N, Kevrekidis I. 通过非线性数据挖掘建立化学动力学简化模型。Processes. 2013 07;2.[25] Papaioannou PG, Talmon R, Kevrekidis IG, Siettos C. 使用流形学习、径向基函数插值和几何谐波进行时间序列预测。《混沌：非线性科学跨学科期刊》. 2022;32(8):083113. 获取网址：https://doi.org/10.1063/5.0094887.[26] Yan X, Bai J, Li X, Chen Z. 降维技术能否在预测中国原油期货波动率时更好地利用不确定性指数的信息？《资源政策》. 2022;75:102521. 获取网址：https://www.sciencedirect.com/science/article/pii/S0301420721005286.[27] Joseph K, Babajide Wintoki M, Zhang Z. 利用投资者情绪预测异常股票收益和交易量：来自在线搜索的证据。International Journal of Forecasting. 2011;27(4):1116-27. 获取网址：https://www.sciencedirect.com/science/article/pii/S0169207011000021.[28] Liu X, keung Lai K. 使用动态支持向量机方法预测日内成交量占比。《系统科学与复杂性杂志》. 2017;30:421-33.[29] Libman D, Haber S, Schaps M. 使用神经网络进行成交量预测。《人工智能前沿》. 2019;2. 获取网址：https://www.frontiersin.org/article/10.3389/frai.2019.00021.[30] Fix E, Hodges JL. 判别分析。非参数判别：一致性性质。《国际统计评论》/ Revue Internationale de Statistique. 1989;57(3):238-47. 获取网址：http://www.jstor. org/stable/1403797.[31] Specht DF. 一种广义回归神经网络。IEEE Transactions on Neural Networks. 1991;2(6):568-76.[32] Bouchaud JP, Potters M. 随机矩阵理论的金融应用：简要综述。载于：《牛津随机矩阵理论手册》。牛津大学出版社；2015. 获取网址：https://doi.org/10.1093/oxfordhb/9780198744191.013.40.[33] Breiman L, Friedman JH. 估计多元回归和相关性的最优变换。《美国统计学会杂志》. 1985;80(391):580-98. 获取网址：https://www.tandfonline.com/doi/abs/10.1080/01621459.1985.10478157.[34] Lorentz GG. 伯恩斯坦多项式。AMS切尔西出版系列。Chelsea Publishing Company；1986. 获取网址：https://books.google.co.uk/books?id=nXiRAKzpdjUC.[35] Hosaka M. 载于：B´ézier曲线与控制点。柏林，海德堡：Springer Berlin Heidelberg；1992. 第117-39页。获取网址：https://doi.org/10.1007/978-3-642-76598-8_9.[36] Piegl L, Tiller W. 载于：B样条曲线与曲面。柏林，海德堡：Springer Berlin Heidelberg；1997. 第81-116页。获取网址：https://doi.org/10.1007/978-3-642-59223-2_3.[37]Mitz R, Shkolnisky Y. ROIPCA：一种基于秩一更新的在线主成分分析算法。CoRR. 2019;abs/1911.11049. 获取网址：http://arxiv.org/abs/1911.11049.[38] Bunch JR, Nielsen CP, Sorensen DC. 对称特征值问题的秩一修正。《物理学文摘》. 1978.[39] Mitz R, Sharon N, Shkolnisky Y. 基于部分谱的对称秩一更新及其在样本外扩展中的应用。SIAM J Matrix Anal Appl. 2017;40:973-97.[40] Melman A. 世俗方程的数值解。《数值数学》. 1995.[41] Bergmeir C, Ben´ítez JM. 关于交叉验证在时间序列预测器评估中的应用。《信息科学》. 2012;191:192-213. 软件可信性数据挖掘。获取网址：https://www.sciencedirect. com/science/article/pii/S0020025511006773.[42] Nicolaou N, Constandinou T. 一种基于非参数乘法回归的非线性因果关系估计器。《神经信息学前沿》. 2016 Jun;10(JUNE)。

[43] Sharpe WF. 夏普比率。《投资组合管理杂志》. 1994;21(1):49-58.[44] Bailey D, Lopez d e Prado M. 经选择偏差、回测过拟合和非正态性调整后的夏普比率。及非正态性。《投资组合管理杂志》。2014年9月；40：94-107。
[45] Kenneth R. French - 数据库；访问日期：2023-02-02。
http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html。
[46] Sch¨olkopf B、Williamson R、Smola A、Shawe-Taylor J、Platt J。用于新颖性检测的支持向量方法。载于：第12届神经信息处理系统国际会议论文集。NIPS'99。美国马萨诸塞州剑桥：MIT出版社；1999年。第582–588页。
[47] Godfrey MD。Peter Henrici 所著《数值分析要素》（John Wiley and Sons，纽约，1964年），328页。海军研究后勤季刊。1965年6月；12(2)：195-5。可从以下网址获取：
https://ideas.repec.org/a/wly/navlog/v12y1965i2p195-195.html。

一种用于求解世俗方程的凸化方法

本节简要介绍我们用于求解第3.1节所述世俗方程的方法，该方法归结为求出以下方程的根：

$$f(\kappa) = 1 + \rho \sum_{j=1}^{d} \frac{z_j^2}{\lambda_j - \kappa}. \tag{60}$$

直接求解方程（60）的主要困难在于，f的定义域中存在大量不连续点。要高效求得方程的 i^{th} 根，可以考虑重新参数化 $\kappa = \lambda_i + \rho\xi$ 和$\delta_j = \frac{\lambda_j - \lambda_i}{\rho}$，从而将方程（60）改写为

$$f_i(\xi) = 1 + \sum_{j=1}^{d} \frac{z_j^2}{\delta_j - \xi}. \tag{61}$$

假设每个特征值都互不相同（这是处理真实数据时一个合理的假设），我们注意到

$$\delta_1 < \delta_2 < \cdots < \delta_i = 0 < \cdots < \delta_n. \tag{62}$$

因此，我们在集合 $(0, \delta_{i+1})$ 中寻找解，其中 $i < n$；并在 $\mathbb{R}_+$ 中寻找解，其中 $i = n$。此外，考虑 $F_i(\gamma) = f_i(\gamma^{-1/2})$，可以进一步将 $f_i(x)$ 转换为凸函数；可以证明，在 $i < n$ 时，该函数在区间 $(\frac{1}{\delta_{i+1}}, \infty)$ 上为凸函数，在 $i = n$时，该函数在区间 $(0, \infty)$ 上为凸函数。
最后，利用递减凸函数的牛顿法[14] 单调收敛于一个解 [47]，这一事实，可以证明：若 γ^* 是 $F_i(\gamma) = 0$的解，则无论初始点取 $\left[\frac{1}{\delta_{i+1}^2}, \gamma^*\right]$中的何值，牛顿法都单调收敛于 γ^*。更详细的解释以及基于割线法的另一种方法，请参阅 [40]。

B 基线方法简述

以下 以下方法作为 OFTER 的基线方法。我们将分别简要介绍它们。 e.

普通最小二乘法（普通最小二乘法）

普通最小二乘法 （普通最小二乘法）是目前最广为人知的回归方法，其中目标变量 y 被建模 d 作为线性组合 特征的

$$y_i = \beta_0 + \sum_{j=1}^{d} \beta_j X_{i,j} + \epsilon_i. \tag{63}$$

我们提醒读者，牛顿法从点 x_0 开始，寻找 $f(x) = 0$ 的解，然后考虑序列

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

直到获得一个解。

系数向量 $\beta = (\beta_0, ..., \beta_d)$ 被选为使 $\sum \epsilon_i^2$ 最大的向量，并可使用闭式公式求得

$$\beta = (X^T X)^{-1} X^T y. \tag{64}$$

由于我们在特征矩阵中纳入了滞后特征，因此该方法与向量自回归回归（VAR）模型非常相似。

主成分回归

为了解决特征之间存在的高度相关性，主成分回归（PCR）首先识别出˜ X的主成分，得到 X，然后保留一定数量的成分 $p < d$。随后使用普通最小二乘法，但针对 X 的前 p 列˜进行回归，而不是针对 X。与 OFTER 类似，我们使用 $\delta = 0.9$ 来选择成分数量。

最小绝对收缩与选择算子

在 X 进行普通最小二乘法之前，另一种降低维度的方法是在目标函数中加入正则化项，见方程（63）。最小绝对收缩和选择算子（LASSO）是一种广受欢迎的方法，其修改后的目标函数形式如下

$$(y - X\beta)^T (y - X\beta) + \lambda \sum_j |\beta_j|. \tag{65}$$

请注意，在这种情况下不存在闭式解，因此需要采用数值优化方法。我们通过交叉验证过程选择λ。

随机森林

决策树是一种受决策分析启发的简单方法，其模型可视化结果类似于一棵由节点和分支组成的树。初始特征空间构成根节点。根节点中的观测会依据一条分割规则划分到“后继节点”中，该规则的选择旨在优化某种准则（通常为熵或基尼不纯度）。这一过程在各个后继节点上递归重复，直至满足某个停止准则。对于每个新的数据点，利用其特征确定其对应的叶节点。随后，可通过对落入每个叶节点的所有观测的目标值取平均来获得预测值。这是一种非参数方法，但方差非常高。

随机森林通过拟合多棵决策树构建。每棵决策树仅使用随机抽取的观测值样本和特征样本进行拟合。随后，将各棵决策树的预测值合并为一个预测值（通常取平均值）。随机森林基于装袋法，旨在降低方差并避免过拟合。在基准任务中，我们使用 200 棵树。

支持向量机（SVM）：

SVM 是另一种受学习理论启发的热门机器学习模型。具体而言，求得一个向量 β 和一个数值 β_0，使其最小化

$$\frac{1}{2}||\beta||^2, \quad \text{s.t} \quad |(y_i - \beta^T x_i - \beta_0)| \leq \epsilon. \tag{66}$$

新点 x' 的预测值为 $\beta_0 + \beta^T x'$。

循环神经网络：

循环神经网络（RNN）是一类专门针对具有时间结构的数据集设计的神经网络。这通常通过引入额外的跳跃连接来实现，这些连接将过去层的输出传递到当前层。由此产生的一个问题是，梯度往往会消失或爆炸。

长短期记忆网络是循环神经网络的一种变体，专门用于避免梯度爆炸或梯度消失问题。其实现方式是引入细胞状态，其中包含前面各个细胞的摘要信息，并由多个不同的门进行控制。记 x_t 为一个数据点的特征向量，c_t 为细胞状态的输入，以及 x_t 为模块的输出。基本长短期记忆网络的公式形式为

细胞状态，以及 h_t 为模块的输出。基本长短期记忆网络的公式形式为

$$\begin{cases} f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \\ i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right), \\ o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right), \\ \hat{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right), \\ c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t . h_t = c_t \odot \tanh(c_t), \end{cases} \tag{67}$$

其中 W_f， U_f， b_f， W_i， U_i， b_i， W_o， U_o， b_o， W_c， U_c和 b_c是可训练参数。

我们还考虑了 LSTM 的另外两种变体。双向长短期记忆网络（Bi-LSTM）使用两个并行且方向相反的 LSTM 网络，使信息能够双向流动。我们还使用门控递归单元（GRU），其架构与 LSTM 相似，但更为简单。

对于这两种层，我们均使用两层和 300 个训练轮次。

C 稳健性分析

在本节中，我们将展示 OFTER-DR-FT 中使用的五个主要参数的变化如何影响其在合成数据集上的表现。我们固定 OFTER 中给出的五个主要参数如下： $\delta = 0.9$， $L = 800$， $C_{\text{min}} = 0.05$， $C_{\text{Original}} = 0.05$，以及 $p_{\text{ADF}} = 0.05$。保持其他四个参数不变，我们依次调整这些参数，并记录其对性能的影响。在表3中，我们展示了将上述各参数调整为给定数值时，目标值与预测值之间的相关性。

总体而言，算法的性能并不显著取决于参数的选择，因为对每个特征进行预测时，性能都较为稳定。不过也有例外；例如， C_{min} 似乎会显著影响算法在 M2.3 和 M2.4 上的性能。

	δ		p_{ADF}		L		C_{min}		C_{original}	
数值	**0.8**	**0.95**	**0.01**	**0.1**	**500**	**1500**	**0.01**	**0.1**	**0.01**	**0.1**
M1.1	0.470	0.459	0.467	0.467	0.441	0.477	0.464	0.464	0.465	0.461
M1.2	0.519	0.511	0.509	0.509	0.503	0.521	0.509	0.514	0.504	0.518
M1.3	0.077	0.071	0.085	0.085	0.085	0.113	0.119	0.116	0.085	0.117
M1.4	0.554	0.558	0.552	0.552	0.555	0.556	0.553	0.556	0.545	0.559
M1.5	0.094	0.107	0.101	0.101	0.147	0.120	0.102	0.110	0.101	0.113
M2.1	0.599	0.623	0.599	0.599	0.596	0.603	0.675	0.600	0.599	0.60
M2.2	0.535	0.535	0.535	0.535	0.534	0.539	0.897	0.535	0.536	0.536
M2.3	0.864	0.799	0.815	0.815	0.776	0.85	0.759	0.401	0.804	0.817
M2.4	0.962	0.964	0.963	0.963	0.963	0.963	0.962	0.963	0.963	0.963
M2.5	0.679	0.681	0.681	0.681	0.678	0.682	0.681	0.678	0.680	0.679
M3.1	0.758	0.758	0.758	0.758	0.751	0.762	0.758	0.758	0.758	0.757
M3.2	0.852	0.851	0.851	0.851	0.85	0.852	0.851	0.850	0.849	0.850
M3.3	0.669	0.664	0.665	0.665	0.662	0.668	0.665	0.666	0.664	0.667
M3.4	0.902	0.903	0.903	0.903	0.902	0.905	0.903	0.902	0.903	0.902
M3.5	0.826	0.827	0.827	0.827	0.825	0.828	0.827	0.826	0.825	0.825

表***3***：*OFTER*稳健性分析。该表展示了通过改变算法的五个参数子集，目标值与合成数据集的 *15* 个特征各自预测值之间的相关性。