Evaluating time series forecasting models

An empirical study on performance estimation methods

Vitor Cerqueira[1,2], Luis Torgo[1,2,3], and Igor Mozetič[4]

[1]*LIAAD-INESC TEC, Porto, Portugal*
[2]*University of Porto, Porto, Portugal*
[3]*Dalhousie University, Halifax, Canada*
[4]*Jožef Stefan Institute, Ljubljana, Slovenia*

May 29, 2019

Performance estimation aims at estimating the loss that a predictive model will incur on unseen data. These procedures are part of the pipeline in every machine learning project and are used for assessing the overall generalisation ability of predictive models. In this paper we address the application of these methods to time series forecasting tasks. For independent and identically distributed data the most common approach is cross-validation. However, the dependency among observations in time series raises some caveats about the most appropriate way to estimate performance in this type of data and currently there is no settled way to do so. We compare different variants of cross-validation and of out-of-sample approaches using two case studies: One with 62 real-world time series and another with three synthetic time series. Results show noticeable differences in the performance estimation methods in the two scenarios. In particular, empirical experiments suggest that cross-validation approaches can be applied to stationary time series. However, in real-world scenarios, when different sources of non-stationary variation are at play, the most accurate estimates are produced by out-of-sample methods that preserve the temporal order of observations.

评价时间序列预测模型

绩效评估方法的实证研究

Vitor Cerqueira、Luis Torgo和Igor Mozeti Zarlac

[1]LIAAD–INESC TEC，波尔图，葡萄牙
[2]波尔图大学，波尔图，葡萄牙
[3]达尔豪西大学，哈利法克斯，加拿大
[4]斯洛文尼亚卢布尔雅那，Jo Bazzef Stefan研究所

2019年5月29日，

性能估计的目的是估计预测模型在看不见的数据上会产生的损失。这些程序是每个机器学习项目中管道的一部分，用于评估预测模型的整体泛化能力。在本文中，我们解决了这些方法的时间序列预测任务的应用。对于独立同分布的数据，最常用的方法是交叉验证。然而，时间序列中的观测之间的依赖性提出了一些关于估计这类数据性能的最适当方法的警告，目前还没有确定的方法。我们使用两个案例研究比较了交叉验证和样本外方法的不同变体：一个是62个真实世界的时间序列，另一个是三个合成时间序列。结果表明，在两种情况下的性能估计方法的显着差异。 特别是，实证实验表明，交叉验证方法可以应用于平稳的时间序列。然而，在现实世界的情况下，当不同来源的非平稳变化在发挥作用时，最准确的估计是由样本外的方法，保持观察的时间顺序。

1 Introduction

Machine learning plays an increasingly important role in science and technology, and performance estimation is part of any machine learning project pipeline. This task is related to the process of using the available data to estimate the loss that a model will incur on unseen data. Machine learning practitioners typically use these methods for model selection, hyper-parameter tuning and assessing the overall generalization ability of the models. In effect, obtaining reliable estimates of the performance of models is a critical issue on all predictive analytics tasks.

Choosing a performance estimation method often depends on the data one is modelling. For example, when one can assume independence and an identical distribution (i.i.d.) among observations, cross-validation [17] is typically the most appropriate method. This is mainly due to its efficient use of data [1]. However, there are some issues when the observations in the data are dependent, such as time series. These dependencies raise some caveats about using standard cross-validation in such data. Notwithstanding, there are particular time series settings in which variants of cross-validation can be used, such as in stationary or small-sized data sets where the efficient use of all the data by cross-validation is beneficial [6].

In this paper we present a comparative study of different performance estimation methods for time series forecasting tasks. Several strategies have been proposed in the literature and currently there is no consensual approach. We applied different methods in two case studies. One is comprised of 62 real-world time series with potential non-stationarities and the other is a stationary synthetic environment [4–6].

In this study we compare two main classes of estimation methods:

- Out-of-sample (OOS): These methods have been traditionally used to estimate predictive performance in time-dependent data. Essentially, out-of-sample methods hold out the last part of the time series for testing. Although these approaches do not make a complete use of the available data, they preserve the temporal order of observations. This property may be important to cope with the dependency among observations and account for the potential temporal correlation between the consecutive values of the time series.

- Cross-validation (CVAL): These approaches make a more efficient use of the available data, which is beneficial in some settings [6]. They assume that observations are i.i.d., though some strategies have been proposed to circumvent this requirement. These methods have been shown to be able to provide more robust estimations than out-of-sample approaches in some time series scenarios [4–6].

A key characteristic that distinguishes these two types of approaches is that OOS methods always preserve the temporal order of observations meaning that a model is never tested on past data. The objective of this study is to address the following research question: How do out-of-sample methods compare to cross-validation approaches in terms of performance estimation ability for different types of time series data?

This paper is an extension to an article published before [12]. In this work, we substantially increase the experimental setup both in methods and data sets used;

1引言

机器学习在科学技术中发挥着越来越重要的作用，性能评估是任何机器学习项目管道的一部分。此任务与使用可用数据来估计模型将在不可见数据上产生的损失的过程有关。机器学习从业者通常使用这些方法进行模型选择，超参数调整和评估模型的整体泛化能力。实际上，获得模型性能的可靠估计是所有预测分析任务的关键问题。

性能估计方法的选择通常取决于建模的数据。例如，当可以假设独立性和同分布（i.i.d.）在观察中，交叉验证[17]通常是最合适的方法。这主要是由于它对数据的有效利用[1]。但是，当数据中的观测值是相关的时，会出现一些问题，例如时间序列。这些依赖性提出了在此类数据中使用标准交叉验证的一些警告。尽管如此，在特定的时间序列设置中，可以使用交叉验证的变体，例如在固定或小型数据集中，通过交叉验证有效使用所有数据是有益的[6]。

在本文中，我们提出了不同的性能估计方法的时间序列预测任务的比较研究。文献中提出了几种战略，目前还没有达成共识的办法。我们在两个案例研究中采用了不同的方法。一个由62个具有潜在非平稳性的真实世界时间序列组成，另一个是静态合成环境[4–6]。

在这项研究中，我们比较了两类主要的估计方法：

- 样本外（OOS）：这些方法传统上用于估计时间相关数据的预测性能。从本质上讲，样本外方法保留了时间序列的最后一部分用于测试。虽然这些方法没有完全利用现有数据，但它们保留了观测的时间顺序。这个属性对于科普观测之间的依赖性和解释时间序列的连续值之间的潜在时间相关性可能是重要的。

- 交叉验证（CVAL）：这些方法可以更有效地利用可用数据，这在某些情况下是有益的[6]。他们假设观察是独立同分布的，尽管已经提出了一些策略来规避该要求。在某些时间序列场景中，这些方法已被证明能够提供比样本外方法更稳健的估计[4–6]。

区分这两种方法的一个关键特征是，OOS方法总是保持观察的时间顺序，这意味着模型永远不会在过去的数据上进行测试。本研究的目的是解决以下研究问题：如何样外的方法比较交叉验证方法的性能估计能力不同类型的时间序列数据?

本文是对[12]之前发表的一篇文章的扩充。在这项工作中，我们大大增加了实验装置在方法和数据集使用;

provide additional analysis such as the impact of stationarity; and a more in-depth and critical discussion of the results.

This paper is structured as follows. The literature on performance estimation for time series forecasting tasks is reviewed in Section 2. Materials and methods are described in Section 3, including the predictive task, time series data sets, performance estimation methodology, and experimental design. The results of the experiments are reported in Section 4. A discussion of our results is carried out in Section 5. Finally, the conclusions of our empirical study are provided in Section 6.

2 Background

In this section we provide a background to this paper. We review the typical estimation methods used in time series forecasting and explain the motivation for this study.

In general, performance estimation methods for time series forecasting tasks are designed to cope with the dependence between observations. This is typically accomplished by having a model tested on observations future to the ones used for training. These include the OOS testing as well as variants of the CVAL method.

2.1 Out-of-sample approaches

Figure 1: Simple out-of-sample procedure: an initial part of the available observations are used for fitting a predictive model. The last part of the data is held out, where the predictive model is tested.

When using OOS performance estimation procedures, a time series is split into two parts: an initial fit period in which a model is trained, and a testing period held out for estimating the loss of that model. This simple approach (`Holdout`) is depicted in Figure 1. However, within this type of procedure one can adopt different strategies regarding training/testing split point, growing or sliding window settings, and eventual update of the models. In order to produce a robust estimate of predictive performance, Tashman [38] recommends employing these strategies in multiple test periods. One might create different sub-samples according to, for example, business cycles [14]. For a more general setting one can also adopt a randomized approach. This is similar to random sub-sampling (or repeated holdout) in the sense that they consist of repeating a learning plus testing cycle several times using different, but possibly overlapping data samples (`Rep-Holdout`). This idea is illustrated in Figure 2, where one iteration of a repeated holdout is shown. A point a is randomly chosen from the available window

我们在方法和数据集方面都大大增加了实验设置;提供了额外的分析，如平稳性的影响;以及对结果进行了更深入和更重要的讨论。

本文件的结构如下。第2节回顾了时间序列预测任务的性能估计文献。材料和方法在第3节中描述，包括预测任务、时间序列数据集、性能估计方法和实验设计。实验结果见第4节。第5节对我们的结果进行了讨论。最后，我们的实证研究的结论提供在第6节。

2背景

在本节中，我们提供了本文的背景。我们回顾了时间序列预测中使用的典型估计方法，并解释了这项研究的动机。

一般来说，时间序列预测任务的性能估计方法的设计，以科普观测之间的依赖性。这通常是通过在用于训练的观测值的未来对模型进行测试来实现的。

这些包括OOS测试以及CVAL方法的变体。

2.1样本外方法

图1：简单的样本外程序：可用观测的初始部分
用于拟合预测模型。数据的最后一部分是测试预测模型的部分。

当使用OOS性能估计过程时，时间序列被分成两个部分：训练模型的初始拟合期，以及用于估计该模型损失的测试期。图1描述了这种简单的方法（保持）。然而，在这种类型的过程中，可以采用不同的策略来训练/测试分裂点，增长或滑动窗口设置，以及最终更新模型。为了对预测性能进行稳健的估计，Tashman [38]建议在多个测试阶段采用这些策略。例如，可以根据商业周期创建不同的子样本[14]。对于更一般的设置，也可以采用随机方法。这类似于随机子采样（或重复保持），因为它们包括使用不同但可能重叠的数据样本（Rep–Holdout）重复多次学习加测试循环。 这个想法在图2中说明，其中显示了重复坚持的一次迭代。 从可用窗口中随机选择点a

(constrained by the training and testing sizes) of a time series Y. This point then marks the end of the training set, and the start of the testing set.

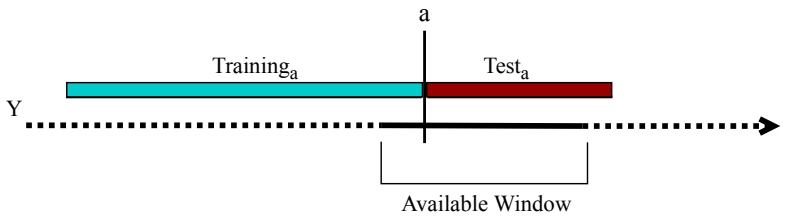


Figure 2: Example of one iteration of the repeated holdout procedure. A point a is chosen from the available window. Then, a previous part of observations are used for training, while a subsequent part of observations are used for testing.

2.2 Prequential

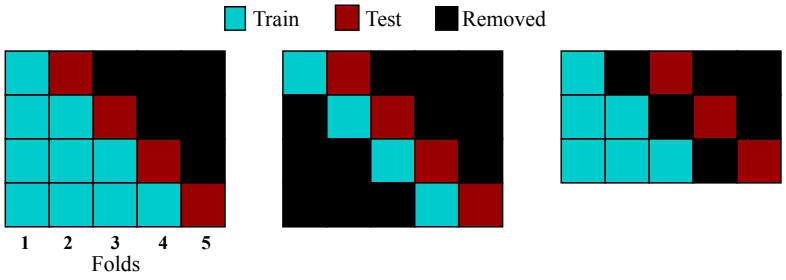


Figure 3: Variants of prequential approach applied in blocks for performance estimation. This strategy can be applied using a growing window (left, right), or a sliding window (middle). One can also introduce a gap between the training and test sets.

OOS approaches are similar to prequential or interleaved-test-then-train evaluation [7, Chapter 2.2]. Prequential is typically used in data streams mining. The idea is that each observation is first used to test the model, and then to train the model. This can be applied in blocks of sequential instances [29]. In the initial iteration, only the first two blocks are used, the first for training and the second for test. In the next iteration, the second block is merged with the first and the third block is used for test. This procedure continues until all blocks are tested (`Preq-Bls`). This procedure is exemplified in the left side of Figure 3, in which the data is split into 5 blocks.

从时间序列Y的可用窗口（受训练和测试大小的约束）中随机选择点a。这一点标志着训练集的结束和测试集的开始。

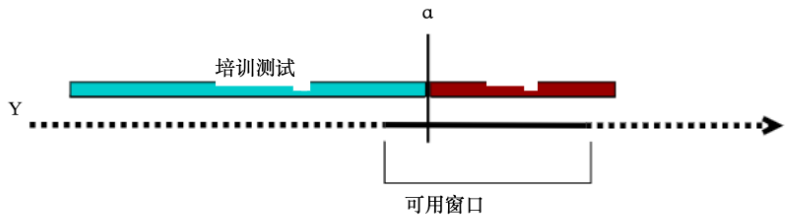


图2：重复保持程序的一次迭代示例。A点A是
从可用窗口中选择。然后，前一部分观测值用于训练，而后一部分观测值用于测试。

2.2前序的

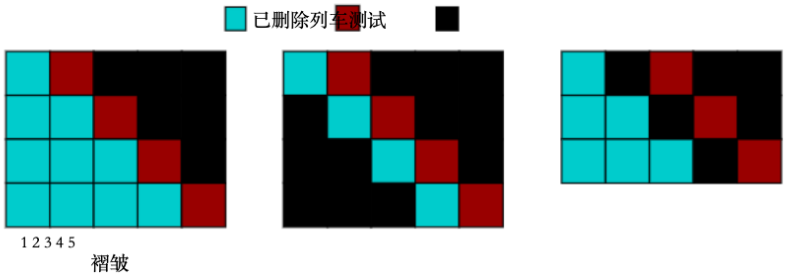


图3：用于性能估计的块中应用的先决方法的变体
是的。可以使用增长窗口（左，右）或滑动窗口（中）应用此策略。还可以在训练集和测试集之间引入间隙。

OOS方法类似于先决或先测试后训练评估[7，第2.2章]。Prequential通常用于数据流挖掘。每个观测值首先用于测试模型，然后用于训练模型。这可以应用于连续实例的块[29]。在初始迭代中，仅使用前两个块，第一个用于训练，第二个用于测试。在下一次迭代中，第二个块与第一个块合并，第三个块用于测试。该程序继续进行，直至所有组织块均已检测（Preq-Bls）。该过程在图3的左侧举例说明，其中数据被分成5个块。

A variant of this idea is illustrated in the middle scheme of Figure 3. Instead of merging the blocks after each iteration (growing window), one can forget the older blocks in a sliding window fashion (`Preq-Sld-Bls`). This idea is typically adopted when past data becomes deprecated, which is common in non-stationary environments. Another variant of the prequential approach is represented in the right side of Figure 3. This illustrates a prequential approach applied in blocks, where a gap block is introduced (`Preq-Bls-Gap`). The rationale behind this idea is to increase the independence between training and test sets.

2.3 Cross-validation approaches

The typical approach when using K-fold cross-validation is to randomly shuffle the data and split it in K equally-sized folds or blocks. Each fold is a subset of the data comprising t/K randomly assigned observations, where t is the number of observations. After splitting the data into K folds, each fold is iteratively picked for testing. A model is trained on K-1 folds and its loss is estimated on the left out fold (`CV`). In fact, the initial random shuffle of observations before splitting into different blocks is not intrinsic to cross-validation [17]. Notwithstanding, the random shuffling is a common practice among data science professionals. This approach to cross-validation is illustrated in the left side of Figure 4.

2.3.1 Variants designed for time-dependent data

Some variants of K-fold cross-validation have been proposed specially designed for dependent data, such as time series [1]. However, theoretical problems arise by applying this technique directly to this type of data. The dependency among observations is not taken into account since cross-validation assumes the observations to be i.i.d.. This might lead to overly optimistic estimations and consequently, poor generalisation ability of predictive models on new observations. For example, prior work has shown that cross-validation yields poor estimations for the task of choosing the bandwidth of a kernel estimator in correlated data [18]. To overcome this issue and approximate independence between the training and test sets, several methods have been proposed as variants of this procedure. We will focus on variants designed to cope with temporal dependency among observations.

The Blocked Cross-Validation [35] (`CV-Bl`) procedure is similar to the standard form described above. The difference is that there is no initial random shuffling of observations. In time series, this renders K blocks of contiguous observations. The natural order of observations is kept within each block, but broken across them. This approach to cross-validation is also illustrated in the left side of Figure 4. Since the random shuffle of observations is not being illustrated, the figure for `CV-Bl` is identical to the one shown for `CV`.

The Modified CV procedure [27] (`CV-Mod`) works by removing observations from the training set that are correlated with the test set. The data is initially randomly shuffled and split into K equally-sized folds similarly to K-fold cross-validation. Afterwards, observations from the training set within a certain temporal range of the observations

图3的中间方案中示出了该想法的变型。代替在每次迭代之后合并块（增长窗口），可以以滑动窗口方式（Preq–Sld–Bls）忘记较旧的块。当过去的数据被弃用时，通常会采用这种想法，这在非静态环境中很常见。图3的右侧表示了先决方法的另一种变体。这示出了在块中应用的先决方法，其中引入了间隙块（Preq–Bls–Gap）。这个想法背后的基本原理是增加训练集和测试集之间的独立性。

2.3交叉验证方法

使用K折交叉验证时的典型方法是随机洗牌数据并将其拆分为K个相等大小的折叠或块。每个折叠是数据的一个子集，包括t/K个随机分配的观测值，其中t是观测值的数量。在将数据分割成K个折叠之后，每个折叠被迭代地挑选用于测试。模型在K–1折叠上训练，并在左侧折叠（CV）上估计其损失。事实上，在划分为不同块之前，观察的初始随机洗牌并不是交叉验证所固有的[17]。尽管如此，随机洗牌是数据科学专业人员的常见做法。这种交叉验证方法如图4的左侧所示。

2.3.1为时间相关数据设计的变量

K折交叉验证的一些变体已经被提出来专门设计用于相关数据，例如时间序列[1]。然而，将这种技术直接应用于这种类型的数据会产生理论问题。由于交叉验证假设观测是独立同分布的，因此不考虑观测之间的依赖性。这可能导致过于乐观的估计，从而导致预测模型对新观测的泛化能力差。例如，先前的工作已经表明，交叉验证对于在相关数据中选择核估计器的带宽的任务产生差的估计[18]。为了克服这个问题和训练集和测试集之间的近似独立性，已经提出了几种方法作为该过程的变体。我们将专注于变量设计，以科普时间之间的依赖性观察。

阻断交叉验证[35]（CV–Bl）程序类似于上述标准形式。不同之处在于，没有初始的随机观察。在时间序列中，这会呈现K个连续观测数据块。观察的自然顺序保持在每个块内，但在它们之间被打破。图4的左侧也说明了这种交叉验证方法。由于未示出观察的随机混洗，CV–B1的图与CV所示的图相同。

修改的CV程序[27]（CV–Mod）通过从训练集中删除与测试集相关的观察值来工作。数据最初随机混洗并分成K个相等大小的折叠，类似于K折叠交叉验证。然后，在观测的一定时间范围内，

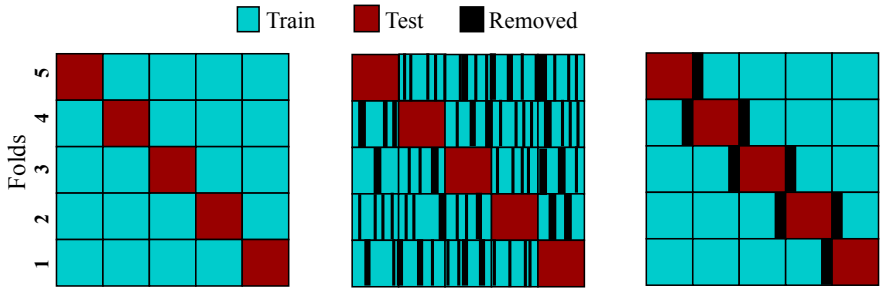


Figure 4: Variants of cross-validation estimation procedures

of the test set are removed. This ensures independence between the training and test sets. However, when a significant amount of observations are removed from training, this may lead to model under-fit. This approach is also described as non-dependent cross-validation [4]. The graph in the middle of Figure 4 illustrates this approach.

The hv-Blocked Cross-Validation (`CV-hvBl`) proposed by Racine [34] extends blocked cross-validation to further increase the independence among observations. Specifically, besides blocking the observations in each fold, which means there is no initial randomly shuffle of observations, it also removes adjacent observations between the training and test sets. Effectively, this creates a gap between both sets. This idea is depicted in the right side of Figure 4.

2.3.2 Usefulness of cross-validation approaches

Recently there has been some work on the usefulness of cross-validation procedures for time series forecasting tasks. Bergmeir and Benítez [4] present a comparative study of estimation procedures using stationary time series. Their empirical results show evidence that in such conditions cross-validation procedures yield more accurate estimates than an OOS approach. Despite the theoretical issue of applying standard cross-validation, they found no practical problem in their experiments. Notwithstanding, the Blocked cross-validation is suggested for performance estimation using stationary time series.

Bergmeir et al. [5] extended their previous work for directional time series forecasting tasks. These tasks are related to predicting the direction (upward or downward) of the observable. The results from their experiments suggest that the hv-Blocked CV procedure provides more accurate estimates than the standard out-of-sample approach. These were obtained by applying the methods on stationary time series.

Finally, Bergmeir et al. [6] present a simulation study comparing standard cross-validation to out-of-sample evaluation. They used three data generating processes and performed 1000 Monte Carlo trials in each of them. For each trial and generating process, a stationary time series with 200 values was created. The results from the simulation suggest that cross-validation systematically yields more accurate estimates,

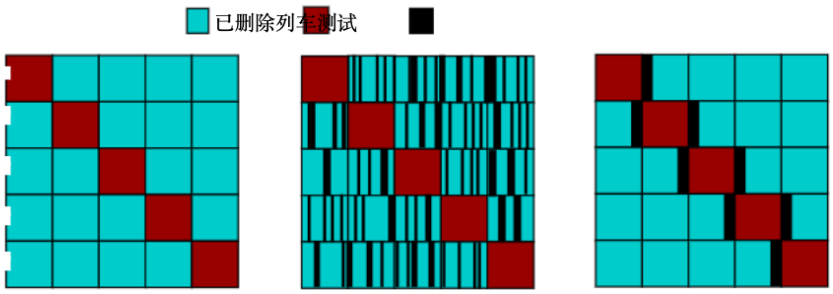


图4：交叉验证估计程序的变体

在测试集的观测值的特定时间范围内的来自训练集的观测值被移除。这确保了训练集和测试集之间的独立性。然而，当从训练中删除大量观测值时，这可能会导致模型欠拟合。这种方法也被描述为非依赖交叉验证[4]。图4中间的图表说明了这种方法。

Racine [34]提出的hv–Blocked Cross–Validation（CV–hvBl）扩展了Blocked Cross–Validation，以进一步提高观察结果之间的独立性。具体来说，除了在每个折叠中阻塞观察值（这意味着没有初始的随机观察值洗牌）之外，它还删除了训练集和测试集之间的相邻观察值。实际上，这在两组之间产生了差距。这个想法在图4的右侧描述。

2.3.2交叉验证方法的可行性

最近有一些工作的有用性的交叉验证程序的时间序列预测任务。Bergmeir和Ben 'itez [4]提出了一个使用平稳时间序列的估计程序的比较研究。他们的实证结果表明，在这种情况下，交叉验证程序比OOS方法产生更准确的估计。尽管应用标准交叉验证存在理论问题，但他们在实验中没有发现实际问题。尽管如此，阻塞交叉验证建议使用平稳时间序列的性能估计。

Bergmeir等人[5]扩展了他们以前的工作，用于定向时间序列预测任务。这些任务与预测可观察的方向（向上或向下）有关。他们的实验结果表明，hv–Blocked CV程序比标准样本外方法提供更准确的估计值。

这些都是通过应用平稳时间序列的方法。

最后，Bergmeir等人。[6]提出了一项模拟研究，比较了标准交叉验证和样本外评估。他们使用了三个数据生成过程，并在每个过程中进行了1000次Monte Carlo试验。对于每个试验和生成过程，创建具有200个值的固定时间序列。模拟结果表明，交叉验证系统地产生更准确的估计，

provided that the model is correctly specified.

In a related empirical study [30], the authors compare estimation procedures on several large time-ordered Twitter datasets. They find no significant difference between the best cross-validation and out-of-sample evaluation procedures. However, they do find that standard, randomized cross-validation is significantly worse than the blocked cross-validation, and should not be used to evaluate classifiers in time-ordered data scenarios.

Despite the results provided by these previous works we argue that they are limited in two ways. First, the used experimental procedure is biased towards cross-validation approaches. While these produce several error estimates (one for each fold), the OOS approach is evaluated in a one-shot estimation, where the last part of the time series is withheld for testing. OOS methods can be applied in several windows for more robust estimates, as recommended by Tashman [38]. By using a single origin, one is prone to particular issues related to that origin.

Second, the results are based on stationary time series, most of them artificial. Time series stationarity is equivalent to identical distribution in the terminology of more traditional predictive tasks. Hence, the synthetic data generation processes and especially the stationary assumption limit interesting patterns that can occur in real-world time series. Our working hypothesis is that in more realistic scenarios one is likely to find time series with complex sources of non-stationary variations.

In this context, this paper provides an extensive comparative study using a wide set of methods for evaluating the performance of uni-variate time series forecasting models. These include several variants of both cross-validation and out-of-sample approaches. The analysis is carried out using a real-world scenario as well as a synthetic case study used in the works described previously [4–6].

2.4 Related work on performance estimation for dependent data

The problem of performance estimation has also been under research in different scenarios where the observations are somehow dependent (non-i.i.d.).

2.4.1 Performance estimation under spatio-temporal dependencies

Geo-referenced time series are becoming more prevalent due to the increase of data collection from sensor networks. In these scenarios, the most appropriate estimation procedure is not obvious as spatio-temporal dependencies are at play. Oliveira et al. [32] presented an extensive empirical study of performance estimation for forecasting problems with spatio-temporal time series. The results reported by the authors suggest that both CVAL and OOS methods are applicable in these scenarios. Like previous work in time-dependent domains [4, 30], Oliveira et al. suggest the use of blocking when using a cross-validation estimation procedure.

只要模型被正确地指定。
在一项相关的实证研究[30]中，作者比较了几个大型时间排序Twitter数据集的估计过程。他们发现最好的交叉验证和样本外评估程序之间没有显着差异。然而，他们确实发现标准的随机交叉验证比阻塞交叉验证明显更差，并且不应该用于评估时间排序数据场景中的分类器。

尽管这些以前的作品提供的结果，我们认为，他们是有限的，在两个方面。首先，所使用的实验程序偏向于交叉验证方法。虽然这些会产生几个误差估计（每个折叠一个），但OOS方法是在一次性估计中进行评估的，其中时间序列的最后一部分被保留用于测试。如Tashman [38]所建议的，OOS方法可以在多个窗口中应用，以获得更稳健的估计。通过使用单一的起源，人们倾向于与该起源相关的特定问题。

其次，结果是基于平稳的时间序列，其中大部分是人为的。
时间序列平稳性在更传统的预测任务的术语中等同于同分布。因此，合成数据生成过程，特别是平稳假设限制了可能发生在现实世界时间序列中的有趣模式。我们的工作假设是，在更现实的情况下，人们很可能会发现时间序列的非平稳变化的复杂来源。

在这种情况下，本文提供了一个广泛的比较研究，使用广泛的方法来评估单变量时间序列预测模型的性能。这些方法包括交叉验证和样本外方法的几种变体。该分析是使用真实世界的场景以及前面描述的工作中使用的综合案例研究进行的[4–6]。

2.4相关数据性能估计的相关工作

性能估计的问题也一直在不同的情况下，其中的意见是某种程度上依赖（非独立同分布）的研究。

2.4.1时空相关性下的性能估计

随着传感器网络数据采集量的增加，地理参考时间序列变得越来越普遍。在这些情况下，最合适的估计程序是不明显的时空依赖性在发挥作用。Oliveira等人。[32]提出了一项关于时空时间序列预测问题的性能估计的广泛实证研究。作者报告的结果表明，CVAL和OOS方法都适用于这些场景。与以前在时间依赖域中的工作一样[4，30]，Oliveira等人建议在使用交叉验证估计程序时使用分块。

2.4.2 Performance estimation in data streams mining

Data streams mining is concerned with predictive models that evolve continuously over time in response to concept drift [16]. Gama et al. [15] provide a thorough overview of the evaluation of predictive models for data streams mining. The authors defend the usage of the prequential estimator with a forgetting mechanism, such as a fading factor or a sliding window.

This work is related to ours in the sense that it deals with performance estimation using time-dependent data. Notwithstanding, the paradigm of data streams mining is in line with sequential analysis [40]. As such, the assumption is that the sample size is not fixed in advance, and predictive models are evaluated as observations are collected. In our setting, given a time series data set, we want to estimate the loss that a predictive models will incur in unseen observations future to that data set.

3 Materials and methods

In this section we present the materials and methods used in this work. First, we will define the prediction task. Second, the time series data sets are described. We then formalize the methodology employed for performance estimation. Finally, we overview the experimental design.

3.1 Predictive task definition

A time series is a temporal sequence of values $Y = \{y_1, y_2, \ldots, y_t\}$, where y_i is the value of Y at time i and t is the length of Y. We remark that we use the term time series assuming that Y is a numeric variable, i.e., $y_i \in \mathbb{R}, \forall\ y_i \in Y$.

Time series forecasting denotes the task of predicting the next value of the time series, y_{t+1}, given the previous observations of Y. We focus on a purely auto-regressive modelling approach, predicting future values of time series using its past lags.

To be more precise, we use time delay embedding [37] to represent Y in an Euclidean space with embedding dimension p. Effectively, we construct a set of observations which are based on the past p lags of the time series. Each observation is composed of a feature vector $x_i \in \mathbb{X} \subset \mathbb{R}^p$, which denotes the previous p values, and a target vector $y_i \in \mathbb{Y} \subset \mathbb{R}$, which represents the value we want to predict. The objective is to construct a model $f : \mathbb{X} \rightarrow \mathbb{Y}$, where f denotes the regression function.

Summarizing, we generate the following matrix:

$$Y_{[n,p]} = \left[\begin{array}{ccccc|c} y_1 & y_2 & \cdots & y_{p-1} & y_p & y_{p+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i-p+1} & y_{i-p+2} & \cdots & y_{i-1} & y_i & y_{i+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-p+1} & y_{t-p+2} & \cdots & y_{t-1} & y_t & y_{t+1} \end{array}\right]$$

2.4.2数据流挖掘中的性能估计

数据流挖掘关注的是随着时间的推移不断演变以响应概念漂移的预测模型[16]。Gama等人[15]提供了对数据流挖掘预测模型评估的全面概述。作者捍卫使用的前置估计与遗忘机制，如衰落因子或滑动窗口。

这项工作与我们的意义上说，它涉及使用时间相关的数据的性能估计。尽管如此，数据流挖掘的范式与序列分析一致[40]。因此，假设样本量事先不固定，并在收集观察结果时评估预测模型。在我们的设置中，给定一个时间序列数据集，我们希望估计预测模型在该数据集未来不可见的观测中将产生的损失。

3材料与方法

在本节中，我们介绍了这项工作中使用的材料和方法。首先，我们将定义预测任务。其次，描述了时间序列数据集。然后，我们正式的性能估计所采用的方法。最后，我们概述了实验设计。

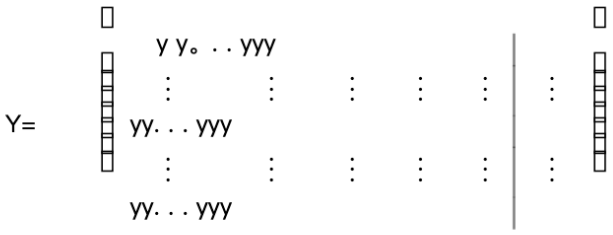
3.1预测任务定义

时间序列是值Y = {y，y，. . .，y}，其中y是Y在时间i的值，t是Y的长度。我们注意到，我们使用术语时间序列假设Y是一个数字变量，即，y∈ R，y∈ Y .

时间序列预测表示在给定Y的先前观测值的情况下预测时间序列的下一个值y的任务。我们专注于一个纯粹的自回归建模方法，预测未来值的时间序列使用其过去的滞后。

为了更精确，我们使用时间延迟嵌入[37]来表示嵌入维数为p的欧几里得空间中的Y。有效地，我们构建了一组基于时间序列过去p个滞后的观测值。每个观测值由一个特征向量x∈ X <$R和一个目标向量y∈ Y <$R组成，其中x ∈ X <$R表示先前的p值，y∈ Y <$R表示我们想要预测的值。目标是构建一个模型f：X → Y，其中f表示回归函数。

总而言之，我们生成以下矩阵：



Taking the first row of the matrix as an example, the target value is y_{p+1}, while the attributes (predictors) are the previous p values $\{y_1, \ldots, y_p\}$. Essentially we assume that there are no time dependencies larger than p.

3.2 Time series data

Two different case studies are used to analyse the performance estimation methods: a scenario comprised of real-world time series and a synthetic setting used in prior work [4–6] for addressing the issue of performance estimation for time series forecasting tasks.

3.2.1 Real-world time series

We analyse 62 real-world time series (RWTS) from different domains. They have different granularity and length as well as unknown dynamics. The time series are described in Table 1 in Appendix 6. In the table, the column p denotes the embedding dimension of the respective time series. Our approach for estimating this parameter is addressed in section 3.4.1. Differencing is the computation of the differences between consecutive observations. This process is useful to remove changes in the level of a time series, thus stabilising the mean [20]. This is important to account for trend and seasonality in time series. The column I represents the number of differences applied to the respective time series in order to make it trend-stationary according to the KPSS test [24]. Finally, the column S represents whether or not a time series is stationary (1 if it is, 0 otherwise).

We analysed the stationarity of the time series comprising the real-world case study. Essentially, a time series is said to be stationary if its characteristics do not depend on the time that the data is observed [20]. In this work we consider a stationarity of order 2. This means that a time series is considered stationary if it has constant mean, constant variance, and an auto-covariance that does not depend on time. Henceforth we will refer a time series as stationary if it is stationary of order 2.

In order to test if a given time series is stationary we follow the wavelet spectrum test described by Nason [31]. This test starts by computing an evolutionary wavelet spectral approximation. Then, for each scale of this approximation, the coefficients of the Haar wavelet are computed. Any large Haar coefficient is evidence of a non-stationarity. An hypothesis test is carried out to assess if a coefficient is large enough to reject the null hypothesis of stationarity. In particular, we apply a multiple hypothesis test with a Bonferroni correction and a false discovery rate [31].

In Figure 5 is shown an example of the application of the wavelet spectrum test to a non-stationary time series. In the graphic, each red horizontal arrow denotes a non-stationarity found by the test. The left-hand side axis denotes the scale of the time series. The right-hand axis represents the scale of the wavelet periodogram and where the non-stationarities are found. Finally, the lengths of the arrows denote the scale of the Haar wavelet coefficient whose null hypothesis was rejected. For a thorough description of this method we refer to the work by Nason [31].

以矩阵的第一行为例，目标值是y，而属性（预测因子）是之前的p值{y，. . .，y}。本质上，我们假设没有大于p的时间依赖性。

3.2 Time series data 🔄 ❓

两个不同的案例研究用于分析性能估计方法：由真实世界时间序列组成的场景和先前工作中使用的合成设置[4–6]，用于解决时间序列预测任务的性能估计问题。

3.2.1真实世界时间序列

我们分析了来自不同领域的62个真实世界的时间序列（RWTS）。它们具有不同的粒度和长度以及未知的动态。时间序列描述见附录6中的表1。在该表中，列p表示相应时间序列的嵌入维数。我们估计该参数的方法在第3.4.1节中讨论。差分是计算连续观测值之间的差异。这个过程有助于消除时间序列水平的变化，从而稳定均值[20]。这对于解释时间序列中的趋势和季节性很重要。列I表示应用于相应时间序列的差异数量，以便根据KPSS测试使其趋势平稳[24]。最后，列S表示时间序列是否是平稳的（如果是，则为1，否则为0）。

我们分析了由真实世界案例研究组成的时间序列的平稳性。
从本质上讲，如果时间序列的特征不依赖于数据被观察的时间，那么它就是平稳的[20]。在这项工作中，我们考虑一个平稳的顺序2。这意味着如果时间序列具有恒定的均值、恒定的方差和不依赖于时间的自协方差，则该时间序列被认为是平稳的。然而，如果一个时间序列是2阶平稳的，我们就称它是平稳的。

为了测试给定的时间序列是否是平稳的，我们遵循Nason [31]描述的小波谱测试。该测试通过计算进化小波谱近似开始。然后，对于这种近似的每个尺度，计算Haar小波的系数。任何大的哈尔系数都是非平稳性的证据。进行假设检验以评估系数是否足够大以拒绝平稳性的零假设。特别是，我们应用了Bonferroni校正和错误发现率的多重假设检验[31]。

在图5中示出了将小波谱检验应用于非平稳时间序列的示例。在图中，每个红色水平箭头表示测试发现的非平稳性。左侧轴表示时间序列的尺度。右手轴表示小波周期图的尺度和发现非平稳性的地方。最后，箭头的长度表示其零假设被拒绝的Haar小波系数的尺度。对于这种方法的全面描述，我们参考Nason的工作[31]。

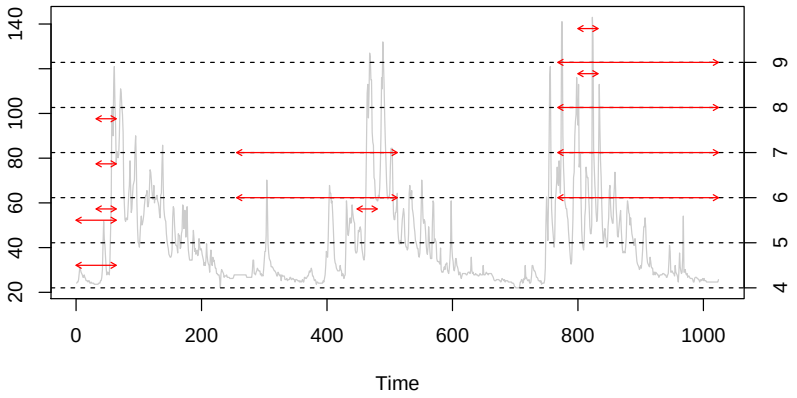


Figure 5: Application of the wavelet spectrum test to a non-stationary time series. Each red horizontal arrow denote a non-stationarity identified by the test.

3.2.2 Synthetic time series

We use three synthetic use cases defined in previous work by Bergmeir et al. [5, 6]. The data generating processes are all stationary and are designed as follows:

S1: A stable auto-regressive process with lag 3, i.e., the next value of the time series is dependent on the past 3 observations – c.f. Figure 6 for a sample graph.

S2: An invertible moving average process with lag 1 – c.f. Figure 7 for a sample graph.

S3: A seasonal auto-regressive process with lag 12 and seasonal lag 1 – c.f. Figure 8 for a sample graph.

For the first two cases, S1 and S2, real-valued roots of the characteristic polynomial are sampled from the uniform distribution $[-r; -1.1] \cup [1.1, r]$, where r is set to 5 [4]. Afterwards, the roots are used to estimate the models and create the time series. The data is then processed by making the values all positive. This is accomplished by subtracting the minimum value and adding 1. The third case S3 is created by fitting a seasonal auto-regressive model to a time series of monthly total accidental deaths in the USA [9]. For a complete description of the data generating process we refer to the work by Bergmeir et al. [4, 6]. Similarly to Bergmeir et al., for each use case we performed 1000 Monte Carlo simulations. In each repetition a time series with 200 values was generated.

3.3 Performance estimation methodology

Performance estimation addresses the issue of estimating the predictive performance of predictive models. Frequently, the objective behind these tasks is to compare different

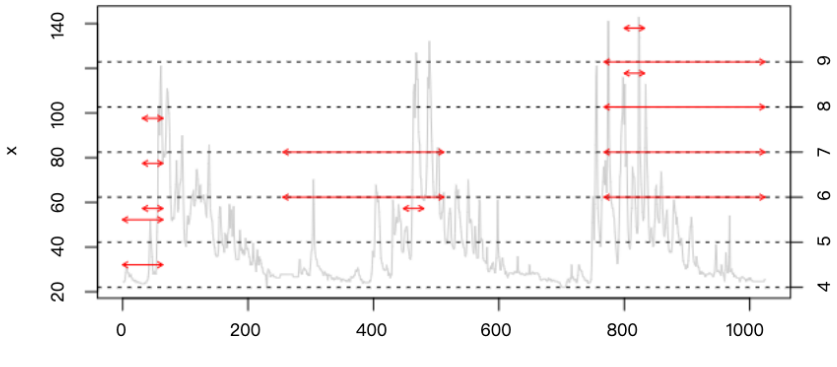


图5：小波谱检验在非平稳时间序列中的应用。
每个红色水平箭头表示测试识别的非平稳性。

3.2.2合成时间序列

我们使用Bergmeir等人在以前的工作中定义的三个合成用例。[5，6]。数据生成过程都是静态的，设计如下：

S1：滞后为3的稳定自回归过程，即，时间序列的下一个值取决于过去的3个观测值– c.f.图6是一个示例图。

S2：滞后1 – c.f.的可逆移动平均过程。图7是一个示例图。

S3：具有滞后12和季节滞后1 – c.f.的季节自回归过程。图8是一个示例图。

For the first two cases, S1 and S2, real–valued roots of the characteristic polynomial are sampled from the uniform distribution [−r; −1.1] ∪ [1.1, r], where r is set to 5 [4]. Afterwards, the roots are used to estimate the models and create the time series. The data is then processed by making the values all positive. This is accomplished by subtracting the minimum value and adding 1. The third case S3 is created by fitting a seasonal auto–regressive model to a time series of monthly total accidental deaths in the USA [9]. For a complete description of the data generating process we refer to the work by Bergmeir et al. [4, 6]. Similarly to Bergmeir et al., for each use case we performed 1000 Monte Carlo simulations. In each repetition a time series with 200 values was generated. 🔄 ❓

3.3业绩估计方法

性能估计解决了估计预测模型的预测性能的问题。通常，这些任务背后的目标是比较不同的

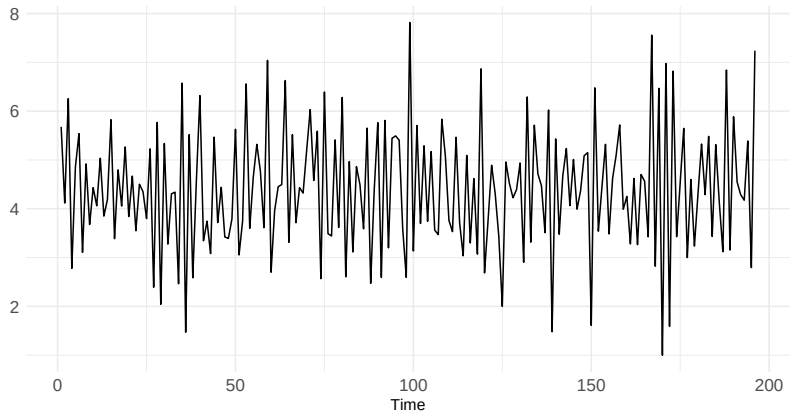


Figure 6: Sample graph of the S1 synthetic case.

solutions for solving a predictive task. This includes selecting among different learning algorithms and hyper-parameter tuning for a particular one.

Training a learning model and evaluating its predictive ability on the same data has been proven to produce biased results due to overfitting [1]. Since then several methods for performance estimation have been proposed in the literature, which use new data to estimate the performance of models. Usually, new data is simulated by splitting the available data. Part of the data is used for training the learning algorithm and the remaining data is used to test and estimate the performance of the model.

For many predictive tasks the most widely used of these methods is K-fold cross-validation [36] (c.f. Section 2 for a description). The main advantages of this method is its universal splitting criteria and efficient use of all the data. However, cross-validation is based on the assumption that observations in the underlying data are independent. When this assumption is violated, for example in time series data, theoretical problems arise that prevent the proper use of this method in such scenarios. As we described in Section 2 several methods have been developed to cope with this issue, from out-of-sample approaches [38] to variants of the standard cross-validation, e.g., block cross-validation [35].

Our goal in this paper is to compare a wide set of estimation procedures, and test their suitability for different types of time series forecasting tasks. In order to emulate a realistic scenario we split each time series data in two parts. The first part is used to estimate the loss that a given learning model will incur on unseen future observations. This part is further split into training and test sets as described before. The second part is used to compute the true loss that the model incurred. This strategy allows the computation of unbiased estimates of error since a model is always tested on unseen observations.

The workflow described above is summarised in Figure 9. A time series Y is split into an estimation set Y^{est} and a subsequent validation set Y^{val}. First, Y^{est} is used to calculate $\hat{g}$, the estimate of the loss that a predictive model m will incur on future

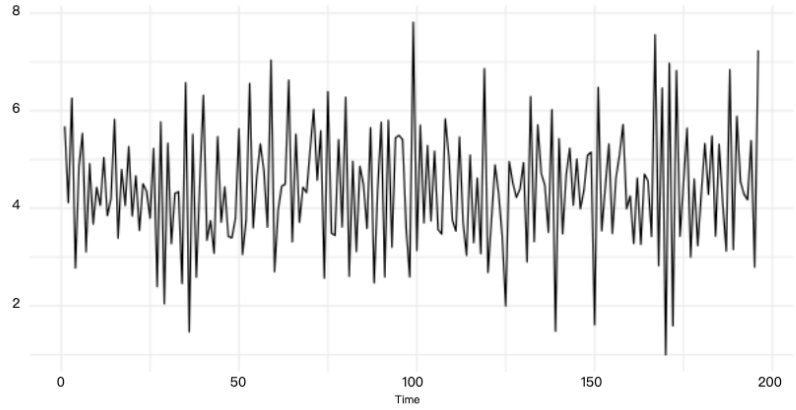


图6：S1合成外壳的示例图。

这些任务背后的目标是比较用于解决预测任务的不同解决方案。这包括在不同的学习算法中进行选择，以及为特定的学习算法进行超参数调整。

训练学习模型并评估其对相同数据的预测能力已被证明会由于过拟合而产生有偏差的结果[1]。从那时起，在文献中提出了几种性能估计方法，这些方法使用新的数据来估计模型的性能。通常，新数据是通过拆分可用数据来模拟的。部分数据用于训练学习算法，其余数据用于测试和估计模型的性能。

对于许多预测任务，这些方法中最广泛使用的是K折交叉验证[36]（参见第2节说明）。该方法的主要优点是其通用的分裂准则和有效地利用所有的数据。然而，交叉验证是基于基础数据中的观察是独立的假设。当违反这一假设时，例如在时间序列数据中，就会出现理论问题，阻止在这种情况下正确使用这种方法。正如我们在第2节中所描述的，已经开发了几种方法来科普这个问题，从样本外方法[38]到标准交叉验证的变体，例如，块交叉验证[35]。

我们在本文中的目标是比较广泛的估计程序，并测试其适用于不同类型的时间序列预测任务。为了模拟现实场景，我们将每个时间序列数据分为两部分。第一部分用于估计给定的学习模型在不可见的未来观测中会产生的损失。如前所述，这一部分进一步分为训练集和测试集。第二部分用于计算模型产生的真实损失。这种策略允许计算误差的无偏估计，因为模型总是在看不见的观测值上进行测试。

图9总结了上述工作流程。时间序列Y被分成估计集Y和后续验证集Y。首先，Y用于计算预测模型m在未来可能发生的损失的估计，

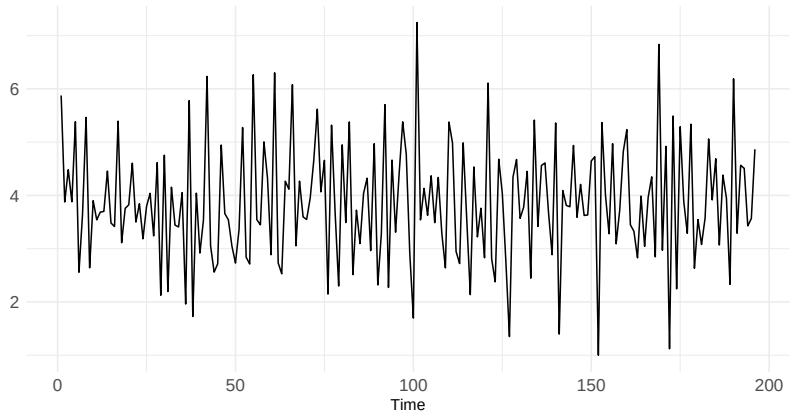


Figure 7: Sample graph of the S2 synthetic case.

new observations. This is accomplished by further splitting Y^{est} into training and test sets according to the respective estimation procedure g_i, $i \in \{1, \ldots, z\}$. The model m is built on the training set and $\hat{g}_i$ is computed on the test set.

Second, in order to evaluate the estimates $\hat{g}_i$ produced by the methods g_i, $i \in \{1, \ldots, z\}$, the model m is re-trained using the complete set Y^{est} and tested on the validation set Y^{val}. Effectively, we obtain L^m, the ground truth loss that m incurs on new data.

In summary, the goal of an estimation method g_i is to approximate L^m by $\hat{g}_i$ as well as possible. In Section 3.4.3 we describe how to quantify this approximation.

3.4 Experimental design

The experimental design was devised to address the following research question: How do the predictive performance estimates of cross-validation methods compare to the estimates of out-of-sample approaches for time series forecasting tasks?

Existing empirical evidence suggests that cross-validation methods provide more accurate estimations than traditionally used OOS approaches in stationary time series forecasting [4–6] (see Section 2). However, many real-world time series comprise complex structures. These include cues from the future that may not have been revealed in the past. Effectively, our hypothesis is that preserving the temporal order of observations when estimating the predictive ability of models is an important component.

3.4.1 Embedding dimension and estimation set size

We estimate the optimal embedding dimension (p) using the method of False Nearest Neighbours [21]. This method analyses the behaviour of the nearest neighbours as we increase p. According to Kennel et al. [21], with a low sub-optimal p many of the nearest neighbours will be false. Then, as we increase p and approach an optimal

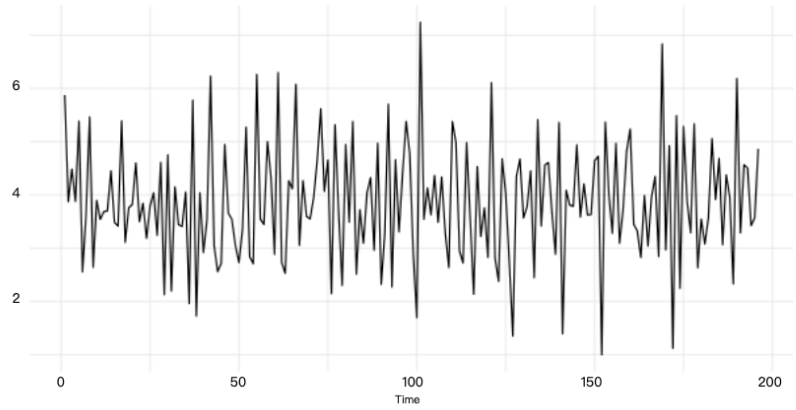


图7：S2合成用例的示例图。

新的观察。这是通过根据相应的估计过程g，i ∈ {1，. . .，z}。模型m是建立在训练集上的，而模型m是在测试集上计算的。

其次，为了评价方法g，i ∈ {1，. . .，z}，使用完整集合Y重新训练模型m，并在验证集合Y上测试模型m。实际上，我们获得了L，即在新数据上引起的地面真实损失。

总之，估算方法的目标是尽可能地近似Lby气体。在3.4.3节中，我们描述了如何量化这种近似。

3.4实验设计

实验设计的目的是解决以下研究问题：如何交叉验证方法的预测性能估计相比，估计的样本外的时间序列预测任务的方法?

现有的经验证据表明，交叉验证方法在平稳时间序列预测中提供了比传统使用的OOS方法更准确的估计[4–6]（见第2节）。然而，许多真实世界的时间序列包含复杂的结构。这些包括来自未来的线索，这些线索在过去可能没有被揭示。实际上，我们的假设是，在估计模型的预测能力时，保持观察的时间顺序是一个重要的组成部分。

3.4.1嵌入维数和估计集大小

我们使用假近邻方法估计最佳嵌入维数（p）[21]。这种方法分析了当我们增加p时最近邻的行为。根据Kennel等人的说法。[21]，对于低的次优p，许多最近邻将是假的。然后，当我们增加p并接近最优值时，

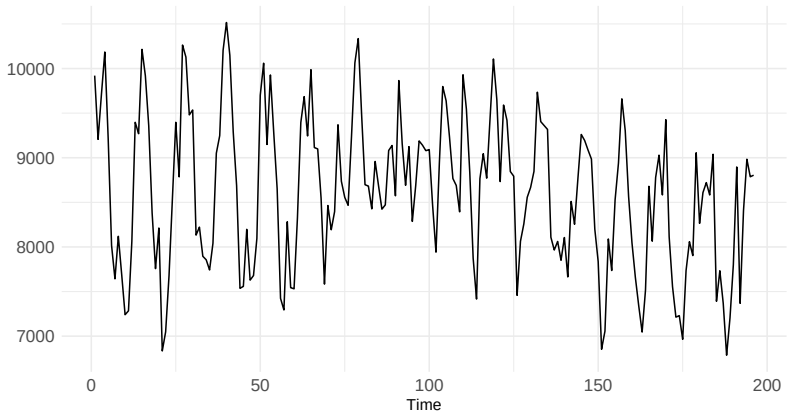


Figure 8: Sample graph of the S3 synthetic case.

embedding dimension those false neighbours disappear. We set the tolerance of false nearest neighbours to 1%. The embedding dimension estimated for each series is shown in Table 1. Regarding the synthetic case study, we fixed the embedding dimension to 5. The reason for this setup is to try to follow the experimental setup by Bergmeir et al. [6].

The estimation set (Y^{est}) in each time series is the first 70% observations of the time series – see Figure 9. The validation period is comprised of the subsequent 30% observations (Y^{val}).

3.4.2 Estimation methods

In the experiments we apply a total of 11 performance estimation methods, which are divided into CVAL variants and OOS aproaches. The cross-validation methods are the following:

`CV` Standard, randomized K-fold cross-validation;

`CV-Bl` Blocked K-fold cross-validation;

`CV-Mod` Modified K-fold cross-validation;

`CV-hvBl` hv-Blocked K-fold cross-validation;

Conversely, the out-of-sample approaches are the following:

`Holdout` A simple OOS approach–the first 70% of Y^E is used for training and the subsequent 30% is used for testing;

`Rep-Holdout` OOS tested in *nreps* testing periods with a Monte Carlo simulation using 70% of the total observations t of the time series in each test. For each period,

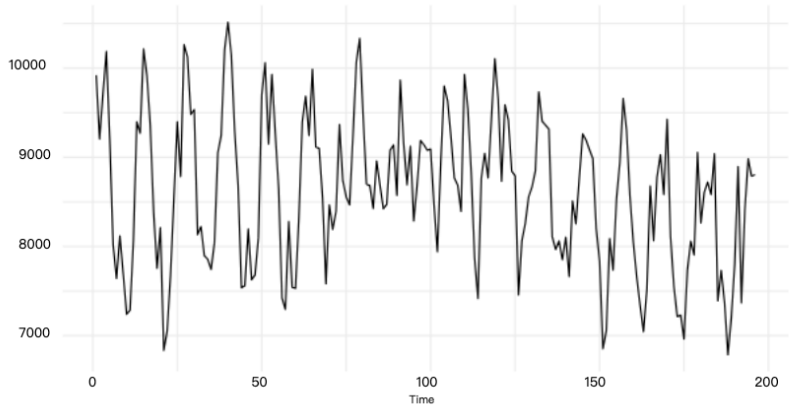


图8：S3合成外壳的示例图。

嵌入维数使伪邻域消失。我们将错误最近邻的容忍度设置为1%。每个系列的嵌入维数估计值见表1。关于合成案例研究，我们将嵌入维数固定为5。这种设置的原因是试图遵循Bergmeir等人的实验设置。[6]。

每个时间序列中的估计集（Y）是时间序列的前70%观测值–参见图9。验证期由后续30%观察结果（Y）组成。

3.4.2估算方法

在实验中，我们总共应用了11种性能估计方法，分为CVAL变体和OOS方法。交叉验证方法如下：

CV标准，随机K折交叉验证；

CV–B1封闭的K折交叉验证；

CV–Mod改良K折交叉验证；

CV–hvBI hv阻断的K折交叉验证；

相反，样本外方法如下：

一个简单的OOS方法–Y的前70%用于训练，
后续30%用于测试；

使用蒙特卡罗模拟在nreps测试期间测试Rep–Holdout OOS，
每次测试中时间序列总观测值t的70%。对于每个时期，

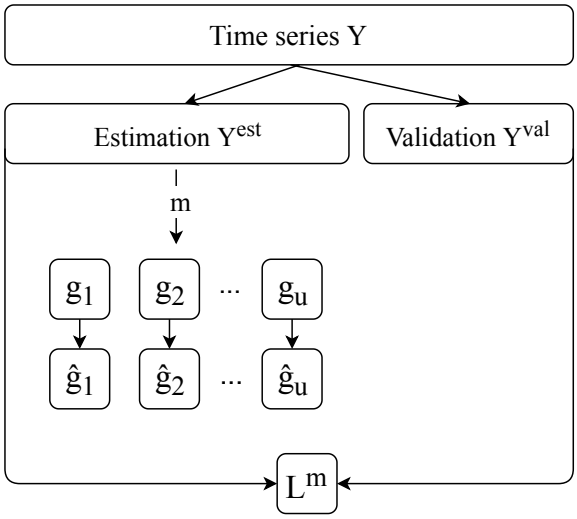


Figure 9: Experimental comparison procedure: A time series is split into an estimation set Y^{est} and a subsequent validation set Y^{val}. The first is used to estimate the error $\hat{g}$ that the model m will incur on unseen data, using z different estimation methods. The second is used to compute the actual error L^m incurred by m. The objective is to approximate L^m by $\hat{g}$ as well as possible.

a random point is picked from the time series. The previous window comprising 60% of t is used for training and the following window of 10% of t is used for testing;

`Preq-Bls` Prequential evaluation in blocks in a growing fashion;

`Preq-Sld-Bls` Prequential evaluation in blocks in a sliding fashion–the oldest block of data is discarded after each iteration;

`Preq-Bls-Gap` Prequential evaluation in blocks in a growing fashion with a gap block–this is similar to the method above, but comprises a block separating the training and testing blocks in order to increase the independence between the two parts of the data;

`Preq-Grow` **and** `Preq-Slide` As baselines we also include the exhaustive prequential methods in which an observation is first used to test the predictive model and then to train it. We use both a growing/landmark window (`Preq-Grow`) and a sliding window (`Preq-Slide`).

We refer to Section 2 for a complete description of the methods. The number of folds K or repetitions *nreps* in these methods is 10, which is a commonly used setting in the literature. The number of observations removed in `CV-Mod` and `CV-hvBl` (c.f. Section 2) is the embedding dimension p in each time series.

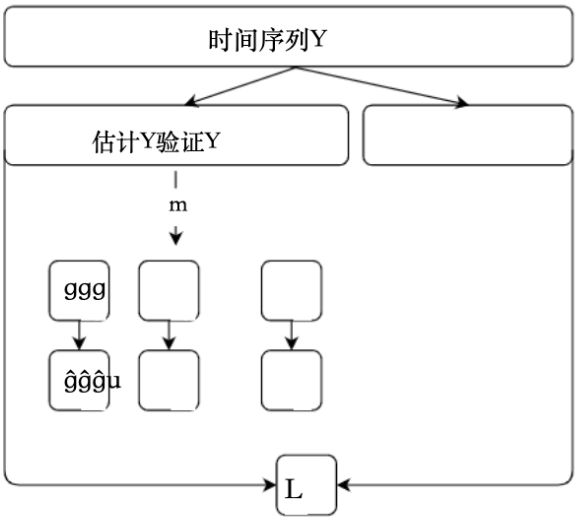


图9：实验比较过程：将时间序列拆分为估计值
集合Y和随后的验证集合Y。第一个用于使用z种不同的估计方法来估计模型m将在未知数据上产生的误差σ g。第二个用于计算m引起的实际误差L。目标是尽可能地近似Lby Bg。

从时间序列中选取随机点。包括t的60%的前一个窗口用于训练，并且t的10%的后一个窗口用于测试;

以增长的方式对区块进行预先评价;

Preq-Sld-Bls以滑动方式在块中进行的预评估-最早的块
的数据在每次迭代后被丢弃;

Preq-Bls-间隙以增长方式对间隙区组进行区组预评价-
这类似于上述方法，但是包括将训练块和测试块分开的块，以便增加数据的两个部分之间的独立性;

Preq-Grow和Preq-Slide作为基线，我们还包括详尽的先决条件
首先使用观测值来测试预测模型，然后对其进行训练的方法。我们使用增长/地标窗口（Preq-Grow）和滑动窗口（Preq-Slide）。

有关方法的完整描述，请参阅第2节。这些方法中的折叠次数K或重复次数nreps为10，这是文献中常用的设置。CV-Mod和CV-hvBl中删除的观察结果数量（参见

第2节）是每个时间序列的嵌入维数p。

3.4.3 Evaluation metrics

Our goal is to study which estimation method provides a $\hat{g}$ that best approximates L^m. Let $\hat{g}_i^m$ denote the estimated loss by the learning model m using the estimation method g on the estimation set, and L^m denote the ground truth loss of learning model m on the test set. The objective is to analyze how well $\hat{g}_i^m$ approximates L^m. This is quantified by the absolute predictive accuracy error (APAE) metric and the predictive accuracy error (PAE) [6]:

$$\text{APAE} = |\hat{g}_i^m - L^m| \tag{1}$$

$$\text{PAE} = \hat{g}_i^m - L^m \tag{2}$$

The APAE metric evaluates the error size of a given estimation method. On the other hand, PAE measures the error bias, i.e., whether a given estimation method is under-estimating or over-estimating the true error.

Another question regarding evaluation is how a given learning model is evaluated regarding its forecasting accuracy. In this work we evaluate models according to root mean squared error (RMSE). This metric is traditionally used for measuring the differences between the estimated values and actual values.

3.4.4 Learning algorithm

The results shown in this work are obtained using a rule-base regression system Cubist [23], a variant of Quinlan's model tree [33]. This method presented the best forecasting results among several other predictive models in a recent study [11]. Notwithstanding, other learning algorithms were tested, namely the lasso [39] and a random forest [41]. The conclusions drawn using these algorithms are similar to the ones reported in the next sections.

4 Empirical experiments

4.1 Results with synthetic case study

In this section we start by analysing the average rank, and respective standard deviation, of each estimation method and for each synthetic scenario (S1, S2, and S3), according to the metric APAE. For example, a rank of 1 in a given Monte Carlo repetition means that the respective method was the best estimator in that repetition. These analyses are reported in Figures 10–12. This initial experiment is devised to reproduce the results by Bergmeir et al. [6]. Later, we will analyse how these results compare when using real-world time series.

The results shown by the average ranks corroborate those presented by Bergmeir et al. [6]. That is, cross validation approaches generally perform better (i.e., show a lower average rank) relative to the simple out-of-sample procedure `Holdout`. This can be concluded from all three scenarios: S1, S2, and S3.

3.4.3评价指标

我们的目标是研究哪种估计方法提供了最好的近似L的估计。令g表示学习模型m使用估计方法g在估计集上的估计损失，并且L表示学习模型m在测试集上的真实损失。目的是分析如何近似L.这是由绝对预测准确性误差（APAE）度量和预测准确性误差（PAE）量化的[6]：

$$\text{Apache} = |\text{标准} g — l| \tag{1}$$

$$\text{PAE} = 100g - L \tag{2}$$

APAE度量评估给定估计方法的误差大小。另一方面，PAE测量误差偏差，即，给定的估计方法是低估还是高估真实误差。

关于评估的另一个问题是如何评估给定的学习模型的预测准确性。在这项工作中，我们评估模型根据均方根误差（RMSE）。该度量传统上用于测量估计值和实际值之间的差异。

3.4.4学习算法

这项工作中显示的结果是使用基于规则的回归系统Cubist [23]获得的，Cubist是Quinlan模型树的变体[33]。在最近的一项研究中，该方法在其他几种预测模型中呈现了最佳的预测结果[11]。尽管如此，其他学习算法也进行了测试，即套索[39]和随机森林[41]。使用这些算法得出的结论与下一节中报告的结论相似。

4经验实验

4.1综合案例研究结果

在本节中，我们首先根据APAE指标分析每种估计方法和每种合成情景（S1，S2和S3）的平均排名和相应的标准差。例如，给定Monte Carlo重复中的秩为1意味着相应的方法是该重复中的最佳估计量。这些分析报告见图10–12。这个最初的实验旨在重现Bergmeir等人的结果。[6]。稍后，我们将分析使用真实世界时间序列时这些结果的比较。

平均秩显示的结果证实了Bergmeir等人提出的结果。[6]。也就是说，交叉验证方法通常执行得更好（即，显示出较低的平均等级）。这可以从所有三个场景得出结论：S1、S2和S3。

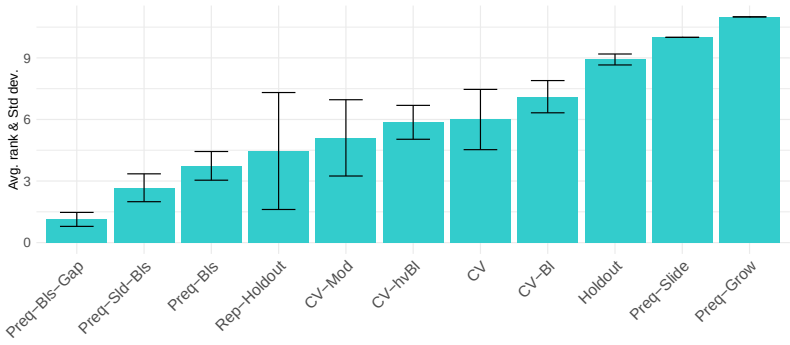


Figure 10: Average rank and respective standard deviation of each estimation methods in case study S1

Focusing on scenario S1, the estimation method with the best average rank is `Preq-Bls-Gap`, followed by the other two prequential variants (`Preq-Sld-Bls`, and `Preq-Bls`). Although the `Holdout` procedure is clearly a relatively poor estimator (worst average rank), the repeated holdout in multiple testing periods (`Rep-Holdout`) shows a better average rank than the cross validation procedures (though with a large standard deviation). Among cross validation procedures, `CV-Mod` presents the best average rank.

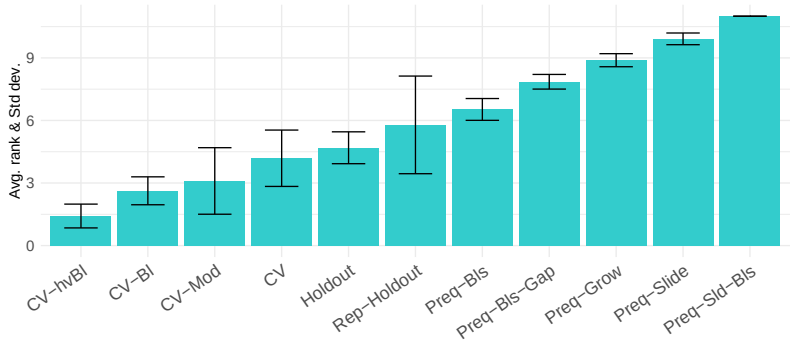


Figure 11: Average rank and respective standard deviation of each estimation methods in case study S2

Scenario S2 shows a seemingly different story relative to S1. In this problem, the prequential variants present the worst average rank, while the cross validation procedures show the best estimation ability. Among all, `CV-hvBl` shows the best average rank. Moreover, `Rep-Holdout` presents again a large standard deviation in rank, relative to

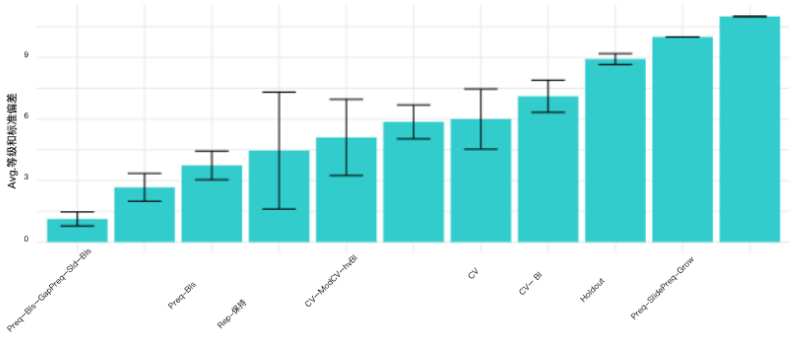


图10：每种估计方法的平均秩和各自的标准差
在案例研究S1中，

聚焦于场景S1，具有最佳平均秩的估计方法是Preq−Bls−Gap，其次是其他两个先决变量（Preq−Sld−Bls和Preq−Bls）。虽然保持程序显然是一个相对较差的估计（最差平均秩），在多个测试期间的重复保持（Rep−Holdout）显示出比交叉验证程序更好的平均秩（尽管标准差较大）。在交叉验证程序中，CV−Mod呈现最好的平均等级。

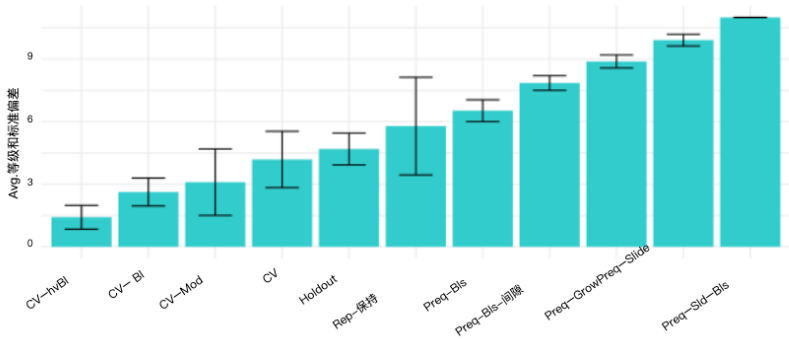


图11：每种估计方法的平均秩和各自的标准差
在案例研究S2中，

场景S2显示了一个相对于S1的看似不同的故事。在这个问题中，序变量呈现最差的平均秩，而交叉验证程序显示出最好的估计能力。其中，CV−hvBl显示最好的平均等级。此外，Rep−Holdout再次呈现出较大的排名标准差，相对于

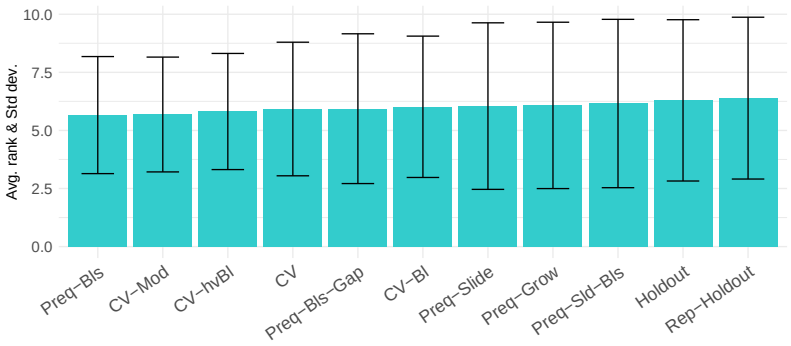


Figure 12: Average rank and respective standard deviation of each estimation methods in case study S3

the remaining estimation methods.

Regarding the scenario S3, the outcome is less clear than the previous two scenarios. The methods show a closer average rank among them, with large standard deviations.

In summary, this first experiment corroborates the experiment carried our by Bergmeir et al. [6]. Notwithstanding, other methods that the authors did not test show an interesting estimation ability in these particular scenarios, namely the prequential variants.

The synthetic scenarios comprise time series that are stationary. However, real-world time series often comprise complex dynamics that break stationarity. When choosing a performance estimation method one should take this issue into consideration. To account for time series stationarity, in the next section we analyze the estimation methods using real-world time series. We will also control for time series stationarity to study its impact on the results.

4.2 Results with real-world case study

In this section we analyze the performance estimation ability of each method using a case study comprised of real-world time series from different domains.

4.2.1 Main results

To accomplish this in Figure 13 we start by analyzing the average rank, and respective standard deviation, of each estimation method using the APAE metric. This graphic tells a different story relative to the synthetic case study. Particularly, the `Rep-Holdout` and `Holdout` show the best estimation ability in terms of the average rank. The method `CV-Bl` is the best estimator among the cross-validation procedures.

In order to study the direction of the estimation error, in Figure 14 we present for each method the percentual difference between the estimation error and the true error according to the PAE metric. In this graphic, values below the zero line denote

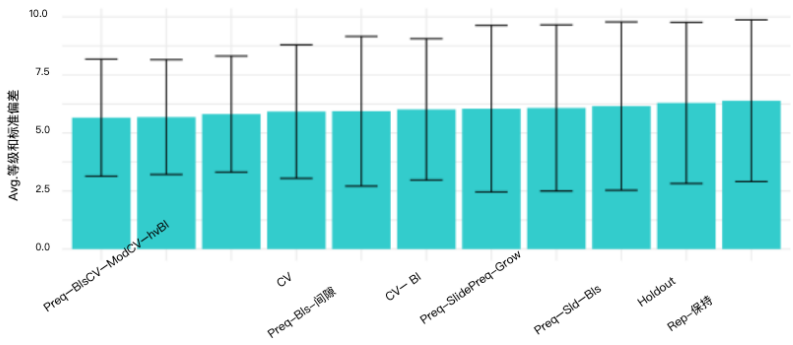


图12：每种估计方法的平均秩和各自的标准差

在案例研究S3中，

其余的估计方法。

关于情景S3，结果不如前两个情景清楚。

这些方法显示出它们之间更接近的平均排名，具有较大的标准差。

总之，第一个实验证实了Bergmeir等人进行的实验[6]。尽管如此，作者没有测试的其他方法在这些特定场景中显示出有趣的估计能力，即先决变量。

合成情景包括静止的时间序列。然而，现实世界的时间序列往往包括复杂的动态，打破平稳性。在选择性能估计方法时，应该考虑这个问题。为了说明时间序列的平稳性，在下一节中，我们将分析使用真实时间序列的估计方法。我们还将控制时间序列平稳性，以研究其对结果的影响。

4.2真实世界案例研究的结果

在本节中，我们使用一个由来自不同领域的真实时间序列组成的案例研究来分析每种方法的性能估计能力。

4.2.1主要成果

为了实现这一点，在图13中，我们首先使用APAE指标分析每种估计方法的平均排名和各自的标准差。此图讲述了与综合案例研究不同的故事。特别地，Rep–Holdout和Holdout在平均秩方面表现出最好的估计能力。方法CV–B1是交叉验证程序中的最佳估计量。

为了研究估计误差的方向，在图14中，我们根据PAE度量为每种方法呈现了估计误差和真实误差之间的百分比差异。在此图中，零线以下的值表示

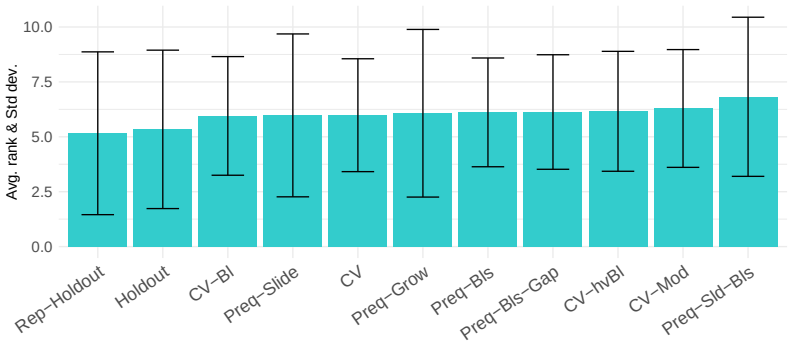


Figure 13: Average rank and respective standard deviation of each estimation methods in case study RWTS

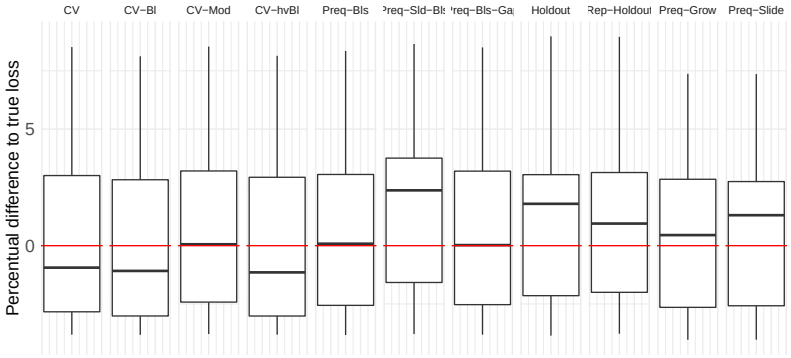


Figure 14: Percentual difference of the estimated loss relative to the true loss for each estimation method in the RWTS case study. Values below the zero line represent under-estimations of error. Conversely, values above the zero line represent over-estimations of error.

under-estimations of error, while values above the zero line represent over-estimations. In general, cross-validation procedures tend to under-estimate the error (i.e. are optimistic estimators), while the prequential and out-of-sample variants tend to over-estimate the error (i.e. are pessimistic estimators).

This result corroborates the results on Twitter time-ordered data [30]. The authors found that all variants of cross-validation procedures tend to under-estimate the errors, while the out-of-sample procedures tend to over-estimate them.

We also study the statistical significance of the obtained results in terms of error size (APAE) according to a Bayesian analysis [2]. Particularly, we employed the Bayes sign test to compare pairs of methods across multiple problems. We define the *region*

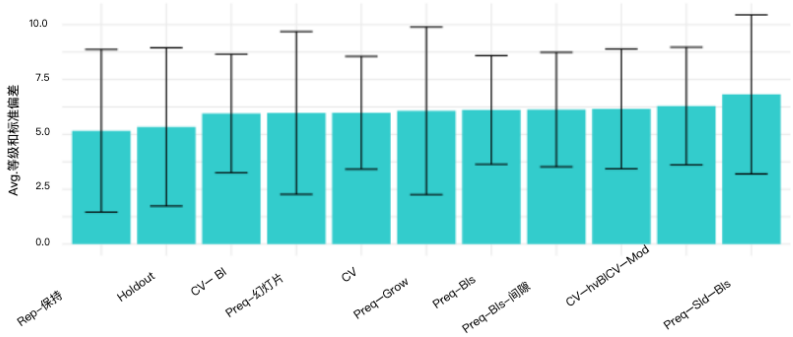


图13：每种估计方法的平均秩和各自的标准差
在案例研究RWTS中

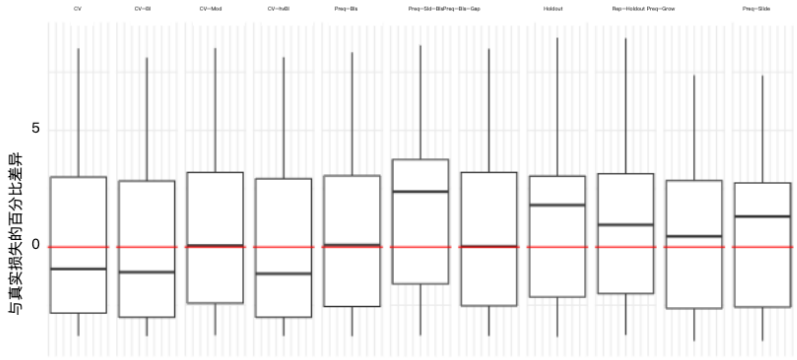


图14：每一个国家的估计损失相对于实际损失的百分比差异
RWTS案例研究中的估计方法。零线以下的值表示低估了误差。相反，高于零线的值表示对误差的高估。

误差的低估，而高于零线的值表示高估。一般来说，交叉验证程序倾向于低估误差（即乐观估计），而先决变量和样本外变量倾向于高估误差（即悲观估计）。

这一结果证实了Twitter时间排序数据的结果[30]。作者发现，交叉验证程序的所有变体都倾向于低估误差，而样本外程序则倾向于高估误差。

我们还根据贝叶斯分析研究了所获得结果在误差大小（APAE）方面的统计显著性[2]。特别是，我们采用贝叶斯符号检验来比较多个问题中的方法对。我们定义这个区域

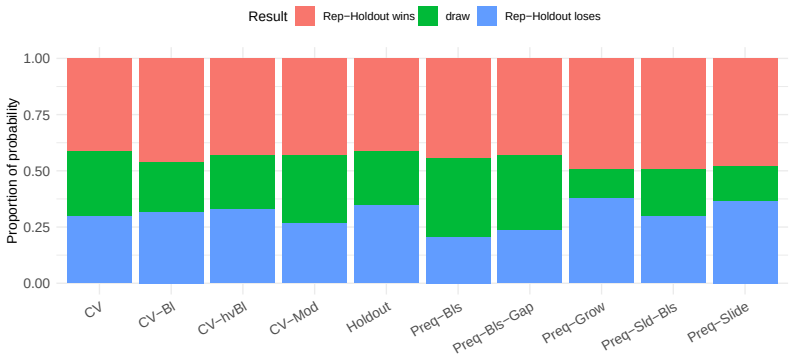


Figure 15: Proportion of probability of the outcome when comparing the performance estimation ability of the respective estimation method with the `Rep-Holdout` method. The probabilities are computed using the Bayes sign test.

of practical equivalence [2] (ROPE) to be the interval [-2.5%, 2.5%] in terms of APAE. Essentially, this means that two methods show indistinguishable performance if the difference in performance between them falls within this interval. For a thorough description of the Bayesian analysis for comparing predictive models we refer to the work by Benavoli et al [2].

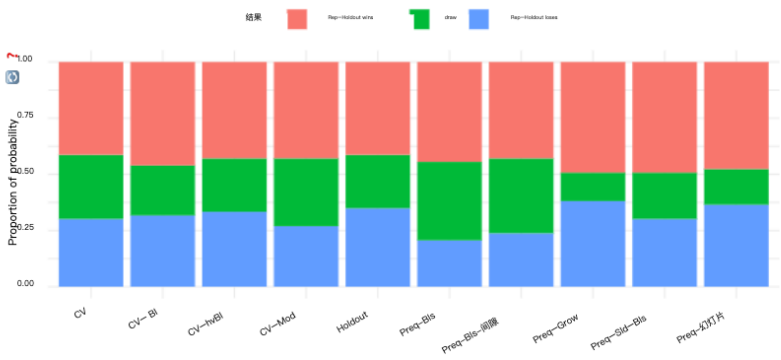
In this experiment we fix the method `Rep-Holdout` as the baseline, since it is the one showing the best average rank (Figure 13). According to the illustration in Figure 15, the probability of `Rep-Holdout` winning (i.e., showing a significantly better estimation ability) is generally larger than the opposite.

4.2.2 Controlling for stationarity

After analyzing the synthetic case study we hypothesized that the results were biased due to the stationarity assumption. In this section we repeat the average rank experiment in the real-world case study controlling for stationarity. We consider a time series stationary according to the analysis carried out in Section 3.2.1.

In Figure 16 we present the results considering only the real world time series that are stationary. According to the average rank, the typical cross-validation approach `CV` presents the best estimation ability, followed by `Rep-Holdout`.

In Figure 17 we present a similar analysis for the non-stationary time series, whose results are considerably different relative to stationary time series. In this scenario, `CV` is one of the worst estimator according to average rank. The out-of-sample approaches `Holdout` and `Rep-Holdout` present the best estimation ability.



We define the region Figure 15: Proportion of probability of the outcome when comparing the performance estimation ability of the respective estimation method with the Rep−Holdout method. The probabilities are computed using the Bayes sign test.

of practical equivalence [2] (ROPE) to be the interval [−2.5%, 2.5%] in terms of APAE. Essentially, this means that two methods show indistinguishable performance if the difference in performance between them falls within this interval. For a thorough description of the Bayesian analysis for comparing predictive models we refer to the work by Benavoli et al [2].

In this experiment we fix the method Rep−Holdout as the baseline, since it is the one showing the best average rank (Figure 13). According to the illustration in Figure 15, the probability of Rep−Holdout winning (i.e., showing a significantly better estimation ability) is generally larger than the opposite.

4.2.2 Controlling for stationarity

在分析了综合案例研究后，我们假设由于平稳性假设，结果是有偏差的。在本节中，我们在控制平稳性的真实案例研究中重复平均秩实验。根据3.2.1节中的分析，我们认为时间序列是平稳的。

在图16中，我们给出了仅考虑平稳的真实的世界时间序列的结果。根据平均秩，典型的交叉验证方法CV表现出最好的估计能力，其次是Rep−Holdout。

在图17中，我们对非平稳时间序列进行了类似的分析，其结果与平稳时间序列有很大不同。在这种情况下，根据平均秩，CV是最差的估计量之一。样本外方法Holdout和Rep−Holdout具有最好的估计能力。

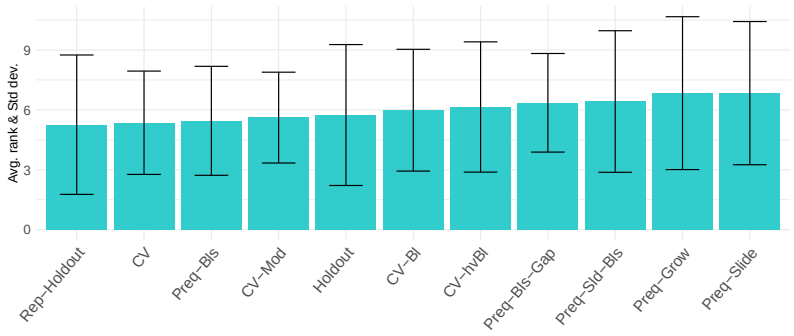


Figure 16: Average rank and respective standard deviation of each estimation methods in case study RWTS for stationary time series (31 time series).

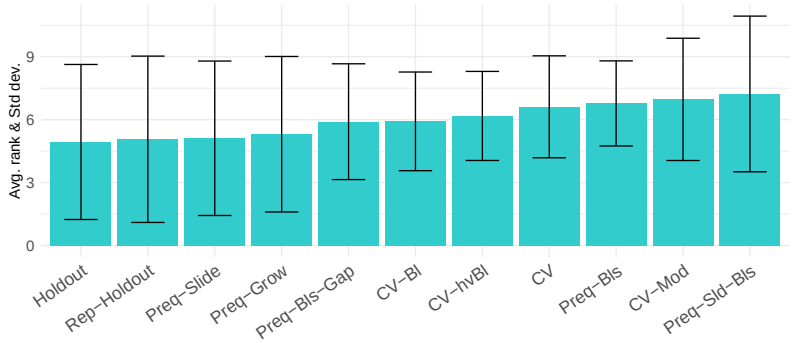


Figure 17: Average rank and respective standard deviation of each estimation methods in case study RWTS for non-stationary time series (31 time series).

4.2.3 Descriptive model

What makes an estimation method appropriate for a given time series is related to the characteristics of the data. For example, in the previous section we analyzed the impact that stationarity has in terms of what is the best estimation method.

The real-world time series case study comprises a set of time series from different domains. In this section we present, as a descriptive analysis, a tree-based model that relates some characteristics of time series according with the most appropriate estimation method for that time series. Basically, we create a predictive task in which the attributes are some characteristics of a time series, and the categorical target variable is the estimation method that best approximates the true loss in that time series. We use CART [8] (classification and regression tree) algorithm for obtaining the model for this task. The characteristics used as predictor variables are the following

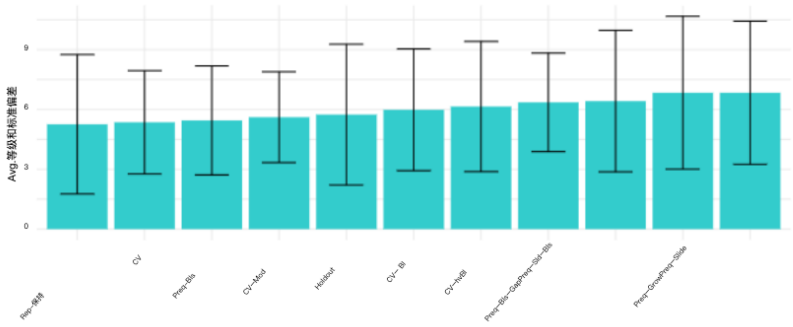


图16：平稳时间序列（31个时间序列）案例研究RWTS中每种估计方法的平均秩和各自的标准差。

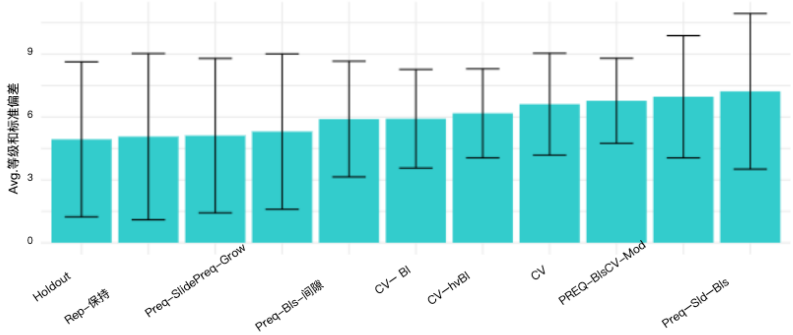


图17：非平稳时间序列（31个时间序列）案例研究RWTS中每种估计方法的平均秩和各自的标准差。

4.2.3描述模型

使估计方法适合给定时间序列的因素与数据的特征有关。例如，在上一节中，我们分析了平稳性对最佳估计方法的影响。

真实世界的时间序列案例研究包括一组来自不同领域的时间序列。在本节中，我们提出了一个基于树的模型，作为描述性分析，该模型将时间序列的一些特征与该时间序列最合适的估计方法相关联。基本上，我们创建了一个预测任务，其中属性是时间序列的一些特征，分类目标变量是最接近该时间序列中真实损失的估计方法。我们使用CART [8]（分类和回归树）算法来获得该任务的模型。用作预测变量的特征如下

summary statistics:

- **Skewness**, for measuring the symmetry of the distribution of the time series;
- 5-th and 95-th Percentiles (**Perc05**, **Perc95**) of the standardized time series;
- Acceleration (**Accel.**), as the average ratio between a simple moving average and the exponential moving average;
- Inter-quartile range (**IQR**), as a measure of the spread of the standardized time series;
- Serial correlation, estimated using a Box-Pierce test statistic;
- Long-range dependence, using a Hurst exponent estimation with wavelet transform;
- Maximum Lyapunov Exponent, as a measure of the level of chaos in the time series;
- a boolean variable, indicating whether or not the respective time series is stationary according to the wavelet spectrum test [31].

The characteristics used in the obtained decision tree are written in boldface. The decision tree is shown in Figure 18. The numbers below the name of the method in each node denote the number of times the respective method is best over the number of time series covered in that node.

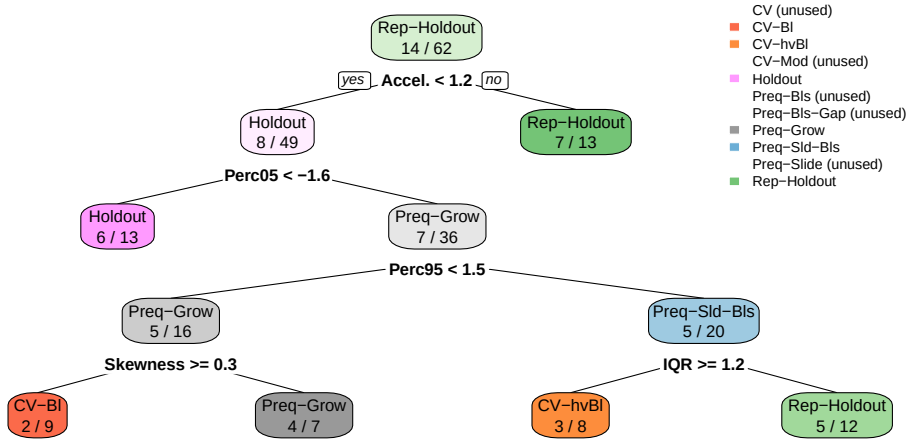


Figure 18: Decision tree that maps the characteristics of time series to the most appropriate estimation method. Graphic created using the *rpart.plot* framework [28].

统计摘要：

- 偏度，用于度量时间序列分布的对称性;
- 标准化时间序列的第5和第95百分位数（Perc 05，Perc 95）；
- 加速度（Accel.），作为简单移动平均线和指数移动平均线之间的平均比率;
- 四分位数间距，作为标准化时间序列分布的量度;
- 用Box–Pierce检验统计量估计的序列相关性;
- 用带小波变换的Hurst指数估计的长程相关性;
- 最大李雅普诺夫指数，作为时间序列中混沌程度的度量;
- 一个布尔变量，表示根据小波谱测试，相应的时间序列是否是平稳的[31]。

在所获得的决策树中使用的特征以黑体字书写。决策树如图18所示。每个节点中方法名称下面的数字表示相应方法在该节点中覆盖的时间序列数量上最佳的次数。

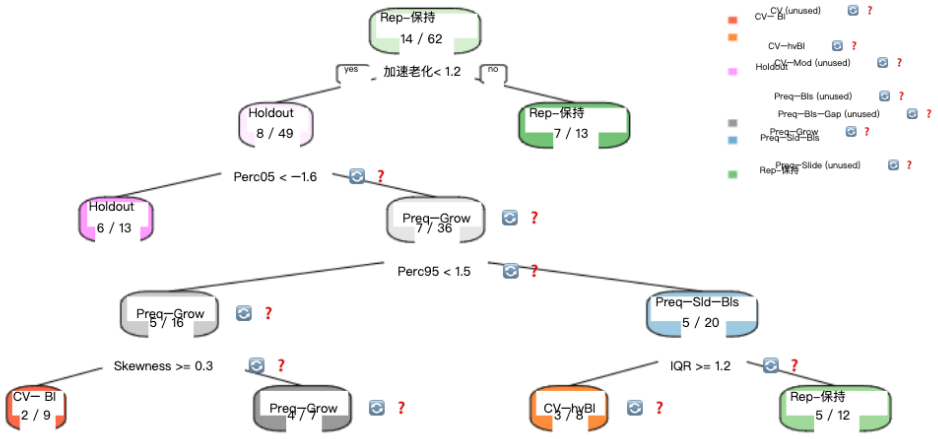


图18：将时间序列的特征映射到最合适的
适当的估计方法。使用rpart.plot框架创建的图形[28]。

Some of the estimation methods do not appear in the tree model. The tree leaves, which represent a decision, are dominated by the `Rep-Holdout` and `Holdout` estimation methods. The estimation methods `CV-Bl`, `Preq-Slide`, `Preq-Grow`, and `CV-hvBl` also appear in other leaves.

The estimation method in the root node is `Rep-Holdout`, which is the best method most of the times across the 62 time series. The first split is performed according to the acceleration characteristic of time series. Basically, if acceleration is not below 1.2, the tree leads to a leaf node with `Rep-Holdout` as the most appropriate estimation method. Otherwise, the tree continues with more tests in order to find the most suitable estimation method for each particular scenario.

5 Discussion

5.1 Impact of the results

In the experimental evaluation we compare several performance estimation methods in two distinct scenarios: (1) a synthetic case study in which artificial data generating processes are used to create stationary time series; and (2) a real-world case study comprising 62 time series from different domains. The synthetic case study is based on the experimental setup used in previous studies by Bergmeir et al. for the same purpose of evaluating performance estimation methods for time series forecasting tasks [4–6].

Bergmeir et al. show in previous studies [3, 4] that the blocked form of cross-validation, denoted here as `CV-Bl`, yields more accurate estimates than a simple out-of-sample evaluation (`Holdout`) for stationary time series forecasting tasks. The method `CV` is also suggested to be "a better choice than OOS[`Holdout`] evaluation" as long as the data are well fitted by the model [6]. To some extent part of the results from our experiments corroborate these conclusions. Specifically, this is verified by the APAE incurred by the estimation procedures in the synthetic case studies.

However, according to our experiments, the results from the synthetic stationary case studies do not reflect those obtained using real-world time series. In general, holdout applied with multiple randomized testing periods (`Rep-Holdout`) provides the most accurate performance estimates. Notwithstanding, for stationary time series `CV` also shows a competitive estimation ability.

In a real-world environment we are prone to deal with time series with complex structures and different sources of non-stationary variations. These comprise nuances of the future that may not have revealed themselves in the past [38]. Consequently, we believe that in these scenarios, `Rep-Holdout` is a better option as performance estimation method relative to cross-validation approaches.

5.2 On the importance of data size

The temporal order preservation by OOS approaches, albeit more realistic, comes at a cost since less data is available for estimating predictive performance. As Bergmeir et al. [6] argue, this may be important for small data sets, where a more efficient use of

有些估计方法在树模型中没有出现。代表决策的树叶由Rep-Holdout和Holdout估计方法支配。估计方法CV-BI、Preq-Slide、Preq-Grow和CV-hvBI也出现在其他叶中。

根节点的估计方法是Rep-Holdout，这是62个时间序列中大多数时候最好的方法。根据时间序列的加速度特性进行第一次分裂。基本上，如果加速度不低于1.2，则树导致叶节点，其中Rep-Holdout作为最适当的估计方法。否则，树继续进行更多的测试，以便为每个特定场景找到最合适的估计方法。

5讨论

5.1结果的影响

在实验评估中，我们比较了两种不同情况下的几种性能估计方法：（1）人工数据生成过程用于创建固定时间序列的合成案例研究;以及（2）包括来自不同领域的62个时间序列的真实案例研究。综合案例研究基于Bergmeir等人先前研究中使用的实验设置。用于评估时间序列预测任务的性能估计方法的相同目的[4–6]。

Bergmeir等人。在以前的研究中[3，4]表明，交叉验证的块形式，在这里表示为CV-BI，对于固定时间序列预测任务，比简单的样本外评估（Holdout）产生更准确的估计。只要模型能很好地拟合数据，CV方法也被认为是“比OOS[保持]评价更好的选择”[6]。我们的部分实验结果在一定程度上证实了这些结论。具体而言，这是验证了合成案例研究中的估计程序所产生的APAE。

然而，根据我们的实验，从合成平稳的情况下，研究的结果并不反映使用现实世界的时间序列所获得的。一般来说，应用于多个随机测试期的保留（Rep-Holdout）提供了最准确的性能估计。尽管如此，对于平稳时间序列，CV也显示出有竞争力的估计能力。

在现实环境中，我们倾向于处理具有复杂结构和不同来源的非平稳变化的时间序列。这些包括过去可能没有揭示的未来的细微差别[38]。因此，我们认为，在这些情况下，Rep-Holdout是一个更好的选择，相对于交叉验证方法的性能估计方法。

5.2数据大小的重要性

OOS方法的时间顺序保留虽然更现实，但也有代价，因为可用于估计预测性能的数据较少。正如Bergmeir等人[6]所认为的那样，这对于小数据集可能很重要，在小数据集中，更有效地使用

the data (e.g. CV) may be beneficial. However, during our experimental evaluation we did not found compelling evidence to back this claim. In the reported experiments we fixed the data size to 200 observations, as Bergmeir et al [6] did. In order to control for data size, we varied this parameter from a size of 100 to a size of 3000, by intervals of 100 (100, 200, ..., 3000). The experiments did not provide any evidence that the size of the synthetic time series had a noticeable effect on the error of estimation methods.

In our experiments the size of the time series in the real-world case study are in the order of a few thousands. For large scale data sets the recommendation by Dietterich [13], and usually adopted in practice, is to apply a simple out-of-sample estimation procedure (Holdout).

5.3 Scope of the real-world case study

In this work we center our study on univariate numeric time series. Nevertheless, we believe that the conclusions of our study are independent of this assumption and should extend for other types of time series. The objective is to predict the next value of the time series, assuming immediate feedback from the environment. Moreover, we focus on time series with a high sampling frequency, specifically, half-hourly, hourly, and daily data. The main reason for this is because high sampling frequency is typically associated with more data, which is important for fitting the predictive models from a machine learning point of view. Standard forecasting benchmark data are typically more centered around low sampling frequency time series, for example the M competition data [26].

6 Final remarks

In this paper we analyse the ability of different methods to approximate the loss that a given predictive model will incur on unseen data. This error estimation process is performed in every machine learning task for model selection and hyper-parameter tuning. We focus on performance estimation for time series forecasting tasks. Since there is currently no settled approach for performance estimation in these settings, our objective is to compare different available methods and test their suitability.

We analyse several methods that can be generally split into out-of-sample approaches and cross-validation methods. These were applied to two case studies: a synthetic environment with stationary time series and a real-world scenario with potential non-stationarities.

In a stationary setting the cross-validation variants are shown to have a competitive estimation ability. However, when non-stationarities are present, they systematically provide worse estimations than the out-of-sample approaches.

Bergmeir et al. [4–6] suggest that for stationary time series one should use cross-validation in a blocked form (CV-Bl). On the other hand, for real-world time series with potential non-stationarities we conclude that approaches that maintain the temporal order of data provide better error estimations. In particular, out-of-sample

数据（例如CV）可能是有益的。然而，在我们的实验评估中，我们没有发现令人信服的证据来支持这一说法。在报告的实验中，我们将数据大小固定为200个观测值，正如Bergmeir等人[6]所做的那样。为了控制数据大小，我们将此参数从大小100更改为大小3000，间隔为100（100，200，...，3000）。实验没有提供任何证据表明合成时间序列的大小对估计方法的误差有明显的影响。

在我们的实验中，时间序列的大小在现实世界的案例研究中是在几千的顺序。对于大规模数据集，Dietterich [13]的建议（通常在实践中采用）是应用简单的样本外估计程序（保持）。

5.3真实世界案例研究的范围

在这项工作中，我们的研究中心单变量数值时间序列。尽管如此，我们相信我们的研究结论是独立于这一假设，并应扩展到其他类型的时间序列。目标是预测时间序列的下一个值，假设来自环境的即时反馈。此外，我们专注于具有高采样频率的时间序列，特别是半小时，每小时和每日数据。其主要原因是因为高采样频率通常与更多数据相关联，这对于从机器学习的角度拟合预测模型非常重要。标准预测基准数据通常更集中于低采样频率时间序列，例如M竞争数据[26]。

6最后评论

在本文中，我们分析了不同方法的能力，以近似的损失，一个给定的预测模型将在看不见的数据。这个误差估计过程在每个机器学习任务中执行，用于模型选择和超参数调整。我们专注于时间序列预测任务的性能估计。由于目前没有固定的方法在这些设置中的性能估计，我们的目标是比较不同的可用方法，并测试其适用性。

我们分析了几种方法，通常可以分为样本外方法和交叉验证方法。这些被应用到两个案例研究：一个合成的环境与固定的时间序列和现实世界的情况下，潜在的nonstationarities。

在一个固定的设置中，交叉验证的变体被证明具有竞争力的估计能力。然而，当存在非平稳性时，它们系统地提供比样本外方法更差的估计。

Bergmeir等人。[4–6]建议对于平稳时间序列，应该使用块形式的交叉验证（CV–BI）。另一方面，对于具有潜在非平稳性的真实世界时间序列，我们得出结论，保持数据时间顺序的方法提供了更好的误差估计。特别是，样本外

applied in multiple testing periods (`Rep-Holdout`) is recommended. In the interest of reproducibility, the methods and data sets are publicly available at `https://github.com/vcerqueira/performance_estimation`.

References

[1] Arlot, S., Celisse, A., et al.: A survey of cross-validation procedures for model selection. Statistics surveys **4**, 40–79 (2010)

[2] Benavoli, A., Corani, G., Demšar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. The Journal of Machine Learning Research **18**(1), 2653–2688 (2017)

[3] Bergmeir, C., Benitez, J.M.: Forecaster performance evaluation with cross-validation and variants. In: Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on, pp. 849–854. IEEE (2011)

[4] Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences **191**, 192–213 (2012)

[5] Bergmeir, C., Costantini, M., Benítez, J.M.: On the usefulness of cross-validation for directional forecast evaluation. Computational Statistics & Data Analysis **76**, 132–143 (2014)

[6] Bergmeir, C., Hyndman, R.J., Koo, B.: A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics & Data Analysis **120**, 70–83 (2018)

[7] Bifet, A., Kirkby, R.: Data stream mining a practical approach (2009)

[8] Breiman, L.: Classification and regression trees. Routledge (2017)

[9] Brockwell, P.J., Davis, R.A.: Time series: theory and methods. Springer Science & Business Media (2013)

[10] Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrated ensemble for time series forecasting. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 478–494. Springer (2017)

[11] Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrage of forecasting experts. Machine Learning pp. 1–32 (2018)

[12] Cerqueira, V., Torgo, L., Smailović, J., Mozetič, I.: A comparative study of performance estimation methods for time series forecasting. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 529–538. IEEE (2017)

[13] Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation **10**(7), 1895–1923 (1998)

建议在多个测试期应用样本外（Rep-Holdout）。为了重现性，方法和数据集可在https：//github.com/vcerqueira/performance_estimation上公开获得。

参考文献

[1]Arlot，S.，Celisse，A.，等：模型选择的交叉验证程序综述。统计调查4，40–79（2010年）

[2]Benavoli，A.，Corani，G.，Dem Bassar，J.，Zaffalon，M.：是时候改变了：通过贝叶斯分析比较多个分类器的教程。The Journal of Machine Learning Research 18（1），2653–2688（2017）

[3]Bergmeir，C.，贝尼特斯，J.M.：用交叉验证和变量评估预报器性能。在：智能系统设计与应用（ISDA），2011年第11届国际会议上，pp。849–854. IEEE（2011年）

[4]Bergmeir，C.，Ben 'itez，J.M.：时间序列预测器评估中交叉验证的使用。信息科学191，192–213（2012）

[5]Bergmeir，C.，Costantini，M.，Ben 'itez，J.M.：方向性预测评估之交叉验证之有用性。计算统计与数据分析76，132–143（2014）

[6]Bergmeir，C.，Hyndman，R.J.，古，B.：自回归时间序列预测交叉验证有效性的注记。计算统计与数据分析120，70–83（2018）

[7]Bifet，A.，柯克比：数据流挖掘的实用方法（2009）

[8]Breiman，L.：分类和回归树电影院（2017）

[9]布罗克韦尔，P.J.，Davis，R.A.：时间序列：理论与方法。施普林格科学与商业媒体（2013）

[10]Cerqueira，V.，托尔戈湖，平托，F.，Soares，C.：时间序列的仲裁集合预测。在：联合欧洲会议机器学习和知识发现数据库，pp。478–494.施普林格（2017）

[11]Cerqueira，V.，托尔戈湖，平托，F.，Soares，C.：预测专家的套利。机器学习1–32（2018）

[12]Cerqueira，V.，托尔戈湖，Smailovi'c，J.，莫泽蒂·穆塞韦尼茨：的比较研究时间序列预测的性能估计方法。2017 IEEE International Conference on Data Science and Advanced Analytics（DSAA），2017年10月17日，第100页。529–538. IEEE（2017）

[13]Dietterich，T.G.：用于比较监督分类的近似统计检验–阳离子学习算法Neural computation 10（7），1895–1923（1998）

[14] Fildes, R.: Evaluation of aggregate and individual forecast method selection rules. Management Science **35**(9), 1056–1065 (1989)

[15] Gama, J., Sebastião, R., Rodrigues, P.P.: On evaluating stream learning algorithms. Machine learning **90**(3), 317–346 (2013)

[16] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 44 (2014)

[17] Geisser, S.: The predictive sample reuse method with applications. Journal of the American statistical Association **70**(350), 320–328 (1975)

[18] Hart, J.D., Wehrly, T.E.: Kernel regression estimation using repeated measurements data. Journal of the American Statistical Association **81**(396), 1080–1088 (1986)

[19] Hyndman, R.: Time series data library. `http://data.is/TSDLdemo`. Accessed: 2017-12-11

[20] Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)

[21] Kennel, M.B., Brown, R., Abarbanel, H.D.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical review A **45**(6), 3403 (1992)

[22] Koprinska, I., Rana, M., Agelidis, V.G.: Yearly and seasonal models for electricity load forecasting. In: Neural Networks (IJCNN), The 2011 International Joint Conference on, pp. 1474–1481. IEEE (2011)

[23] Kuhn, M., Weston, S., Keefer, C., code for Cubist by Ross Quinlan, N.C.C.: Cubist: Rule- and Instance-Based Regression Modeling (2014). R package version 0.0.18

[24] Kwiatkowski, D., Phillips, P.C., Schmidt, P., Shin, Y.: Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? Journal of econometrics **54**(1-3), 159–178 (1992)

[25] Lichman, M.: UCI machine learning repository (2013). URL `http://archive.ics.uci.edu/ml`

[26] Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R.: The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of forecasting **1**(2), 111–153 (1982)

[27] McQuarrie, A.D., Tsai, C.L.: Regression and time series model selection. World Scientific (1998)

[14]菲尔德斯，R.：综合和个别预测方法选择规则的评估。
Management Science 35（9），1056–1065（1989）

[15]Gama，J.，Sebasti Pastao河，罗德里格斯，P. P.：关于流学习算法的评价
Rithms机器学习90（3），317–346（2013）

[16]G兹利奥莱，特斯特切岛，Bifet，A.，Pechenizkiy，M.，Bouchachia，A.：概念
漂移适应研究综述。ACM计算调查（CSUR）46（4），44（2014）

[17]Geisser，S.：预测样本重用方法及其应用。Journal of the
美国统计协会70（350），320–328（1975）

[18]哈特，J.D.，Wehrly，T. E.：使用重复测量的核回归估计–
段数据。美国统计协会杂志81（396），1080–1088（1986）

[19]Hyndman，R.：时间序列数据库。http://data.is/TSDLdemo.访问日期：
2017–12–11

[20]Hyndman，R.J.，Athanasopoulos，G.：预测：原则与实践。
OTexts（2018）

[21]犬舍，医学学士，布朗河，澳–地Abarbanel，H.D.：确定嵌入维数
用于使用几何构造的相空间重构。Physical review A 45（6），3403（1992）

[22]科普林斯卡岛Rana，M.，V.G. Agelovan：电力的年度和季节模型
负荷预测在：神经网络（IJCNN），2011年国际联合会议上，pp。1474–1481. IEEE（2011
年）

[23]Kuhn，M.，韦斯顿，S.，Keefer，C.，罗斯·昆兰（Ross Quinlan），N.C.C.
Cubist：基于规则和实例的回归建模（2014）。R软件包版本0.0.18

[24]Kwiatkowski，D.，菲利普斯，P.C.，施密特，P.，Shin，Y.：测试零假设–
平稳性与单位根的对立：我们如何确定经济时间序列有单位根？计量经济学杂志54（1–
3），159–178（1992）

[25]Lichman，M.：UCI机器学习库（2013）。http：//www.hkk.com.cn
ics.uci.edu/ml

[26]Makridakis，S。安德森，A。Carbone，R。菲尔，R。Hibon，M.，Lewandowski，
R.，Newton，J.，Parzen，E.，Winkler，R.：外推（时间序列）方法的准确性：预测竞赛
的结果。Journal of Forecasting 1（2），111–153（1982）

[27]McQuarrie，A. D.，蔡志龙：回归与时间序列模型选择。世界
科学（1998年）

[28] Milborrow, S.: rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart' (2018). URL `https://CRAN.R-project.org/package=rpart.plot`. R package version 3.0.6

[29] Modha, D.S., Masry, E.: Prequential and cross-validated regression estimation. Machine Learning **33**(1), 5–39 (1998)

[30] Mozetič, I., Torgo, L., Cerqueira, V., Smailović, J.: How to evaluate sentiment classifiers for Twitter time-ordered data? PLoS ONE **13**(3), e0194317 (2018)

[31] Nason, G.: A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **75**(5), 879–904 (2013)

[32] Oliveira, M., Torgo, L., Costa, V.S.: Evaluation procedures for forecasting with spatio-temporal data. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 703–718. Springer (2018)

[33] Quinlan, J.R.: Combining instance-based and model-based learning. In: Proceedings of the tenth international conference on machine learning, pp. 236–243 (1993)

[34] Racine, J.: Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. Journal of econometrics **99**(1), 39–61 (2000)

[35] Snijders, T.A.: On cross-validation for predictor evaluation in time series. In: On Model Uncertainty and its Statistical Implications, pp. 56–69. Springer (1988)

[36] Stone, M.: Cross-validation and multinomial prediction. Biometrika pp. 509–515 (1974)

[37] Takens, F.: Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80, chap. Detecting strange attractors in turbulence, pp. 366–381. Springer Berlin Heidelberg, Berlin, Heidelberg (1981). DOI 10.1007/BFb0091924

[38] Tashman, L.J.: Out-of-sample tests of forecasting accuracy: an analysis and review. International journal of forecasting **16**(4), 437–450 (2000)

[39] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)

[40] Wald, A.: Sequential analysis. Courier Corporation (1973)

[41] Wright, M.N.: ranger: A Fast Implementation of Random Forests (2015). R package version 0.3.0

Appendix

[28]米尔博罗，S.：rpart.plot：Plot 'rpart' Models：An Enhanced Version of 'plot.rpart'（2018）.网址https://CRAN.R-project.org/package=rpart.plot。
R软件包版本3.0.6

[29]Modha，D.S.，Masry，E.：预验证和交叉验证回归估计。
Machine Learning 33（1），5-39（1998）

[30]Mozeti Zaghic，I.，托尔戈湖，Cerqueira，V.，Smailovi'c，J.：如何评估Twitter时间排序数据的情感分类器？PLoS ONE 13（3），e0194317（2018）

[31]Nason，G.：二阶平稳性和近似置信度的检验-

局部平稳时间序列的局部自协方差区间。皇家统计学会杂志：系列B（统计方法）75（5），879-904（2013）

[32]Oliveira，M.，托尔戈湖，Costa，V.S.：预测的评价程序
时空数据在：联合欧洲会议机器学习和知识发现数据库，pp。703-718.施普林格（2018）

[33]Quinlan，J.R.：结合基于实例和基于模型的学习。在：Pro-
第十届机器学习国际会议的主席，pp。236-243（1993年）

[34]拉辛，J.：相关数据的一致交叉验证模型选择：hv-
块交叉验证。计量经济学杂志99（1），39-61（2000）

[35]Snijders，T.A.：时间序列中预测因子评估的交叉验证。In：On
模型的不确定性及其统计意义，pp。56-69. 03 The Dog（1988）

[36]斯通，M.：交叉验证和多项式预测。Biometrika pp. 509—515
(1974)

[37]Takens，F.：动力系统与湍流，沃里克，1980：会议录
在沃里克1979/80大学举行的研讨会，第一章。探测湍流中的奇异吸引子。366-381.
Springer柏林海德堡，柏林，海德堡（1981年）。DOI 10.1007/BFb0091924

[38]Tashman，L.J.：预测准确性的样本外检验：一个分析与再检验
风景国际预测杂志16（4），437-450（2000）

[39]Tibshirani，R.：通过套索进行回归收缩和选择。Journal of the
皇家统计学会。系列B（方法学），pp. 267-288（1996年）

[40]Wald，A.：序贯分析。05 The Dog of the Dog（1973）

[41]Wright，M.N.：ranger：A Fast Implementation of Random Forests（2015）. R
软件包版本0.3.0

附录

Table 1: Time series data sets and respective summary. The column p denotes the size of the embedding dimension, I denotes the number of differences applied to the time series to make it trend-stationary, and S represents whether or not the de-trended time series is stationary (1 if it is).

ID	Time series	Data source	Data characteristics	Size	p	I	S
1	Rotunda AEP	Porto Water Consumption from different locations in the city of Porto [10]	Half-hourly values from Nov. 11, 2015 to Jan. 11, 2016	3000	30	0	0
2	Preciosa Mar			3000	9	1	0
3	Amial			3000	11	0	0
4	Global Horizontal Radiation	Solar Radiation Monitoring [10]	Hourly values from Apr. 25, 2016 to Aug. 25, 2016	3000	23	1	0
5	Direct Normal Radiation			3000	19	1	1
6	Diffuse Horizontal Radiation			3000	18	1	1
7	Average Wind Speed			3000	10	1	0
8	Humidity	Bike Sharing [10]	Hourly values from Jan. 1, 2011 Mar. 01, 2011	1338	11	0	0
9	Windspeed			1338	12	0	1
10	Total bike rentals			1338	8	0	1
11	AeroStock 1	Stock price values from different aerospace companies [10]	Daily stock prices from January 1988 through October 1991	949	6	1	1
12	AeroStock 2			949	13	1	0
13	AeroStock 3			949	7	1	1
14	AeroStock 4			949	8	1	1
15	AeroStock 5			949	6	1	1
16	AeroStock 6			949	10	1	1
17	AeroStock 7			949	8	1	1
18	AeroStock 8			949	8	1	1
19	AeroStock 9			949	9	1	1
20	AeroStock 10			949	8	1	1
21	CO.GT	Air quality indicators in an Italian city [25]	Hourly values from Mar. 10, 2004 to Apr. 04 2005	3000	30	1	0
22	PT08.S1.CO			3000	8	1	0
23	NMHC.GT			3000	10	1	0
24	C6H6.GT			3000	13	0	1
25	PT08.S2.NMHC			3000	9	0	0
26	NOx.GT			3000	10	1	1
27	PT08.S3.NOx			3000	10	1	0
28	NO2.GT			3000	30	1	0
29	PT08.S4.NO2			3000	8	0	0
30	PT08.S5.O3			3000	8	0	1
31	Temperature			3000	8	1	0
32	RH			3000	23	1	0
33	Humidity			3000	10	1	0

表1：时间序列数据集和相应总结。列p表示大小
l表示应用于时间序列以使其趋势平稳的差异的数量，并且S表示去趋势时间序列是否是平稳的（如果是，则为1）。

ID时间序列数据源数据特征大小p l S			
1圆形大厅AEP波尔图水			3000 30 0 0
2 Preciosa Mar 3000 9 1 0	消费从	2015年11月11日至2016年1月11日半小时值	
3 Amial	中的不同位置		3000 11 0 0
4全球水平辐射	城市波尔图[10]		3000 23 1 0
5直接正常辐射3000 19 1 1	太阳辐射	2016年4月25日至2016年8月25日的小时值	
6水平辐射3000 18 1 1	监测[10]		
7平均风速			3000 10 1 0
8个湿度			1338 11 0 0
9风速小时值自2011年1月1日1338 12 0 1	自行车共享[10]		
10 自行车出租总额2011年3月1日1338 8 0 1			
11 AeroStock 1			949 6 1 1
12 AeroStock 2 949 13 1 0			
13 AeroStock 3 949 7 1 1			
14 2009年10月31日	股票价格从		
15 2009年10月31日	不同航天	1988年1月至1991年10月的每日股票价格	
16 AeroStock 6 949 10 1 1	公司[10]		
17 AeroStock 7 949 8 1 1			
18 10德国航空公司8 949 8 1 1			
19 1000 1000 1000 1000			
20 AeroStock 10			949 8 1 1
21 CO.GT			3000 30 1 0
22 粤ICP备16038888号–1			
23 NMHC.GT 3000 10 1 0			
24 C6H6.GT 3000 13 0 1			
25 PT08.S2.NMHC 3000 9 0 0			
26 NOx.GT 3000 10 1 1			
27 PT08.S3.NOx 3000 10 1 0	空气质素指标	2004年3月10日至2005年4月4日的小时值	
28 粤ICP备16033550号–1	意大利城市[25]		
29 粤ICP备16038888号–1			
30 粤ICP备15038888号–1			
31 温度3000 8 1 0			
32 RH 3000 23 1 0			
33 湿度			3000 10 1 0

Table 2: Continuation of Table 1

ID	Time series	Data source	Data characteristics	Size	p	I	S
34	Electricity Total Load	Hospital Energy Loads [10]	Hourly values from Jan. 1, 2016 to Mar. 25, 2016	3000	19	0	1
35	Equipment Load			3000	30	0	1
36	Gas Energy			3000	10	1	1
37	Gas Heat Energy			3000	13	1	1
38	Water heater Energy			3000	30	0	1
39	Total Demand	Australian Electricity [22]	Half-hourly values from Jan. 1, 1999 to Mar. 1, 1999	2833	6	0	1
40	Recommended Retail Price			2833	19	0	0
41	Sea Level Pressure	Ozone Level Detection [25]	Daily values from Jan. 2, 1998 to Dec. 31, 2004	2534	9	0	1
42	Geo-potential height			2534	7	0	1
43	K Index			2534	7	0	1
44	Flow of Vatnsdalsa river	Data market [19]	Daily, from Jan. 1, 1972 to Dec. 31, 1974	1095	11	0	0
45	Rainfall in Melbourne		Daily, from from 1981 to 1990	3000	29	0	0
46	Foreign exchange rates		Daily, from Dec. 31, 1979 to Dec. 31, 1998	3000	6	1	0
47	Max. temperatures in Melbourne		Daily, from from 1981 to 1990	3000	7	0	1
48	Min. temperatures in Melbourne		Daily, from from 1981 to 1990	3000	6	0	1
49	Precipitation in River Hirnant		Half-hourly, from Nov. 1, 1972 to Dec. 31, 1972	2928	6	1	0
50	IBM common stock closing prices		Daily, from Jan. 2, 1962 to Dec. 31, 1965	1008	10	1	0
51	Internet traffic data I		Hourly, from Jun. 7, 2005 to Jul. 31, 2005	1231	10	0	1
52	Internet traffic data II		Hourly, from Nov. 19, 2004 to Jan. 27, 2005	1657	11	1	0
53	Internet traffic data III		from Nov. 19, 2004 to Jan. 27, 2005 – Data collected at five minute intervals	3000	6	1	0
54	Flow of Jokulsa Eystri river		Daily, from Jan. 1, 1972 to Dec. 31, 1974	1096	21	0	0
55	Flow of O. Brocket		Daily, from Jan. 1, 1988 to Dec. 31, 1991	1461	6	1	0
56	Flow of Saugeen river I		Daily, from Jan. 1, 1915 to Dec. 31, 1979	1400	6	0	0
57	Flow of Saugeen river II		Daily, from Jan. 1, 1988 to Dec. 31, 1991	3000	30	0	0
58	Flow of Fisher River		Daily, from Jan. 1, 1974 to Dec. 31, 1991	1461	6	0	1
59	No. of Births in Quebec		Daily, from Jan. 1, 1977 to Dec. 31, 1990	3000	6	1	1
60	Precipitation in O. Brocket		Daily, from Jan. 1, 1988 to Dec. 31, 1991	1461	29	0	0
61	Min. temperature	Porto weather [10]	Daily values from Jan. 1, 2010 to Dec. 28, 2013	1456	8	0	1
62	Max. temperature			1456	10	0	0

表2：表1的延续

ID	时间序列	数据源	数据特征	大小	p	I	S
34	电力总负荷						3000 19 0 1
35	设备负荷3000 30 0 1	医院能源[第10话]	2016年1月1日至2016年3月25日的小时值				
36	燃气能源3000 10 1 1						
37	燃气热能3000 13 1 1						
38	热水器节能3000 30 0 1						
39	总需求	澳大利亚电力[22]	1999年1月1日至1999年3月1日的半小时值				2833 6 0 1
40	建议零售价						2833 19 0 0
41	海平面气压	臭氧水平检测[25]	1998年1月2日至2004年12月31日的日值				2534 9 0 1
42	位势高度2534 7 0 1						
43	上证综指2534 7 0 1						
44	Vatnsdalsa河流量		每日，1972年1月1日至1974年12月31日				1095 11 0 0
45	1981 – 1990年墨尔本日降雨量3000 29 0 0						
46	外汇汇率1979年12月31日至12月31日，						3000 6 1 0
47	最大温度在梅尔–伯恩		1998 1981年至1990年每日3000 7 0 1				
48	最低气温在梅尔–伯恩	数据市场[19]	1981年至1990年每日3000 6 0 1				
49	降水在River Hir–nant		1972年11月1日至1972年12月31日，每半小时一次				2928 6 1 0
50	IBM普通股收盘价价格		每日，1962年1月2日至1965年12月31日				1008 10 1 0
51	2005年6月7日至7月31日的互联网流量数据I小时，		2005				1231 10 0 1
52	2004年11月19日至2004年1月27日，互联网流量数据II每小时，		2005				1657 11 1 0
53	2004年11月19日至2005年1月27日互联网流量数据三–		每隔5分钟收集一次数据				3000 6 1 0
54	1972年1月1日至12月31日，		1974				1096 21 0 0
55	O的流动。1988年1月1日至12月31日，		1991				1461 6 1 0
56	1915年1月1日至12月31日，索金河的流量，		1979				1400 6 0 0
57	1988年1月1日至12月31日，		1991				3000 30 0 0
58	费希尔河每日流量，1974年1月1日至12月31日，		1991				1461 6 0 1
59	号出生在魁北克日报，从1977年1月1日至12月31日，		1990				3000 6 1 1
60	降水量O. 1988年1月1日至12月31日，		1991				1461 29 0 0
61	最低温度	波尔图天气[10]	2010年1月1日至2013年12月28日的日值				1456 8 0 1
62	最大温度						1456 10 0 0