2017 International Conference on Data Science and Advanced Analytics

A Comparative Study of Performance Estimation Methods for Time Series Forecasting

Vitor Cerqueira
LIAAD-INESCTEC &
University of Porto
Porto, Portugal
Email: vmac@inesctec.pt

Luis Torgo
LIAAD-INESCTEC &
University of Porto
Porto, Portugal
Email: ltorgo@inesctec.pt

Jasmina Smailović
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana
Slovenia
Email: jasmina.smailovic@ijs.si

Igor Mozetič
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana
Slovenia
Email: igor.mozetic@ijs.si

***Abstract*—Performance estimation denotes a task of estimating the loss that a predictive model will incur on unseen data. These procedures are part of the pipeline in every machine learning task and are used for assessing the overall generalisation ability of 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 these datasets and currently there is no settled way to do so. We compare different variants of cross-validation and different variants of out-of-sample approaches using two case studies: One with 53 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 synthetic time series. However, in real-world scenarios the most accurate estimates are produced by the out-of-sample methods, which preserve the temporal order of observations.**

***Keywords*-performance estimation; model selection; cross validation; time series.**

I. Introduction

Machine learning plays an increasingly important role in science and technology. Performance estimation is part of any machine learning task pipeline. This task is related to a procedure 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, meta-parameter tuning and assessing the overall generalisation ability of the models. In effect, obtaining reliable estimates of the performance of models is a critical issue in predictive analytics tasks.

Choosing a performance estimation method often depends on the data one is trying to model. For example, when one can assume independence and an identical distribution (i.i.d) among observations, cross-validation is typically the most appropriate method. This is mainly due to its efficient use of data [1].

However, there are problems in which the observations in the data are dependent, such as time series. This raises some caveats about using standard cross-validation in such datasets. Notwithstanding, there are particular time series settings in which variants of cross-validation can be used, such as in stationary or small-sized datasets where the efficient use of all the data by cross-validation is beneficial [2].

In this paper we present a comparative study of different performance estimation methods for time series forecasting task. 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 53 real-world time series with potential non-stationarities and the other is a stationary synthetic environment [2]–[4].

In this study we compare two types of 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 control the dependency among observations and account for the temporal evolution of the time series.
- Cross-validation (CV): These approaches make a more efficient use of the available data, which is beneficial in some settings [2]. 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 [2]–[4].

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?

The literature on performance estimation for time series forecasting tasks is reviewed in section II. The general methodology for performance estimation is described in section III. Afterwards, the case studies are presented in section IV and the respective experiments in the following section V. A brief discussion is carried out is section VI. Finally, section VII concludes the paper.

II. Literature Review

In this section we provide a background to this paper. We review the typical estimation methods used in time series

978-1-5090-5004-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DSAA.2017.7

forecasting and explain the motivation and originality of our work.

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 out-of-sample (OOS) testing as well as variants of the cross-validation (CV) method.

A. Out-of-sample approaches

In 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. Within this approach 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 [5] recommends employing these strategies in multiple test periods. One might create different sub-samples according to, for example, business cycles [6]. For a more general setting one can also adopt a randomized approach [7]. 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.

OOS approaches are similar to prequential or interleaved-test-then-train evaluation. 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 [8]. 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. This approach is also related to holdout evaluation for data streams in scenarios with concept drift [9, Chapter 2.2.1].

B. Cross-validation approaches

Some variants of K-fold cross-validation have been proposed specially designed for dependent data, such as time series [1].

The idea behind 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 n/K randomly assigned observations, where n 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.

Theoretical problems arise by applying this technique directly to time series data. The dependency between observations is not taken into account since cross-validation assumes that the values of the time series are i.i.d.. This might lead to overly optimistic estimations and consequently, poor generalisation ability of 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 [10]. To overcome this issue and approximate independence between the training and test sets, several methods have been proposed as variants of this procedure.

The Blocked Cross-Validation [11] procedure proposed is similar to the standard form described above. The difference is that there is no initial random shuffling of observations. This renders K blocks of contiguous observations.

The hv-Blocked Cross-Validation proposed by Racine [12] extends blocked cross-validation to further increase the independence among observations. Specifically, besides blocking the observations in each fold, it also removes adjacent observation between the training and test sets. Effectively, this creates a gap between both sets.

The Modified CV procedure [13] 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 range of the observations 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 [3].

Recently there has been some work on the usefulness of cross-validation procedures for time series forecasting tasks.

Bergmeir and Benítez [3] 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. [4] 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. [2] 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 is created. The results from the simulation suggest that, provided that the model is correctly specified, cross-validation systematically yields more accurate estimates.

Despite the results provided by these previous works we argue that they are optimistic in two ways. First, the results are 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 [5]. 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 intricacies, such as pink noise or fractional integration.

In this context, this paper provides an extensive comparison study using a wide set of performance estimation methods. 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 [2]–[4].

III. Performance Estimation Methodology

This section formalises the task of performance estimation for time series forecasting. Our main objective in this paper is to compare different performance estimation procedures and test their suitability in these settings.

A time series is a temporal sequence of values $Y = \{y_1, y_2, \ldots, y_n\}$, where y_i is the value of Y at time i and n 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_{n+1}, given the previous observations of Y.

A. Performance estimation

Performance estimation addresses the issue of estimating the predictive performance of predictive models. Frequently, the objective behind these tasks is to compare different solutions for solving a predictive task. This includes selecting among different learning algorithms and meta-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 [14] (c.f. section II 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 II several methods have been developed to cope with this issue, from out-of-sample approaches [5] to variants of the standard cross-validation, e.g., block cross-validation [11].

B. Methodology

Our goal in this paper is to compare a wide set of estimation procedures, and test their suitability for time series forecasting tasks.

In order to emulate a realistic scenario we split the 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 1. A time series Y is split into an estimation set Y^E and a subsequent validation set Y^V. First, Y^E is used to estimate $\hat{E}$, the loss estimate that a predictive model m will incur on new observations. This is accomplished by further splitting Y^E into training and test sets according to the respective estimation procedure f_i, $i \in \{1, \ldots, z\}$. The model m is built on the training set and $\hat{E}$ is computed on the test set.

Second, in order to evaluate the estimations $\hat{E}_i$ produced by the methods $f_i, i \in \{1, \ldots, z\}$, the model m is re-trained using the complete estimation set Y^E and tested on the validation set Y^V. Effectively, we obtain E, the ground true loss that m incurs on new data.

In summary, the goal of an estimation method f_i is to approximate E by $\hat{E}_i$ as well as possible. In section V-A2 we describe how to quantify this approximation.

IV. 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 [2]–[4] for addressing the issue of performance estimation for time series forecasting tasks.

A. Real-world time series

We analyse 53 time series from different domains. They have different granularity and length as well as unknown dynamics. The time series are described in Table II.

B. Synthetic time series

We use three synthetic use cases defined in previous work by Bergmeir et al. [2], [4]. 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 essentially dependent on the past 3 observations – c.f. Figure 2 for a sample graph.

S2: An invertible moving average process with lag 1 – c.f. Figure 3 for a sample graph.

S3: A seasonal auto-regressive process with lag 12 (seasonal lag 1) – c.f. Figure 4 for a sample graph.

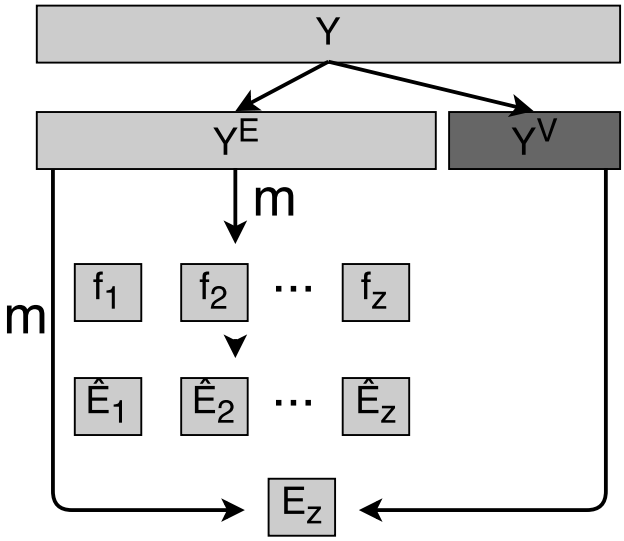

Fig. 1: Experimental comparison procedure: A time series is split into an estimation set Y^E and a subsequent validation set Y^V. The first is used to estimate the error $\hat{E}$ that the model m will incur on unseen data, using z different estimation methods. The second is used to compute the actual error E incurred by m. The objective is to approximate E by $\hat{E}$ as well as possible.

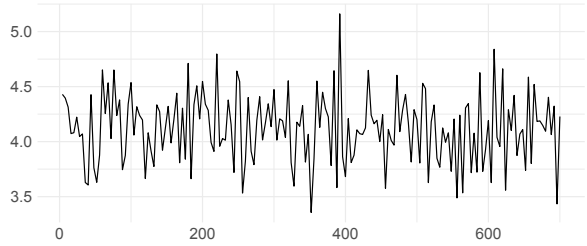

Fig. 2: Sample graph of the S1 synthetic case.

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 [3]. 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 [15]. For a complete description of data generating process we refer to the work by Bergmeir et al. [2], [3]. For each use case we performed 200 Monte Carlo simulations. In each repetition a time series with 700 values was generated.

V. Experimental Evaluation

In this section we present experiments carried out to analyse the performance estimation methods for time series forecasting tasks. These were designed to address the following research question: How do the predictive performance estimates of cross-validation methods relate to the estimates of out-of-sample approaches for time series forecasting tasks?

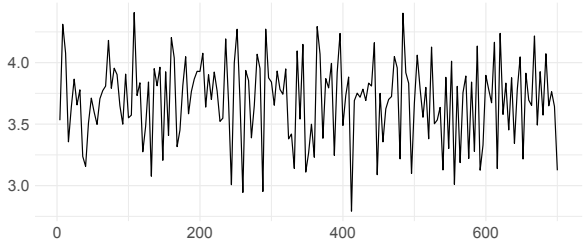

Fig. 3: Sample graph of the S2 synthetic case.

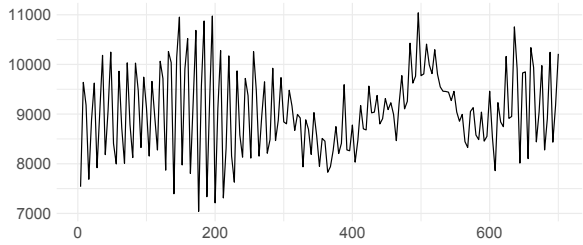

Fig. 4: Sample graph of the S3 synthetic case.

Existing empirical evidence suggests that cross-validation methods provide more accurate estimations than traditionally used OOS approaches in stationary time series forecasting [2]–[4] (see section II). 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.

The study was performed using the *R* language, and specifically, its performanceEstimation framework [16].

A. Experimental Setup

We focus on a purely auto-regressive modelling approach, predicting future values of time series using its past lags/observations. To accomplish this we follow the ideas regarding time-delay embedding [17]. In this context, a time series is reconstructed into a higher dimensional space with embedding dimension h. Effectively, we generate the following matrix:

$$Y_{[n,h]} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{h-1} & y_h \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i-h+1} & y_{i-h+2} & \cdots & y_{i-1} & y_i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-h+1} & y_{n-h+2} & \cdots & y_{n-1} & y_n \end{bmatrix} \quad (1)$$

Each row denotes an embedding vector $v_r, \forall\, r \in \{1, \ldots, t-h+1\}$. We then use the standard regression toolbox to solve the prediction task $y_{t+1} = f(v_r)$. Essentially we assume that

there are no long term time dependencies in the series and thus the embedding vectors are deemed as essentially uncorrelated.

We estimate the optimal embedding dimension (h) for the real-world scenario using the method of False Nearest Neighbours [18]. This method analyses the behaviour of the nearest neighbours as we increase h. According to the authors of the method [18], with a low sub-optimal h, many of the nearest neighbours will be false. Then, as we increase h and approach an optimal embedding dimension those false neighbours disappear. The embedding dimension in the synthetic case study is fixed to 5 [2].

The estimation set (Y^E) in each time series is the first 70% observations of the time series – see Figure 1. The validation period is comprised of the subsequent 30% observations (Y^V).

1) Estimation methods: In the experiments we apply a total of 10 performance estimation methods, which are divided into cross-validation (CV) variants and out-of-sample (OOS) aproaches. The cross-validation methods are the following:

CV.KF Standard K-fold cross-validation.

CV.BKF Blocked K-fold cross-validation.

CV.MKF Modified K-fold cross-validation.

CV.hvBKF hv-Blocked K-fold cross-validation.

The number of folds K in these methods is 10, which is a commonly used setting in the literature. The number of observations removed in CV.MKF and CV.hvBKF (c.f. section II) is the embedding dimension h in each time series.

The out-of-sample approaches are the following:

OOS.H Holdout: the first 70% of Y^E is used for training and the subsequent 30% is used for testing.

OOS.PB Prequential evaluation in blocks. The number of blocks is set to 10.

OOS.MC60 OOS tested in *nreps* testing periods with a Monte Carlo simulation using 60% of the total observations n of the time series in each test. For each period, a random point is picked from the time series. The previous window comprising 42% of n is used for training and the following window of 18% of n is used for testing. In this case *nreps* is set to 10.

OOS.MC20 Similar to OOS.MC60 but with a training window of 14%, a testing window of 6% and *nreps* set to 20. We used two different settings of multiple testing in OOS to test for robustness and check how the training/test window affects the results.

OOS.GW and OOS.SW As baselines we also include the exhaustive OOS alternatives in which an observation is first used to test the predictive model and then to train it. We use both a growing/landmark window (OOS.GW) and a sliding window (OOS.SW).

For a complete description of each method we refer to section II. Table I summarizes the estimation methods used as well as their specs.

2) Evaluation metrics: Our goal is to analyse which estimation method provides an $\hat{E}$ that best approximates E.

TABLE I: Summary of performance estimation procedures used in the experiments.

ID	Description
$CV.KF$	K Fold Cross-Validation
$CV.BKF$	Blocked K Fold Cross-Validation
$CV.MKF$	Modified K Fold Cross-Validation
$CV.hvBFK$	hv Blocked K Fold Cross-Validation
$OOS.H$	OOS Holdout
$OOS.PB$	Prequential in K Blocks
$OOS.MC60$	OOS with Monte Carlo Simulation using 60% of Y
$OOS.MC20$	OOS with Monte Carlo Simulation using 20% of Y
$OOS.GW$	OOS Growing Window
$OOS.SW$	OOS Sliding Window

Specifically, let E_f^m denote the estimated loss by the learning model m using the estimation method f on the estimation set, and E^m denote the ground truth loss of learning model m on the validation set. The objective is to analyse how well E_f^m approximates E^m. This is quantified by the predictive accuracy error (PAE) metric [2]:

$$PAE = \hat{E}_f^m - E^m \tag{2}$$

Across each case study, a given performance estimation method is evaluated in two dimensions according to PAE: 1) **error size**, by taking the absolute value of PAE – this measures the magnitude of the difference between the estimated and the actual error; and 2) **error bias**, by measuring the median PAE value across experiments. The error bias shows if a given method is under-estimating or over-estimating the 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 two distinct metrics: root mean squared error (RMSE) and mean absolute error (MAE). These are traditionally used for measuring the differences between the estimated and actual values. We include the two metrics in our experiments in the interest of robustness and because they have been used in previous studies [2]. RMSE and MAE are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

3) Learning models: To provide robustness to our experiments we apply the following three learning algorithms:

NN: A feed forward neural network with a single hidden layer [19];

DT: A CART regression tree [20];

LM: A linear model with a Ridge regularization [21].

The meta-parameters of each model are optimised using a grid search with a nested estimation procedure within each predictive performance estimation procedure.

TABLE II: Datasets and respective summary from the real-world case study.

ID	Time series	Data source	Data characteristics
1 2 3	Rotunda AEP Preciosa Mar Ameal	Porto Water Consumption from different locations in the city of Porto [22]	Half-hourly values from Nov. 11, 2015 to Jan. 11, 2016 (2929 values)
4 5 6 7	Global Horizontal Radiation Direct Normal Radiation Diffuse Horizontal Radiation Average Wind Speed	Solar Radiation Monitoring [23], [24]	Hourly values from Apr. 25, 2016 to Aug. 25, 2016 (2950 values)
8 9 10 11	Temperature Humidity Windspeed Total bike rentals	Bike Sharing [25]	Daily values from Jan. 1, 2011 to Dec. 31, 2012 (731 values)
12 13 14	Humidity Windspeed Total bike rentals		Hourly values from Jan. 1, 2011 to Mar. 01, 2011 (1338 values)
15 16 17 18 19 20 21 22 23 24	AeroStock1 AeroStock2 AeroStock3 AeroStock4 AeroStock5 AeroStock6 AeroStock7 AeroStock8 AeroStock9 AeroStock10	Stock price values from different aerospace companies [26]	Daily stock prices from January 1988 through October 1991 (949 values)
25 26 27 28 29 30 31 32 33 34 35 36 37	CO.GT PT08.S1.CO NMHC.GT C6H6.GT PT08.S2.NMHC NOx.GT PT08.S3.NOx NO2.GT PT08.S4.NO2 PT08.S5.O3 Temperature RH Humidity	Air quality indicators in an Italian city [27]	Hourly values from Mar. 10, 2004 to Apr. 04 2005 (9357 values)
38 39 40 41	Electricity Total Load Equipment Load Gas Energy Gas Heat Energy	Hospital Energy Loads [28]	Hourly values from Jan. 1, 2016 to Mar. 25, 2016 (2000 values)
42 43	Total Demand Recommended Retail Price	Australian Electricity [29]	Half-hourly values from Jan. 1, 1999 to Mar. 1, 1999 (2833 values)
44 45 46 47 48 49 50	SP DAX FTSE NIKKEI BOVESPA EU EM	Returns at Istanbul Stock Exchange with seven other international indices [30]	Daily values from Jan. 5, 2009 to Feb. 22, 2011 (536 values)
51	Flow of Vatnsdalsa river	Icelandic river [31]	Daily values from Jan. 1, 1972 to Dec. 31, 1974 (1095 values)
52 53	Min. temperature Max. temperature	Porto weather [32]	Daily values from Jan. 1, 2010 to Dec. 28, 2013 (1457 values)

B. Results

1) Predictive accuracy error size: The results regarding the error size of the performance estimation methods are presented in Figures 5 to 10. These follow the guidelines of Demšar [33] regarding statistical comparison of methods over multiple data sets. In this analysis we use the absolute value of the PAE measure (c.f. Section V-A2) of each estimation method across the experiments. Then the Friedman test is applied to obtain the average ranking of methods. To check for statistical differences of the average rankings we use a significance level of 0.05.

Each diagram shows the ranking of the methods according to the Friedman test. A lower rank represents better performance. The horizontal lines connecting the methods show the significance of the difference among ranks. Pairs of models not connected with a horizontal line indicate significant difference in their ranks for a given experiment.

Figures 5 to 7 denote the results of the analysis in the synthetic case studies S1, S2 and S3, respectively. In this scenario we fixed the learning models to the neural network in the interest of conciseness. Similar conclusions are drawn using the other two learning models.

The results show significant differences among distinct methods, particularly in the S1 and S2 cases. Overall, the cross-validation approaches seem to present a comparable estimation ability relative to the out-of-sample approaches. However, the results are not as evident as reported in previous work by Bergmeir et al. [2]. We argue that this is mainly due to the differences in the size of the time series. They use synthetic time series with a length of 200 observations whereas we increase the data to 700 values. Consequently, in small-sized data sets the benefits of cross-validation approaches are more apparent due to their better use of data relative to out-of-sample approaches. We increased the size of the time series to provide more fair comparisons.

In the synthetic case the results do not alter significantly for distinct evaluation metrics. However, some differences can be visible between case studies. For example, in Figure 7, the CV.MKF method is the most accurate estimator. On the other hand, it presents one of the worst rankings in the other two synthetic case studies.

Figures 8 to 10 illustrate the results for each learning model applied to the real-world case study. These show different results relative to the synthetic scenario.

Although the differences are not significant, the out-of-sample approaches show systematically better ranks than cross-validation approaches. These suggest that in a real-world setting it is beneficial to keep the temporal order of observations.

In the synthetic case studies, OOS.H is significantly better than OOS applied in multiple testing periods, OOS.MC60 or OOS.MC20. In the real-world scenario the inverse is observed, though with no statistical significance. In this case, OOS.MC60 or OOS.MC20 consistently show better ranking than OOS.H. This suggests that in real-world time series with potential non-stationarities it is benefitial to test in multiple periods using a randomized strategy [5]. Comparing the two OOS strategies that use multiple testing periods, in most cases OOS.MC60 shows a better rank in the real-world data relative to OOS.MC20.

The exhaustive approaches, OOS.GW and OOS.SW, are computationally expensive and do not seem to provide better estimates in comparison to the other, cheaper approaches.

The results on the real-world case are robust across the two evaluation metrics. Moreover, some differences can be captured by using different learning models but these do not seem to alter the results significantly.

2) Median predictive accuracy error: Table III presents the results of the median PAE (predictive accuracy error – c.f. Section V-A2). Ideally, an estimation procedure should approach the median PAE of 0. A positive value indicates an over-estimation of error ($\hat{E} > E$), while a negative one represents under-estimation of error or over-fitting ($\hat{E} < E$). We use the median instead of a mean value of PAE to control for outlier values.

The numbers in the table suggest that the out-of-sample approaches provide the most accurate estimates, on average and across the case studies. This holds for both evaluation metrics and the three learning models used.

VI. Discussion

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 53 time series from different domains and with unknown dynamics. 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 [2]–[4].

Bergmeir et al. show in previous studies [3], [34] that the blocked form of cross-validation, denoted here as CV.BKF, yields more accurate estimates than an out-of-sample evaluation (OOS.H) for stationary time series forecasting tasks. The CV.KF is also suggested to be "a better choice than OOS evaluation" as long as the data are well fitted by the model [2].

To some extent part of the results from our experiments corroborate these conclusions. Specifically, this is verified by the predictive accuracy error size incurred by the estimation procedures in the synthetic case studies.

However, according to our experiments, the results from the synthetic stationary case studies poorly reflect the results on real-world data. In a realistic setting the results suggest that estimation methods that preserve the temporal order of observations provide more accurate error estimates. This is shown in the experimental evaluation both in terms of error size as well as error bias. In particular, OOS.MC60 shows a noticeable high ranking in the real-world case study across all the problems. Cross-validation approaches, especially standard K-fold cross-validation, systematically provide worse estimates than the out-of-sample methods.

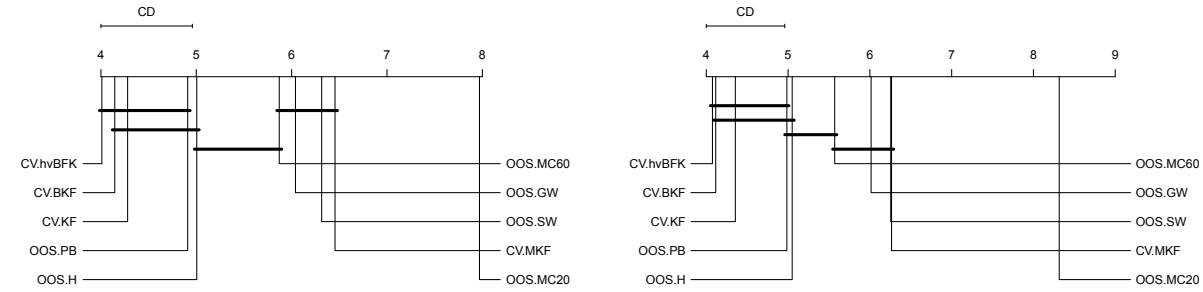

Fig. 5: Friedman test on the **S1 synthetic data** and the NN learning model. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

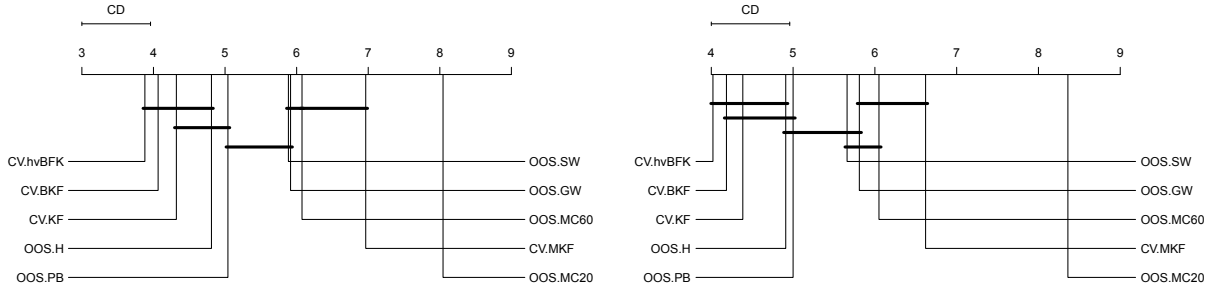

Fig. 6: Friedman test on the **S2 synthetic data** and the NN learning model. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

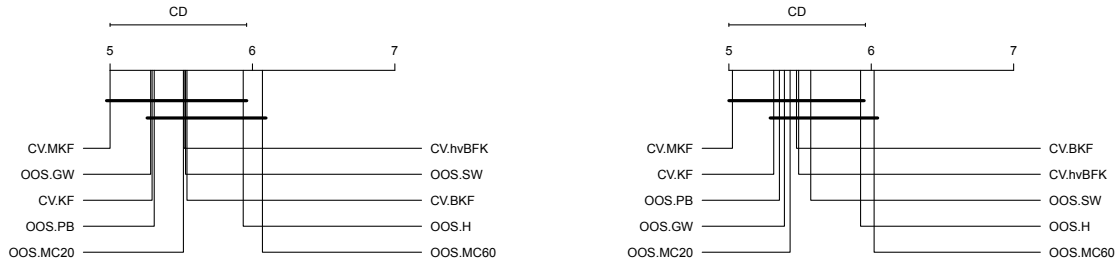

Fig. 7: Friedman test on the **S3 synthetic data** and the NN learning model. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

In a real-world environment we are prone to deal with time series with complex structures, such as long-range dependence, pink noise or fractional integration. These comprise nuances of the future that may not have revealed themselves in the past [5]. Consequently, and following the results of our experiments, in these scenarios we conclude that using the out-of-sample approaches yields more reliable performance estimates than cross-validation approaches.

VII. Final Remarks

In this paper we analyse the ability of different methods to approximate the loss that a given learning algorithm will incur on unseen data. This error estimation process is performed in every machine learning task for model selection and meta-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 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 be able to handle the dependency among observations. However, under the realistic scenario, they systematically provide worse estimations than the out-of-sample approaches.

Bergmeir et al. [2]–[4] suggest that for stationary time series one should use cross-validation in a blocked form. On the

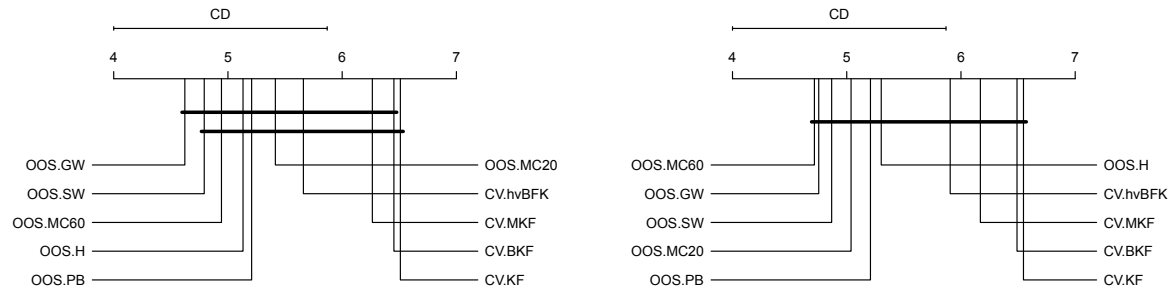

Fig. 8: Friedman test on the real-world time series datasets and the **NN learning model**. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

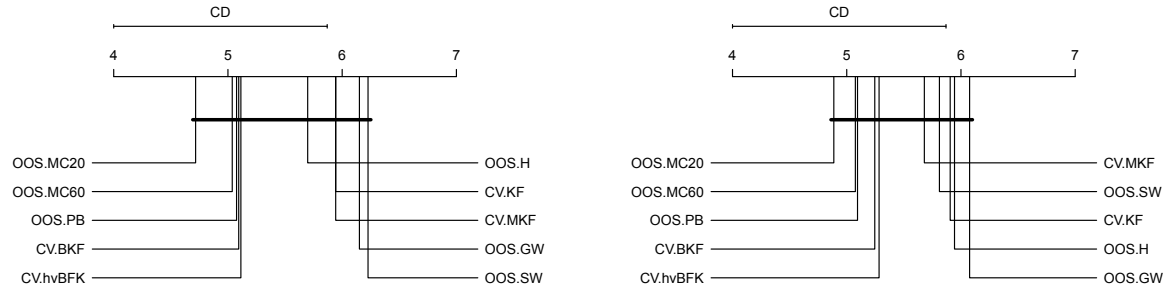

Fig. 9: Friedman test on the real-world time series datasets and the **DT learning model**. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

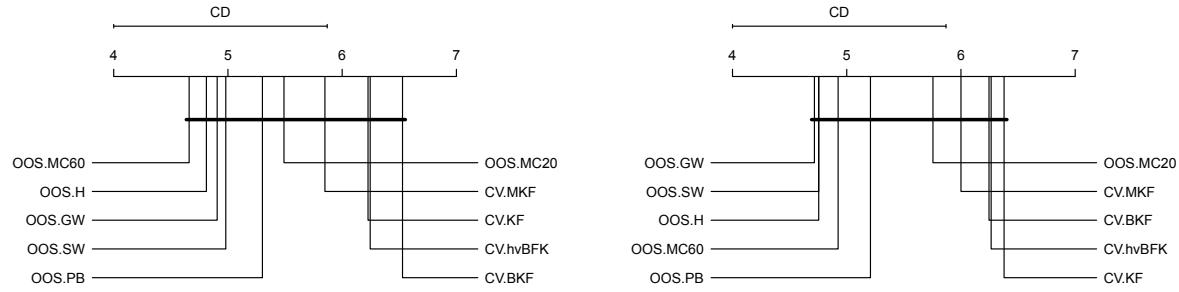

Fig. 10: Friedman test on the real-world time series datasets and the **LM learning model**. Critical difference diagrams show ranking of the performance estimation procedures by the RMSE metric (left) and by the MAE metric (right).

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 applied in multiple testing periods results in remarkably high estimates, relative to the other tested alternatives.

In the interest of reproducibility, the methods and datasets are publicly available at https://github.com/vcerqueira/performance_estimation.

ACKNOWLEDGMENT

Vitor Cerqueira and Luís Torgo acknowledge financing by Project "NORTE-01-0145-FEDER-000036", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). Jasmina Smailović and Igor Mozetič acknowledge financial support from the H2020 FET project DOLFINS (grant no. 640772), and the Slovenian Research Agency (research core funding no. P2-0103).

REFERENCES

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

[2] C. Bergmeir, R. J. Hyndman, B. Koo *et al.*, "A note on the validity of cross-validation for evaluating time series prediction," *Monash University Dept of Econometrics and Business Statistics Working Paper*, 2015.

[3] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.

TABLE III: Median predictive accuracy error (PAE) of the different performance estimation methods with distinct evaluation metrics, data sets and learning models. A method with the most accurate result for each problem is highlighted in bold.

		Data	**CV.KF**	**CV.BKF**	**CV.MKF**	**CV.hvBKF**	**OOS.H**	**OOS.PB**	**OOS.MC60**	**OOS.MC20**	**OOS.GW**	**OOS.SW**
NN	RMSE	Real	-0.016	-0.016	0.006	-0.016	0.009	-0.084	**0.000**	0.001	0.005	0.005
		S1	**0.000**	-0.002	0.077	**0.000**	0.009	0.040	0.064	0.177	0.018	0.039
		S2	-0.016	-0.015	0.161	-0.009	**0.005**	0.053	0.090	0.276	0.037	0.058
		S3	-76.997	-93.982	-42.502	-60.624	-37.584	-35.956	-111.506	-92.722	-56.942	**-31.962**
	MAE	Real	-0.030	-0.029	0.005	-0.011	0.004	-0.019	**0.001**	**0.001**	0.005	0.005
		S1	0.002	0.002	0.055	**0.001**	0.002	0.030	0.035	0.123	0.016	0.018
		S2	-0.011	-0.010	0.088	-0.008	**0.005**	0.041	0.059	0.201	0.012	0.018
		S3	-72.115	-59.102	-42.229	-24.513	-39.985	**-16.915**	-100.757	-77.970	-30.724	-27.612
DT	RMSE	Real	-0.126	-0.015	0.007	-0.010	0.006	**0.005**	0.009	0.008	0.009	0.007
		S1	-0.012	-0.009	0.024	-0.008	0.005	0.022	0.017	0.073	-0.008	**-0.004**
		S2	-0.015	-0.016	0.008	-0.015	-0.011	**0.007**	0.015	0.067	-0.022	-0.022
		S3	-78.501	-59.134	-26.714	-49.290	17.197	-20.782	11.758	**-0.409**	-1.352	-4.435
	MAE	Real	-0.042	-0.002	0.008	0.004	0.017	0.005	**0.004**	0.016	0.009	0.006
		S1	-0.003	-0.004	0.022	**-0.002**	0.007	0.021	0.012	0.065	-0.003	0.005
		S2	-0.012	-0.010	0.009	-0.010	-0.010	**0.003**	0.006	0.057	-0.014	-0.016
		S3	-62.379	-45.430	-21.725	-39.645	12.115	-7.688	15.813	0.751	**0.585**	5.571
LM	RMSE	Real	-0.034	-0.040	-0.011	-0.045	-0.006	-0.043	-0.020	-0.014	**0.004**	0.005
		S1	-0.006	-0.006	**0.000**	-0.006	-0.004	0.002	0.002	0.013	-0.008	-0.009
		S2	-0.012	-0.011	**-0.001**	-0.011	-0.005	**-0.001**	0.009	0.008	-0.015	-0.016
		S3	-41.939	-37.042	-28.962	-34.269	**-1.126**	-14.314	2.735	-18.777	-17.145	-14.369
	MAE	Real	-0.014	-0.018	0.003	-0.021	**-0.001**	-0.007	-0.028	-0.019	0.004	0.005
		S1	0.001	0.001	0.006	0.001	**0.000**	0.005	0.003	0.018	0.002	**0.000**
		S2	-0.007	-0.009	0.004	-0.009	-0.004	**0.000**	0.004	0.017	-0.009	-0.010
		S3	-31.583	-21.759	-26.131	-19.745	-1.081	-9.812	**-0.560**	-7.397	-3.840	-7.146

[4] C. Bergmeir, M. Costantini, and J. M. Benítez, "On the usefulness of cross-validation for directional forecast evaluation," *Computational Statistics & Data Analysis*, vol. 76, pp. 132–143, 2014.

[5] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International journal of forecasting*, vol. 16, no. 4, pp. 437–450, 2000.

[6] R. Fildes, "Evaluation of aggregate and individual forecast method selection rules," *Management Science*, vol. 35, no. 9, pp. 1056–1065, 1989.

[7] L. Torgo, *Data mining with R: learning with case studies*. Chapman & Hall/CRC Boca Raton, FL:, 2011.

[8] D. S. Modha and E. Masry, "Prequential and cross-validated regression estimation," *Machine Learning*, vol. 33, no. 1, pp. 5–39, 1998.

[9] A. Bifet and R. Kirkby, "Data stream mining a practical approach," 2009.

[10] J. D. Hart and T. E. Wehrly, "Kernel regression estimation using repeated measurements data," *Journal of the American Statistical Association*, vol. 81, no. 396, pp. 1080–1088, 1986.

[11] T. A. Snijders, "On cross-validation for predictor evaluation in time series," in *On Model Uncertainty and its Statistical Implications*. Springer, 1988, pp. 56–69.

[12] J. Racine, "Consistent cross-validatory model-selection for dependent data: hv-block cross-validation," *Journal of econometrics*, vol. 99, no. 1, pp. 39–61, 2000.

[13] A. D. McQuarrie and C.-L. Tsai, *Regression and time series model selection*. World Scientific, 1998.

[14] M. Stone, "Cross-validation and multinomial prediction," *Biometrika*, pp. 509–515, 1974.

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

[16] L. Torgo, *An Infra-Structure for Performance Estimation and Experimental Comparison of Predictive Models*, 2013, R package version 0.1.1.

[17] F. Takens, *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, ch. Detecting strange attractors in turbulence, pp. 366–381.

[18] M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical review A*, vol. 45, no. 6, p. 3403, 1992.

[19] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002, iSBN 0-387-95457-0.

[20] T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive Partitioning and Regression Trees*, 2015, R package version 4.1-10.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[22] "Águas de douro e paiva," https://www.addp.pt/home.php, accessed: 2017-01-30.

[23] S. Andreas, A.; Wilcox, "Solar radiation monitoring station (sorms): Humboldt state university, arcata, california (data); nrel report no. da-5500-56515." http://dx.doi.org/10.5439/1052559, 2007.

[24] V. Cerqueira, L. Torgo, and C. Soares, "Arbitrated ensemble for solar radiation forecasting," in *International Work-Conference on Artificial Neural Networks*. Springer, Cham, 2017, pp. 720–732.

[25] H. Fanaee-T and J. Gama, "Event labeling combining ensemble detectors and background knowledge," *Progress in Artificial Intelligence*, pp. 1–15, 2013.

[26] P. Vlachos, "Statlib project repository," *Carnegie Mellon University*, 2000.

[27] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[28] EERE, "Commercial and residential hourly load profiles for all tmy3 locations in the united states," http://en.openei.org/datasets/files/961/pub/, accessed: 2016-11-21.

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

[30] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid rbf neural networks model as a forecaster," *Statistics and Computing*, vol. 24, no. 3, pp. 365–375, 2013.

[31] H. Tong, B. Thanoon, and G. Gudmundsson, "Threshold time series modeling of two icelandic riverflow systems1," *JAWRA Journal of the American Water Resources Association*, vol. 21, no. 4, pp. 651–662, 1985.

[32] "freemeteo: Weather forecasts," http://freemeteo.com.pt, accessed: 2017-01-30.

[33] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

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