Neural Processing Letters (2023) 55:10073–10091
https://doi.org/10.1007/s11063-023-11239-8

Model Selection for Time Series Forecasting An Empirical Analysis of Multiple Estimators

Vitor Cerqueira[1] · **Luis Torgo**[1] · **Carlos Soares**[2,3,4]

Accepted: 8 March 2023 / Published online: 22 March 2023

Abstract
Evaluating predictive models is a crucial task in predictive analytics. This process is especially challenging with time series data because observations are not independent. Several studies have analyzed how different performance estimation methods compare with each other for approximating the true loss incurred by a given forecasting model. However, these studies do not address how the estimators behave for model selection: the ability to select the best solution among a set of alternatives. This paper addresses this issue. The goal of this work is to compare a set of estimation methods for model selection in time series forecasting tasks. This objective is split into two main questions: (i) analyze how often a given estimation method selects the best possible model; and (ii) analyze what is the performance loss when the best model is not selected. Experiments were carried out using a case study that contains 3111 time series. The accuracy of the estimators for selecting the best solution is low, despite being significantly better than random selection. Moreover, the overall forecasting performance loss associated with the model selection process ranges from 0.28 to 0.58%. Yet, no considerable differences between different approaches were found. Besides, the sample size of the time series is an important factor in the relative performance of the estimators.

Keywords Model selection · Performance estimation · Cross-validation · Time series · Forecasting

✉ Vitor Cerqueira
vitor.cerqueira@dal.ca

Luis Torgo
ltorgo@dal.ca

Carlos Soares
csoares@fe.up.pt

1 Faculty of Computer Science, Dalhousie University, Halifax, Canada

2 Fraunhofer AICOS Portugal, Porto, Portugal

3 INESC TEC, Porto, Portugal

4 University of Porto, Porto, Portugal

1 Introduction

Estimating the predictive performance of models is a crucial stage in the data science pipeline.

Performance estimation methods are used to solve two main problems:

- To provide a reliable estimate of thef performance of a given predictive model. These estimates inform the end-user of the expected generalization ability of that model;
- To use these estimates to perform model selection, i.e., to select a predictive model among a set of possible alternatives. The alternatives may be different learning algorithms or different parameter settings of the same learning algorithm.

Despite their similarity, model selection and performance estimation are two different problems [1, 2]. On the same data set, an estimator may provide the best loss estimations, on average, but not the best model ranking for selection purposes. This idea is illustrated in Fig. 1 . In this example, there are four predictive models: A1, A2, A3, and A4. These are shown in the x-axis of Fig. 1. The true test set loss is depicted by the left-hand side bars (light-blue). Thus, the correct ranking of the models is A1 > A2 > A3 > A4. A1 is the best model as it shows the lowest test loss. Two estimators, E1 (middle bars) and E2 (right-hand side bars), are used to approximate the error of each model. The estimator E1 produces the best loss approximations (nearest to the true error) relative to estimator E2, on average. However, its estimated ranking (A2 > A1 > A4 > A3) is different from the true ranking and worse than the one produced by E2. Despite providing worse average performance estimates, E2 outputs a perfect ranking of the models. This example shows that one estimator is better for performance estimation (E1), and the other for model selection (E2).

This work addresses this problem with a particular focus on time series forecasting tasks. The time dependency among observations poses a challenge to several estimation methods that assume observations are independent.

Several previous papers have addressed the performance estimation problem for forecasting tasks (e.g. [3–7]). Yet, they do not analyze how different estimation methods behave for model selection. This work addresses this limitation. This matter was explored before by

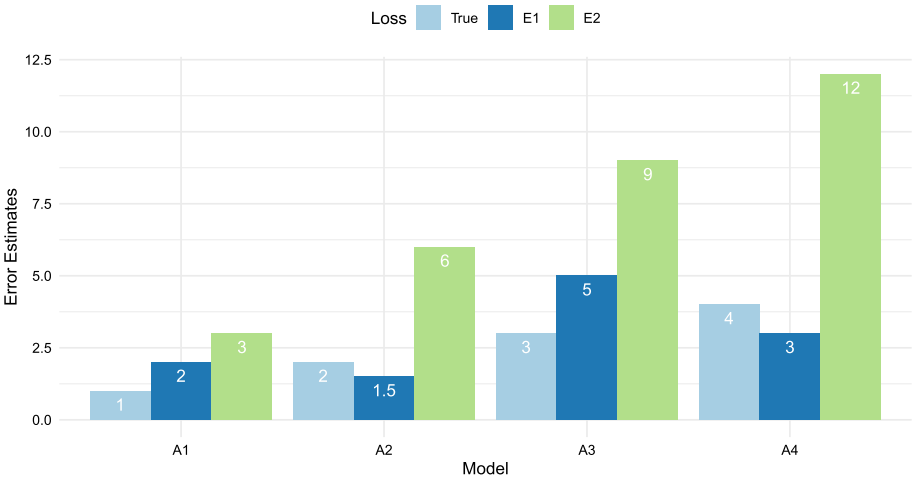

Fig. 1 True error (labeled *True*) and two error estimates (using estimators E1 and E2) of the predictive performance of four models: A1, A2, A3, and A4. The estimator E1 shows the best estimates, on average. However, E2 produces a perfect ranking of the models

Breiman and Spector [1] for i.i.d. data sets. They found interesting differences in the relative performance of estimation methods when applied to model selection and performance estimation.

The goal of this work is to study different estimators (e.g. K-fold cross-validation, Holdout) for model selection in time series forecasting tasks, in which the observations are not i.i.d.. Given a pool of alternative models, this work studies: (i) how often the best solution is picked (the one which maximizes forecasting performance on test data); and (ii) how much performance is lost when it does not. In other words, the goal is to analyze how different estimators rank the available predictive models by their performance in unseen observations. This work particularly emphasizes the top-ranked model, which is the one most probably selected for deployment. That is, for predicting future observations of the domain under study.

Most estimation procedures involve repeating the application of a model to different data folds. The estimation results across these folds are typically combined using the arithmetic mean. Accordingly, the selected model is the one with the lowest average estimated error. This work also studies the possibility of combining the results across folds using the average rank. The average rank method is a non-parametric approach. This aspect is beneficial to smooth the effect of large errors (outliers) in particular folds [8].

A set of experiments were carried out using 10 estimation methods, 3111 data sets, and 50 forecasting models. The results show that the accuracy of the estimators for selecting the best model ranges from 7 to 10%. Overall, the forecasting performance loss incurred due to incorrect model selection ranges from 0.28 to 0.58%. The experiments were controlled by the sample size of the time series. As expected, the performance loss is considerably larger for smaller time series. Finally, regarding the strategy for combining the results across folds, the average rank leads to a comparable performance relative to the average error.

In summary, the contribution of this paper is an extensive study comparing a set of performance estimation methods for model selection in time series forecasting tasks. To our knowledge, this paper is the first to quantify the impact of using a particular estimator for model selection in forecasting problems. The data and experiments carried out are publicly available online.[1]

This paper is organized as follows. In the next section, the literature related to this work is reviewed. This includes state-of-the-art methods for performance estimation and machine learning approaches for forecasting problems. Section 3 formalizes the tasks concerning time series forecasting and model selection. The experiments are presented in Sect. 4 and discussed in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related Work

This section provides an overview of the literature related to this work. It briefly describes performance estimation methods designed for time series forecasting models (Sect. 2.1). Previous papers are listed, and the contributions of this paper are highlighted. Section 2.2 overviews the literature concerning the application of machine learning methods for forecasting.

[1] https://github.com/vcerqueira/model_selection_forecasting.

2.1 Performance Estimation in Forecasting

2.1.1 Methods

Several approaches have been proposed for evaluating the predictive performance of models in time series. According to Arlot and Celisse [2], *cross-validation* denotes the process of splitting the data, once or multiple times, for estimating the performance of a predictive model. In this process, part of the data is used to fit a model while the remaining observations are used for testing. All methods analyzed in this work follow this procedure. However, this paper refers to cross-validation as the class of approaches which follow the splitting method of K-fold cross-validation and which assume independence among data points. Other procedures, such as prequential [9], work under different assumptions. For example, prequential assumes the observations are ordered and arrive sequentially. Therefore, these are designated accordingly.

Arguably, the most common approach to assess the performance of a forecasting model is the *holdout* procedure, also known as *out-of-sample* evaluation (e.g. [4]). The initial part of the time series is used to fit the model, and the latest part of the data is used to test it. Tashman [6] recommends applying this approach in multiple testing periods. Indeed, Cerqueira et al. [5] show that the holdout applied in multiple, randomized, periods leads to the best estimation ability relative to several other state-of-the-art estimators for non-stationary time series.

K-fold cross-validation works by randomly assigning the available observations to K different but equally-sized folds. Each fold is then iteratively used for testing a predictive model. This model is built using all remaining observations. This process is theoretically inadequate to evaluate time series models because the observations are not independent [2, 10]. However, Bergmeir et al. [3, 4] show that cross-validation approaches can be successfully applied to estimate the performance of forecasting models. For example, they showed that, in some scenarios, the K-fold cross-validation procedure described above provides better results than out-of-sample evaluation [4]. Notwithstanding, several methods have been proposed as extensions to the K-fold cross-validation. These aim at mitigating the problem of independence among observations assumed by K-fold cross-validation, which does not hold in time series. Some of these extensions include blocked cross-validation [11], modified cross-validation [12], and *hv*-blocked cross-validation [13]. Bergmeir et al. [3] and Cerqueira et al. [5] compare different estimators for evaluating forecasting models. They suggest using the blocked form of K-fold cross-validation for stationary time series.

The *prequential* method is also a common evaluation approach for time-dependent data [9]. This method is also referred to as *time series cross-validation* by practitioners. Prequential denotes the process in which an observation (or batch of observations) is first used for testing a predictive model and then updating it. Prequential is the most common solution in data stream mining tasks [14]. For more general time series, prequential approaches are typically applied in contiguous blocks of data. Moreover, prequential can be applied in different manners. For example, using a growing window or a sliding window. In a growing window, the latest observations are used to expand the data set. On the other hand, in a sliding window approach, older observations are discarded as new ones become available thereby keeping the training and testing sizes constant.

The above-mentioned estimation methods are described in more detail in Sect. 4.4.

2.1.2 Empirical Comparisons

The methods described above have been studied in different studies according to how well they approximate the loss a predictive model incurs in a test set [3–5]. The objective was to assess: (i) the magnitude of their estimation errors, and (ii) the direction of the error, i.e., whether the estimators under-estimate or over-estimate the loss of the respective model. These two quantities allow us to analyze which estimators provide the most reliable approximations, on average. They enable the quantification of the generalization ability of models, which helps the end-user decide whether or not a model can be deployed. However, as illustrated in Fig. 1, the estimator with the best approximations is not necessarily the most appropriate for model selection [1]. Correctly identifying the relative performance of predictive models is an important feature for model selection.

2.2 Machine Learning Methods for Forecasting

Without loss of generality, this work focuses on typical machine learning regression algorithms for forecasting. These are applied in an auto-regressive manner using time-delay embedding, which is formalized in Sect. 3. This section lists several papers which address the applicability of machine learning approaches to forecasting.

Makridakis et al. [15] reported that machine learning approaches performed worse relative to traditional methods such as ARIMA [16], exponential smoothing [17], or a simple seasonal random walk. These conclusions were drawn from 1045 monthly time series with a low sample size (an average of 118 observations). However, in a more recent work, Makridakis and his colleagues [18] show that machine learning approaches provide better forecasting performance for SKU demand forecasting when compared to these traditional methods. Moreover, Cerqueira et al. [19] show that the conclusions drawn in [15] are only valid for small time series. In the M5 forecasting competition [20], the `lightgbm` gradient boosting method [21], which is a popular machine learning algorithm, was the method used in the winning solution and some of the runner-ups.[2]

Other studies have also shown that standard regression methods can be successfully applied to time series forecasting problems. Cerqueira et al. [22, 23] develop a dynamic ensemble for forecasting. The ensemble is heterogeneous and comprised of several machine learning methods, along with other traditional forecasting approaches. Corani et al. [24] propose a method based on Gaussian processes for automatic forecasting. Automatic forecasting is the process of creating forecasting models with minimal input from the user.

Deep neural networks are increasingly applied to this type of problem and several architectures have been developed. Examples include N-BEATS [25] or DeepAR [26], among others. A notable work is that by Smyl [27], which developed a hybrid method that combines a recurrent neural network with exponential smoothing. Finally, the temporal fusion transformer is an adaptation of the transformer architecture for forecasting problems [28]. LSTM (long short-term memory) is a special type of recurrent neural network that has been applied in many application domains. Examples include wind speed forecasting [29] or traffic flow prediction [30].

Prophet [31] is also a popular forecasting approach, which is based on an additive model. NeuralProphet [32] extends this method by leveraging deep learning.

[2] https://github.com/Mcompetitions/M5-methods.

Table 1 Summary of the learning algorithms

ID	Algorithm	Parameter	Value
SVR	Support Vector Regr.	Kernel	{Linear, RBF Polynomial, Laplace}
		Cost	{1, 5}
		ϵ	{0.1, 0.01}
MARS	Multivar. A. R. Splines	Degree	{1, 3}
		No. terms	{5, 10, 20}
		Forward thresh	{0.001}
RF	Random forest	No. trees	{250, 500}
		Mtry	{5, 10}
PPR	Proj. pursuit regr.	No. terms	{2, 4}
		Method	{super smoother, spline}
RBR	Rule-based regr	No. iterations	{1, 5, 10, 25}
MLP	Multi-layer Perceptron	Units Hid. Lay. 1	{10, 15}
		Units Hid. Lay. 2	{0, 5}
GLM	Generalised Linear Regr	Penalty mixing	{0, 0.25, 0.5, 0.75, 1}
GP	Gaussian Processes	Kernel	{Linear, RBF Polynomial, Laplace}
		Tolerance	{0.001}
PCR	Principal Comp. Regr	*Default*	–
PLS	Partial Least Regr	Method	{kernel, SIMPLS}
XGB	Gradient Boosting	*Auto*[a]	–

[a] Automatically optimized using a grid search based on the *tsensembler* R package

Global forecasting models have been increasingly used for forecasting [33, 34]. The idea behind these approaches is to pool the historical observations of multiple time series to build a forecasting model.

This paper focuses on several standard regression algorithms as the pool of alternative forecasting models. These are detailed in Table 1. Notwithstanding, the comparisons carried out could be applied to any forecasting model.

3 Problem Definition

This section defines two tasks. First, the time series forecasting problem is formalized from an auto-regressive perspective (Sect. 3.1). Then, the model selection problem is defined (Sect. 3.2). Finally, Sect. 3.3 overviews the average rank method for algorithm selection across multiple data sets and how it is applied for model selection within a single data set.

3.1 Auto-Regression

Let Y denote a time series $Y = \{y_1, y_2, \ldots, y_n\}$, in which y_i is the i-th out of n time-ordered observations. The goal is to predict the future values of such time series. Without loss of

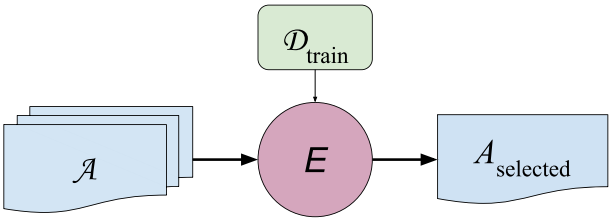

Fig. 2 Workflow for model selection. The estimation method E tests different algorithms $\mathrm{A} \in \mathcal{A}$ using the training data. The algorithm with the best performance estimation is selected for predicting future observations

generality, this work focuses on one-step-ahead forecasting. Accordingly, at the i-th time step, a forecasting model aims at predicting the value of the next observation (y_{i+1}).

Without loss of generality, the predictive task can be formalized using time delay embedding by constructing a set of observations of the form (X, y). The target value y_i is modeled based on the past p values before it: $X_i = \{y_{i-1}, y_{i-2}, \ldots, y_{i-p}\}$. This process leads to a multiple regression problem where each $y_i \in \mathcal{Y} \subset \mathbb{R}$ represents the i-th observation we want to predict, and $X_i \in \mathcal{X} \subset \mathbb{R}^p$ represents the corresponding explanatory variables. Effectively, the time series is transformed into the data set $\mathcal{D}(X, y) = \{X_i, y_i\}_{p+1}^{n}$.

When addressing a time series forecasting problem as an auto-regressive task, one of the challenges is to determine the number of lags. One common approach is the number of lags p is usually estimated according to the False Nearest Neighbors method [35].

3.2 Model Selection

Model selection denotes the process of using the available training data to select a predictive model M among a set of m alternatives $\mathcal{A} = \{A_1, \ldots, A_m\}$. As depicted in Fig. 2, each alternative is evaluated using an estimation method E (e.g. K-fold cross-validation), and a training data set $\mathcal{D}_{train}$. The model which maximizes the predictive performance (or minimizes the loss) according to the estimator is selected and used in future observations (a test set).

The goal is to find and select $\mathrm{A}_* \in \mathcal{A}$, which is the model with the best performance on the test set. Therefore, the model selection problem can be formalized as follows:

$$\mathrm{A}_{\text{selected}} \in \underset{\mathrm{A} \in \mathcal{A}}{\operatorname{argmin}} \big(L(\mathrm{A}, \mathcal{D}_{train}, E) \big) \tag{1}$$

where L represents the loss metric that quantifies the expected error of a predictive model, and E is the estimation method applied to estimate such measure.

In Eq. 1, the function L takes as input the learning algorithm, the training data, and the estimation method. The output of L is the expected error of the respective learning algorithm. Then, model $\mathrm{A}_{\text{selected}}$ is selected, which is the one that minimizes the expected error. Ideally, $\mathrm{A}_{\text{selected}}$ represents A_*, which denotes the model with the best performance in a test set. However, this is not necessarily the case as E is only able to provide an approximation of the true error of any model.

The goal of this paper is to analyze a set of estimation methods $\mathcal{E} = \{E_1, E_2, \ldots, E_q\}$ according to their ability to find A_*, and their behavior when they do not. This can be regarded as an analysis of the ranking ability of the set of estimators, $\mathcal{E}$, in which one is particularly interested in the top-ranked model. Then, it is analyzed how much performance is lost when

these estimators do not select the best possible model. Section 4 presents a set of experiments that compare a set of estimators from this perspective.

3.3 Average Rank

The average rank is a procedure typically applied to carry out a statistical comparison of multiple learning algorithms over multiple data sets [36]. The rank of a predictive model denotes its position in terms of average performance on a set of problems relative to its competitors. A rank of 1 in a given data set means that the respective model was the best-performing one. Effectively, the average rank represents the average relative position of a given predictive model. As Benavoli et al. [37] explain, the average rank method is a non-parametric approach that does not assume the normality of the sample means and is robust to outliers. This approach is often used for algorithm selection [38].

Here, we hypothesize that the average rank may be a useful approach for model selection within a single data set. Following Eq. 1, the selected model is typically the one that minimizes the expected error. This expected error is estimated by averaging (with the arithmetic mean) the error of each model across several folds. However, the average error is amenable to outliers. For example, a model may incur a large error in a single fold which will significantly affect its average. On the other hand, the average rank approach may amplify small errors [36]. In this context, experiments were carried out to test whether the average rank is a better approach to combining the results of multiple models across multiple folds within the same problem.

Yang [8] explored this idea before in the context of regression tasks. The author refers to this process as voting cross-validation and concluded that it leads to a comparable performance relative to the standard error-based cross-validation. We investigate this issue in the context of time series forecasting problems.

4 Experiments

The experiments carried out in this paper aim at comparing different estimators for model selection in time series forecasting tasks. They are designed to address the following research questions:

1. **RQ1**: What is the selection accuracy of different performance estimation methods for model selection? That is, how often do their estimates lead to picking the best model (the one which maximizes the true predictive performance in new observations);
2. **RQ2**: What is the forecasting performance lost when performance estimation methods do not pick the best model?
3. **RQ3**: Do the experimental results vary when controlling for the sample size of the time series?
4. **RQ4**: How does the average rank (voting cross-validation) compare with the average error for aggregating the results across folds for model selection?
5. **RQ5**: What is the relative computational cost of each estimator?

Sect. 4.1 describes the time series data sets used in the experiments. The experimental design is explained in Sect. 4.2, which includes how each estimation method was applied. Then, Sect. 4.3 overview the evaluation process which includes three different metrics. Section 4.4 lists all learning algorithms and estimation methods used in the experiments. Finally, the results of the experiments are described in Sect. 4.5.

4.1 Data Sets

The experiments in this paper are carried out using 3111 real-world time series. All the daily time series with at least 500 observations from the M4 case study [39] were retrieved. The sample size constraint (at least 500 data points) is included as it is an important component for training machine learning models [19]. The query returned 2937 time series. These cover several domains of application, including demographics, finance, industry, and economics. The remaining 174 time series were retrieved from a previous related study [5]. 149 out of these 174 time series were retrieved from the benchmark database *tsdl* [40]. The remaining 25 time series were collected from the study by Cerqueira et al. [23]. All data sets are available for reproducing the experiments (c.f. footnote 1).

4.2 Experimental Design

A realistic scenario was used to compare the different performance estimation methods for model selection. First, the available time series was split into two parts: an estimation set, which contains the initial 70% of observations; and a test set, which contains the subsequent 30% of observations. The general goal is to select the model which provides the best performance on the test set. In practice, one cannot perform direct estimations on this set as it represents future observations. Therefore, the estimation set must be used to assess which is the best model.

For each estimator, $E_i \in \mathcal{E}$, the model selection process is carried out according to the workflow illustrated in Fig. 2 and defined in Eq. 1, in which the training data represents the estimation set.

This process results in a set of models $\{\mathrm{A}_{E_1}, \ldots, \mathrm{A}_{E_q}\}$, where A_{E_i} is the model selected by estimator E_i. Finally, each estimator is evaluated according to its model selection ability using the test set. The evaluation process is described in the next section.

4.3 Evaluation

Let A_E denote the model selected by the estimation method E. $\mathrm{RMSE}(\mathrm{A}_E)$ represents the generalization root mean squared error of that model in a given time series problem. Hopefully, A_E is equal to A_*, which represents the best model that should be selected. In such a case, $\mathrm{RMSE}(\mathrm{A}_E)$ would be optimal according to the pool of available models.

$\mathrm{RMSE}(\mathrm{A}_E)$ is used to quantify and compare different estimation methods. Specifically, the percentage difference between the error of the model selected by the estimator $\mathrm{E}(\mathrm{RMSE}(\mathrm{A}_E))$ and the error of A_* is computed. This can be formalized as follows:

$$\text{Selection Loss } (E) = \frac{\mathrm{RMSE}(\mathrm{A}_E) - \mathrm{RMSE}(\mathrm{A}_*)}{\mathrm{RMSE}(\mathrm{A}_*)} \times 100 \tag{2}$$

where $\mathrm{SelectionLoss}(E)$ denotes the generalization error associated with the estimator E for a given data set. This error is zero in case E selects A_*, which is the correct choice. Otherwise, there is a positive performance loss associated with the model selection process, which is quantified by Eq. 2.

This analysis is carried out with multiple time series problems. Thus, from the definition in Eq. 2 the following statistics summarise the quality of an estimation method for model selection:

- Selection Accuracy (SA): how often a performance estimation method picks the best possible forecasting model. This can be quantified as the proportion of times that RMSE(E) is equal to zero;
- Average Loss when Wrong (ALW): The average loss incurred by picking the wrong forecasting model (not selecting A_*). This loss is quantified by taking the average of RMSE(E) across all the problems when A_E is different from A_*;
- Average Loss (AL): A combination of the two previous measurements: ALW but also taking into account when E selects A_*, in which case the loss is zero.

The three metrics listed above enable comparison between different estimators according to their ability to select the best forecasting model among a set of alternatives.

4.4 Learning Algorithms and Estimation Methods

For each time series problem, each estimator compares 50 alternative algorithms and selects the one which maximizes the expected performance. The models are obtained using different parameter settings of the following learning algorithms: support vector regression [41], multivariate adaptive regression splines [42], random forests [43], projection pursuit regression [44], rule-based regression based on Cubist [45], multi-layer perceptron [46], generalized linear regression [47], Gaussian processes [41], principal components regression [48], partial least squares regression [48], and extreme gradient boosting [49]. The algorithms and corresponding parameters are described in Table 1.

The experiments are focused on regression learning algorithms, which have shown competitive forecasting performance relative to traditional approaches such as ARIMA [16] or exponential smoothing [17] (c.f. Sect. 2.2). Similar studies could be carried out for traditional approaches.

A total of 10 performance estimation methods were applied. These are described below:

- `K-fold cross-validation` (`CV`): First, the time series observations are randomly shuffled and split into K folds. Then, each fold is iteratively selected for testing. A model is trained on K-1 folds and tested in the remaining one. This approach breaks the temporal order of observations, which is problematic for dependent data such as time series [2]. However, it has been shown that `CV` is applicable in some time series scenarios [4];
- `Blocked K-fold cross-validation` (`CV-Bl`): This approach is identical to `CV`. The difference is that `CV-Bl` does not shuffle the observations before assigning them to different folds. This leads to K folds of contiguous observations. Bergmeir et al [3] and Cerqueira et al [5] recommend this approach for estimating the performance of forecasting models if the time series is stationary;
- `Modified Cross-validation` (`CV-Mod`): The modified cross-validation is a variant of `CV` which attempts at decreasing the dependency between training and testing observations [12]. First, time series observations are randomly shuffled into K folds, similarly to `CV`. Then, in each iteration of the cross-validation procedure, some of the training observations are removed. Particularly, the training observations which are within p (the size of the auto-regressive process) observations of any testing point are removed. While this process increases the independence among observations, a considerable number of observations are removed;
- `hv-Blocked K-fold cross-validation` (`CV-hvBl`): The hv-blocked K-fold cross-validation [13] is a variant of `CV-Bl`. Similarly to `CV-Mod`, it removes some instances to decrease the dependency among observations. Specifically, for each iteration,

the adjacent p observations between training and testing are removed. This process creates a small gap between the two sets;

- `Holdout`: This method represents the typical out-of-sample estimation approach, in which the final part of the time series is held out for testing. This process runs in a single iteration, in which the initial 70% of observations are used for training, while the subsequent 30% ones used for testing;
- `Repeated Holdout` (`Rep-Holdout`): An extension of `Holdout` in which this process is repeated K times in multiple, randomized, testing periods [5, 6]. For each one of the K iterations, a random point is chosen in the time series. Then, the 60% of observations (out of the total time series length n) before this point are used for training, while the subsequent 10% of observations (out of n) are used for testing. Note that the window for selecting this random point is restricted by the size of the training and testing sets [5]. This method is also known as Monte-Carlo cross-validation [50];
- `Prequential in Blocks` (`Preq-Bls`): The prequential evaluation methodology is applied using K blocks of data and a growing window [9]. In the first iteration, the initial block containing the first n/K observations is used for training, while the subsequent block (also containing n/K observations) is used for testing. Then, these two blocks are merged together and used for training in the second iteration. In this iteration, the third block of data is used for testing. This process continues until the last block is tested. The `Preq-Bls` procedure is commonly used for evaluating time series models. This method is often referred to as time series cross-validation[3,4];
- `Prequential in Sliding Blocks` (`Preq-Sld-Bls`): This method represents a variant of `Preq-Bls` but is applied with a sliding window. This means that, after each iteration, the oldest block of data is discarded. Therefore, in each iteration, a single block of observations is used for training, and another one is used for testing;
- `Trimmed Prequential in Blocks` (`Preq-Bls-Trim`): A variant of `Preq-Bls` in which the initial splits are discarded due to low sample size: The initial iterations use a training sample size that may not be representative of the complete available time series, which may bias the results. Formally, according to this method, only the final 60% of the K iterations of the `Preq-Bls` procedure are considered. For example, if `Preq-Bls` splits the time series into 10 blocks, the method `Preq-Bls-Trim` only considers (and averages) the results on the last 60% (i.e. 6) iterations;
- `Prequential in Blocks with a Gap` (`Preq-Bls-Gap`): A final variant of `Preq-Bls`, in which a gap is introduced between the training and testing sets. In each iteration, there is a block of n/K observations splitting the training and test sets. Similarly to `CV-Mod` e `CV-hvBl`, the motivation for this process is to increase the independence between the two sets.

The work by Cerqueira et al. [5] provides more information on these methods, including visual representations. Two approaches are used to combine the results of the K iterations of each estimator: the average error according to the arithmetic mean, and the average rank.

The number of folds or repetitions K of the estimation methods is set to 10, where applicable.

[3] https://robjhyndman.com/hyndsight/tscv/.

[4] The *model_selection* module from the *scikit-learn* Python library designates this method as **TimeSeriesSplits.**

Table 2 Results for all methods over the 3111 time series

	Average error			Average rank		
	SA	ALW	AL	SA	ALW	AL
CV-Bl	**0.10**	0.34 ± 0.72	0.29 ± 0.68	0.08	**0.32 ± 0.64**	**0.28 ± 0.64**
Preq-Bls	**0.09**	**0.33 ± 0.70**	**0.28 ± 0.68**	0.08	0.34 ± 0.68	0.30 ± 0.66
CV-hvBl	**0.10**	0.35 ± 0.72	0.30 ± 0.69	0.09	**0.33 ± 0.64**	**0.28 ± 0.63**
CV-Mod	**0.07**	**0.33 ± 0.68**	**0.30 ± 0.66**	0.06	0.35 ± 0.70	0.31 ± 0.68
Preq-Bls-Gap	**0.09**	**0.36 ± 0.79**	**0.30 ± 0.75**	0.07	0.37 ± 0.73	0.32 ± 0.70
CV	**0.09**	**0.38 ± 0.82**	**0.32 ± 0.78**	0.07	0.40 ± 0.74	0.35 ± 0.73
Preq-Sld-Bls	**0.07**	0.39 ± 1.06	0.34 ± 1.03	0.06	**0.37 ± 0.99**	**0.33 ± 0.93**
Preq-Bls-Trim	**0.09**	0.41 ± 0.86	0.35 ± 0.83	0.08	**0.40 ± 0.76**	**0.35 ± 0.74**
Rep-Holdout	**0.08**	**0.42 ± 0.88**	**0.35 ± 0.85**	0.07	0.40 ± 0.79	0.36 ± 0.75
Holdout	0.07	0.67 ± 1.79	0.58 ± 1.64	0.07	0.67 ± 1.79	0.58 ± 1.64

Methods are ordered by increasing values of AL . Each metric is bold according to which approach (average rank or average error) provides the best result for each estimator

4.5 Results

This section describes the results of the experiments. Each research question is addressed in turn.

4.5.1 RQ1: Selection Accuracy of the Estimators

The main results are presented in Table 2. This table shows the score of each estimator over the 3111 time series, both for the average error and average rank approaches. In the table, the methods are ordered by increasing values of AL, obtained either by average rank or average error.

The selection accuracy of the estimators for finding the best predictive model ranges from 7 to 10%. These values represent a significant improvement relative to a random selection procedure, which has an expected accuracy of 2% (1 over 50 possible alternative models). Notwithstanding, this degree of accuracy across all methods means that a given estimator will most probably fail to select the most appropriate model in the available pool. In relative terms, CV-Bl and CV-hvBl show the best score (10%) while CV-Mod, Preq-Sld-Bls, and Holdout show the worst one (7%).

Overall, the selection accuracy scores (SA) for finding the best predictive model highlight the importance of studying the subsequent shortfall: how much forecasting performance is lost by picking the wrong model.

4.5.2 RQ2: Performance Loss During Model Selection

The ALW and AL scores quantify the average performance loss of the estimator. Both are presented for completeness in Table 2. Notwithstanding, the analysis focuses on the AL metric as it also incorporates the cases in which the estimator selects the correct model.

The scores of AL range from 0.28% (CV-Bl, Preq-Bls, and CV-hvBl) and 0.58% (Holdout). As explained before, this metric represents the average (median) difference between the loss of the forecasting model selected by the respective estimator and the loss of

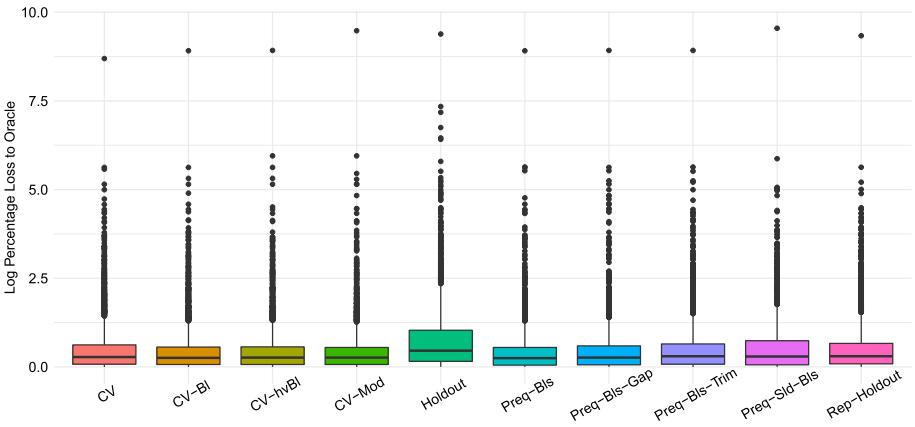

Fig. 3 Distribution of the AL (log-scaled) incurred by each estimation method relative to the Oracle using all time series. The results suggest a large number of outliers for all methods. This means that, in some time series, the selected model suffers large errors relative to the model that should have been selected

the best possible solution available, in which the average is computed across all time series. Essentially, by applying one of the best estimators one can expect a performance loss of about 0.28% concerning the best predictive model in the available pool. Note that these results are highly dependent on the particular pool of available forecasting algorithms. The values for average loss are considerable. In many domains of application, each increment in forecasting performance has a considerable financial impact on organizations [51]. Therefore, it is crucial to maximize this performance. However, unless the domain is sensitive to small differences in performance, the chosen estimation method is not a critical factor for performance.

Overall, except for `Holdout` which shows an AL significantly higher than the rest, the results are comparable across all estimation methods. Especially when considering the dispersion (IQR), which is quite significant. Even the standard cross-validation procedure (`CV`), which is a poor method for performance estimation for forecasting [5], only shows an AL difference of 0.04% to the best estimators. This shows that breaking the temporal order of observations during model selection is, in general, not problematic for time series forecasting tasks.

In general, all estimators present considerable variability in both ALW and AL. This dispersion is further explored in Fig. 3, which shows the distribution of the loss of each estimator relative to the oracle. This loss is non-negative and, as described before, it is zero when the estimator selects the best possible solution. The figure shows that all estimators suffer large errors in some data sets. This means that, while the median errors are comparable, one may be exposed to a significant performance loss irrespective of the estimation method used.

4.5.3 RQ3: Sensitivity Analysis on the Sample Size

This section analyses the effect of the time series sample size on the experimental results. A previous study by Cerqueira et al. [19] showed that the training sample size of time series is an important factor in the relative forecasting performance among different predictive models. Thus, the aim is to study this effect in estimation methods when applied to model selection. To accomplish this, the time series are split into two groups: a group that consists of 371

Table 3 Results for each method in 371 time series with a sample size below 1000 observations

	Average error			Average rank		
	SA	ALW	AL	SA	ALW	AL
CV-hvBl	**0.09**	1.53 ± 2.96	1.28 ± 2.89	0.06	**1.35 ± 2.67**	**1.19 ± 2.41**
CV	0.08	**1.45 ± 3.08**	1.30 ± 2.78	**0.09**	1.45 ± 3.10	**1.21 ± 2.79**
Preq-Bls-Trim	**0.08**	1.66 ± 3.29	1.39 ± 3.19	0.07	**1.46 ± 2.91**	**1.30 ± 2.76**
CV-Mod	**0.07**	**1.54 ± 2.93**	**1.33 ± 2.95**	0.05	1.54 ± 3.14	1.34 ± 3.04
CV-Bl	**0.08**	1.55 ± 2.97	**1.34 ± 2.88**	0.06	**1.53 ± 2.83**	1.38 ± 2.70
Rep-Holdout	**0.09**	1.64 ± 3.22	1.40 ± 2.96	0.07	**1.57 ± 2.80**	**1.36 ± 2.74**
Preq-Bls-Gap	**0.07**	1.65 ± 3.21	1.48 ± 3.01	0.06	**1.53 ± 3.21**	**1.36 ± 3.05**
Preq-Bls	**0.08**	1.60 ± 2.72	**1.39 ± 2.65**	0.06	**1.59 ± 3.13**	1.44 ± 2.89
Preq-Sld-Bls	**0.04**	**1.77 ± 3.23**	**1.70 ± 3.19**	0.02	1.79 ± 2.97	1.76 ± 2.98
Holdout	0.06	2.43 ± 5.75	2.25 ± 5.77	0.06	2.43 ± 5.75	2.25 ± 5.77

Methods are ordered by increasing values of AL. Each metric is bold according to which approach (average rank or average error) provides the best result for each estimator

Table 4 Results for each method in 2740 time series with sample size above 1000 observations

	Average error			Average rank		
	SA	ALW	AL	SA	ALW	AL
Preq-Bls	**0.09**	**0.29 ± 0.58**	**0.24 ± 0.57**	0.08	0.30 ± 0.56	0.26 ± 0.56
CV-Bl	**0.10**	0.30 ± 0.56	0.25 ± 0.56	0.09	**0.28 ± 0.52**	**0.24 ± 0.52**
CV-hvBl	**0.10**	0.30 ± 0.57	0.26 ± 0.57	0.09	**0.29 ± 0.53**	**0.24 ± 0.51**
CV-Mod	**0.07**	**0.30 ± 0.54**	**0.26 ± 0.53**	0.06	0.30 ± 0.56	0.28 ± 0.56
Preq-Bls-Gap	**0.09**	**0.31 ± 0.60**	**0.26 ± 0.61**	0.07	0.32 ± 0.57	0.28 ± 0.58
Preq-Sld-Bls	0.07	0.33 ± 0.81	0.28 ± 0.77	0.07	**0.31 ± 0.73**	**0.27 ± 0.69**
CV	**0.09**	**0.33 ± 0.65**	**0.29 ± 0.62**	0.07	0.35 ± 0.60	0.32 ± 0.60
Preq-Bls-Trim	**0.09**	0.36 ± 0.69	**0.30 ± 0.68**	0.08	**0.35 ± 0.61**	0.31 ± 0.61
Rep-Holdout	**0.08**	0.36 ± 0.70	**0.30 ± 0.68**	0.07	**0.35 ± 0.63**	0.31 ± 0.62
Holdout	0.07	0.59 ± 1.41	0.51 ± 1.33	0.07	0.59 ± 1.41	0.51 ± 1.33

Methods are ordered by increasing values of AL. Each metric is bold according to which approach (average rank or average error) provides the best result for each estimator

time series (out of 3111) with less than 1000 observations; and another group with 2740 time series with a sample size above 1000 data points. Then, the previous analysis is carried out for each group of time series.

The results are shown in Tables 3and 4, in which the first presents the results for the time series with a low sample size, and the second shows the results for the group of time series with a sample size above 1000 observations. There are considerable differences between the two groups. All metrics are noticeably better for larger sample sizes. This result indicates that the model selection task is easier for all estimators if more data is available, which is not surprising.

The results for the group of larger time series (more than 1000 observations) do not vary considerably relative to the results shown in Table 2 for all time series. This is expected because most of the available time series are part of this group. Notwithstanding, the group

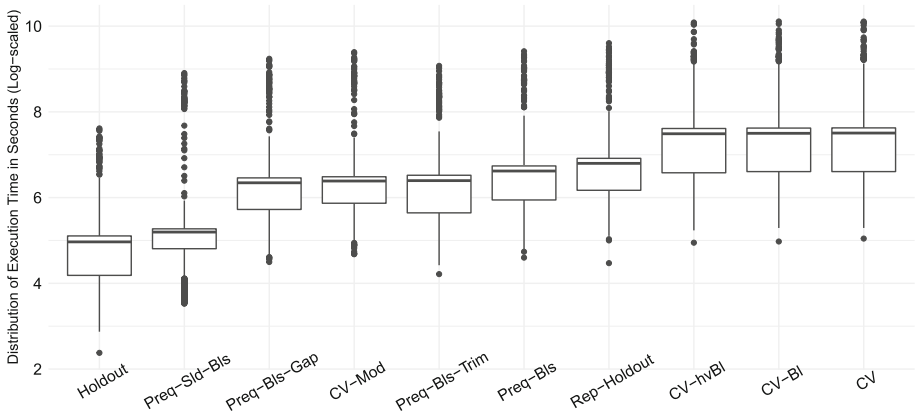

Fig. 4 Distribution of the execution time in seconds (log-scaled), of each estimator across the 3111 time series

of smaller time series (less than 1000 data points) contains 371 time series, which is a considerable amount. In terms of relative performance for this group of time series, `CV-hvBl`, `CV`, and `Preq-Bls-Trim` show the best estimation ability.

An interesting result is how much `Preq-Bls-Trim` improved relative to `Preq-Bls` in the group of small time series. As a reminder, `Preq-Bls` is a common procedure used to evaluate predictive models applied in time-dependent scenarios. For example, this strategy is implemented in the widely used *scikit-learn* [52] Python library as `TimeSeriesSplits`. Our results show that `Preq-Bls-Trim`, which discards the initial iterations of `Preq-Bls`, presents better results for smaller time series while requiring fewer computational resources. This indicates that, for small sample sizes, the initial iterations of the estimation procedure may not be representative of the complete time series. They significantly distort the estimation, and this leads to a poor model selection by `Preq-Bls`. By discarding the initial iterations, the remaining folds are more representative of the complete time series.

4.5.4 RQ4: Analysis of the Averaging Approaches

Regarding the comparison between average error and average rank (also known as voting cross-validation [8]), the results are comparable. The average error leads to a systematic, but marginally, better selection accuracy. Across all data sets, the ALW and AL scores are comparable. The average rank shows slightly better ALW and AL scores when the time series comprises less than 1000 observations. Note that, in the case of `Holdout`, the average error and average rank are identical because this estimator relies on a single iteration. Since the average rank requires the extra computation of computing ranks, the average error may be a preferable approach for simplicity.

4.5.5 RQ5: Execution Time Analysis

The final research question analyses the execution time of each estimator. The execution time denotes the time a given estimator takes to carry out the estimation process and produce a ranking of the predictive models under comparison.

The results of this analysis are shown in Fig. 4, which shows the distribution of the execution time of each estimator across the 3111 data sets. The estimators are ordered (from

left to right) by median execution time. Holdout is the quickest estimator while CV is the slowest one, on average. In general, the execution time is correlated with the number of iterations and the amount of data used by the estimator.

5 Discussion

This work analyzed the ability of several estimators for model selection in time series forecasting tasks. Experiments were carried out using 3111 univariate time series, 10 estimation methods, and 50 auto-regressive models. Each estimator tests each one of the 50 models using the available data and selects one for making predictions in a test set.

An important finding is that, given the number of possible alternative models (50), all estimators show low accuracy in selecting the best available model (**RQ1**). Notwithstanding, the scores are significantly better than a random selection procedure. Accordingly, this lead to the study of the differences in performance between the model selected by each estimator and the model that should have been selected (the one maximizing forecasting performance on test data). Overall, the average difference across all time series ranges from about 0.28 to about 0.58%. However, there is a large variability in the results across all estimation methods (**RQ2**). These values may be significant in many domains and show that the evaluation process of time series models is an important task that should not be overlooked.

Another finding is that the results vary considerably when controlling for time series sample size. For larger time series (comprising more than 1000 observations), the results are consistent with the conclusions presented above. On the other hand, the model selection task is significantly more difficult when smaller sample sizes are available for all estimation methods. The best overall estimator incurs an average performance loss of 0.28%, but this value increases to 1.19% for the group of time series with less than 1000 data points (**RQ3**).

Finally, the execution time of each approach was analyzed. These correlate with the amount of data each estimator uses (**RQ4**).

Considerable distinctions were found in the relative performance of the estimators when applied for performance estimation [5] and when used for model selection. For performance estimation, Cerqueira et al. [5] report that CV is the worst estimator, across the 174 time series that they study. For model selection purposes, which is the topic of this paper, CV is competitive with the best approaches, especially for smaller sample sizes. Cerqueira et al. [5] did not find any impact of the time series sample size on the relative performance of the estimators. This factor is important for model selection, though it is also important to remark that the process for analyzing this impact is different in the two studies. Moreover, this impact is noticeable both in relative terms, where the ranking of the estimators is different, and in absolute terms, in the sense that larger sample sizes lead to better overall model selection results.

In terms of AL, there is a 0.3% gap between the best estimator (CV-Bl applied with average rank) and the worst estimator (Holdout). This gap decreases to 0.06%, except for Holdout. The relevance of the differences in the AL scores diminishes further when inspecting the dispersion (using IQR) across the time series, which is often more than double the average score. In this context, there is no significant difference between the estimators analyzed in this work, except for Holdout.

6 Conclusions and Future Work

Different estimation methods were studied for model selection in time series forecasting tasks. While these methods have been studied for performance estimation [3–5], model selection is a different problem. Therefore, the best estimator for performing model selection may not be the most appropriate for performance estimation.

The estimation methods were analyzed from two main perspectives. First, this work aimed at quantifying their accuracy: how often each estimator selects the best model, i.e., the one with the best performance on test data. The second objective was to study how much performance was lost when the estimator did not select the best model.

The experiments carried out in the paper are available in an online repository (c.f. footnote 1).

The analysis was based on a set of experiments. These experiments included 3111 time series, 10 estimation methods, and 50 predictive models. The results of the analysis suggest the following conclusions. All estimation methods show low accuracy for finding the most appropriate model, though this score is better than a random selection procedure. The overall performance loss during model selection ranges between 0.28 and 0.58%. No considerable difference was found among the different compared methods. The exception is `Holdout`, which should be avoided unless there is a large sample size available. If the domain is sensitive to small differences in performance, `Preq-Bls` and `CV-Bl` presents the overall best model selection ability.

The results also suggest that: (i) taking the average rank of models, instead of the average error, leads to a comparable performance in terms of model selection; (ii) all estimators improve their performance when more data is available, and (iii) the execution time of each method is highly correlated with the amount of data it uses for model selection.

The idea of studying how much performance is lost during model selection may foster additional research on this topic. For example, carry a similar analysis for i.i.d. data sets and standard machine learning tasks such as regression or classification. Another possible future research direction is carrying out a similar study but in settings involving multiple decisions. For example, selecting not only the predictive model but also the most appropriate pre-processing steps.

Author Contributions All authors contributed to writing and research.

Funding The work of L. Torgo was undertaken, in part, thanks to funding from the Canada Research Chairs program; the work of Carlos Soares was partially funded by projects ConnectedHealth (no. 46858), supported by Competitiveness and Internationalisation Operational Programme (POCI) and Lisbon Regional Operational Programme (LISBOA 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), by the project Safe Cities - Inovação para Construir Cidades Seguras, with the reference POCI-01-0247-FEDER-041435, co-funded by the European Regional Development Fund (ERDF), through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under the PORTU- GAL 2020 Partnership Agreement, by project NextGenAI - Center for Responsible AI (2022-C05i0102-02), supported by IAPMEI, and also by FCT plurianual funding for 2020–2023 of LIACC (UIDB/00027/2020_UIDP/00027/202

Data Availibility All experiments and data are publicly available (c.f. footnote 1)

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose

Consent to participate Not applicable

Consent for publication Not applicable

Ethics approval Not applicable

References

1. Breiman L, Spector P (1992) Submodel selection and evaluation in regression. The x-random case. International statistical review/revue internationale de Statistique pp. 291–319
2. Arlot S, Celisse A et al (2010) A survey of cross-validation procedures for model selection. Stat Surv 4:40–79
3. Bergmeir C, Benítez JM (2012) On the use of cross-validation for time series predictor evaluation. Inf Sci 191:192–213
4. Bergmeir C, Hyndman RJ, Koo B (2018) A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput Stat Data Anal 120:70–83
5. Cerqueira V, Torgo L, Mozetič I (2020) Evaluating time series forecasting models: an empirical study on performance estimation methods. Mach Learn 109:1–32
6. Tashman LJ (2000) Out-of-sample tests of forecasting accuracy: an analysis and review. Int J Forecast 16(4):437–450
7. Mozetič I, Torgo L, Cerqueira V, Smailović J (2018) How to evaluate sentiment classifiers for twitter time-ordered data? PLoS ONE 13(3):e0194,317
8. Yang Y (2007) Consistency of cross validation for comparing regression procedures. Ann Stat 35(6):2450–2473
9. Dawid AP (1984) Present position and potential developments: Some personal views statistical theory the prequential approach. J R Stat Soc Ser A (General) 147(2):278–290
10. Opsomer J, Wang Y, Yang Y (2001) Nonparametric regression with correlated errors. Stat Sci 16(2):134–153
11. Snijders TA (1988) On model uncertainty and its statistical implications. Springer, pp 56–69
12. McQuarrie AD, Tsai CL (1998) Regression and time series model selection. World Scientific
13. Racine J (2000) Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. J Econ 99(1):39–61
14. Gama J, Rodrigues PP, Sebastião R (2009) In: Proceedings of the 2009 ACM symposium on Applied Computing, pp 1496–1500
15. Makridakis S, Spiliotis E, Assimakopoulos V (2018) Statistical and machine learning forecasting methods: Concerns and ways forward. PLoS ONE 13(3):e0194,889
16. Chatfield C (2000) Time-series forecasting. CRC press
17. Gardner ES Jr (1985) Exponential smoothing: the state of the art. J Forecast 4(1):1–28
18. Spiliotis E, Makridakis S, Semenoglou AA, Assimakopoulos V (2022) Comparison of statistical and machine learning methods for daily sku demand forecasting. Oper Res 22(3):3037–3061
19. Cerqueira V, Torgo L, Soares C (2022) A case study comparing machine learning with statistical methods for time series forecasting: size matters. J Intell Inf Syst 59:1–19
20. Makridakis S, Spiliotis E, Assimakopoulos V (2020) The m5 accuracy competition: results, findings and conclusions. Int J Forecast 38:1346
21. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) In: Advances in neural information processing systems, pp 3146–3154
22. Cerqueira V, Torgo L, Oliveira M, Pfahringer B (2017) In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (IEEE, 2017), pp 242–251
23. Cerqueira V, Torgo L, Pinto F, Soares C (2019) Arbitrage of forecasting experts. Mach Learn 108(6):913–944
24. Corani G, Benavoli A, Augusto J, Zaffalon M (2020) Automatic forecasting using gaussian processes. arXiv preprint arXiv:2009.08102
25. Oreshkin BN, Carpov D, Chapados N, Bengio Y (2019) N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437
26. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) Deepar: Probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
27. Smyl S (2020) A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int J Forecast 36(1):75–85
28. Lim B, Arık SÖ, Loeff N, Pfister T (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 37(4):1748–1764

29. Chen MR, Zeng GQ, Lu KD, Weng J (2019) A two-layer nonlinear combination method for short-term wind speed prediction based on elm, enn, and lstm. IEEE Internet Things J 6(4):6997–7010
30. Zhao F, Zeng GQ, Lu KD (2019) Enlstm-wpeo: Short-term traffic flow prediction by ensemble lstm, nnct weight integration, and population extremal optimization. IEEE Trans Veh Technol 69(1):101–113
31. Taylor SJ, Letham B (2018) Forecasting at scale. Am Stat 72(1):37–45
32. Triebe O, Hewamalage H, Pilyugina P, Laptev N, Bergmeir C, Rajagopal R (2021) Neuralprophet: Explainable forecasting at scale. arXiv preprint arXiv:2111.15397
33. Bandara K, Hewamalage H, Liu YH, Kang Y, Bergmeir C (2021) Improving the accuracy of global forecasting models using time series data augmentation. Pattern Recogn 120:108,148
34. Hewamalage H, Bergmeir C, Bandara K (2022) Global models for time series forecasting: A simulation study. Pattern Recogn 124:108,441
35. Kennel MB, Brown R, Abarbanel HD (1992) Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical Rev A 45(6):3403
36. Brazdil PB, Soares C (2000) European conference on machine learning. Springer, pp 63–75
37. Benavoli A, Corani G, Mangili F (2016) Should we really use post-hoc tests based on mean-ranks? J Mach Learn Res 17(1):152–161
38. Abdulrahman SM, Brazdil P, van Rijn JN, Vanschoren J (2018) Speeding up algorithm selection using average ranking and active testing by introducing runtime. Mach Learn 107(1):79–108
39. Makridakis S, Spiliotis E, Assimakopoulos V (2020) The m4 competition: 100,000 time series and 61 forecasting methods. Int J Forecast 36(1):54–74
40. Hyndman R, Yang Y (2019) tsdl: Time series data library. https://finyang.github.io/tsdl/, https://github.com/FinYang/tsdl
41. Karatzoglou A, Smola A, Hornik K, Zeileis A (2004) kernlab-an s4 package for kernel methods in r. J Stat Softw 11(9):1–20
42. Milborrow S (2012) earth: multivariate adaptive regression spline models
43. Wright MN (2015) ranger: a fast implementation of random forests. R package
44. Friedman JH, Stuetzle W (1981) Projection pursuit regression. J Am Stat Assoc 76(376):817–823
45. Kuhn M, Weston S, Keefer C (2014) N.C.C. code for Cubist by Ross Quinlan, Cubist: rule- and instance-based regression modeling. R package version 0.0.18
46. Cannon AJ (2017) monmlp: Multi-layer perceptron neural network with optional monotonicity constraints. https://CRAN.R-project.org/package=monmlp. R package version 1.1.5
47. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22
48. Mevik BH, Wehrens R, Liland KH (2016) pls: partial least squares and principal component regression. https://CRAN.R-project.org/package=pls. R package version 2.6-0
49. Chen T, Guestrin C (2016) In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794
50. Picard RR, Cook RD (1984) Cross-validation of regression models. J Am Stat Assoc 79(387):575–583
51. Jain CL (2017) Answers to your forecasting questions. J Bus Forecast 36(1):3
52. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830