Schnaubelt, Matthias

Working Paper

A comparison of machine learning model validation schemes for non-stationary time series data

FAU Discussion Papers in Economics, No. 11/2019

Provided in Cooperation with:
Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics

Suggested Citation: Schnaubelt, Matthias (2019) : A comparison of machine learning model validation schemes for non-stationary time series data, FAU Discussion Papers in Economics, No. 11/2019, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Nürnberg

This Version is available at:
https://hdl.handle.net/10419/209136

Discussion Papers
in Economics

No. 11/2019

A comparison of machine learning model validation schemes for non-stationary time series data

Matthias Schnaubelt
University of Erlangen-Nürnberg

ISSN 1867-6707

Friedrich-Alexander-Universität Erlangen-Nürnberg
Institute for Economics
https://www.iwf.rw.fau.de/discussion-papers/

A comparison of machine learning model validation schemes for non-stationary time series data

Matthias Schnaubelt[a,1]

[a] *University of Erlangen-Nürnberg, Department of Statistics and Econometrics, Lange Gasse 20, 90403 Nürnberg, Germany*

Abstract

Machine learning is increasingly applied to time series data, as it constitutes an attractive alternative to forecasts based on traditional time series models. For independent and identically distributed observations, cross-validation is the prevalent scheme for estimating out-of-sample performance in both model selection and assessment. For time series data, however, it is unclear whether forward-validation schemes, i.e., schemes that keep the temporal order of observations, should be preferred. In this paper, we perform a comprehensive empirical study of eight common validation schemes. We introduce a study design that perturbs global stationarity by introducing a slow evolution of the underlying data-generating process. Our results demonstrate that, even for relatively small perturbations, commonly used cross-validation schemes often yield estimates with the largest bias and variance, and forward-validation schemes yield better estimates of the out-of-sample error. We provide an interpretation of these results in terms of an additional evolution-induced bias and the sample-size dependent estimation error. Using a large-scale financial data set, we demonstrate the practical significance in a replication study of a statistical arbitrage problem. We conclude with some general guidelines on the selection of suitable validation schemes for time series data.

Keywords: machine learning; model selection; model validation; time series; cross-validation

Email address: matthias.schnaubelt@fau.de (Matthias Schnaubelt)

[1]The author has benefited from helpful discussions with Ingo Klein, Jonas Dovern, Thomas Fischer, Christopher Krauss and Alexander Glas.

1. Introduction

Machine learning methods are increasingly used for time series predictions. Multiple reasons contribute to their popularity: First, they are able to choose relevant variables from a large number of input candidates and neglect unimportant ones. Second, they can handle high-dimensional data sets where the number of features may even exceed the sample size. Third, machine learning models are general-purpose methods in the sense that they do not make assumptions about the underlying process generating the time series, and are able to cope well with non-linear dynamics (Zhang et al., 1998, 2001; Zhang, 2003). Consequently, they are able to generalize well on unseen data and have been shown to outperform classical time series models for various prediction tasks. Given these advantages, it is not surprising that machine learning has also been extensively applied to economic and business time series.[2]

An important step in the application of machine learning is model selection, with the goal of choosing the best model among a number of candidates by considering a suitable error measure. However, as first noted by Larson (1931), estimation of the model error by using data previously used for training is likely to be too optimistic, as the model may overfit the given data (Hastie et al., 2009), giving rise to the resubstitution error. Therefore, to select the best model from a number of choices or to assess the generalization ability of a final model, available data is usually divided into at least one training and one validation set. As the latter is used to estimate the error of a model trained on the former, one expects to obtain a better estimate of the model's out-of-sample predictive performance. Using as many validation sets as data points available then leads to *leave-one-out cross-validation* as proposed by Stone (1974), Allen (1974) and Geisser (1975). Subsequently, several extensions to the leave-one-out scheme have been proposed, of which k-fold cross-validation (Geisser, 1975) is most common: Data is randomly divided into k equally-sized subsets, which are used once for computing a validation error, and $k-1$ times for model training. The final error estimate is the average of all validation subset errors. Today, *k-fold cross-validation* can be considered as the standard validation method for most machine learning applications.

To theoretically show the applicability of cross-validation, one must assume that observations are independent and identically distributed. In case of time series, however, this assumption does

[2] For example, financial machine learning has been used for stock market return predictions (Kim, 2003; Schnaubelt et al., 2018; Huck, 2019), option valuation (Andreou et al., 2008), market volatility forecasts (Mittnik et al., 2015), and commodity price prediction (Weron, 2014; Panapakidis and Dagoumas, 2016; Lago et al., 2018). A corresponding survey can be found in Henrique et al. (2019). Another area of application is demand prediction, e.g., for electric utility load forecasts (Baliyan et al., 2015; Hong and Fan, 2016) or retail sales predictions (Beheshti-Kashi et al., 2015). Yet other works examine the use of machine learning models for macroeconomic forecasting (Teräsvirta et al., 2005; Wohlrabe and Buchen, 2014; Plakandaras et al., 2015).

not hold as future observations may depend on past ones. To properly account for the time-dependence of observations, empirical research based on traditional time series models typically reverts to validation schemes that keep the temporal order of observations between training and validation sets. Hence, we refer to these methods as *forward-validation* methods. One example of forward-validation would be a rolling-origin scheme, where the model is trained on a block of data that grows with every split, and validation errors are computed on all later data. Tashman (2000) reviews several different forms of forward-validation. As an alternative to forward-validation, cross-validation may be used, ignoring the fact that the assumption of independence is violated. This may still seem favorable, because cross-validation uses the available data more efficiently in the sense that every observation contributes to the final validation error.

The practical consequences of using cross-validation for time series validation have been analyzed in several studies. Bergmeir and Benitez (2012) conduct an empirical study on both synthetic stationary and real-world time series to compare the performance of cross-validation to the simplest forward-validation scheme, namely last-block validation. Their results reveal no practical consequences of using cross-validation in a time-series context. Further, cross-validation seems to deliver more robust error estimates when compared to last-block validation. However, they conclude that their study is limited to stationary time series. Subsequently, Bergmeir et al. (2014) perform a similar study, but instead evaluate the directional accuracy of regression models as error metric. They conclude that blocked cross-validation is preferable to forward-validation for small samples. More recently, Bergmeir et al. (2018) provide a theoretical justification that cross-validation is applicable to time-series validation for purely autoregressive stationary models as long as all relevant lags are appropriately embedded in the feature matrix. Finally, Cerqueira et al. (2017) extend on the experiments of Bergmeir and Benitez (2012) by including more complex forward-validation schemes. For synthetic stationary time series, they conclude that cross-validation is applicable. For real-world data sets however, they find evidence that forward-validation methods perform better. While it is commonplace to use cross-validation for non-time-series applications, no clear consensus has been reached on the proper method that should be used to validate machine learning models applied to time series data.

One of the reasons why results from synthetic time series and real-world data sets differ is the evolution of the data-generating processes: For real-world data, not only the time-dependence of observations is important, but a more realistic additional assumption is that the data-generating process slowly evolves over time. These underlying dynamics invalidate the assumption of a glob-

ally stationary time series, i.e., perturb stationarity.[3] For example, Guegan (2007) lists a number of phenomena for stock returns yielding such dynamics, among them changes in the unconditional variance, cyclical components, and jump-induced regime shifts. With this paper, we address the open question of choosing the proper validation scheme for time series data by considering the impact of slowly-evolving data-generating processes on validation scheme performance. Further, we seek to provide guidance on the selection of suitable validation schemes in light of slowly-evolving data-generating processes.

In detail, our contributions are as follows: First, extending on current literature, we introduce a research design for the comparative analysis of validation scheme performance. We leverage locally stationary autoregressive processes (Dahlhaus, 2012), i.e., autoregressive stochastic processes with time-varying parameter curves, to generate homogeneous sets of synthetic time series that allow for a fine-grained control of both the type and strength of the stationarity perturbation, which would not be possible with real-world data.

Second, we present simulation studies for choosing the appropriate validation scheme in light of slowly-evolving dynamics. Specifically, we present the results of extensive simulation studies and comprehensively assess key performance metrics of various validation schemes for a broad range of synthetic time series, both in a regression and a classification setting, and for various machine learning models. The specifications of parameter curves are motivated by stylized facts of economic and financial time series, such as cycles or regime shifts. Our study comprises eight common cross- and forward-validation schemes and benchmarks them against last-block schemes. Performance is first assessed in terms of mean squared estimate error, i.e., the difference between the validation scheme's estimate of the out-of-sample error and the actual out-of-sample error. Next, we split this error into a bias and a variance component which we analyze separately, as a low bias is preferred for model assessment, while a low variance is beneficial when the goal is model selection. Our experiments explicitely consider the most important influencing dimensions of validation scheme performance: the type of slowly evolving process dynamics, the strength of the perturbation of stationarity, and the sample size.

We demonstrate that, depending on the type of process dynamics, the application of cross-validation schemes to time series data comes at great risk: While theoretically applicable and empirically resulting in the lowest errors in case of global stationarity, cross-validation performs considerably worse in situations where processes are allowed to evolve over time. Furthermore, our results

[3] We use the word 'dynamics' to describe the evolution of the data-generating process.

show that forward-validation methods can achieve a better performance, both in terms of bias and variance, than cross-validation methods in such situations. In the context of autoregressive processes, this is especially true when non-periodic changes of the autoregression coefficients are present. When the strength of the stationarity perturbation is gradually increased, the error of cross-validation estimates is fastest-growing, and forward-validation methods become preferable. Finally, for very high perturbation strengths, last-block validation becomes preferable in terms of variance. By deriving an approximation for the prediction error of a time-varying autoregressive model, we interpret these results in terms of a trade-off between evolution-induced bias and the sample-size dependent generalization ability of the model. Overall, our results suggest that forward-validation methods are not only preferable because they are inherently look-ahead free, but furthermore because they may better estimate the out-of-sample error in presence of slowly changing dynamics.

Third, we demonstrate the practical significance of these findings by showing that similar performance differences arise in a real-world financial data set. Specifically, we employ a well-established financial machine learning model for statistical arbitrage in stock markets, and perform a large-scale comparative study of validation schemes on daily stock price data of all S&P 500 constituents from 1990 to 2015. The results are very similar to those using synthetic data with time-dependent, non-periodic autoregression coefficients: We find that randomized cross-validation performs worst, and forward-validation schemes yield the lowest estimate errors. The magnitude in performance difference between cross- and forward-validation is substantial, as we find that it is in the order of 10 percent of the overall error estimate.

The rest of this paper is organized as follows: We begin by reviewing cross-validation and forward-validation in Section 2. Then, we introduce the methodology of our simulation study in Section 3. Our results are presented in Section 4, and we conclude in Section 5.

2. Cross- and forward-validation schemes

The ability of a model to generalize, i.e., to perform well on unseen data, is of utmost importance in the process of model selection and model validation. Commonly, the generalization error $\mathcal{L}_\mathcal{D}$ is defined as

$$\mathcal{L}_\mathcal{D} = \mathbb{E}_{\boldsymbol{X},Y}\left[\ell(\boldsymbol{X}, Y, \hat{f}(\boldsymbol{X};\hat{\theta})) \mid \mathcal{D}\right]. \tag{1}$$

Therein, we write the available training data as $\mathcal{D} = \{(\boldsymbol{x}_t, y_t)\}_t$, where input (feature) vectors of dimension d are denoted as $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$, and scalar output (target) variables are denoted by

$y \in \mathcal{Y} \subset \mathbb{R}$. The model $\hat{f}$ is trained on $\mathcal{D}$ yielding fitted parameters $\hat{\theta}$, and ℓ is a loss function $\ell(\boldsymbol{x}, y, \hat{y}) : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$. While keeping the training data fixed, the expectation is taken with respect to random variables $\boldsymbol{X}$ and Y. The prediction error (expected generalization error) $\mathcal{L}$ is then given by additionally taking the expectation over all possible realizations of the training data $\mathcal{D}$, i.e., $\mathcal{L} = \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}_{\mathcal{D}}\right]$ (Hastie et al., 2009).

These definitions assume that all data is drawn independently from an underlying distribution. Generally, this assumption is violated for time series data, as observations depend on past values and the data-generating dynamics might evolve over time. Consequently, the definition of the (expected) generalization error is no longer adequate in a time series context, and must be altered to explicitly cope with the time-ordering of the data. The most natural and pragmatic approach is to consider the generalization error as expected error on unobserved future data.[4] We therefore assume that we have training data $\mathcal{D}$ of size T and denote future test data as $\mathcal{D}_{test}$.

During model construction and selection, future test data are unknown. To nevertheless estimate the out-of-sample prediction error $\mathcal{L}$, one often divides available training data into a number of train/test splits (or folds) by means of a validation scheme. For given training data $\mathcal{D}$ and number of splits k, a validation scheme $\mathcal{V}(\mathcal{D}; k)$ is specified by a set of k non-overlapping data splits, i.e.,

$$\mathcal{V}(\mathcal{D}; k) = \left\{(\mathcal{I}_i^t, \mathcal{I}_i^v) \mid \mathcal{I}_i^t, \mathcal{I}_i^v \subset \{0, \ldots, T-1\}; \mathcal{I}_i^t \cap \mathcal{I}_i^v = \emptyset\right\}_{i=0}^{k-1}. \tag{2}$$

For every split i, the model is trained on data indexed by the index set $\mathcal{I}_i^t$, and the loss is determined on validation data indexed by $\mathcal{I}_i^v$. The estimate of the error is then the average out-of-sample loss over all k splits of the validation scheme $\mathcal{V}$:

$$\hat{\mathcal{L}}(\mathcal{D}; \mathcal{V}) = \frac{1}{k} \sum_{i=0}^{k-1} \frac{1}{|\mathcal{I}_i^v|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}(\mathcal{I}_i^v)} \ell(\boldsymbol{x}, y, \hat{f}(\boldsymbol{x}; \hat{\theta}_{\mathcal{I}_i^t})). \tag{3}$$

Therein, $|\mathcal{I}_i^v|$ denotes the cardinality of the i-th validation index set, $\hat{f}(\boldsymbol{x}; \hat{\theta}_{\mathcal{I}_i^t})$ denotes the predictor trained on the i-th set of training data, and $\mathcal{D}(\mathcal{I}_i^v)$ denotes the i-th set of validation data. Depending on the exact choice of $\mathcal{V}$, we can arange existing validation schemes into two main groups: First, cross-validation schemes that use all available training data in every split, and second, forward-validation schemes that preserve the time-ordering of the data. Figure 1 illustrates all splitting schemes that are introduced in the following.

[4] This definition is also used for example by McDonald et al. (2011) and Kuznetsov and Mohri (2014) in the derivation of generalization bounds for time series prediction of non-stationary mixing stochastic processes.

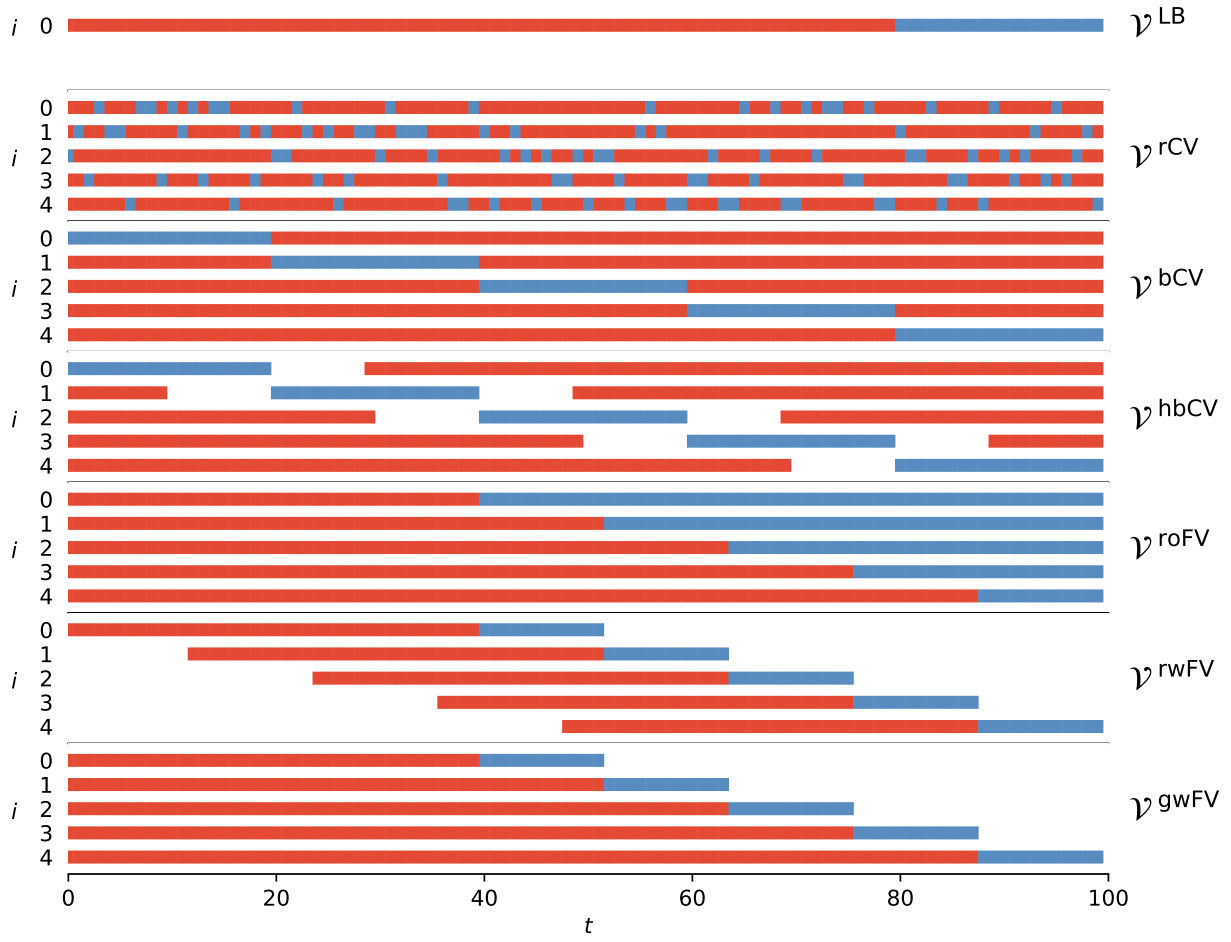


Figure 1: **Illustration of validation data splitting schemes.** We visualize different data splitting schemes $\mathcal{V}$ by their respective validation index sets $\mathcal{I}_i^v$ (blue) and training index sets $\mathcal{I}_i^v$ (red), where i denotes the split index. We show the following schemes: $\mathcal{V}^{\text{LB}}$: last-block validation, $\mathcal{V}^{\text{rCV}}$: random cross-validation, $\mathcal{V}^{\text{bCV}}$: blocked cross-validation, $\mathcal{V}^{\text{hbCV}}$: h-blocked cross-validation, $\mathcal{V}^{\text{roFV}}$: rolling-origin forward-validation, $\mathcal{V}^{\text{rwFO}}$: rolling-window forward-validation, $\mathcal{V}^{\text{gwFO}}$: growing-window forward-validation. With the exception of $\mathcal{V}^{\text{LB}}$, sets are displayed for $k = 5$ splits.

2.1. Cross-validation

Cross-validation schemes have a long history, are commonly used to validate machine learning models and have a sound theoretical foundation – see Arlot and Celisse (2010) for a recent survey. Cross-validation can be traced back to the works of Stone (1974), Allen (1974) and Geisser (1975). Following Arlot and Celisse (2010), we can further distinguish two types of cross-validation: First, *exhaustive data splitting* schemes average over one split for every possible training set of some fixed size. Prominent examples are the *leave-one-out* (Stone, 1974; Allen, 1974; Geisser, 1975), where $k = T$ and $\mathcal{I}_i^v = \{i\}$, and the *leave-p-out* (Shao, 1993), in which every possible subset of p samples is once used as validation set, leading to a total of $k = \binom{T}{p}$ splits. Second, *partial data splitting* schemes do not use all possible subsets, and therefore have a reduced computational complexity.

Well established is *k-fold cross-validation* (Geisser, 1975), which divides the available training data into k subsets of approximately equal size, i.e., $|\mathcal{I}_i^v| \approx \frac{T}{k}\ \forall\ i \in \{0, \dots, k-1\}$. Each of the k subsets is then used as validation set for a model trained on the remaining $k-1$ subsets. Observations in these subsets are drawn randomly without replacement, and we therefore refer to this validation scheme as random cross-validation defined as

$$\mathcal{V}^{\text{rCV}}(k) := \left\{(\mathcal{I}_i^t = \bar{\mathcal{I}}_i^v,\ \mathcal{I}_i^v = \{\pi(i), \pi(i+k), \pi(i+2k), \dots\})\right\}_{i=0}^{k-1}, \tag{4}$$

where π denotes a random permutation of the index set.

The statistical properties of cross-validation are difficult to derive and depend on the actual framework used (Arlot and Celisse, 2010). For i.i.d. observations, the bias of cross-validation has been shown to be positive and decreasing with the size of the training set, which, in case of k-fold cross-validation, increases with the number of splits k. For a linear regression setting and in an asymptotic expansion, the variance of the cross-validation estimate has been shown to decrease with the number of splits k (Burman, 1989).

Bergmeir et al. (2018) provide a theoretical argument that standard random cross-validation is applicable to time series forecasting for purely autoregressive models. This result relies on a number of assumptions such as a stationary autoregressive process and uncorrelated errors, which occurs for example when the time series is embedded appropriately.

Modified versions of k-fold cross-validation have been proposed for use specifically with dependent time series data: Snijders (1988) proposes the use of continuous parts of the time series as validation sets. Transferred to cross-validation, this means that instead of randomly selecting any k subsets of similar size, one divides the data into k time-continuous blocks of observations. Following Bergmeir and Benitez (2012), we refer to this splitting scheme as *blocked cross-validation*, which is formally given by

$$\mathcal{V}^{\text{bCV}}(k) := \left\{\left(\mathcal{I}_i^t = \bar{\mathcal{I}}_i^v, \mathcal{I}_i^v = \left\{\lfloor \tfrac{i}{k} T \rfloor, \lfloor \tfrac{i}{k} T \rfloor + 1, \dots, \lfloor \tfrac{i+1}{k} T \rfloor - 1\right\}\right)\right\}_{i=0}^{k-1}. \tag{5}$$

Under the name *h-block validation*, Burman et al. (1994) introduce an exhaustive splitting scheme that extends on leave-one-out, but removes blocks of size h from either side of every single validation observation to block out potentially dependent observations. In the following, Racine (2000) show that h-block validation is asymptotically inconsistent in the sense of Shao (1993). As a solution, Racine (2000) introduce the so-called hv-block cross-validation, where the validation data of each fold is a continuous block of $2v+1$ observations. To remove dependencies between training and validation subsets, h observations adjacent to either side of the validation set are left out. The model is trained on all $N_{train} - 2h - 2v - 1$ remaining observations. The validation block is rolled

forward by one observation, such that a total of $k = N_{train} - 2v$ validation folds is considered. As this exhaustive data splitting scheme is computationally expensive, Bergmeir and Benitez (2012) consider a blocked form of cross-validation, which divides data into k blocks of continuous data and removes h observations from either side of the validation block. Bergmeir et al. (2014) further analyze this scheme for directional accuracy forecasts of stationary data. In the following, we refer to this scheme as *h-blocked k-fold cross-validation* specified by

$$\mathcal{V}^{\text{hbCV}}(k,h) := \left\{\left(\mathcal{I}_i^t = \bar{\mathcal{I}}_i^v, \mathcal{I}_i^v = \left\{\lfloor \tfrac{i}{k}T \rfloor + h, \lfloor \tfrac{i}{k}T \rfloor + h + 1, \ldots, \lfloor \tfrac{i+1}{k}T \rfloor - h - 1\right\}\right)\right\}_{i=0}^{k-1}. \quad (6)$$

The proper value of h generally depends on the data and should be chosen such that observations that are at least h apart become independent.

2.2. Forward-validation

Compared to cross-validation, forward-validation schemes strictly demand that validation set observations succeed all observations in the respective training set. As such, forward-validation is inherently free of look-ahead bias. On the other hand, forward-validation splits no longer cover all available data, i.e., $\mathcal{I}_i^t \cup \mathcal{I}_i^v = \{0, \ldots, T-1\}$ does not generally hold. This limits the available training data when compared to cross-validation, and data are no longer efficiently used.

The simplest validation scheme is *hold-out* (Devroye and Wagner, 1979; Arlot and Celisse, 2010): A model is trained once on data specified by an index set $\mathcal{I}^t$ and then evaluated on its non-empty complement $\bar{\mathcal{I}}^v$. Generally, $\mathcal{I}^t$ is chosen randomly; however, when using the latest part of the training data as validation set, hold-out represents the simplest forward-validation scheme. The splitting scheme of *last-block validation* is given by

$$\mathcal{V}^{\text{LB}}(f) := \left\{\left(\mathcal{I}^t = \{0, \ldots, \lfloor (1-f)\cdot T \rfloor - 1\},\ \mathcal{I}^v = \bar{\mathcal{I}}^t\right)\right\}, \quad (7)$$

where $f \in (0,1)$ defines the fraction of data used for validation.

In his review, Tashman (2000) discusses several forward-validation schemes that have been used for time series validation. *Rolling-origin evaluation* (for one of the first descriptions see, for example, Armstrong and Grohman, 1972) is a scheme that perpeptually transfers an observation from the future validation set to the training set, and retrains the model after each transfer. Due to large computational costs, we instead consider a modified scheme where the origin is moved forward by blocks of observations. Data before (after) the respective split's origin is used for training (validation). We define this scheme by

$$\mathcal{V}^{\text{roFV}}(k, f_{min}) := \left\{\left(\mathcal{I}_i^t = \bar{\mathcal{I}}_i^v, \mathcal{I}_i^v = \left\{\lfloor f_{min}T + i\kappa^{\text{roFV}} \rfloor, \ldots, T-1\right\}\right)\right\}_{i=0}^{k-1}, \quad (8)$$

where, as before, k denotes the number of splits, $f_{min} \in [\frac{1}{T}, 1)$ is the minimum fraction of data used as training set, and $\kappa^{\text{roFV}} = \frac{(1-f_{min})T}{k}$ denotes the number of observations the origin is shifted by. This scheme has the theoretical advantage that more recent observations, which may be most representative of future dynamics, are used with higher weight, as they appear in more than one validation set. On the other hand, observations at the end of the training data set are validated using a model that has been trained on very early data only.

To overcome this disadvantage, validation data can be limited to a fixed number of observations following the split's training data. As before, the amount of training data increases with every split as it is taken from a flexible-size *growing window*. Growing-window validation was first used by Makridakis (1990) for time series forecasting, and later also by Leitch and Tanner (1991), Thoma (1994) and Pesaran and Timmermann (1995) for further economic research questions. As before, we introduce a blocked variant of growing-window validation to limit computational cost, and define this scheme as

$$\mathcal{V}^{\text{gwFV}}(k, f_{min}) := \{(\mathcal{I}_i^t = \left\{0, \ldots, \lfloor f_{min}T + i\kappa^{\text{gwFV}} \rfloor - 1\right\}, \\ \mathcal{I}_i^v = \left\{\lfloor f_{min}T + i\kappa^{\text{gwFV}} \rfloor, \ldots, \lfloor f_{min}T + (i+1)\kappa^{\text{gwFV}} \rfloor - 1\right\})\}_{i=0}^{k-1}, \quad (9)$$

where $\kappa^{\text{gwFV}} = \frac{(1-f_{min})T}{k}$ is the number of observations the window grows by.

Instead of letting the training set grow with each split, one can also use a fixed-size *rolling window* as training data, as, for example, used by Callen et al. (1996) and Swanson and White (1997). Swanson and White (1997) discuss the advantage of rolling-window validation over growing-window validation for econometrics in non-stationary environments, as "the model is allowed to update by discarding older and less relevant observations" (Swanson and White, 1997, p. 444). We define this growing-window validation in blocked form as

$$\mathcal{V}^{\text{rwFV}}(k, f) := \{(\mathcal{I}_i^t = \left\{\lfloor i\kappa^{\text{rwFV}} \rfloor, \ldots, \lfloor fT + i\kappa^{\text{rwFV}} \rfloor - 1\right\}, \\ \mathcal{I}_i^v = \left\{\lfloor fT + i\kappa^{\text{rwFV}} \rfloor, \ldots, \lfloor fT + (i+1)\kappa^{\text{rwFV}} \rfloor - 1\right\})\}_{i=0}^{k-1}, \quad (10)$$

where f specifies the constant fraction of data that is used for training and $\kappa^{\text{rwFV}} = \frac{(1-f)T}{k}$ is the number of observations the window is rolled forward by.

3. Methodology

We evaluate the performance of these validation schemes in extensive Monte Carlo simulation studies, that we describe in the following. Unlike analytical analyses, this allows us to consider a

number of different models and various data-generating time-evolving processes. The goal of our simulations is to compare the estimate error of different validation schemes, i.e., assess how well a given validation scheme estimates the out-of-sample prediction error using available training data only. Our study design is motivated by several works in this area – see, among others, Bergmeir et al. (2018) and Cerqueira et al. (2017).

For all simulation studies, we run 1000 replications, each of which consists of three steps: First, we generate a time series with T observations that serves as the data source for both training and out-of-sample test data. These observations are used to generate features and targets via time-delay embedding. We use the first 80 percent of data as training sample $\mathcal{D}$, and the last 20% are held back as test sample $\mathcal{D}_{test}$. We use data from two sources: We employ synthetic data from a broad range of different locally stationary autoregressive processes that use time-varying parameters to perturb global stationarity, as well as financial data in an extensive real-world application study. The generation of our data sets and the choice of underlying processes are further detailed in Section 3.1. Second, we compute the out-of-sample model loss as ground-truth loss. To this end, we first train a given model f on all training data $\mathcal{D}$ to obtain a parameter estimate $\hat{\theta}$. Then, we generate predictions for the held-out test data $\mathcal{D}_{test}$ and compute the empirical generalization error $\mathcal{L}$ using a loss function ℓ. Our selection of models is described in Section 3.2 and the respective loss functions are outlined in Section 3.3. Third, using training data $\mathcal{D}$ only, and using the same model as in the previous step, we apply different validation schemes $\mathcal{V}$ to obtain the estimated error $\hat{\mathcal{L}}(\mathcal{D}; \mathcal{V})$ using equation 3. Finally, we calculate the error for each of these estimates as $\hat{\mathcal{L}}(\mathcal{D}; \mathcal{V}) - \mathcal{L}$. Values closer to zero are preferred as these indicate better estimates of out-of-sample performance. Finally, we aggregate results from all replications.

3.1. Data generation

3.1.1. Data generating processes

The goal of our study is to analyze the performance of validation methods in situations where global stationarity is perturbed by evolving process dynamics. To this end, we study locally stationary autoregressive processes (Dahlhaus, 2012). Compared to conventional autoregressive processes, these allow for parameters that change slowly over time, rendering them the perfect test bed for the goal of our study: First, autoregressive processes have been extensively used in the literature as data generating processes for related studies (see, for example, Bergmeir and Benitez, 2012; Bergmeir et al., 2018; Cerqueira et al., 2017; Fischer et al., 2018) and are simple yet extensively used in applications. Second, introducing time-dependent parameters as a perturbation of process stationarity to describe slowly changing dynamics is a natural choice. Third, studying synthetic data from locally stationary processes allows us to precisely control the type and strength of perturbation.

Compared to using real-world datasets, we are therefore able to simulate sets of homogeneous times series with similar characteristics.

To formalize the notion of locally stationary processes, consider a time-varying autoregressive processes of order p (tvAR(p)) that follows

$$X_t - \mu_t = \sum_{j=1}^{p} \varphi_{t,j} \cdot (X_{t-j} - \mu_{t-j}) + \sigma_t \varepsilon_t, \tag{11}$$

with ε_t being an independent random variable of zero mean and unit variance. In the formulation above, this process in defined in discrete time $t \in \mathbb{Z}$. Following Dahlhaus (2012), we rewrite the process in rescaled time $u = t/T$ by rescaling all parameter curves to the unit interval, which results in the process equation

$$X_{t,T} - \mu(u) = \sum_{j=1}^{p} \varphi_j(u) \cdot (X_{t-j,T} - \mu(u - j/T)) + \sigma(u)\varepsilon_t. \tag{12}$$

Therein, we use a time-varying parameter vector $\boldsymbol{\varphi} : \mathbb{R} \to \mathbb{R}^p$, time-varying variance $\sigma : \mathbb{R} \to (0, \infty)$ and time-varying mean $\mu : \mathbb{R} \to \mathbb{R}$. Outside the unit interval, we assume that all parameter curves retain the value at the boundary, i.e., $\boldsymbol{\varphi}(u) = \boldsymbol{\varphi}(0)$, $\sigma(u) = \sigma(0)$ and $\mu(u) = \mu(0)$ for $u < 0$ and $\boldsymbol{\varphi}(u) = \boldsymbol{\varphi}(1)$, $\sigma(u) = \sigma(1)$ and $\mu(u) = \mu(1)$ for $u > 1$. We use the concept of rescaled time to study the sample size-dependent performance of different validation methods: In rescaled time, we can keep the functional form of the parameter curves constant, and at the same time vary the number of observations available for every locally stationary process.

Following Bergmeir and Benitez (2012), the order p of the autoregressive process and the initial parameter $\tilde{\boldsymbol{\varphi}}$ are randomly determined prior to each replication, which allows us to obtain average results for a large number of different processes: The order p is drawn with equal probability from $\{1, \ldots, p_{max}\}$. Then, p random roots of the characteristic polynomial r_i are sampled from the interval $[-r_{max}, -1.1] \cup [1.1, r_{max}]$ with equal probability, and the initial autoregressive parameter values $\tilde{\varphi}_i$ are determined. Following Cerqueira et al. (2017), we set p_{max} to 5 and r_{max} to 5.

We perform simulation studies for a broad range of parameter curves yielding different types of data generating processes (DGP) – compare Table 1 for a summary. Every DGP specifies some time-dependence of either the mean $\mu(u)$, the autoregression coefficients $\boldsymbol{\varphi}(u)$ or the variance of the innovations $\sigma^2(u)$. Our selection of DGP is partially motivated by different types of non-stationarities present in financial time series, e.g., cyclical dynamics, a decay of autocorrelation over time or jump-induced regime changes – for an overview of different non-stationary phenomena in finance, see Guegan (2007).

As baselines, we include a noise process with no structure at all (*BASE-NOISE*), as well as a globally stationary autoregressive process (*BASE-AR*). In a first set of DGP, we perturb global stationarity by adding a time-varying mean to the process. The DGP *MEAN-JUMP* adds discrete jumps to the otherwise constant mean, where the mean process is modeled as compound Poisson process (Ross, 1996, p. 82 ff.) with rate λ and jump sizes that are independently normally distributed with variance s. We specify the rate of the Poisson process in terms of the expected number of jumps γ, i.e., $\lambda = \gamma/T$, which ensures that the expected number of jumps is constant when changing sample size. The jump process $J_{\gamma,s}(u)$ is then given by

$$J_{\gamma,s}(u) = \sum_{i=1}^{N(uT)} S_i, \text{ where } S_i \sim \mathcal{N}(0, s) \text{ and } N(t) \sim P\left(\lambda = \gamma/T\right). \tag{13}$$

The variance of this process after T steps, i.e., for $u = 1$, is γs.

The DGP *MEAN-RW* includes a stochastic drift of the mean modeled by a continuous-time Wiener process $W_\beta(u)$ for $u \in [0, 1]$ (Steele, 2001, p. 29 ff.) that assumes Gaussian increments $W_\beta(u+\delta_0) - W_\beta(u) \sim \mathcal{N}(0, \beta\delta_u)$ with strength parameter β. The formulation as a Wiener process in rescaled time u ensures that the variance at $u = 1$ is equal to β independent of the selected T. The related discrete-time random walk has normally distributed increments with variance β/T.

In a second set of non-stationary DGP, we consider processes with time-vaying autoregression coefficients. We include a deterministic linear (*COEF-LIN*) and exponential decay (*COEF-EXP*) with strength parameters α and τ, respectively, as well as a periodic time dependence following a sine curve (*COEF-SINE*). In addition to a fixed strength parameter α, the latter is governed by a phase ϕ and frequency ω, which are held fixed for all u during a single replication. We randomly determine frequency (uniformly sampled from the interval $[1, 5]$) and phase for each replication to rule out a systematic dependence of results on some fixed phase or frequency. In addition to these deterministic autoregression coefficient curves, we consider randomly changing curves: Starting from the initial roots r_i of the autoregressive process, we add realized values of some stochastic process in every time step, and redetermine the coefficients $\boldsymbol{\varphi}(u)$ in every time step t. In case new roots fall into the interval $[-1.1, 1.1]$, we clip values to the boundaries of this interval.[5] The DGP *COEF-RW* lets the roots r_i of the autoregression process evolve according to a Wiener process with increment variance $\beta\delta_u$, i.e., similar to the DGP *MEAN-RW*. The DGP *COEF-JUMP* perturbs the processes' roots by adding discrete jumps according to a compound Poisson process (defined along the lines of equation 13).

[5] Unlike directly changing autoregression coefficients, this procedure ensures stationarity of the process at all times.

DGP name	Parameter curve specification		
	Mean $\mu(u)$	**AR coefficients** $\boldsymbol{\varphi}(u)$	**Variance** $\sigma^2(u)$
BASE-NOISE	0	$\mathbf{0}$	1
BASE-AR	0	$\tilde{\boldsymbol{\varphi}}$	1
MEAN-JUMP	Compound Poisson process $J_{\gamma,s}(u)$ with expected number of jumps $\gamma > 0$ and jump variance $s > 0$	$\tilde{\boldsymbol{\varphi}}$	1
MEAN-RW	Wiener process $W_\beta(u)$ with variance parameter $\beta > 0$	$\tilde{\boldsymbol{\varphi}}$	1
COEF-LIN	0	Linear decay according to $\tilde{\boldsymbol{\varphi}} \cdot (1 - \alpha u)$ with rate $\alpha > 0$	1
COEF-EXP	0	Exponential decay according to $\tilde{\boldsymbol{\varphi}} \cdot e^{-\tau u}$ with constant $\tau > 0$	1
COEF-SINE	0	Periodic curve according to $\tilde{\boldsymbol{\varphi}} \cdot (1 - \alpha + \alpha \sin(\omega u + \phi))$ with strength $\alpha > 0$, random frequency ω and phase ϕ	1
COEF-JUMP	0	AR roots evolving as $r_i^{t+1} = r_i^t + J_{\gamma,s}(u)$ with expected number of jumps $\gamma > 0$ and jump variance $s > 0$	1
COEF-RW	0	AR roots evolving as $r_i^{t+1} = r_i^t + W_\beta(u)$ with variance parameter $\beta > 0$	1
VAR-EXP	0	$\tilde{\boldsymbol{\varphi}}$	Exponential decay according to $e^{-2\tau u}$, with constant $\tau > 0$
VAR-JUMP	0	$\tilde{\boldsymbol{\varphi}}$	Compound Poisson process $1 + J_{\gamma,s}(u)$ with expected number of jumps $\gamma > 0$ and jump variance $s > 0$

Table 1: **Overview of data generating processes (DGP) for the simulation study**. This table specifies the types of locally stationary autoregressive processes used for the Monte Carlo experiments, which all follow the general form $X_{t,T} - \mu(u) = \sum_{j=1}^{p} \varphi_j(u) \cdot (X_{t-j,T} - \mu(u - j/T)) + \sigma(u)\varepsilon_t$. Each DGP uses no more than one non-constant parameter curve: Either the mean $\mu(u)$, the autoregression coefficients $\boldsymbol{\varphi}(u)$ or the process variance $\sigma^2(u)$ are varied as the rescaled time u progresses. All processes use standard normal innovations ε_t. $\tilde{\boldsymbol{\varphi}}$ denotes the initial AR coefficient that is randomly determined prior to each replication.

A third set of non-stationarity DGP introduces time-varying variance of the autoregressive process. The DGP *VOLA-EXP* includes an exponentially decaying variance with decay constant 2τ, which is also considered in Chandler (2010). Similar to previously described jump processes, the DGP *VOLA-JUMP* adds a series of dicrete jumps to the then piecewise constant volatility $\sigma^2(u)$ of the process.

To generate a realization $x_{t,T}$ for a given DGP $X_{t,T}$, we proceed as follows: First, we use i.i.d. standard normal innovations ε_t. Second, a burn-in phase of 500 observations for $u < 0$ and therefore with a constant parameter curve, is used to initialize the process. Third, the process evolves according to its process equation and parameter curves, and a sample of T obervations is generated from the rescaled time interval $u = t/T \in (0, 1]$.
To construct feature vectors, we consider d lags of the realized time series $x_{t,T}$, i.e., take the vectors

$$\boldsymbol{x}_t = (x_{t-1,T}, x_{t-2,T}, \ldots, x_{t-d,T}) \tag{14}$$

as features. We set d to 5 to embed every possible autoregressive model in this specification. We assess validation methods in both a regression and a classification setting, which allows us to check for robustness under changes of the target variable. Additionally, a classification problem is not affected by changes in the scale of the time series. In the regression setting, the target is the value of the time series at time t, i.e.,

$$y_t^r = x_{t,T}. \tag{15}$$

For classification, the target encodes the directional evolution of the time series with respect to its last value as

$$y_t^c = \begin{cases} 1 & \text{if } x_{t,T} \geq x_{t-1,T}, \\ 0 & \text{else.} \end{cases} \tag{16}$$

3.1.2. Financial data

In addition to synthetic data from locally stationary autoregressive processes, we analyze validation schemes for the example of a large-scale financial data set, which allows us to study a set of homogenous time series from a single domain in detail.
We obtain daily total return indices for all stocks that have been part of the S&P 500 index in the time period from January 1990 to October 2015. Data are corrected for dividend payments, corporate actions and stock splits. In every replication of our Monte Carlo study, we randomly create a subset of these data as follows: First, and following a moving block bootstrap scheme (Lahiri, 2003, p. 25 ff.), we sample a continuous block of total return data that comprises 1240

trading days, i.e., approximately 5 calendar years. Second, we randomly choose a sub-universe of 250 stocks out of all constituents of the S&P 500 index at the first day of the moving block.
With this subsample of data, we construct a classification task following Krauss et al. (2017): First, we generate feature vectors: We calculate the stock's simple return in a time window Δt days prior to the respective day, where $\Delta t \in \{\{1, 2, \ldots, 20\} \cup \{40, 60, \ldots, 240\}\}$. The first feature therefore contains yesterday's return, and the last feature the return realized over the past 240 days. The first 240 days of data are considered only for feature generation, and discarded afterwards. Second, we calculate a binary target: We consider a stock's return and compare it to the median return of all stocks in the sample for the same day. We assign a one if the stock's return exceeds the median, i.e., outperforms the market, and zero otherwise. After constructing feature vectors and the target, we are left with 1000 observations for all of the 250 stocks, giving a total data set size of 250000. Observations from all stocks are used jointly to train a single model.

3.2. Prediction models

Validation scheme performance may also depend on the selected prediction model, for example through different learning curves, i.e., the ability to generalize on unseen data depending on the size of available training data. Therefore, we select the following machine learning models to obtain robust results from the simulation study:

- Linear models (LR): As basic linear models, we include a standard linear regression for the regression target, and logistic regression in case of the classification setup (Hastie et al., 2009). In the default implementation, the latter uses L2 regularization with strength 1.0.

- Random forests (RF): We include a random forest model for both the classification and the regression setting (Breiman, 2001). We use a maximum depth of 10 and an ensemble of 100 decision trees.

- Feed-forward artificial neural networks (NN): We use a multilayer perceptron model (Hastie et al., 2009) with one hidden layer of 10 neurons and the rectified linear activation function. The activation function of the single output neuron is the sigmoid function in the classification case and the identity function for the regression setup.

This selection is motivated by the following considerations: First, these models serve as representatives for widespread model classes. The simultaneous use of regression and classification models allows us to obtain results for two very common application scenarios of machine learning models. Second, implementations of these models are readily available in numerous machine learning

libraries, which further enhances the reproducibility of our results. Third, most of these models have been used in similar studies, for example by Cerqueira et al. (2017).

3.3. Error measures

For regression models, we evaluate model performance using the fraction of unexplained variance, $FVU = \sum(y_t^r - \hat{y}_t^r)^2 / \sum(y_t^r - \bar{y}^r)^2$, where $\bar{y}^r$ denotes the arithmetic mean of the true target values $\hat{y}_t^r$. This error measure relates to the coefficient of determination (R^2 score) as $R^2 = 1 - FVU$. Our choice of regression error measure is motivated by three reasons: First, it is commonly used to evaluate regression model performance and easy to interpret. Second, compared to other common error measures such as the (R)MSE, it is scale-independent and allows to compare model performance from different, heterogeneous time series as used in the simulation study. Third, the specific choise of R^2 based on the work of Kvalseth (1985), who compares different forms of the coefficient of determination and concludes that this definition fulfills most of the desired properties. The corresponding loss function is therefore given by the quadratic loss $\ell^{\text{SE}}(\boldsymbol{x}, y^r, \hat{y}^r) = C(y^r - \hat{y}^r)^2$ in which the constant C is given by the denominator of the coefficient of determination.

In the case of classification models, scale-independence is achieved by definition of the target y^c and model performance is measured by the misclassification error (i.e., the inverse accuracy score), corresponding to the 0-1 loss given by $\ell^{\text{0-1}}(\boldsymbol{x}, y^c, \hat{y}^c) = \mathbf{1}\{\hat{y}^c \neq y^c\}$.

4. Empirical results

In this section, we present empirical results from our simulation studies. In a first set of studies (Sections 4.1 and 4.3), we use a fixed sample size and set the perturbation strength of processes such that parameter variations are similar in magnitude. The goal of these first studies is to explore the influence of different types of stationarity perturbation, and to understand whether differences in a validation scheme's performance are driven by bias or variance. Section 4.2 interprets these results in terms of a trade-off between a bias introduced by the processes' evolution and the sample-size-dependent generalization ability of the model. In a subsequent analysis (Section 4.4), we specifically consider the influence of time series length and perturbation strength for selected processes. Finally, Section 4.5 presents results from our real-world application study.

4.1. Overview – constant sample size and constant perturbation strength

The first simulation study targets a large-scale comparison of validation scheme performance for different data generating processes and models with a fixed time series length of $T = 10000$.[6] The

[6] This time series length may be considered a medium length times series. For a single time series at daily frequency, this corresponds to about 27 years of data, which is very reasonable in a number of applications. For financial machine

parametrization of our locally stationarity processes is such that similar perturbation strengths are achieved across processes, whenever applicable. For processes with time-varying autoregression coefficients, we target a relatively small variation in coefficients in the order of 10 percent. Consequently, we set $\alpha = 0.1$ and $\tau = -\ln(0.9)$ for deterministic coefficient curves. For the DGPs *COEF-JUMP* and *COEF-RW*, we set $\gamma = 100$, $s = 0.001$ as well as $\beta = \gamma s = 0.1$, respectively.[7] For processes with changing mean, we set $\gamma = 100$, $s = 0.01$ and $\beta = 0.01$, which both correspond to a root mean square translation distance of 0.1. For the process with decaying variance, we set $\tau = -0.5 \cdot \ln(0.01)$.

As baselines, we consider last-block validation schemes that use the last 10 percent (*LB10*) and the last 30 percent (*LB30*) of the data as validation set. For cross- and forward-validation schemes, we use the same number of splits for comparability and to achieve roughly similar computational costs for all methods. We use $k = 10$ splits, which is a commonly used value (Cerqueira et al., 2017). As cross-validation methods, we consider random cross-validation (*rCV*), blocked cross-validation (*bCV*) and hv-blocked cross-validation (*hbCV*). For the latter, we set the gap size h to the maximum order of the autoregressive process, i.e., $h = 5$. Further, we include rolling-origin forward-validation (*roFV*), rolling-window forward-validation (*rwFV*) and growing-window forward-validation (*gwFV*). All forward-validation schemes require to choose a (minimal) size of the training data. In order to not loose too many training samples when compared to cross-validation methods, we set the (minimum) size of training data to $f_{(min)} = 0.4$.

We discuss most results for the regression case (Table 2), and those from the classification setting only in case of material differences. As a first performance metric, the table lists the mean of the squared estimation error $[\hat{\mathcal{L}}(\mathcal{D}; \mathcal{V}) - \mathcal{L}]^2$, which we denote as $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$. Corresponding results for the classification setting are listed in Table 5 of the Appendix. Further, we compute average ranks for each validation scheme according to the squared estimation error, and perform statistical tests as suggested by Demšar (2006): We apply the Friedman test with the null hypothesis of equal ranks, which we can reject for all cases. We therefore apply the post-hoc Nemenyi test, which rejects the null hypothesis of similar ranks for a pair of validation methods if their average ranks differ by at least a critical difference CD. We visualize average ranks and corresponding critical differences for selected data generating processes in Figures 2, 3 and 5.

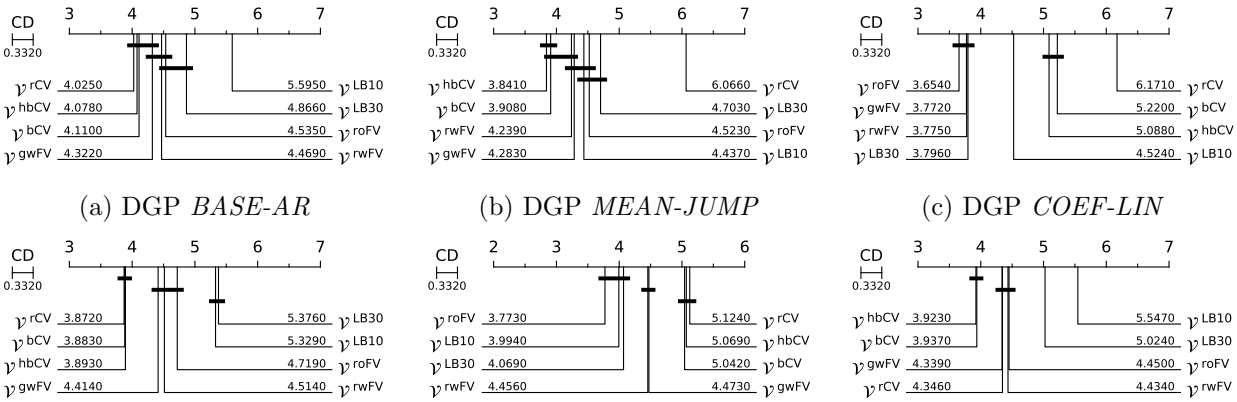
Globally stationary baseline cases – superiority of cross-validation: For the baseline processes

learning, typical sample sizes might even be much larger, consider as an example the case of using returns from 1000 trading days (four years) and 100 stocks as targets.

[7] As the roots of the autoregressive process are chosen from the interval $[1.1, 5]$, the root mean square translation distance $\sqrt{\gamma s} \approx 0.3162$ corresponds to a perturbation in the order of 10%.

		$MSE_{\hat{\mathcal{L}}(\mathcal{V})} \times 10^2$							
Model	**DGP**	$\mathcal{V}^{\text{LB10}}$	$\mathcal{V}^{\text{LB30}}$	$\mathcal{V}^{\text{rCV}}$	$\mathcal{V}^{\text{bCV}}$	$\mathcal{V}^{\text{hbCV}}$	$\mathcal{V}^{\text{roFV}}$	$\mathcal{V}^{\text{rwFV}}$	$\mathcal{V}^{\text{gwFV}}$
LR	*BASE-AR*	0.0874	0.0460	**0.0311**	0.0317	0.0317	0.0415	0.0355	0.0351
	BASE-NOISE	0.0011	0.0005	**0.0004**	**0.0004**	**0.0004**	0.0005	0.0013	0.0009
	MEAN-JUMP	**1.0897**	1.6382	2.6031	1.1309	1.1284	1.5253	1.1136	1.0985
	MEAN-RW	0.9797	1.6437	2.3023	0.9722	0.9698	1.3489	**0.9269**	0.9330
	COEF-LIN	0.1186	0.0875	0.1896	0.1672	0.1672	**0.0850**	0.0940	0.0940
	COEF-EXP	0.1107	0.0775	0.1734	0.1516	0.1516	**0.0738**	0.0831	0.0830
	COEF-SIN	0.4869	0.4278	0.2769	**0.2647**	**0.2647**	0.3386	0.3014	0.3011
	COEF-JUMP	**0.7271**	0.9769	1.7918	1.6197	1.6199	0.8944	1.2555	1.2570
	COEF-RW	**0.9441**	1.4328	2.6053	2.2412	2.2414	1.1964	1.6163	1.5814
	VOLA-EXP	0.1110	0.0602	0.0429	**0.0372**	**0.0372**	0.0473	0.0435	0.0431
	VOLA-JUMP	0.1264	0.0642	**0.0439**	0.0445	0.0445	0.0593	0.0534	0.0531
RF	*BASE-AR*	0.0944	0.0511	**0.0356**	**0.0365**	**0.0365**	0.0476	0.0669	0.0488
	BASE-NOISE	0.0063	0.0047	**0.0026**	**0.0026**	**0.0026**	0.0046	0.0152	0.0058
	MEAN-JUMP	2.9740	4.5452	5.0637	2.9944	2.9862	4.0674	2.8569	**2.8234**
	MEAN-RW	2.1257	3.3833	3.8528	2.1053	2.0979	3.0168	**1.9833**	1.9945
	COEF-LIN	0.1273	0.0831	0.1764	0.1539	0.1535	0.0761	**0.0697**	0.0786
	COEF-EXP	0.1262	0.0784	0.1792	0.1555	0.1550	0.0739	**0.0705**	0.0790
	COEF-SIN	0.4644	0.4319	0.2847	**0.2708**	0.2714	0.3433	0.3417	0.3206
	COEF-JUMP	**0.7439**	0.9382	1.6800	1.5142	1.5154	0.8589	1.2136	1.1979
	COEF-RW	**0.9757**	1.3904	2.3033	1.9565	1.9572	1.1730	1.4440	1.4278
	VOLA-EXP	0.1042	0.0650	0.0611	0.0448	**0.0445**	0.0567	0.0689	0.0499
	VOLA-JUMP	0.1431	0.0748	0.0471	0.0462	**0.0461**	0.0700	0.0795	0.0616
NN	*BASE-AR*	0.0994	0.0526	**0.0357**	0.0365	0.0364	0.0490	0.0478	0.0436
	BASE-NOISE	0.0048	0.0033	**0.0019**	0.0020	**0.0019**	0.0025	0.0066	0.0034
	MEAN-JUMP	1.1686	1.6664	2.5698	1.1269	1.1213	1.5798	1.1201	**1.1133**
	MEAN-RW	0.8782	1.4165	2.3028	0.8308	0.8267	1.1940	0.8057	**0.8052**
	COEF-LIN	0.1221	0.0746	0.1794	0.1562	0.1563	**0.0719**	0.0775	0.0829
	COEF-EXP	0.1204	0.0846	0.1848	0.1613	0.1614	**0.0804**	0.0840	0.0891
	COEF-SIN	0.4465	0.4211	0.2698	0.2576	**0.2570**	0.3291	0.3042	0.2962
	COEF-JUMP	**0.7582**	0.9948	1.7815	1.6005	1.6004	0.9037	1.2408	1.2426
	COEF-RW	**0.9840**	1.4257	2.5093	2.1562	2.1571	1.1991	1.5599	1.5481
	VOLA-EXP	0.1286	0.0791	0.0619	0.0529	**0.0528**	0.0610	0.0559	0.0562
	VOLA-JUMP	0.1365	0.0600	0.0427	**0.0411**	0.0417	0.0539	0.0533	0.0497

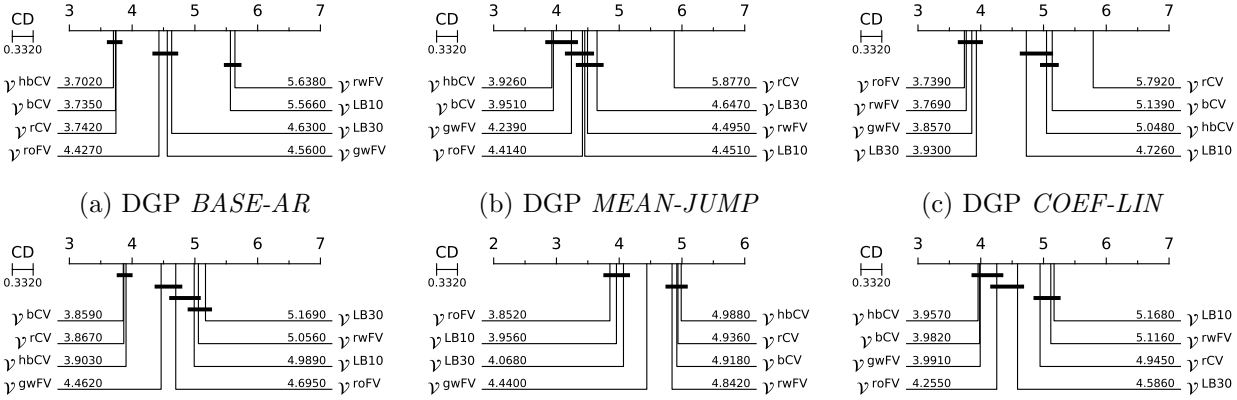
Table 2: **Comparison of mean squared estimate errors for the regression case.** This table shows the mean squared estimate error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$ for different validation schemes and different data generating processes (DGP), multiplied by 100. Higher values indicate a larger deviation between the in-sample validation error estimate and the true out-of-sample error. The maximum (minimum) values of each row are shown in bold (underlined). Results for the linear regression (LR), random forests (RF) and feed-forward neural network (NN) are shown separately. In total, eight validation schemes are listed: Last block validation using the last 10 percent (*LB10*) or 30 percent of the data (*LB30*); cross-validation in the randomized variant (*rCV*) as well as the blocked (*bCV*) and *h*-blocked form (*hbCV*); forward-validation in rolling-origin (*roFV*), rolling-window (*rwFV*) and growing-window (*gwFV*) variants.



(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

Figure 2: **Critical difference (CD) plots for selected data generating processes (DGP) and the linear regression model.** The plots display averaged ranks of mean squared estimate errors. Lower ranks indicate lower estimate errors, i.e., better estimation of the out-of-sample error. Horizontal bars connect validation schemes for which the average rank difference is statistically not significant at the 5% level (post-hoc Nemenyi test).



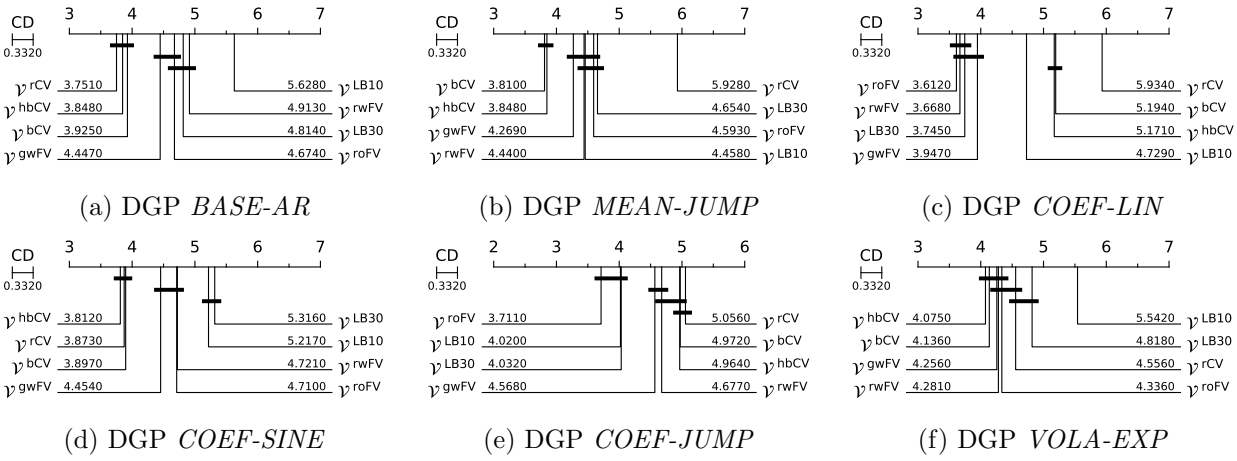
(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

Figure 3: **Critical difference (CD) plots for selected data generating processes (DGP) and the random forest regression model.** Compare Figure 2 for a detailed description.

(*BASE-AR* and *BASE-NOISE*), cross-validation schemes clearly achieve the lowest mean squared error values: For the example of the linear regression model, randomized cross-validation (*rCV*) and modifications (*bCV* and *hbCV*) achieve mean squared errors of 0.0311 and 0.0317, respectively, while the best forward-validation method, growing-window forward-validation (*gwFW*), achieves a value of 0.0351. Differences in squared error ranks between randomized cross-validation and its blocked modifications are not statistically significant.

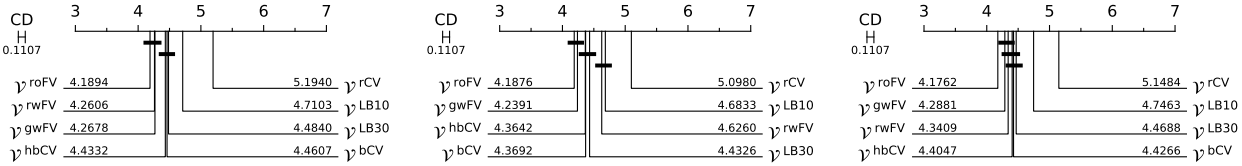
Adding a time-varying mean – breakdown of random cross-validation: When adding a time-varying mean that evolves according to a random walk or a jump process (DGP *MEAN-RW* and *MEAN-JUMP*, respectively), we observe that now randomized cross-validation performs worst: Looking at the average ranks in Figures 2b and 3b, we find that random cross-validation achieves an average



(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

Figure 4: **Critical difference (CD) plots for selected data generating processes (DGP) and the feed-forward neural network regression model.** Compare Figure 2 for a detailed description.



(a) Linear regression (b) Random forest (c) Artificial neural network

Figure 5: **Critical difference (CD) plots for all perturbed data generating processes (DGP).** Perturbed DGP refer to all DGP but *BASE-AR* and *BASE-NOISE*. Compare Figure 2 for a detailed description.

rank of only roughly 6, whereas all other validation schemes (including last-block schemes) come out at average ranks well below 5. With average ranks close to 3.9, blocked forms of cross-validation are least affected by the time-varying mean. Regarding mean squared errors, we make very similar observations also for the DGP *MEAN-RW* (compare Table 2).

Adding time-varying coefficients – advantage of forward-validation: Next, we consider DGPs with time-evolving coefficients, which we further divide into non-periodic (*COEF-LIN*, *COEF-EXP*, *COEF-RW* and *COEF-JUMP*) and periodic parameter curves (*COEF-SINE*).

For non-periodic parameter curves, we make the following observations: First, random cross-validation is the worst cross-validation method in terms of mean squared error. For the example of random forest regression and linearly decaying coefficients (*COEF-LIN*), it achieves a $MSE_{\hat{\mathcal{L}}}$ of 0.1764, which is considerably higher than for blocked cross-validation (0.1539). Second, for all of these DGP and across models, we observe that forward-validation schemes are able to achieve a better $MSE_{\hat{\mathcal{L}}}$ than cross-validation schemes. We find that differences in average ranks between cross-validation and forward-validation methods are statistically significant. Third, the inferiority of last-block validation observed in the baseline cases is no longer given: For the example of

linearly decaying regression coefficients (DGP *COEF-LIN*), last-block validation with 30 percent hold-out data and the linear model achieves a $MSE_{\hat{\mathcal{L}}}$ of 0.0875. The best forward-validation scheme performs only slightly better with an $MSE_{\hat{\mathcal{L}}}$ of 0.0850.

For periodic coefficient curves (DGP *COEF-SINE*), this picture is reversed: The average ranking of validation schemes (Figures 2d, 3d and 4d) and the $MSE_{\hat{\mathcal{L}}}$ (Table 2) are similar to those of the globally stationary baseline case (DGP *BASE-AR*): Cross-validation schemes yield lower $MSE_{\hat{\mathcal{L}}}$ values than both last-block and forward-validation schemes.

When considering an exponentially decaying volatility of the process (DGP *VOLA-EXP*, Figures 2f, 3f and 4f), we observe that the two blocked variants of cross-validation (namely, *hCV* and *hbCV*) yield the lowest average ranks, independent of the model considered. Random cross-validation leads to a statistically significantly higher average rank. The best forward-validation scheme is growing-window forward-validation (*gwFV*), followed by either rolling-origin or rolling-window forward-validation. Differences between the different forward-validation schemes and to blocked cross-validation schemes are not found to be statistically significant. We obtain similar conclusions from the respective $MSE_{\hat{\mathcal{L}}}$ values (Table 2).

Turning to a process with volatility jumps (DGP *VOLA-JUMP*), we find that random cross-validation yields $MSE_{\hat{\mathcal{L}}}$ values very similar to those of blocked cross-validation. Also, forward-validation yields higher $MSE_{\hat{\mathcal{L}}}$ values than cross-validation, independent of the actual scheme used. In most settings with time-dependent volatility, we find that last-block schemes perform worse than cross- or forward-validation.

Differences between machine learning models: To compare validation schemes across models, Figure 5 shows critical difference plots for results from all perturbed DGP and different models. We observe that the ranking of validation schemes is nearly identical for the linear regression and neural network models: Forward-validation schemes lead the ranking, with the rolling-origin scheme achieving the highest average rank. Next are blocked cross-validation schemes and last-block validation with 30 percent validation data. With considerable distance, random cross-validation achieves the last rank. For the random forest model, the ranking is similar with one exception: Rolling-window forward-validation ranks lower than with the other models. We make similar observations in the detailed critical difference plot for the random forest model (e.g., for the DGP *COEF-JUMP*, Figure 3e).

Comparison to the classification setting: Next, we compare results from the regression case to those of the classification case (compare Table 5 as well as Figures 10 and 11 in the Appendix), while keeping the strengths of stationarity perturbations at the same level as before. In general, we find that differences between validation schemes are less pronounced in the classification setting.

Consider the example of linearly decaying autoregression coefficients (DGP *COEF-LIN*), where cross-validation and forward-validation achieve similar mean squared errors across models (Table 5). With a maximum difference in average ranks of 0.38, validation schemes perform very similarly (compare Figure 10c). In comparison, we find rank differences of 2.52 for the regression setup. Also, we observe that the previously observed disadvantage of random cross-validation over its blocked modifications with non-periodic dynamics is not marked, as only relatively small differences in mean squared error can be observed. Overall, a non-stationary perturbation of the process affects the binarized target variable less than a continuous one. This leads to smaller differences between validation schemes in the classification setting, which are most apparent in the relative performance difference between random and blocked cross-validation.

4.2. An interpretation in terms of an approximation of the prediction error

To gain an intuition for these results, we now consider an approximation of the prediction error for an AR(1) model with time-varying coefficient $\varphi(u)$ and time-varying variance $\sigma(u)$. In a simplified regression setting, the model is assumed to be trained on data from the rescaled-time segment $\left[u_t - \frac{b_T}{2},\ u_t + \frac{b_T}{2}\right]$ and validated with respect to the locally stationary process at time u_v. The prediction error, expressed in terms of the R^2 score, can be approximated as

$$\mathcal{L}^{R^2} \approx \underbrace{\varphi(u_v)^2}_{\text{unperturbed score}} - \underbrace{\left(\varphi(u_v) - \varphi(u_t) + \frac{b_T^2}{24}\mu(u_t)\right)^2}_{\text{bias induced by process evolution}} - \underbrace{\frac{1}{b_T T}\left(1 - \varphi(u_t)^2\right)}_{\text{estimation error}}. \tag{17}$$

The derivation of this result can be found in Appendix A, and $\mu(u_t)$ is a term that depends on second derivatives of the local autocovariance function of the process with respect to time, evaluated at u_t. This approximation allows us to identify the major drivers of the prediction error: The first term is the maximum achievable R^2 score that would asymptotically be obtained in the unperturbed case (i.e., for $\varphi(u)$ and $\sigma(u)$ constant). The second driver is a bias term that reduces the maximum score. It is due to the fact that the model is fit to a different process than it is validated on, and depends on the difference in autoregression coefficients at times u_v and u_t. The further away validation data is from training data, the larger this term is likely to become if the process dynamics is non-periodic. However, the evolution-induced bias does not depend on the sample size T. The third term further decreases the maximum score by an estimation error that decreases with the available sample size T.

To further illustrate generalization abilities of different models, Figure 6 compares average empirical prediction errors from the three investigated models for varying sample sizes and different rates α of a linear decay. The dependence of the prediction score on sample size T and decay rate α

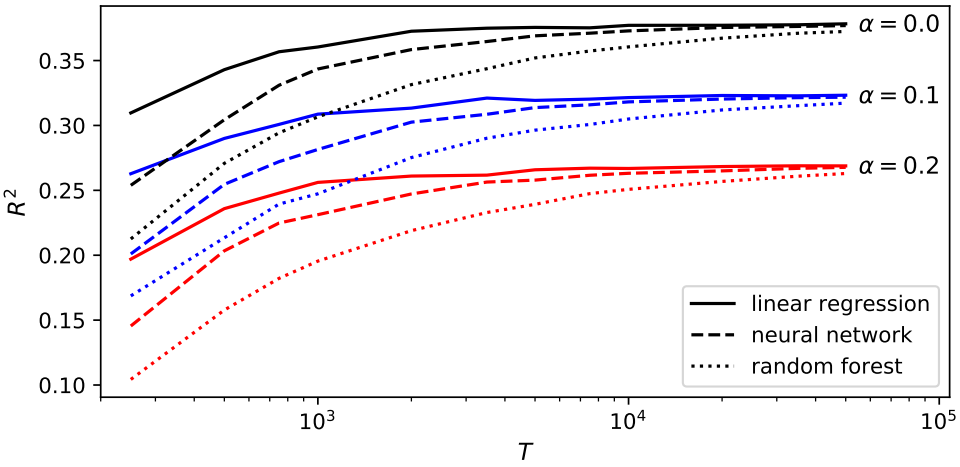


Figure 6: **Generalization ability of different models.** This plot shows the out-of-sample prediction score of the linear, neural network and random forest regression models, expressed in terms of the mean coefficient of determination R^2, for different sample sizes T. The decay rate α of the autoregression coefficient is varied from $\alpha = 0$ (global stationarity) to $\alpha = 0.2$. Results are calculated from 1000 time series, each generated with randomly chosen initial autoregression coefficients.

agrees with the simplified approximation: The larger the sample size T, the closer the prediction score gets to the highest possible value, which is composed of the unperturbed score and the bias term (first and second term in equation 17). This behavior of the estimation error is described by the factor $\frac{1}{T}$ in the last term of equation 17. Although this result is derived from estimating an autoregressive model, we empirically recover similar learning curves for all models. While all models achieve similar scores for large sample sizes, random forest regression profits most from an increase in sample size, i.e., has the steepest learning curve. The bias term is driven by the decay rate α, and reduces the maximum score, which is seen from the approximate parallel shift between the black, blue and red sets of curves.

These results allow us to give a qualitative interpretation of the performance differences in validation schemes in terms of two drivers: First, forward-validation schemes train the model on less observations than cross-validation schemes, which enters the estimation error via the $\frac{1}{T}$-dependence. On the one hand, the bias induced by the process' evolution may be smaller for forward-validation schemes, as validation samples are closer to training data.

Consider first cases where the process evolution is due to changes in the parameter curve $\varphi(u)$: For the non-periodic dynamics, the absolute difference in autoregression parameters $|\varphi(u_v) - \varphi(u_t)|$ tends to increase with the temporal difference $|u_v - u_t|$. Thus, forward-validation may be more applicable as the temporal difference between training and validation samples is smaller, leading to the reduced validation errors observed in case of decaying, random-walking or jumping au-

toregression coefficients. If, however, the underlying dynamics is periodic or the sample size is small, cross-validation methods may be preferred due to their more efficient use of available data. This leads to a smaller estimation error when compared to forward-validation, which offsets the evolution-induced bias from the processes' dynamics.

In cases where instead the variance $\sigma(u)$ is time-dependent, the evolution-induced bias is affected only through the term $\frac{b_T^2}{24}\mu(u_t)$. It does not explicitly depend on the rescaled time u_v of the validation sample, and thus leads to a similar bias contribution when comparing forward- and cross-validation. Consequently, and as observed in the simulation study, mean squared estimate errors from cross-validation are lower than from forward-validation.

4.3. Analysis of bias-variance decomposition

We proceed by splitting the mean squared error into a bias and a variance component given by $MSE_{\hat{\mathcal{L}}(\mathcal{V})} = BIAS^2_{\hat{\mathcal{L}}(\mathcal{V})} + VAR_{\hat{\mathcal{L}}(\mathcal{V})}$, where bias and variance are given by $BIAS_{\hat{\mathcal{L}}(\mathcal{V})} = \mathbb{E}_{\mathcal{D}}[\hat{\mathcal{L}} - \mathcal{L}_{PE}]$ and $VAR_{\hat{\mathcal{L}}(\mathcal{V})} = \mathbb{E}_{\mathcal{D}}[(\hat{\mathcal{L}} - \mathbb{E}[\hat{\mathcal{L}}])^2]$, respectively. If the goal of applying a validation scheme is model assessment, i.e., the estimation of the expected generalization error, a small bias is preferable (cf., Arlot and Celisse, 2010). If instead the goal is model selection, a smaller variance usually leads to a better selection performance (cf., Arlot and Celisse, 2010; Cawley and Talbot, 2010): An estimator of model performance with large, but constant bias still works well in terms of selecting the best model; a large variance of the estimator however may lead to an "overfitting in model selection" (Cawley and Talbot, 2010). We therefore seek to understand whether differences in mean squared estimator performance are dominated by differences in bias or variance.

First, we consider the quotient between squared bias and variance, i.e., the fraction ${BIAS_{\hat{\mathcal{L}}}}^2/VAR_{\hat{\mathcal{L}}}$, to determine the main driver of our results. Results for the linear regression model are presented in Panel A of Table 3; results were found to by fairly independent of the actual model. We observe that in most cases, only a negligible part of the mean squared estimate error can be attributed to the bias term. Exceptions are the DGP *MEAN-JUMP*, *MEAN-RW* and *LIN-DECAY*, for which squared bias and variance are of a similar magnitude.

Next, we analyze the bias component (Panel B of Table 3), which is interpreted as follows: A positive sign of the bias indicates that the error estimated by the validation scheme is larger than the true out-of-sample generalization error, i.e., the error is overestimated, and vice versa. We find a positive bias for the globally stationary base case (*BASE-AR*), which is consistent with expectation, as errors from cross-validation folds are based on a model that is trained on less observations than the final model (Arlot and Celisse, 2010, p. 68). When stationarity is perturbed by time-dependent parameters, the sign of the bias varies with both the type of DGP and the employed validation scheme. The DGP *COEF-LIN* presents one exception as we find that the bias is always negative.

DGP	$\mathcal{V}^{\text{LB10}}$	$\mathcal{V}^{\text{LB30}}$	$\mathcal{V}^{\text{rCV}}$	$\mathcal{V}^{\text{bCV}}$	$\mathcal{V}^{\text{hbCV}}$	$\mathcal{V}^{\text{roFV}}$	$\mathcal{V}^{\text{rwFV}}$	$\mathcal{V}^{\text{gwFV}}$
Panel A	$\boldsymbol{BIAS}_{\hat{\mathcal{L}}(\mathcal{V})}{}^{\mathbf{2}}/\boldsymbol{VAR}_{\hat{\mathcal{L}}(\mathcal{V})}$							
BASE-AR	0.0066	0.0001	0.0004	0.0113	0.0114	0.0013	0.0541	0.0440
MEAN-JUMP	0.0050	0.0045	0.6701	0.0727	0.0711	0.0000	0.0348	0.0110
MEAN-RW	0.0007	0.0011	0.6590	0.0661	0.0645	0.0001	0.0239	0.0061
COEF-LIN	0.0601	0.3185	1.1646	1.0864	1.0860	0.3420	0.4948	0.5116
COEF-SIN	0.0010	0.0022	0.0185	0.0003	0.0003	0.0016	0.0082	0.0061
COEF-JUMP	0.0008	0.0013	0.0213	0.0000	0.0000	0.0009	0.0002	0.0010
COEF-RW	0.0010	0.0009	0.0438	0.0003	0.0003	0.0009	0.0006	0.0001
VOLA-EXP	0.0015	0.0090	0.0130	0.0029	0.0028	0.0045	0.0048	0.0041
VOLA-JUMP	0.0024	0.0013	0.0011	0.0118	0.0118	0.0015	0.0465	0.0379
Panel B	$\boldsymbol{BIAS}_{\hat{\mathcal{L}}(\mathcal{V})} \times \mathbf{10^2}$							
BASE-AR	0.2389	0.0161	0.0365	0.1885	0.1889	0.0734	0.4271	0.3848
MEAN-JUMP	-0.7403	-0.8595	-10.2230	-2.7696	-2.7379	-0.0438	-1.9356	-1.0950
MEAN-RW	-0.2650	-0.4326	-9.5661	-2.4556	-2.4251	0.0916	-1.4703	-0.7521
COEF-LIN	-0.8201	-1.4548	-3.1949	-2.9516	-2.9511	-1.4724	-1.7645	-1.7837
COEF-SIN	0.2198	-0.3090	-0.7104	0.0928	0.0950	-0.2292	0.4959	0.4279
COEF-JUMP	0.2381	-0.3568	-1.9334	0.0677	0.0735	-0.2837	0.1556	0.3541
COEF-RW	0.3099	-0.3595	-3.3082	-0.2675	-0.2574	-0.3338	-0.3198	0.0915
VOLA-EXP	0.1280	-0.2318	-0.2350	-0.1030	-0.1024	-0.1452	0.1447	0.1321
VOLA-JUMP	0.1741	0.0897	0.0679	0.2283	0.2285	0.0937	0.4875	0.4405
Panel C	$\boldsymbol{VAR}_{\hat{\mathcal{L}}(\mathcal{V})} \times \mathbf{10^2}$							
BASE-AR	0.0869	0.0461	0.0312	0.0314	0.0314	0.0415	0.0337	0.0337
MEAN-JUMP	1.0853	1.6325	1.5596	1.0553	1.0545	1.5268	1.0772	1.0876
MEAN-RW	0.9800	1.6435	1.3885	0.9128	0.9119	1.3502	0.9062	0.9283
COEF-LIN	0.1120	0.0664	0.0877	0.0802	0.0802	0.0634	0.0629	0.0622
COEF-SIN	0.4869	0.4272	0.2721	0.2649	0.2649	0.3384	0.2992	0.2995
COEF-JUMP	0.7273	0.9766	1.7562	1.6213	1.6215	0.8945	1.2565	1.2570
COEF-RW	0.9441	1.4329	2.4984	2.2427	2.2430	1.1965	1.6168	1.5829
VOLA-EXP	0.1110	0.0598	0.0424	0.0371	0.0371	0.0472	0.0433	0.0430
VOLA-JUMP	0.1263	0.0642	0.0439	0.0441	0.0441	0.0593	0.0511	0.0512

Table 3: **Bias-variance decomposition.** This table displays the bias-variance ratio $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}{}^2/VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ (Panel A), the bias $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ (Panel B) and the variance $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ (Panel C) for the linear regression model. In total, eight validation schemes are listed: Last block evaluation using the last 10 percent ($LB10$) or 30 percent of the data ($LB30$); cross-validation in the random variant (rCV) as well as the blocked (bCV) and h-blocked form ($hbCV$); forward-validation in rolling-origin ($roFV$), rolling-window ($rwFV$) and growing-window ($gwFV$) variants.

This can be explained by the fact that by construction, autoregression coefficients decline over time, which in turn leads to a decreasing predictability (i.e., the amount of autocorrelation the model can exploit). Hence, the out-of-sample error is always larger than any error calculated in-sample. For more complicated time-dependencies of the autocorrelation coefficients, the achievable predictability varies non-monotonically, which leads to the observed changes in bias sign. For all but one perturbed cases, random cross-validation accumulates the bias with the largest magnitude and with a negative sign, i.e., underestimates the out-of-sample generalization error. When we compare the bias magnitude magnitude for blocked cross-validation variants with forward-validation, we find

no clear prevalence of schemes.

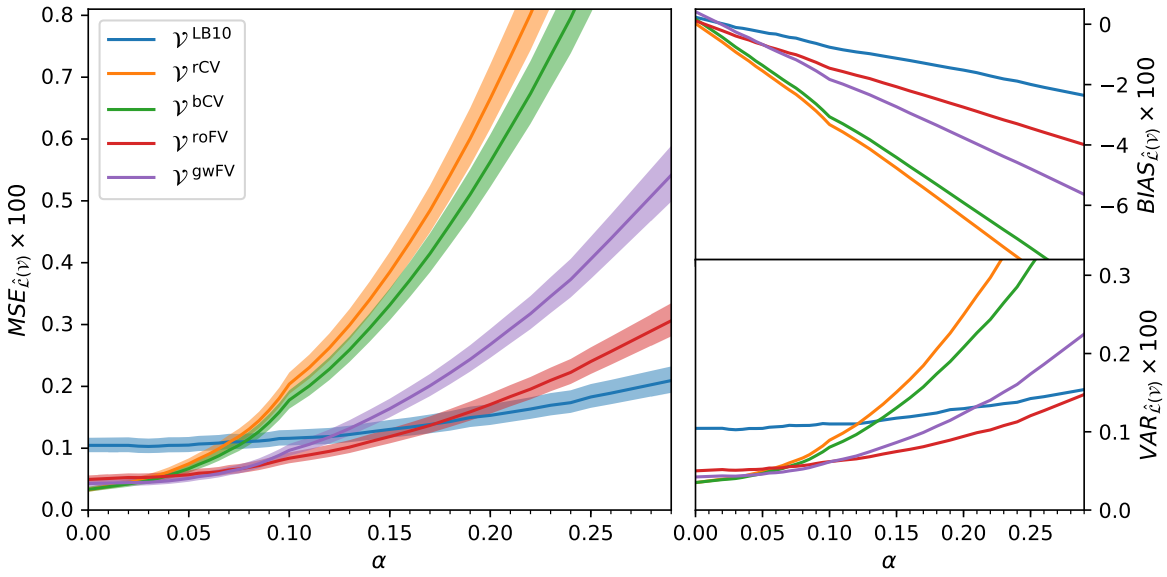
Finally, we consider the variance of the validation scheme (Panel C of Table 3), which in most cases contributes the largest part of the mean squared estimate error. Consequently, differences between validation schemes resemble those observed from the mean squared error. If we further look at cases where the mean squared error is to a large extent driven by the bias component, specifically at the DGP *MEAN-JUMP*, *MEAN-RW* and *COEF-LIN*, we find that previous conclusions also hold true regarding the variance term: Among all cross-validation methods, random cross-validation is affected most severely when global stationarity is perturbed. Also, we find that when non-periodic changes of the autoregression coefficients are present, variance of forward-validation methods are mostly smaller when compared to cross-validation methods.

4.4. Influence of perturbation strength and number of observations

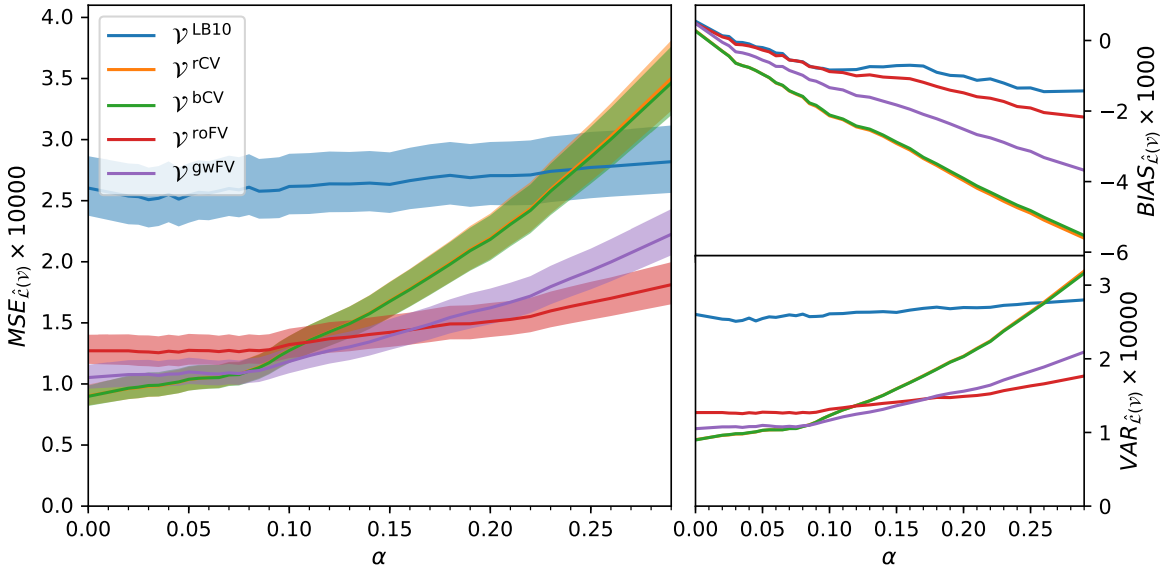
In the following, we analyze the performance of validation schemes for increasing perturbation strengths, and focus on the example of linearly decaying autoregression coefficients for the sake of interpretability. We vary the rate of decay α in the interval $[0, 0.3]$, where $\alpha = 0$ corresponds to a globally stationary process, and a value of $\alpha = 0.3$ means that autoregression coefficients at time $u = 1$ are reduced to 70% of their initial values. For every parameter setting, we run 1000 replications with a sample size of $T = 10000$.

Results for the linear regression model are given in Figure 7a: The left subplot displays the mean squared error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$ of different validation schemes, together with the bootstrapped estimates of the 2.5%- and 97.5%-quantiles. The smaller right plots visualize $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ and $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ components separately. We find that relative performance differences in terms of the mean squared error can be divided into three regimes, depending on the decay rate α: For small α, validation schemes other than last-block validation yield a similar mean squared error. As α increases, mean squared errors for all validation schemes increase, however at different rates. The mean squared error of random cross-validation (rCV) rises fastest, followed by by the blocked variant (bCV). At a critical decay rate $\alpha \approx 0.06$, cross-validation and forward-validation methods yield clearly different values of $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$. In this central regime, forward-validation methods ($gwFV$ and $roFW$) have the lowest mean squared error. As α increases to values above 0.18, last-block validation is the scheme with the lowest error. In terms of the variance, we observe a very similar picture. The magnitude of bias on the other hand increases almost linearly at different rates, with cross-validation again being affected most severely.

When considering the classification case (Figure 7b), we obtain a qualitatively very similar picture. However, the critical perturbation strength is substantially larger ($\alpha \approx 0.15$ compared to $\alpha \approx 0.06$ found in the regression setting), and blocked as well as randomized cross-validation schemes yield



(a) Linear regression model



(b) Logistic regression model used for classification

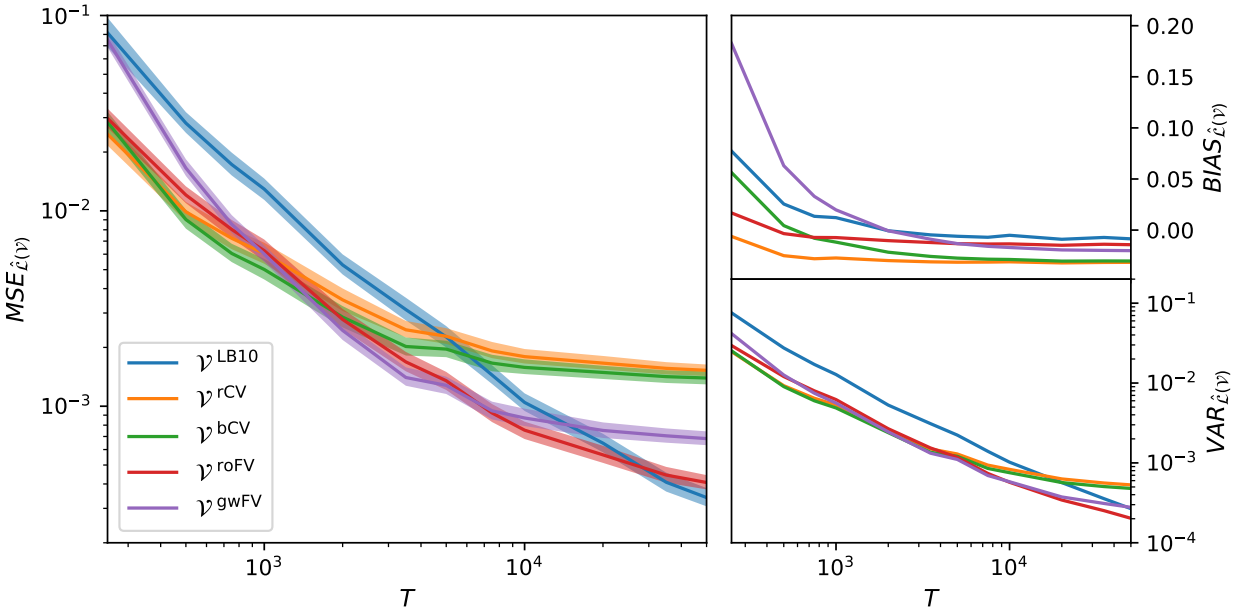
Figure 7: **Performance of validation schemes for increasing perturbation strengths.** The plots display main performance metrics for the DGP *COEF-LIN* and different models when the strength of stationarity perturbation is increased. The strength of stationarity perturbation is expressed in terms of the linear decay rate α, where $\alpha = 0$ corresponds to globally stationary data. For every setting of α, we run 1000 replications with a sample size of $T = 10000$. The mean squared error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$ (left plots), the bias $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ (upper right plots) and the variance $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ (lower right plots) are used as performance metrics. For the mean squared error, filled areas show the interval between bootstrapped 2.5%- and 97.5%-quantiles. Results for six validation schemes are shown: Last block evaluation using the last 10 percent of the data (*LB10*); cross-validation in the random variant (*rCV*) as well as the blocked (*bCV*) form; forward-validation in rolling-origin (*roFV*) and growing-window (*gwFV*) variants.

similar performances. The use of a binary classification target may mediate some of the effects of the stationarity perturbation and thus increase robustness, as only the directional accuracy and not the magnitude of the prediction error is evaluated.

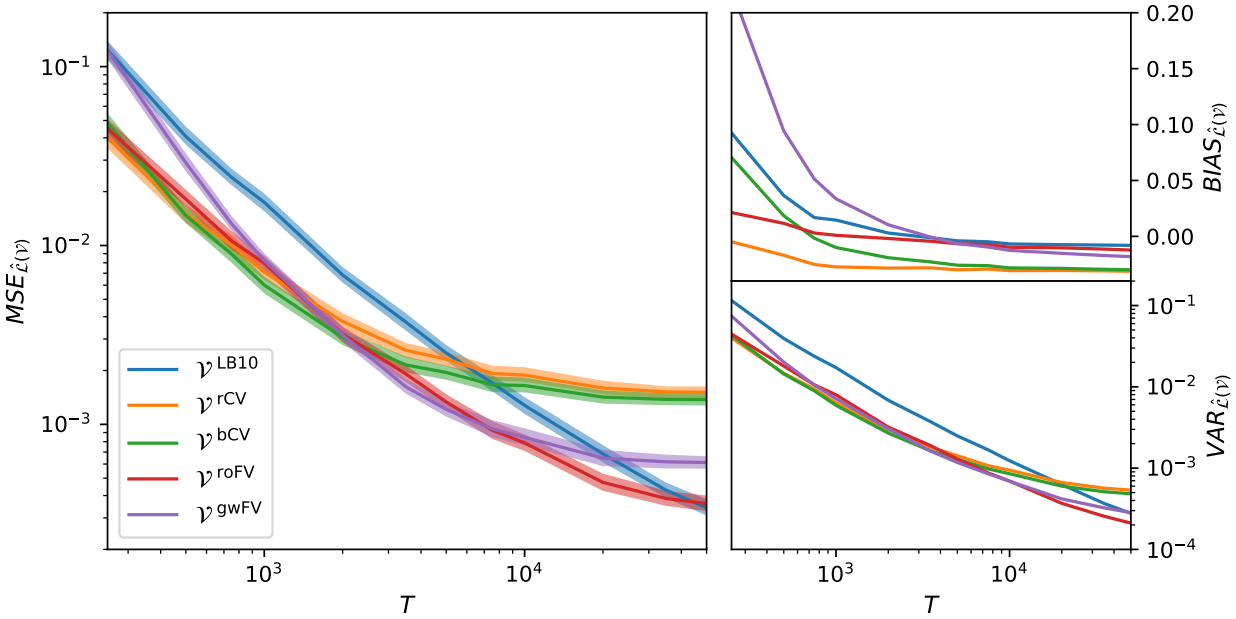
To further analyze the influence of sample size, we vary the time series length T from $T = 250$ to $T = 50000$ (Figure 8). The linear decay rate of autoregression coefficients is fixed at $\alpha = 0.1$, i.e., the autoregression coefficients at $u = 0$ have a value of 90% of their initial value at $u = 1$. Since we formulate this dynamics in rescaled time, increasing the sample size results in more and more observations in the vicinity of any locally stationary process at time u_0; the strength of the perturbation over the unit rescaled-time interval stays however unchanged.
For larger samples sizes, we recover the result that forward-validation schemes perform better, i.e., achieve lower mean squared error values than cross-validation. Asymptotically, the estimation error term in equation 17 tends towards zero as it depends on $\frac{1}{T}$. Despite different generalization abilities, we find little differences in this behavior between linear regression and random forest regression (Subfigures 8a and 8b). The estimate bias induced by the process' evolution dominates, and the mean squared estimate error approaches a constant as this bias term is approximately independent of sample size. For large sample sizes, last-block validation has the lowest $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$: The evolution-induced bias is lowest, as observations from the last block that are used for validation are closest in time to out-of-sample observations, and estimation errors are negligible. However, for samples sizes smaller than 2000, cross-validation and especially blocked cross-validation start to rank highest among all validation methods: Cross-validation gains in performance as the estimation error term becomes more important than the bias induced by the processes' evolution. In this regime, the mean squared prediction error is largely driven by the $\frac{1}{T}$-dependence of the estimation error. When comparing these results to those obtained from using a classification target (Figure 9), we obtain a qualitatively similar picture. However, the perturbation strength was set to a higher value to recover this similarity. Further, growing-window validation does not perform considerably worse than other validation schemes for small sample sizes in the classification setting.

4.5. Application example: Machine learning for statistical arbitrage

To analyze how results from synthetic time series relate to real-world data sets, we now present results from a large-scale simulation study using financial market data from S&P 500 constituents (compare Section 3.1.2). The machine learning models predict whether a certain stock under- or overperforms when compared to the general market. Results are obtained from 1000 moving block bootstrap replications with randomly selected stock universes. Table 4 lists main performance indicators.



(a) Linear regression model



(b) Random forest regression model

Figure 8: **Influence of sample size on validation scheme performance in the regression setting.** These plots show the performance of different validation schemes for samples sizes that change from 250 to 50000, while the linear decay rate of autoregression coefficients is held constant at $\alpha = 0.1$. We use a linear regression as well as a random forest regression model to perform 1000 replications. Plots show the mean squared error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$, the bias $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ and the variance $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ of the estimate error. For the mean squared error, filled areas show the interval between bootstrapped 2.5%- and 97.5%-quantiles. Results for six validation schemes are shown: Last block evaluation using the last 10 percent of the data ($LB10$); cross-validation in the random variant (rCV) as well as the blocked (bCV) form; forward-validation in rolling-origin ($roFV$) and growing-window ($gwFV$) variants.

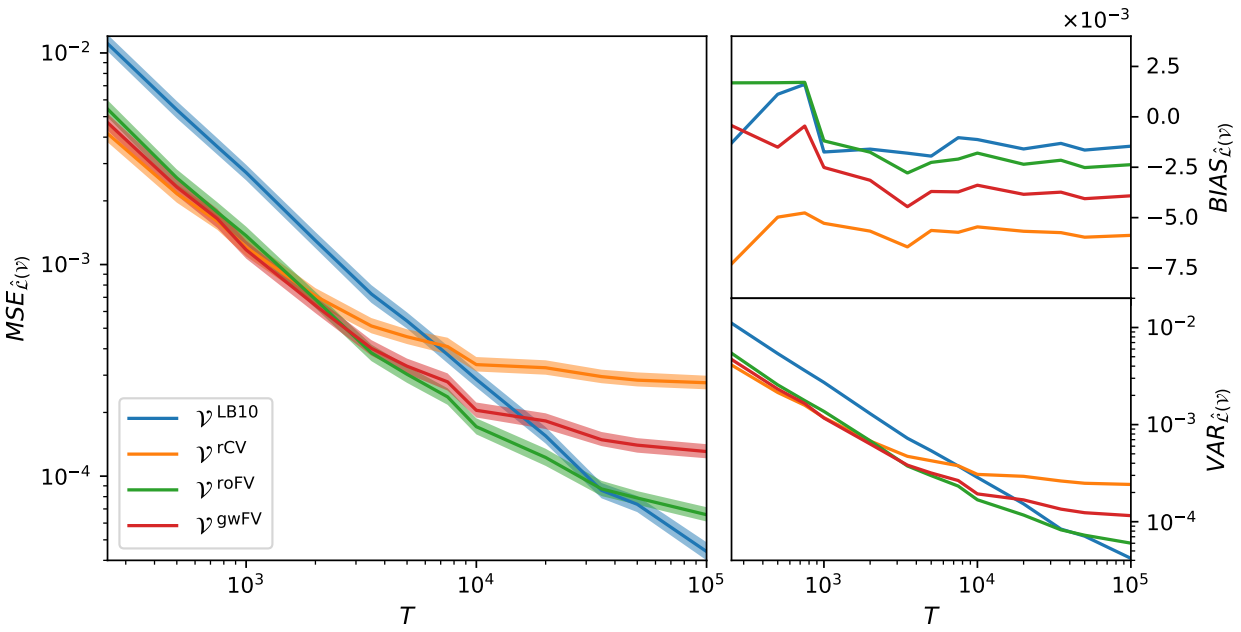


Figure 9: **Influence of sample size on validation scheme performance in the classification setting.** These plots show the performance of different validation schemes for samples sizes that change from 250 to 100000, while the linear decay rate of autoregression coefficients is held constant at $\alpha = 0.25$. We use a logistic regression model to perform 1000 replications. Subplots show the mean squared error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$, the bias $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ and the variance $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ of the estimate error. For the mean squared error, filled areas show the interval between bootstrapped 2.5%- and 97.5%-quantiles. Results for four validation schemes are shown: Last block evaluation using the last 10 percent of the data ($LB10$); random cross-validation (rCV); forward-validation in rolling-origin ($roFV$) and growing-window ($gwFV$) variants.

We observe that, in terms of the mean squared error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$ (Panel A), forward-validation schemes outperform cross-validation and last-block schemes. Across models, random cross-validation (rCV) yields the largest mean squared error, while rolling-origin forward-validation ($roFV$) achieves the lowest values. Looking at the respective bias components (Panel B), we find that randomized cross-validation also accumulates the largest bias. Bias signs are generally negative, i.e., all validation schemes underestimate the out-of-sample error and overestimate the accuracy. Last-block and forward-validation schemes achieve the lowest bias magnitudes. Regarding the variance component $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$, last-block validation with 10 percent validation data ($LB10$) has the highest values. Randomized cross-validation is similar in variance when compared to its blocked variants. Rolling-origin forward-validation achieves the lowest variance values, while other forward-validation variants are similar to cross-validation. Hence, differences in mean squared estimate error between cross- and forward-validation schemes are to a larger extent driven by differences in bias, which is qualitatively similar to observations from the DGP *COEF-LIN*.

This exemplary study illustrates the importance of accounting for time-evolving dynamics in the selection of suitable validation schemes for time-series data. We find that real-world performance differences between validation schemes qualitatively resemble those found with synthetic data when

Model	$\mathcal{V}^{\text{LB10}}$	$\mathcal{V}^{\text{LB30}}$	$\mathcal{V}^{\text{rCV}}$	$\mathcal{V}^{\text{bCV}}$	$\mathcal{V}^{\text{hbCV}}$	$\mathcal{V}^{\text{roFV}}$	$\mathcal{V}^{\text{rwFV}}$	$\mathcal{V}^{\text{gwFV}}$
Panel A	$\boldsymbol{MSE_{\hat{\mathcal{L}}(\mathcal{V})} \times 10^4}$							
LR	0.4689	0.3943	0.5776	0.4479	0.4473	**0.3353**	0.3863	0.3939
RF	0.4185	0.3320	1.0738	0.4616	0.4577	**0.2797**	0.3322	0.3484
NN	0.5019	0.3951	0.5201	0.4325	0.4392	**0.3258**	0.3763	0.3826
Panel B	$\boldsymbol{BIAS_{\hat{\mathcal{L}}(\mathcal{V})} \times 10^2}$							
LR	**-0.1700**	-0.2117	-0.5020	-0.3013	-0.3008	-0.1941	-0.1879	-0.2468
RF	-0.1863	-0.1693	-0.8872	-0.4113	-0.4074	**-0.1656**	-0.1928	-0.2747
NN	-0.1501	-0.1885	-0.4318	-0.2748	-0.2766	-0.1764	**-0.1694**	-0.2255
Panel C	$\boldsymbol{VAR_{\hat{\mathcal{L}}(\mathcal{V})} \times 10^4}$							
LR	0.4404	0.3498	0.3259	0.3574	0.3572	**0.2979**	0.3514	0.3333
RF	0.3842	0.3037	0.2870	0.2927	0.2920	**0.2526**	0.2953	0.2733
NN	0.4799	0.3599	0.3340	0.3573	0.3631	**0.2950**	0.3479	0.3321

Table 4: **Performance of validation schemes in a real-world stock performance classification task.** Panel A shows the mean squared estimate error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$. Panels B and C show the bias $BIAS_{\hat{\mathcal{L}}(\mathcal{V})}$ and variance $VAR_{\hat{\mathcal{L}}(\mathcal{V})}$ components, respectively. 1000 Monte Carlo replications were used. The maximum (minimum) values of each row in terms of magnitude are shown in bold (underlined). In total, eight validation schemes are listed: Last block validation using the last 10 percent ($LB10$) or 30 percent of the data ($LB30$); cross-validation in the standard, randomized variant (rCV) as well as the blocked (bCV) and h-blocked form ($hbCV$); forward-validation in rolling-origin ($roFV$), rolling-window ($rwFV$) and growing-window ($gwFV$) variants.

introducing non-periodic changes of autoregression coefficients. The observed performance differences in this application example are substantial: For instance, the difference between the root mean square error of random cross-validation and rolling-origin validation amounts to 0.5074%. This error is comparably large, as typical binary balanced accuracy scores for stock prediction tasks are only in the order of 54% (Krauss et al., 2017). Generally, the actual dynamics governing the non-stationarity of financial time series data are much more complex than the simplified examples considered in the simulation study, and results are therefore not directly comparable. Nevertheless, results from synthetic data with time-evolving dynamics may help with the selection of validation schemes for real-world applications.

5. Conclusion

This paper comprehensively examines the performance of common cross- and forward-validation methods for model selection and model assessment. In the proposed Monte Carlo study design, we compare the performances using synthetic data from a broad selection of locally stationary processes and a statistical arbitrage application. These data sets have in common that the data-generating process evolves over time, which is a realistic assumption about any real-world process, but unfortunately perturbs global stationarity.

Generally, the choice of a suitable validation scheme depends on a number of factors, among

them the sample size and model used. This is further complicated when considering time-evolving processes, as the type of dynamics plays a major role and already small perturbations of stationarity may derange validation methods to a larger extent. Nevertheless, we can derive the following guidelines from our experiments: Using cross-validation for time-series applications comes at a great risk. While theoretically applicable, we find that random cross-validation often is associated with the largest bias and variance when compared to all other validation schemes. In most cases, blocked variants of cross-validation have a similar or better performance, and should therefore be preferred if cross-validation is to be used. If global stationarity is perturbed by non-periodic changes in autoregression coefficients, we find that forward-validation may be preferred over cross-validation. Within forward-validation schemes, we find that rolling-origin and growing-window schemes often achieve the best performance. A closer look on the effect of the perturbation strength reveals that there exist three performance regimes: For small perturbations, cross- and forward-validation methods perform similarly. For intermediate perturbation strengths, forward-validation performs better. For still higher perturbation strengths, last-block validation performs best.

With the help of simplified expression for the prediction error of a time-varying AR(1) process, we interpret these results in terms of a trade-off between the bias induced by the slow evolution of the process and the estimation error that depends on the amount of a split's training data.

We demonstrate the practical significance of these results with a large-scale Monte Carlo study that performs replications of a statistical arbitrage problem on S&P 500 stock data. Consistent with results from synthetic time series with non-periodic dynamics, we find that forward-validation schemes outperform cross-validation schemes both in terms of bias and variance. Compared to typical accuracies from statistical arbitrage models, differences in the performance of validation schemes are substantial.

References

Allen, D.M., 1974. The relationship between variable selection and data agumentation and a method for prediction. Technometrics 16, 125–127.

Andreou, P.C., Charalambous, C., Martzoukos, S.H., 2008. Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters. European Journal of Operational Research 185, 1415–1433.

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79.

Armstrong, J.S., Grohman, M.C., 1972. A comparative study of methods for long-range market forecasting. Management Science 19, 211–221.

Baliyan, A., Gaurav, K., Mishra, S.K., 2015. A review of short term load forecasting using artificial neural network models. Procedia Computer Science 48, 121–125.

Beheshti-Kashi, S., Karimi, H.R., Thoben, K.D., Lütjen, M., Teucke, M., 2015. A survey on retail sales forecasting and prediction in fashion markets. Systems Science & Control Engineering 3, 154–161.

Bergmeir, C., Benitez, J.M., 2012. On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213.

Bergmeir, C., Costantini, M., Benitez, J.M., 2014. On the usefulness of cross-validation for directional forecast evaluation. Computational Statistics & Data Analysis 76, 132–143.

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

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Burman, P., 1989. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika 76, 503–514.

Burman, P., Chow, E., Nolan, D., 1994. A cross-validatory method for dependent data. Biometrika 81, 351–358.

Callen, J.L., Kwan, C.C.Y., Yip, P.C.Y., Yuan, Y., 1996. Neural network forecasting of quarterly accounting earnings. International Journal of Forecasting 12, 475–482.

Cawley, G.C., Talbot, N.L.C., 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research 11, 2079–2107.

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

Chandler, G., 2010. Order selection for heteroscedastic autoregression: A study on concentration. Statistics & Probability Letters 80, 1904–1910.

Dahlhaus, R., 2012. Locally stationary processes, in: Rao, T.S., Rao, S.S., Rao, C.R. (Eds.), Time Series Analysis: Methods and Applications. Elsevier. volume 30 of *Handbook of Statistics*, pp. 351 – 413.

Dahlhaus, R., Giraitis, L., 1998. On the Optimal Segment Length for Parameter Estimates for Locally Stationary Time Series. Journal of Time Series Analysis 19, 629–655.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Devroye, L., Wagner, T., 1979. Distribution-free performance bounds for potential function rules. IEEE Transactions on Information Theory 25, 601–604.

Fischer, T.G., Krauss, C., Treichel, A., 2018. Machine learning for time series forecasting – a simulation study. FAU Discussion Papers in Economics No. 2/2018.

Geisser, S., 1975. The predictive sample reuse method with applications. Journal of the American Statistical Association 70, 320–328.

Guegan, D., 2007. Global and local stationary modelling in finance: Theory and empirical evidence. Documents de travail du Centre d'Economie de la Sorbonne 2007 53.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction. Springer Series in Statistics. 2nd ed., Springer-Verlag, New York.

Henrique, B.M., Sobreiro, V.A., Kimura, H., 2019. Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications 124, 226–251.

Hong, T., Fan, S., 2016. Probabilistic electric load forecasting: A tutorial review. International Journal of Forecasting 32, 914–938.

Huck, N., 2019. Large data sets and machine learning: Applications to statistical arbitrage. European Journal of Operational Research 278, 330–342.

Kim, K.K., 2003. Financial time series forecasting using support vector machines. Neurocomputing 55, 307–319.

Krauss, C., Do, X.A., Huck, N., 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research 259, 689–702.

Kuznetsov, V., Mohri, M., 2014. Generalization bounds for time series prediction with non-stationary processes, in: Auer, P., Clark, A., Zeugmann, T., Zilles, S. (Eds.), Algorithmic Learning Theory, Springer International Publishing. pp. 260–274.

Kvalseth, T.O., 1985. Cautionary note about R2. The American Statistician 39, 279–285.

Lago, J., De Ridder, F., De Schutter, B., 2018. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Applied Energy 221, 386–405.

Lahiri, S.N., 2003. Resampling methods for dependent data. Springer Series in Statistics, Springer-Verlag, New York.

Larson, S.C., 1931. The shrinkage of the coefficient of multiple correlation. Journal of Educational Psychology 22, 45–55.

Leitch, G., Tanner, J.E., 1991. Economic forecast evaluation: Profits versus the conventional error measures. The American Economic Review 81, 580–590.

Makridakis, S., 1990. Note—sliding simulation: A new approach to time series forecasting. Management Science 36, 505–512.

McDonald, D.J., Shalizi, C.R., Schervish, M., 2011. Generalization error bounds for stationary autoregressive models. arXiv: 1103.0942.

Mittnik, S., Robinzonov, N., Spindler, M., 2015. Stock market volatility: Identifying major drivers and the nature of their impact. Journal of Banking & Finance 58, 1–14.

Panapakidis, I.P., Dagoumas, A.S., 2016. Day-ahead electricity price forecasting via the application of artificial neural network based models. Applied Energy 172, 132–151.

Pesaran, M.H., Timmermann, A., 1995. Predictability of stock returns: Robustness and economic significance. The Journal of Finance 50, 1201–1228.

Plakandaras, V., Gupta, R., Gogas, P., Papadimitriou, T., 2015. Forecasting the U.S. real house price index. Economic Modelling 45, 259–267.

Racine, J., 2000. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. Journal of Econometrics 99, 39–61.

Ross, S.M., 1996. Stochastic processes. 2nd ed., John Wiley & Sons, New York.

Schnaubelt, M., Fischer, T., Krauss, C., 2018. Separating the signal from the noise – financial machine learning for Twitter. FAU Discussion Papers in Economics No. 14/2018.

Shao, J., 1993. Linear model selection by cross-validation. Journal of the American Statistical Association 88, 486–494.

Snijders, T.A.B., 1988. On cross-validation for predictor evaluation in time series, in: Dijkstra, T.K. (Ed.), On Model Uncertainty and its Statistical Implications, Springer Berlin Heidelberg. pp. 56–69.

Steele, J.M., 2001. Stochastic calculus and financial applications. Number 45 in Applications of mathematics, Springer, New York.

Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological) 36, 111–133.

Swanson, N.R., White, H., 1997. Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models. International Journal of Forecasting 13, 439–461.

Tashman, L.J., 2000. Out-of-sample tests of forecasting accuracy: An analysis and review. International Journal of Forecasting 16, 437–450.

Teräsvirta, T., van Dijk, D., Medeiros, M.C., 2005. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. International Journal of Forecasting 21, 755–774.

Thoma, M.A., 1994. Subsample instability and asymmetries in money-income causality. Journal of Econometrics 64, 279–306.

Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting 30, 1030–1081.

Wohlrabe, K., Buchen, T., 2014. Assessing the macroeconomic forecasting performance of boosting: Evidence for the United States, the Euro area and Germany. Journal of Forecasting 33, 231–242.

Zhang, G., Eddy Patuwo, B., Y. Hu, M., 1998. Forecasting with artificial neural networks:: The state of the art. International Journal of Forecasting 14, 35–62.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175.

Zhang, G.P., Patuwo, B.E., Hu, M.Y., 2001. A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers and Operations Research 28, 381–396.

Appendix A Approximation of the expected generalization error

The purpose of this section is to derive an approximation of the expected generalization error from fitting an autoregressive process to an interval of data generated from a time-varying autoregressive process. The expected generalization error is estimated as the out-of-sample error outside of this interval. Consider the first-order time-varying locally stationary autoregressive process with zero mean given in rescaled time u by

$$X_{t,T} = \varphi(u)X_{t-1,T} + \sigma(u)\varepsilon_t, \tag{18}$$

which has the local autocovariance function $c(u,j) = \frac{\sigma(u)^2}{1-\varphi(u)^2}\varphi(u)^{|j|}$. In the following, we assume that we fit a stationary AR(1) process and obtain a parameter estimate $\hat{\varphi}$ on observations from the rescaled-time segment $\left[u_t - \frac{b_T}{2},\ u_t + \frac{b_T}{2}\right]$ that constitues the training data $\mathcal{D}$. We denote this trained model as $\hat{f}$. First, we derive the generalization error $\mathcal{L}_\mathcal{D}$ under the quadratic loss function at some validation time $u_v \notin \left[u_t - \frac{b_T}{2},\ u_t + \frac{b_T}{2}\right]$, assuming that the validation sample follows the local stationary approximation of the process $\tilde{X}_t(u_v)$ at time u_v:

$$\begin{aligned}
\mathcal{L}_\mathcal{D} &= \mathbb{E}_{(X,Y)\sim\tilde{X}_t(u_v)}\left[(Y - \hat{f}(X))^2 \mid \mathcal{D}\right] \\
&= \mathbb{E}\left[(\varphi(u_v)X + \sigma(u_v)\varepsilon - \hat{\varphi}X)^2 \mid \mathcal{D}\right] \\
&= (\varphi(u_v) - \hat{\varphi})^2\,\mathbb{E}\left[X^2\right] + 2\,(\varphi(u_v) - \hat{\varphi})\,\sigma(u_v)\mathbb{E}\left[X\varepsilon\right] + \sigma(u_v)^2\mathbb{E}\left[\varepsilon^2\right] \\
&= (\varphi(u_v) - \hat{\varphi})^2\,\tfrac{\sigma(u_v)^2}{1-\varphi(u_v)^2} + \sigma(u_v)^2,
\end{aligned} \tag{19}$$

Therein, we follow our convention of a time-delay embedding for target and feature values, i.e., $Y = X_t$ and $X = X_{t-1}$, respectively. Calculating the expected generalization error by additionally averaging over all possible realizations of the training data $\mathcal{D}$ yields an expression that is a function of the expected parameter value $\mathbb{E}[\hat{\varphi}(u_t)]$ and its variance $Var[\hat{\varphi}(u_t)]$:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_\mathcal{D}\left[\mathcal{L}_G\right] \\
&= \left((\varphi(u_v) - \mathbb{E}[\hat{\varphi}(u_t)])^2 + Var[\hat{\varphi}(u_t)]\right)\tfrac{\sigma(u_v)^2}{1-\varphi(u_v)^2} + \sigma(u_v)^2
\end{aligned} \tag{20}$$

To further calculate the prediction error, we use the following approximations of $\mathbb{E}[\hat{\varphi}(u_t)]$ and $Var[\hat{\varphi}(u_t)]$ from the segment with center u_t in the limit $b_T^2 + \frac{1}{b_T T} \to 0$ (Dahlhaus, 2012, p. 359):[8]

[8] In the given form of the approximation, we assume that a trivial data taper $h(x) = 1$ is used, simplifying the expression as $d_K = \frac{\int_0^1 h^2(x)(x-0.5)^2dx}{\int_0^1 h^2(x)dx} = \frac{1}{12}$ and $v_K = \frac{\int_0^1 h^4(x)dx}{\left[\int_0^1 h^2(x)dx\right]^2} = 1$.

$$\mathbb{E}[\hat{\varphi}(u_t)] = \varphi(u_t) - \frac{b_T^2}{24}\mu(u_t) + o\left(b_T^2\right) \tag{21}$$

$$Var[\hat{\varphi}(u_t)] = \frac{1}{b_T T}\sigma(u_t)^2 R(u_t)^{-1} + o\left(\frac{1}{b_T T}\right) \tag{22}$$

In these equations, the shorthand notations $R(u) = c(u,0) = \frac{\sigma(u)^2}{1-\varphi(u)^2}$, $r(u) = c(u,1) = R(u)\varphi(u)$ and

$$\mu(u_t) = R(u_t)^{-1}\left[\left(\frac{\partial^2}{\partial u^2}R(u)\right)\varphi(u_t) + \left(\frac{\partial^2}{\partial u^2}r(u)\right)\right]_{u=u_t} \tag{23}$$

are used. In the following, we suppose that Assumptions 2.1 from Dahlhaus and Giraitis (1998) are fulfilled. Specifically, we assume that the third derivatives of the variance $|\delta^3\sigma^2(u)/\partial u^3|$ and of the autoregression parameter $|\partial^3\varphi(u)/\partial u^3|$ are uniformly bounded for $u \in [0,1]$. Further, we assume that $|\varphi(u)| < 1 \;\forall\; u \in [0,1]$ and that $P(X_{t,T} = 0) = 0 \;\forall\; t \in [1,T]$. These assumptions hold for most data generating processes studied in the simulation study, for example linear or exponential decays of the autoregression parameters or the variance. As such, this approximation yields an appropriate framework to study the general form of the generalization error under time-dependent parameter curves, and may provide an intuitive interpretation of results. Substitution of above approximations into equation 20 yields

$$\mathcal{L} = \left[\left(\varphi(u_v) - \varphi(u_t) + \tfrac{b_T^2}{24}\mu(u_t)\right)^2 + \tfrac{1}{b_T T}\tfrac{\sigma(u_t)^2}{R(u_t)}\right] R(u_v) + \sigma(u_v)^2 + o\left(b_T^4\right) + o\left(\tfrac{1}{b_T T}\right). \tag{24}$$

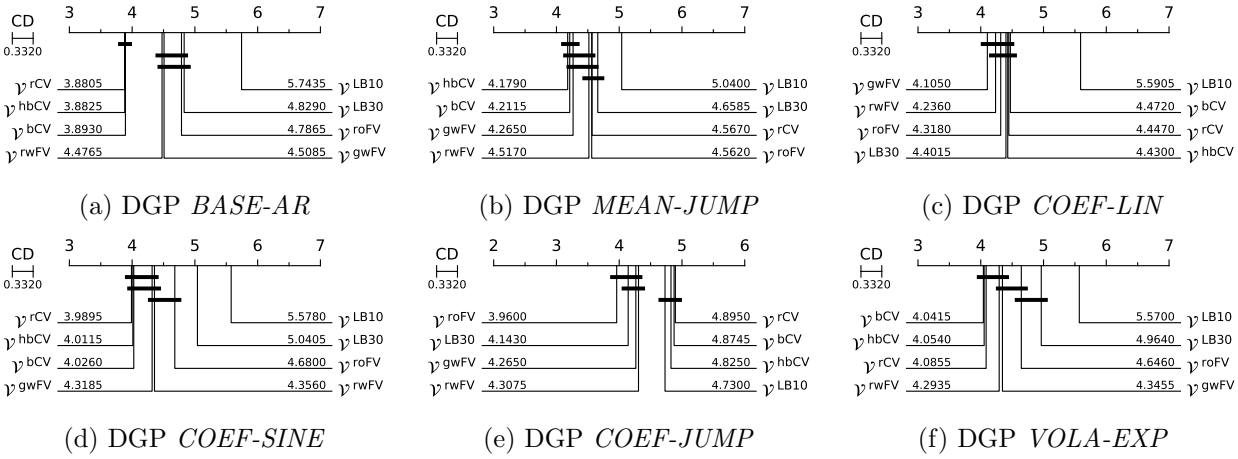
Dividing by the variance at rescaled time u_v and subtracting from 1 yields the expected generalization error in terms of the R^2 score:

$$\mathcal{L}^{R^2} = \varphi(u_v)^2 - \left(\varphi(u_v) - \varphi(u_t) + \tfrac{b_T^2}{24}\mu(u_t)\right)^2 - \tfrac{1}{b_T T}\left(1 - \varphi(u_t)^2\right) + o\left(b_T^4\right) + o\left(\tfrac{1}{b_T T}\right). \tag{25}$$

Appendix B Empirical results for classification models

		$\boldsymbol{MSE}_{\hat{\mathcal{L}}(\mathcal{V})} \times 10^2$							
Model	**DGP**	$\mathcal{V}^{\text{LB10}}$	$\mathcal{V}^{\text{LB30}}$	$\mathcal{V}^{\text{rCV}}$	$\mathcal{V}^{\text{bCV}}$	$\mathcal{V}^{\text{hbCV}}$	$\mathcal{V}^{\text{roFV}}$	$\mathcal{V}^{\text{rwFV}}$	$\mathcal{V}^{\text{gwFV}}$
LR	*BASE-AR*	0.0251	0.0134	0.0093	**0.0092**	**0.0092**	0.0125	0.0107	0.0108
	BASE-NOISE	0.0270	0.0140	**0.0095**	**0.0095**	**0.0095**	0.0132	0.0106	0.0108
	MEAN-JUMP	0.0428	0.0415	0.0398	0.0372	0.0371	0.0384	0.0391	**0.0360**
	MEAN-RW	0.0471	0.0449	0.0379	0.0356	0.0356	0.0416	0.0368	**0.0350**
	COEF-LIN	0.0277	0.0143	0.0136	0.0135	0.0136	0.0138	0.0127	**0.0125**
	COEF-EXP	0.0268	0.0140	0.0118	0.0118	0.0118	0.0129	0.0115	**0.0114**
	COEF-SINE	0.0358	0.0240	**0.0164**	0.0165	0.0165	0.0208	0.0182	0.0178
	COEF-JUMP	0.0567	0.0532	0.0792	0.0791	0.0791	**0.0492**	0.0653	0.0650
	COEF-RW	0.0666	0.0711	0.1082	0.1083	0.1082	**0.0643**	0.0863	0.0854
	VOLA-EXP	0.0266	0.0153	0.0103	**0.0102**	**0.0102**	0.0136	0.0115	0.0115
	VOLA-JUMP	0.0267	0.0139	**0.0094**	0.0095	0.0095	0.0129	0.0110	0.0109
RF	*BASE-AR*	0.0252	0.0136	0.0096	**0.0095**	**0.0095**	0.0122	0.0126	0.0109
	BASE-NOISE	0.0268	0.0138	0.0097	0.0099	**0.0098**	0.0125	0.0122	0.0109
	MEAN-JUMP	0.0803	0.0781	0.0736	0.0682	0.0681	0.0721	0.0642	**0.0636**
	MEAN-RW	0.0689	0.0846	0.0627	0.0573	0.0573	0.0735	0.0550	**0.0543**
	COEF-LIN	0.0262	0.0147	0.0124	0.0125	0.0125	0.0132	0.0137	**0.0121**
	COEF-EXP	0.0263	0.0139	0.0123	0.0121	0.0122	0.0125	0.0136	**0.0117**
	COEF-SINE	0.0373	0.0241	**0.0158**	0.0159	0.0159	0.0206	0.0195	0.0178
	COEF-JUMP	0.0501	0.0530	0.0760	0.0760	0.0761	**0.0473**	0.0645	0.0634
	COEF-RW	0.0626	0.0683	0.1028	0.1028	0.1029	**0.0608**	0.0813	0.0798
	VOLA-EXP	0.0266	0.0178	0.0121	0.0117	**0.0116**	0.0165	0.0130	0.0122
	VOLA-JUMP	0.0292	0.0181	**0.0098**	**0.0098**	0.0099	0.0155	0.0131	0.0116
NN	*BASE-AR*	0.0248	0.0129	**0.0089**	0.0090	**0.0089**	0.0118	0.0100	0.0099
	BASE-NOISE	0.0240	0.0127	0.0089	**0.0088**	0.0089	0.0111	0.0103	0.0099
	MEAN-JUMP	0.0458	0.0430	0.0376	0.0352	0.0353	0.0389	0.0364	**0.0341**
	MEAN-RW	0.0431	0.0418	0.0326	0.0304	0.0305	0.0374	0.0314	**0.0299**
	COEF-LIN	0.0259	0.0146	0.0135	0.0133	0.0133	0.0131	**0.0127**	**0.0127**
	COEF-EXP	0.0269	0.0141	0.0132	0.0130	0.0130	0.0128	**0.0123**	**0.0123**
	COEF-SINE	0.0347	0.0228	**0.0153**	**0.0153**	0.0154	0.0196	0.0178	0.0175
	COEF-JUMP	0.0524	0.0566	0.0794	0.0795	0.0797	**0.0494**	0.0650	0.0654
	COEF-RW	0.0674	0.0714	0.1067	0.1072	0.1077	**0.0640**	0.0821	0.0831
	VOLA-EXP	0.0265	0.0140	**0.0090**	0.0093	0.0093	0.0121	0.0104	0.0102
	VOLA-JUMP	0.0225	0.0141	0.0095	0.0096	**0.0094**	0.0115	0.0112	0.0108

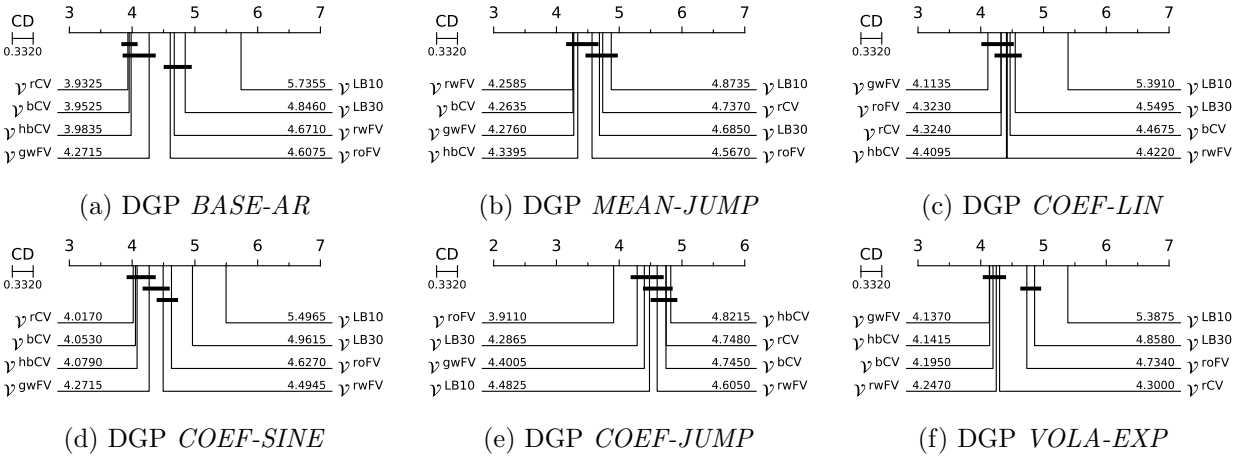
Table 5: **Comparison of mean squared estimate errors for the classification case.** This table shows the mean squared estimate error $MSE_{\hat{\mathcal{L}}(\mathcal{V})}$ for different validation schemes and different data generating processes (DGP), multiplied by 100. Higher values indicate a larger deviation between the in-sample validation error estimate and the true out-of-sample error. The maximum (minimum) values of each row are shown in bold (underlined). Results for the logistic regression (LR), random forests (RF) and feed-forward neural network (NN) are shown separately. In total, eight validation schemes are listed: Last block validation using the last 10 percent (*LB10*) or 30 percent of the data (*LB30*); cross-validation in the randomized variant (*rCV*) as well as the blocked (*bCV*) and h-blocked form (*hbCV*); forward-validation in rolling-origin (*roFV*), rolling-window (*rwFV*) and growing-window (*gwFV*) variants.



(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

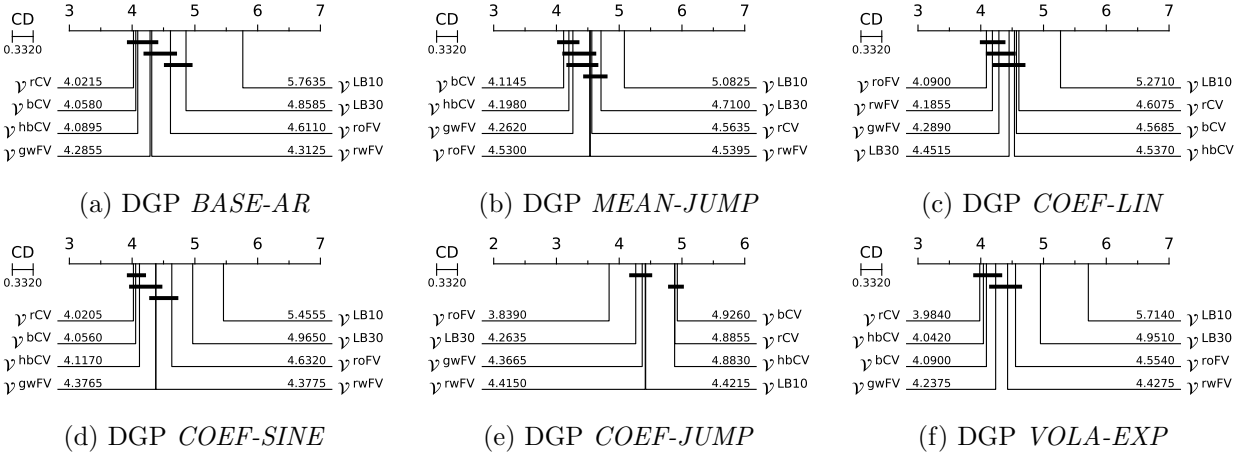
Figure 10: **Critical difference (CD) plots of selected data generating processes (DGP) with the logistic regression classification model.** Compare Figure 2 for a detailed description of the plots.



(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

Figure 11: **Critical difference (CD) plots of selected data generating processes (DGP) with the random forest classification model.** Compare Figure 2 for a detailed description of the plots.



(a) DGP *BASE-AR* (b) DGP *MEAN-JUMP* (c) DGP *COEF-LIN*

(d) DGP *COEF-SINE* (e) DGP *COEF-JUMP* (f) DGP *VOLA-EXP*

Figure 12: **Critical difference (CD) plots of selected data generating processes (DGP) with the random forest classification model.** Compare Figure 2 for a detailed description of the plots.