Information Sciences 191 (2012) 192–213

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

On the use of cross-validation for time series predictor evaluation

Christoph Bergmeir *, José M. Benítez

Department of Computer Science and Artificial Intelligence, E.T.S. de Ingenierías Informática y de Telecomunicación, CITIC-UGR, University of Granada, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 30 November 2010
Received in revised form 24 October 2011
Accepted 28 December 2011
Available online 4 January 2012

Keywords:
Cross-validation
Time series
Predictor evaluation
Error measures
Machine learning
Regression

ABSTRACT

In time series predictor evaluation, we observe that with respect to the model selection procedure there is a gap between evaluation of traditional forecasting procedures, on the one hand, and evaluation of machine learning techniques on the other hand. In traditional forecasting, it is common practice to reserve a part from the end of each time series for testing, and to use the rest of the series for training. Thus it is not made full use of the data, but theoretical problems with respect to temporal evolutionary effects and dependencies within the data as well as practical problems regarding missing values are eliminated. On the other hand, when evaluating machine learning and other regression methods used for time series forecasting, often cross-validation is used for evaluation, paying little attention to the fact that those theoretical problems invalidate the fundamental assumptions of cross-validation. To close this gap and examine the consequences of different model selection procedures in practice, we have developed a rigorous and extensive empirical study. Six different model selection procedures, based on (i) cross-validation and (ii) evaluation using the series' last part, are used to assess the performance of four machine learning and other regression techniques on synthetic and real-world time series. No practical consequences of the theoretical flaws were found during our study, but the use of cross-validation techniques led to a more robust model selection. To make use of the "best of both worlds", we suggest that the use of a blocked form of cross-validation for time series evaluation became the standard procedure, thus using all available information and circumventing the theoretical problems.

1. Introduction

Time series forecasting methods have become indispensable tools in a broad field of applications to understand and forecast, e.g., technical, physical, and economic data. They are nowadays used to make important decisions with far-reaching consequences, and evaluating the procedures' performance is crucial. Assessing an estimate of the error a predictor produces on average can be done for different purposes. Besides the common case, where a general measure of the reliability and accuracy of a system is needed, evaluation often becomes necessary to choose the best one out of different methods and/or parameter sets. Also, researchers proposing a new method are interested in the question, whether the new method performs better than the state-of-the-art methods. This is usually determined by the application and comparison of all the methods on a set of benchmarking data or within competitions. The steps usually involved in such an assessment are (i) computing the methods' results on the test data (ii) calculating the errors of these results with respect to reference data under the use of a loss function, (iii) computing an error measure from the errors, and (iv) using a model selection procedure to examine the distribution of the error measure and/or to find the best model that would be used in the final application.

* Corresponding author.
E-mail addresses: c.bergmeir@decsai.ugr.es (C. Bergmeir), j.m.benitez@decsai.ugr.es (J.M. Benítez).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.12.028

In classification and regression, the standard procedures are to use a quadratic loss function (as a normal distribution of the error is assumed), the root mean squared error as error measure, and cross-validation as the model selection procedure. However, in time series prediction such a consensus does not exist yet. Characteristics of the series such as the amount of observed values, the periodicity, or the complexity of the generating processes can be very different, as well as the types of forecasts needed (one-step-ahead, one-to-24-step-ahead, etc.). With the different nature of series, types of forecasts, and evaluation purposes, not only the methods that perform well and yield accurate forecasts vary, but also the evaluation techniques that guarantee robust, meaningful error estimates. The errors computed should be the ones that are relevant for the intended application, and the measure should be a good summary of the error distribution. With respect to this, important questions are whether the procedures achieve adequacy and diversity. Adequacy means that for each relevant horizon enough forecasts are available, whereas diversity means that the measured error should not depend on special events within the time series [47]. Finally, the model selection procedure should represent and use the distribution of the error measure reasonably.

In traditional forecasting of economic data, the analyzed series mainly represent yearly, quarterly, or monthly acquired data, so that series hardly reach a length longer than a couple of hundreds of values. Furthermore, data generation processes are often complex and poorly represented in the time series itself, so that the amount of information that can potentially be extracted and used for forecasting is limited. Methods widely used are linear methods (made popular by Box and Jenkins [8]) such as autoregression (AR), moving average (MA), or combinations of these (ARMA) Based on these methods, various approaches exist to tackle non-stationarity in the series. A possibility is to use the derivative of the series as input (ARIMA). Or seasonal decomposition procedures, e.g., the STL procedure [16], are used to decompose the series into (deterministic) trend and seasonality, and a stochastic process of the residuals that is stationary.

Regarding non-linear methods, threshold AR models (TAR) are used to partition the (stationary) time series into linear pieces, so that they can be modeled by linear methods. A regime-switching mechanism then decides, which linear model to use [48]. These models have evolved to a host of choices, and combining them with machine learning techniques is getting increasingly popular [6,20,24]. Neural networks are also popular [7,21,52], but as for this type of time series problems complex methods do not necessarily yield better results (this was one of the conclusions of the M3 forecasting competition [36]), their use is not always appropriate.

For evaluation, usually a part at the end of each series is reserved and not used during model generation. This is often called out-of-sample evaluation [47]. To avoid confusion with other evaluation techniques, we will call validation using a set taken from the end of the series *last block* validation.

Another application scenario is the problem a forecaster faces when implementing a concrete application in fields where time series are longer and the underlying processes are better represented, such as in electrical load or price forecasting, traffic-related data or other technical or physical data. Within these applications, the forecaster is typically interested in the forecast of only one concrete time series, and methods that perform well on that particular series. Machine learning techniques and traditional methods seem to outperform each other on time series with different characteristics. Though these characteristics are to the best of our knowledge not well specified so far [21], machine learning techniques usually work well on long series with high-frequency data [7]. So, in this scenario the use of neural networks, machine learning techniques in general and other regression techniques is popular [1,2,13,14,27,39], also as it is often possible to find non-linear patterns in the data, due to the large amount of observed values. The machine learning techniques and other general regression methods used in this context take lagged values of the time series as input to perform an autoregression. With the use of regression methods, also evaluation techniques known from this field are applied to the time series forecasts. Besides the use of statistical measures such as the Akaike and Bayesian information criteria (AIC, BIC), it has become common practice to use cross-validation techniques within regression, and so such methods are also used in the literature to evaluate autoregressions on time series [1,13,39].

We observe that, especially with respect to the model selection procedure, researchers might often be unaware of the advantages and risks of the methods they use. Cross-validation makes full use of the data, i.e., all available data is used both for testing and training. Hence, diversity and adequacy of the evaluation is achieved. But, as during autoregression the same values are used both as part of the input and as reference data, the training set and the test set are not independent if randomly chosen. And the time series might be generated by a process that evolves over time, thus hurting the fundamental assumptions of cross-validation that the data are independent and identically distributed (i.i.d.) [3]. On the other side, besides simulating the real-world application, last block evaluation solves these theoretical problems straightforwardly, and practical problems of missing values during training (as they are reserved for testing) do not arise. But it does not make full use of the data, so that especially in a competition situation, where the validation set is not available to the participants, some problems with respect to adequacy and diversity arise. Only one forecast per series and horizon can be calculated, and the error measure might, rather than being representative, reflect characteristics of the validation set not present neither in the rest of the series nor in future data.

The most important questions regarding this issue seem to be whether the theoretical problems are relevant in practice and therefore standard cross-validation might mislead the user and is applied erroneously in such situations, whether advantages of cross-validation prevail, yielding more robust results, and whether the theoretical shortcomings of cross-validation can be solved in a practical manner. With the aim of gaining further insight into the issue, we present a comprehensive and rigorous empirical study on this, using various cross-validation and last block techniques to evaluate machine learning and general regression methods. Performing an empirical study on this topic is also motivated by the fact, that asymptotic behavior known from theory of the evaluation methods might be quite different from their performance on small test sets [12].

Furthermore, we analyze the different problems of cross-validation in more detail. The problem of dependencies within training and test set can be solved by using blocks of data rather than choosing data randomly. The problem of time evolving effects is closely related to stationarity of the series, so that it can be tackled with known tools for detecting and removing stationarity from the series.

Complete experimental results as well as high-resolution color graphics and complementary information can be found at http://sci2s.ugr.es/dicits/papers/CV-TS.

The remainder of the paper is structured as follows. Section 2 presents traditional methods of evaluation in detail, regarding data splitting in training and test set, and error measures proposed for forecast evaluation. Section 3 details how regression techniques are used for forecasting and how they are evaluated, using cross-validation. Section 4 discusses the design of the experiments that are carried out within our work, and Section 5 shows the results. Section 6 summarizes the conclusions drawn from the paper.

2. Traditional predictor evaluation

The most common approach for data partitioning within traditional forecast evaluation is last block evaluation, as choosing the validation set in this way typically corresponds to the later use case of the system (the continuous forecasting of upcoming values), and the model can be trained and used as in a normal application situation. Furthermore, as we assume that the future depends on the past, the natural dependencies in the data are respected.

However, also within last block evaluation there exist different possibilities for choosing how to use the available data for model building and for evaluation, which are discussed in Section 2.1.

After choice and computation of pairs of forecasts and known reference values, important issues are the choice of a loss function, i.e., how the errors are computed and scaled, and the choice of an error measure that defines which errors are averaged in what way. Literature on this topic is discussed in Section 2.2.

2.1. Data partitioning and forecast horizons

Depending on the length of the last block and the process applied to the individual series, it may be the case that only few forecasts per time series and/or horizon are available (see Section 2.1.1). Adequacy and diversity of the error measure may then be obtained by averaging over different series, or over different horizons, see Section 2.1.2.

2.1.1. Individual series

When evaluating forecasts on individual series, there are mainly four possibilities for training and evaluation, which we name similar to Tashman [47] *fixed-origin*, *rolling-origin-recalibration*, *rolling-origin-update*, and *rolling-window evaluation*. In the following, let the *forecast origin* be the time point of the last known value, from which the forecast is performed. For example, if a daily time series over a certain period, that ends on day t with value x_t, is used to forecast the value x_{t+k} of day $t + k$, the forecast origin is t.

Fixed-origin evaluation is typically applied during forecasting competitions. A forecast for each value present in the test set is computed using only the training set. The forecast origin is fixed to the last point in the training set. So, for each horizon only one forecast can be computed. Obvious drawbacks of this type of evaluation are, that characteristics of the forecast origin might heavily influence evaluation results, and, as only one forecast per horizon is present, averaging is not possible within one series and one horizon.

Within rolling-origin-recalibration evaluation, forecasts for a fixed horizon are performed by sequentially moving values from the test set to the training set, and changing the forecast origin accordingly. For each forecast, the model is recalibrated using all available data in the training set, which often means a complete retraining of the model.

Rolling-origin-update evaluation is probably the normal use case of most applications. Forecasts are computed in analogy to rolling-origin-recalibration evaluation, but values from the test set are not moved to the training set, and no model recalibration is performed. Instead, past values from the test set are used merely to update the input information of the model. Both types of rolling-origin evaluation are often referred to as n-step-ahead evaluation, with n being the forecast horizon used during the evaluation. Tashman [47] argues that model recalibration probably yields better results than updating. But recalibration may be computationally expensive, and within a real-world application, the model typically will be built once by experts, and later it will be used with updated information as new values are available, but it will certainly not be rebuilt.

Rolling-window evaluation is similar to rolling-origin evaluation, but the amount of data used for training is kept constant, so that as new data is available, old data from the beginning of the series is discarded. Rolling-window evaluation is only applicable if the model is rebuilt in every window, and has merely theoretical statistical advantages, that might be noted in practice only if old values tend to disturb model generation.

2.1.2. Different series and horizons

Using forecasts of different horizons to compute an average error raises some problems. Values of different horizons have different statistical properties. With increasing horizon the uncertainty and the variance increases. This issue might be

addressed with a loss function that takes into account the horizon (as defined, e.g., by Kunst and Jumah [35]). But as the relative performance of methods depends on the forecast horizon used [18,36], an average calculated from forecasts of different horizons is potentially misleading.

Time series of different lengths are normally combined by keeping the length of the validation set constant, though depending on the overall procedure this may not be necessary. When averaging over different time series, it should be taken into account that there are very different types of time series. A method might be suited well for a certain type of series, but show weak performance on other types. So, without notions neither of the intended application nor of the time series types present in the evaluation database, averaging over different time series might be misleading. An obvious case of this problem is that time series with different time intervals such as yearly, monthly, and daily data should not be used for the computation of an averaged error measure.

A good test database should contain (according to Tashman [47]) heterogeneous groups of homogeneous time series. Within a homogeneous group adequacy may be achieved, and within the groups, diversity may be achieved. Thus, a good test database could compensate for the shortcomings of fixed-origin evaluation, what is especially important during forecasting competitions.

2.2. Accuracy measures

The purpose of error measures is to obtain a clear and robust summary of the error distribution [15]. It is common practice to calculate error measures by first calculating a loss function (usually eliminating the sign of the single errors) and then computing an average. Let in the following y_t be the observed value at time t, also called the reference value, and let $\hat{y}_t$ be the forecast for y_t. The error E_t is then computed by $y_t - \hat{y}_t$. Hyndman and Koehler [31] give a detailed review of different accuracy measures used in forecasting and classify the measures into these groups:

2.2.1. Scale-dependent measures

Standard error measures, where absolute errors $\mathrm{AE}_t = |y_t - \hat{y}_t|$ or squared errors $\mathrm{SE}_t = (y_t - \hat{y}_t)^2$ are averaged by arithmetic mean M or median MD, leading to the mean absolute error MAE, the median absolute error MDAE, the mean squared error MSE or the root mean squared error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}. \tag{1}$$

As these widely used standard error measures are scale-dependent, they cannot be used to compare and average errors across heterogeneous time series.

2.2.2. Percentage errors

To overcome scale-dependency, the error can be divided by the reference value, thus defining the percentage error:

$$\mathrm{PE}_t = 100\ \frac{y_t - \hat{y}_t}{y_t}. \tag{2}$$

In analogy to scale-dependent measures, the mean absolute percentage error:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|100\ \frac{y_t - \hat{y}_t}{y_t}\right|, \tag{3}$$

the median absolute percentage error MDAPE, the root mean squared percentage error RMSPE, or the root median squared percentage error RMDSPE can be computed.

The main problem regarding these measures is that they have to deal with infinite values if one y_t equals zero, or with undefined values, if $\hat{y}_t = y_t = 0$ for one t. Time series containing zero values are common in many applications, e.g., in return on investment curves, or in traffic delay time series. Also, if the time series consists of real numbers and it is unlikely that exact zeros occur, percentage errors at small values are high (as the denominator gets small), and so the measures show a skewed distribution. By using the median for averaging (MDAPE, RMDSPE), these problems are easier to deal with, as single infinite or undefined values do not necessarily result in an infinite or undefined measure.

The so-called symmetric measures such as sMAPE or sMDAPE try to overcome these shortcomings. However, they are not as symmetric as their names suggest: the symmetry with respect to the interchange of forecast and reference value is obtained at the cost of loosing the symmetry that forecasts only differing in sign should result in the same error [26]. Furthermore, in their original definitions these measures are even able to take negative values; the definition of the sMAPE as it is used in the NNGC1[1] competition circumvents at least the latter shortcoming by using absolute values in the denominator:

$$\mathrm{sMAPE} = \frac{1}{n}\sum_{t=1}^{n} 100\ \frac{|y_t - \hat{y}_t|}{m_t},\ \text{with}\ \ m_t = \frac{|y_t| + |\hat{y}_t|}{2}. \tag{4}$$

[1] http://www.neural-forecasting-competition.com.

But the main problems when dealing with reference values and forecasts close and equal to zero persist.

2.2.3. Relative errors

Another possibility is not to scale using the reference value, but with the help of the error of a benchmark method *B*, where normally the naïve method is used, which uses the last known reference value as forecast. The relative error is defined as:

$$\mathrm{RE}_t = \frac{y_t - \widehat{y_t}}{y_t - \hat{y}_{tB}}, \tag{5}$$

where $\hat{y}_{tB}$ is the forecast for y_t obtained by the benchmark method. Using the RE, e.g., the mean relative absolute error MRAE, or the median relative absolute error MDRAE can be defined. However, general problems of the percentage error measures persist. If the naïve method is used as benchmark, and two subsequent values are zeros, the relative error measures might evaluate to infinity. If in addition the forecast is correct and takes a value of zero, the result is undefined.

2.2.4. Relative measures

Instead of calculating a relative error for each forecast value, averaged error measures can be computed for the forecasting method and a benchmark method. Using, e.g., the MAE, the relative MAE can be defined as:

$$\mathrm{RELMAE} = \frac{\mathrm{MAE}}{\mathrm{MAE}_B}. \tag{6}$$

The RELRMSE with the naïve method as benchmark is also known under the name of Theil's U [31]. Relative measures are able to circumvent many of the problems the other error measures have, as they do not have problems with zeros in the forecasts or the reference values (only if all values would be zero). A shortcoming of these measures is, that they cannot be computed as straightforward over various time series as measures based on percentage errors or relative errors. A second averaging step has to be performed to calculate the average of the relative measures computed on various time series. This is especially a problem, if there is only one forecast per time series and/or horizon available (as it is the case in forecasting competitions). Then, calculating, e.g., the RELMAE for each series with only one forecast and later calculating the mean over this measure across different series results in computing the overall MRAE, with all problems discussed.

Another problem, present in both relative errors and relative measures, is that the use of a benchmark may introduce unexpected or undesired behavior of the measures. The desired behavior of comparing the methods' performance to a benchmark may lead to the behavior that the measure represents rather characteristics of the benchmark. E.g., if there is a negative relation of lag one present in the series, the series tends to short-term oscillations and is less smooth, so that the benchmark will perform badly. Then, low values of the RELMAE in this situation, if compared or averaged with values of a similar series with positive feedback of lag one may be misleading. This shortcoming can be overcome by the usage of a more sophisticated benchmark like an ARIMA model. However, this might induce new problems and complicate error calculation and interpretability in general. Additionally, the performance of the naïve forecast may be sensitive to the horizon used, which might also lead to misinterpretation.

2.2.5. Others

As illustrated, all commonly used error measures have shortcomings, and no commonly accepted measure exists so far that is robust, scale-independent, easy to compute, use, and interpret. There is ongoing research on this topic. Hyndman and Koehler [31] propose to scale errors using the in-sample error of a benchmark method, namely the naïve method. They define the scaled error as

$$\mathrm{SE}_t = \frac{y_t - \hat{y}_t}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}. \tag{7}$$

With the help of this error, e.g., the mean absolute scaled error MASE or the median absolute scaled error MDASE can be computed. The MASE has the advantage, that it has no problems with zeros in the input data in practice and that it can be used in a straightforward way to average across time series. However, its interpretation might be difficult and it uses a benchmark method, which has the shortcomings already discussed.

Chen and Yang [15] present several further normalization approaches, as well as some measures theoretically founded on the Kullback–Leibler divergence. Some of these are further discussed by Kunst and Jumah [35].

3. Cross-validation and regression for time series

Cross-validation is one of the most important tools in the evaluation of regression and classification methods. Its use is outlined in Section 3.1. If general regression techniques are used in time series forecasting (see Section 3.2), cross-validation is often used for evaluation in these applications as well, in spite of the theoretical problems that may arise. On the contrary, in traditional forecasting standard cross-validation receives little attention due to both theoretical and practical problems. Instead, a variety of validation techniques customized for time series can be found in the literature, which are discussed in Section 3.4.

Table 1
Example of a time series preprocessed for regression, using the last four values (lagged values) to predict the current value. Lines in boldface are values that are used for testing within the current iteration of cross-validation, the other ones are used for training.

Index	Lag4	Lag3	Lag2	Lag1	Target
		...	...		
7	14	10	26	11	−13
8	10	26	11	−13	−15
9	**26**	**11**	**−13**	**−15**	**−8**
10	11	−13	−15	−8	35
11	−13	−15	−8	35	40
12	**−15**	**−8**	**35**	**40**	**−8**
13	**−8**	**35**	**40**	**−8**	**−16**
14	**35**	**40**	**−8**	**−16**	**7**
15	40	−8	−16	7	17
		...	...		

3.1. Cross-validation and related methods in regression

In regression and classification, the main concern is usually the generalization ability of the system, i.e., its performance on unseen data. To get a valid estimation of this performance, data used for testing is typically not used for model building.

This raises two problems. Firstly, the data used for testing has to be omitted during training, although the system would probably yield better results if it had been trained with all available data, especially if the amount of data is small. Secondly, in a statistical sense the data available to the researcher is only one possible realization of a stochastic process sample, and so also the acquired error measure is one sample of a stochastic variable that has its possible realizations and probability distribution. This relates directly to adequacy and diversity of the accuracy measure. Depending on the amount of data available, the test set often cannot be chosen large, as the data in the test set cannot be used during training, which decreases adequacy. Diversity is decreased, as the measure computed might represent characteristics only observable in the test set, not in the rest of the data.

To tackle these problems, in classification and regression it is common practice to use *k-fold cross-validation* [3,46], where all available data is randomly partitioned into k sets. Then, the whole training or model fitting procedure, as well as the calculation of the error is performed k times, with every set being once used as the test set, and the other sets being used for model building. So, the method finally acquires k independent realizations of the error measure, and all data is used as well for training as for testing. Averaging the k obtained error measures yields an overall error measure that typically will be more robust than single measures.

In traditional regression and model selection theory, another popular way is to consider complexity of the models, as more complex models are more likely to overfit the data, which yields bad generalization abilities. In this context, a model with not more than the necessary amount of complexity is called a parsimonious model [19]. Model complexity is usually defined by the amount of parameters the model requires. Measures computing this kind of error, i.e., the error of fit with penalizing the amount of parameters are, e.g., the Akaike and Bayesian information criteria (AIC, BIC). Some authors discuss the relation of these measures and cross-validation. Shao [43] demonstrates that AIC and leave-one-out cross-validation (LOOCV) converge and asymptotically show the same behavior. However, Arlot and Celisse [3] argue that in a practical situation cross-validation is applicable to a wide range of problems, so that without knowledge of the data it is likely to yield better results than penalized information criteria.

3.2. Regression for time series prediction

As the name autoregressive (AR) model suggests, AR calculates an estimate for a future value using determined lagged values from the past. So, a regression of the time series on itself is performed. To use standard regression techniques for autoregression, time series must be preprocessed through an embedding step. The lags that are to be used as inputs are identified and a data matrix is built as seen in Table 1.

If autoregression is performed in this way, technically cross-validation can be performed as in normal regression. However, as the embedding procedure may use heavily overlapping parts of the time series (as illustrated in Table 1), the data used for regression is not statistically independent, so that one of the key assumptions of cross-validation is not met [3].

3.3. Stationarity and cross-validation

An important concept in time series research is stationarity, which means that the basic statistics of the time series do not change over time. A series is defined to be stationary (see, e.g., Cryer and Chan [19]), if for any time points $t_1, \ldots, t_n$, and any lag parameter k, the joint distribution of $x_{t_1}, \ldots, x_{t_n}$ and $x_{t_1-k}, \ldots, x_{t_n-k}$ is the same. From this follows especially, with $n = 1$, that the x_t are identically distributed, which is important for our work as this is one of the assumptions of cross-validation.

A related but weaker definition, sometimes referred to as second-order stationarity, is to define a series as (second-order) stationary if the mean remains constant throughout the series, and the autocorrelation of two values only depends on the

relative position of the values (the lag) within the series. From this definition follows that all values of the series have the same mean and variance. If all the joint distributions are Gaussian, the two concepts of stationarity are identical [19].

Non-stationarity has to be taken into account throughout the whole modeling process, not only during model selection. Depending on the type of stationarity, it can be easily removed by a preprocessing step such as differentiation (as done in ARIMA models) or a procedure that removes trend and seasonality. Also, with the Dickey–Fuller unit root test [42], the series can be checked for stationarity. If non-stationarity cannot be removed by such a preprocessing step, the model building procedure may require a processing step that determines, which parts of the series to include in the modeling, as proposed by Deco et al. [22], or prediction of the series might even be an impossible task [22,33].

Furthermore, for non-stationary series last block evaluation might be misleading as well, as the block chosen for testing might be very different from the training data, and the unknown future may also be different from the training data, the test data, or from both of these. Following Inoue and Kilian [32], and also Kunst [34], we could argue that in time series forecasting, the last block of a series might be the most relevant one, being probably most related to the data to predict. However, cases are easily imaginable where this is not the case, and if the last block in fact is the most important part, we suggest that it would be more appropriate to take this information into account while building the model (e.g., by weighting), and not just for its evaluation. Because of these difficulties it is common practice in time series forecasting to assume stationary time series.

Also, w.r.t. the application of cross-validation, the problem of dependent data can be dealt with more easily in the framework of stationarity. As the autocorrelation function only depends on the lags, it can be analyzed as a function of these. In a stationary series, often the number of lags with significant autocorrelation is small [19]. So we can assume that there is a constant h such that x_i and x_j are approximately independent, if $|i-j| > h$ [3,37,41].

It is worth noting, that actually the method outlined in Section 3.2 also is motivated by a stationarity assumption, as choosing particular lagged values as inputs for the forecasting procedure would not make sense, if their dependence on the value that is to be forecast would continuously change. So, if we assume that the model was built taking into account all lagged values with relevant correlations, the order of the model gives a good estimation on the number of values that are dependent and should therefore be omitted during cross-validation.

3.4. Cross-validation and related methods for time series

In addition to the theoretical problems of unmet basic assumptions, using cross-validation for the evaluation of traditional forecasting procedures leads to practical problems. As the partitions are chosen randomly, missing values have to be taken into account during model construction. Depending on the forecasting method, this can be a straightforward or a very difficult task. E.g., if the autocorrelation function is used during the model construction process, missing values might skew that function and eventually yield bad results.

Also, practical consequences of the theoretical problems have been observed in some cases in the literature, which will be discussed in the following. And various evaluation methods especially for time series have been proposed.

3.4.1. Methods based on the last block

As evaluation techniques based on the last block circumvent the discussed problems, various authors describe evaluation techniques based on the last block. Hjorth [29,30] proposes a procedure called "forward validation", which basically computes a weighted sum of one-step-ahead forecasts by the rolling-origin-recalibration procedure. The weights are normalized and depend on the number of parameters and the forecast origin: As the model is recalibrated, depending on the forecast origin more or less data is available for model construction. Finally, the method yielding the minimal error in forward validation is chosen. Additionally, the author presents an estimation for the bias that might be introduced by the model selection procedure in the error measure. Therefore, the forward validation procedure is performed using the chosen model subsequently on subseries of the last block, so that a set of error samples is obtained. The difference between the error measure of the last block and the mean of these subsequently calculated error measures can then be used as an estimate for the bias introduced by the model selection procedure. Wagenmakers et al. [50] use the so-called accumulative prediction error (APE) for model selection, which is the sum of one-step-ahead forecasts computed by rolling-origin-recalibration evaluation. The authors argue that this process has a strong theoretical justification by being related to the AIC and BIC. The advantage using the APE instead of AIC or BIC is according to them, that the APE is not only sensitive to the number of parameters, but also to their functional form. Both the forward validation method and the APE use rolling-origin-recalibration evaluation.

Inoue and Kilian [32] show, that information criteria such as AIC or BIC asymptotically perform better than evaluation on the last block of the series. They also admit the shortcoming that evaluation with the last block only uses a part of the information available, and so looses potentially important information.

3.4.2. Cross-validation with omission of dependent data

A brief review for methods of this type of cross-validation is given by Arlot and Celisse [3]. To solve the theoretical problems of cross-validation with correlated data, stationarity is assumed, and not only data that is used for testing is removed from the training set, but also data that is not independent from the data that is used for testing. The procedure is described, e.g., by McQuarrie and Tsai [37], and is often called modified cross-validation (MCV) [3]. We call it in the following *non-dependent cross-validation*. Burman et al. [9] present a related approach which they call h-block cross-validation. They

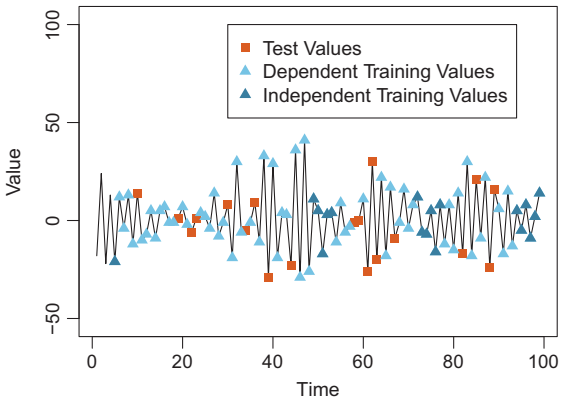

Fig. 1. Example of values that are used for training and test within one iteration of a 5-fold cross-validation procedure. Assuming a forecast procedure that uses the last four values to predict the next one, for every point in the test set values in a radius of four cannot be used for training, if independence has to be achieved. This leads to areas, in which nearly all points have to be excluded from training.

perform LOOCV, and remove the values correlated with the test values in both axis directions from the training set. Hart [28] presents a procedure he names time series cross-validation (TSCV), which simultaneously estimates an optimal bandwidth and the autocorrelation function of the data. Kunst [34] proposes the use of cross-validation with removed dependent values with a subsequent bootstrapping procedure. A related field is non-parametric regression with correlated errors. As in this field, no autoregression is performed and the series are not necessarily stationary, it can be seen as a preprocessing step for trend estimation within time series problems. Opsomer et al. [38] show that standard cross-validation chooses smaller bandwidths in a kernel estimator regression framework if autocorrelation of the error is high, so that the method overfits the data. They state, that depending on the regression method and the parameters to choose, short-range dependence has weak influence, whereas long-range dependence often has a high influence on the model selection process. Carmack et al. [10] present a method they call far casting cross-validation (FCCV), which is similar to h-block cross-validation, and tackles especially the problem of multivariate data by defining a neighborhood radius of dependent data to remove. They also present results with their method in a bandwidth selection framework for non-parametric regression, and show that LOOCV underestimates the error in certain cases.

Depending on the amount of lags used and the number of folds during cross-validation, omission of dependent values can lead to a significant loss of data or even to the removal of all data available for training (see Fig. 1). So, non-dependent cross-validation methods are only applicable in certain cases, where folds contain a low percentage of the overall data, or the amount of relevant lags is small. It has to be noted, that LOOCV is not only computationally costly, but in contrast to k-fold cross-validation it is also asymptotically not consistent (which would mean that with the amount of available data going to infinity, the probability of selecting the best model goes to one) [44].

3.4.3. Cross-validation with blocked subsets

Snijders [45] uses cross-validation with "non-interrupted time series as validating sub-samples". We call this type of cross-validation in the following *blocked cross-validation*. In particular, the last block can be one of these validating sub-samples. In that early study, the author compares blocked cross-validation with last block evaluation, using basic linear forecasting methods. As there is no clear tendency in the results, the use of last block evaluation is suggested as it is less costly to compute: last block evaluation only involves one training and evaluation step, whereas cross-validation involves these steps for every sub-sample. Racine [41] presents a method named *hv*-block cross-validation. It extents the h-block method in the way that not a single value is used for testing, but a block of data of size v. The author points out, that the method is asymptotically consistent for general stationary processes. In his experimental study that focuses on model selection capabilities and does not explicitly state error values, he shows that *hv*-blocked cross-validation yields better model selection performance than v-blocked cross-validation (where $h = 0$, so that no dependent values are omitted), if the series are large, i.e., longer than 500 values; on series with 5000 values his method achieves 10% higher probability for choosing the correct model. However, in practice such long series are often not available, and the task is not the choice of a true model (often there is no true underlying model at all), but the determination of a model that yields good forecasting performance. In particular, Kunst [34] showed that the model with the best forecasting performance is not always the true model, i.e., the model that generated the data.

4. Design of the experiments

As seen in Section 3.4, many authors state the theoretical problems when cross-validation is to be used for time series prediction, and many methods have been proposed to circumvent these problems. Furthermore, some results on synthetic

Table 2
The parameter grid that is generated for the neural network method, which has two parameters, size and decay. The size is chosen by the model selection procedures from {3, 5, 9}, and the decay from {0.00316, 0.0147, 0.1}.

	Size	Decay
1	3	0.00316
2	5	0.00316
3	9	0.00316
4	3	0.01470
5	5	0.01470
6	9	0.01470
7	3	0.10000
8	5	0.10000
9	9	0.10000

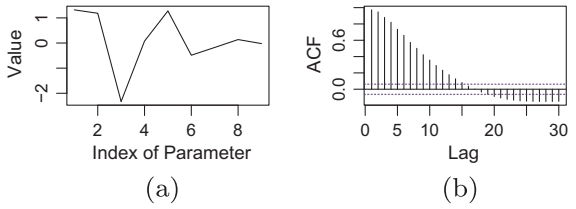

Fig. 2. (a) Generated parameters to simulate an AR or MA model, respectively. In order to obtain a stationary (AR) or invertible (MA) process, the coefficients have to fulfill certain constraints that are respected within our generation process, e.g., the sum of all coefficients has to be smaller than one and the absolute value of the last coefficient has to be smaller than one [19]. Higher coefficients tend to be small. (b) Autocorrelation function of a time series generated using the AR model with the parameters from (a).

and real-world data within bandwidth selection in regression suggest, that standard cross-validation might favor overfitting methods.

But it remains unclear, if these problems have significant consequences in practice when using machine learning techniques in real-world applications. This question is important, as in such a scenario often standard cross-validation is used for evaluation. Another question is, if standard cross-validation could be replaced by another, theoretically better-founded method (last block, blocked cross-validation, etc.), that is equally robust and easy in its use, yielding the same quality of results.

To address those issues we have developed a thorough experimental study with the following objectives:

- To determine if dependency within the data has effects on the cross-validation, e.g., in the way, that the cross-validation procedure systematically underestimates the error. This can be done by comparing randomly chosen evaluation sets to blocked sets.
- To determine if effects of temporal evolution can be found, by comparing evaluations that use data from the end of the series to evaluations that use data from somewhere in between. It has to be noted, that we will consider in this study only (second-order) stationary series, which is common practice in time series forecasting, as stated in Section 3.3.
- To determine if cross-validation yields a more robust error measure, by making full use of the data in the way that it uses all data for training and testing.

In order to cover a broad amount of application situations, the experiments on different model selection procedures will be carried out using machine learning and general regression methods, synthetic and real-world datasets, and various error measures.

4.1. Applied models and algorithms

We have considered a selection of representative machine learning and general regression methods: a support vector regression, a multi-layer perceptron, a linear fitting method, and lasso regression [23]. All methods used are available in packages within the statistical computing language R [40]. We use the implementation of an epsilon support vector regression algorithm with a radial kernel from the LIBSVM [11] (that is wrapped in R by the package `e1071`), and employ a multi-layer perceptron of the `nnet` package [49]. Furthermore, we use lasso regression and the linear fit model present in the R base package. The latter one is a traditional (linear) forecasting method, i.e., an AR model with fixed order that can be applied in the same way as the other regression methods (without the potential need for treatment of missing values during order determination). In the following, the methods will be called `svmRadial`, `nnet`, `lasso`, and `lm`.

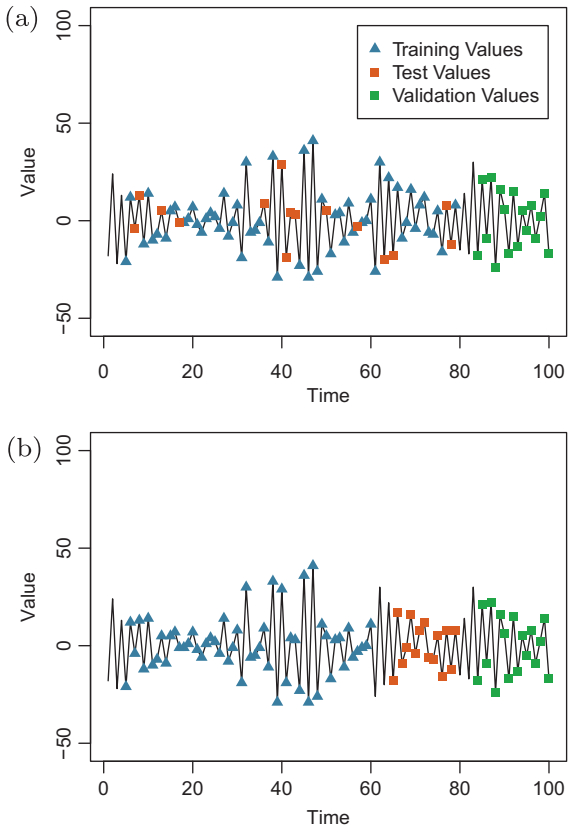

Fig. 3. Within the study, a validation set is withhold from model building and model selection: the out-set. From the other values, the in-set, values are chosen for training and testing according to the model selection procedure, for example (a) one set of standard cross-validation, and (b) last block evaluation. Due to the embedding, values from the very beginning of the series cannot be used as targets, and to make the validation independent of the in-set, the first values of the out-set are omitted as well.

All methods are applied with different parameter configurations. Therefore, for each method a parameter grid is determined empirically (on time series that are available in R, but not used throughout our study, e.g., the "canadian lynx" dataset), which is fixed throughout all of the experiments. The model selection procedures choose for each model and time series the best parameter combination from the grid. The `nnet` has two parameters, size and decay. The model selection procedures choose the size from {3, 5, 9}, and the decay from {0.00316, 0.0147, 0.1}. As an example, the grid is shown in Table 2. The `svmRadial` method has two parameters, cost and gamma, which we defined to be chosen from {0.1, 1, 10, 100, 1000}, and {0.0001, 0.001, 0.01, 0.2}, respectively. The `lasso` has one parameter, fraction, that was chosen from {0.10, 0.36, 0.63, 0.90}. The linear model `lm` has no free parameters to be determined during model selection.

4.2. Benchmarking data

Both synthetic and real-world data were used throughout our study, in order to analyze the evaluation methods' behavior under controlled conditions and in real-world application scenarios. Using the data, three use cases were defined for the experiments, see Section 4.2.3. All data is made available in the KEEL-dataset repository.[2]

4.2.1. Synthetic data

Linear and non-linear time series are simulated for the study. Linear series are generated by an ARMA process:

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_k y_{t-k} + w_t - \beta_1 w_{t-1} - \beta_2 w_{t-2} - \cdots - \beta_l w_{t-l}, \tag{8}$$

where l and k are the numbers of lags that influence the current value, and $w_t, \ldots, w_{t-l}$ are i.i.d. Gaussian distributed random variables. In order to obtain a process that has a stationary AR part and an invertible MA part, the coefficients $\alpha_1, \ldots, \alpha_k$, and $\beta_1, \ldots, \beta_l$, have to be chosen in a way that the roots of their characteristic polynomials have an absolute value greater than one, respectively [19]. In analogy to model fitting, where the final model usually is checked by unit root tests for stationarity

[2] http://sci2s.ugr.es/keel/timeseries.php.

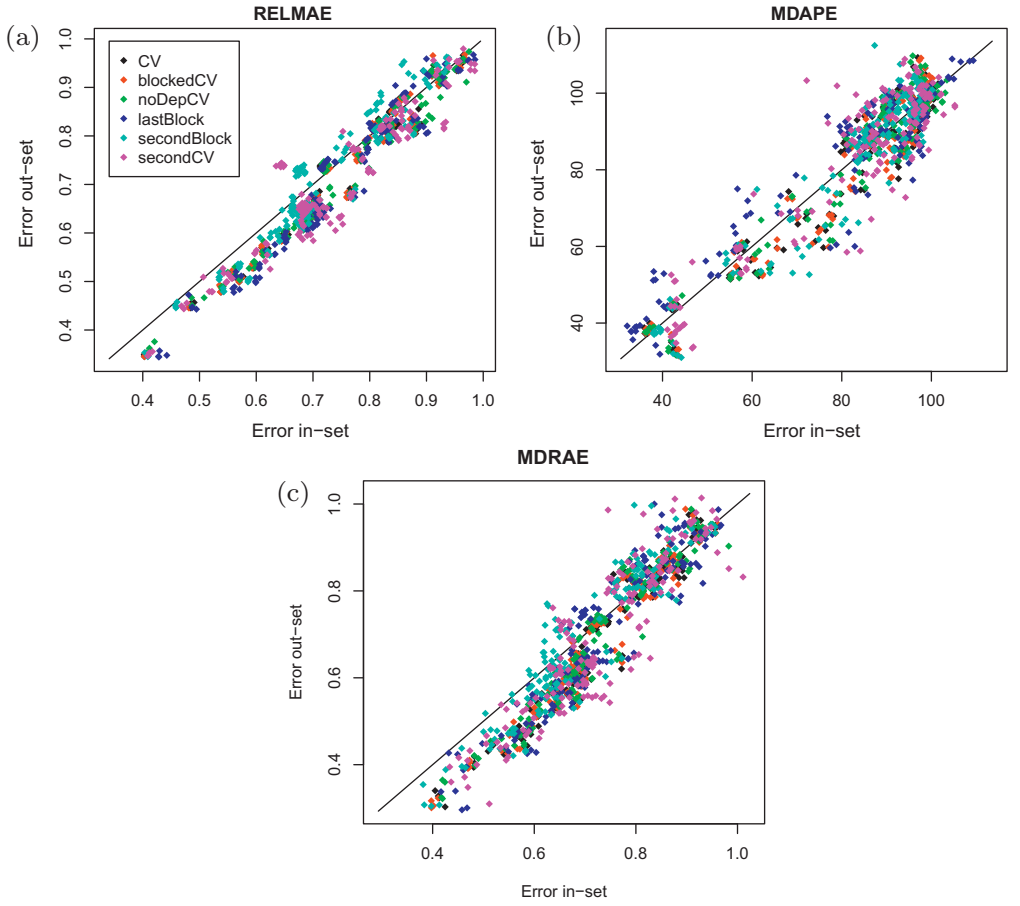

Fig. 4. Point plots for scenario (AS1) (synthetic data with few significant lags), using different error measures. Every point represents the in-set and out-set errors of one method applied to one dataset, with its parameters selected by one of the model selection procedures. As we use four methods and, within this scenario 40 datasets, 160 points are present for every one of the six model selection procedures.

and invertibility (e.g., by using the Dickey–Fuller unit root test [42]), parameters could be generated randomly and then checked for validity. However, if l (or k, respectively) is large finding valid parameters is not trivial (see Fig. 2) and random generation may take a long time, as lots of potential parameters have to be generated and tested. Instead of this generate-and-test approach, we sample the roots of the characteristic polynomials randomly from a uniform distribution in the interval $[-root_{max}, -1.1] \wedge [1.1, root_{max}]$, with $root_{max}$ being a parameter to be chosen. From this roots, the coefficients can then be computed by algebraic standard methods. It has to be noted, that the characteristic polynomials are constrained to have real-valued roots. Initial values are chosen randomly, and the first $2 \cdot max(l, k) + 1$ values of every series are discarded to remove effects of these values. The procedure is used to simulate AR processes by setting the coefficients of the MA part to zero, and vice versa.

Non-linear time series are simulated by a similar procedure, introducing non-linearities in the following way. Parameters for an AR model are generated as described above. Then, for every lag, a non-linear function is chosen randomly from $\cos(x)$, $\sin(x)$, $\arctan(x)$, $\tanh(x)$, and $\exp\left(-\frac{x}{c}\right)$ (where c is a constant value; throughout our experiments we used $c = 10{,}000$). Series are simulated as within the AR model, but with application of the corresponding non-linear function to every $y_{t-1}, \ldots, y_{t-l}$.

4.2.2. Real-world data

Data from the Santa Fe forecasting competition [51] and the NNGC1[3] competition are used.

The Santa Fe competition data set consists of six time series sets, all of them with several thousands of values. Five of these sets are taken from real-world applications, i.e., laser generated data, physiological data, currency exchange rate data, astrophysical data, and audio data. The sixth series is computer generated. Out of this data, we use five time series: the laser generated data, two series of the physiological data, the computer generated series, and a continuous part of the astrophysical

[3] http://www.neural-forecasting-competition.com/datasets.htm.

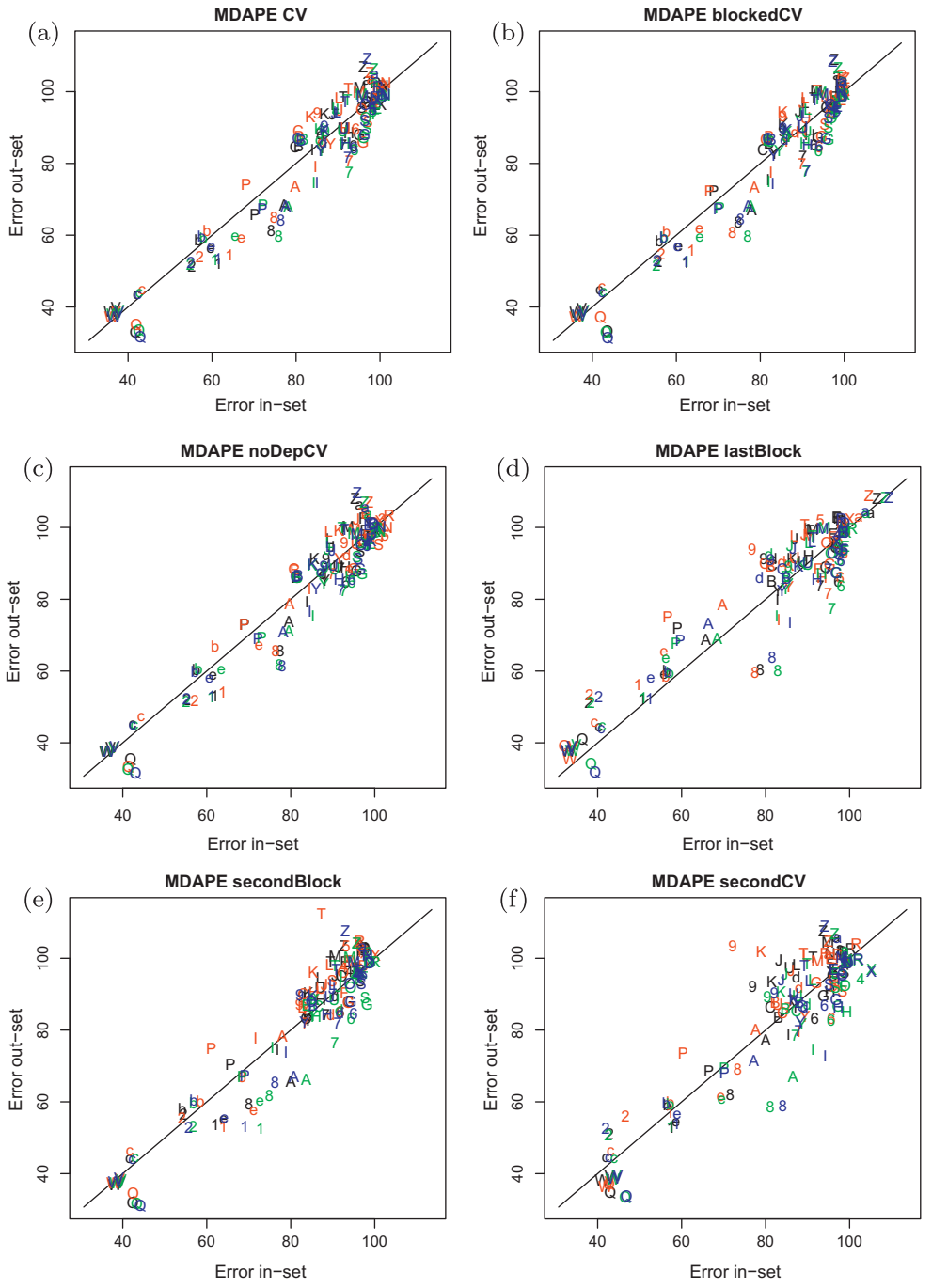

Fig. 5. Point plots for scenario (AS1), using MDAPE as error measure, with single plots for every model selection procedure. Each symbol indicates a different dataset and each color a method. The methods are: (black) `svmRadial`, (red) `nnet`, (blue) `lasso`, (green) `lm`. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

data. The other data is not used as it requires special treatment (time stamps instead of uniform intervals, missing values, learning one concept from many time series), which is not the focus of our work.

From the NNGC1 data, the high-frequency data is used, i.e., weekly, daily, and hourly data. The weekly data are economic data related to the oil industry (gasoline prices, amount of imports to the U.S., etc.). The daily time series are measures of traffic volume passing several tunnels, and the hourly data are average late arrival times and arrival times of airports and metro systems.

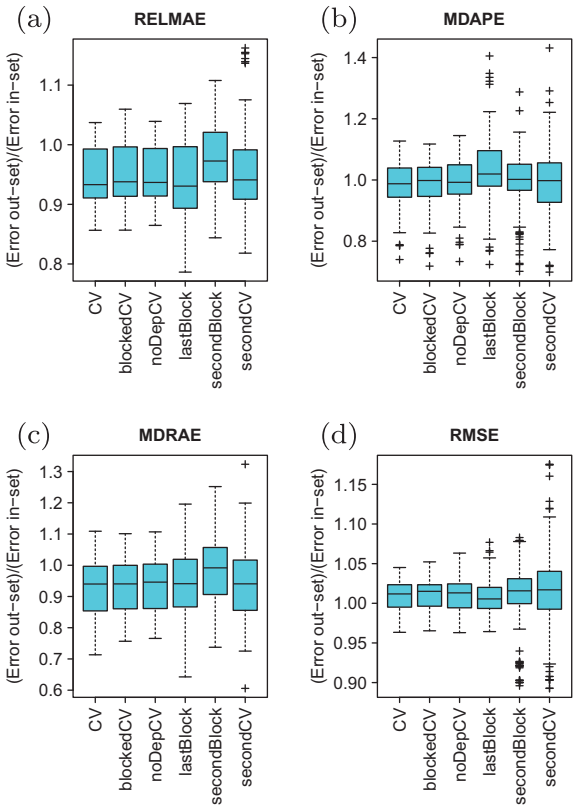

Fig. 6. Box plots of scenario (AS1).

4.2.3. Application scenarios

Application of cross-validation without dependent values is not feasible if relevant lags are too large. So, synthetic and real-world data was used with only few and small relevant lags. Additionally, a simulation study was performed with data containing larger and more lags. Then, cross-validation without dependent values cannot be applied. Thus, three different scenarios were analyzed within the study:

- For application scenario one (AS1), synthetic data with significant autocorrelations in only few and small lags (between one and five) were considered, so that cross-validation without dependent values is applicable. Time series with 1000 values were generated with the simulation methods for AR, MA, ARMA, and non-linear time series presented in Section 4.2.1. The ARMA series were simulated with the same order of the AR and the MA part, and autocorrelations in the first one to five lags were used, i.e., $l = k$, $l, k \in \{1, 2, 3, 4, 5\}$. For every lag, values of 5.0 and 10.0 were used for the $root_{max}$ parameter. So, in total 40 time series were simulated.
- To analyze behavior of the methods on time series that have autocorrelations in more and larger lags, application scenario two (AS2) uses synthetic data with autocorrelations in the last 10–30 lags. Cross-validation without dependent values has to be omitted then. Series were simulated in analogy to scenario (AS1), but with k, $l \in \{10, 15, 20, 25, 30\}$.
- Application scenario three (AS3) considers real-world data (with using four lags, to enable the use of cross-validation without dependent values). All real-world series are tested with the Augmented Dickey–Fuller test [42] for stationarity. The series that do not pass the test are not used. Though the non-stationary series are most likely to show poor performance under cross-validation, as stated in Section 3.3, experiments are likely to be only valid for these particular series. Also, it is likely that the forecasting models would show poor performance as well, possibly indicating that these series would require special preprocessing for non-stationarity, which is not the focus of this study. So, finally a total of 29 real-world time series is used, five from the Santa Fe competition, and 24 from the NNGC1 competition data.

4.3. Data preparation and partitioning

During this study we will use rolling-origin-update evaluation with one-step-ahead forecasting, since this is the most common way such models are used.

As discussed earlier, last block evaluation simulates the typical real-world application scenario of forecasting systems. So, we withhold from every series a percentage p_{ds} of values as "unknown future" for validation. In the following, we will call

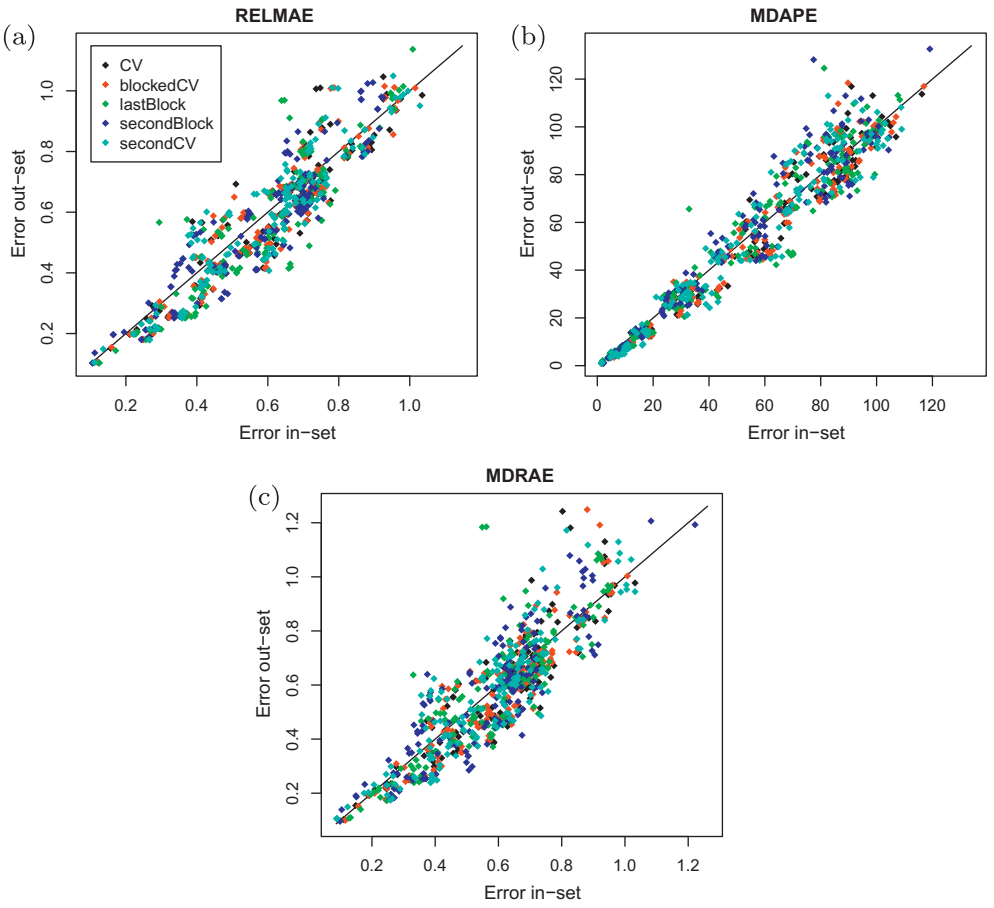

Fig. 7. Point plots for scenario (AS2) (synthetic data with more relevant lags), using different error measures. Every point represents the in-set and out-set errors of one method applied to one dataset, with its parameters selected by one of the model selection procedures. As we use four methods and, within this scenario 40 datasets, 160 points are present for every one of the five model selection procedures.

this dataset the validation set, the out-of-sample set, or shortly the out-set, as it is completely withhold from all other model building and model selection processes. The remaining data accordingly will in the following be called the in-set. The out-set is chosen in such a way that the later in-set evaluation can be performed without problems. That is, values from its beginning are removed according to the number of lags that later will be used for training, so that the out-set is independent of the other data, and it is chosen in a way that the amount of data remaining in the in-set can be partitioned equally, e.g., if 5-fold cross-validation is to be used, the remaining amount of data will be divisible by five. The process is illustrated in Fig. 3. Throughout our experiments, we use $p_{ds} = 0.8$, i.e., 20% of the data is used as out-set.

The in-set data is used for model building and model selection in the following way: the lags l_{ds} to be used for forecasting are chosen. For synthetic series, the lags are known, as they were specified during data generation, for the real-world data they have to be estimated. To have the possibility to use all model selection procedures (especially the procedure that removes dependent values, see Section 4.4), four lags are used for real-world series. This seems feasible, as the focus of this study does not lie in the actual performance of the methods, but in the performance of the model selection procedures. Then, the data is embedded as already discussed in Section 3.2.

4.4. Compared model selection procedures

From the embedded versions of the data, training and test sets are generated according to different model selection strategies. The procedures are then used for choosing the parameters of each method, as the model selection procedures are applied with each method and each parameter configuration, and for each method, the parameter set that produced the minimal error within the model selection procedure is chosen. This error is furthermore used as an estimate of the overall error, and is later compared to the errors that the methods (with these parameter configurations) produce on the out-set. We consider six different model selection strategies, named `CV`, `blockedCV`, `noDepCV`, `lastBlock`, `secondBlock`, and `secondCV`.

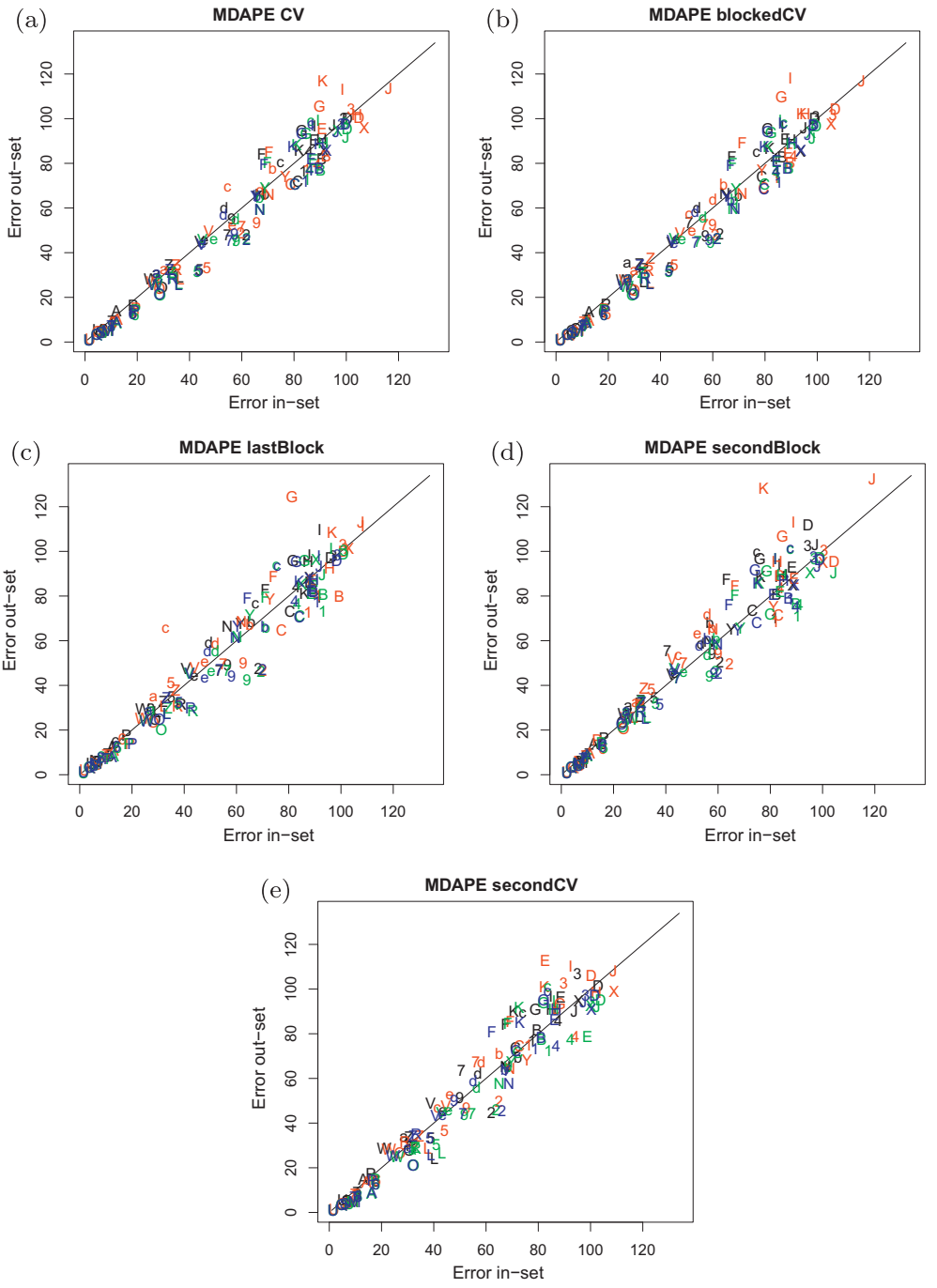

Fig. 8. Point plots for scenario (AS2), using MDAPE as error measure, with single plots for every model selection procedure. Each symbol indicates a different dataset and each color a method. The methods are: (black) `svmRadial`, (red) `nnet`, (blue) `lasso`, (green) `lm`. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For `CV`, standard 5-fold cross-validation is applied, i.e., the embedded data is partitioned randomly into five sets, and within five turns every set is used as test set, while the other sets are used for training. For `blockedCV`, 5-fold cross-validation is applied on data that is not partitioned randomly, but sequentially into five sets. So, the problem of dependent values is resolved (except for some values at the borders of the blocks, which can be removed). The problem that a system evolving over time might have generated the time series and render results of cross-validation incorrect remains. During `noDepCV`, 5-fold cross-validation without the dependent values is applied: the sets generated for `CV` are used, but according to the lags used for embedding, dependent values are removed from the training set. As stated earlier, depending on the lags used a lot

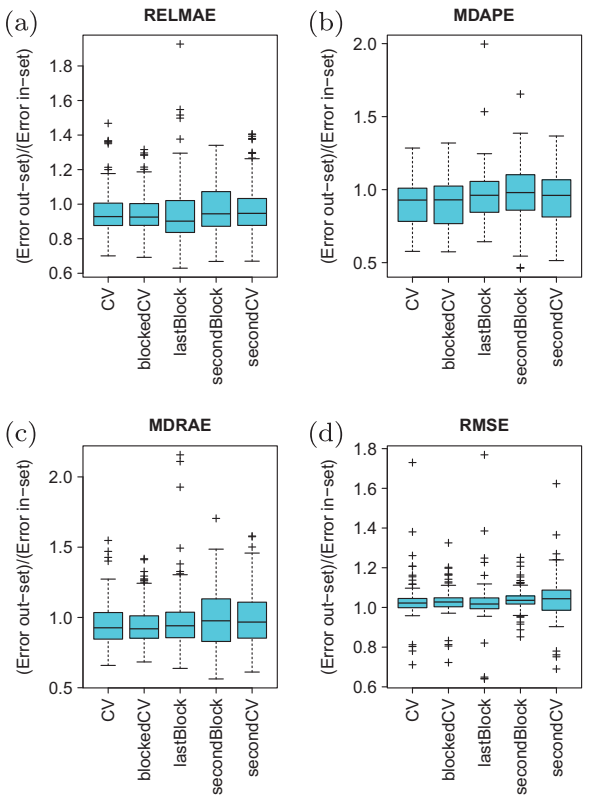

Fig. 9. Box plots of scenario (AS2).

of values have to be removed, so that this model selection procedure only can be applied if the amount of lags is not large compared to the number of cross-validation subsets, i.e., in AS1 and AS3. The evaluation type `lastBlock` uses only the last set of `blockedCV` for evaluation. To study in more detail the effects of using a connected evaluation set or random chosen values, and the effect of using data from the end or from somewhere in between, `secondBlock` evaluation uses not the last but the second block of `blockedCV` for evaluation, and `secondCV` uses only the second subset of `CV`. The different types of data sampling within the model selection procedures illustrates Fig. 3.

The other methods discussed in Section 3.4, namely forward validation and APE, cannot be used, as no recalibration of the models is performed within our experiments.

4.5. Computed error measures

Though we do not compute average measures over multiple time series, as the study employs various different time series, only the use of scale-independent error measures is feasible in order to achieve results that can be compared among themselves. As some of the series contain zero values, measures based on percentage errors or on scaled errors are only applicable, if a robust averaging is used, e.g., the median. So, we calculate the MDAPE and MDRAE. Furthermore, as for every time series and the horizon used many values are available, the use of relative measures (e.g., RELMAE) is possible. Within these, we use the naïve forecast (i.e., the last known value) as a benchmark. It has to be noted, that by using this benchmark, a difference between blocked and unblocked validation modes exists, as during unblocked modes the naïve forecasts might also be present in the training set as lagged and target values, whereas this is not the case for blocked validation modes (if the borders of the sets are removed).

Furthermore, the additional validation procedure we apply enables the scaling of the error directly by the error on the out-set. For this purpose, also scale-dependent measures such as the RMSE can be used.

4.6. Plots and statistical tests

The in-set error, estimated by the model selection procedure, is compared to the error on the out-set. If the model selection procedure produces a good estimate for the error the two errors should be very similar. Therefore, we analyze plots of points (E_{in-set}, $E_{out-set}$). If the errors are equal, these points all lie on a line with origin zero and gradient one. In the following, we call this type of evaluation *point plots*.

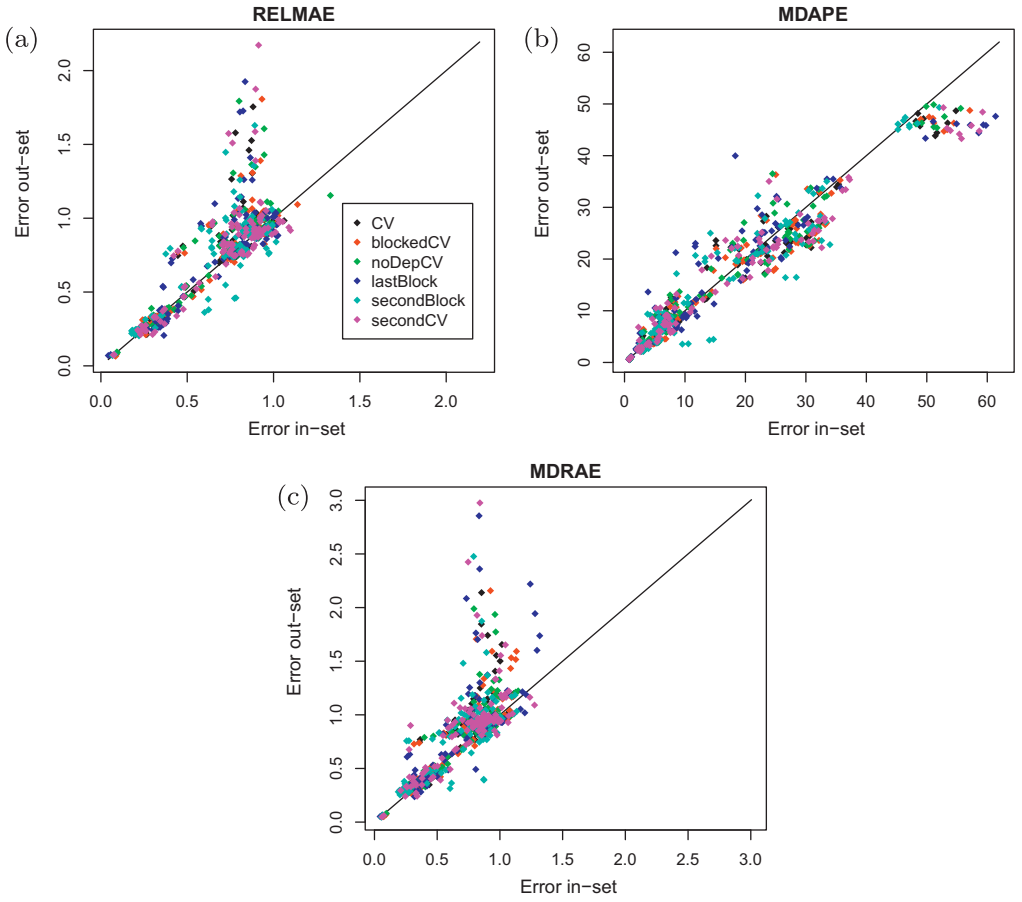

Fig. 10. Point plots for scenario (AS3) (real-world data), using different error measures. Every point represents the in-set and out-set errors of one method applied to one dataset, with its parameters selected by one of the model selection procedures. As we use four methods and, within this scenario 29 datasets, 116 points are present for every one of the six model selection procedures.

Additionally to the point plots, we analyze box-and-whisker plots containing directly the value of the quotient ($E_{out-set}/E_{in-set}$), which is especially interesting with the use of scale-dependent measures like the RMSE, as with using the quotient a normalization takes place, so that the results are comparable.

Statistical significance of the results is explored with the following non-parametric tests: the Friedman test in its implementations of García et al. [25] is used to determine if the distributions of the quotient ($E_{out-set}/E_{in-set}$) for the model selection procedures differ in their location parameter (the median). And the Fligner–Killeen test [17] (that is available in R) is used to determine if these distributions differ in their dispersion.

5. Experimental results and analysis

The complete results can be found at http://sci2s.ugr.es/dicits/papers/CV-TS. In the following, a selection of the results is presented.

5.1. Plots of the results

Fig. 4 shows point plots for scenario AS1, using different error measures. It can be observed that the RELMAE yields a less scattered distribution than MDAPE and MDRAE. The in-set error tends to overestimate the out-set error, especially when using relative measures, i.e., RELMAE or MDRAE. No systematical difference between different model selection procedures can be determined in this plot. To further examine the different model selection procedures, Fig. 5 shows the results of Fig. 4 in more detail, only for the MDAPE measure, but with the results of every model selection procedure in a different plot. Fig. 6 shows the results of Fig. 5 as a box plot, again including all error measures of Fig. 4, and furthermore including the RMSE. Figs. 5 and 6 show graphically that the methods using only a single set for evaluation, i.e., `lastBlock`, `secondBlock`, and `secondCV` lead in general to more disperse, less robust results. Few differences between the model selection procedures

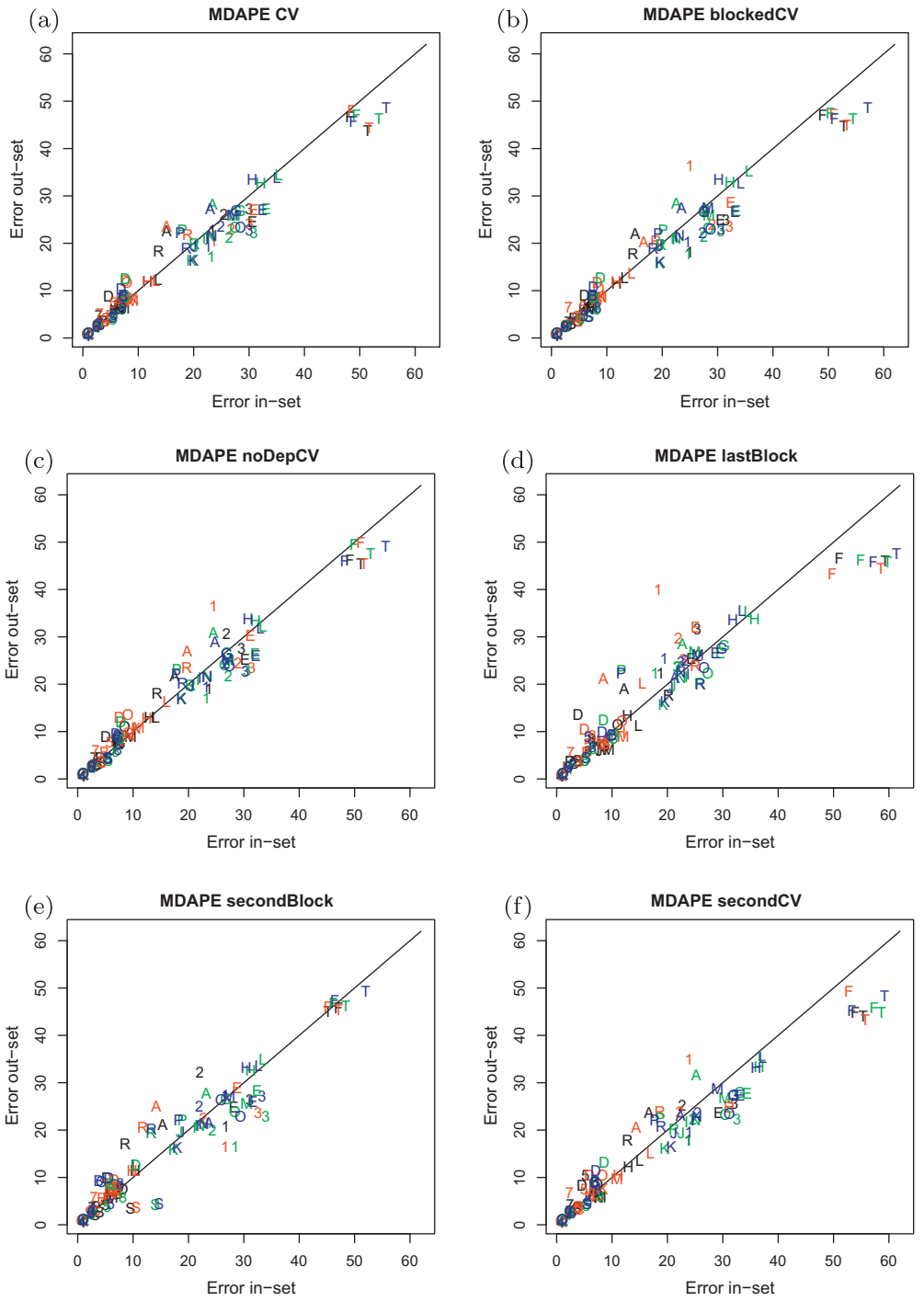

Fig. 11. Point plots for scenario (AS3), using MDAPE as error measure, with single plots for every model selection procedure. Each symbol indicates a different dataset and each color a method. The methods are: (black) `svmRadial`, (red) `nnet`, (blue) `lasso`, (green) `lm`. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the form of a bias are present. Furthermore, the difference is not in the way expected w.r.t. the theoretical problems of cross-validation (due to the dependencies within the values, an underestimation when applying cross-validation could occur).

Within the scenario AS2, `noDepCV` is not applicable any more. Point plots and box plots analogous to the plots of scenario AS1 are shown in Figs. 7–9. The results confirm the results of scenario AS1.

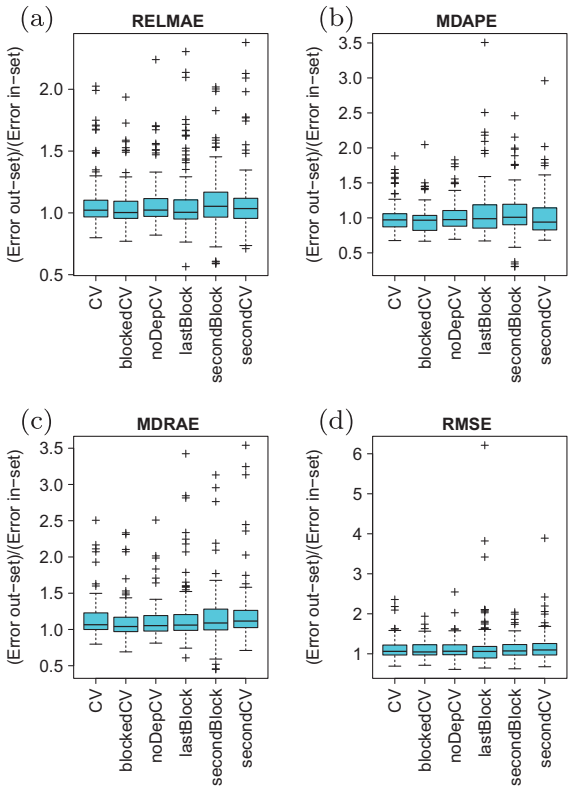

Fig. 12. Box plots of scenario (AS3).

Figs. 10–12 show the results of scenario AS3, where real-world data is used. The results on real-world data basically confirm the simulation results, but contain more noise. For real-world data *a priori* it is unknown, which delayed values are relevant, and no sophisticated methods to solve such problems were used within our study, so on some datasets the methods perform worse than the naïve forecast. This leads to the "traces" in Fig. 10, when measures that employ a benchmark are used.

5.2. Some results of parameter selection

Tables 3 and 4 show examples in scenario AS3 for the differences in choosing the parameters by distinct model selection procedures, and error measures, respectively. The examples illustrate that both the model selection procedure and the error measure used influences in the choice of the parameter set.

5.3. Statistical evaluation

The Friedman test shows high significance ($p < 0.001$) in the difference of medians for the groups of all model selection procedures, for all combinations of scenarios and error measures, except for AS3 with RMSE, where a p-value of 0.126 was obtained. As the data results from different methods (e.g., not from a test and a control sample), we cannot expect the resulting points to have exactly the same distributions. So, the statistical significance in the difference shows merely that the amount of data points is high enough to obtain a stable result. As a statistical test does not express the size or relevance of the difference found, their consequences for the practice are often unclear [53]. Also, in the forecasting community the use of statistical tests in general is discussed controversially [4,5]. Therefore, we perform an analysis of the medians and their differences in Table 5. The table shows that with respect to under or overestimation of the error, no difference between the model selection procedures can be found. The differences in the accuracy with which the in-set error predicts the out-set error are small, and vary with the characteristics of the data and the error measures. So, e.g., choice of the error measure seems more relevant than choice of the model selection procedure.

Table 6 shows results of the Fligner–Killeen test. Though the difference between `lastBlock` and the cross-validation procedures is not always statistically significant, the table clearly shows the trend that the difference between the last block evaluation and the cross-validation methods is bigger than difference among the cross-validation methods.

Table 3
Parameter sets that are chosen by the different model selection procedures, using MDAPE, for the laser data of the Santa Fe competition. The numbers represent the index of the parameter set within the parameter grid, as shown in Table 2 for `nnet`. For `svmRadial`, the relevant sets 18, 19, and 20 all define the parameter gamma to be 0.2, and the parameter cost to be 10, 100, and 1000, respectively. In the method `lasso`, 4 means the only present parameter fraction is set to 0.9.

	svmRadial	nnet	lasso
CV	19	4	4
blockedCV	19	8	4
noDepCV	20	4	4
lastBlock	19	3	4
secondBlock	18	7	4
secondCV	18	4	4

Table 4
Parameter sets that are chosen by the different error measures, using standard cross-validation, for the laser data of the Santa Fe competition. The numbers represent the index in the parameter grid, as in Table 3.

	svmRadial	nnet	lasso
RELMAE	19	3	4
MDAPE	19	4	4
MDRAE	19	4	3
RMSE	19	3	4

Table 5
Medians and differences in the median. The columns are: CV, bCV, and lB: Median of ($E_{out-set}/E_{in-set}$) values for the procedures `CV`, `blockedCV`, and `lastBlock`, diminished by one. The optimal ratio of the errors is one (which would result in a zero in the table), as then the in-set error equals the out-set error, and hence is a good estimate. Negative values in the table indicate a greater in-set error, i.e., the out-set error is overestimated. A positive value, on the contrary, indicates underestimation. CV-lB, CV-bCV, and bCV-lB: differences of the absolute values of CV, bCV, and lB. A negative value indicates that the minuend in the difference leads to a value nearer to one, that is, to a better estimate of the error.

		CV	bCV	lB	CV-lB	CV-bCV	bCV-lB
RELMAE	AS1	−0.067	−0.062	−0.069	−0.002	0.005	−0.007
	AS2	−0.072	−0.075	−0.098	−0.026	−0.003	−0.023
	AS3	0.022	0.003	0.005	0.017	0.019	−0.002
MDAPE	AS1	−0.012	−0.002	0.020	−0.007	0.011	−0.018
	AS2	−0.071	−0.070	−0.039	0.033	0.001	0.031
	AS3	−0.028	−0.033	−0.012	0.017	−0.005	0.022
MDRAE	AS1	−0.060	−0.060	−0.059	0.001	0.000	0.001
	AS2	−0.074	−0.081	−0.059	0.014	−0.007	0.021
	AS3	0.065	0.041	0.060	0.005	0.025	−0.020
RMSE	AS1	0.012	0.015	0.006	0.006	−0.003	0.010
	AS2	0.022	0.027	0.017	0.005	−0.005	0.010
	AS3	0.061	0.046	0.060	0.002	0.015	−0.013

Table 6
p-Values of the Fligner test. First column: Fligner test for differences in variance, applied to the group of all model selection procedures (6 procedures for AS1 and AS3, and 5 procedures for AS2). Columns 2–4: Tests of interesting pairs of methods (without application of a post hoc procedure).

		All	CV, lB	bCV, lB	CV, bCV
RELMAE	AS1	0.000	0.002	0.000	0.307
	AS2	0.010	0.119	0.087	0.849
	AS3	0.005	0.115	0.050	0.618
MDAPE	AS1	0.000	0.018	0.004	0.444
	AS2	0.108	0.943	0.424	0.529
	AS3	0.005	0.007	0.013	0.784
MDRAE	AS1	0.000	0.020	0.007	0.701
	AS2	0.001	0.346	0.369	0.983
	AS3	0.054	0.256	0.163	0.737
RMSE	AS1	0.000	0.448	0.508	0.707
	AS2	0.000	0.230	0.047	0.433
	AS3	0.078	0.028	0.008	0.644

6. Conclusions

In this paper, we reviewed the methodology of evaluation in traditional forecasting and in regression and machine learning methods used for time series. We observed that with the use of general regression methods and machine learning techniques, also techniques usually used for their evaluation (especially cross-validation) are used within the time series problems. This raises theoretical concerns, as the data contains dependencies and time-evolving effects may occur. On the other hand, in traditional forecasting only the last part of every series is typically used for evaluation, thus not exhausting the available information.

Because of the shortcomings that the commonly used methods have, various other methods have been proposed in the literature. We grouped these in Section 3.4 into methods based on the last block, methods that use non-dependent cross-validation, and blocked cross-validation methods.

In order to analyze the shortcomings of the popular methods and to evaluate the potential benefit from the use of other methods in common application situations, we performed a thorough empirical study. It includes the comparison of six model selection procedures (among others cross-validation and last block evaluation) on forecasts of four different methods for synthetic and real-world time series.

Using standard 5-fold cross-validation, no practical effect of the dependencies within the data could be found, regarding whether the final error is under- or overestimated. On the contrary, last block evaluation tends to yield less robust error measures than cross-validation and blocked cross-validation. The non-dependent cross-validation procedure also yields robust results, but might lead to a waste of data and therewith is not applicable in many cases.

Regarding time-evolving effects, no differences could be found, as using the last block and using a block taken from somewhere within the data (we used the second block of the blocked cross-validation) showed a similar behavior. This is not surprising, as we limited the study to stationary time series. However, assuming stationary time series is a common and reasonable assumption in many applications.

Though no practical problems with standard cross-validation could be found, we suggest the use of blocked cross-validation, together with an adequate control for stationarity, since it makes full use of all available information both for training and testing, thus yielding a robust error estimate. And theoretical problems are solved in a practical manner, as assuming stationarity and approximate independence after a certain amount of lags coincides with common assumptions of many application scenarios.

Acknowledgements

This work was supported in part by the Spanish Ministry of Science and Innovation (MICINN) under Project TIN-2009-14575. C. Bergmeir holds a scholarship from the Spanish Ministry of Education (MEC) of the "Programa de Formación del Profesorado Universitario (FPU)".

References

[1] N. Amjady, F. Keynia, Application of a new hybrid neuro-evolutionary system for day-ahead price forecasting of electricity markets, Applied Soft Computing Journal 10 (3) (2010) 784–792.
[2] R.D.A. Araujo, Swarm-based translation-invariant morphological prediction method for financial time series forecasting, Information Sciences 180 (24) (2010) 4784–4805.
[3] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Statistics Surveys 4 (2010) 40–79.
[4] J.S. Armstrong, Significance tests harm progress in forecasting, International Journal of Forecasting 23 (2) (2007) 321–327.
[5] J.S. Armstrong, Statistical significance tests are unnecessary even when properly done and properly interpreted: Reply to commentaries, International Journal of Forecasting 23 (2) (2007) 335–336.
[6] J.L. Aznarte, J.M. Benítez, Equivalences between neural-autoregressive time series models and fuzzy systems, IEEE Transactions on Neural Networks 21 (9) (2010) 1434–1444.
[7] S.D. Balkin, J.K. Ord, Automatic neural network modeling for univariate time series, International Journal of Forecasting 16 (4) (2000) 509–515.
[8] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, 1970.
[9] Prabir Burman, Edmond Chow, Deborah Nolan, A cross-validatory method for dependent data, Biometrika 81 (2) (1994) 351–358.
[10] P.S. Carmack, W.R. Schucany, J.S. Spence, R.F. Gunst, Q. Lin, R.W. Haley, Far casting cross-validation, Journal of Computational and Graphical Statistics 18 (4) (2009) 879–893.
[11] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001 <http://www.csie.ntu.edu.tw/cjlin/libsvm>.
[12] C. Chatfield, Model uncertainty and forecast accuracy, Journal of Forecasting 15 (7) (1996) 495–508.
[13] B.-J. Chen, M.-W. Chang, C.-J. Lin, Load forecasting using support vector machines: a study on eunite competition 2001, IEEE Transactions on Power Systems 19 (4) (2004) 1821–1830.
[14] Y. Chen, B. Yang, Q. Meng, Y. Zhao, A. Abraham, Time-series forecasting using a system of ordinary differential equations, Information Sciences 181 (1) (2011) 106–114.
[15] Zhuo Chen, Yuhong Yang, Assessing forecast accuracy measures. Technical Report 2004–10, Iowa State University, Department of Statistics & Statistical Laboratory, 2004.
[16] R.B. Cleveland, W.S. Cleveland, J.E. McRae, I. Terpenning, Stl: a seasonal-trend decomposition procedure based on loess (with discussion), Journal of Official Statistics 6 (1990) 3–73.
[17] W.J. Conover, Mark E. Johnson, Myrle M. Johnson, Comparative study of tests for homogeneity of variances, with applications to the outer continental shelf bidding data, Technometrics 23 (4) (1981) 351–361.
[18] S.F. Crone, M. Hibon, K. Nikolopoulos, Advances in forecasting with neural networks? Empirical evidence from the nn3 competition on time series prediction, International Journal of Forecasting 27 (3) (2011) 635–660.
[19] Jonathan D. Cryer, Kung-Sik Chan, Time Series Analysis With Applications in R, Springer, 2008. ISBN 978-0-387-75958-6.
[20] C.G. da Silva, Time series forecasting with a non-linear model and the scatter search meta-heuristic, Information Sciences 178 (16) (2008) 3288–3299.

[21] J.G. De Gooijer, R.J. Hyndman, 25 Years of time series forecasting, International Journal of Forecasting 22 (3) (2006) 443–473.
[22] G. Deco, R. Neuneier, B. Schürmann, Non-parametric data selection for neural learning in non-stationary time series, Neural Networks 10 (3) (1997) 401–407.
[23] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, H. Ishwaran, K. Knight, J.-M. Loubes, P. Massart, D. Madigan, G. Ridgeway, S. Rosset, J.I. Zhu, R.A. Stine, B.A. Turlach, S. Weisberg, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2) (2004) 407–499.
[24] M. Gan, H. Peng, X. Peng, X. Chen, G. Inoussa, A locally linear rbf network-based state-dependent ar model for nonlinear time series modeling, Information Sciences 180 (22) (2010) 4370–4383.
[25] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (10) (2010) 2044–2064.
[26] P. Goodwin, R. Lawton, On the asymmetry of the symmetric mape, International Journal of Forecasting 15 (4) (1999) 405–408.
[27] C. Hamzaçebi, Improving artificial neural networks' performance in seasonal time series forecasting, Information Sciences 178 (23) (2008) 4550–4559.
[28] Jeffrey D. Hart, Automated kernel smoothing of dependent data by using time series cross- validation, Journal of the Royal Statistical Society: Series B (Methodological) 56 (3) (1994) 529–542.
[29] J.S.U. Hjorth, Computer Intensive Statistical Methods, Validation, Model Selection and Bootstrap, Chapman and Hall, 1994.
[30] Urban Hjorth, Model selection and forward validation, Scandinavian Journal of Statistics 9 (2) (1982) 95–105.
[31] R.J. Hyndman, A.B. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting 22 (4) (2006) 679–688.
[32] A. Inoue, L. Kilian, On the selection of forecasting models, Journal of Econometrics 130 (2) (2006) 273–306.
[33] T.Y. Kim, K.J. Oh, C. Kim, J.D. Do, Artificial neural networks for non-stationary time series, Neurocomputing 61 (1–4) (2004) 439–447.
[34] R. Kunst, Cross validation of prediction models for seasonal time series by parametric bootstrapping, Austrian Journal of Statistics 37 (2008) 271–284.
[35] Robert M. Kunst, Adusei Jumah, Toward a theory of evaluating predictive accuracy, Economics Series, vol. 162, Institute for Advanced Studies, 2004.
[36] S. Makridakis, M. Hibon, The m3-competition: results, conclusions and implications, International Journal of Forecasting 16 (4) (2000) 451–476.
[37] Allan D.R. McQuarrie, Chih-Ling Tsai, Regression and Time Series Model Selection, World Scientific Publishing, 1998.
[38] J. Opsomer, Y. Wang, Y. Yang, Nonparametric regression with correlated errors, Statistical Science 16 (2) (2001) 134–153.
[39] N.M. Pindoriya, S.N. Singh, S.K. Singh, An adaptive wavelet neural network-based energy price forecasting in electricity markets, IEEE Transactions on Power Systems 23 (3) (2008) 1423–1432.
[40] R Development Core Team, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2009, ISBN 3-900051-07-0.
[41] J. Racine, Consistent cross-validatory model-selection for dependent data: hv-block cross-validation, Journal of Econometrics 99 (1) (2000) 39–61.
[42] S.E. Said, D.A. Dickey, Testing for unit roots in autoregressive-moving average models of unknown order, Biometrika 71 (1984) 599–607.
[43] J. Shao, An asymptotic theory for linear model selection, Statistica Sinica 7 (1997) 221–264.
[44] Jun Shao, Linear model selection by cross-validation, Journal of the American Statistical Association 88 (422) (1993) 486–494.
[45] T.A.B. Snijders, On cross-validation for predictor evaluation in time series, in: T.K. Dijkstra (Ed.), On Model Uncertainty and its Statistical Implications, 1988, pp. 56–69.
[46] M. Stone, Cross-validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society: Series B (Methodological) 36 (2) (1974) 111–147.
[47] L.J. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, International Journal of Forecasting 16 (4) (2000) 437–450.
[48] H. Tong, Non-Linear Time Series: A Dynamical System Approach, Clarendon Press, Oxford, 1990.
[49] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S, fourth ed., Springer, New York, 2002. ISBN 0-387-95457-0.
[50] E.-J. Wagenmakers, P. Grünwald, M. Steyvers, Accumulative prediction error and the selection of time series models, Journal of Mathematical Psychology 50 (2) (2006) 149–166.
[51] A.S Weigend, N.A. Gershenfeld, (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, 1994.
[52] G.P. Zhang, A neural network ensemble method with jittered training data for time series forecasting, Information Sciences 177 (23) (2007) 5329–5346.
[53] S.T. Ziliak, D.N. McCloskey, Size matters: the standard error of regressions in the american economic review, Journal of Socio-Economics 33 (5) (2004) 527–546.