Computational Statistics and Data Analysis 120 (2018) 70–83

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

A note on the validity of cross-validation for evaluating autoregressive time series prediction

Christoph Bergmeir [a,*], Rob J. Hyndman [b], Bonsoo Koo [b]

[a] Faculty of Information Technology, Monash University, Melbourne, Australia
[b] Department of Econometrics & Business Statistics, Monash University, Melbourne, Australia

ARTICLE INFO

Article history:
Received 10 June 2016
Received in revised form 30 October 2017
Accepted 3 November 2017
Available online 22 November 2017

Keywords:
Cross-validation
Time series
Autoregression

ABSTRACT

One of the most widely used standard procedures for model evaluation in classification and regression is *K*-fold cross-validation (CV). However, when it comes to time series forecasting, because of the inherent serial correlation and potential non-stationarity of the data, its application is not straightforward and often replaced by practitioners in favour of an out-of-sample (OOS) evaluation. It is shown that for purely autoregressive models, the use of standard *K*-fold CV is possible provided the models considered have uncorrelated errors. Such a setup occurs, for example, when the models nest a more appropriate model. This is very common when Machine Learning methods are used for prediction, and where CV can control for overfitting the data. Theoretical insights supporting these arguments are presented, along with a simulation study and a real-world example. It is shown empirically that *K*-fold CV performs favourably compared to both OOS evaluation and other time-series-specific techniques such as non-dependent cross-validation.

1. Introduction

Cross-validation (CV) (Stone, 1974; Arlot and Celisse, 2010) is one of the most widely used methods to assess the generalizability of algorithms in classification and regression (Hastie et al., 2009), and is subject to ongoing active research (e.g., Budka and Gabrys, 2013; Borra and Di Ciaccio, 2010; Bergmeir et al., 2014; Moreno-Torres et al., 2012). However, when it comes to time series prediction, practitioners are often unsure of the best way to evaluate their models. There is often a feeling that we should not be using future data to predict the past. In addition, the serial correlation in the data, along with possible non-stationarities, make the use of CV appear problematic as it does not account for these issues (Bergmeir and Benítez, 2012). Practitioners usually resort to out-of-sample (OOS) evaluation instead, where a section from the end of the series is withheld for evaluation. In this way, only one evaluation on a test set is considered, whereas with the use of cross-validation, various such evaluations are performed. So, by using OOS, the benefits of CV, especially for small datasets, cannot be exploited. One important part of the problem is that in the traditional forecasting literature, OOS evaluation is the standard evaluation procedure, partly because fitting of standard models such as exponential smoothing (Hyndman et al., 2008) or ARIMA models are fully iterative in the sense that they start estimation at the beginning of the series. In addition, some research has demonstrated cases where standard CV fails in a time series context. For example, Opsomer et al. (2001) show that standard CV underestimates bandwidths in a kernel estimator regression framework if autocorrelation of the error is high, so that the method overfits the data. As a result, several CV techniques have been developed especially for the

* Correspondence to: Faculty of Information Technology, P.O. Box 63 Monash University, Victoria 3800, Australia.
E-mail address: christoph.bergmeir@monash.edu (C. Bergmeir).

https://doi.org/10.1016/j.csda.2017.11.003
0167-9473/

dependent case (Györfi et al., 1989; Burman and Nolan, 1992; Burman et al., 1994; McQuarrie and Tsai, 1998; Racine, 2000; Kunst, 2008).

This study addresses the problem in the following way. When purely (non-linear, non-parametric) autoregressive methods are applied to forecasting problems, as is often the case (e.g., when using Machine Learning methods), the aforementioned problems of CV are largely irrelevant, and CV can and should be used without modification, as in the independent case. To the best of our knowledge, this is the first paper to justify the use of the standard K-fold CV in the dependent setting without modification. Our paper draws the line of applicability of the procedure between models that have uncorrelated errors, and models that are heavily misspecified and thereby do not produce uncorrelated errors. In practice, this means that the method can be used without problems to detect overfitting, as overfitted models have uncorrelated errors. Underfitting should be tackled beforehand, e.g. by testing residuals for serial correlation. We provide a theoretical proof and additional results of simulation experiments and a real-world example to justify our argument.

2. Cross-validation for the dependent case

Cross-validation for the dependent setting has been studied extensively in the literature, including Györfi et al. (1989), Burman and Nolan (1992) and Burman et al. (1994). Let $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ be a time series. Traditionally, when K-fold CV is performed, K randomly chosen numbers out of the vector $\boldsymbol{y}$ are removed. This removal invalidates the CV in the dependent setting because of the correlation between errors in the training and test sets. Therefore, Burman and Nolan (1992) suggest bias correction, whereas Burman et al. (1994) propose h-block CV whereby the h observations preceding and following the observation are left out in the test set. We call this procedure non-dependent cross-validation, as it leaves out the possibly dependent observations and only considers data points that can be considered to be independent.

However, both bias correction and the h-block CV method have their limitations including inefficient use of the available data.

Let us now consider a purely autoregressive model of order p

$$y_t = g(\boldsymbol{x}_t, \boldsymbol{\theta}) + \varepsilon_t, \tag{1}$$

where ε_t is a shock or disturbance term, $\boldsymbol{\theta}$ is a parameter vector, $\boldsymbol{x}_t \in \mathbb{R}^p$ consists of lagged values of y_t and $g(\boldsymbol{x}_t, \boldsymbol{\theta}) = \mathrm{E}_{\boldsymbol{\theta}}[y_t|\boldsymbol{x}_t]$. Here $g(\cdot)$ could be a linear or nonlinear function, or even a nonparametric function. Thus, $g(\cdot)$ could be a totally unspecified function of the lagged values of y_t up to pth order.

Here, the lag order of the model is fixed and the time series is *embedded* accordingly, generating a matrix that is then used as the input for a (nonparametric, nonlinear) regression algorithm. The embedded time series with order p and a fixed forecast horizon of $h = 1$ is defined as follows:

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_p & y_{p+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-p} & y_{t-p+1} & \cdots & y_{t-1} & y_t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-p} & y_{n-p+1} & \cdots & y_{n-1} & y_n \end{bmatrix}. \tag{2}$$

Thus each row is of the form $[\boldsymbol{x}_t', y_t]$, and the first p columns of the matrix contain predictors for the last column of the matrix.

Recall the usual K-fold CV method, where the training data is partitioned into K separate sets, say $J = \{J_1, \ldots, J_K\}$. Define $J_{k^-} = \cup_{j \neq k} J_j$, so that for a particular evaluation k in the cross-validation, J_k is used as test set, and the remaining sets combined, J_{k^-}, are used for model fitting. Instead of reducing the training set by removing the h observations preceding and following the observation y_t of the test set, we leave out the entire set of rows corresponding to $t \in J_k$ in matrix (2). Fig. 1 illustrates the procedure.

Provided (1) is true, the rows of the matrix (2) are conditionally uncorrelated because $y_t - \hat{g}(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}) = \hat{\varepsilon}_t$ is simply a regression error, and $\{\hat{\varepsilon}_t\}$ are asymptotically independent and identically distributed (i.i.d.) provided g is estimated appropriately. Consequently, omitting rows of the matrix will not affect the bias or consistency of the estimates.

In practice, although we do not know the correct p and other hyper-parameters of the model, if our model is sufficiently large and flexible (and therefore liable to be overfitted), errors will be uncorrelated. In the case of correlated errors, the CV procedure will be biased, but this can be relatively easily tackled by testing the residuals for serial correlation.

It is worth mentioning that this method leaves the entire row related to the chosen test set out instead of test set components only. As a result, we lose much less information embedded in the data in this way than in the h-block CV.

3. The theoretical result

Let $y_1, \ldots, y_n$ be the observations from a stationary process where each y_t has distribution P on $\mathbb{R}$. We consider pure AR(p) models in order to discuss the validity of our method. Without loss of generality and for ease of notation, we focus on the leave-one-out CV because generalization to the K-fold CV is straightforward.

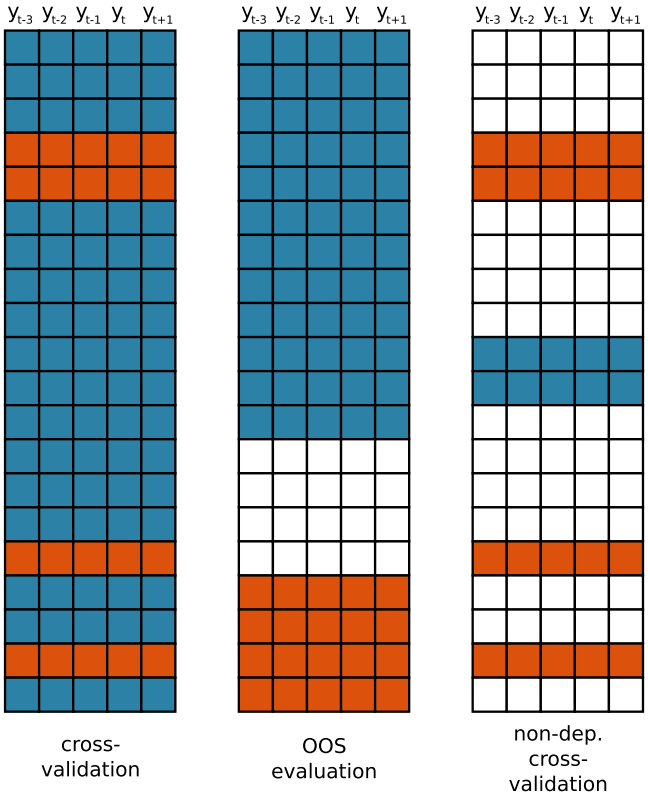

Fig. 1. Training and test sets for different cross-validation procedures for an embedded time series. Rows chosen for training are shown in blue, rows chosen for testing in orange, rows shown in white are omitted due to dependency considerations. The example shows one fold of a 5-fold CV, and an embedding of order 4. So, for the dependency considerations, 4 values before and after a test case cannot be used for training. We see that the non-dependent CV considerably reduces the available training data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Consider the following nonlinear regression model,

$$y_t = g(\boldsymbol{x}_t, \boldsymbol{\theta}) + \varepsilon_t, \tag{3}$$

where $g(\cdot)$ is a continuous and differentiable function with respect to $\boldsymbol{\theta}$ for all $\boldsymbol{x}_t = (y_{t-1}, y_{t-2}, \ldots, y_{t-p})'$ and ε_t is a regression error. The objective function related to the estimation of $\hat{\boldsymbol{\theta}}$ is given as

$$Q(\boldsymbol{\theta}) = \sum_{t=p+1}^{n} (y_t - g(\boldsymbol{x}_t, \boldsymbol{\theta}))^2.$$

By definition, $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} Q(\boldsymbol{\theta})$. Suppose $\{\tilde{y}_t\}_{t=1}^{m}$ is another process that has the same distribution as the sample data $\{y_t\}_{t=1}^{n}$ and $\tilde{\boldsymbol{x}}_t = (\tilde{y}_{t-1}, \tilde{y}_{t-2}, \ldots, \tilde{y}_{t-p})$. For example, $\{\tilde{y}_t\}_{t=1}^{m}$ may be the future data. Then, the prediction error we consider in the nonlinear models is given as

$$\mathrm{PE} = \mathrm{E}\{\tilde{y} - g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}})\}^2,$$

where $\hat{\boldsymbol{\theta}}$ is the solution to the set of nonlinear equations

$$q(\hat{\boldsymbol{\theta}}) = \sum_{t=p+1}^{n} \left(\frac{\partial}{\partial \boldsymbol{\theta}} g(\tilde{\boldsymbol{x}}_t, \hat{\boldsymbol{\theta}})\right)\left(\tilde{y}_t - g(\tilde{\boldsymbol{x}}_t, \hat{\boldsymbol{\theta}})\right) = \sum_{t=p+1}^{n} \left(\frac{\partial}{\partial \boldsymbol{\theta}} g(\tilde{\boldsymbol{x}}_t, \hat{\boldsymbol{\theta}})\right)\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) = 0,$$

where $\tilde{\varepsilon}_t(\boldsymbol{\theta}) = \tilde{y}_t - g(\tilde{\boldsymbol{x}}_t, \boldsymbol{\theta})$. By construction, $\tilde{\varepsilon}_t$ has the same distribution as ε_t. Meanwhile, we consider estimating PE by cross-validation. Here the training sample is $\{(\boldsymbol{x}_j, y_j); j = p+1, \ldots, n, j \neq t\}$ and the test sample is $\{(\boldsymbol{x}_t, y_t)\}$. We leave out the entire row of matrix (2) corresponding to the test set. An estimator of PE using cross-validation is

$$\widehat{\mathrm{PE}} = \frac{1}{n-p} \sum_{t=p+1}^{n} \{y_t - g(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_{-t})\}^2,$$

where $\hat{\boldsymbol{\theta}}_{-t}$ is the leave-one-out estimate of $\boldsymbol{\theta}$, which is the solution to the following set of nonlinear equations

$$q(\hat{\boldsymbol{\theta}}_{-t}) = \sum_{\substack{j=p+1 \\ j\neq t}}^{n} \left(\frac{\partial}{\partial \boldsymbol{\theta}} g(\boldsymbol{x}_j, \hat{\boldsymbol{\theta}}_{-t}) \right) \left(y_j - g(\boldsymbol{x}_j, \hat{\boldsymbol{\theta}}_{-t}) \right) = \sum_{\substack{j=p+1 \\ j\neq t}}^{n} \left(\frac{\partial}{\partial \boldsymbol{\theta}} g(\boldsymbol{x}_j, \hat{\boldsymbol{\theta}}_{-t}) \right) \varepsilon_j(\hat{\boldsymbol{\theta}}_{-t}) = 0,$$

where $q(\hat{\boldsymbol{\theta}}_{-t})$ is obtained from differentiating $Q_{-t}(\theta)(= \sum_{\substack{j=p+1 \\ j\neq t}}^{n} (y_j - g(\boldsymbol{x}_j, \boldsymbol{\theta}))^2)$ with respect to θ.

In order to make the cross-validation work, $\widehat{\text{PE}}$ should approximate PE closely. To this end, we need a set of assumptions as follows.

Assumptions

A1 (**Stationary nonlinear AR**(p) **process**) A sequence of $\{y_t\}_{t=1}^n$ is represented by the nonlinear AR(p) process as in (3). The nonlinear AR(p) process is assumed to be stationary and ergodic.
A2 (**Consistency of the Nonlinear Least Squares**) $\hat{\boldsymbol{\theta}}_{-t}$ is a consistent estimator of $\boldsymbol{\theta}$.
A3 (**Errors are MDS**) The disturbances $\{\varepsilon_t\}$ satisfy the following.

(i) $\{\varepsilon_t, \mathscr{F}_t\}$ form a sequence of martingale differences (MDS) where $\mathscr{F}_t$ is the sigma field generated by $\{\varepsilon_s, y_s; s \leq t\}$. Note that i.i.d errors are MDS.
(ii) $\text{var}(\varepsilon_t|\boldsymbol{x}_t) = \sigma^2$ and $\text{E}|\varepsilon_t|^{4+\delta} < K$ for some $K < \infty$ and $\delta > 0$.
(iii) $\{\varepsilon_t\}$ have absolutely continuous distribution with respect to Lebesgue measure. This assumption is satisfied with errors typically used, including normally distributed errors.

Assumption **A1** states our model to which the CV is applied to, and ensures that no misspecification is considered in our setup. Note that (3) is a standard AR(p) process and our setup covers common applications of CV for dependent data. Moreover, under certain regularity conditions, the sequence $\{y_t\}$ satisfies the strong-mixing condition below (see Mokkadem, 1988, for more details):

(a) $|P(A \cap B) - P(A)P(B)| \leq \alpha(k)P(A)$, for all $A \in \mathscr{F}_t$ and $B \in \mathscr{F}_{t+k}$ for any t and k,
(b) $\sum_{k\geq 1}\alpha(k) < \infty$.

Assumption **A2** and **A3** are required to make sure that the estimator is consistent and the error structure can be such that the proposed CV method could work. Note that due to stationarity of $\{y_t\}_{t=1}^n$, conditions for the consistency of $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}_{-t}$ are equivalent. Regarding **A2**, for the leave-one-out estimator to be consistent, there are a variety of conditions suggested in the literature, e.g. Andrews (1987). For instance, the following set of conditions ensures the consistency of $\hat{\boldsymbol{\theta}}_{-t}$.

(i) The parameter space Θ in which $\boldsymbol{\theta}$ belongs to is a compact subset of $\mathbb{R}^k$.
(ii) $g(\cdot, \boldsymbol{\theta})$ is continuous and differentiable in $\boldsymbol{\theta}$ for all $(y_t, \boldsymbol{x}_t)$.
(iii) Define $Q_{-t}(\boldsymbol{\theta})$ as

$$Q_{-t}(\boldsymbol{\theta}) = \sum_{\substack{j=p+1 \\ j\neq t}}^{n} (y_j - g(\boldsymbol{x}_j, \boldsymbol{\theta}))^2.$$

Then, $\frac{1}{n-p-1}Q_{-t}(\boldsymbol{\theta}) \xrightarrow{p} Q(\boldsymbol{\theta})$ uniformly in $\boldsymbol{\theta}$, where $Q(\boldsymbol{\theta})$ is a nonstochastic function which attains a unique minimum at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$.

A3.(i) ensures that errors are serially uncorrelated, which will be the key component for the success of the cross-validation. It is worth noting that **A2** and **A3** are not strong assumptions but rather are standard ones, e.g., when Machine Learning methods are used for prediction. For instance, the i.i.d. errors are nested in the martingale difference sequence. However, errors with discrete values, say $\varepsilon_t \in \{-1, 1\}$ do not satisfy **A3**.

Theorem 1. *Suppose that Assumptions **A1**–**A3** hold. Then,*

$$\widehat{PE} \xrightarrow{p} PE. \tag{4}$$

Proof. See Appendix. ■

Additionally, we consider what would happen if the aforementioned assumptions are violated. To begin with, as the employed CV method makes use of features of residuals, residuals should emulate error terms for the success of this type of CV. Suppose that **A1** is violated, that is, the model is misspecified. The residuals will no longer have zero serial correlation. Then, the violation of **A3** follows naturally since ε_t is no longer a MDS, which renders Lemma 1 in the appendix invalid. Therefore, the CV does not work any longer. This is stated implicitly in our proof since we assume that $y_t = g(\boldsymbol{x}_t, \boldsymbol{\theta}) + \varepsilon_t$

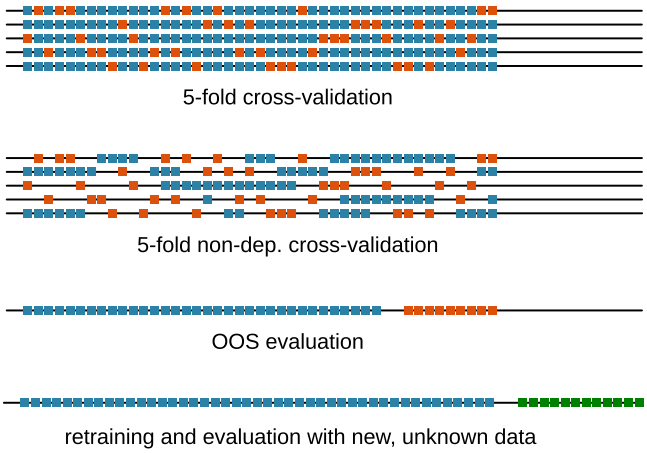

Fig. 2. Training and test sets used for the experiments. The blue and orange dots represent values in the training and test set, respectively. The green dots represent future data not available at the time of model building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where ε_t is a MDS. Because **A2** ensures that $\widehat{\text{PE}}$ approximates PE closely through consistency of the OLS estimator of $\boldsymbol{\phi}$, its violation invalidates the proposed CV not only for the dependent setup but also even for the independent setup.

Finally, we also note that the model (3) is quite general. Provided that the errors are serially uncorrelated, Theorem 1 and its proof ensure that the proposed CV is valid for a linear or nonlinear parametric regression. Its validity is extended to models considered "nonparametric" such as regression splines, smoothing splines, locally polynomial regression, and neural networks, which can all be represented by (3).

4. Monte Carlo experimental study

We perform Monte Carlo experiments illustrating the consequences of Theorem 1. In the following, we discuss the general setup of our experiments, as well as the error measures, model selection procedures, forecasting algorithms, and data generating processes employed. The experiments are performed using the R programming language (R Core Team, 2014).

4.1. General setup of the experiments

The experimental design is based on the setup of Bergmeir et al. (2014). Each time series is partitioned into a set available to the forecaster, called the *in-set*, and a part from the end of the series not available at this stage (the *out-set*), which is considered the unknown future. In our experiments, we use series with a total length of 200 values, and we use 70% of the data (140 observations) as in-set, the rest (60 observations) is withheld as the out-set. The in-set is partitioned according to the model selection procedure, models are built and evaluated, and $\widehat{\text{PE}}$ is calculated. In this way, we get an estimate of the error on the in-set. Then, models are built using all data from the in-set, and evaluated on the out-set data. The error on the out-set data is considered the true error PE, which we are estimating by $\widehat{\text{PE}}$. Fig. 2 illustrates the procedure.

Analogous to the theoretical proof, we can calculate PE as a mean squared error (MSE) as follows:

$$\text{PE}(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}}) = \frac{1}{n}\sum_{t=1}^{n}\{y_{T+t} - \boldsymbol{g}(\boldsymbol{x}_{T+t}, \hat{\boldsymbol{\theta}})\}^2, \tag{5}$$

where T is the end of the in-set and hence $\{y_{T+t}, \boldsymbol{x}_{T+t}\}$, $t = 1, 2, \ldots, n$, comprises the out-set.

The question under consideration is how well $\widehat{\text{PE}}$ estimates PE. We evaluate this by assessing the error between $\widehat{\text{PE}}$ and PE, across all series involved in the experiments. We use a mean absolute error (MAE) to assess the size of the effect and call this measure "mean absolute predictive accuracy error" (MAPAE). It is calculated in the following way:

$$\text{MAPAE} = \frac{1}{m}\sum_{j=1}^{m}\left|\widehat{\text{PE}}_j(\hat{\boldsymbol{\theta}}_{-t}, \boldsymbol{x}_t) - \text{PE}_j(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}})\right|,$$

where m is the number of series in the Monte-Carlo study. Furthermore, to see if any bias is present, we use the mean of the predictive accuracy error (MPAE), defined analogously as

$$\text{MPAE} = \frac{1}{m}\sum_{j=1}^{m}\left(\widehat{\text{PE}}_j(\hat{\boldsymbol{\theta}}_{-t}, \boldsymbol{x}_t) - \text{PE}_j(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}})\right).$$

4.2. Error measures

For the theoretical proof it was convenient to use the MSE, but in the experiments we use the root mean squared error (RMSE) instead, defined as

$$\mathrm{PE}^{\mathrm{RMSE}}(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}}) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\{y_{T+t} - \boldsymbol{g}(\boldsymbol{x}_{T+t}, \hat{\boldsymbol{\theta}})\}^2}. \tag{6}$$

The RMSE is more common in applications as it operates on the same scale as the data, and as the square root is a bijective function on the non-negative real numbers, the developed theory holds also for the RMSE. Furthermore, we also perform experiments with the MAE, to explore if the theoretical findings can apply to this error measure as well. In this context, the MAE is defined as

$$\mathrm{PE}^{\mathrm{MAE}}(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{x}}) = \frac{1}{n}\sum_{t=1}^{n}\left|y_{T+t} - \boldsymbol{g}(\boldsymbol{x}_{T+t}, \hat{\boldsymbol{\theta}})\right|. \tag{7}$$

4.3. Model selection procedures

The following model selection procedures are used in our experiments:

5-fold CV This denotes normal 5-fold CV, for which the rows of an embedded time series are randomly assigned to folds.

LOOCV Leave-one-out CV. This is very similar to the *5-fold CV* procedure, but the number of folds is equal to the number of rows in the embedded matrix, so that each fold consists of only one row of the matrix.

nonDepCV Non-dependent CV. The same folds are used as for the *5-fold CV*, but rows are removed from the training set if they have a lag distance smaller than p from a row in the test set, where p is the maximal model order (5 in our experiments).

OOS Classical out-of-sample evaluation, where a block of data from the end of the series is used for evaluation.

4.4. Forecasting algorithms

In the experiments, one-step-ahead prediction is considered. We fit linear autoregressions with up to 5 lagged values, i.e., AR(1) to AR(5) models. Furthermore, as a non-linear method we use a standard multi-layer perceptron (MLP) neural network model, available in R in the package `nnet`. It uses the BFGS algorithm for model fitting, and we set the parameters of the network to a size of 5 hidden units and a weight decay of 0.00316. For the MLP, up to 5 lagged values are used.

We use these relatively simple methods because they are sufficient to show that CV works in this context. The experiments could be easily expanded to other popular methods, such as support vector regression, Random Forests, etc. It is worth noting that the results would not change much. This is because, irrespective of models and methods used for forecasting, our approach is concerned with the residuals (see Assumption **A3** in Section 3), and therefore the disturbance term or noise, which have a universal set of features such as zero mean and zero correlations between errors. This is important in that those features ensure the justification of using the standard K-fold CV without modification.

4.5. Data generating processes

We implement three different experiments, each involving 1000 Monte Carlo trials. In the first two experiments, we generate data from stationary AR(3) processes and invertible MA(1) processes, respectively. We use the stochastic design of Bergmeir et al. (2014), so that for each Monte Carlo trial, new coefficients for the data generating process (DGP) are generated, and we can explore larger areas of the parameter space and achieve more general results, not related to the parameters/coefficients of a particular DGP.

The purpose of Experiment 1 with AR(3) processes is to illustrate how the methods perform when the true model, or very similar models to the DGP, are used for forecasting. Experiment 2 shows a situation in which the true model is not among the forecasting models, but the models can still reasonably well fit the data. This is the case for an MA process which can approximate an AR process with a large number of lags. In practice, usually a relatively low number of AR lags is sufficient to model such data.

The third experiment is a counterexample; i.e., a situation where the CV procedures break down. We use a seasonal AR process as the DGP with a significant lag 12 (seasonal lag 1). As the models taken into account only use up to the first five lags, the models should not be able to fit well such data. We obtain the parameters for the DGP by fitting a seasonal AR model to a time series that shows monthly totals of accidental deaths in the USA, from 1973 to 1978 (Brockwell and Davis, 1991). This dataset is included in a standard installation of R. It is illustrated in Fig. 3. We use the seasonal AR model as a DGP in the Monte Carlo experiments.

All series in the experiments are made entirely positive by subtracting the minimum and adding 1.

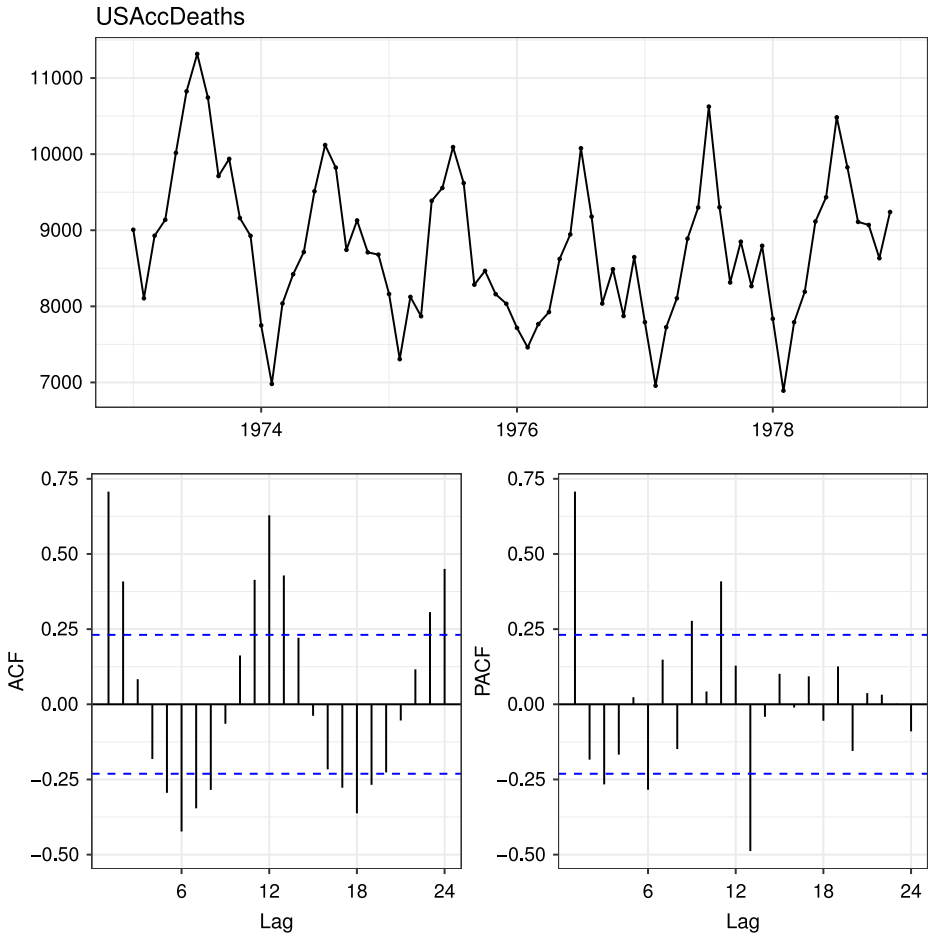

Fig. 3. Series used to obtain a DGP for the Monte Carlo experiments. The ACF and PACF plots clearly show the monthly seasonality in the data.

5. Results and discussion

In the following, we present results and offer discussions of the findings.

5.1. Results for linear model fitting

The top panel of Table 1 shows the results for Experiment 1, where AR(3) processes are used as DGPs. We see that for RMSE, the *5-fold CV* and *LOOCV* procedures achieve values around 0.09 for MAPAE, whereas the *OOS* procedure has higher values around 0.16, so that the CV procedures achieve more precise error estimates. The *nonDepCV* procedure performs considerably worse, which is due to the fact that the fitted models are less accurate as they are fitted with less data. Regarding the MPAE, *LOOCV* achieves consistently low-biased estimates with absolute values smaller than 0.003, whereas *OOS* and *5-fold CV* have absolute values up to 0.01 and 0.007, respectively, comparable to each other. The findings hold in a similar way for the MAE.

The middle panel of Table 1 shows the results for Experiment 2, where we use MA(1) processes as the DGPs. Regarding the MAPAE, we see similar results as in Experiment 1; i.e., the CV procedures yield more precise error estimates than the *OOS* procedure. Regarding the MPAE, *OOS* now slightly outperforms the CV procedures with absolute values between 0.003 and 0.01. The CV procedures have values between 0.008 and 0.013, and 0.005 and 0.011, respectively. Similar findings hold for the MAE error measure. The *nonDepCV* procedure again is not competitive.

Finally, the bottom panel of Table 1 shows the results for Experiment 3. In this experiment, where all models are heavily misspecified, we see that the advantage of the CV procedures w.r.t. MAPAE has nearly vanished, and the CV estimates are more biased than the estimates obtained with the *OOS* procedure.

Table 1
Fitted model: linear AR. Series length: 200.

	# Lags	RMSE		MAE	
		MAPAE	MPAE	MAPAE	MPAE
DGP: AR(3)					
5-fold CV	AR(1)	0.098	−0.000	0.084	−0.004
	AR(2)	0.089	0.004	0.078	0.000
	AR(3)	0.090	0.006	0.077	0.001
	AR(4)	0.092	0.006	0.078	0.001
	AR(5)	0.094	0.007	0.080	0.001
LOOCV	AR(1)	0.098	−0.002	0.084	−0.005
	AR(2)	0.089	0.002	0.077	−0.002
	AR(3)	0.090	0.002	0.077	−0.002
	AR(4)	0.091	0.001	0.078	−0.003
	AR(5)	0.093	0.001	0.079	−0.003
nonDepCV	AR(1)	0.423	0.411	0.283	0.271
	AR(2)	0.510	0.505	0.341	0.336
	AR(3)	0.630	0.628	0.419	0.418
	AR(4)	1.014	1.014	0.620	0.619
	AR(5)	6.137	6.137	2.580	2.580
OOS	AR(1)	0.170	−0.010	0.143	−0.002
	AR(2)	0.157	−0.004	0.134	0.002
	AR(3)	0.158	−0.002	0.135	0.003
	AR(4)	0.160	−0.002	0.136	0.003
	AR(5)	0.163	−0.001	0.139	0.003
DGP: MA(1)					
5-fold CV	AR(1)	0.113	0.013	0.096	0.007
	AR(2)	0.106	0.008	0.090	0.003
	AR(3)	0.102	0.012	0.087	0.007
	AR(4)	0.100	0.009	0.085	0.005
	AR(5)	0.100	0.012	0.085	0.006
LOOCV	AR(1)	0.113	0.011	0.096	0.005
	AR(2)	0.105	0.005	0.090	0.001
	AR(3)	0.101	0.009	0.086	0.004
	AR(4)	0.099	0.005	0.084	0.001
	AR(5)	0.098	0.007	0.084	0.002
nonDepCV	AR(1)	0.264	0.244	0.201	0.179
	AR(2)	0.359	0.352	0.266	0.258
	AR(3)	0.482	0.480	0.343	0.340
	AR(4)	0.862	0.861	0.545	0.543
	AR(5)	10.225	10.224	4.038	4.037
OOS	AR(1)	0.192	−0.010	0.161	−0.002
	AR(2)	0.181	−0.006	0.153	−0.001
	AR(3)	0.173	−0.003	0.145	0.003
	AR(4)	0.171	−0.003	0.144	0.004
	AR(5)	0.171	−0.005	0.143	0.001
DGP: AR(12)					
5-fold CV	AR(1)	150.890	−43.549	128.103	−41.810
	AR(2)	154.210	−50.193	129.217	−46.432
	AR(3)	158.004	−59.821	132.424	−54.444
	AR(4)	166.364	−80.904	139.967	−71.794
	AR(5)	172.824	−95.194	145.233	−83.965
LOOCV	AR(1)	150.661	−44.063	127.822	−42.250
	AR(2)	154.152	−52.161	129.122	−47.950
	AR(3)	157.858	−62.983	132.123	−56.868
	AR(4)	166.496	−84.866	139.968	−74.659
	AR(5)	173.410	−100.682	145.731	−88.044
nonDepCV	AR(1)	206.934	126.776	159.916	84.506
	AR(2)	245.753	187.501	181.995	126.540
	AR(3)	332.597	292.792	223.966	184.539
	AR(4)	690.263	664.060	382.009	354.904
	AR(5)	8101.953	8090.237	3090.719	3077.161

(*continued on next page*)

Table 1 (*continued*)

	# Lags	RMSE		MAE	
		MAPAE	MPAE	MAPAE	MPAE
OOS	AR(1)	157.690	−25.556	135.948	−16.516
	AR(2)	161.896	−28.484	138.869	−18.247
	AR(3)	165.107	−34.500	140.313	−22.344
	AR(4)	171.390	−40.088	145.932	−25.820
	AR(5)	177.577	−42.417	152.985	−28.486

5.2. Results for MLP model fitting

Table 2 shows the analogous results where neural networks have been used for forecasting. The experiments essentially confirm the findings for the linear models. If only one lagged value is used, the model fitting procedure has difficulties and the resulting models are not competitive, yielding high values of both MAPAE and MPAE throughout all model selection procedures and experiments.

For the first experiment, the CV methods show advantages in the sense that they yield more precise error estimates (lower MAPAE), and a comparable bias (as measured by MPAE) compared to the *OOS* procedure.

These advantages are also seen in Experiment 2. For Experiment 3, where models are heavily misspecified, the advantages of the CV procedures for MAPAE have mainly vanished and the disadvantages of high bias prevail.

6. Real-world data example

In addition to the Monte Carlo experiments, we perform in this section a study with a real-world data series. We use the well-known yearly sunspot series, available, e.g., from Tong (1993) and in R as series `sunspot.year` in the base package. It contains the amounts of yearly sunspots from 1700 to 1988, with 289 observations in total. Following Tong (1993), we transform the data as a preprocessing step as follows:

$$y_t = 2(1 + x_t)^{0.5} - 1.$$

Here, x_t is the original time series and y_t is the transformed version, shown in Fig. 4.

Following the configuration of our Monte Carlo experiments, we use 30% of the data as out-set, perform *5-fold CV*, and perform *OOS* evaluation accordingly with 20% of the in-set data. Namely, the out-set contains the last 86 values of the series, and the *OOS* test set contains the last 40 values of the in-set.

Tong (1993) describes that the AIC criterion for a linear model chooses a lag order of 9, and that non-linear prediction methods outperform linear ones for this time series. In particular, he fits a self-exciting threshold AR model of order 11.

Following the suggestion that non-linear models are appropriate for this series, we focus here on the neural network model as used in our Monte Carlo experiments. This choice is also motivated by the fact that we are not primarily interested in model performance but we want to evaluate the model selection procedures, and therewith a model with more hyper-parameters will allow better insights. We use the following parameter grid, which is in line with the Monte Carlo experiments and has been used in similar experiments before (Bergmeir and Benítez, 2012). The parameter grid is built from size $= \{3, 4, 5, \ldots, 15\}$, decay $= \{0, 0.00316, 0.0147, 0.05, 0.075, 0.1\}$, and maximal lag order $p = \{1, 2, 3, \ldots, 20\}$. As model selection procedures, we use *5-fold CV* and *OOS* as in the Monte Carlo experiments.

The cross-validation gives us a genuine out-of-sample prediction for each data point. We use the predictions across all folds to build a residual series, and apply the Ljung–Box test (Ljung and Box, 1978 implemented in R in function `Box.test`) to this series. As we are considering out-of-sample predictions, we set degrees of freedom to 0. As mentioned earlier, in the literature models with a maximal lag order of 11 are fitted to this series, so we set the amount of lags in the Ljung–Box test to 20, to capture relevant remaining autocorrelation.

All possible combinations from the defined hyper-parameter grid give us 1560 different model configurations in total. For each of these model configurations, we get a residual series as described above that is checked with the Ljung–Box test for serial correlation. In total, the residual series of 763 model configurations pass the test. From these model configurations, we choose the configuration that results in the minimal cross-validated RMSE, and the one that results in the minimal *OOS* RMSE. Results are shown in Table 3.

The model chosen by *5-fold CV* has parameters $p = 7$ and size $= 3$, and results in an out-set error of 2.247. Compare this to the model that is selected by the *OOS* procedure, which has parameters $p = 3$ and size $= 9$, and a slightly higher out-set error of 2.281. We note that the latter model configuration is the configuration that gives the best *OOS* error not only among configurations that pass the Ljung–Box test, but among all the configurations used in the experiments.

We see that the cross-validation procedure chooses a model configuration that has a reasonable lag structure and is with only 3 hidden units not overfitting. The model selected using *OOS* uses less lags and more hidden units, which seems to be an inferior choice, however, both model configurations perform similar on the out-set, with the model chosen by *OOS* yielding to a slightly higher out-set error.

Table 2
Fitted model: Neural networks. Series length: 200.

	# Lags	RMSE		MAE	
		MAPAE	MPAE	MAPAE	MPAE
DGP: AR(3)					
5-fold CV	AR(1)	0.769	−0.724	0.665	−0.632
	AR(2)	0.151	0.027	0.107	0.009
	AR(3)	0.195	0.040	0.133	0.017
	AR(4)	0.207	0.057	0.135	0.028
	AR(5)	0.258	0.075	0.159	0.040
LOOCV	AR(1)	0.769	−0.727	0.663	−0.632
	AR(2)	0.152	0.009	0.109	−0.005
	AR(3)	0.170	0.028	0.118	0.005
	AR(4)	0.201	0.033	0.129	0.012
	AR(5)	0.205	0.018	0.135	−0.004
nonDepCV	AR(1)	0.580	−0.109	0.505	−0.207
	AR(2)	0.844	0.838	0.606	0.605
	AR(3)	0.885	0.862	0.659	0.643
	AR(4)	0.842	0.826	0.643	0.639
	AR(5)	0.771	0.750	0.603	0.597
OOS	AR(1)	0.818	−0.729	0.693	−0.619
	AR(2)	0.232	0.008	0.180	0.013
	AR(3)	0.262	0.011	0.198	0.016
	AR(4)	0.295	0.002	0.215	0.013
	AR(5)	0.326	0.013	0.240	0.024
DGP: MA(1)					
5-fold CV	AR(1)	0.289	−0.253	0.252	−0.228
	AR(2)	0.159	0.027	0.114	0.012
	AR(3)	0.182	0.068	0.125	0.043
	AR(4)	0.187	0.061	0.127	0.038
	AR(5)	0.204	0.065	0.132	0.040
LOOCV	AR(1)	0.296	−0.265	0.258	−0.237
	AR(2)	0.157	0.012	0.115	0.002
	AR(3)	0.178	0.019	0.122	0.007
	AR(4)	0.185	0.030	0.124	0.017
	AR(5)	0.176	0.000	0.120	−0.012
nonDepCV	AR(1)	0.352	0.215	0.256	0.107
	AR(2)	0.712	0.704	0.533	0.531
	AR(3)	0.761	0.755	0.589	0.587
	AR(4)	0.727	0.719	0.578	0.575
	AR(5)	0.659	0.649	0.534	0.532
OOS	AR(1)	0.368	−0.285	0.305	−0.239
	AR(2)	0.247	0.005	0.191	0.013
	AR(3)	0.267	0.015	0.198	0.021
	AR(4)	0.276	0.006	0.206	0.016
	AR(5)	0.284	0.042	0.217	0.047
DGP: AR(12)					
5-fold CV	AR(1)	154.137	−37.919	130.981	−37.587
	AR(2)	157.743	−49.764	132.260	−45.950
	AR(3)	152.300	−38.585	127.494	−38.171
	AR(4)	155.298	−55.475	130.786	−51.674
	AR(5)	158.104	−62.537	132.781	−58.896
LOOCV	AR(1)	153.873	−38.041	130.153	−37.107
	AR(2)	162.565	−63.654	135.013	−56.950
	AR(3)	169.791	−82.204	140.290	−70.038
	AR(4)	163.306	−69.063	136.467	−61.480
	AR(5)	164.931	−79.233	136.795	−70.160
nonDepCV	AR(1)	195.267	103.594	156.112	68.756
	AR(2)	195.145	96.722	155.378	64.679
	AR(3)	204.659	125.448	158.247	84.718
	AR(4)	208.368	136.821	162.852	96.170
	AR(5)	205.205	125.914	162.812	89.028

(*continued on next page*)

Table 2 (*continued*)

	# Lags	RMSE		MAE	
		MAPAE	MPAE	MAPAE	MPAE
OOS	AR(1)	159.824	−23.745	137.788	−15.963
	AR(2)	163.665	−33.175	139.471	−21.946
	AR(3)	168.314	−17.899	141.165	−9.545
	AR(4)	173.594	−28.360	144.881	−17.344
	AR(5)	179.041	−29.841	150.671	−18.436

Table 3
Chosen model configurations and out-set errors on the yearly sunspot data, using 5-fold CV and OOS evaluation.

	Parameters			RMSE
	p	Size	Decay	Out-set
CV	7	3	0.1	2.247
OOS	3	9	0	2.281

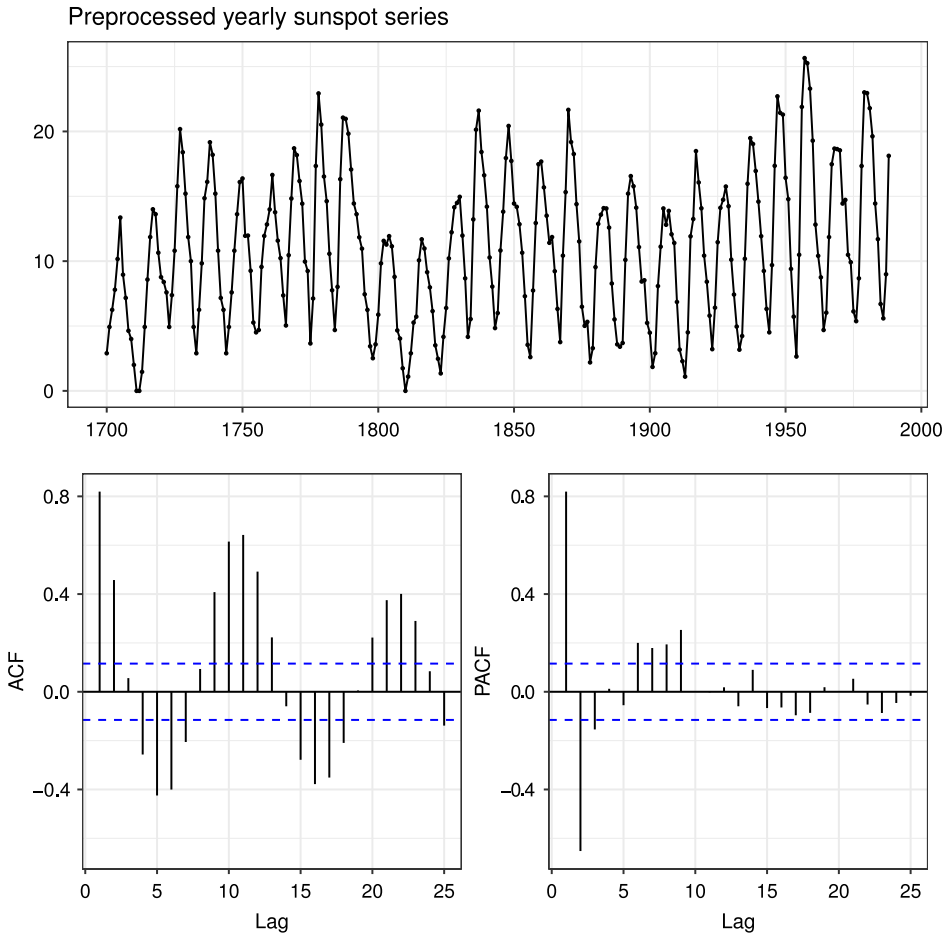

Fig. 4. Yearly sunspot series after preprocessing.

7. Conclusions

In this work we have investigated the use of cross-validation procedures for time series prediction evaluation when purely autoregressive models are used, which is a very common situation; e.g., when using Machine Learning procedures for time series forecasting. In a theoretical proof, we have shown that a normal K-fold cross-validation procedure can be used if the residuals of our model are uncorrelated, which is especially the case if the model nests an appropriate model. In

the Monte Carlo experiments, we have shown empirically that even if the lag structure is not correct, as long as the data are fitted well by the model, cross-validation without any modification is a better choice than OOS evaluation. We have then in a real-world data example shown how these findings can be used in a practical situation. Cross-validation can adequately control overfitting in this application, and only if the models underfit the data and lead to heavily correlated errors, are the cross-validation procedures to be avoided as in such a case they may yield a systematic underestimation of the error. However, this case can be easily detected by checking the residuals for serial correlation, e.g., using the Ljung–Box test.

Acknowledgement

Koo acknowledges the financial assistance of the Australian Research Council Discovery Grant No. DP150104292.

Appendix. Proof of Theorem 1

Proof of Theorem 1. Suppose that **A1** is satisfied and hence the order of the autoregressive process is p. Following Burman and Nolan (1992),

$$\begin{aligned}\mathrm{PE} &= \int \Big[g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - \tilde{y}\Big]^2 dF \\ &\approx \int \Big[g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - g(\tilde{\boldsymbol{x}}, \boldsymbol{\theta})\Big]^2 dF + \int \tilde{\varepsilon}^2 dF \\ &= \int \Big[g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - \mathrm{E}g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) + \mathrm{E}g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - g(\tilde{\boldsymbol{x}}, \boldsymbol{\theta})\Big]^2 dF + \int \tilde{\varepsilon}^2 dF,\end{aligned}$$

where F is the distribution of the process $\{\tilde{y}_k\}_{k=1}^n$. Therefore, with a bit of algebra, PE becomes

$$\int \Big[g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - \mathrm{E}g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}})\Big]^2 dF + \int \Big[\mathrm{E}g(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) - g(\tilde{\boldsymbol{x}}, \boldsymbol{\theta})\Big]^2 dF + \int \tilde{\varepsilon}^2 dF, \tag{A.1}$$

whereas, in a similar vein, $\widehat{\mathrm{PE}}$ matches

$$\frac{1}{n-p}\sum_{t=p+1}^{n}\Big[g(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_{-t}) - \mathrm{E}g(\boldsymbol{x}_t, \hat{\boldsymbol{\theta}}_{-t})\Big]^2 + \int \Big[\mathrm{E}g(\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{-t}) - g(\boldsymbol{x}, \boldsymbol{\theta})\Big]^2 dF_n + \int \varepsilon^2 dF_n, \tag{A.2}$$

where F_n is the empirical distribution of the test sample. The second and third terms of the above two equations, (A.1) and (A.2) are asymptotically identical. So we can focus on the first term of each equation. The first term of (A.1) becomes

$$\frac{1}{(n-p)^2}\sum_{t=p+1}^{n}\sum_{j=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) = \frac{1}{n-p}\sum_{j=p+1}^{n}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) + \frac{1}{(n-p)(n-p-1)}\sum_{\substack{j=p+1\\ j\neq t}}^{n}\sum_{t=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}), \tag{A.3}$$

whereas the first term of (A.2) is $(n-p)^{-1}\sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t})$. Intuitively, the probability limit of the first term in (A.3) is asymptotically equivalent to the probability limit of the first term of (A.2), and, due to leaving the entire row out, the second term of (A.3) is a martingale difference sequence that converges to zero in probability. This will be shown in Lemma 1. It is worth noting that this is compatible with the condition Burman and Nolan (1992) provide under their setup; that is, for any $t < j$,

$$\mathrm{E}\left[\varepsilon_t\varepsilon_j|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_j\right] = 0. \tag{A.4}$$

That is, the use of standard cross-validation in our setup is valid as long as

$$\mathrm{E}\sum_{t=p+1}^{n}\sum_{j=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) \approx \mathrm{E}\sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t}), \tag{A.5}$$

which follows from Lemma 1. ■

Lemma 1. *Suppose that **A1–A3** hold. Then, there exists an arbitrarily small constant $c > 0$ such that*

$$E\left[\sum_{t=p+1}^{n}\sum_{j=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) - \sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t})\right]^4 \leq cn^4.$$

Proof of Lemma 1. Note that

$$\begin{aligned}
&\sum_{t=p+1}^{n}\sum_{j=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) - \sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t}) \\
&= \underbrace{\sum_{j=p+1}^{n}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t})}_{A.6.1} + \underbrace{\sum_{\substack{j=p+1\\ j\neq t}}^{n}\sum_{t=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}})}_{A.6.2} \qquad \text{(A.6)} \\
&= A.6.1 + A.6.2.
\end{aligned}$$

For $A.6.1$,

$$\begin{aligned}
\sum_{j=p+1}^{n}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \sum_{j=p+1}^{n}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t}) &= \underbrace{\sum_{j=p+1}^{n}\left(\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \mathrm{E}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}})\right)}_{A.6.1.1} - \underbrace{\sum_{j=p+1}^{n}\left(\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t}) - \mathrm{E}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t})\right)}_{A.6.1.2} \\
&\quad + \underbrace{\sum_{j=p+1}^{n}\left(\mathrm{E}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \mathrm{E}\varepsilon_j^2(\hat{\boldsymbol{\theta}}_{-t})\right)}_{A.6.1.3}.
\end{aligned}$$

Note that $\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})$ and $\varepsilon_j(\hat{\boldsymbol{\theta}}_{-t})$ have the same distribution as long as both $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}_{-t}$ are consistent due to **A1** and **A2**. Therefore, $A.6.1.3$ is of smaller order than $A.6.1.1$ and $A.6.1.2$. For $A.6.1.1$, the summand are martingale difference sequences and hence using Burkholder's inequality, we obtain,

$$\mathrm{E}\left|\sum_{j=p+1}^{n}\left(\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \mathrm{E}\varepsilon_j^2(\hat{\boldsymbol{\theta}})\right)\right|^4 \leq c\mathrm{E}\left|\sum_{t=1}^{n}\left(\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}}) - \mathrm{E}\tilde{\varepsilon}_j^2(\hat{\boldsymbol{\theta}})\right)^2\right|^2 \leq cn^2.$$

For $A.6.1.2$, it is analogous. Let us turn our attention to $A.6.2$. For $A.6.2$, due to symmetry of a covariance matrix,

$$\sum_{\substack{j=p+1\\ j\neq t}}^{n}\sum_{t=p+1}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}) = 2\sum_{j=p+2}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\sum_{t=p+1}^{t<j}\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}}).$$

Therefore, the proof boils down to showing that the impact of the covariance terms, that is, sum of the off-diagonal elements of the variance–covariance matrix becomes negligible, i.e.

$$\mathrm{E}\left[\sum_{\substack{j=p+1\\ j\neq t}}^{n}\sum_{t=1}^{n}\varepsilon_j(\hat{\boldsymbol{\theta}})\varepsilon_t(\hat{\boldsymbol{\theta}})\right]^2 = 2\mathrm{E}\left[\sum_{j=p+2}^{n}\tilde{\varepsilon}_j(\hat{\boldsymbol{\theta}})\sum_{t=p+1}^{t<j}\tilde{\varepsilon}_t(\hat{\boldsymbol{\theta}})\right]^2 \leq cn^4. \qquad \text{(A.7)}$$

Note that $\{\tilde{\varepsilon}_t\}$ and $\{\varepsilon_t\}$ are stationary martingale difference sequences and any linear combination of martingales defined on the same filtration is also a martingale. The proposed CV method exactly works towards ensuring the uncorrelatedness between residuals. Once omitting rows of the matrix and applying the law of iterated expectation, (A.7) holds in the similar fashion as in the proof of Lemma 5.3 as in Burman and Nolan (1992). ■

References

Andrews, D.W., 1987. Consistency in nonlinear econometric models: A generic uniform law of large numbers. Econometrica 1465–1471.

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

Bergmeir, C., Benítez, J.M., 2012. On the use of cross-validation for time series predictor evaluation. Inform. Sci. 191, 192–213.

Bergmeir, C., Costantini, M., Benítez, J.M., 2014. On the usefulness of cross-validation for directional forecast evaluation. Comput. Statist. Data Anal. 76, 132–143.

Borra, S., Di Ciaccio, A., 2010. Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. Comput. Statist. Data Anal. 54 (12), 2976–2989.

Brockwell, P.J., Davis, R.A., 1991. Time Series: Theory and Methods. Springer, New York.

Budka, M., Gabrys, B., 2013. Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation. IEEE Trans. Neural Netw. Learn. Syst. 24 (1), 22–34.

Burman, P., Chow, E., Nolan, D., 1994. A Cross-validatory method for dependent data. Biometrika 81 (2), 351–358.

Burman, P., Nolan, D., 1992. Data-dependent estimation of prediction functions. J. Time Series Anal. 13 (3), 189–207.

Györfi, L., Härdle, W., Sarda, P., Vieu, P., 1989. Nonparametric Curve Estimation from Time Series. Springer Verlag, Berlin.

Hastie, T., Tibshirani, R., Friedman, J., 2009. Elements of Statistical Learning. Springer, New York.

Hyndman, R.J., Koehler, A.B., Ord, J.K., Snyder, R.D., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer, Berlin, URL http://www.exponentialsmoothing.net.
Kunst, R., 2008. Cross validation of prediction models for seasonal time series by parametric bootstrapping. Austral. J. Statist. 37, 271–284.
Ljung, G.M., Box, G.E.P., 1978. On a measure of lack of fit in time series models. Biometrika 297–303.
McQuarrie, A.D.R., Tsai, C.-L., 1998. Regression and time series model selection. World Scientific Publishing.
Mokkadem, A., 1988. Mixing properties of ARMA processes. Stochastic Process. Appl. 29 (2), 309–315.
Moreno-Torres, J., Saez, J., Herrera, F., 2012. Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Trans. Neural Netw. Learn. Syst. 23 (8), 1304–1312.
Opsomer, J., Wang, Y., Yang, Y., 2001. Nonparametric regression with correlated errors. Statist. Sci. 16 (2), 134–153.
R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.
Racine, J., 2000. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. J. Econometrics 99 (1), 39–61.
Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. J. R. Stat. Soc. Ser. B Stat. Methodol. 36 (2), 111–147.
Tong, H., 1993. Non-linear Time Series: A Dynamical System Approach. Clarendon Press.