The validity of backtesting for evaluation of autoregressive time series predictions

by
Michael Fotakis 536534
MSc Tilburg University

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Econometrics and Mathematical Economics

Supervisor:
Denis Kojevnikov

May 2021

Contents

Acknowledgements

First and foremost I would like to thank my thesis supervisor Dr. Denis Kojevnikov for his invaluable support during this master thesis. Without his advice, directions and assistance this endeavour would not have been possible. My gratitude extends to Dr. Christoph Bergmeir of Monash University for his useful input regarding the coding part of my thesis. Additionally, I would like to thank Aegon Nederland's Model Validation department and especially Pieter-Jan whose support during my internship at the department and feedback on my thesis played no small part in its completion. Last but not least, I would like to sincerely thank my family and friends, Costis, Nitsa, George, Ilka, Dimitris and Nikos who supported me throughout my studies.

Abstract

This thesis examines the validity of backtesting in the autoregressive predictions context. A Monte Carlo simulation exercise was adopted to compare the performance of popular cross-validation methods to walk-forward testing in cases of structural breaks and autocorrelations in the error structure. Despite the limited use of the method in time series prediction evaluations, it manages to very closely compete with cross-validation in most data generating processes and even achieve lower errors when heavy error autocorrelation is present. Additionally, in small real data samples, backtesting implemented with the walk-forward method yields lower error figures and remains competitive even at bigger samples.

Chapter 1

Introduction

The term backtesting in time series modelling describes the process of testing the performance of a predictive model using historical data. This ex-post evaluation checks how well the model would have performed in terms of predicting the present with past data. This in turn could, generally speaking, provide an indicator for the suitability of the model in question for predicting the future using present data. In the time series setting, backtesting generally involves reserving some part of the series and using observations further back in time to predict for the reserved data. Contrary to traditional time series prediction methodology where all present information is used to predict the unknown, unobserved future, backtesting a time-series model involves using past information to predict present information. Since present information is available, such a prediction can easily be evaluated when compared to the actual observed quantity. As a method for evaluating time series predictions, it is closely related to cross-validation. More specifically, cross-validation is a wider concept since it is often the case that more recent data are used to predict observations of the past for evaluation purposes.

Bergmeir, Hyndman & Koo (2018) conduct a Monte Carlo simulation experiment to test similar evaluation methods for autoregressive time series such as 5-fold and leave-one-out cross-validation, cross-validation for dependent data among others. They use autoregressive models of varying orders to predict series which where generated using an autoregressive and a moving average sequence in order to examine performance under correct and incorrect specification.

This thesis assumes the same approach to test the validity of backtesting in the purely autoregressive framework implemented with the walk-forward testing method in Kaastra & Boyd (1996). This testing method, popular in the commodity trading and neural network literature, involves splitting the data in train and test sets while maintaining the natural order of the time series data. The train/test set window moves forward in time, the model is examined across all possible time windows and prediction errors and accuracy are averaged over the number of iterations (hence the relation with CV). This particular method seems particularly attractive for time series with structural changes e.g. series with an intercept change at a specific time period. Apart from examining the validity of backtesting in comparison to information criteria, this thesis will also consider how backtesting would handle the case of structural breaks and an autocorrelated error structure in the simulated data generating processes on top of the commonplace independent and identically distributed errors case.

In general, this research proposal aims to examine whether backtesting in this particular framework has potential advantages over other CV methods, how well does it perform under correct or incorrect specification, whether it responds positively to distortions encountered in real data (e.g. non-iid errors, structural changes) and how efficiently it could be applied to real-world applications.

The structure of the thesis is given here. In Chapter 2, the current literature revolving the issues discussed in this thesis are reviewed. In Chapter 3, the technical analysis is described in detail and results are presented, analyzed and explained. Finally, Chapter 4 gives the main conclusions of the thesis and discusses potential implications and recommendations for further research.

Chapter 2

Literature review

Backtesting is mainly used in the risk management sector to evaluate either the performance of a trading strategy or the quality of a firm's risk measurement. How often a risk measurement succeeds in predicting potential losses for a portfolio or whether a given strategy yields gains or losses is often tested ex-post. In other words, how many correct instances of current losses a risk measurement predicts or how well a trading strategy would perform in the present is tested exclusively with past data giving a sense of predictive quality. For example, Harvey & Liu (2015) showcase the use of backtesting for evaluating trading strategies. Similarly, Campbell (2005) gives a thorough introduction to the backtesting procedures for VaR (Value at Risk) measures. More specifically, a transformation of the returns of a portfolio into a binary function is presented for the evaluation of its performance using historical data.

In a more abstract time series setting (not necessarily financial analysis), backtesting generally involves reserving some part of the series to use as the "unknown" future and using observations further back in time to predict for the reserved data. Contrary to traditional time series prediction methodology where all present information is used to predict the unknown, unobserved future, backtesting a time-series model involves using past information to predict present information. Since present information is naturally available, such a prediction can easily be evaluated when compared to the actual observed quantity of interest. In this setting, backtesting can be implemented through training and test set splits[1] of the data and evaluating ex-post based on the observations in the test set.

It is obvious that such an evaluation design is closely related to cross-validation. More specifically, cross-validation is considered a wider concept since it is often the case that, with random sampling, more recent data are used to predict observations of the past for evaluation purposes while for backtesting the opposite is generally true. Cross-validation with time series data has been thoroughly examined in the pertinent literature e.g. McQuarrie & Tsai (1998), Racine (2000), Arlot & Celisse (2009), Bergmeir & Benítez (2012) and remains a heavily discussed topic.

In predictive neural networks literature the data is often used in a smaller sample window that moves forward in time. In this sub-sample, both training and test sets are not random, like in standard cross-validation, but their observations are kept in chronological order. The test set always comes right after the training set in time and iteration after iteration, the whole sub-sample is moved forward by one observation/time period. This procedure is mentioned as walk-forward sliding window in Kaastra & Boyd (1996) used for financial forecasting and is also present in Karathanasopoulos et al. (2016) used for predicting the crack spread and Gudelek et al. (2017) used for developing a trading model.

This sliding window approach combines generally lower error figures averaged over a number of iterations (in comparison to non-iterative methods such as OOS validation), also existent in cross-validation, with the fact that the order of the data is kept unchanged which naturally conforms to the definition of backtesting. Unchanged data order might be especially useful in the context of time series since correlation between subsequent observations is often significant. Here, the sliding window approach is presented not for neural networks but for the case of linear time series modelling. Especially interesting is the comparison between the sliding window method and the use of information criteria used for model selection when the research objective is prediction quality. Additionally, it is also examined how well this method performs compared to the cross-validation methods used for the same purpose in Bergmeir et al. (2018) where an array of random sub-sampling methods are evaluated on their forecasting performance.

The choice of the sliding window or walk forward testing approach is motivated by several factors. First, its usage in the relevant literature is quite limited. Also, while used for practical applications, its theoretical

[1] For the definition of training and test data sets see James, Witten, Hastie & Tibshirani (2013)

properties are not explored along with its potential advantages and disadvantages. Additionally, this method of validation has a particular appeal for the cases that the series of interest is one with autocorrelation present in the idiosyncratic error or even structural breaks. Even though traditional cross-validation methods have been adjusted to deal with heavy correlation between adjacent time series observations, a method that does not shuffle the data and discards a potentially smaller amount of observations might prove competitive in the autoregressive context. The latter would specifically mean more accurate predictions and consequently more consistent model selection. Such strengths as well as potential drawbacks are lacking in the current literature.

Chapter 3

Prediction evaluation design

In this chapter, the validity of the walk-forward testing procedure is examined. Firstly, a simulation exercise is conducted to examine the theoretical performance of cross-validation and walk-forward testing on synthetic data (section 3.1). Simulation results can be found in section 3.2. The comparison is extended to real data as well (real data applications are found in section 3.3).

3.1 Simulation experiments

The Monte Carlo experiment implemented in this study is an extension of the experiment conducted in Bergmeir et al. (2018). The experiment is extended to accommodate the cases where the data are generated using structural breaks and autocorrelated errors. These additional data generating processes are tested in the same way as in Bergmeir et al. (2018) using multiple autoregressive models meaning models in which the variable of interest is given as a linear function of its past values. Additionally, the walk forward testing method is also adapted into the experiment to test its performance against the cross validation methods in Bergmeir et al. (2018). The procedures are evaluated based on error metrics i.e how well the prediction method manages to predict actual values observed in the data (either simulated or real). The methodology adopted in this study for the calculation of the error measures is described in further detail in the following paragraphs.

The experiment is initiated by simulating a series from one of the data generating processes described in Bergmeir et al. (2018) and those described in Table 3.1. In all DGPs excluding those with autocorrelated errors the error structure is an independently and identically distributed stochastic process following a standard normal distribution (zero mean and variance equal to one). All DGPs are initially simulated as zero mean processes. The whole series is then made positive by subtracting its minimum and adding 1 to all observations. This transformation is applied in order not to offset the error metrics of interest with negative values.

Table 3.1: Description of the additional data generating processes used in the extended version of the experiment. Real data series are also included in the exercise and are discussed in section 3.3.

DGP	Description
AR(3) w/ break	AR(3) process consisting of two different set of random parameters for its two halves. Stationarity constraints dictate that the mean of the process should remain constant. For this reason, the different sets of parameters are such to yield a stationary process with the same mean and variance for every sub-sample with different parameters (such models with breaks expressed in non-constant parameters are also described in Chow (1960) and Lin & Teräsvirta (1994)). The stationarity of the generated process is also confirmed with ADF tests. In notation: $$y_t = \sum_{i=1}^{3} \phi_i y_{t-i} + \varepsilon_t,$$ where $\varepsilon_t \sim N(0,1)$ and $\phi_i = \phi_{i1}, i = 1, \dots \frac{T}{2}, \phi_i = \phi_{i2} \neq \phi_{i1}$ for $i = \frac{T}{2} + 1, \dots, T$ (all ϕ's yield a stationary solution of the above).

MA(1) w/ break	MA(1) process generated similarly to the AR(3) with a break process. Different parameters for the two halves of the series but statistical properties are kept unchanged. Stationarity checks are also performed. In notation: $$y_t = z_t + \theta_i z_{t-i} + \varepsilon_t,$$ where both $\varepsilon_t, z_t \sim N(0,1)$ and $\theta_i = \theta_{i1}$ for $i = 1,\ldots,\frac{T}{2}$, $\theta_i = \theta_{i2} \neq \theta_{i1}$ for $t = \frac{T}{2}+1,\ldots,T$ (θ yields a stationary solution for the above).
AR(3) w/ AR errors	In this case the error structure is generated not as a white noise but as a stationary AR(1) process whose parameters are chosen randomly in each simulation run. In notation: $$y_t = \sum_{i=1}^{3} \phi_i y_{t-i} + \varepsilon_t,$$ where ϕ_i such that the above is stationary and $\varepsilon_t = b\varepsilon_{t-1} + u_t, u_t \sim N(0,1)$.
AR(3) w/ MA errors	Error structure used follows a MA(1) process. Parameters are again chosen randomly. In notation: $$y_t = \sum_{i=1}^{3} \phi_i y_{t-i} + \varepsilon_t,$$ where ϕ_i such that the above is stationary and $\varepsilon_t = v_t + bv_{t-1} + u_t, v_t, u_t \sim N(0,1)$.
MA(1) w/ AR errors	Moving average process of order one. Error structure is a stationary AR(1) process. Parameters for the error process and MA(1) process are chosen randomly. In notation: $$y_t = z_t + \theta z_{t-1} + \varepsilon_t$$ where θ such that the above is stationary and $\varepsilon_t = b\varepsilon_{t-1} + u_t, u_t \sim N(0,1)$.
MA(1) w/ MA errors	Moving average process of order one. Error structure is an MA(1) process. Parameters for the error process and MA(1) process are chosen randomly. In notation: $$y_t = z_t + \theta z_{t-1} + \varepsilon_t$$ where θ such that the above is stationary and $\varepsilon_t = v_t + bv_{t-1} + u_t, v_t, u_t \sim N(0,1)$.

After the series is simulated and transformed, it is used to estimate a variety of linear autoregressive models and evaluate their predictive power. Specifically, autoregressive models of order one up to five are used for each data generating process to estimate the analogous autoregressive parameters and predict future values based on those parameters. In order to evaluate predictions, the simulated series sample is not used in its whole to estimate the AR polynomial parameters but is instead split into separate partitions for estimation (also known as training) and prediction-evaluation (also known as testing). The way the partitions are set for the testing of the cross validation methods that are described in Bergmeir et al. (2018) are similarly adapted here. Necessary for the comprehension of the experiment are the different validation methods used and are described below:

- **OOS (Out-of-sample validation)**
  The simple way of validating a model. The data are split once into training and test sets (not with random sampling, the test set is always the last block of the data). The model is fit on the training set and evaluated on the test set. Error measures like mean squared error (MSE), mean absolute error (MAE) are calculated from the actual test data and the fitted values of the model (predictions). Predictions for the test set observations are evaluated based on the true values of the dependent variable. Since the test observations are not used in fitting the model, its accuracy can be estimated more accurately.

- **k-fold cross-validation**
  Data is split in k random subsets. Each time, a single of the k subsets is used as a test set and the model is trained on the rest of the data. The process is repeated for every subset. Error measures of interest are averaged over the k validations for better accuracy. Train and test sets do not necessarily

are ordered. Splits are made with random sampling. Here, k is chosen equal to 5 similarly to Bergmeir et al. (2018) for consistency. Additionally, a k between 5 and 10 has been shown to be enough for accurate measures without suffering from calculation complexity (James et al. 2013).

- **LOOCV (Leave-one-out cross-validation)**
  A special case of k-fold CV with k = n, where n is the sample size. This means that only one observation at a time is used as the test set. The model is trained on all observations except one and evaluated on the one excluded. Error measures are again extracted and averaged over n times where every observation has been used for validation. LOOCV is more flexible than k-fold since it uses more data to train but it incurs computational cost and its performance improvement with respect to k-fold rarely justify it. Performance compared to 5-fold CV is very similar as showcased in Bergmeir et al. (2018) and is not included here.

- **k-fold CV for dependent data** (referred as "noDepCV" in the output tables)
  Functions similarly to the standard k-fold but in order to control for dependence in the data (like the one existent in time series) some observations before and after those used for evaluation are discarded. This way, advantages of CV are exploited even with dependent data but a considerable number of observations is discarded and the smaller sample size often leads to inaccurate predictions.

- **Walk-forward testing (a.k.a rolling scheme or sliding window)**
  In this case, training and test sets are taken sequentially in similar fashion to OOS validation but not all observations are used. This allows for the train/test data "window" to slide forward in time. Each time the window is moved forward by one observation and a training observation is discarded (from the start) and one is added at the end of the test set. This way, the model is trained and evaluated on updated data for every new iteration and at the same time the order of the data is maintained.

To implement the walk-forward testing procedure, the series is split differently. Like the experiment in Bergmeir et al. (2018), the series is separated into in-set and out-set. The in-set is used to estimate and evaluate predictions by extracting error measures with a traditional train/test split described in James et al. (2013). The in-set is then used in its entirety to retrain the model and the out-set is used as a validation set to the retrained model thus estimating the "generalization error" (Bergmeir et al. 2018). The particularity of the walk-forward testing method is that it involves repeated iterations and each time, both the in-set and out-set are pushed forward in time by one observation. The in-set and out-set consist, by design, of subsequent observations and so, contrary to the respective cross validation experiment, the whole in-set/out-set block is iteratively moved ahead and estimation and prediction are repeated. A small percentage of the observations is discarded in each iteration to allow for a sliding in-set/out-set window (ten percent). The loss of information in this procedure is minimal and a small percentage such as ten percent allows for multiple iterations even at moderate sample sizes. More iterations to average over mean that averaged error measures are more precise.

The available observations in each iterative step are divided into in-set/out-set based on a 70%/30% split while the train/test partition within the in-set is based on an 80%/20% split, see also Bergmeir et al. (2018). This split is repeated in each iteration of walk forward testing since the window of observations slides ahead but in the rest of the validation methods, the out-set is set only once and used only once once the cross-validation procedure is run solely on the in-set.

The performance of the partitioning method (referring both to cross-validation and walk-forward testing) on a given DGP is derived from the magnitude of the difference between the corresponding error metrics in the in-set and out-set. The metrics in question are prediction accuracy error (PAE) and absolute prediction accuracy error (APAE) which are calculated by using the difference in root mean square error (RMSE) and mean absolute error (MAE) in the in-set and out-set. In notation, these calculations are expressed as

$$PAE = (M_{in} - M_{out}) \tag{3.1}$$

and

$$APAE = |M_{in} - M_{out}| \tag{3.2}$$

which are averaged over the the number of simulation runs to get $MPAE$ and $MAPAE$ respectively, in notation as follows.

$$MPAE = \frac{1}{m}\sum(M_{in} - M_{out}) \tag{3.3}$$

and

$$MAPAE = \frac{1}{m}\sum |M_{in} - M_{out}|. \tag{3.4}$$

In the above, m represents the number of simulation runs and M represents the error measure and is indexed by *in* or *out* corresponding to the partition in which the error measure was calculated. RMSE and MAE are calculated as:

$$RMSE = \frac{1}{N_{set}} \sum_{i \in set} (y_i - \hat{y}_i)^2 \tag{3.5}$$

and

$$MAE = \frac{1}{N_{set}} \sum_{i \in set} |y_i - \hat{y}_i| \tag{3.6}$$

where $i \in set$ gives all the observations in the given validation set (whether the test set or the out-set) and N_{set} denotes the size of said sub-sample. Also, denoted by y_i and $\hat{y}_i$ respectively are the true value of the observation i in the data and the predicted value for observation i.

When generating an autoregressive series, either for use directly in the experiment or indirectly as error structure, stationarity is guaranteed since the random parameters are chosen such that the autoregressive polynomial has roots between 1.1 and 5. Similarly, for the moving average processes, polynomial roots between 1.1 and 1.2 ensure stationarity. Stationarity is a necessary condition here. Since resampling methods are employed, it is important that the series has the same stochastic distribution regardless of the sub-sample examined. As mentioned before, the parameters for the lagged values are different random parameters chosen by setting a different seed in each Monte Carlo run and specifically as a linear function of the simulation run. This way, parameters are different in each run but results of the whole process can be replicated.

The new data generating processes that are used to extend the simulation experiment are described here. Firstly, in order to introduce a series with structural break the stochastic series in Bergmeir et al. (2018) are adapted. Due to the stationarity condition necessary for estimation and prediction, series with structural breaks cannot be generated by shifting the intercept. Instead, the series is generated by using two different sets of random parameters for the series (the number of of breaks within the series is one resulting a series with different parameters in its two parts). Parameters are such that mean and variance are maintained throughout. Just as with the simple stochastic DGPs, in each run a different seed is chosen for the generation of parameters.

Another extension of the experiment is the introduction of autocorrelation in the error term. New DGPs are used where the error term was set manually to reflect that. For the DGPs with autocorrelated errors, the series generation is similar to the simple stochastic DGPs in Bergmeir et al. (2018) but the simulation error is not a white noise process. Significant autocorrelation is introduced by adding an error term that is in turn generated either as an AR(1) or MA(1) process parameters for which are chosen randomly in each run.

One deviation of the experiment carried out here from the one in Bergmeir et al. (2018) is in the sample size of the simulated series. The chosen sample size is 500 instead of 200. While a small sample size is useful to grasp real application performance with scarce data, evaluating performance on more substantial samples is also insightful for a more generalized picture needed here for the substantiation of the use of backtesting. To aid in comprehension, the procedure followed here is also given in pseudo-code (see Algorithms 1 and 2 in the appendix).

3.2 Simulation results

The simulation experiment results are presented in Tables 3.2 and 3.3 for the cross-validation methods of Bergmeir et al. (2018) and walk-forward testing respectively. Given are the prediction accuracy errors figures for the RMSE and MAE metric calculated as described in the previous sections. For a comparison to a much more traditional evaluation technique, the average Akaike information criterion (AIC) is also showcased (averaged over the number of simulation runs). The average AIC given in Table 3.3 would be calculated the same way for the models in Table 3.2 and the figures are not repeated. It is noted that adding variables to a model improves the fit but might result in overfitting which negatively affects model predictions. This criterion (Akaike 1974) penalizes larger models resulting in a measure to evaluate models depending on their fit/parsimonity trade-off. Comparison to AIC is imperative since it is one of the tools most used for model selection. For the ”best” model a minimized AIC is sought. It is important to note that when interpreting the prediction accuracy error figures, a metric for a given model and method with the

lowest absolute value means better performance. Comparing the corresponding figures in the two tables, the first key takeaway is that walk-forward testing is performing very similarly to the CV methods for most data generating processes and even outperforming them for some.

Table 3.2: Prediction accuracy errors using the Bergmeir et al. (2018) cross-validation methods on all DGPs, n = 500. Number of simulations = 1000.

	# Lags	RMSE		MAE	
		MAPAE	MPAE	MAPAE	MPAE
Stochastic AR(3)					
5-fold CV	AR(1)	0.0611	0.0056	0.0529	0.0032
	AR(2)	0.0538	0.0044	0.0468	0.0021
	AR(3)	0.0538	0.0053	0.0468	0.0027
	AR(4)	0.0539	0.0053	0.0470	0.0028
	AR(5)	0.0543	0.0061	0.0474	0.0035
noDepCV	AR(1)	0.0779	0.0525	0.0647	0.0401
	AR(2)	0.0826	0.0665	0.0667	0.0509
	AR(3)	0.0973	0.0862	0.0776	0.0663
	AR(4)	0.1157	0.1087	0.0911	0.0837
	AR(5)	0.1368	0.1329	0.1068	0.1025
OOS	AR(1)	0.0906	0.0041	0.0780	0.0054
	AR(2)	0.0807	0.0018	0.0687	0.0030
	AR(3)	0.0812	0.0023	0.0693	0.0033
	AR(4)	0.0814	0.0016	0.0695	0.0029
	AR(5)	0.0819	0.0023	0.0696	0.0036
Stochastic MA(1)					
5-fold CV	AR(1)	0.0702	0.0071	0.0606	0.0046
	AR(2)	0.0645	0.0042	0.0550	0.0029
	AR(3)	0.0611	0.0061	0.0526	0.0039
	AR(4)	0.0595	0.0060	0.0511	0.0039
	AR(5)	0.0582	0.0071	0.0501	0.0049
noDepCV	AR(1)	0.0771	0.0307	0.0653	0.0231
	AR(2)	0.0859	0.0589	0.0705	0.0455
	AR(3)	0.0927	0.0745	0.0754	0.0570
	AR(4)	0.1111	0.1006	0.0890	0.0777
	AR(5)	0.1277	0.1213	0.1008	0.0938
OOS	AR(1)	0.1027	-0.0007	0.0885	0.0023
	AR(2)	0.0933	-0.0013	0.0797	0.0020
	AR(3)	0.0886	-0.0013	0.0766	0.0014
	AR(4)	0.0876	0.0004	0.0759	0.0029
	AR(5)	0.0861	0.0009	0.0737	0.0036
Seasonal ARIMA					
5-fold CV	AR(1)	115.1834	-21.9630	96.7376	-21.1397
	AR(2)	117.0713	-32.1678	98.4162	-29.5819
	AR(3)	119.3455	-39.5577	100.1148	-36.0202
	AR(4)	124.2539	-51.6009	104.3097	-45.6987
	AR(5)	128.6290	-61.7764	107.7119	-54.1881
noDepCV	AR(1)	119.0875	10.7730	99.1899	4.6100
	AR(2)	121.6447	17.9357	100.7751	9.4933
	AR(3)	125.5325	27.0583	103.4676	15.8845
	AR(4)	129.9525	33.3091	106.3693	20.1499
	AR(5)	134.8378	39.6226	110.0953	23.8675
OOS	AR(1)	143.4039	-2.9361	119.4262	1.1792
	AR(2)	144.5708	-4.5383	120.7964	-0.3288
	AR(3)	147.0885	-3.9486	122.7743	0.0309
	AR(4)	150.5957	-4.0077	126.4228	1.0284
	AR(5)	152.6405	-5.4721	128.5837	0.0508
AR(3) w/ break					

5-fold CV	AR(1)	0.1698	0.1698	0.0934	0.0934
	AR(2)	0.2852	0.2852	0.2205	0.2205
	AR(3)	0.1014	0.1014	0.0788	0.0788
	AR(4)	0.0623	0.0623	0.0548	0.0548
	AR(5)	0.0605	0.0605	0.0517	0.0517
noDepCV	AR(1)	0.3556	0.3556	0.2070	0.2070
	AR(2)	0.4539	0.4539	0.3098	0.3098
	AR(3)	0.2713	0.2713	0.1985	0.1985
	AR(4)	0.2104	0.2104	0.1657	0.1657
	AR(5)	0.3110	0.3110	0.2519	0.2519
OOS	AR(1)	0.9686	0.9686	0.7228	0.7228
	AR(2)	1.1121	1.1121	0.8415	0.8415
	AR(3)	0.6537	0.6537	0.5474	0.5474
	AR(4)	0.6192	0.6192	0.5443	0.5443
	AR(5)	0.7209	0.7209	0.6172	0.6172
MA(1) w/ break					
5-fold CV	AR(1)	0.0836	0.0836	0.0761	0.0761
	AR(2)	0.0925	0.0925	0.0820	0.0820
	AR(3)	0.0829	0.0829	0.0785	0.0785
	AR(4)	0.0613	0.0613	0.0553	0.0553
	AR(5)	0.0553	0.0553	0.0511	0.0511
noDepCV	AR(1)	0.2086	0.2086	0.1734	0.1734
	AR(2)	0.2926	0.2926	0.2297	0.2297
	AR(3)	0.2987	0.2987	0.2332	0.2332
	AR(4)	0.2610	0.2610	0.2066	0.2066
	AR(5)	0.2931	0.2931	0.2408	0.2408
OOS	AR(1)	0.4436	0.4436	0.3926	0.3926
	AR(2)	0.6457	0.6457	0.5638	0.5638
	AR(3)	0.7156	0.7156	0.6164	0.6164
	AR(4)	0.6334	0.6334	0.5521	0.5521
	AR(5)	0.6177	0.6177	0.5409	0.5409
AR(3) w/ AR errors					
5-fold CV	AR(1)	0.0255	-0.0173	0.0321	-0.0179
	AR(2)	0.0338	-0.0336	0.0249	-0.0249
	AR(3)	0.0374	-0.0374	0.0244	-0.0244
	AR(4)	0.0335	-0.0335	0.0217	-0.0217
	AR(5)	0.0314	-0.0314	0.0171	-0.0171
noDepCV	AR(1)	0.0182	0.0047	0.0249	-0.0056
	AR(2)	0.0119	0.0073	0.0077	-0.0014
	AR(3)	0.0342	0.0342	0.0217	0.0214
	AR(4)	0.0587	0.0587	0.0370	0.0370
	AR(5)	0.0726	0.0726	0.0505	0.0505
OOS	AR(1)	0.0211	0.0192	0.0338	0.0129
	AR(2)	0.0181	0.0181	0.0236	0.0227
	AR(3)	0.0165	0.0165	0.0269	0.0269
	AR(4)	0.0178	0.0178	0.0269	0.0269
	AR(5)	0.0166	0.0166	0.0294	0.0294
AR(3) w/ MA errors					
5-fold CV	AR(1)	0.0549	-0.0478	0.0433	-0.0216
	AR(2)	0.0610	-0.0610	0.0443	-0.0443
	AR(3)	0.0542	-0.0542	0.0282	-0.0282
	AR(4)	0.0443	-0.0443	0.0143	-0.0143
	AR(5)	0.0357	-0.0357	0.0085	-0.0080
noDepCV	AR(1)	0.0431	-0.0282	0.0343	-0.0072
	AR(2)	0.0182	-0.0127	0.0171	-0.0071
	AR(3)	0.0142	0.0106	0.0221	0.0190
	AR(4)	0.0289	0.0216	0.0308	0.0308

	AR(5)	0.0465	0.0465	0.0509	0.0509
OOS	AR(1)	0.0227	0.0214	0.0595	0.0589
	AR(2)	0.0121	0.0061	0.0356	0.0356
	AR(3)	0.0129	0.0012	0.0420	0.0420
	AR(4)	0.0190	0.0095	0.0536	0.0536
	AR(5)	0.0193	0.0168	0.0589	0.0589
MA(1) w/ AR errors					
5-fold CV	AR(1)	0.0638	-0.0638	0.0356	-0.0356
	AR(2)	0.0549	-0.0549	0.0371	-0.0371
	AR(3)	0.0613	-0.0613	0.0283	-0.0283
	AR(4)	0.0461	-0.0461	0.0190	-0.0190
	AR(5)	0.0360	-0.0360	0.0118	-0.0118
noDepCV	AR(1)	0.0480	-0.0480	0.0200	-0.0200
	AR(2)	0.0110	-0.0110	0.0201	-0.0009
	AR(3)	0.0144	0.0114	0.0309	0.0309
	AR(4)	0.0271	0.0180	0.0317	0.0317
	AR(5)	0.0323	0.0323	0.0388	0.0388
OOS	AR(1)	0.0295	0.0295	0.0503	0.0503
	AR(2)	0.0161	0.0161	0.0421	0.0421
	AR(3)	0.0106	-0.0079	0.0393	0.0393
	AR(4)	0.0058	0.0022	0.0407	0.0407
	AR(5)	0.0128	0.0128	0.0516	0.0516
MA(1) w/ MA errors					
5-fold CV	AR(1)	0.0959	-0.0959	0.0709	-0.0709
	AR(2)	0.0791	-0.0791	0.0444	-0.0444
	AR(3)	0.0857	-0.0857	0.0480	-0.0480
	AR(4)	0.0686	-0.0686	0.0448	-0.0448
	AR(5)	0.0613	-0.0613	0.0432	-0.0432
noDepCV	AR(1)	0.0903	-0.0903	0.0696	-0.0696
	AR(2)	0.0215	-0.0215	0.0275	-0.0023
	AR(3)	0.0228	-0.0228	0.0196	0.0080
	AR(4)	0.0168	-0.0058	0.0284	0.0130
	AR(5)	0.0286	0.0286	0.0352	0.0275
OOS	AR(1)	0.0473	0.0473	0.0446	0.0446
	AR(2)	0.0261	0.0035	0.0408	0.0408
	AR(3)	0.0233	-0.0233	0.0275	0.0275
	AR(4)	0.0252	-0.0092	0.0356	0.0338
	AR(5)	0.0231	-0.0082	0.0355	0.0345

Table 3.3: Prediction accuracy errors using the Walk-forward testing procedure, n = 500. Akaike information criterion is also listed (last column). Number of simulations = 1000.

# Lags	RMSE		MAE		AIC
	MAPAE	MPAE	MAPAE	MPAE	
Stochastic AR(3)					
AR(1)	0.0775	0.0028	0.0677	0.0033	1449.44
AR(2)	0.0703	0.0001	0.0602	0.0007	1410.86
AR(3)	0.0710	0.0007	0.0609	0.0010	1408.53
AR(4)	0.0713	0.0006	0.0611	0.0009	1409.43
AR(5)	0.0718	0.0012	0.0613	0.0016	1410.38
Stochastic MA(1)					
AR(1)	0.0917	-0.0018	0.0777	0.0008	1546.82
AR(2)	0.0838	-0.0023	0.0711	0.0003	1491.75
AR(3)	0.0798	-0.0022	0.0678	0.0001	1464.77
AR(4)	0.0780	-0.0007	0.0664	0.0012	1448.34
AR(5)	0.0763	0.0003	0.0653	0.0021	1438.46

Seasonal ARIMA					
AR(1)	134.5255	-2.7368	112.9500	1.1871	8245.58
AR(2)	135.3352	-5.3037	113.7143	-1.2689	8242.23
AR(3)	137.3391	-5.2872	115.2063	-1.6940	8239.26
AR(4)	139.6598	-5.6044	117.6478	-1.2567	8235.45
AR(5)	140.7849	-7.1528	118.7747	-2.2577	8231.48
AR(3) w/ break					
AR(1)	0.6538	0.6538	0.4686	0.4686	1764.30
AR(2)	1.0629	1.0629	0.7345	0.7345	1664.75
AR(3)	0.8311	0.8311	0.5806	0.5806	1640.72
AR(4)	0.7721	0.7721	0.5592	0.5592	1642.62
AR(5)	0.7655	0.7655	0.5621	0.5621	1644.50
MA(1) w/ break					
AR(1)	0.2689	0.2689	0.2231	0.2231	1674.00
AR(2)	0.4977	0.4977	0.3993	0.3993	1675.99
AR(3)	0.6500	0.6500	0.4959	0.4959	1677.98
AR(4)	0.6576	0.6576	0.4923	0.4923	1675.59
AR(5)	0.6666	0.6666	0.4956	0.4956	1677.40
AR(3) w/ AR errors					
AR(1)	0.0641	-0.0641	0.0422	-0.0414	1505.71
AR(2)	0.0492	-0.0492	0.0126	-0.0112	1442.16
AR(3)	0.0472	-0.0472	0.0033	-0.0015	1436.00
AR(4)	0.0445	-0.0445	0.0015	0.0002	1436.70
AR(5)	0.0431	-0.0431	0.0031	0.0025	1437.69
AR(3) w/ MA errors					
AR(1)	0.0713	-0.0713	0.0205	-0.0101	1594.88
AR(2)	0.0630	-0.0630	0.0147	-0.0114	1512.77
AR(3)	0.0690	-0.0690	0.0134	-0.0014	1490.48
AR(4)	0.0592	-0.0592	0.0161	0.0143	1469.15
AR(5)	0.0543	-0.0543	0.0204	0.0198	1465.15
MA(1) w/ AR errors					
AR(1)	0.0455	-0.0455	0.0328	0.0082	1577.34
AR(2)	0.0502	-0.0502	0.0331	0.0045	1518.84
AR(3)	0.0709	-0.0709	0.0166	0.0048	1497.54
AR(4)	0.0625	-0.0625	0.0066	0.0066	1477.82
AR(5)	0.0581	-0.0581	0.0174	0.0174	1472.72
MA(1) w/ MA errors					
AR(1)	0.0481	-0.0481	0.0272	-0.0272	1792.43
AR(2)	0.0769	-0.0769	0.0162	-0.0162	1653.48
AR(3)	0.0934	-0.0934	0.0263	-0.0230	1618.91
AR(4)	0.0831	-0.0831	0.0476	-0.0136	1561.59
AR(5)	0.0881	-0.0881	0.0497	-0.0174	1548.79

For the simpler stochastic DGPs (AR(3), MA(1), seasonal ARIMA), 5-fold cross-validation performs better than every other method but walk-forward testing comes exceptionally close and outperforms the rest of the cross-validation methods. Better performance is seen in the overall lower figures than those for the other CV methods and isolated exceptions do not affect how the performance of the method should be perceived. For every model from the range of the five autoregressive models chosen 5-fold CV achieves metrics approximately 0.01 lower than the ones given by walk-forward testing the data generating processes.

Interestingly, for the DGPs with a structural break, 5-fold CV again outperforms walk-forward testing. This implies that in this case, the data does not necessarily need to be maintained in chronological order to achieve better predictions, at least in terms of estimating the generalization error described above. Prediction accuracy error metrics for five-fold CV are closer to zero across for both RMSE and MAE compared to the walk-forward testing method. The difference here is much more substantial for both the AR and MA data generating processes.

What can also be in the tables (3.2 and 3.3) is that for the moving average processes with autocorrelated error structures the walk-forward testing procedure produces figures with lower absolute values for all

estimated models than five-fold cross-validation. OOS is the strongest contender here as it yields the lowest absolute value metrics. Regardless, walk-forward testing remains very competitive for all processes and even outperforms the non-dependent cross-validation used in Bergmeir et al. (2018).

In terms of prediction accuracy errors, for the moving average processes with autocorrelated errors the best performing method is simple OOS testing. For the rest, 5-fold cross-validation is the most dominant method as it achieves prediction accuracy error measures closer to zero than the competing cross-validation methods. Bergmeir et al. (2018) also implement leave-one-out-cross-validation (LOOCV) but since results are very similar to 5-fold CV across the board it is deemed redundant for this exercise. Examining performance in terms of the raw measures themselves also presents interest. Achieving the "generalization error" is not always wanted. Walk-forward testing remains competitive in terms of raw RMSE and MAE as well. In the case of the new DGPs introduced, walk-forward testing is slightly behind in performance but for the simple AR(3) and MA(1) processes it achieves lower RMSE and MAE.

As for model selection, one could only draw conclusions for the case of the AR(3) processes where the true data generating process is included in the attempted models. Based on which of the five models attempted achieves the lowest error measures, the best model is determined. AIC also becomes relevant in this comparison as the model with the lowest AIC would be the one selected for use. One would expect that the procedures that involve multiple iterations, like CV and walk-forward testing would achieve better results when the attempted model is the one used to generate the data and outperform the AIC selection. In this case, contrary to both methods examined here, AIC selects the true model, although the other procedures only miss the true model by one autoregressive order (see first five rows in Tables 3.2, 3.3). This is not the case for the raw RMSE and MAE figures with which both methods select the correct model (first five rows of Tables A.1, A.2). These results are expected since the model is relatively simple and so every method is successful in identifying the true model. Any shortcomings of CV and walk-forward testing in terms of PAE are to be taken lightly since the range of the models examined here is narrow and the prediction accuracy error measure design is not the primary tool for model selection. In the rest of the AR(3)-based DGPs, AIC remains successful excluding the case of the moving-average errors where walk-forward testing also manages to pinpoint the true number of lags present thanks to the lower MPAE, MAPAE figures for MAE. Five-fold CV does not identify the AR(3) model as the best model for the modified AR(3) DGPs.

3.3 Applications on real data

Since simulated data do not tell the whole story, the procedures are also applied to real data. In order to examine the methods on a broader scale, four different data series are selected for this purpose. They are of economic interest and are characterized by different granularity and sample size to test the simulated outcomes. First, real interest rate for the USA provided by The World Bank DataBank[1] (Figure 3.1), second, the harmonised index of consumer prices (HICP) for inflation rate in the Euro area provided by the European Central Bank Statistical Data Warehouse[2] (Figure 3.2). Additionally, the euro area M3 monetary aggregate (European Central Bank Statistical Data Warehouse[3]). Lastly, US dollar to euro exchange rate data[4] are also included in the analysis.

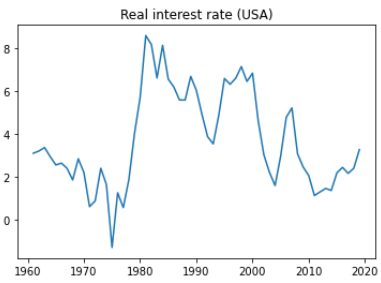
Before testing, the series are tested for stationarity. All series indicate presence of unit roots (ADF tests are conducted) and so they are differenced at the start of the procedure to end up with suitable stationary series. Prediction accuracy error measures for 5-fold CV and walk-forward testing are given in Tables 3.4 and 3.5.

[1] https://data.worldbank.org/indicator/FR.INR.RINR?locations=US

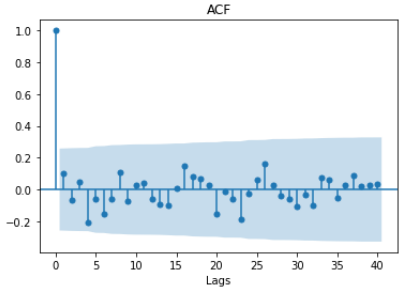
[2] https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=122.ICP.M.U2.N.000000.4.ANR

[3] https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=117.BSI.M.U2.Y.V.M30.X.I.U2.2300.Z01.A

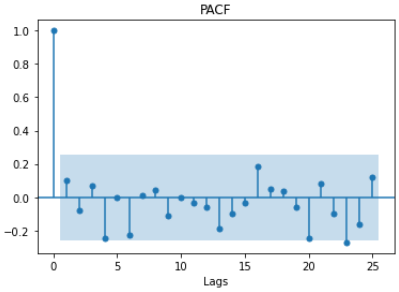
[4] https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=120.EXR.D.USD.EUR.SP00.A



(a)

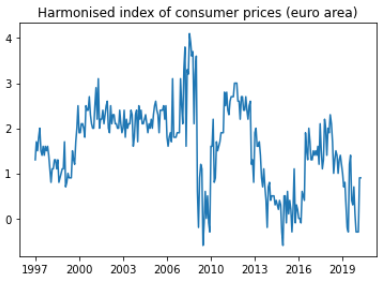


(b)

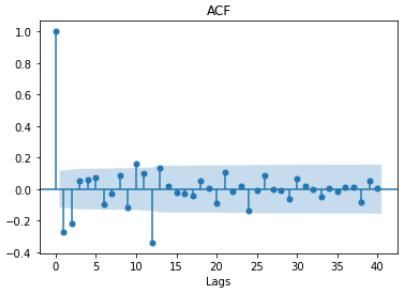


(c)

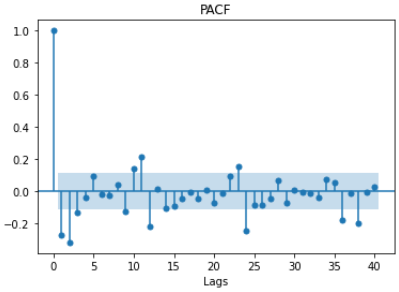
Figure 3.1: (a): Real interest rate (%) of the United States, annual frequency ranging from 1961 to 2020 (Sample size: 59). (b): Autocorrelation function of the final, differenced series, given for 40 lags. Shaded regions represent the 95% confidence interval. (c): Partial autocorrelation function of the final, differenced series, given for 25 lags. Shaded regions represent the 95% confidence interval.



(a)



(b)



(c)

Figure 3.2: (a): Harmonised index of consumer prices, monthly frequency ranging from April 1997 to February 2021 (Sample size: 290). (b): Autocorrelation function of the final, differenced series, given for 40 lags. Shaded regions represent the 95% confidence interval. (c): Partial autocorrelation function of the final, differenced series, given for 25 lags. Shaded regions represent the 95% confidence interval.

From the different cross-validation procedures, only five-fold cross-validation is performed since it generally outperforms the rest on the simulated data for prediction accuracy errors. For the sake of comparison, walk-forward testing is also applied here. Prediction accuracy error figures are listed in Tables 3.4 and 3.5.

Table 3.4: 5-fold cross-validation applied on real data series. Number of iterations = 1000.

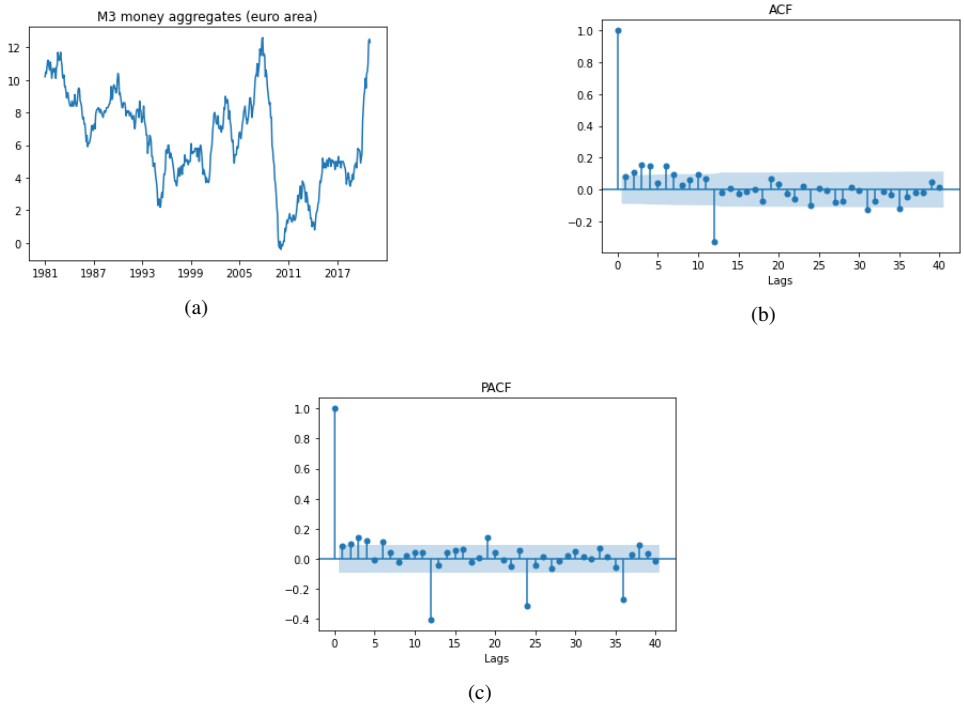
# Lags	RMSE		MAE	
	MAPAE	MPAE	MAPAE	MPAE
Real interest rate (US)				
AR(1)	0.5803	0.5803	0.5201	0.5201
AR(2)	0.6773	0.6773	0.5705	0.5705
AR(3)	0.7221	0.7221	0.6142	0.6142
AR(4)	0.8713	0.8713	0.7412	0.7412
AR(5)	1.1707	1.1707	0.9143	0.9143
HICP				
AR(1)	0.0550	0.0550	0.0109	-0.0109
AR(2)	0.0685	0.0685	0.0002	-0.0002
AR(3)	0.0733	0.0733	0.0029	-0.0029
AR(4)	0.0763	0.0763	0.0052	-0.0052
AR(5)	0.0734	0.0734	0.0031	0.0031
Monetary aggregate M3 (euro area)				
AR(1)	0.1030	0.1030	0.0826	0.0826
AR(2)	0.0986	0.0986	0.0763	0.0763
AR(3)	0.0850	0.0850	0.0634	0.0634
AR(4)	0.0888	0.0888	0.0650	0.0650
AR(5)	0.0885	0.0885	0.0637	0.06371
US/euro exchange rate				
AR(1)	0.000568	0.000568	0.000037	0.000037
AR(2)	0.000569	0.000569	0.000037	0.000037
AR(3)	0.000572	0.000572	0.000039	0.000039
AR(4)	0.000576	0.000576	0.000041	0.000041
AR(5)	0.000577	0.000577	0.000043	0.000043

Table 3.5: Walk-forward testing applied on real data series.

# Lags	RMSE		MAE	
	MAPAE	MPAE	MAPAE	MPAE
Real interest rate (US)				
AR(1)	0.1135	0.1135	0.1163	0.1163
AR(2)	0.1296	0.1296	0.1236	0.1236
AR(3)	0.1006	0.1006	0.0563	0.0563
AR(4)	0.1375	0.1375	0.1349	0.1349
AR(5)	0.1331	0.1331	0.1246	0.1246
HICP				
AR(1)	0.0209	-0.0209	0.0295	-0.0295
AR(2)	0.0185	0.0185	0.0025	-0.0025
AR(3)	0.0126	0.0126	0.0045	-0.0045
AR(4)	0.0131	0.0131	0.0051	-0.0051
AR(5)	0.0095	0.0095	0.0021	-0.0021
Monetary aggregate M3 (euro area)				
AR(1)	0.0659	0.0659	0.0615	0.0615

AR(2)	0.0666	0.0666	0.0628	0.0628
AR(3)	0.0414	0.0414	0.0365	0.0365
AR(4)	0.0346	0.0346	0.0275	0.0275
AR(5)	0.0311	0.0311	0.0232	0.0232
US/euro exchange rate				
AR(1)	0.0015685	0.0015685	0.0006320	0.0006320
AR(2)	0.0015700	0.0015700	0.0006325	0.0006325
AR(3)	0.0015695	0.0015695	0.0006303	0.0006303
AR(4)	0.0015726	0.0015726	0.0006348	0.0006348
AR(5)	0.0015691	0.0015691	0.0006324	0.0006324

Walk-forward testing is performing better for the real interest rate series with lower PAE measures across the board. Walk-forward testing manages to beat 5-fold CV for RMSE PAE measures as well and achieves very similar MAE PAE figures. In the smaller samples examined here (59 for the interest rate data, 290 for the HICP data), walk-forward testing clearly outperforms 5-fold cross-validation. Similarly, CV is also outperformed by walk-forward testing in the case of the M3 aggregate data especially for the RMSE figures. In a much larger sample, as is for the exchange rate data, results are different. It is observed that CV achieved much lower prediction accuracy error metrics for this data series and walk-forward testing, while its figures are quite small, does not manage to stay competitive against 5-fold CV.

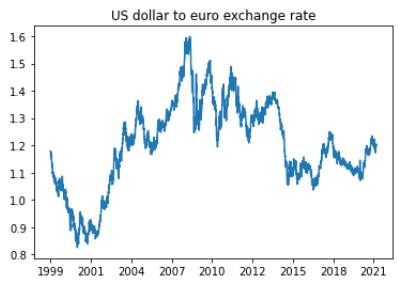


(a) (b) (c)

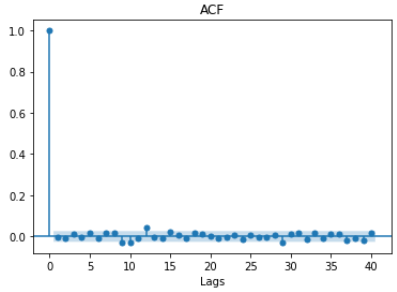
Figure 3.3: (a): Monetary aggregate M3 vis-a-vis euro area, monthly frequency ranging from January 1981 to February 2021 (Sample size: 482). (b): Autocorrelation function of the final, differenced series, given for 40 lags. Shaded regions represent the 95% confidence interval. (c): Partial autocorrelation function of the final, differenced series, given for 25 lags. Shaded regions represent the 95% confidence interval.

The picture painted by the raw RMSE, MAE in Tables 3.6 and 3.7 is similar. For the first three data series (real interest rate, HICP, M3 aggregate), walk-forward testing achieves lower figures than 5-fold CV. Again, for the exchange rate data, 5-fold CV performs slighty better that walk-forward testing. It is worth noting that with a large sample size like this, the differences observed are much slimmer. Even though the prediction accuracy error design with the MPAE, MAPAE measures makes more sense in a synthetic data context , it is not the intuitive metric when using real data. Since the designation of the out-set keeps a significant amount of observations unused in estimation, something one would typically avoid, using the standard RMSE and MAE on the whole sample is more representative for assessing the suitability of a given

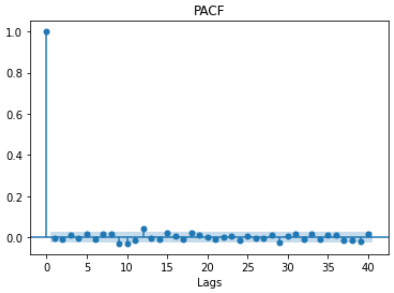
method. As seen in the tables, walk-forward testing constitutes the superior method at small samples. Only in the case of exchange rates, where the sample size is just over 5700 observations 5-fold CV marginally outperforms walk-forward testing.



(a)



(b)



(c)

Figure 3.4: (a): ECB reference exchange rate, US dollar/Euro, daily frequency ranging from 4th January 1981 to 24th April 2021 (Sample size: 5708). (b): Autocorrelation function of the final, differenced series, given for 40 lags. Shaded regions represent the 95% confidence interval. (c): Partial autocorrelation function of the final, differenced series, given for 25 lags. Shaded regions represent the 95% confidence interval.

From the range of five models used here, the model with the lowest metrics would be naturally selected as the better among the range. For real interest rate, 5-fold CV chooses the AR(1) while walk-forward testing selects the AR(5) with both RMSE and MAE. For the HICP data, 5-fold CV chooses the AR(2) with RMSE and the AR(4) with MAE. Walk-forward testing chooses the AR(3) with RMSE and the AR(1) with MAE. For the M3 data, 5-fold RMSE selects the AR(4), MAE the AR(3) while walk-forward testing selects the AR(5) with both measures. For the exchange rate data, the smallest model is chosen across the board. AIC is also reported in Table 3.7. The AR(1) is chosen for interest rate, AR(5) for HICP, AR(4) for M3 and AR(1) for the exchange rate data. Having that said, not knowing the true model, as is with real data, AIC and metric selections cannot be correctly evaluated here.

Table 3.6: RMSE, MAE, 5-fold cross-validation for real data series.

# Lags	RMSE	MAE
Real interest rate (US)		
AR(1)	1.5569	1.2762
AR(2)	1.6483	1.3389
AR(3)	1.7138	1.3764
AR(4)	1.7953	1.4437
AR(5)	2.1007	1.6230
HICP		
AR(1)	0.5481	0.3593
AR(2)	0.5270	0.3495
AR(3)	0.5300	0.3462

AR(4)	0.5339	0.3449
AR(5)	0.5333	0.3487
Monetary aggregate M3 (euro area)		
AR(1)	0.4657	0.3598
AR(2)	0.4670	0.3621
AR(3)	0.4608	0.3527
AR(4)	0.4601	0.3535
AR(5)	0.4634	0.3571
US/euro exchange rate		
AR(1)	0.0074671	0.0053498
AR(2)	0.0074686	0.0053498
AR(3)	0.0074717	0.0053522
AR(4)	0.0074762	0.0053558
AR(5)	0.0074806	0.0053595

Table 3.7: RMSE, MAE, walk-forward testing on real data.

# Lags	RMSE	MAE	AIC
Real interest rate (US)			
AR(1)	1.1589	0.9159	177.3608
AR(2)	1.1764	0.9354	179.0453
AR(3)	1.1865	0.8845	180.8357
AR(4)	1.1419	0.8545	179.9961
AR(5)	1.1425	0.8555	181.9958
HICP			
AR(1)	0.4516	0.3219	444.6139
AR(2)	0.4561	0.3262	415.5617
AR(3)	0.4497	0.3230	411.9500
AR(4)	0.4514	0.3220	413.4100
AR(5)	0.4502	0.3242	412.8034
Monetary aggregate M3 (euro area)			
AR(1)	0.4381	0.3412	555.3772
AR(2)	0.4409	0.3463	552.4968
AR(3)	0.4281	0.3273	545.7998
AR(4)	0.4193	0.3201	540.2109
AR(5)	0.4184	0.3192	542.2093
US/euro exchange rate			
AR(1)	0.0084388	0.0059377	-39925.090
AR(2)	0.0084400	0.0059379	-39923.714
AR(3)	0.0084403	0.0059357	-39922.241
AR(4)	0.0084440	0.0059415	-39920.281
AR(5)	0.0084421	0.0059412	-39920.027

Chapter 4

Conclusions

In this section, the main results are discussed and conclusions are drawn from these results. This study examined one way backtesting could be implemented in the time series setting and specifically to evaluate model predictions and subsequent model selection based on the quality of said predictions. A Monte Carlo simulation experiment was conducted adapting the experiment of Bergmeir et al. (2018) to accommodate structural breaks and autocorrelated errors in the data generating processes to evaluate the performance of information criteria, cross-validation methods and backtesting implemented using the walk-forward testing method in Kaastra & Boyd (1996). Real data are also tested with these methods and analogous comparisons are made.

The main result of this exercise is that, despite the very limited use in similar time series and financial applications, backtesting implemented with a sliding "window" of data is very competitive in achieving accurate predictions even against the best performing cross-validation methods utilised in this experiment. Both in terms of prediction accuracy error measures and "raw" RMSE and MAE figures, which are more traditional in such types of model selection, the form of backtesting applied here is only slightly falling behind for some of the artificial data generating process and manages to outperform 5-fold cross-validation for processes with significant autocorrelation present in the error structure.

The encouraging performance of the method on processes with significant autocorrelation along with the fact that the models used to estimate and predict are very simple AR(p) models hint that the method could handle heavy misspecification which is very well the case with real data whose true generating process is rarely, if at all, known. Indeed, when it comes to the real data series used here, backtesting outperforms 5-fold CV especially at relatively small samples. Both PAE and traditional error measures confirm that backtesting implemented with a walk-forward testing (a.k.a sliding window) could provide prediction and model selection advantages over cross-validation in heavily misspecified time series contexts.

Having that said, this study is limited by the fact that theoretical properties of the walk forward testing method in large samples are not straightforward as with some types of cross-validation. Asymptotic properties of the procedure and error estimates would greatly benefit towards better comprehension of predictions and model selections given by the method. This thesis recommends that further research is required to establish in which scenarios such an undertaking would be beneficial for the researcher. Theoretical properties of such a method have to be thoroughly examined and the use of models and data generating processes with higher complexity need to be explored.

References

Akaike, H. (1974), 'A new look at the statistical model identification', *IEEE transactions on automatic control* **19**(6), 716–723.

Arlot, S. & Celisse, A. (2009), 'A survey of cross-validation procedures for model selection'.

Bergmeir, C. & Benítez, J. M. (2012), 'On the use of cross-validation for time series predictor evaluation', *Information Sciences* **191**, 192–213.

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

Campbell, S. D. (2005), 'A review of backtesting and backtesting procedures'.

Chow, G. C. (1960), 'Tests of equality between sets of coefficients in two linear regressions', *Econometrica: Journal of the Econometric Society* pp. 591–605.

Gudelek, M. U., Boluk, S. A. & Ozbayoglu, A. M. (2017), A deep learning based stock trading model with 2-d cnn trend detection, *in* '2017 IEEE Symposium Series on Computational Intelligence (SSCI)', IEEE, pp. 1–8.

Harvey, C. R. & Liu, Y. (2015), 'Backtesting', *The Journal of Portfolio Management* **42**(1), 13–28.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer.

Kaastra, I. & Boyd, M. (1996), 'Designing a neural network for forecasting financial', *Neurocomputing* **10**, 215–236.

Karathanasopoulos, A., Dunis, C. & Khalil, S. (2016), 'Modelling, forecasting and trading with a new sliding window approach: the crack spread example', *Quantitative Finance* **16**(12), 1875–1886.

Lin, C.-F. J. & Teräsvirta, T. (1994), 'Testing the constancy of regression parameters against continuous structural change', *Journal of econometrics* **62**(2), 211–228.

McQuarrie, A. & Tsai, C. (1998), 'Regression and time series model selection–world scientific', *New Jersey* .

Racine, J. (2000), 'Consistent cross-validatory model-selection for dependent data: hv-block cross-validation', *Journal of econometrics* **99**(1), 39–61.

Appendix

Algorithm 1: Cross-validation testing procedure on simulated data for calculation of prediction accuracy errors.

```
for (n in 1, 2, ..., number of simulations) do
    for (all data generating processes) do
        set seed (differs with n);
        generate synthetic series according to Table 3.1;
        for (all CV methods) do
            determine in-set/out-set partitions and number of iterations of each method;
            for (every iteration of the CV method in current run) do
                for (AR order from 1 to 5) do
                    estimate model of current order with train data;
                    evaluate model on test data;
                end
            end
            for (AR order from 1 to 5) do
                estimate model of current order with the whole in-set data;
                evaluate model on out-set data;
                determine prediction accuracy error measures by subtracting average error
                  measures calculated on the in-set and measures calculated on the out-set;
            end
        end
    end
end
```

Algorithm 2: Walk-forward testing procedure on simulated data for calculation of prediction accuracy errors.

```
for (n in 1, 2, ..., number of simulations) do
    for (all data generating processes) do
        set seed (differs with n);
        generate synthetic series according to Table 3.1;
        for (every iteration of the WFT method) do
            determine the in-set and out-set;
            for (AR order from 1 to 5) do
                estimate model of current order with train data;
                evaluate model on test data;
                estimate model of current order again, now with the all of the in-set data;
                evaluate model on out-set data;
            end
        end
        for (AR order from 1 to 5) do
            determine prediction accuracy error measures by subtracting average error measures
              calculated on the in-set and measures calculated on the out-set;
        end
    end
end
```

Table A.1: RMSE, MAE measures using the Bergmeir et al. (2018) cross-validation methods on all DGPs. Number of simulations = 1000, sample size n = 500.

	# Lags	RMSE	MAE
Stochastic AR(3)			
5-fold CV	AR(1)	1.0504	0.8388
	AR(2)	1.0079	0.8045
	AR(3)	1.0064	0.8035
	AR(4)	1.0080	0.8050
	AR(5)	1.0099	0.8065
noDepCV	AR(1)	1.0973	0.8757
	AR(2)	1.0700	0.8533
	AR(3)	1.0874	0.8671
	AR(4)	1.1114	0.8858
	AR(5)	1.1367	0.9056
OOS	AR(1)	1.0489	0.8410
	AR(2)	1.0052	0.8055
	AR(3)	1.0034	0.8041
	AR(4)	1.0043	0.8051
	AR(5)	1.0061	0.8066
Stochastic MA(1)			
5-fold CV	AR(1)	1.1554	0.9226
	AR(2)	1.0932	0.8734
	AR(3)	1.0650	0.8510
	AR(4)	1.0484	0.8375
	AR(5)	1.0390	0.8301
noDepCV	AR(1)	1.1789	0.9411
	AR(2)	1.1478	0.9160
	AR(3)	1.1334	0.9041
	AR(4)	1.1430	0.9113
	AR(5)	1.1531	0.9190
OOS	AR(1)	1.1476	0.9204
	AR(2)	1.0877	0.8725
	AR(3)	1.0576	0.8485
	AR(4)	1.0428	0.8365
	AR(5)	1.0328	0.8287
Seasonal ARIMA			
5-fold CV	AR(1)	998.8598	799.2195
	AR(2)	994.2633	795.3548
	AR(3)	990.6431	792.2216
	AR(4)	985.2711	788.0291
	AR(5)	980.2467	783.8283
noDepCV	AR(1)	1031.5958	824.9693
	AR(2)	1044.3668	834.4300
	AR(3)	1057.2591	844.1264
	AR(4)	1070.1811	853.8777
	AR(5)	1081.6456	861.8840
OOS	AR(1)	1017.8867	821.5384
	AR(2)	1021.8928	824.6080
	AR(3)	1026.2522	828.2727
	AR(4)	1032.8644	834.7562
	AR(5)	1036.5509	838.0672

AR(3) w/ break			
5-fold CV	AR(1)	1.5003	1.1644
	AR(2)	1.3879	1.1101
	AR(3)	1.3124	1.0326
	AR(4)	1.3077	1.0281
	AR(5)	1.3109	1.0293
noDepCV	AR(1)	1.6861	1.2779
	AR(2)	1.5566	1.1994
	AR(3)	1.4823	1.1524
	AR(4)	1.4558	1.1389
	AR(5)	1.5613	1.2296
OOS	AR(1)	2.2991	1.7937
	AR(2)	2.2148	1.7310
	AR(3)	1.8647	1.5012
	AR(4)	1.8647	1.5176
	AR(5)	1.9713	1.5948
MA(1) w/ break			
5-fold CV	AR(1)	1.3448	1.0713
	AR(2)	1.3529	1.0768
	AR(3)	1.3516	1.0768
	AR(4)	1.3487	1.0736
	AR(5)	1.3498	1.0737
noDepCV	AR(1)	1.4698	1.1686
	AR(2)	1.5530	1.2246
	AR(3)	1.5674	1.2315
	AR(4)	1.5484	1.2248
	AR(5)	1.5876	1.2634
OOS	AR(1)	1.7048	1.3878
	AR(2)	1.9061	1.5587
	AR(3)	1.9842	1.6147
	AR(4)	1.9208	1.5704
	AR(5)	1.9122	1.5635
AR(3) w/ AR errors			
5-fold CV	AR(1)	1.1147	0.8857
	AR(2)	1.0304	0.8186
	AR(3)	1.0220	0.8128
	AR(4)	1.0245	0.8158
	AR(5)	1.0276	0.8189
noDepCV	AR(1)	1.1367	0.8980
	AR(2)	1.0714	0.8420
	AR(3)	1.0937	0.8586
	AR(4)	1.1168	0.8745
	AR(5)	1.1316	0.8865
OOS	AR(1)	1.1512	0.9165
	AR(2)	1.0821	0.8662
	AR(3)	1.0760	0.8640
	AR(4)	1.0759	0.8644
	AR(5)	1.0756	0.8654
AR(3) w/ MA errors			
5-fold CV	AR(1)	1.2216	0.9764
	AR(2)	1.1007	0.8761
	AR(3)	1.0778	0.8553

	AR(4)	1.0583	0.8439
	AR(5)	1.0568	0.8422
noDepCV	AR(1)	1.2412	0.9908
	AR(2)	1.1490	0.9133
	AR(3)	1.1425	0.9025
	AR(4)	1.1241	0.8889
	AR(5)	1.1390	0.9011
OOS	AR(1)	1.2907	1.0569
	AR(2)	1.1678	0.9560
	AR(3)	1.1331	0.9254
	AR(4)	1.1120	0.9117
	AR(5)	1.1093	0.9091
MA(1) w/ AR errors			
5-fold CV	AR(1)	1.1745	0.9388
	AR(2)	1.1082	0.8871
	AR(3)	1.0816	0.8569
	AR(4)	1.0673	0.8483
	AR(5)	1.0657	0.8455
noDepCV	AR(1)	1.1904	0.9544
	AR(2)	1.1520	0.9233
	AR(3)	1.1543	0.9161
	AR(4)	1.1314	0.8991
	AR(5)	1.1341	0.8961
OOS	AR(1)	1.2679	1.0247
	AR(2)	1.1791	0.9663
	AR(3)	1.1349	0.9245
	AR(4)	1.1156	0.9081
	AR(5)	1.1146	0.9090
MA(1) w/ MA errors			
5-fold CV	AR(1)	1.4596	1.1695
	AR(2)	1.2738	1.0301
	AR(3)	1.2221	0.9855
	AR(4)	1.1561	0.9202
	AR(5)	1.1444	0.9068
noDepCV	AR(1)	1.4652	1.1707
	AR(2)	1.3315	1.0722
	AR(3)	1.2851	1.0415
	AR(4)	1.2188	0.9780
	AR(5)	1.2344	0.9775
OOS	AR(1)	1.6028	1.2850
	AR(2)	1.3565	1.1153
	AR(3)	1.2846	1.0610
	AR(4)	1.2154	0.9988
	AR(5)	1.1976	0.9846

Table A.2: RMSE, MAE measures, walk-forward testing. Number of simulations = 1000, sample size n = 500.

# Lags	RMSE	MAE
Stochastic AR(3)		
AR(1)	1.0479	0.8392

AR(2)	1.0044	0.8039
AR(3)	1.0029	0.8026
AR(4)	1.0043	0.8040
AR(5)	1.0061	0.8056
Stochastic MA(1)		
AR(1)	1.1467	0.9191
AR(2)	1.0873	0.8715
AR(3)	1.0574	0.8478
AR(4)	1.0426	0.8357
AR(5)	1.0332	0.8282
Seasonal ARIMA		
AR(1)	1019.3437	822.3744
AR(2)	1023.4679	825.3353
AR(3)	1027.6679	828.7279
AR(4)	1033.8075	834.4938
AR(5)	1037.2777	837.7937
AR(3) w/ break		
AR(1)	2.1499	1.6725
AR(2)	2.2944	1.7343
AR(3)	2.1222	1.6063
AR(4)	2.0909	1.6005
AR(5)	2.0892	1.6081
MA(1) w/ break		
AR(1)	1.6164	1.2959
AR(2)	1.8516	1.4779
AR(3)	2.0092	1.5750
AR(4)	2.0266	1.5815
AR(5)	2.0416	1.5891
AR(3) w/ AR errors		
AR(1)	1.0989	0.8770
AR(2)	1.0395	0.8392
AR(3)	1.0360	0.8424
AR(4)	1.0369	0.8441
AR(5)	1.0375	0.8446
AR(3) w/ MA errors		
AR(1)	1.2276	1.0088
AR(2)	1.1232	0.9281
AR(3)	1.0867	0.8973
AR(4)	1.0648	0.8837
AR(5)	1.0612	0.8808
MA(1) w/ AR errors		
AR(1)	1.2147	0.9927
AR(2)	1.1356	0.9404
AR(3)	1.0915	0.8984
AR(4)	1.0708	0.8812
AR(5)	1.0677	0.8829
MA(1) w/ MA errors		
AR(1)	1.5321	1.2269
AR(2)	1.3017	1.0794
AR(3)	1.2327	1.0239
AR(4)	1.1610	0.9592

AR(5)	1.1416	0.9448