Research in International Business and Finance 51 (2020) 101072

Contents lists available at ScienceDirect

Research in International Business and Finance

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

Full length Article

Forecasting financial time-series using data mining models: A simulation study[★]

Imad Bou-Hamad[a,*], Ibrahim Jamali[b]

[a] Department of Business Information and Decision Systems, Olayan School of Business, American University of Beirut, P.O. Box: 11-0263, Riad El-Solh, 1107-2020, Beirut, Lebanon
[b] Department of Finance, Accounting and Managerial Economics, Olayan School of Business, American University of Beirut, Beirut 1107 2020, P.O. Box 11-0236, Riad El-Solh Street, Lebanon

ARTICLE INFO

Keywords:
Random forests
Artificial neural networks
Static forecasting
Dynamic forecasting, Financial time series
Persistence
AR(1)-GARCH(1,1)

ABSTRACT

In this paper, we examine the static and dynamic predictive ability of artificial neural networks and random forests for financial time series within a simulation context. Our simulation design, in which we generate data from an AR(1)-GARCH(1,1) model, allows for several degrees of persistence in the mean equation to mimic the behavior of short and long-horizon asset returns. While the true data generating process beats the data mining techniques in terms of static forecasting, the novelty in this paper is to demonstrate that the data mining techniques outperform the true model under a dynamic forecasting scheme for moderate to highly persistent time series. We provide an empirical application using one-day and long-horizon returns on two exchange rates. Our empirical findings corroborate our simulation results in that the data mining models exhibit superior predictive ability for highly persistent time series. We discuss the importance of our findings for asset allocation and portfolio management.

1. Introduction

Financial time series exhibit well-known stylized features. One the one hand, asset prices are non-stationary (or unit root) processes while asset returns show little persistence. On the other hand, volatility clustering and non-normality (i.e., leptokurtosis) are salient features of asset returns. Since the seminal contribution of Engle (1982) and Bollerslev (1986), the literature in financial economics commonly employs conditional heteroskedasticity models, such as the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, to account for volatility clustering and non-normality in asset returns. In addition, financial economists routinely employ Autoregressive Moving Average (ARMA) models to predict asset returns (See, for example, Gospodinov and Jamali, 2011).

[★] We thank the editor-in-chief, an anonymous handling associate editor and two anonymous referees for numerous helpful comments and suggestions, which greatly improved the presentation and contents of the paper. Authors are listed in alphabetical order and this reflects an equal contribution to the manuscript. The second author gratefully acknowledges financial support from the University Research Board of the American University of Beirut.

* Corresponding author.
E-mail addresses: ib12@aub.edu.lb (I. Bou-Hamad), ij08@aub.edu.lb (I. Jamali).

https://doi.org/10.1016/j.ribaf.2019.101072
Received 26 August 2018; Received in revised form 23 July 2019; Accepted 25 July 2019
Available online 01 August 2019
0275-5319/

With the advent of big data, data mining techniques have become increasingly more popular for predicting financial and macroeconomic time series. Artificial Neural Networks (ANNs), in particular, have been extensively used in forecasting asset prices. Yu et al. (2010); Franses and Van Griensven (1998) and Franses and Van Homelen (1998) explore the predictive ability of ANNs in foreign exchange markets. Refenes et al. (1994); White (1988) and Kara et al. (2011) assess the forecasting ability of ANNs in equity markets while Swanson and White (1995) study the predictive ability of ANNs in fixed income markets. Maasoumi et al. (1994); Swanson and White (1997) and Teräsvirta et al. (2005) study the predictive power of ANNs for macroeconomic series.[1] Existing research which explores the predictive power of ANNs in commodity markets includes Kohzadi et al. (1996) and Dbouk and Jamali (2018). The predictive performance of classification techniques, such as random forests, have also been examined in the literature (Kumar and Thenmozhi, 2016; Bou-Hamad, 2017).

This paper offers a simulation study that assesses the predictive ability of ANNs and RFs for financial time series. Namely, we simulate data from an Autoregressive (AR) mean model whose conditional variance follows a Generalized Autoregressive Conditional Heteroskedasticity (GARCH). We allow several degrees of persistence in the mean (i.e., AR) equation to mimic the behavior of asset returns. More specifically, we simulate AR processes with high, medium and low persistence by varying the persistence (i.e, AR coefficient) of the mean equation. We do so to mimic the properties of the short and long-horizon asset returns, which exhibit low and high persistence, respectively, and assume that the variance of the errors (of the mean equation) follows a GARCH process to account for conditional heteroskedasticity in the variance of asset returns.

While a sizeable literature compares the predictive ability of data mining techniques for financial time series (i.e, prices or returns), only a few studies examine the forecasting ability of machine learning techniques using simulated data. Fischer et al. (2018) use a simulation setting to compare the predictive ability of time series models to machine learning techniques while McNelis (2005) examines the predictive performance of ANNs for financial time series using simulated data. Our study contributes to the latter stream of research by offering an assessment of the predictive power or ANNs and RFs using a simulation study. Knowing the true Data Generating Process (DGP) has the advantage of allowing for comparisons between the predictive ability of the "true" time series model and machine learning techniques. Our paper also relates to a strand of research which investigates the predictive power of artificial neural networks and data mining techniques for asset return volatility (Vortelinos, 2014, 2017) and bank failures (Le and Viviani, 2018; Wanke et al., 2016).

A number of interesting results emerge from our analysis and one of our results, in particular, is novel. Our findings suggest that, as expected, forecasting from the true DGP generally yields superior predictive performance, according to conventional statistical forecast accuracy measures, when a static (i.e., one-step-ahead rolling) forecasting scheme is used. Our findings also suggest that, interestingly, the data mining techniques appear to marginally underperform the true DGP in a dynamic (i.e., multistep-ahead) forecast exercise for low-persistence time series. However, as the persistence parameter is increased, the gains from employing data mining techniques become more pronounced. In fact, the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) are almost halved when forecasting highly persistent time series. This latter finding, which to the best of our knowledge is novel to the literature, suggest that forecasters obtain superior out-of-sample forecasting performance when predicting asset prices in a dynamic context.

The rest of the paper is organized as follows. We first introduce our simulation setup and our two forecasting schemes. We then present the predictive data mining models whose out-of-sample forecasting performance we compare to the true DGP. The statistical forecast accuracy measures that we employ are presented in the following section while our forecasting results and our discussion of the results are presented in the penultimate section. We then provide an empirical application using short and long-horizon exchange rate returns. The final section offers some concluding remarks.

2. Simulation design

We simulate data from an Autoregressive (AR) Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model. More specifically, the true DGP from which are data are simulated is an AR(1)-GARCH(1,1) given by:

$$y_t = \varphi y_{t-1} + u_t$$

$$u_t = v_t \sigma_t^2;\ v_t \tilde{} t(7)$$

$$\sigma_t^2 = \mu + \alpha_1 u_{t-1}^2 + \beta \sigma_{t-1}^2$$

The mean equation generating the simulated asset returns is a AR(1). The variance (of the mean model's) error term is assumed to follow a GARCH (1,1) model. As noted in the introduction, we elect to use the prior DGP given that asset returns exhibit pronounced conditional heteroskedasticity. The error term is assumed to follow a *t*-distribution to account for the leptokurtic distribution (i.e., fat tails) of asset returns.

We simulate samples of 1000, 2000, 5000 and 10,000 observations, respectively. That is, the sample sizes are progressively larger in each simulation. The values of the parameters in the variance equation are: $\mu = 0.02$, $\alpha_1 = 0.04$, $\beta = 0.95$. We calibrate the

[1] McNelis (2005) provides a textbook treatment of predicting financial time series using ANNs.

parameters of the variance equation of the model as well as the degrees of freedom of the *t*-distribution of the error term based on our estimation results for the exchange rate returns of the Australian Dollar and Swiss Franc against the US dollar. We vary the persistence parameter of the mean equation's AR model, φ, in every simulation. More specifically, φ is assumed to take on the values -0.04, 0.1, 0.5 or 0.9 to mimic the behavior of very low, low, medium and high persistence asset returns, respectively.

3. Forecasting methodology

In order to assess the out-of-sample forecasting ability of the competing models, we delineate our simulated samples into in-sample (or training) and out-of-sample (or validation) period. The in-sample period is used to estimate (or train) the models while the out-of-sample period is preserved for out-of-sample (or validation) assessment of the forecasting ability of the models. Throughout our forecast assessment exercise, our validation period is selected to be the last 252 observations. This amounts to simulated daily return data for one year. That is, we generate daily forecasts of return for one year of trading.

We generate the out-of-sample forecasts from our true DGP and data mining techniques using two forecasting schemes. Under the first, rolling one-step-ahead forecasts are generated from the true DGP and the data mining techniques by using a rolling (fixed) window of varying sizes. When the sample size is 1000 observations, for example, the window size is 747 observations while it is 1747, 4747 and 9747 observations when the sample sizes are 2000, 5000 and 1000 observations, respectively. The rolling window forecasting scheme drops the oldest observation and adds the newest observation so as to maintain a constant window size. For example, for a sample size of 1000, the forecast of y_t for observation 748 is generated by estimating the model using observations 1 to 747 and generating a one-step-ahead forecast. Similarly, the forecast of y_t for observation 749 is generated by estimating the model using observations 2 to 748.

This forecasting scheme is referred to as the *static* forecasting. Under the second forecasting scheme, the models are estimated over the entirety of the training period and multi-step-ahead (i.e., 252) forecasts are generated from the model. The latter forecasting scheme is referred to as *dynamic* forecasting. With a dynamic forecasting scheme, forecasts generated for earlier periods are used as lagged dependent variables to generate multi-step-ahead forecasts whereas a static forecasting scheme only uses actual values to generate the forecasts.

4. Predictive data mining models

4.1. Random forests (RF)

A random forest is a popular predictive data mining technique that was introduced by Breiman (2001). More specifically, an RF is a tree-based ensemble method used to predict either a categorical response (classification) or a numerical response (regression) variable. The advantage of an RF over a single tree is that an RF, like any ensemble learning tools, boosts the predictive performance of a weak learner (a single tree) via a voting (classification) or averaging (regression) scheme. We focus here on the regression aspect of an RF since our response is a numerical variable. More formally, let Y designate a numerical response and $X = (X_1, ..., X_p)^T$ a p-dimensional vector of predictors assuming an unknown joint distribution $P_{XY}(X, Y)$. The aim is to estimate a prediction function $f(x)$ for Y that minimizes the expected value of a loss function $Loss(Y, f(X))$

$$E_{XY}(Loss(Y, f(X))$$

Goldstein et al. (2010) point out that minimizing the expected value for squared error loss produces the condition expectation:

$$f(X) = E(Y|X = x)$$

Ensembles build f f based on a set of so-called “base learners” $l_1(x), ..., l_K(x)$. A Kth base learner in a random forest is a tree denoted by $l_k(X, \theta_k)$, where θ_k is a collection of random variables that are independent for $k = 1, ..., K$. In regression the base learners are averaged to compute the prediction function. Hence, the predicted Y is given by

$$f(x) = \frac{1}{K}\sum_{k=1}^{K} l_k(x)$$

The trees forming a random forest are built using the binary recursive partitioning algorithm presented in Breiman et al. (1984). Recursive partitioning constructs a decision tree by partitioning each node in the tree into two descendants based on a splitting criterion. The nodes that are not further split are known as terminal nodes and constitute the final partition of the predictor space. The typical splitting criterion in the context of regression is the mean squared error at the node

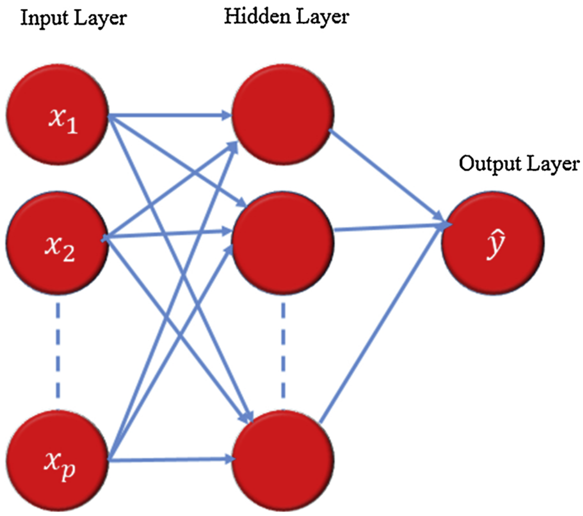

Fig. 1. Architecture of an Artificial Neural Network.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y - \bar{y})^2$$

where n is the number of observations in the node. Let D denote a training dataset of size N. The general algorithm of RF goes as follows:

1 Take B bootstrap samples of size N from D
2 Fit a tree for each bootstrap sample using the binary recursive partitioning
3 Select randomly mtree predictors from the available p predictors
4 Find the best split among all splits on the *mtree* predictors and split the node into two descendants
5 Each tree is fully-grown and not pruned. The splitting is stopped when a minimum node size is reached

The final prediction of the response variable is the mean prediction of individual trees forming the random forest.

4.2. Artificial neural networks

Artificial Neural Networks (ANNs) are machine learning-based techniques widely used for predictive data mining tasks. ANNs mimic the structure of biological neural networks where neurons are interconnected and learn from experience. An artificial neural network architecture consists of nodes (neurons) arranged in layers that are interconnected through a system of weights. Various neural network architectures have been developed in the literature (Krenker et al., 2011). However, the famous type that has been used to solve most of the problems is feed-forward neural networks. Such networks include an input layer, one or more hidden layers and an output layer. Fig. 1 illustrates a schematic diagram of a feed-forward neural network with one hidden layer. Input nodes take as input the values of the predictors. Their outputs do not change from their inputs. Hidden layer nodes take as input the processed output from the input layer. Hence, the input of a hidden layer node is computed as follow. First, a weighted sum of inputs is computed and an activation function is applied to this sum. For example, consider the values of a vector of predictors $(x_1, \ldots, x_p)^T$, the output of an input layer node j is computed as follows:

$$\alpha_j + \sum_{i=1}^{p} w_{ij}x_i$$

where $\alpha_{j,} w_{1,j}, \ldots, w_{pj}$ are weights that take initially random values and adjusted as the network learns. The next step is to apply an activation function g to this sum. The most popular activation function is logistic function

$$g(s) = \frac{1}{1 + e^{-s}}$$

Finally, the output layer obtains input values from the hidden layer and the same activation function is applied to create the output. Fig. 1 depicts the typical architecture of ANN.

The common algorithm used to estimate and update the weights is the back-propagation algorithm (Rumelhart et al., 1986). Multiple studies suggest using this algorithm (Haykin, 1994; Reed and Marks, 1999; Rojas, 2013). However, this method suffers from a low learning speed (Castillo et al., 2006). Several alternatives have been developed to increase the learning speed including Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which is used in this paper.

The main parameters in a random forest that may require tuning are the number of randomly selected predictors (mtree) for splitting at each node, and the number of trees in the forest (ntree). Although there is no explicit theory supporting the tuning of these two parameters, the default of mtree is P/3 in regression RF, where P is the total number of predictors (Cutler et al., 2012). In our case, we consider 12 lags (predictors), hence mtree is set to a value of 4. As for the number of trees in the forest, generalization error decreases in general as ntree increases. Oshiro et al (2012) recommend a number of trees between 64 and 128. We tried different settings of the number of trees and found that 100 trees were enough to produce accurate estimates. In MLP, one hidden layer with sufficient large number of nodes is enough for the "universal approximation" property (Hornik et al., 1989; Hornik, 1993). As for the number of nodes in the hidden layer, Blum (1992, p. 60) suggests, as a rule of thumb, a value between the input layer size and the output layer size. The input layer size is the number of predictors (12 lags) and the output layer size is the number of response variables (here is one). Therefore, we used a number of hidden nodes between 1 and 12. The lowest error obtained with seven nodes in the hidden layer. We used the packages "randomForest" and "nnet" in the R programming language to perform our analyses.

4.3. Assessing forecast accuracy

We employ conventional statistical accuracy measures to assess the out-of-sample forecast accuracy of the competing models. The first statistical accuracy measure which we employ is the Root Mean Squared Error (RMSE) given by:

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

where y_t and $\hat{y}_t$ are, respectively, the actual and forecast values at time t, respectively and T_f is the number of observations in the validation (or out-of-sample forecast) dataset. The RMSE uses a quadratic loss function and thereby weights under and over-forecasts of the same magnitude in the same way.

The second statistical forecast accuracy measure that we use is the Mean Absolute Error (MAE). The MAE employs an absolute loss function and is given by:

$$MAE = \frac{1}{T_f} \sum |y_t - \hat{y}_t|$$

While ranking models based on RMSE and MAE provides a good first indication of the statistical forecast accuracy of the competing models, a more careful assessment of whether the differences in loss functions among the competing models are statistically significant is warranted. In order to assess whether the differences in Mean Squared Errors (MSEs) and MAEs among the models are significant, we employ the Diebold and Mariano (1995), henceforth DM, test.

Let $d_{t+1} \equiv \Delta L_{t+1} = L(y_{t+1} - \hat{y}_{t+1}) - L(y_{t+1} - \hat{y}_{t+1}^B)$ denote the difference in loss functions. More specifically, let $L(y_{t+1} - \hat{y}_{t+1})$ denote the loss function of the RF or ANN models and $L(y_{t+1} - \hat{y}_{t+1}^B)$ denote the loss function of our benchmark model, denoted by "B", which is the AR(1)-GARCH(1,1). The null of equation predictive accuracy that the DM statistic tests is H_0: $E(d_{t+1}) = 0$.

The DM statistic is given by:

$$DM = {}^{\bar{d}}\!/_{\hat{\sigma}_{\bar{d}}}$$

where $\bar{d}$ is the sample mean of the differences in the loss function and $\hat{\sigma}_{\bar{d}}$ is a consistent estimate of the standard deviation of $\bar{d}$.

Given that the data mining models and the AR(1)-GARCH(1,1) models are non-nested, the DM statistic follows a standard normal distribution.[2] We report a modified version of the DM statistic. The modification, proposed by Harvey et al. (1997), consists of a degrees of freedom adjustment of the variance of DM statistic. In line with the suggestions of Harvey et al. (1997), we use the critical values from the t-distribution.

[2] When the models are nested, the DM statistic follows a non-standard distribution.

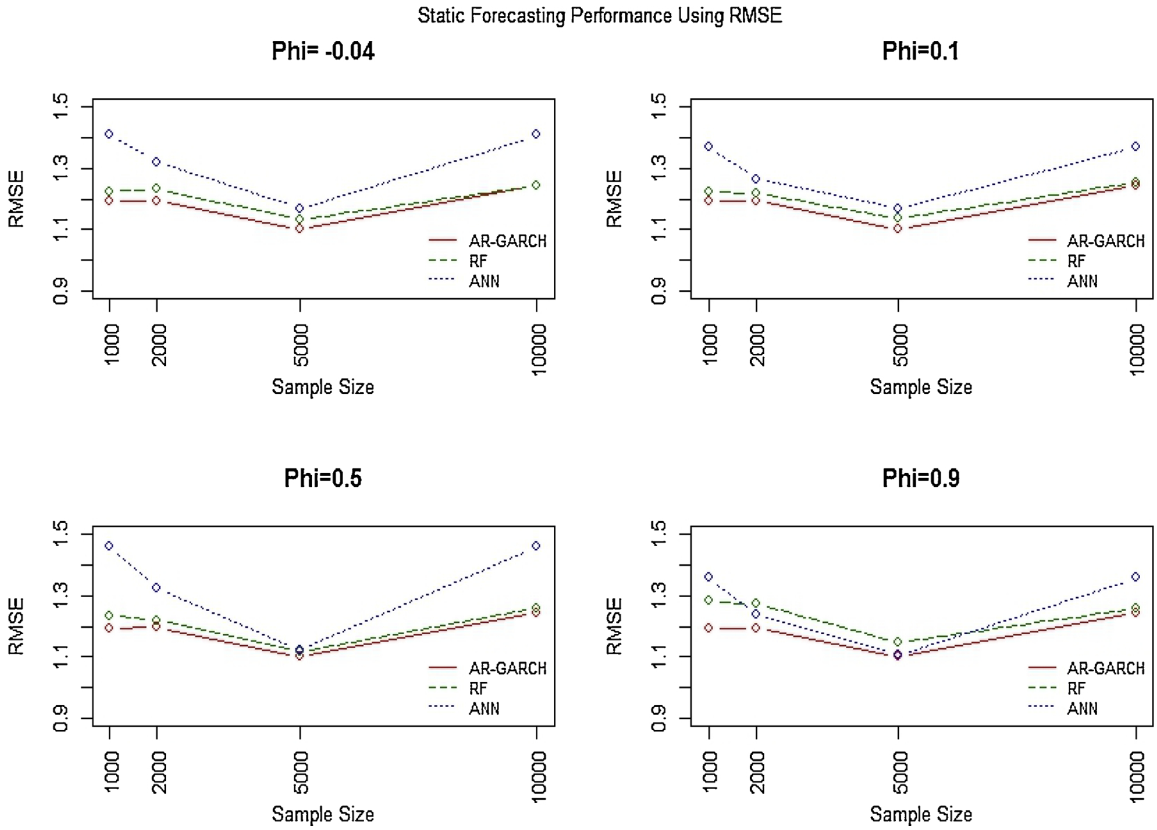

Fig. 2. The figure presents the RMSEs of the one-step-ahead (i.e., static) forecasts for the three models as a function of the sample size and for different values of the parameter phi, which is the autoregressive (or persistence) parameter.

5. Statistical forecast accuracy: results and discussion

5.1. Assessment of static forecast accuracy

Our analysis of the results commences with a visual inspection of the RMSEs and MAEs for the one-step-ahead (i.e. static) forecasts of the three models. More specifically, Figs. 2 and 3 display the RMSE and MAE of each of the models as a function of the sample size for each value of the persistence parameter.

The same information is provided in tabular format in Table 1.

Figs. 2 and 3 show that the AR(1)-GARCH(1,1) outperforms the competing models in terms of out-of-sample forecast accuracy according to the RMSE and the MAE. The same observation emerges from Table 1. In all scenarios of different values of phi and sample sizes, the RMSE and MAE of the AR(1)-GARCH(1,1) are consistently lower than that of the RF and ANN. This result is in line with *a priori* expectations. We test next for whether the differences in MAE and RMSE among the true DGP and the two data mining techniques are significant.

5.2. Assessment of dynamic forecast accuracy

We turn next to assessing the multi-step-ahead forecast accuracy of the models. Figs. 4 and Figure 55 present the RMSE and MAE of each of the models as a function of the sample size for each value of the persistence parameter while Table 2 presents the same results in a tabular form.

A very interesting result emerges from comparing the predictive accuracy of the true DGP to that of the data mining techniques.

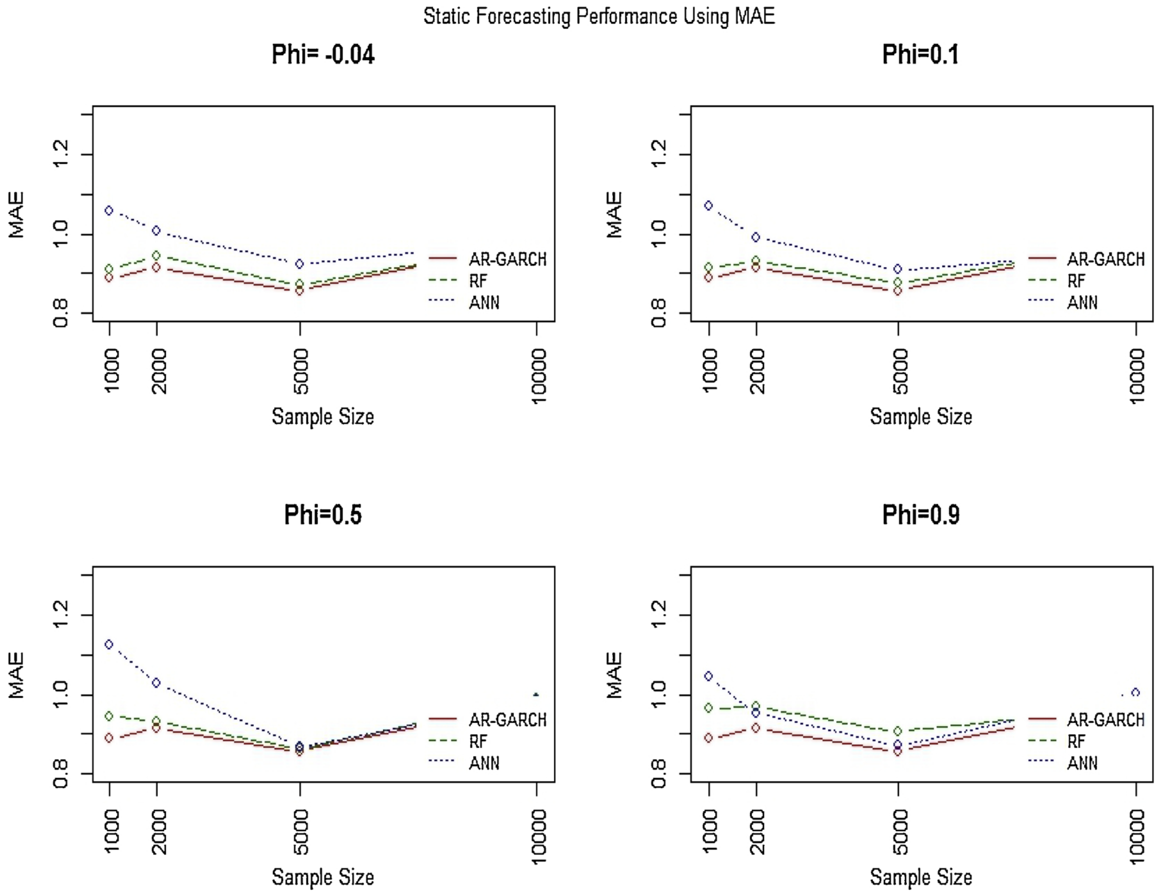

Fig. 3. The figure presents the MAEs of the one-step-ahead (i.e., static) forecasts for the three models as a function of the sample size and for different values of the parameter phi, which is the autoregressive (or persistence) parameter.

Table 1
Statistical Forecast Accuracy for Static Forecasts.

	φ = -0.04			φ = 0.1			φ = 0.5			φ = 0.9		
Sample Size	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN
Panel A: Root Mean Squared Error												
1000	1.194	1.225	1.406	1.194	1.223	1.367	1.194	1.235	1.457	1.195	1.281	1.356
2000	1.195	1.231	1.319	1.195	1.218	1.265	1.196	1.219	1.322	1.195	1.271	1.240
5000	1.102	1.133	1.169	1.102	1.136	1.169	1.102	1.117	1.121	1.102	1.149	1.108
10,000	1.245	1.245	1.406	1.245	1.254	1.367	1.245	1.258	1.457	1.245	1.259	1.356
Panel B: Mean Absolute Error												
1000	0.890	0.912	1.058	0.890	0.913	1.068	0.890	0.945	1.124	0.890	0.964	1.043
2000	0.914	0.944	1.006	0.914	0.929	0.991	0.914	0.933	1.026	0.914	0.970	0.952
5000	0.857	0.871	0.924	0.857	0.876	0.910	0.858	0.865	0.867	0.857	0.907	0.872
10,000	0.978	0.978	0.982	0.978	0.978	0.958	0.978	0.990	0.984	0.978	0.969	1.002

Notes: The table provides the Root Mean Squared Errors (RMSEs) and Mean Absolute Errors (MAEs) for the one-step-ahead (i.e., static) forecasts from the competing models. The true Data Generating Process is the AR(1)-GARCH(1,1) model.

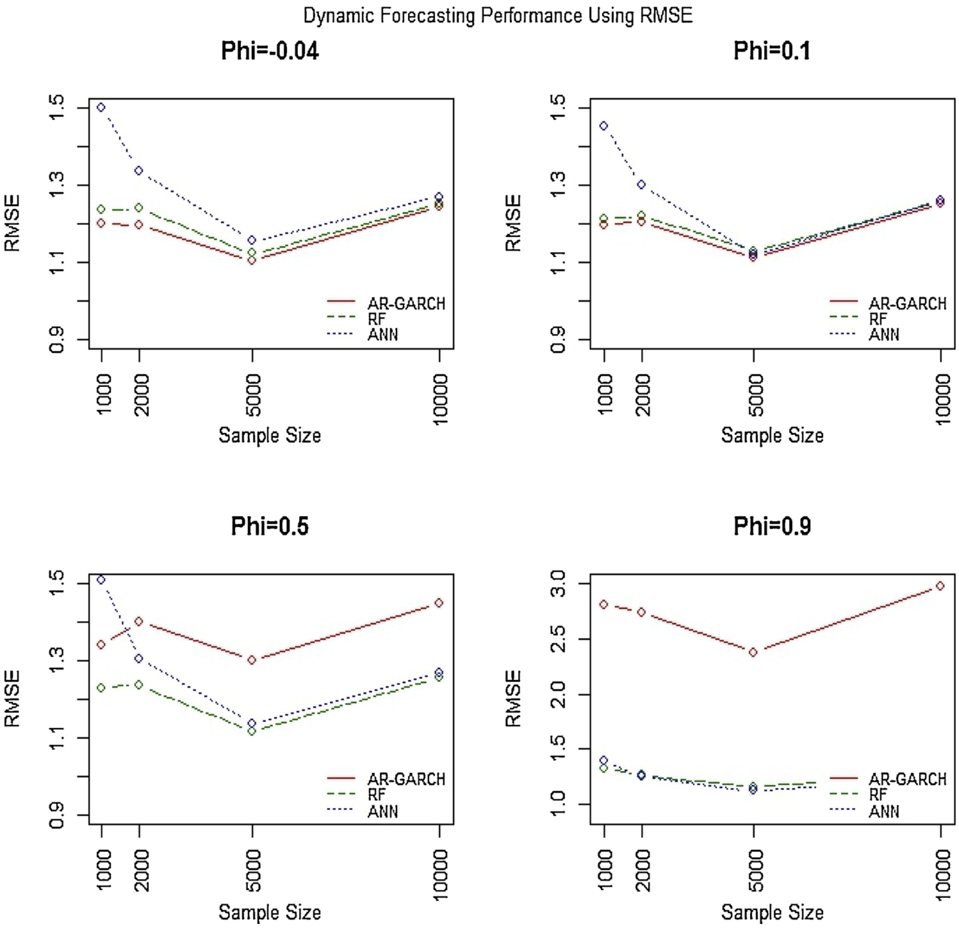

Fig. 4. The figure presents the RMSEs of the multi-step-ahead (i.e., static) forecasts for the three models as a function of the sample size and for different values of the parameter phi, which is the autoregressive (or persistence) parameter.

While the AR(1)-GARCH(1,1) continues to outperform the data mining techniques, and particularly the ANN, for a low persistence parameter (φ = -0.04), the RMSE and MAE of the RF are lower than that of the true DGP for all sample sizes when the persistence parameter is 0.5 or 0.9. Similarly, for sample sizes of 2000 and above, the RMSE and MAE of the ANN are lower than those of the true DGP albeit the difference in amplitude is not large. The lower error of RF and ANN is more pronounced when the persistence parameter is very high (0.9).

Our most striking result arises when we compare the out-of-sample predictive accuracy of the AR(1)-GARCH(1,1) model to that of the RF and ANN when the persistence parameter is high. In fact, when the true DGP exhibits high persistence, the out-of-sample dynamic predictive accuracy of the RF and ANN are remarkably better than that of the AR(1)-GARCH(1,1). The RMSE and MAE of the data mining techniques are almost half those of the AR(1)-GARCH(1,1).

A potential explanation of the superior dynamic predictive ability is the mean reversion of long-horizon forecasts of the AR(1)-GARCH(1,1) models. That is, while long-horizon forecasts from AR(1)-GARCH(1,1) revert back to their mean, the data mining techniques may better capture the dynamics at long horizons since these models do not impose a mean reverting behavior on the long-run forecasts.

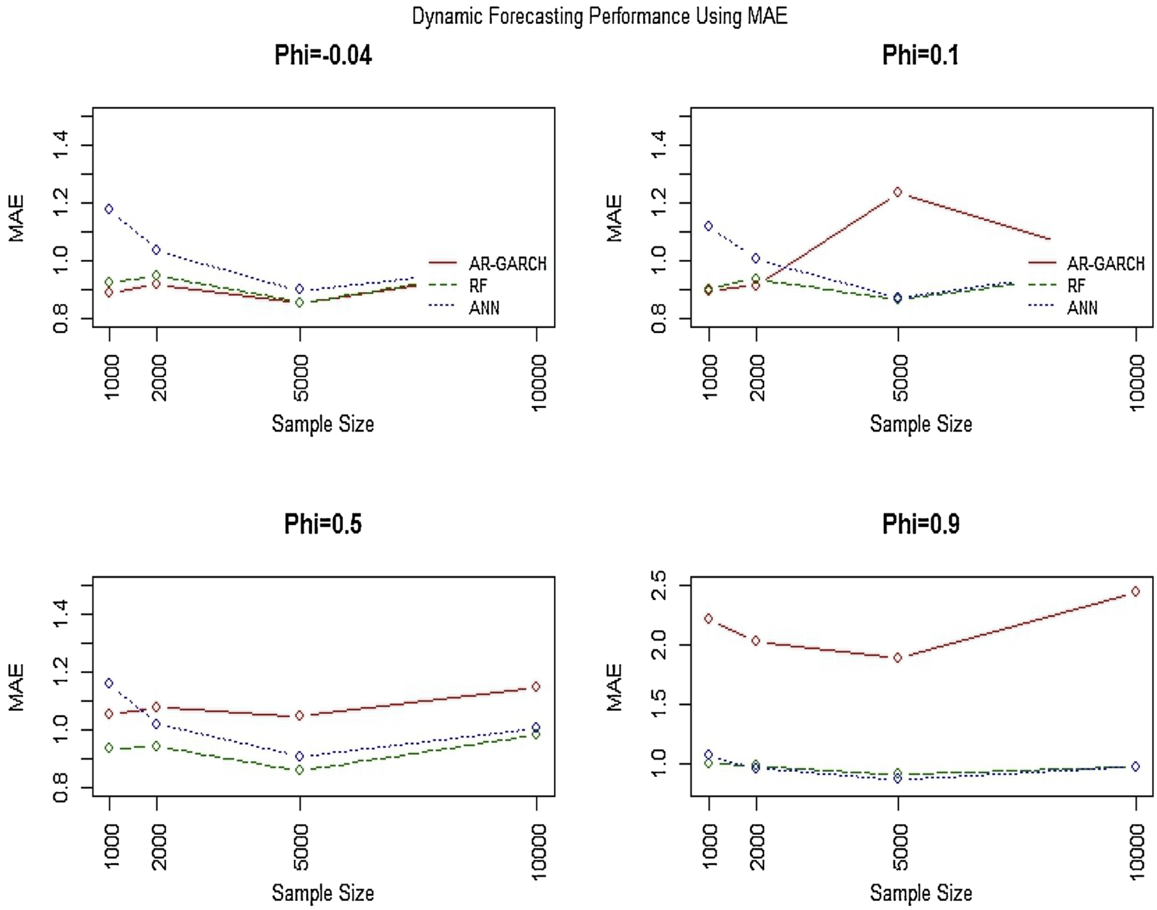

Fig. 5. The figure presents the MAEs of the multi-step-ahead (i.e., static) forecasts for the three models as a function of the sample size and for different values of the parameter phi, which is the autoregressive (or persistence) parameter.

Table 2
Statistical Forecast Accuracy for Dynamic Forecasts.

	φ = -0.04			φ = 0.1			φ = 0.5			φ = 0.9		
Sample Size	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN	AR-GARCH	RF	ANN
Panel A: Root Mean Squared Error												
1000	1.199	1.233	1.498	1.195	1.211	1.452	1.341	1.231	1.511	2.807	1.321	1.395
2000	1.196	1.237	1.334	1.204	1.218	1.298	1.402	1.238	1.306	2.745	1.269	1.253
5000	1.102	1.121	1.154	1.110	1.129	1.119	1.302	1.119	1.138	2.381	1.162	1.121
10,000	1.244	1.251	1.269	1.253	1.259	1.257	1.450	1.256	1.269	2.983	1.265	1.254
Panel B: Mean Absolute Error												
1000	0.890	0.922	1.174	0.897	0.903	1.117	1.055	0.938	1.158	2.205	1.006	1.079
2000	0.918	0.950	1.036	0.916	0.938	1.005	1.077	0.941	1.018	2.030	0.984	0.963
5000	0.857	0.858	0.899	1.233	0.864	0.873	1.048	0.863	0.910	1.889	0.915	0.877
10,000	0.976	0.991	0.989	0.985	0.979	0.987	1.146	0.981	1.005	2.440	0.974	0.981

Notes: The table provides the Root Mean Squared Errors (RMSEs) and Mean Absolute Errors (MAEs) for the one-step-ahead (i.e., static) forecasts from the competing models. The true Data Generating Process is the AR(1)-GARCH(1,1) model.

5.3. Testing for equal predictive accuracy

In order to ascertain that the predictive gains from using data mining techniques are statistically significant for long-horizon forecasts of highly persistent time series, we test for equal predictive accuracy using the DM test. Tables 5 and 6 provide, respectively, the results of the DM test for static and dynamic forecasts. As noted before, we use the Harvey et al. (1997) variant of the DM test for quadratic and absolute value loss functions (Tables 3 and 4).

In line with our earlier observations, the DM test results confirm that, when assessing dynamic forecast accuracy, the difference in

Table 3
Tests of Equal Predictive Accuracy for Static (i.e., rolling one-step-ahead) Forecasts.

	φ = -0.04		φ = 0.1		φ = 0.5		φ = 0.9	
Sample Size	RF	ANN	RF	ANN	RF	ANN	RF	ANN
Panel A: Quadratic Loss Function (MSE)								
1000	−2.361	−4.419	−1.916	3.336	−1.332	−4.517	−2.189	−3.802
2000	−3.313	−2.970	−2.344	2.454	−1.578	−3.854	−2.290	−2.826
5000	−2.480	−2.966	−2.233	−4.780	−1.409	−1.068	−2.153	−0.318
10,000	0.068	−0.497	−0.651	0.951	−1.068	−1.839	−0.564	−2.184
Panel B: Absolute Value Loss Function (MAE)								
1000	−1.806	−3.828	1.227	−5.753	−2.239	−4.836	−2.356	−3.998
2000	−3.094	−2.556	1.155	−2.907	−1.147	−4.308	−2.123	−2.428
5000	−1.206	−3.459	1.221	−2.699	−0.707	−0.639	−2.436	−1.158
10,000	0.096	−0.247	−0.007	1.498	−1.442	−0.610	0.410	−1.798

Notes: The table provides the results of the Harvey et al. (1997) of Diebold and Mariano (1995) test of equal predictive accuracy. RF refers to the Random Forest forecast and ANN refers to the Artificial Neural Network forecast. The RF and ANN's predictive accuracy is compared to that of the AR(1)-GARCH(1,1) model, which is the true Data Generating Process (DGP). *, **, *** denote, respectively, statistical significance at the 10%, 5% and 1% levels.

Table 4
Tests of Equal Predictive Accuracy for Dynamic (i.e., multi-step-ahead) Forecasts.

	φ = -0.04		φ = 0.1		φ = 0.5		φ = 0.9	
Sample Size	RF	ANN	RF	ANN	RF	ANN	RF	ANN
Panel A: Quadratic Loss Function (MSE)								
1000	−1.662	−6.156	−0.989	−4.410	2.506***	−3.308	2.811***	2.644***
2000	−2.844	−4.685	−2.414	−4.671	2.770***	1.500*	2.603***	2.475***
5000	−1.452	−3.877	−1.234	−0.260	3.720***	3.345***	4.386***	4.319***
10,000	−0.519	−1.652	−0.377	−0.216	3.905***	3.177***	4.268***	4.228***
Panel B: Absolute Value Loss Function (MAE)								
1000	−1.683	−6.658	−0.343	−5.399	3.550***	−1.970	3.923***	3.489***
2000	−2.221	−4.084	−1.717	−3.574	3.409***	1.141*	3.790***	3.472***
5000	−0.000	−3.367	0.261	−0.214	4.507***	3.536***	5.084***	5.136***
10,000	−1.206	−0.807	0.449	−0.066	3.379***	2.565***	5.451***	4.827***

Notes: The table provides the results of the Harvey et al. (1997) of Diebold and Mariano (1995) test of equal predictive accuracy. RF refers to the Random Forest forecast and ANN refers to the Artificial Neural Network forecast. The RF and ANN's predictive accuracy is compared to that of the AR(1)-GARCH(1,1) model, which is the true Data Generating Process (DGP). *, **, *** denote, respectively, statistical significance at the 10%, 5% and 1% levels.

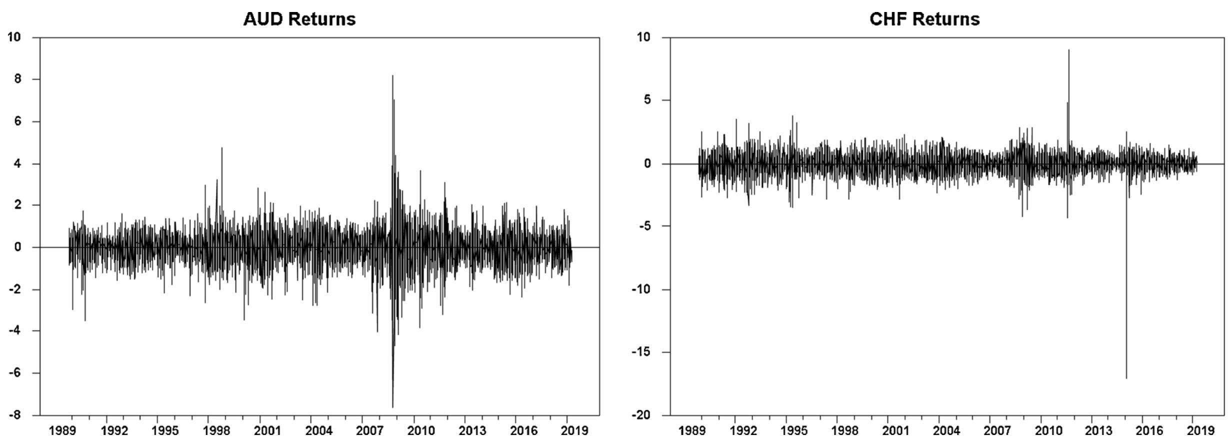

Fig. 6. Time series dynamics of one-day returns on AUD and CHF.

MSEs and MAEs between the data mining techniques and the AR(1)-GARCH(1,1) are significant, at the 1% level, for all sample sizes when the persistence parameter $\varphi = 0.9$. More specifically, our results clearly indicate that the data mining techniques outperform the true DGP.

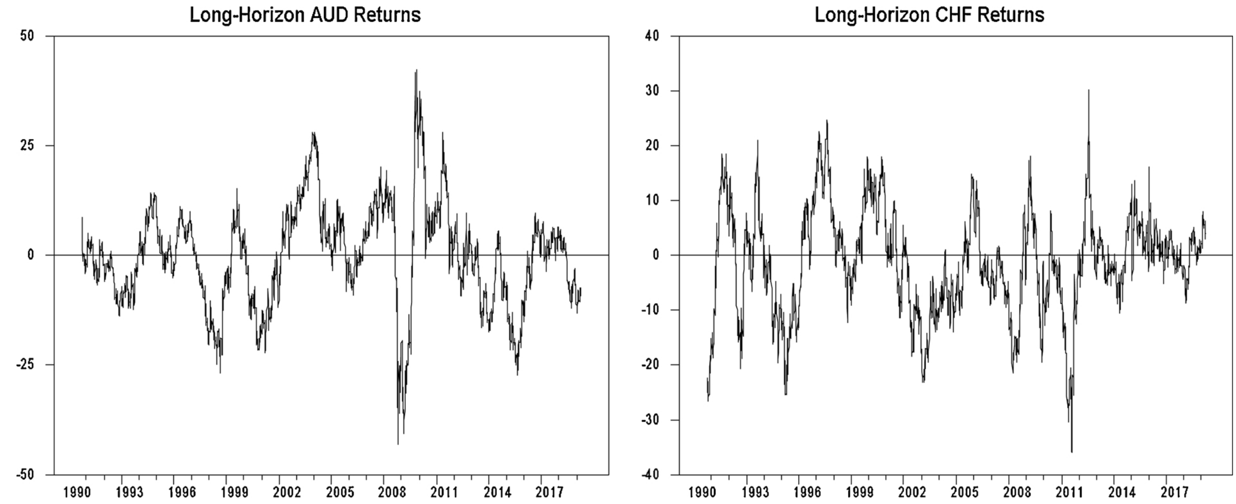

Fig. 7. Time series dynamics of long-horizon returns on AUD and CHF. Long-horizon returns are cumulative exchange rate returns over a 252-day period.

Table 5
Full-sample estimation results for exchange rate returns.

	AUD	CHF
Panel A: One-Day Returns		
Mean Equation		
ω	0.013***	0.002
	0.005	0.006
Ø	−0.033***	−0.042***
	0.011	0.0100
Variance Equation		
μ	0.002***	0.002***
	0.000	0.000
α_1	0.045***	0.033***
	0.005	0.004
β	0.950***	0.961***
	0.005	0.004
Degrees of Freedom		
ν	6.967***	6.223***
	0.521	0.549
Panel B: Long-Horizon Returns		
Mean Equation		
ω	−0.008	−0.006
	0.009	0.009
Ø	0.996***	0.995***
	0.000	0.001
Variance Equation		
μ	0.007***	0.009***
	0.001	0.002
α_1	0.041***	0.025***
	0.004	0.003
β	0.951***	0.963***
	0.005	0.005
Degrees of Freedom		
ν	11.287***	7.787***
	1.324	0.608

Notes: This table provides the full-sample estimation results for the AR(1)-GARCH(1,1) model with *t*-distributed errors for AUD and CHF returns. The results are provided for one-day and long-horizon returns.

When the persistence parameter is 0.5, our results also point to some success of the RF and ANN's dynamic forecasts vis-à-vis the AR(1)-GARCH(1,1). Namely, under a quadratic loss function, the RF dynamic forecasts outperform the true DGP at the 5% level when the sample size is 1000, 2000 and 10,000 while the ANN outperforms the AR(1)-GARCH(1,1) at the 5% level or better for sample sizes of 5000 and 10,000. Similarly, the RF outperforms the true DGP at the 1% level for sample sizes of 1000, 2000 and 10,000. In

Table 6
Out-of-Sample Statistical Accuracy of Forecasts of AUD and CHF Returns.

	AUD		CHF	
	RMSE	MAE	RMSE	MAE
Panel A: One-Day Returns (Low Persistence)				
Static Forecasts				
AR-GARCH	0.541	0.424	0.355	0.269
RF	0.563	0.443	0.359	0.276
ANN	0.551	0.433	0.370	0.285
Dynamic Forecasts				
AR-GARCH	0.542	0.425	0.356	0.269
RF	0.553	0.434	0.362	0.278
ANN	0.545	0.431	0.366	0.278
Panel B: Long-Horizon Returns (High Persistence)				
Static Forecasts				
AR-GARCH	0.727	0.584	0.585	0.442
RF	0.763	0.610	0.635	0.488
ANN	0.747	0.599	0.592	0.452
Dynamic Forecasts				
AR-GARCH	8.450	7.645	6.121	5.863
RF	0.769	0.616	0.647	0.488
ANN	0.750	0.599	0.610	0.464

Notes: The table provides the RMSE and MAE of the static and dynamic forecasts of the competing models for one-day and long-horizon returns.

contrast, with the exception of the ANN for sample size of 1000 and a persistence parameter of 0.9, the RF and ANN fail to outperform the AR(1)-GARCH(1,1) under a static forecasting scheme. In sum, the DM test results corroborate the superior performance of the data mining techniques relative to the true DGP for moderately or highly persistent time series under a dynamic forecasting scheme.

6. Empirical application

We collect daily observations on the exchange rates of the Australian Dollar (AUD) and Swiss Franc (CHF) against the US Dollar from Datastream for the period October 16, 1989 and March 28, 2019. Continuously compounded one-day returns on each of the two exchange rates are computed as:

$$r_t = 100 \times \ln(\frac{P_t}{P_{t-1}})$$

The one-day exchange rate returns are known to exhibit low persistence. We also work with one-year (i.e, long-horizon) returns which we compute as the moving sum of one-day returns over a 252-day period.[3] These are cumulative returns on the two exchange rates over a one-year horizon. The long-horizon returns exhibit persistence that is close to one (i.e., high persistence). The time series dynamics of the one-day and long-horizon returns are provided in Figs. 6 and Figure 77.

The AR(1)-GARCH(1,1) model is given by:

$$r_t = \omega + \varphi r_{t-1} + u_t$$

$$u_t = v_t \sigma_t^2;\ v_t \tilde{}\, t(v)$$

$$\sigma_t^2 = \mu + \alpha_1 u_{t-1}^2 + \beta \sigma_{t-1}^2$$

where v is the degrees of freedom parameter. The full-sample estimation results of the AR(1)-GARCH(1,1) model for the one-day and long-horizon returns are provided in Table 5.

The estimation results provided in Table 5 show that one-day returns exhibit little persistence as evinced by a highly significant coefficient φ of -0.03 for AUD and -0.04 for CHF. In contrast, the long-horizon returns exhibit high persistence with a coefficient φ of 0.996 for AUD and 0.995 for CHF.

In line with our simulation design, we estimate the AR(1)-GARCH(1,1), RF and ANN models over an in-sample period and generate static and dynamic forecasts over an out-of-sample period consisting of 252 observations. In our empirical application, the in-sample period is October 16, 1989 to April 10, 2018 when we employ one-day returns while it is December 10, 1990 to April 10, 2018 when long-horizon returns are employed. The out-of-sample forecasting period is April 11, 2018 to March 28, 2018. The RMSE and MAE of the competing models are provided in Table 6.

The results in Table 6 show that the statistical accuracy of the AR(1)-GARCH(1,1) forecasts is comparable to, or higher than, that of the RF and ANN. This is true for static forecasts of one-day and long-horizon returns as well as for the dynamic forecasts of the one-

[3] Given that we use continuously compounded returns, the returns are additive across time.

day returns. In contrast, the dynamic forecasts of long-horizon returns generated from the RF and ANN exhibit significantly higher statistical accuracy than the AR(1)-GARCH(1,1) forecasts.[4] In sum, our empirical results confirm the conclusions of the simulation study in that the data mining models exhibit superior predictive ability for time series with high persistence.

7. Concluding remarks

This paper examines the static and dynamic predictive ability of artificial neural networks and random forests for financial time series within a simulation context. We simulate data from an Autoregressive (AR) mean model whose conditional variance follows a Generalized Autoregressive Conditional Heteroskedasticity (GARCH). Our simulation design allows for several degrees of persistence in the mean (i.e., AR) equation to mimic the behavior of short and long-horizon asset returns. More specifically, we simulate AR processes with high, medium and low persistence by varying the persistence of the mean equation. We also assume that the variance of the errors (of the mean equation) follows a GARCH process to account for conditional heteroskedasticity in the variance of asset returns.

We generate the out-of-sample forecasts from the true AR(1)-GARCH(1,1) and the data mining techniques using two forecasting schemes. The first is a static forecasting scheme under which we generate one-step-ahead out-of-sample forecasts by using a rolling window of fixed size which drops the oldest observation and adds a new observation. The second forecasting scheme is a dynamic one under which we generate multi-step-ahead forecasts for the entire validation period.

Strikingly, our results indicate that the data mining techniques outperform the true model, namely the AR(1)-GARCH(1,1), under a dynamic forecasting scheme for moderate to highly persistent time series. The superior performance of the data mining techniques becomes more pronounced the higher the degree of persistence is. The latter result is novel to the literature and suggests that portfolio managers making long-horizon portfolio allocation decisions or risk managers wanting to assess portfolio risk over a one-year horizon can obtain more accurate forecasts of the portfolio return dynamics from data mining techniques than from the true model.

We provide an empirical application which involves forecasting the one-day and long-horizon returns on the AUD/USD and CHF/USD exchange rates. Our empirical results confirm our conclusions from the simulation study in that the data mining methods exhibit superior predictive ability for financial time series with high persistence. Our results have important implications for asset allocation. Portfolio managers conducting asset allocation over a one-year horizon can generate better forecasts of asset returns using data mining models. The more accurate forecasts will imply a lesser need for portfolio rebalancing and hence lower transaction costs and higher returns.

References

Blum, A., 1992. Neural Networks in C++. Wiley, NY, pp. 697.
Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Kluwer Academic Publishers, New York. NY.
Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.
Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. J. Econometrics 31, 307–327.
Bou-Hamad, I., 2017. Bayesian credit ratings: A random forest alternative approach. Commun. Stat. - Theory Methods 46, 7289–7300.
Castillo, E., Guijarro-Berdiñas, B., Fontenla-Romero, O., Alonso-Betanzos, A., 2006. A very fast learning method for neural networks based on sensitivity analysis. J. Mach. Learn. Res. 7, 1159–1182.
Cutler, A., Cutler, D.R., Stevens, J.R., 2012. Random forests. Ensemble Machine Learning. Springer, Boston, MA.
Dbouk, W., Jamali, I., 2018. Predicting daily oil prices: linear and non-linear models. Res. Int. Bus. Finance 46, 149–165.
Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. J. Bus. Econ. Stat. 20, 134–144.
Engle, R.F., 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50, 987–1007.
Franses, P.H., Van Griensven, K., 1998. Forecasting exchange rates using neural networks for technical trading rules. Stud. Nonlinear Dyn. Econometrics 2, 109–114.
Franses, P.H., Van Homelen, P., 1998. On forecasting exchange rates using neural networks. Appl. Financ. Econ. 8, 589–596.
Fischer, T., Krauss, C., Treichel, A., 2018. Machine Learning for Time Series Forecasting – a Simulation Study, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Working Paper No.02/2018.
Gospodinov, N., Jamali, I., 2011. Risk premiums and predictive ability of BAX futures. J. Futures Mark. 31, 534–561.
Harvey, D., Leybourne, S., Newbold, P., 1997. Testing the equality of prediction mean squared errors. Int. J. Forecast. 13, 281–291.
Haykin, S., 1994. Neural Networks, a Comprehensive Foundation (No. BOOK). Macmilan.
Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2, 359–366.
Hornik, K., 1993. Some new results on neural network approximation. Neural Netw. 6, 1069–1072.
Kara, Y., Boyacioglu, M.A., Baykan, Ö.K., 2011. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. Expert Syst. Appl. 38, 5311–5319.
Kohzadi, N., Boyd, M.S., Kermanshahi, B., Kaastra, I., 1996. A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing 10, 169–181.
Krenker, A., Bester, J., Kos, A., 2011. Introduction to the artificial neural networks. Artificial Neural Networks-Methodological Advances and Biomedical Applications. InTech.
Kumar, M., Thenmozhi, M., 2016. Forecasting stock index movement: a comparison of support vector machines and random forest. Indian Institute of Capital Markets 9th Capital Markets Conference Paper.
Le, H.H., Viviani, J.L., 2018. Predicting bank failure: an improvement by implementing a machine-learning approach to classical financial ratios. Res. Int. Bus. Finance 44, 16–25.
Maasoumi, E., Khotanzed, A., Abaye, A., 1994. Artificial neural networks for some macroeconomic series: a first report. Econometric Rev. 13, 105–122.
McNelis, P.D., 2005. Neural Networks in Finance: Gaining Predictive Edge in the Market, 2005. Elsevier Academic Press, Burlington, USA.
Oshiro, T.M., Perez, P.S., Baranauskas, J.A., 2012. How many trees in a random forest? July. International Workshop on Machine Learning and Data Mining in Pattern

[4] The Harvey et al. (1997) of the Diebold and Mariano (1995) test confirms that the differences in RMSE and MAE are significant only for the dynamic forecasts of long-horizon returns. These results are available from the authors upon request.

Recognition. Springer, Berlin, Heidelberg, pp. 154–168.
Reed, R., Marks, R.J., 1999. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press.
Refenes, A.N., Zapranis, A., Francis, G., 1994. Stock performance modeling using neural networks: a comparative study with regression models. Neural Netw. 7, 375–388.
Rojas, R., 2013. Neural Networks: A Systematic Introduction. Springer Science & Business Media.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.
Swanson, N.R., White, H., 1995. A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. J. Bus. Econ. Stat. 13, 265–275.
Swanson, N.R., White, H., 1997. A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. Rev. Econ. Stat. 79, 540–550.
Teräsvirta, T., Van Dijk, D., Medeiros, M.C., 2005. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: a re-examination. Int. J. Forecast. 21, 755–774.
Vortelinos, D.I., 2014. Optimally sampled realized range-based volatility estimators. Res. Int. Bus. Finance 30, 34–50.
Vortelinos, D.I., 2017. Forecasting realized volatility: HAR against principal components combining, neural networks and GARCH. Res. Int. Bus. Finance 39, 824–839.
Wanke, P., Azad, M.A.K., Barros, C.P., 2016. Predicting efficiency in Malaysian Islamic banks: a two-stage TOPSIS and neural networks approach. Res. Int. Bus. Finance 36, 485–498.
White, H., 1988. Economic prediction using neural networks: the case of IBM daily stock returns. Proceedings of the IEEE International Conference on Neural Networks 2. pp. 451–458.
Yu, L., Wang, S., Lai, K.K., 2010. Foreign Exchange Rate Forecasting With Artificial Neural Networks. Springer Science & Business Media.