Boosting multi-step autoregressive forecasts

Souhaib Ben Taieb SBENTAIE@ULB.AC.BE
Machine Learning Group, Computer Science Department, Faculty of Sciences,
Université Libre de Bruxelles, Brussels, Belgium.

Rob J Hyndman ROB.HYNDMAN@MONASH.EDU
Department of Econometrics and Business Statistics,
Monash University, Clayton VIC 3800, Australia.

Abstract

Multi-step forecasts can be produced recursively by iterating a one-step model, or directly using a specific model for each horizon. Choosing between these two strategies is not an easy task since it involves a trade-off between bias and estimation variance over the forecast horizon. Using a nonlinear machine learning model makes the tradeoff even more difficult. To address this issue, we propose a new forecasting strategy which boosts traditional recursive linear forecasts with a direct strategy using a boosting autoregression procedure at each horizon. First, we investigate the performance of the proposed strategy in terms of bias and variance decomposition of the error using simulated time series. Then, we evaluate the proposed strategy on real-world time series from two forecasting competitions. Overall, we obtain excellent performance with respect to the standard forecasting strategies.

1. Introduction

Forecasts guide decisions in many areas of scientific, industrial and economic activity such as in meteorology, telecommunication and finance. In many real-life scenarios, the forecaster encounters a multi-step forecasting problem where forecasts are required for short, medium or long horizons. In particular, multi-step forecasting of a univariate time series consists in predicting several future observations of a given sequence of historical observations.

Although time series from real world phenomena typically behave nonlinearly (Kantz & Schreiber, 2004), time series forecasting is very much dominated by linear methods (Gooijer & R. J. Hyndman, 2006). This is partly due to mathematical convenience and the robust performance of linear forecasting methods (Fan & Yao, 2005). Nonlinear models have also been considered to allow more flexibility in the forecasts. Examples include bilinear models, k-nearest-neighbor methods and neural network models (Teräsvirta, Tjø stheim, & Granger, 2010). However, empirical studies indicate that the potential gain from complex nonlinear methods is not always realized due to overfitting and estimation variance. In addition, real-world time series often possess either approximately linear, or moderately nonlinear behavior that does not require a complex nonlinear model to be captured.

Traditionally, multi-step forecasting has been handled recursively, where a single time series model is estimated and each forecast is computed using previous forecasts. Another approach builds a separate time series model for each forecasting horizon, and forecasts are computed directly by the estimated model (Chevillon, 2007). Choosing between these two strategies involves a trade-off between bias and estimation variance and depends also on the size of the time series, the level of noise and the nonlinearity of the model. In brief, selecting one of the two strategies is not an easy task in real-world forecasting problems.

We propose a new method, called the boost strategy, to address the two previous issues, i.e. the weak nonlinearity in many real-world time series and the problem of choosing between recursive and direct forecasts. The idea is to boost recursive linear forecasts with a direct strategy using several *small* and *nonlinear* adjustments at each forecast horizon. To do so, a boosting autoregression procedure is applied at each horizon on the residuals from the recursive linear forecasts using a so-called weak learner; that is a learner with large bias relative to variance.

Our boost strategy has many advantages: (i) it balances flexibility and robustness since the linear recursive forecasts serve as a first robust approximation and flexibility is

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

added by allowing several nonlinear boosting components at each forecast horizon; (ii) it allows nonlinearities with the boosting components without sacrificing much variance thanks to the reduced variance of the weak learners; and (iii) it avoids the difficult choice between recursive and direct forecasts.

We evaluate the boost strategy in two steps. In the first step, we decompose the mean squared error (MSE) of the forecasts and analyze the bias and variance components over the horizon. We begin by a theoretical analysis of the bias and variance components for two steps ahead. Then we conduct a simulation study with two data generating processes (DGP) for the general case of h steps ahead. Then, we consider real-world time series and compare the performance of the boost strategy with the recursive and direct strategies on roughly 500 time series from the M3 and NN5 forecasting competitions. Overall, the boost strategy consistently produces better out-of-sample forecasts and thus is very attractive for multi-step forecasting tasks.

2. Multi-step forecasting strategies

We begin by discussing the problem of multi-step forecasting and describe the recursive and the direct strategies for producing multi-step forecasts.

Consider a univariate time series $\boldsymbol{Y}_T = \{y_1, \dots, y_T\}$ comprising T observations. We would like to forecast the H future observations $\{y_{T+1}, \dots, y_{T+H}\}$. We assume that the data are described by a possibly non-linear autoregressive process of the form

$$y_t = f(\boldsymbol{x}_{t-1}) + \varepsilon_t \text{ with } \boldsymbol{x}_{t-1} = [y_{t-1}, \dots, y_{t-d}]', \quad (1)$$

where $\{\varepsilon_t\}$ is a Gaussian white noise process with zero mean and variance σ^2. The time series is therefore specified by a function f, an embedding dimension d, and a noise term ε_t. We assume that we do not know f or d.

If we consider the MSE as the error measure to be minimized, then the optimal forecast at horizon h is the conditional mean $\mu_{t+h|t} = \mathbb{E}[y_{t+h} \mid \boldsymbol{x}_t]$ and the goal of forecasting is to estimate it.

The recursive strategy. One strategy for producing multi-step forecasts, called *recursive*, centers on building a time series model of the same form as (1), aiming to minimize the one-step-ahead prediction error variance. The unknown future values are then obtained dynamically by repeatedly iterating the model and by replacing (plugging in) the unknown future values with their own forecasts. In other words, it entails a model of the form

$$y_t = m(\boldsymbol{z}_{t-1}; \boldsymbol{\phi}) + e_t \quad (2)$$

with $\boldsymbol{z}_{t-1} = [y_{t-1}, \dots, y_{t-p}]'$ where p is an estimation of the embedding dimension d and $\mathbb{E}[e_t] = 0$. Note that $e_t = f(\boldsymbol{x}_{t-1}) - m(\boldsymbol{z}_{t-1}; \boldsymbol{\phi}) + \varepsilon_t$ is the forecast error of the model m. The parameters $\boldsymbol{\phi}$ are estimated by

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \sum_t \left[y_t - m(\boldsymbol{z}_{t-1}; \boldsymbol{\phi})\right]^2.$$

Then, forecasts are obtained recursively, $\hat{\mu}_{T+h|T} = m^{(h)}(\boldsymbol{z}_T; \hat{\boldsymbol{\phi}})$ where $m^{(h)}$ is the recursive application of m and $h = 1, \dots, H$.

The direct strategy. A second strategy, called *direct*, tailors the forecasting model directly to the forecast horizon. That is, different forecasting models are used for each forecast horizon:

$$y_t = m_h(\boldsymbol{r}_{t-h}; \boldsymbol{\gamma}_h) + e_{t,h}, \quad (3)$$

where $\boldsymbol{r}_{t-h} = [y_{t-h}, \dots, y_{t-h-p_h}]'$. For each model, the parameters $\boldsymbol{\gamma}_h$ are estimated as follows

$$\hat{\boldsymbol{\gamma}}_h = \underset{\boldsymbol{\gamma}_h}{\operatorname{argmin}} \sum_t \left[y_t - m_h(\boldsymbol{r}_{t-h}; \boldsymbol{\gamma}_h)\right]^2.$$

Then forecasts are obtained for each horizon from the corresponding model, $\hat{\mu}_{T+h|T} = m_h(\boldsymbol{r}_T; \hat{\boldsymbol{\gamma}}_h)$ with $h = 1, \dots, H$.

If we knew f and d, then the recursive strategy and the direct strategy would be equivalent when f is linear, but not when f is nonlinear. Because of minimizing the one-step prediction error, when f is nonlinear the recursive strategy is biased while the direct strategy achieves the optimal error in a mean squared error sense (Fan & Yao, 2005; Atiya, Elshoura, Shaheen, & El-sherif, 1999).

In practice, which strategy is better is an empirical matter as we must estimate unknown functions m (Eq. (2)) and m_h (Eq. (3)) from a finite-sample dataset. The performance of both strategies depends notably on the nonlinearity of f, the embedding dimension d, the level of noise σ^2, the size of the time series T, the estimation algorithm and the forecast horizon h. So, the choice between the recursive and the direct strategy is not an easy task in applications.

3. The boost strategy

We propose to boost the recursive forecasts from a simple autoregressive (AR) linear model by using a direct strategy, allowing several *small* and *nonlinear* adjustments at each horizon h. Each adjustment tries to catch the discrepancy between the linear recursive forecasts and the true conditional mean at horizon h.

In other words, we begin with a simple autoregressive linear model,

$$y_t = \underbrace{c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p}}_{m(\boldsymbol{z}_{t-1}; \boldsymbol{\phi})} + e_t$$

and produce forecasts from it using the recursive strategy $m^{(h)}(\boldsymbol{z}_t; \hat{\phi})$. At this stage, our forecasts are equivalent to

those from the traditional linear AR model. We consider these forecasts as a first approximation and we add a further direct step to adjust for data features which cannot or are not represented by the linear autoregressive fit.

More precisely, a boosting procedure is used to model the potential signal left in the residuals from the AR forecasts, at each horizon h, with a direct strategy. To ensure that the adjustments are *small*, every boosting procedure requires the choice of a so-called weak learner; that is a learner with large bias relative to variance. Formally, we have

$$y_t = m^{(h)}(\boldsymbol{z}_{t-h}; \hat{\phi}) + \sum_{i=1}^{I_h} \nu l_i(y_{t-j}, y_{t-k}; \boldsymbol{\psi}_i) + e_{t,h}, \quad (4)$$

where I_h is the number of boosting iterations at horizon h, νl_i is the weak component at iteration i with ν being a shrinkage factor, and $y_{t-j}, y_{t-k} \in \boldsymbol{r}_{t-h}$.

One should note that the weak components in (4) are assured to be *weak* because of three reasons: the shrinkage factor ν shrinks the estimate towards zero, the restriction to bivariate interactions between variables, and the weak learner $l(\cdot; \psi)$.

Several weak learners have been used in the boosting literature, notably stumps (trees with two terminal nodes) (Friedman, 2001) and smoothing splines (Bühlmann & Yu, 2003). Penalised regression splines (P-splines) (Eilers & Marx, 1996) have also been considered in Schmid and Hothorn (2008) as a better alternative to smoothing splines in terms of computational time. We use P-splines as the weak learner in our implementation of the boost strategy.

P-splines require the selection of two additional parameters: the number of knots and the smoothing parameter. However, Ruppert (2002) has shown that the number of knots does not have much effect on the estimation provided enough knots are used. The weakness of the P-spline is measured by its degrees of freedom (df). Bühlmann and Yu (2003) and Schmid and Hothorn (2008) proposed that the smoothing parameter be set to give a small value of df (i.e., df $\in [3, 4]$), and that this number be kept fixed in each boosting iteration.

Algorithm 1 gives the different steps of the boost strategy to estimate the conditional mean $\mu_{T+h|T}$ at each horizon with a gradient boosting approach (Friedman, 2001). We assume the number of boosting iterations at each horizon is given. In practice, one may use a cross-validation procedure to identify the best number of iterations. This hyperparameter is particularly important as it controls the trade-off between the bias and variance of the estimation (Bühlmann & Yu, 2003).

First, an AR model is fitted (line 1). Then at each horizon h, the first (possibly crude) estimate is simply the recursive

Algorithm 1 The boost strategy

$\{y_1, \dots, y_T\}$:	Time series with T observations.
H:	Forecasting horizon.
$l(\cdot; \boldsymbol{\psi})$:	Weak learner.
$[I_1, \dots, I_H]$:	Number of iterations for each horizon.
$0 < \nu \leq 1$:	Shrinkage parameter.

1: Fit an AR(p): $y_t = \underbrace{c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p}}_{m(\boldsymbol{z}_{t-1}; \boldsymbol{\phi})} + e_t$
2: **for** $h \leftarrow 1, \dots, H$ **do**
3: $\quad \hat{F}_h^{(0)}(\boldsymbol{r}_{t-h}) = m^{(h)}(\boldsymbol{z}_t; \hat{\phi})$
4: $\quad$ **for** $i \leftarrow 1, \dots, \boldsymbol{I}_h$ **do**
5: $\quad\quad \tilde{y}_t^i = -\frac{\frac{1}{2}\partial(y_t - F_h(\boldsymbol{r}_{t-h}))^2}{\partial F_h(\boldsymbol{r}_{t-h})}\Big|_{F_h(\boldsymbol{r}_{t-h}) = F_h^{(i-1)}(\boldsymbol{r}_{t-h})}$
6: $\quad\quad\quad = (y_t - \hat{F}_h^{(i-1)}(\boldsymbol{r}_{t-h}))$
7: $\quad\quad$ **for all** $y_{t-a}, y_{t-b} \in \boldsymbol{r}_{t-h}$ **do**
8: $\quad\quad\quad \{(y_{t-a}, y_{t-b}, \tilde{y}_t^i)\}_{t=1}^T \rightarrow l_i(y_{t-a}, y_{t-b}; \hat{\boldsymbol{\psi}}_i)$
9: $\quad\quad$ **end for**
10: $\quad\quad (j, k) = \underset{(a,b)}{\operatorname{argmin}} \sum_{t=1}^T [\tilde{y}_t^i - \nu l_i(y_{t-a}, y_{t-b}; \hat{\boldsymbol{\psi}}_i)]^2$
11: $\quad\quad \hat{F}_h^{(i)}(\boldsymbol{r}_{t-h}) = \hat{F}_h^{(i-1)}(\boldsymbol{r}_{t-h}) + \nu l_i(y_{t-j}, y_{t-k}; \hat{\boldsymbol{\psi}}_i)$
12: $\quad$ **end for**
13: $\quad \hat{\mu}_{T+h|T} = \hat{F}_h^{(I_h)}(\boldsymbol{r}_T)$
14: $\quad\quad\quad = m^{(h)}(\boldsymbol{z}_T; \hat{\boldsymbol{\phi}}) + \sum_{i=1}^{\boldsymbol{I}_h} \nu l_i(y_{T-j}, y_{T-k}; \hat{\boldsymbol{\psi}}_i)$
15: **end for**

forecasts from the AR(p) model (line 3). This first estimate is boosted during I_h iterations to improve the forecasts. More precisely, at each iteration i, a new pseudo-response is calculated. In the case of quadratic loss, it is simply the residuals from the previous iteration (line 6). This pseudo-response is then regressed against all possible pairs of variables with the weak learner (line 8) and the estimate with the largest contribution to the fit is selected (line 10) and added to the estimation of the previous iteration (line 11). The final forecasts is then the sum of the AR(p) forecasts and several boosting components.

In brief, we enrich the class of linear autoregressive model with additional nonlinear terms and bivariate interactions at each forecasting horizon. By using a boosting procedure to extend the linear autoregressive model to a broader class of models, we allow the modelling of more complex dependencies without sacrificing much variance. Also, limiting the possible interactions is motivated by the fact that we expect many real-world time series to depend on lower-order interactions (Friedman, 2001; Duvenaud, Nickisch, & Rasmussen, 2011). Finally, the boost strategy is very attractive as it avoids making a choice between the recursive and the direct strategies which can be a difficult task in real-world

applications.

4. Related work

This work considers boosting in the contexts of multi-step forecasting. In the machine learning literature, boosting is well known for classification with AdaBoost (Freund & Schapire, 1996), but much less attention has been paid to regression settings. Some extensions of AdaBoost to regression include Drucker (1997) and Shrestha and Solomatine (2006). A gradient boosting approach has also been proposed for regression (Friedman, 2001).

In the forecasting community, boosting has received even less attention and the literature is rather sparse. Assaad, Boné, and Cardot (2008) considered recurrent neural networks as weak learners with an adapted AdaBoost and compared their method with local approaches on two time series. Audrino and Bühlmann (2003, 2009) used a gradient boosting approach to model volatility in financial applications. Boosting has only recently been considered in the macroeconometric literature with (Shafik & Tutz, 2009; Bai & Ng, 2009; Buchen & Wohlrabe, 2011). Economic forecasting is also considered in Robinzonov, Tutz, and Hothorn (2012) with a boosting procedure to estimate nonlinear additive autoregressive models. Finally, a gradient boosting approach has been used recently in a load forecasting competition and ranked among the top five competitors (Ben Taieb & R. Hyndman, 2013).

5. Bias and variance analysis

A performance analysis of the forecasting strategies can be accomplished through an examination of the error decomposition into the bias and variance components (Geman, Bienenstock, & Doursat, 1992). Let $g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)$ denote the forecasts of a given strategy at horizon h using the input vector $\boldsymbol{z}_t$ and using the set of parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}$. These parameters are estimated using $\boldsymbol{Y}_T$, a time series with T observations. So, the input vector $\boldsymbol{z}_t$ and the set of parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}$ can change for each sample $\boldsymbol{Y}_T$. In addition, the input vector $\boldsymbol{z}_t$ can be different from $\boldsymbol{x}_t$, the "real" input vector defined in (1). Let us also define $g(\boldsymbol{z}_t; \boldsymbol{\theta}_T; h) = \mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)\right]$.

Assuming the process defined in (1) is stationary, the MSE of the given strategy at horizon h is decomposed as follows.

$$
\begin{aligned}
&\text{MSE}_h \\
&= \mathbb{E}_{\boldsymbol{x}_t}\left[\underbrace{\mathbb{E}_{\varepsilon,\boldsymbol{Y}_T}\left[(y_{t+h} - g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h))^2 \mid \boldsymbol{x}_t\right]}_{\text{MSE}_h(\boldsymbol{x}_t)}\right] \\
&= \underbrace{\mathbb{E}_{\boldsymbol{x}_t,\varepsilon}\left[(y_{t+h} - \mu_{t+h|t})^2 \mid \boldsymbol{x}_t\right]}_{\text{Noise } N_h} \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{x}_t}\left[(\mu_{t+h|t} - g(\boldsymbol{z}_t; \boldsymbol{\theta}_T; h))^2\right]}_{\text{Bias } B_h} \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{x}_t,\boldsymbol{Y}_T}\left[(g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h) - g(\boldsymbol{z}_t; \boldsymbol{\theta}_T; h))^2 \mid \boldsymbol{x}_t\right]}_{\text{Variance } V_h}
\end{aligned} \tag{5}
$$

where $\mathbb{E}_x$ and $\mathbb{E}[\cdot|x]$ denote the expectation over x and the expectation conditional on x, respectively.

We can see that the MSE of the forecasts $g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)$ at horizon h can be decomposed into three different components, namely the noise term N_h, the squared bias term B_h and the estimation variance term V_h. So, this decomposition is identical to the usual decomposition used in machine learning (Geman et al., 1992). However, in contrast with usual regression problems, multi-step forecasting is dealing with time-dependent data and requires learning dependent tasks with different noise level changing with the forecasting horizon h.

At horizon $h = 1$, the problem of multi-step forecasting reduces for all strategies to the estimation of the function f since we have the simple expression $\mu_{t+1|t} = f(\boldsymbol{x}_t)$. In the following, we consider other horizons. To simplify the derivations, we will perform a theoretical analysis for two steps ahead using similar arguments to Ben Taieb and Atiya (2014). Then, we will perform Monte Carlo simulations to analyze bias and variance for the general case of h steps ahead.

5.1. Theoretical analysis

Assume that the time series is generated by the nonlinear autoregressive process defined in (1). First, we can compute y_{t+2} using a Taylor series approximation up to second-order terms, which gives us

$$
\begin{aligned}
&y_{t+2} \\
&\quad = f(f(\boldsymbol{x}_t) + \varepsilon_{t+1}, y_t, \ldots, y_{t-d+2}) + \varepsilon_{t+2} \\
&\quad \approx f(f(\boldsymbol{x}_t), \ldots, y_{t-d+2}) + \varepsilon_{t+1} f_{x_1} + \tfrac{1}{2}(\varepsilon_{t+1})^2 f_{x_1 x_1} + \varepsilon_{t+2},
\end{aligned}
$$

where f_{x_1} and $f_{x_1 x_1}$ are the first and second derivatives of f with respect to its first argument, respectively. The conditional expectation $\mu_{t+2|t}$ is then given by

$$
\begin{aligned}
\mu_{t+2|t} &= \mathbb{E}[y_{t+2}|\boldsymbol{x}_t] \\
&= f(f(\boldsymbol{x}_t), y_t, \ldots, y_{t-d+2}) + \tfrac{1}{2}\sigma^2 f_{x_1 x_1}
\end{aligned}
$$

In order to compute the bias and variance terms at $h = 2$, $B_2(\boldsymbol{x}_t)$ and $V_2(\boldsymbol{x}_t)$ as defined in (5), we consider that the forecasts of each strategy can be modeled as a sum of three terms: the true function value we are trying to estimate, that is the conditional mean $\mu_{t+2|t} = \mathbb{E}[y_{t+2}|\boldsymbol{x}_t]$, an offset term denoted by $\delta(\boldsymbol{z}_t; \boldsymbol{\theta})$ and a variability term denoted by $\eta(\boldsymbol{z}_t; \boldsymbol{\theta})\varepsilon_\eta$ where $\eta(\boldsymbol{z}_t; \boldsymbol{\theta})$ is a deterministic component giving the standard deviation of the term, and ε_η is a stochastic component with $\mathbb{E}[\varepsilon_\eta] = 0$ and $\mathbb{E}[\varepsilon_\eta^2] = 1$.

The offset term $\delta(\boldsymbol{z}_t;\boldsymbol{\theta})$ is the discrepancy from the conditional mean $\mu_{t+2|t}$ arising from (i) the lack of flexibility of the considered forecasting model (i.e. the model, with its parameters $\boldsymbol{\theta}$, is not powerful enough to reconstruct $\mu_{t+2|t}$ accurately), (ii) potential missing variables in the inputs ($\boldsymbol{z}_t$ not equal to $\boldsymbol{x}_t$), and (iii) an inadequate estimation algorithm for the parameters $\boldsymbol{\theta}$ (i.e. even if the model is powerful, the training algorithm may fall short of finding the right parameters).

The variability term $\eta(\boldsymbol{z}_t;\boldsymbol{\theta})\varepsilon_\eta$ represents the variability of the forecasts, and it arises due to (i) the finite-sampledness of the time series $\boldsymbol{Y}_T = \{y_1,\dots,y_T\}$ used to estimate $\boldsymbol{\theta}$, (ii) the number of input variables in $\boldsymbol{z}_t$ potentially including redundant or meaningless variables, and (iii) the complexity of the model that may make it too flexible.

We now write the forecasts of the different strategies using the previous terminology. To simplify notation, we will remove the dependence on the size of the time series T.

Forecasts of the recursive strategy. To produce forecasts at horizon $h=2$, the recursive strategy first estimate a one-step model as in (2) and produce forecasts for $h=1$, that is

$$g(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}};1) = \underbrace{\underbrace{\underbrace{f(\boldsymbol{x}_t)}_{\mu_{t+1|t}} + \delta(\boldsymbol{z}_t;\boldsymbol{\phi})}_{m(\boldsymbol{z}_t;\boldsymbol{\phi})} + \eta(\boldsymbol{z}_t;\boldsymbol{\phi})\varepsilon_\eta}_{m(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}})}$$

Then, forecasts at horizon $h=2$ are obtained recursively and can be computed, after some simplification using a Taylor series expansion, as follows

$$\begin{aligned}
&g(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}};2)\\
&= m(m(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}}),\dots,y_{t-p+2};\hat{\boldsymbol{\phi}})\\
&= f(f(\boldsymbol{x}_t),\dots,y_{t-p+2})\\
&\quad + \delta(f(\boldsymbol{x}_t),\dots,y_{t-p+2};\boldsymbol{\phi}) + \eta(f(\boldsymbol{x}_t),\dots,y_{t-p+2};\boldsymbol{\phi})\varepsilon_{\eta_2}\\
&\quad + \delta(\boldsymbol{z}_t;\boldsymbol{\phi})m_{z_1} + \frac{1}{2}[\delta(\boldsymbol{z}_t;\boldsymbol{\phi})]^2 m_{z_1z_1}\\
&\quad + \eta(\boldsymbol{z}_t;\boldsymbol{\phi})\varepsilon_{\eta_1}m_{z_1} + \frac{1}{2}[\eta(\boldsymbol{z}_t;\boldsymbol{\phi})\varepsilon_{\eta_1}]^2 m_{z_1z_1}
\end{aligned}$$

where ε_{η_1} and ε_{η_2} are the stochastic components of the variability term around points $\boldsymbol{z}_t$ and $[f(\boldsymbol{z}_t),\dots,y_{t-p+2}]$ respectively, and m_{z_1} and $m_{z_1z_1}$ are respectively the first and second derivatives of the model m with respect to its first argument.

Forecasts of the direct strategy. A model is estimated to directly produce forecast for horizon $h=2$ and can be written as

$$g(\boldsymbol{r}_t;\hat{\boldsymbol{\gamma}};2) = \underbrace{\underbrace{\mu_{t+2|t} + \delta(\boldsymbol{r}_t;\boldsymbol{\gamma})}_{m_2(\boldsymbol{r}_t;\boldsymbol{\gamma})} + \eta(\boldsymbol{r}_t;\boldsymbol{\gamma})\varepsilon_\eta}_{m_2(\boldsymbol{r}_t;\hat{\boldsymbol{\gamma}})}$$

In contrast with the forecasts of the recursive strategy, we can see that the conditional mean $\mu_{t+2|t}$ appears in the previous expression.

Forecasts of the boost strategy. A first linear AR(p) model is fitted and can be written as

$$m(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}}) = \underbrace{\underbrace{f(\boldsymbol{x}_t)}_{\mu_{t+1|t}} + \delta(\boldsymbol{z}_t;\boldsymbol{\phi})}_{c+\phi_1 y_{t-1}+\cdots+\phi_p y_{t-p}} + \eta(\boldsymbol{z}_t;\boldsymbol{\phi})\varepsilon_\eta$$

Then, the forecasts at horizon $h=2$ are produced with

$$\begin{aligned}
&m(\underbrace{m(\boldsymbol{z}_t;\hat{\boldsymbol{\phi}}),\dots,y_{t-p+2}}_{\hat{\boldsymbol{z}}_{t+1}};\hat{\boldsymbol{\phi}})\\
&= (c+\phi_1 c) + (\phi_1^2+\phi_2)y_t + (\phi_1\phi_2+\phi_3)y_{t-1}\\
&\quad + \cdots + (\phi_1\phi_{p-1}+\phi_p)y_{t-p+2} + (\phi_1\phi_p)y_{t-p+1}\\
&\quad + \underbrace{\phi_1\eta(\boldsymbol{z}_t;\boldsymbol{\phi})\varepsilon_{\eta_1} + \eta(\hat{\boldsymbol{z}}_{t+1};\boldsymbol{\phi})\varepsilon_{\eta_2}}_{(\phi_1+1)\eta(\boldsymbol{\phi})\varepsilon_\eta}
\end{aligned}$$

where the variability terms have been simplified since in the linear model, the variability is independent on the input and only depends on the set of parameters $\boldsymbol{\phi}$.

After adding the adjustments from the boosting procedure, the forecasts of the boost strategy can be written as

$$\begin{aligned}
&g(\boldsymbol{r}_t;[\hat{\boldsymbol{\phi}};\hat{\boldsymbol{\psi}}];2)\\
&= (c+\phi_1 c) + (\phi_1^2+\phi_2)y_t + (\phi_1\phi_2+\phi_3)y_{t-1}\\
&\quad + \cdots + (\phi_1\phi_{p-1}+\phi_p)y_{t-p+2} + (\phi_1\phi_p)y_{t-p+1}\\
&\quad + (\phi_1+1)\eta(\boldsymbol{\phi})\varepsilon_{\eta_0}\\
&\quad + \underbrace{\sum_{i=1}^{I_2}\nu[\underbrace{l_i(y_{t-j},y_{t-k};\boldsymbol{\psi}_i) + \eta_i(y_{t-j},y_{t-k};\boldsymbol{\psi}_i)*\varepsilon_{\eta_i}}_{l_i(y_{t-j},y_{t-k};\hat{\boldsymbol{\psi}}_i)}]}_{\sum_{jk} M_{jk}(y_{t-j},y_{t-k};\boldsymbol{\psi}_{jk})+\eta_{jk}(y_{t-j},y_{t-k};\boldsymbol{\psi}_{jk})*\varepsilon_{\eta_{jk}}}
\end{aligned}$$

where $\hat{\boldsymbol{\phi}} = [\hat{\phi}_1,\dots,\hat{\phi}_p]$, $\hat{\boldsymbol{\psi}} = [\hat{\boldsymbol{\psi}}_1,\dots,\hat{\boldsymbol{\psi}}_I]$, and where we have used the fact that the sum of the boosting components over the iterations can also be written as a sum over the different interactions, that is

$$\begin{aligned}
&\sum_{i=1}^{I_2}\nu l_i(y_{t-j},y_{t-k};\hat{\boldsymbol{\psi}}_i)\\
&= \sum_{jk}\sum_{S_{jk}=\{i|a=j\wedge b=k\}}\nu l_i(y_{t-a},y_{t-b};\hat{\boldsymbol{\psi}}_i)\\
&= \sum_{jk} M_{jk}(y_{t-j},y_{t-k};\hat{\boldsymbol{\psi}}_{jk}),
\end{aligned}$$

where $\hat{\boldsymbol{\psi}}_{jk} = \{\hat{\boldsymbol{\psi}}_i | i \in S_{jk}\}$.

Bias and variance comparison. Recall that we want to estimate $\mu_{t+2|t} = f(f(\boldsymbol{x}_t),\dots,y_{t-d+2}) + \frac{1}{2}\sigma^2 f_{x_1x_1}$. Let us now calculate the sum of the bias and variance components as defined in (5), for horizon $h=2$ using the previous expressions for the forecasts of the three strategies. In other words, we compute

$$
\begin{aligned}
&B_2(\boldsymbol{x}_t) + V_2(\boldsymbol{x}_t) \\
&= (\mu_{t+2|t} - g(\boldsymbol{z}_t; \boldsymbol{\theta}; 2))^2 \\
&\quad + \mathbb{E}_{\boldsymbol{Y}_T}\left[(g(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}; 2) - g(\boldsymbol{z}_t; \boldsymbol{\theta}; 2))^2 \mid \boldsymbol{x}_t\right]
\end{aligned}
$$

For the recursive strategy, we have

$$
\begin{aligned}
&B_2^{\text{REC}}(\boldsymbol{x}_t) + V_2^{\text{REC}}(\boldsymbol{x}_t) \\
&= \Big[\mu_{t+2|t} - \Big(f(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}) && (6)\\
&\quad + \delta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\phi}) + \delta(\boldsymbol{z}_t; \boldsymbol{\phi}) m_{z_1} \\
&\quad + \frac{1}{2}[\delta(\boldsymbol{z}_t; \boldsymbol{\phi})]^2 m_{z_1 z_1} + \frac{1}{2}[\eta(\boldsymbol{z}_t; \boldsymbol{\phi})]^2 m_{z_1 z_1}\Big)\Big]^2 && (7)\\
&+ [\eta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\phi})]^2 + [\eta(\boldsymbol{z}_t; \boldsymbol{\phi}) m_{z_1}]^2 \\
&\quad + \frac{1}{2}[\eta(\boldsymbol{z}_t; \boldsymbol{\phi})]^4 m_{z_1 z_1}^2 && (8)\\
&\quad + 2\eta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\phi})\eta(\boldsymbol{z}_t; \boldsymbol{\phi}) m_{z_1} \mathbb{E}[\varepsilon_{\eta_1} \varepsilon_{\eta_2}] \\
&\quad + \eta(\boldsymbol{z}_t; \boldsymbol{\phi})^2 \eta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\phi}) m_{z_1 z_1} \mathbb{E}[\varepsilon_{\eta_1}^2 \varepsilon_{\eta_2}] && (9)
\end{aligned}
$$

where we used the fact that $\mathbb{E}[\varepsilon_\eta^3] = 0$ and $\mathbb{E}[\varepsilon_\eta^4] = 3$ for the standard normal distribution.

For the direct strategy, we have

$$
\begin{aligned}
&B_2^{\text{DIRECT}}(\boldsymbol{x}_t) + V_2^{\text{DIRECT}}(\boldsymbol{x}_t) \\
&= \left[\mu_{t+2|t} - m_2(\boldsymbol{r}_t; \boldsymbol{\gamma})\right]^2 && (10)\\
&\quad + \eta(\boldsymbol{r}_t; \boldsymbol{\gamma})^2 && (11)
\end{aligned}
$$

For the boost strategy, we have

$$
\begin{aligned}
&B_2^{\text{BOOST}}(\boldsymbol{x}_t) + V_2^{\text{BOOST}}(\boldsymbol{x}_t) \\
&= \big[\mu_{t+2|t} - \\
&\quad \big((c + \phi_1 c) + (\phi_1^2 + \phi_2) y_t + (\phi_1 \phi_2 + \phi_3) y_{t-1} \\
&\qquad + \cdots + (\phi_1 \phi_{p-1} + \phi_p) y_{t-p+2} + (\phi_1 \phi_p) y_{t-p+1} \\
&\qquad\quad + \sum_{jk} M_{jk}(y_{t-j}, y_{t-k}; \boldsymbol{\psi}_{jk})\Big)\Big]^2 && (12)\\
&\quad + (\phi_1 + 1)^2 \eta(\boldsymbol{\phi})^2 + \underbrace{\sum_{i=1}^{I_2} \frac{\eta_i(y_{t-j}, y_{t-k}; \boldsymbol{\psi}_i)^2}{\tau^2}}_{\sum_{jk} \eta_{jk}(y_{t-j}, y_{t-k}; \boldsymbol{\psi}_{jk})^2},
\end{aligned}
$$

where $\nu = \frac{1}{\tau}$ and we assumed $\varepsilon_{\eta_0} \perp \varepsilon_{\eta_{jk}}$ and $\varepsilon_{\eta_{ab}} \perp \varepsilon_{\eta_{jk}}$.

Let us now compare the boost strategy with the recursive and direct strategies, beginning with the bias component. For the recursive strategy, since the model m is used recursively, we can see in (6) and (7) that the offset $\delta(\cdot; \phi)$ at $h = 1$ is propagated to $h = 2$. In addition, the offset is amplified when the model m produces a function that has large variations (i.e. m_{z_1} and $m_{z_1 z_1}$ are large in magnitude). For the direct strategy, the offset of the model at $h = 1$ does not appear in (10). So provided that the model m_2 is flexible enough to estimate the conditional mean and enough data is available, the bias can be arbitrarily small. For the boost strategy, because we require the recursive AR model to be linear, the propagation of errors is limited since m_{z_1} is constant, $m_{z_1 z_1} = 0$, $\delta(\cdot, \phi) = \delta(\phi)$ and $\eta(\cdot, \phi) = \eta(\phi)$. Even if the linear recursive forecasts are biased at some horizon, the nonlinear boosting components can adjust the bias as can be see in (12).

We now turn to the variance components. For the recursive strategy, we can see in (8)–(9) that, similar to the offset in the bias, the variance terms get amplified. For the direct strategy, we can see in (11) that the variance will depend on the variability induced by the input $\boldsymbol{r}_t$, the set of parameters $\boldsymbol{\gamma}$ and the size of the time series T. This variability can be particularly large for complex nonlinear models which contain many interactions in $\boldsymbol{r}_t$ or have a large set of parameters $\boldsymbol{\gamma}$. For the boost strategy, the variance is limited, on the one hand by the recursive AR model being linear, and on the other hand because the boosting components allow only bivariate interactions and are shrunk towards zero with the shrinkage factor ν. Limiting interactions to two is not a strong limitation since we expect real-world time series to depend on lower-order interactions.

Furthermore, considering the fact that the direct strategy selects the model at each horizon independently, the errors $e_{t,h}$ from the different models in (3) can be autocorrelated; that is information is left in the errors. With the boost strategy, a direct approach is used after extracting the recursive linear forecasts from the observations. By doing so, selecting the direct models independently has a smaller effect compared to a pure direct strategy.

Finally, if we consider the case of an infinitely long time series, and when f is nonlinear, the direct strategy dominates the recursive strategy which is biased (see p. 348 of Teräsvirta et al. (2010)). When f is linear, the direct and the recursive strategy are equivalent (see p. 118 of Fan and Yao (2005)). In the same case, the boost strategy is equivalent to the direct strategy if the maximum order of interaction in the function f is two. If it is more than two, the boost strategy will be biased.

5.2. Analysis by Monte Carlo simulations

We conduct a simulation study to shed some light on the performance of the boost strategy in terms of bias and variance components over the forecasting horizon. The methodology is similar to the one performed in Berardi and Zhang (2003) except that we consider forecasting multi-step ahead instead of one-step ahead.

Data generating processes. We consider a nonlinear and a linear AR process in the simulation study (see Appendix A.1 of the supplementary material). The nonlinear process has been considered in (Medeiros, Teräsvirta, & Rech,

2006) and (Kock & Teräsvirta, 2011) to compare different forecasting methods in a nonlinear setting. Also, considering a linear process allows us to evaluate the costs of extending the hypothesis space beyond linear functions for the different strategies when the true DGP is indeed linear.

Experimental setup. To show the importance of the size of the time series T for each strategy, we will compare different sizes, namely $T \in \{50, 100, 400\}$.

For the implementation of the recursive and direct strategies, we use a single hidden-layer feedforward neural network model.

Finally, we select the different model hyperparameters using a time-series cross-validation procedure (also called rolling origin). See Appendices A.1 and A.3 of the supplementary material for more details.

Bias and variance estimation. Expression (5) gives the decomposition of the MSE for a given strategy at horizon h. The different parts of the decomposition, namely the noise, the squared bias and the variance, can be estimated by replacing expectation with averages over a large number of samples.

For each DGP, we generate 1000 independent time series. To measure the bias and variance, we use an independent time series composed of 2000 observations from the same DGP for testing purposes. For each simulated time series, the first three hundred simulated values are discarded to stabilize the time series.

For the linear process, the conditional mean has been calculated analytically. For the nonlinear process, we used simulations to generate a large set of possible future continuations using different random terms. An average of these continuations gives us an estimation of the conditional mean.

Results. Figure 1 gives for the nonlinear DGP and different values of T, the MSE (first column), the squared bias (second column), the variance (third column) and the squared bias plus variance (fourth column). The same information is given in Figure 2 for the linear DGP. In the first column, corresponding to the MSE, the bias and variance components of the different strategies are hidden by substantial noise, making comparisons between strategies difficult. Consequently, we consider the three other columns to compare the strategies and use the MSE as a measure of the predictability of the time series relative to the mean (the red line).

First of all, we can see that for both DGPs, the variance components (third column) are much larger than the bias components (second column). In fact, the variance is roughly 10 times larger than the bias for $T = 50$ and $T = 100$. The relative importance of variance is decreasing for $T = 400$ but is still roughly three times larger than the bias.

Figure 1 shows that, for the nonlinear DGP, the boost strategy has reduced the variance consistently over the forecasting horizon compared to the recursive and the direct strategy. This decrease in variance has not induced a huge increase in bias for $T = 50$ and $T = 100$, thus making the forecasts of the boost strategy better. For $T = 400$, we see an increase in bias for the boost strategy but since the variance is still more important, the boost strategy has similar performance than the recursive strategy and still better performance than the direct strategy.

Figure 2 shows that, for the linear DGP, the direct strategy suffers from a high variance compared to the recursive strategy. The boost strategy has a similar variance to the recursive strategy, which is better than the direct strategy.

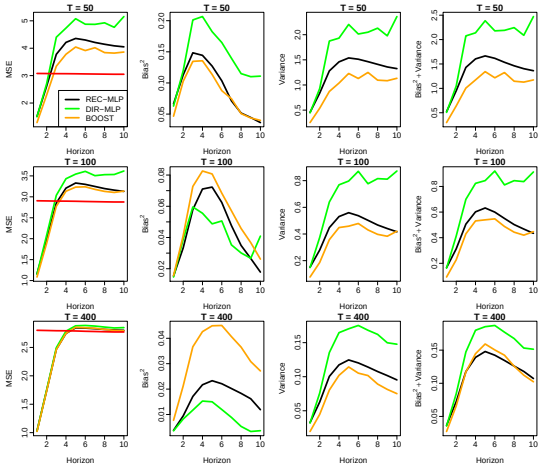

Figure 1. MSE decomposition for the nonlinear DGP with $H = 10$. MSE (first column), squared bias (second column), variance (third column) and squared bias + variance (fourth column).

6. Real-world experiments

We now evaluate our boost strategy on time series from the M3 (Makridakis & Hibon, 2000) and the NN5 forecasting competitions[1].

Real-world time series. The M3 competition dataset consists of 3003 monthly, quarterly, and annual time series. We have considered $M = 339$ monthly time series with a range of lengths between $T = 117$ and $T = 126$ with $H = 18$.

The NN5 competition dataset comprises $M = 111$ daily time series with $T = 735$ observations and $H = 56$ days (8 weeks).

[1]Forecasting competition for artificial neural networks and computational intelligence, 2008, www.neural-forecasting-competition.com/NN5/.

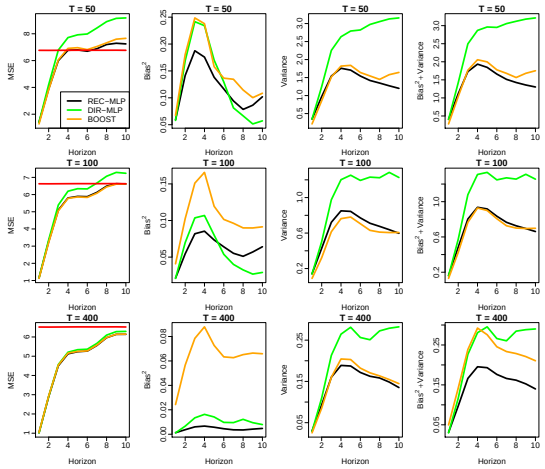

Figure 2. MSE decomposition for the linear DGP with $H = 10$. MSE (first column), squared bias (second column), variance (third column) and squared bias + variance (fourth column).

Preprocessing and setup. For both competitions, we deseasonalized the time series using STL (Seasonal-Trend decomposition based on Loess smoothing) (Cleveland, Cleveland, McRae, & Terpenning, 1990). Of course, the seasonality has been restored after forecasting. For the M3 competition, some time series have a trend. We used the KPSS test to check if first differencing is required.

For both competitions, we used the symmetric mean absolute percentage error (sMAPE), averaged across the M time series. The sMAPE was the main measure used in the M3 and the NN5 competitions. Note that we didn't use MSE since time series have different scales. See Appendices A.2 and A.3 of the supplementary material for more details.

Results. Table 1 gives the results for the M3 competition where the sMAPE is given together with the ranking calculated with respect to sMAPE (in brackets) for each strategy at each horizon. The last row gives the sMAPE averaged over all horizons. Numbers in bold represent the lowest error and underlined numbers represent the highest rank. The same information is given in Table 2 for the NN5 competition where each row corresponds to the sMAPE averaged over 7 consecutive days.

First of all, we can see that the recursive strategy has better forecasts than the direct strategy in the M3 competition (Table 1) and vice versa for the NN5 competition (Table 2). One explanation can be that the M3 time series are short ($T \leq 126$) and very noisy while the NN5 time series are longer ($T = 735$) and less noisy.

For both competitions, we can see that the boost strategy consistently gets better results both in terms of sMAPE and rankings. This confirms the advantage of the boost strategy which avoids the difficult task of choosing between recursive and direct forecasts and is able to produce forecasts with better or close performance to the best between the recursive and direct strategies.

Table 1. Forecast accuracy measures for the M3 competition. **Bold** : lowest error; Underlined : highest rank.

Horizon	REC-MLP	DIR-MLP	BOOST
1	7.41±0.83 (2.02)	**7.30±0.82 (2.01)**	7.47±0.85 (1.98)
2	**7.98±1.05 (1.99)**	7.99±1.00 (1.99)	8.09±1.03 (2.02)
3	8.30±0.89 (1.94)	8.30±0.74 (2.09)	**8.14±0.87 (1.97)**
4	9.42±1.00 (1.96)	9.37±1.01 (2.05)	**9.20±0.97 (1.98)**
5	10.65±1.14 (1.94)	11.30±1.26 (2.08)	**10.56±1.15 (1.98)**
6	11.92±1.18 (2.01)	12.79±1.28 (2.07)	**11.52±1.18 (1.91)**
7	11.93±1.18 (1.95)	12.52±1.17 (2.09)	**11.64±1.16 (1.96)**
8	12.58±1.28 (1.92)	13.72±1.28 (2.09)	**12.58±1.28 (1.99)**
9	12.54±1.19 (2.00)	13.57±1.24 (2.03)	**12.14±1.15 (1.97)**
10	11.51±1.08 (2.06)	12.34±1.13 (2.03)	**11.33±1.12 (1.91)**
11	**11.49±1.02 (2.01)**	12.74±1.10 (2.04)	11.58±1.00 (1.96)
12	12.16±1.12 (1.97)	12.74±1.18 (2.07)	**11.87±1.13 (1.96)**
13	13.76±1.17 (2.00)	15.17±1.31 (2.09)	**13.73±1.20 (1.92)**
14	13.58±1.18 (1.99)	14.38±1.17 (2.07)	**13.45±1.17 (1.95)**
15	13.50±1.18 (1.96)	15.28±1.32 (2.13)	**13.33±1.15 (1.91)**
16	14.11±1.26 (2.00)	15.03±1.32 (2.11)	**13.55±1.24 (1.89)**
17	14.60±1.32 (1.98)	15.39±1.30 (2.11)	**13.73±1.22 (1.91)**
18	17.39±1.53 (2.05)	17.48±1.50 (2.05)	**16.68±1.47 (1.90)**
1-18	11.93±0.95 (1.95)	12.63±0.96 (2.14)	**11.70±0.95 (1.92)**

Table 2. Forecast accuracy measures for the NN5 competition. **Bold** : lowest error; Underlined : highest rank.

Horizon	REC-MLP	DIR-MLP	BOOST
1-7	17.63±0.73 (2.19)	17.01±0.76 (1.96)	**16.78±0.72 (1.85)**
8-14	19.46±0.90 (2.17)	18.91±0.84 (2.05)	**18.10±0.85 (1.78)**
15-21	26.96±1.53 (2.09)	26.16±1.32 (1.96)	**25.91±1.30 (1.94)**
22-28	20.78±1.31 (2.10)	19.76±1.11 (2.00)	**19.29±1.08 (1.90)**
29-35	32.41±1.33 (1.91)	32.03±1.35 (2.09)	**31.96±1.37 (2.00)**
36-42	25.95±1.41 (2.07)	24.81±1.27 (2.11)	**24.45±1.30 (1.82)**
43-49	20.48±1.23 (2.14)	19.48±1.00 (1.95)	**19.29±1.06 (1.92)**
50-56	18.30±0.90 (2.04)	17.21±0.81 (2.14)	**16.69±0.83 (1.82)**
1-56	22.75±0.70 (2.22)	21.86±0.54 (2.05)	**21.51±0.57 (1.73)**

7. Conclusions

We have introduced a new strategy for producing multi-step forecasts of a univariate time series. The strategy starts with linear recursive forecasts from a traditional AR model and adjusts them with a direct strategy using a boosting procedure at each horizon.

The paper makes methodological, theoretical and empirical contributions to multi-step forecasting. In terms of methodology, we propose a new method for multi-step forecasting that combines the best features of the recursive and direct forecasting strategies. In terms of theory, we analyze the bias and variance components of the newly proposed method and compare the results with the recursive and direct strategies when the forecast horizon is two steps ahead. A simulation study is also performed to consider the general case of h steps ahead. In term of empirical usefulness, we evaluate our method on time series from two forecasting competitions and demonstrate that our method is very attractive for multi-step forecasting tasks.

Acknowledgments

We would like to thank Catharina Olsen for her comments and three reviewers for their constructive suggestions.

References

Assaad, M., Boné, R., & Cardot, H. (2008). A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion*, *9*(1), 41–55.

Atiya, A. F., El-shoura, S. M., Shaheen, S. I., & El-sherif, M. S. (1999). A comparison between neural-network forecasting techniques–case study: river flow forecasting. *IEEE Transactions on Neural Networks*, *10*(2), 402–409.

Audrino, F. & Bühlmann, P. (2003). Volatility estimation with functional gradient descent for very high-dimensional financial time series. *Journal of computational finance*, 1–23.

Audrino, F. & Bühlmann, P. (2009). Splines for financial volatility. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*(3), 655–670.

Bai, J. & Ng, S. (2009). Boosting diffusion indices. *Journal of Applied Econometrics*, *629*(March), 607–629.

Ben Taieb, S. & Atiya, A. F. (2014). *A bias and variance analysis for multi-step time series forecasting*. Manuscript submitted for publication.

Ben Taieb, S. & Hyndman, R. (2013). A gradient boosting approach to the Kaggle load forecasting competition. *International Journal of Forecasting*, 1–19.

Berardi, V. & Zhang, G. (2003). An empirical investigation of bias and variance in time series forecasting: modeling considerations and error evaluation. *Neural Networks, IEEE Transactions on*, *14*(3), 668–79.

Buchen, T. & Wohlrabe, K. (2011). Forecasting with many predictors: Is boosting a viable alternative? *Economics Letters*, *113*(1), 16–18.

Bühlmann, P. & Yu, B. (2003). Boosting With the L2 Loss: Regression and Classification. *Journal of the American Statistical Association*, *98*, 324–339.

Chevillon, G. (2007). Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, *21*(4), 746–785.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, *6*(1), 3–73.

Drucker, H. (1997). Improving regressors using boosting techniques. In *International conference on machine learning* (pp. 107–115).

Duvenaud, D., Nickisch, H., & Rasmussen, C. (2011). Additive Gaussian processes. In *Advances in neural information processing systems* (pp. 1–9).

Eilers, P. & Marx, B. (1996). Flexible smoothing with B-splines and penalties. *Statistical science*, *11*(2), 89–121.

Fan, J. & Yao, Q. (2005). *Nonlinear time series : Nonparametric and parametric methods*. Springer New York.

Freund, Y. & Schapire, R. R. E. (1996). Experiments with a New Boosting Algorithm. In *International conference on machine learning* (pp. 148–156).

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, *4*(1), 1–58.

Gooijer, J. G. D. & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*(3), 443–473.

Kantz, H. & Schreiber, T. (2004). *Nonlinear time series analysis*. New York, NY, USA: Cambridge University Press.

Kock, A. & Teräsvirta, T. (2011). Forecasting with nonlinear time series models. *Oxford Handbook of Economic Forecasting*.

Makridakis, S. G. & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476.

Medeiros, M. C., Teräsvirta, T., & Rech, G. (2006). Building neural network models for time series: a statistical approach. *Journal of Forecasting*, *25*(1), 49–75.

Robinzonov, N., Tutz, G., & Hothorn, T. (2012). Boosting techniques for nonlinear time series models. *AStA Advances in Statistical Analysis*, *96*, 99–122.

Ruppert, D. (2002). Selecting the Number of Knots for Penalized Splines. *Journal Of Computational And Graphical Statistics*, *11*, 735–757.

Schmid, M. & Hothorn, T. (2008). Boosting additive models using component-wise P-Splines. *Computational Statistics & Data Analysis*, *53*(2), 298–311.

Shafik, N. & Tutz, G. (2009). Boosting nonlinear additive autoregressive time series. *Computational Statistics & Data Analysis*, *53*(7), 2453–2464.

Shrestha, D. L. & Solomatine, D. P. (2006). Experiments with AdaBoost.RT, an improved boosting scheme for regression. *Neural computation*, *18*(7), 1678–710.

Teräsvirta, T., Tjø stheim, D., & Granger, C. W. J. (2010). *Modelling Nonlinear Economic Time Series*. Advanced Texts in Econometrics. OUP Oxford.

A. Appendix

A.1. Monte Carlo simulations

The nonlinear AR process is given by

$$y_t = -0.17 + 0.85y_{t-1} + 0.14y_{t-2} - 0.31y_{t-3} + 0.08y_{t-7} + 12.80\, G_1(\boldsymbol{y}_{t-1}) + 2.44\, G_2(\boldsymbol{y}_{t-1}) + \varepsilon_t$$

with $G_1(\boldsymbol{y}_{t-1}) = (1+\exp\{-0.46(0.29y_{t-1} - 0.87y_{t-2} + 0.40y_{t-7} - 6.68)\})^{-1}$ and $G_2(\boldsymbol{y}_{t-1}) = (1+\exp\{-1.17 \times 10^3(0.83y_{t-1} - 0.53y_{t-2} - 0.18y_{t-7} + 0.38)\})^{-1}$ where $\varepsilon_t \sim \text{NID}(0,1)$.

We set the error variance to a value which assures a descent predictability for the time series. Medeiros et al. (2006) build this process by fitting an artificial neural network with two hidden units to the annual sunspot series. In Kock and Teräsvirta (2011), this process has been used to compare different forecasting methods in a nonlinear setting.

The linear AR process is given by

$$y_t = 1.32y_{t-1} - 0.52y_{t-2} - 0.16y_{t-3} + 0.18y_{t-4} - 0.26y_{t-5} + 0.19y_{t-6} + \varepsilon_t,$$

where $\varepsilon_t \sim \text{NID}(0,1)$.

This process exhibits cyclic behaviour and was selected by fitting an AR(6) model to the famous annual sunspot series. Because it is a linear process, the variance of ε_t simply scales the resulting series. Consequently, we set the error variance to one without loss of generality.

For all strategies, the maximum number of lagged values p were selected from $2,\ldots,12$ and $2,\ldots,7$ for the nonlinear and linear DGP, respectively.

A.2. Real world experiments

The M3 competition dataset consists of 3003 monthly, quarterly, and annual time series. The time series of the M3 competition have a variety of features. Some have a seasonal component, some possess a trend, and some are just fluctuating around some level. The length of the time series ranges between 14 and 126. We have considered time series with a range of lengths between $T = 117$ and $T = 126$. So, the number of considered time series turns out to be $M = 339$. For these time series, the competition required forecasts for the next $H = 18$ months, using the given historical data.

The NN5 competition dataset comprises $M = 111$ time series representing roughly two years of daily cash withdrawals ($T = 735$ observations) at ATM machines at one of the various cities in the UK. For each time series, the competition required to forecast the values of the next $H = 56$ days (8 weeks), using the given historical data.

Let us define the forecast error for the mth time series at horizon h as $e^m_{T+h} = \hat{y}^m_{T+h} - y^m_{T+h}$ where $m \in \{1,\ldots,M\}$ and $h \in \{1,\ldots,H\}$.

The symmetric mean absolute percentage error (sMAPE) at horizon h is defined as

$$\boxed{\text{sMAPE}_h = \text{mean}(\text{sAPE}^m_h)}$$

where

$$\text{sAPE}^m_h = 200 * \frac{|e^m_{T+h}|}{(|\hat{y}^m_{T+h}| + |y^m_{T+h}|)}$$

For all strategies, the maximum number of lagged values p were selected from the set $2,\ldots,5$ and $2,\ldots,10$ for the M3 and the NN5 competition, respectively.

A.3. Learning models and computational details

For the recursive and direct strategies, we use a single hidden-layer feedforward neural network model. This choice is motivated by the fact that neural networks have proven to be one of the most effective machine learning models in the forecasting literature. We allowed the number of hidden nodes and the parameter for weight decay to vary in $\{0,1,2,3,5\}$ and $\{0, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3\}$, respectively.

For the boost strategy, we used P-splines with 20 equally spaced knots and four degrees of freedom for the weak learners. For the hyperparameter values, we set the value of ν to 0.2 and the maximum number of iterations to 500.

The different model hyperparameters have been selected using a time-series cross-validation procedure with an initial training set containing 70% of the data and a validation set with the remaining 30%.

All the experiments have been carried out using the `R` programming language. We used the *nnet* package for the neural network model and the *mboost* package for gradient boosting. `R` code to perform all experiments will be made available on github[2].

[2] https://github.com/bsouhaib