A Bias and Variance Analysis for Multi-Step-Ahead Time Series Forecasting

Souhaib Ben Taieb and Amir F. Atiya, *Senior Member, IEEE*

***Abstract*—Multi-step-ahead forecasts can either be produced recursively by iterating a one-step-ahead time series model or directly by estimating a separate model for each forecast horizon. In addition, there are other strategies, some of them combine aspects of both aforementioned concepts. In this work we present a comprehensive investigation into the bias and variance behavior of multi-step-ahead forecasting strategies. We provide a detailed review of the different multi-step-ahead strategies. Subsequently, we perform a theoretical study that derives the bias and variance for a number of forecasting strategies. Finally, we conduct a Monte Carlo experimental study that compares and evaluates the bias and variance performance of the different strategies. From the theoretical and the simulation studies we analyze the effect of different factors, such as the forecast horizon and the time series length, on the bias and variance components, and on the different multi-step-ahead strategies. Several lessons are learned, and recommendations are given concerning the advantages, disadvantages, and best conditions of use of each strategy.**

***Index Terms*—Multi-step-ahead forecasting, time series, machine learning, nearest neighbors, neural networks, bias, variance, Monte Carlo simulation.**

I. Introduction

LINEAR forecasting methods, such as ARIMA and exponential smoothing [1], have been dominantly used in the majority of forecasting applications. This is because they are robust methods, and are fairly well-understood due to decades of development and analysis. However, in the last two decades nonlinear forecasting methods have proved themselves and are making inroads into many applications. Examples of nonlinear methods are some statistical models, such as bilinear models, regime-switching models and functional-coefficient models [2], [3]. Alternatively, they could be machine learning models [4], [5]. Examples include K-nearest-neighbor [6], [7], neural networks [8]–[10], support vector machines [11], boosting [12], and fuzzy neural networks [13]–[15].

There are many situations where the time series behaves nonlinearly, and therefore a nonlinear model would be the appropriate choice [16]. For instance, a time series could exhibit some form of saturation effect (for example the variable's effect becomes less pronounced as it increases), or it could switch between two or more different regimes (for example economic expansion and recession) [2]. In addition to model development, it is imperative to have a parallel effort to understand the inner workings of these nonlinear models. This will shed light into their strengths and weaknesses, and therefore channel the research effort into alleviating their weaknesses. In addition, it can help guiding the user to the proper selection of models.

As a step towards the understanding of nonlinear forecasting methods, we consider in this paper the multi-step-ahead forecasting problem and provide an in-depth analysis, using theoretical arguments, qualitative analysis, and simulation experiments. Forecasting multi-step-ahead, rather than a single-step, is more prevalent in the majority of applications. In spite of this, it has been much less studied, partly because it is a more difficult problem. This is because further steps have more uncertainty and are typically harder to forecast, and also because of the potentially complex interaction between the different steps-ahead, i.e. forecasting horizons. There are three major strategies for the multi-step-ahead forecasting problem. In the *recursive* strategy the forecasting model is trained to forecast one-step ahead. Subsequently, forecasting h-step ahead is accomplished by iterating the forecasts h times, using previously forecasted values as inputs as needed. In the *direct* strategy, separate forecasting models are trained to directly forecast each h-step ahead. In the *joint* strategy, also called multi-input multi-output, or MIMO, (available for nonlinear forecasting models only) one multi-output model is trained to forecast the whole horizon in one shot. In this work we are *not* going to study or propose *forecasting models* (such as neural networks, SVR's, etc), but study *multi-step-ahead forecasting strategies*, i.e. the way we apply any forecasting model to obtain the forecasts for the whole horizon, e.g the recursive strategy, the direct strategy, and several other ones.

There has been some theoretical and empirical work comparing between the recursive and the direct strategies for linear models [17]–[23]. A summary of the findings is given by [24]. Most studies agree that the direct strategy is superior for the case of misspecified models, i.e. when the considered class of models does not contain the true model. Otherwise, for well-specified models, the recursive strategy may be better. Essentially, there are two conflicting factors. Misspecification introduces a bias that gets worse with horizon. On the other hand, the direct strategy's error terms are serially correlated (one can determine that by a simple analysis of the ARMA process). The direct strategy has also an effectively smaller in-sample set, and can therefore suffer from a higher variance of the forecasts.

For the nonlinear case only little work has appeared in the literature. For example, [25] was one of the earlier studies that compared the two (direct and recursive) strategies with respect to their asymptotic efficiency. Also, [26] compared the

S. Ben Taieb is with the Department of Computer Science, Université Libre de Bruxelles, Belgium (email: sbentaie@ulb.ac.be).

A. F. Atiya is with the Department of Computer Engineering, Cairo University, Giza, Egypt (email: amir@alumni.caltech.edu).

mean squared error (MSE) of both strategies, in addition to a third strategy that has considered some combination of joint and recursive forecasting. In their study, which is based on neural network models, the direct strategy gave the best results. The works by [27] and [8], which reviewed the literature for nonlinear forecasting models, studied both direct and recursive strategies for such nonlinear models. The work of [28] derived a joint forecasting strategy for support vector regression. Moreover, they compared it with the recursive and the direct strategies, and found that the joint strategy is the better of the three. The study by [29] compared between the direct and the recursive strategies using the concept of grey relational analysis. The work by [30] compared between various multi-step-ahead forecasting strategies. Moreover, they studied the problem characteristics that may affect the outperformance of one strategy over another. However, this study does not include many of the approaches that were developed later. The work by [31] compared between the three basic multi-step strategies for some flood forecasting problem using a neural network as underlying model. They discovered that the recursive strategy is the best for long-term forecasting. For short-term forecasting the recursive strategy, tied with the direct strategy, beat the joint strategy.

Other than nonlinear forecasting models, some researchers considered multi-step strategies with GARCH-type underlying forecasting models. For example [32] considered three strategies: recursive, direct, and mixed data sampling (MIDAS) for multi-step volatility forecasting. The MIDAS strategy is an approach that uses higher frequency time series as inputs to forecast time-aggregated values. They found out that the recursive strategy holds an edge at short horizons, while the MIDAS strategy considerably beats the other two for larger horizons.

Other work has focused on developing novel strategies, combining properties of the three recursive, direct, and joint strategies (see section II). For example, [33] provided a review and a comparison of several multi-step strategies using the NN5 benchmark time series. They propose other variants and combinations of the three basic strategies. They show that the joint strategy is superior, but also some of the other derivative strategies (from the basic three strategies) are competitive.

All the aforementioned studies have considered only overall measures such as the MSE to evaluate and compare the different multi-step-ahead strategies for nonlinear forecasting. However, the error in any forecasting problem can be decomposed into the sum of two opposing components: the bias and the variance. This decomposition is such a fundamental concept that it exists in other problems that involve some form of estimation, such as classification, regression and parameter estimation. The present study investigates the behavior of the bias and the variance components for each forecasting strategy. The bias represents the consistent offset of the forecast, away from the true value. For example, if for a family of forecasting models the forecast for a specific x is always higher by some amount than the true value, then the bias is positive. The variance represents the variation of the forecast around its mean. So, a forecasting model that, for different realization of the training set, produces highly variable forecasts, then it has a high variance. A more complex model (i.e. a model with a large number of parameters) typically has a low bias and a high variance, and vice versa for a simple model. The reason is that a simple model has less parameters. This will make the model less powerful, and therefore less able to fit any shape, leading to a large bias. On the other hand the smaller number of parameters leads to smaller sensitivity to the data, and therefore smaller variance.

Analyzing the behavior of bias and variance is paramount to understanding the inner mechanics of forecasting strategies. These are fundamental concepts that reveal the relation with model complexity, model misspecification, and data adequacy. An in-depth study could therefore give insights into the different interactions between time series length, model complexity, forecasting horizon (i.e. step-ahead), and the strategy's performance. It also could guide the selection of multi-step strategies. For example, if we are encountering a short time series, then we are more vulnerable to overfitting, and it is prudent to use a multi-step strategy that minimizes the variance. On the other hand, if our application pays much attention to the direction of the forecast (rather than value), then it is recommended to use a low-bias strategy, because a large bias can deteriorate the forecasted direction. The only study that considered bias and variance for nonlinear forecasting, and is therefore the closest to our work is the study [34], which presented an empirical investigation for one-step ahead forecasting using neural networks. However, there is a distinctive difference between their study and ours. First, we consider multi-step forecasting instead of one-step forecasting, which is a fundamental difference. Second, they compared *forecasting models* (or learning algorithms), while we compare *multi-step strategies* (like the recursive, direct, and other strategies). A notable theoretical study on bias and variance analysis for multi-step forecasting for linear models was presented by [35]. He proved that standard model selection criteria, such as Akaike, Schwartz's BIC, and leave one out methods are biased estimators of the MSE for multi-step models. These criteria are generally under-penalizing for over-parameterization, and he suggested instead the leave-h-out cross validation criterion, which he proves has better properties. His study, however, did not consider or compare multi-step strategies.

In summary, the contributions of our work are the following:

- Review in detail the strategies for multi-step-ahead forecasting.
- Present a theoretical analysis for the bias and variance for several of these strategies.
- Conduct a simulation study for the comparison of the different strategies from the perspective of bias and variance components.
- Provide a qualitative reasoning concerning the strengths and weaknesses of each strategy in their bias and variance performance.

A brief summary of the findings is that the direct strategy generally has the smallest bias, and becomes superior for longer time series. On the other hand, the joint strategy has the smallest variance. Taking into account the combined effect

of bias and variance, we recommend as the overall winner a variant of the joint strategy, whereby the forecasting model is trained to simultaneously forecast only a portion rather than the whole horizon at a time.

The paper is organized as follows. The next section provides the MSE decomposition for multi-step forecasting. Section II presents the different forecasting strategies. A theoretical bias and variance analysis for two steps ahead is proposed in Section IV. Section V gives the methodology of the simulations with an explanation of the bias and variance estimation as well as the simulation details. Section VI presents a discussion of the results. Finally, Section VII gives a summary and concludes the work.

II. Multi-step-ahead forecasting strategies

Given a univariate time series $\{y_1, \ldots, y_T\}$ comprising T observations, we want to forecast the H next observations of the time series, $\{y_{T+1}, \ldots, y_{T+H}\}$.

Time series of different fields or applications can have different resolutions (e.g. yearly, monthly, daily, hourly, etc), and that could lead to different time series lengths T. Also, depending on the required horizon H, forecasts can be typically classified into short, medium or long term forecasts. Typically, the further in the future we attempt to forecast the harder it can be because of the larger uncertainty.

We will assume the time series $\{y_1, \ldots, y_T\}$ is a realization of an autoregressive process of the form

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

which is specified by a function f, a lag order (or number of lagged variables) d and a noise term $\{\varepsilon_t\}$, which is a stochastic iid noise process with $\mathbb{E}[\varepsilon_t] = 0$ and $\mathbb{E}[\varepsilon_t^2] = \sigma^2$.

Different forms or values of these three components can produce time series with very different characteristics. The autoregressive process in (1) is also called the data generating process (DGP) for the time series $\{y_1, \ldots, y_T\}$. In particular, time series generated by different DGPs can have very different forecastability properties.

In practice, we do not have access to the true DGP (if it exists). The only information we have is one time series realization from that DGP. The goal is to produce the best forecasts (according to an accuracy measure) based on this time series realization.

We will consider the mean squared error (MSE) as forecast error measure. Let us denote $g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_T; h)$ the h-step ahead forecast from input $\boldsymbol{x}_t$ using the set of parameters $\hat{\boldsymbol{\theta}}_T$, that have been estimated from the time series $\boldsymbol{Y}_T = \{y_1, \ldots, y_T\}$. Then, the MSE at horizon h is defined as

$$\text{MSE}_h(\boldsymbol{x}_t) = \mathbb{E}_{\varepsilon, \boldsymbol{Y}_T}\left[(\boldsymbol{y}_{t+h} - g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_T; h))^2\right]. \tag{2}$$

It can be shown that the optimal h-step ahead forecast, i.e. the forecast that has the minimum MSE at horizon h, is the conditional expectation given by $\mu_{t+h|t} = \mathbb{E}[y_{t+h}|\boldsymbol{x}_t]$ (see, for example [36]). In this article, we will assume the goal of multi-step-ahead forecasting is to estimate $\mu_{t+h|t}$ for $h = 1, \ldots, H$ using one time series realization $\{y_1, \ldots, y_T\}$.

For one-step-ahead forecasts, that is $h = 1$, we have the expression $\mu_{t+1|t} = f(\boldsymbol{x}_t)$. So, the problem of forecasting reduces to the estimation of the function f and the lag order d, given in expression (1).

For multi-step forecasts, that is $h > 1$, the problem is more difficult and does not necessarily reduces to the estimation of the function f and the lag order d. In fact, to produce multi-step-ahead forecasts, we need a forecasting strategy which typically involves estimating one or more models which are not necessarily of the same form as f and may not have the same lag order d as the function f.

Multi-step-ahead forecasting strategies can be classified based on whether actual forecasts are used to generate the next forecasts or if the forecasts are generated directly without use of these intermediate forecasts. The former group of strategies is called *recursive* strategies while the latter group is called *direct* strategies. Any multi-step forecasting strategy could also have some aspects of both recursive and direct approaches. In what follows we describe different variants of both recursive and direct strategies.

A. Recursive and direct strategies

The recursive strategy estimates one model m given by

$$y_t = m(\boldsymbol{z}_{t-1}; \boldsymbol{\theta}) + e_t\,, \tag{3}$$

with $\boldsymbol{z}_t = [y_t, \ldots, y_{t-p+1}]'$ and $\mathbb{E}[e_t] = 0$.

This strategy trains one model which focuses solely on the one-step ahead forecasting. This means that the set of parameters $\boldsymbol{\theta}$ are estimated by minimizing a one-step error criterion with

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_t \left(y_t - m(\boldsymbol{z}_{t-1}; \boldsymbol{\theta})\right)^2, \tag{4}$$

where $\boldsymbol{\Theta}$ denotes the parameter space.

After estimating the set of parameters $\boldsymbol{\theta}$, the forecasts are computed recursively. This means that intermediate forecasts are used as input variables for forecasting successive time series values.

The forecasts are computed as $\hat{\mu}_{T+h|T} = m^{(h)}(\boldsymbol{z}_T; \hat{\boldsymbol{\theta}})$ for all $h = 1, \ldots, H$, where $m^{(h)}$ means applying the model m recursively h times starting from $\boldsymbol{z}_T$. For example $\hat{\mu}_{T+2|T} = m^{(2)}(\boldsymbol{z}_T; \hat{\boldsymbol{\theta}}) = m(\hat{\boldsymbol{z}}_{T+1}; \hat{\boldsymbol{\theta}})$ where $\hat{\boldsymbol{z}}_{T+1}$ includes intermediate forecasts in place of actual time series values for the times where such are not yet known.

One advantage of the recursive strategy is the computational time since it requires learning a single model. This strategy will be denoted as **REC**.

Instead of using one parameter for all horizons, a variation of REC is to estimate a different set of parameters $\hat{\boldsymbol{\theta}}^{(h)}$ for each horizon h by minimizing an h-step error criterion. In other words, we compute for each horizon h,

$$\hat{\boldsymbol{\theta}}^{(h)} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_t \left[y_t - m^{(h)}(\boldsymbol{z}_{t-h}; \boldsymbol{\theta})\right]^2, \tag{5}$$

Hence, a different set of parameters is used at each horizon h.

The forecasts are computed as the recursive strategy with $\hat{\mu}_{T+h|T} = m^{(h)}(\boldsymbol{z}_T; \hat{\boldsymbol{\theta}}^{(h)})$ for all $h = 1, \ldots, H$.

One advantage of this strategy is the use of an h-step criterion instead of an 1-step criterion as with REC . This strategy will be denoted as **RECMULTI**.

In the *direct* strategy for each horizon there is a different forecasting model, that is trained to specifically forecast only the h-step ahead value. It uses lagged input variables up to time t for forecasting time $t+h$, and therefore does not attempt to use intermediate forecasts. It is given by

$$y_t = m_h(\boldsymbol{r}_{t-h}; \boldsymbol{\theta}_h) + e_{t,h} \tag{6}$$

where $\boldsymbol{r}_{t-h} = [y_{t-h}, \ldots, y_{t-h-p_h}]'$ and $h = 1, \ldots, H$.

This strategy uses a different set of parameters $\boldsymbol{\theta}_h$ for each horizon h and computed by

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h \in \boldsymbol{\Theta}_h}{\operatorname{argmin}} \sum_t \left[y_t - m_h(\boldsymbol{r}_{t-h}; \boldsymbol{\theta}_h) \right]^2 . \tag{7}$$

Then the forecasts are obtained for each horizon from the corresponding model, that is $\hat{\mu}_{T+h|T} = m_h(\boldsymbol{z}_T; \hat{\boldsymbol{\theta}}_h)$.

One advantage of the direct strategy is its flexibility as it allows different number of lags p_h and a different parameter set for each horizon h. This strategy will be denoted as **DIR**.

The three previous strategies have notably been considered in [24] for the linear case and in [26] for the nonlinear case.

B. Combination of recursive and direct strategies

In addition to the basic recursive and direct strategies, researchers have considered several combinations of these two strategies.

A straightforward approach to combine the recursive and the direct strategies is to take a weighted average of forecasts from the REC and the DIR strategies. Researchers have considered more advanced combination schemes which modify the model building and/or the forecasting procedures.

The **DirRec** strategy [37], also called the mixed strategy [38], uses a different set of parameters $\boldsymbol{\theta}_h$ for each horizon h as in the DIR strategy but includes the previous forecasts with the input variables. In other words, the parameters are estimated as follows

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h \in \boldsymbol{\Theta}_h}{\operatorname{argmin}} \sum_t \left[y_t - [m_h(\hat{m}_{h-1}, \ldots, \hat{m}_1, \boldsymbol{r}_{t-h}; \boldsymbol{\theta}_h)] \right]^2 . \tag{8}$$

where $\hat{m}_h$ is a shorthand for $m_h(\boldsymbol{r}_{t-h}; \hat{\boldsymbol{\theta}}_h)$. Then the forecasts are obtained for each horizon from the corresponding model, that is $\hat{\mu}_{T+h|T} = m_h(\hat{m}_{h-1}, \ldots, \hat{m}_1, \boldsymbol{r}_T; \hat{\boldsymbol{\theta}}_h)$.

So, the model at horizon h is not learnt independently from the previous models. However, because the number of input variables grows linearly with the horizon, this strategy requires a variable selection/elimination method and so is computationally demanding.

The **MSVR** strategy, proposed in [39], is another combination of the REC and DIR architectures. If we assume $H = L \times R$, this strategy estimates the first L direct models as in (7). Then, the estimated models are used R times to produce the H forecasts. In other words, the forecasts are obtained as

$$\hat{\mu}_{T+h|T} = \begin{cases} m_l(\boldsymbol{r}_T; \hat{\boldsymbol{\theta}}_l) & \text{if } h \leq L \\ m_l(\hat{m}_{l-1}, \ldots, \hat{m}_1, \boldsymbol{r}'_T; \hat{\boldsymbol{\theta}}_l) & \text{if } h > L \end{cases} \tag{9}$$

where $l = (h-1)\%L + 1$, $\hat{m}_l$ is a shorthand for $m_l(\boldsymbol{r}_T; \hat{\boldsymbol{\theta}}_l)$ and $\boldsymbol{r}'_T \subset \boldsymbol{r}_T$.

One advantage of the MSVR strategy is the gain in computational time, since it requires learning L direct models with $L = \frac{H}{R} \leq H$. However, because estimated values are used to forecast next values, there is a trade-off between computational time and forecast accuracy.

The **RECTIFY** strategy [40] seeks to combine the best properties of both the recursive and direct forecasting strategies. The rationale behind the strategy is to first apply a linear recursive forecasting model, and then adjust its forecasts using a nonlinear model trained with the direct strategy. This adjustment is meant to correct the bias that is typical of a recursive linear system. The parameters are estimated as follows

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h \in \boldsymbol{\Theta}_h}{\operatorname{argmin}} \sum_t \left[y_t - [m^{(h)}(\boldsymbol{z}_{t-h}; \hat{\boldsymbol{\theta}}) + m_h(\boldsymbol{r}_{t-h}; \boldsymbol{\theta}_h)] \right]^2 . \tag{10}$$

Then the forecasts are obtained as $\hat{\mu}_{T+h|T} = m^{(h)}(\boldsymbol{z}_T; \hat{\boldsymbol{\theta}}) + m_h(\boldsymbol{r}_T; \hat{\boldsymbol{\theta}}_h)$. One advantage of this strategy is that it avoids the difficult task of choosing between the recursive and the direct strategies.

C. Multi-horizon strategies

The forecasting strategies in sections II-A and II-B can be classified into the set of *single-horizon* strategies, that is strategies where the forecasting model considers each horizon in isolation. There is no attempt to connect in some way the forecasts of different steps-ahead.

Another set of strategies, called the *multi-horizon* strategies, is based on forecasting several horizons in one shot. The forecasting error function that is minimized during the training process takes into account simultaneously the forecast errors of several horizons, and therefore there is one set of parameters shared between these horizons.

The rationale behind the multi-horizon strategies is that the different models m_h in (6) share some common characteristics because of the serial correlation in time series data. Thus, estimating the parameters of each model m_h in a joint manner could be beneficial since (i) it allows exploiting the relatedness between the different horizons' forecasting tasks to improve generalization performance, (ii) it avoids potential irregularities in consecutive forecasts due to using very different models at each horizon and (iii) it compensates for the small sample size using additional samples from these other related tasks.

This is a special case of the broader concept of *multi-task regression* [41] or *multiresponse* regression [42], developed in the machine learning and the statistics literature.

We present here the three multi-horizon strategies which have been proposed in the literature, each of which makes use of a different formulation of the objective function, and this, consequently, reflects on the way the parameters are optimized.

The **DIRJOINT** strategy uses one set of parameters $\boldsymbol{\theta}$ that is shared with *all* the horizons, and this is estimated by minimizing the average error over the entire horizon, i.e.

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_t \frac{1}{H} \sum_{h=1}^{H} \left[y_t - m(\boldsymbol{r}_{t-h}; \boldsymbol{\theta}) \right]^2 . \tag{11}$$

This strategy has been called the *MIMO* strategy, when used with nearest neighbors [43], and has also been called the *JOINT* strategy in the context of neural networks forecasting models [44].

An example of how to apply this strategy for neural networks (NN) is to consider a NN architecture with an input layer of size p, corresponding to the time series lags, and an output layer of size H, corresponding to the different horizon forecasts. Thus, the neural network is designed so that it simultaneously produces all H-step ahead forecasts in its output layer.

The **SJOINT** strategy [45] takes a middle approach between the DIR and the DIRJOINT strategies. Instead of having one parameter set for all horizons or one parameter set for each horizon, the SJOINT strategy forms several blocks of consecutive horizons and use ones set of parameters for each group. In other words, assuming $H = L \times R$, this strategy obtains L different multi-horizon models where the parameters are estimated as

$$\hat{\boldsymbol{\theta}}_l = \underset{\boldsymbol{\theta}_l \in \boldsymbol{\Theta}_l}{\operatorname{argmin}} \sum_t \frac{1}{R} \sum_{h=(l-1)R+1}^{l \times R} [y_t - m_l(\boldsymbol{r}_{t-h}; \boldsymbol{\theta}_l)]^2 . \quad (12)$$

with $l = 1, \ldots, L$.

In addition to the set of parameters of the different models, the SJOINT strategy requires the selection of the number of groups L.

The **DIRJOINTL** strategy bundles horizons that are in the vicinity of the current horizon. In other words, we compute for each horizon h,

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h \in \boldsymbol{\Theta}_h}{\operatorname{argmin}} \sum_t \tfrac{1}{i+j+1} \sum_{h'=h-i}^{h+j} [y_t - m_h(\boldsymbol{r}_{t-h'}; \boldsymbol{\theta})]^2,$$

where i and j are respectively the number of horizons before and after the current horizon that are included in the objective function.

This strategy has been considered in [46] with $i = j = 1$ for traffic flow forecasting using neural networks.

This previous group is of the *direct* type. There are also multi-horizon strategies for recursive models $m^{(h)}$ (see (5)), where the parameters are estimated in a joint manner.

As with the DIRJOINT strategy, RECMULTI (or rather REC) can be extended to minimize an H-step error to select *one* set of parameter for all horizons h. In other words, we compute

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \sum_t \frac{1}{H} \sum_{h=1}^{H} \left[y_t - m^{(h)}(\boldsymbol{z}_{t-h}; \boldsymbol{\theta}) \right]^2, \quad (13)$$

where $m^{(h)}$ is an iterative h-step ahead forecast obtained by recursive application of a one-step ahead model.

This means that the parameters are optimized taking into account their whole effect on all future steps, rather than the myopic one-step ahead view of REC . This strategy is akin to the backpropagation through time approach for neural networks [26], [47] whereby a pass through the whole horizon in the training process ensures considering the effect on every step ahead. Once trained, the forecasts are obtained recursively as with REC . We will denote this strategy as **RECJOINT**.

We can also define the recursive version of JOINTL , denoted **RECJOINTL**, for which the set of parameters are computed as

$$\hat{\boldsymbol{\theta}}_h = \underset{\boldsymbol{\theta}_h \in \boldsymbol{\Theta}_h}{\operatorname{argmin}} \sum_t \tfrac{1}{i+j+1} \sum_{h'=h-i}^{h+i} \left[y_t - m^{(h)}(\boldsymbol{r}_{t-h'}; \boldsymbol{\theta}) \right]^2$$

The forecasts are then obtained as for REC .

Note that the idea of minimizing an h-step ahead error has also been considered in [48] to better match the feature of a time series.

For all the previously described multi-horizon strategies (both direct and recursive), we can associate a set $L_h \subseteq \{1 \ldots, H\}$ to each horizon h, which represents the set of horizons involved in the objective of horizon h. Then each multi-horizon strategy can be described with $\{L_1, \ldots, L_H\}$.

For example, the DIR strategy has $L_h = h$. This means the only horizon included in the objective function for horizon h is the horizon h itself. For the DIRJOINT strategy, $L_h = \{1 \ldots, H\}$; this means all the horizons are included in the objective function. Finally, for JOINTL , $L_h = \{h - i, \ldots, h, \ldots, h + i\}$.

III. Mean squared multi-step forecast error decomposition

An in depth analysis of the performance of forecasting strategies can be accomplished through an examination of the MSE decomposition into the bias and the variance components. Given a realization $\boldsymbol{Y}_T$, we denote $g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)$, the h-step ahead forecasts of a given strategy for the input $\boldsymbol{x}_t$. For each realization, a lag order p and a set of parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}$ are estimated, and could possibly be different from one realization to the other. In particular, the estimated lag order p could possibly be different from the "real" lag order d defined in (1). The variation of the forecasts $g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)$ with the different realizations of $\boldsymbol{Y}_T$ gives rise to the variance component. In addition, the variation of $g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h)$ is around a certain mean: $\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T}; h) \right]$ (or rather a mean curve that is a function of h). This represents the forecast averaged over all possible variations of the training set. The difference between that mean and the optimal forecast $\mu_{t+h|t}$ represents the bias.

The MSE of the given strategy at horizon h is decomposed as follows.

$$
\begin{aligned}
&\text{MSE}_h \\
&= \mathbb{E}_{\boldsymbol{x}_t}\left[\underbrace{\mathbb{E}_{\varepsilon,\boldsymbol{Y}_T}\left[(y_{t+h}-g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h))^2 \mid \boldsymbol{x}_t\right]}_{\text{MSE}_h(\boldsymbol{x}_t)}\right] \\
&= \mathbb{E}_{\boldsymbol{x}_t}\Bigg[\underbrace{\mathbb{E}_{\varepsilon}\left[(y_{t+h}-\mu_{t+h|t})^2 \mid \boldsymbol{x}_t\right]}_{N_h(\boldsymbol{x}_t)} \\
&\quad + \underbrace{(\mu_{t+h|t}-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2}_{B_h(\boldsymbol{x}_t)} \qquad (14) \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{Y}_T}\left[(g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2 \mid \boldsymbol{x}_t\right]}_{V_h(\boldsymbol{x}_t)}\Bigg] \\
&= \underbrace{\mathbb{E}_{\boldsymbol{x}_t,\varepsilon}\left[(y_{t+h}-\mu_{t+h|t})^2 \mid \boldsymbol{x}_t\right]}_{N_h} \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{x}_t}\left[(\mu_{t+h|t}-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2\right]}_{B_h} \qquad (15) \\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{x}_t,\boldsymbol{Y}_T}\left[(g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2 \mid \boldsymbol{x}_t\right]}_{V_h}
\end{aligned}
$$

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 $g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)$ at horizon h can be decomposed into three different components, namely the noise term N_h, the squared bias term B_h and the variance term V_h.

The noise component N_h is the irreducible error that cannot be eliminated. This means that even with optimal forecasts (i.e. being able to perfectly estimate $\mu_{t+h|t}$ so that $B_h = 0$ and $V_h = 0$), MSE_h will be equal to N_h.

In contrast to the noise component, the bias and variance depend on the employed forecasting strategy. The bias, denoted by B_h in (15), represents the consistent offset of the forecast, away from the true value, i.e. the conditional mean. For example, a trend deviation in the forecast or the use of a linear model with a nonlinear time series may affect the bias component. In fact, the bias term can decomposed further into

$$
\begin{aligned}
&B_h \qquad (16) \\
&= \mathbb{E}_{\boldsymbol{x}_t}\left[(\mu_{t+h|t}-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2\right] \qquad (17) \\
&= \mathbb{E}_{\boldsymbol{x}_t}\Bigg[\underbrace{(\mu_{t+h|t}-g(\boldsymbol{x}_t;\boldsymbol{\theta}^*;h)}_{A} \qquad (18) \\
&\quad + \underbrace{g(\boldsymbol{x}_t;\boldsymbol{\theta}^*;h)-\mathbb{E}_{\boldsymbol{Y}_T}\left[g(\boldsymbol{x}_t;\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}_T};h)\right])^2}_{B}\Bigg] \qquad (19)
\end{aligned}
$$

The A term represents the discrepancy between the conditional mean and the best potential of the model family we consider. For example, consider that $\mu_{t+h|t}$ is nonlinear, while we consider a linear forecasting model, then $g(\boldsymbol{x}_t;\boldsymbol{\theta}^*;h)$ is the forecast, having perfect estimation of the linear parameter. In such a case the A term is a measure of the limitation of the selected model (or model family), not taking into account parameter or model estimation errors. The B term represents the error due to having the time series limited to T observations, i.e. it expresses how finite-sampledness will affect the bias. For example, even if $\mu_{t+h|t}$ can be correctly estimated with $g(\boldsymbol{x}_t;\boldsymbol{\theta}^*;h)$ (the A term cancels), then the B term will still remain since there is typically a parameter estimation error, because we are only considering a time series with T observations.

The variance, denoted by V_h in (15), represents the variation of the forecast around its mean. For example, a small sample time series or a too complex model might affect the estimation variance component.

A more complex model will tend to have a low bias, as it is powerful enough to be able to produce any shape of the fit. On the other hand, a simple model will be less malleable and will produce large bias. Concerning the variance, in complex models the fit will tend to be much more volatile, because it is more sensitive to the data and the random terms. Simple models, on the other hand, have less sensitivity, because they have less parameters and hence less "knobs" that can be used to tune any solution.

The ideal configuration is to have both a low bias and a low variance. However, this ideal configuration is never achieved in practice as decreasing the bias will increase the variance and vice versa. So, the role of model selection is to find a trade-off between the bias and the variance to obtain the smallest MSE.

The goal of this work is to analyze the bias and the variance components of the forecasting strategies over the forecast horizon. The MSE decomposition given in (15) is identical to the usual decomposition used in the machine learning field [49]. However, in contrast with usual supervised learning problems, the multi-step forecasting problem is dealing with time-dependent data (time series) and requires the learning of dependent tasks with different noise level changing with the forecast horizon.

IV. Theoretical analysis of the bias and the variance

We present here a theoretical analysis that analyzes the bias and variance behavior for the different multi-step forecasting strategies. We will consider the strategies from Sections II-A and II-C but not from Section II-B. For simplicity, consider the two-step ahead case. The observed findings will generally apply to the general h-step ahead situation. In any case, some Monte Carlo experiments for the case of general forecast horizon h will be performed later in the paper, in order to study and validate the behavior as we vary h.

Assume that the time series is generated by the nonlinear autoregressive process defined in (1). Then, at horizon $h = 1$ we have the simple expression $\mu_{t+1|t} = \mathbb{E}[y_{t+1}|\boldsymbol{x}_t] = f(\boldsymbol{x}_t)$. This means that the conditional expectation at horizon $h = 1$ is simply equal to the iteration function f. At horizon $h = 2$, the conditional expectation $\mu_{t+2|t} = \mathbb{E}[y_{t+2}|\boldsymbol{x}_t]$ can be obtained as follows. First we compute y_{t+2} using Taylor series approximation and keeping up to second-order terms. This gives the

following expression:

$$y_{t+2} \tag{20}$$
$$= f(f(\boldsymbol{x}_t) + \varepsilon_{t+1}, \ldots, y_{t-d+2}) + \varepsilon_{t+2} \tag{21}$$
$$\approx f(f(\boldsymbol{x}_t), \ldots, y_{t-d+2}) + \varepsilon_{t+1} f_{x_1} + \tfrac{1}{2}(\varepsilon_{t+1})^2 f_{x_1 x_1} + \varepsilon_{t+2}, \tag{22}$$

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 as

$$\mu_{t+2|t} = \mathbb{E}[y_{t+2}|\boldsymbol{x}_t] \approx f(f(\boldsymbol{x}_t), \ldots, y_{t-d+2}) + \tfrac{1}{2}\sigma^2 f_{x_1 x_1} \tag{23}$$

Now let us perform the MSE decomposition. First we derive the noise component for the first two horizons, that is $N_1(\boldsymbol{x}_t)$ and $N_2(\boldsymbol{x}_t)$, as defined in (14).

The noise component at horizon $h = 1$ is simply equal to the variance of the stochastic noise, this means that $N_1(\boldsymbol{x}_t) = \sigma^2$. At horizon $h = 2$, it can be obtained as

$$N_2(\boldsymbol{x}_t) \tag{24}$$
$$= \mathbb{E}_\varepsilon \left[(y_{t+2} - \mu_{t+2|t})^2\right]$$
$$\approx \mathbb{E}_\varepsilon \left[\left(\varepsilon_{t+1} f_{x_1} + \tfrac{1}{2}\varepsilon_{t+1}^2 f_{x_1 x_1} + \varepsilon_{t+2} - \tfrac{1}{2}\sigma^2 f_{x_1 x_1}\right)^2\right]$$
$$= \sigma^2[1 + f_{x_1}^2] + \tfrac{1}{2}\sigma^4 f_{x_1 x_1}^2. \tag{25}$$

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

Note that the noise component does not depend on the forecasting strategy or the learning algorithm, but only on the data generation process (DGP). One can observe from expression (25) that the noise term becomes larger for more strongly nonlinear DGP's (because, then $f_{x_1 x_1}$ will be larger). Note also that it is justified to use a Taylor series approximation, and retain a few terms. The majority of forecasting applications in practice possess either approximately linear, or slightly or moderately nonlinear DGP. This is the nature of systems encountered in practice, which have relatively simple relations among the successive time series values, and this is evidenced by the success of simple forecasting methods [50]. Having a slightly or moderately nonlinear behavior makes a Taylor series approximation fairly accurate.

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 (14), we model the forecasts of each strategy as a sum of three terms: the true function value we are trying to estimate, which 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 factor giving the standard deviation of the term, and ε_η is a stochastic component with $\mathbb{E}[\varepsilon_\eta] = 0$ and $\mathbb{E}[\varepsilon_\eta^2] = 1$ (see Eqs. (26), and (34).

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 $\boldsymbol{Y}_T = \{y_1, \ldots, y_T\}$ 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 the 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 estimates a one-step model as in (3) and produces forecasts for $h = 1$, that is

$$g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}; 1) = \underbrace{\underbrace{\underbrace{f(\boldsymbol{x}_t)}_{\mu_{t+1|t}} + \delta(\boldsymbol{z}_t; \boldsymbol{\theta})}_{m(\boldsymbol{z}_t; \boldsymbol{\theta})} + \eta(\boldsymbol{z}_t; \boldsymbol{\theta})\varepsilon_\eta}_{m(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}})} \tag{26}$$

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

$$g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}; 2) \tag{27}$$
$$= m(m(\boldsymbol{z}_t; \hat{\boldsymbol{\theta}}), \ldots, y_{t-p+2}; \hat{\boldsymbol{\theta}}) \tag{28}$$
$$\approx f(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}) \tag{29}$$
$$+ \delta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\theta}) + \eta(f(\boldsymbol{x}_t), \ldots, y_{t-p+2}; \boldsymbol{\theta})\varepsilon_{\eta_2} \tag{30}$$
$$+ \delta(\boldsymbol{z}_t; \boldsymbol{\theta}) m_{z_1} + \frac{1}{2}[\delta(\boldsymbol{z}_t; \boldsymbol{\theta})]^2 m_{z_1 z_1} \tag{31}$$
$$+ \eta(\boldsymbol{z}_t; \boldsymbol{\theta})\varepsilon_{\eta_1} m_{z_1} + \frac{1}{2}[\eta(\boldsymbol{z}_t; \boldsymbol{\theta})\varepsilon_{\eta_1}]^2 m_{z_1 z_1} \tag{32}$$

where ε_{η_1} and ε_{η_2} are the stochastic components of the variability term around points $\boldsymbol{z}_t$ and $[f(\boldsymbol{z}_t), \ldots, y_{t-p+2}]$ respectively, and m_{z_1} and $m_{z_1 z_1}$ 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

$$g(\boldsymbol{x}_t; \hat{\boldsymbol{\theta}}_2; 2) \tag{33}$$
$$= \underbrace{\underbrace{\mu_{t+2|t} + \delta(\boldsymbol{r}_t; \boldsymbol{\theta}_2)}_{m_2(\boldsymbol{r}_t; \boldsymbol{\theta}_2)} + \eta(\boldsymbol{r}_t; \boldsymbol{\theta}_2)\varepsilon_\eta}_{m_2(\boldsymbol{r}_t; \hat{\boldsymbol{\theta}}_2)} \tag{34}$$

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 other strategies. Except the recursive strategy, all the forecasting strategies presented in section II use the h-step error as their objective. Thus, the forecasts of these strategies at horizon $h = 2$ can be written as

$$g(\boldsymbol{x}_t;\hat{\boldsymbol{\gamma}};2) \tag{35}$$

$$= \underbrace{\underbrace{\mu_{t+2|t} + \delta(\boldsymbol{r}_t;\boldsymbol{\gamma})}_{m_2(\boldsymbol{r}_t;\boldsymbol{\gamma})} + \eta(\boldsymbol{r}_t;\boldsymbol{\gamma})\varepsilon_\eta}_{m_2(\boldsymbol{r}_t;\hat{\boldsymbol{\gamma}})} \tag{36}$$

where $\boldsymbol{\gamma}$ is the set of parameter, and this will be different for each strategy. For example, for RECMULTI , $\boldsymbol{\gamma} = \boldsymbol{\theta}^{(2)}$ and for DIRJOINT , $\boldsymbol{\gamma} = \boldsymbol{\theta}$. The number of parameters involved and the objective used to identify them will have an impact on both bias and variance components.

Recall that we would like to estimate $\mu_{t+2|t} \approx f(f(\boldsymbol{x}_t),\ldots,y_{t-d+2}) + \frac{1}{2}\sigma^2 f_{x_1x_1}$. In the following, we will calculate the sum of the squared bias and variance components as defined in (15), for horizon $h = 2$ using the previous expressions for the forecasts of the different strategies. In other words, we compute

$$B_2(\boldsymbol{x}_t) + V_2(\boldsymbol{x}_t) \tag{37}$$

$$\approx \left(\mu_{t+2|t} - g(\boldsymbol{z}_t;\boldsymbol{\theta};2)\right)^2 \tag{38}$$

$$+ \mathbb{E}_{\boldsymbol{Y}_T}\left[\left(g(\boldsymbol{z}_t;\hat{\boldsymbol{\theta}};2) - g(\boldsymbol{z}_t;\boldsymbol{\theta};2)\right)^2 \mid \boldsymbol{x}_t\right] \tag{39}$$

We will perform three pairwise comparisons of closely related strategies in terms of the bias and variance components: REC vs DIR , RECMULTI vs DIR and DIRJOINT vs DIR . For the first two comparisons, we will limit ourselves to horizon $h = 2$, for clarity and simplicity. Generally, similar arguments will apply to further horizons. For the last comparison, we consider a general horizon h.

A. One-step recursive and direct strategies

For the recursive strategy (REC), we have

$$B_2^{\text{REC}}(\boldsymbol{x}_t) + V_2^{\text{REC}}(\boldsymbol{x}_t) \tag{40}$$

$$\approx \Bigg\{ \delta(f(\boldsymbol{x}_t),\ldots,y_{t-p+2};\boldsymbol{\theta}) + \delta(\boldsymbol{z}_t;\boldsymbol{\theta})m_{z_1} \tag{41}$$

$$+ \frac{1}{2}[\delta(\boldsymbol{z}_t;\boldsymbol{\theta})]^2 m_{z_1z_1} + \frac{1}{2}[\eta(\boldsymbol{z}_t;\boldsymbol{\theta})]^2 m_{z_1z_1} - \tfrac{1}{2}\sigma^2 f_{x_1x_1} \Bigg\}^2 \tag{42}$$

$$+ [\eta(f(\boldsymbol{x}_t),\ldots,y_{t-p+2};\boldsymbol{\theta})]^2 + [\eta(\boldsymbol{z}_t;\boldsymbol{\theta})m_{z_1}]^2 \tag{43}$$

$$+ \frac{1}{2}[\eta(\boldsymbol{z}_t;\boldsymbol{\theta})]^4 m_{z_1z_1}^2 \tag{44}$$

$$+ 2\eta(f(\boldsymbol{x}_t),\ldots,y_{t-p+2};\boldsymbol{\theta})\eta(\boldsymbol{z}_t;\boldsymbol{\theta})m_{z_1}\mathbb{E}[\varepsilon_{\eta_1}\varepsilon_{\eta_2}] \tag{45}$$

$$+ \eta(\boldsymbol{z}_t;\boldsymbol{\theta})^2\eta(f(\boldsymbol{x}_t),\ldots,y_{t-p+2};\boldsymbol{\theta})m_{z_1z_1}\mathbb{E}[\varepsilon_{\eta_1}^2\varepsilon_{\eta_2}] \tag{46}$$

where we used Eqs. (26)-(34), and the fact that $\mathbb{E}[\varepsilon_\eta^3] = 0$ and $\mathbb{E}[\varepsilon_\eta^4] = 3$ for the standard normal distribution. Note that the quantity inside the curly brackets represents the bias term.

For the direct strategy, we have

$$B_2^{\text{DIRECT}}(\boldsymbol{x}_t) + V_2^{\text{DIRECT}}(\boldsymbol{x}_t) \tag{47}$$

$$\approx \left[\mu_{t+2|t} - m_2(\boldsymbol{r}_t;\boldsymbol{\theta}_2)\right]^2 \tag{48}$$

$$+ \eta(\boldsymbol{r}_t;\boldsymbol{\theta}_2)^2 \tag{49}$$

By comparing the bias and variance terms for both strategies at $h = 2$, we observe the following:

- In contrast with DIR , the bias and variance components of REC at horizon $h = 1$ are affecting the bias and variance components at horizon $h = 2$. For DIR , although bias and variance components at $h = 2$ are dependent on $h = 1$ because of the temporal structure of the time series, the dependence is not explicit.
- Let us take the extreme case of very large training set. We can then assume that the model of the recursive strategy at horizon $h = 1$ has been almost perfectly estimated, so that $\delta(\cdot;\boldsymbol{\theta}) \approx 0$ and $\eta(\cdot;\boldsymbol{\theta}) \approx 0$. Then B_2^{REC} reduces to $\frac{1}{2}\sigma^2 f_{x_1x_1}$, and hence the bias has not been completely eliminated. For the direct strategy, the same assumption results in B_2^{DIR} being approximately zero.
- The bias for REC is amplified when the forecasting model produces a function that has large variations (i.e. m_{z1} and $m_{z_1z_1}$ are large in magnitude). It is well-known that complex models tend to give low bias but could get functions with large variations. In the situation of REC a complex model's originally low bias will get worse because of the large variations. Moreover, if the original bias is low to begin with, then the only direction for it is to increase in magnitude.
- Because of the quantity $\frac{1}{2}[\eta(\boldsymbol{z}_t;\boldsymbol{\theta})]^2 m_{z_1z_1}$ in the bias equation, a high variance (i.e. a high $\eta(\boldsymbol{z}_t;\boldsymbol{\theta})$) tends to make the bias worse, which is in some way paradoxic to the well-known conflicting relation of bias and variance.
- In comparison with DIR , REC generally tends to obtain a worse variance. This underperformance is also amplified for functions having large variations (i.e. m_{z_1} and $m_{z_1z_1}$ are large in magnitude).
- Even though REC seems to be inferior to DIR regarding the bias and variance performance, it does sometimes have some advantages. It reduces the problem of h-step ahead forecasting to an estimation of only one function f, thereby breaking up the problem into more manageable pieces. This could be advantageous for highly nonlinear time series, or when the stochastic error term grows with the horizon, drowning out the deterministic component and making it hard to forecast directly. Also, for very short time series, DIR has a comparatively smaller training set, thus penalizing its variance.

B. Multi-step recursive and direct strategies

By comparing the objectives of RECMULTI and DIR strategies, which are given in (5) and (7), we can see that they are both minimizing an h-step forecasting error.

RECMULTI and DIR reduce or avoid the error accumulation over the horizon compared to REC strategy, which minimizes a 1-step forecasting error. The difference between RECMULTI and DIR is that the former minimizes an h-step *recursive* error while the latter minimizes an h-step *direct* error.

After estimation, RECMULTI produces the forecasts similarly to REC , that is applying the estimated model recursively

h times. The only difference with REC is that a different parameter $\boldsymbol{\theta}^{(h)}$ will be used for each h-step forecast. DIR produces the forecasts directly using the estimated model.

Let us compare these two strategies at horizon h. For RECMULTI , we have

$$B_2^{\text{RECMULTI}}(\boldsymbol{x}_t) + V_2^{\text{RECMULTI}}(\boldsymbol{x}_t) \tag{50}$$

$$\approx \left[\mu_{t+2|t} - m^{(2)}(\boldsymbol{r}_t; \boldsymbol{\theta}^{(2)})\right]^2 \tag{51}$$

$$+ \eta(\boldsymbol{r}_t; \boldsymbol{\theta}^{(2)})^2 \tag{52}$$

where $\boldsymbol{\theta}^{(2)}$ is the set of parameter obtained by minimizing a 2-step recursive error. The same expression for DIR is given in (47).

Assuming the right input variables are given, optimality can be achieved by DIR with an infinitely large training set (thus wiping out the variance term) and a flexible learning model able to learn the function (thus cancelling the bias term). For RECMULTI we need to be able to find a model with a set of parameters that when applied recursively h times is able to estimate the function. In other words, RECMULTI incurs restrictions on the model that may somewhat limit its ability to fit the true DGP, leading to some bias.

With a finite but large dataset, we expect DIR to have a better bias and variance properties. In fact, with a large dataset, DIR will be able to correctly estimate the potential complex function at horizon h while RECMULTI can lose some performance because of the recursive application of the model. This analysis is also valid when comparing other direct-type strategies with recursive-type strategies, like for example DIRJOINT versus RECJOINT , or DIRJOINTL versus RECJOINTL .

However, with small datasets, DIR selects every model independently at each horizon, thus effectively decoupling the different horizon's forecasts. In reality the DGP dictates some structure that governs the evolution of the time series across the horizon. Decoupling the forecasts, as done in DIR will therefore lead to a relatively higher variance (for small data sets). This argument is also valid and even more pronounced when comparing between DIR and DIRJOINT as will be seen in the next section.

C. Single-horizon and multi-horizon strategies

To learn the model at horizon h, single-horizon strategies (SIN), such as DIR and RECMULTI , include only data samples for that horizon. Multi-horizon strategies (MUL) such as DIRJOINT and RECJOINT use additional samples from other horizons to learn the same model, thus increasing the effective size of the training set.

To show the advantage of multi-horizon strategies, we can use a measure such as the number of residuals per parameter (RPP), which can be defined as the number of residuals divided by the number of parameters. This is a measure of data adequacy, as it gauges the amount of data per tunable parameter. The variance should generally vary inversely with the RPP. Of course, the RPP will depend on the forecast horizon for the number of residuals and the learning algorithm for the number of parameters.

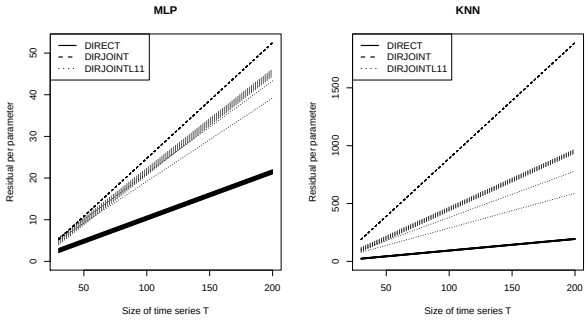

Fig. 1: Residuals per parameter for three strategies for different sizes of time series.

Let us consider a single-layer neural network (MLP) with p input nodes, u hidden units. At horizon h, the residual per parameters for single-horizon strategies is given by

$$\text{RPP}_h^{\text{SIN}} = \frac{T - p - h + 1}{(p+1) \times u + u + 1} \tag{53}$$

and for the multi-horizon strategies, by

$$\text{RPP}_h^{\text{MUL}} = \frac{|L_h|(T - p - max(L_h) + 1)}{(p+1) \times u + (u+1) \times |L_h|} \tag{54}$$

where L_h (defined in section II-C) is the set of horizons considered in the objective for horizon h.

If we consider a non-parametric learning algorithm, such as K-nearest neighbors (KNN), then the residual per parameters is simply equal to the number of residuals since we have only one hyper-parameter, K, the number of neighbors. For single-horizon strategies, the number of residuals per parameter is then given by

$$\text{RPP}_h^{\text{SIN}} = T - p - h + 1 \tag{55}$$

and for the multi-horizon strategies, by

$$\text{RPP}_h^{\text{MUL}} = |L_h|(T - p - max(L_h) + 1) \tag{56}$$

For illustration purposes, Figure 1 gives the number of residuals per parameter for strategies DIR , DIRJOINT and JOINTL (with $i = j = 2$) for all horizons in $\{1, \ldots, H = 10\}$. The size of time series T is ranging in $[50, 200]$ and $p = 2$. On the left, the results are shown for MLP with $u = 2$ and, on the right, for KNN.

From Figure 1, we can see that for both NN and KNN, $\text{RPP}_h^{\text{MUL}} > \text{RPP}_h^{\text{SIN}}$. In others words, multi-horizon strategies have a higher RPP compared to single-horizon strategies. In addition, we can also see that the difference for the case of KNN is much higher than the case of MLP, mainly because of the larger number of parameters to estimate. Therefore, KNN may benefit more from multi-horizon strategies over single-horizon strategies compared to MLP.

The increase of RPP can be particularly useful since by using more observations, the variance term at each horizon can

be decreased, and this is particularly effective for small sample time series. However, it comes with a drawback due to the fact that one set of parameters is used to estimate jointly several functions. This can reduce the flexibility of the forecasting strategy, which may in turn increase the bias.

So, to obtain better forecast with multi-horizon strategies compared to single-horizon strategies at horizon h, the decrease in variance must be higher than the increase in bias. In other terms, multi-horizon strategies will beat single-horizon strategies at horizon h (in an MSE-sense) if

$$\underbrace{V_h^{\texttt{SIN}}(\boldsymbol{x}_t) - V_h^{\texttt{MUL}}\ (\boldsymbol{x}_t)}_{\text{Decrease in variance}} > \underbrace{[B_h^{\texttt{MUL}}\ (\boldsymbol{x}_t)]^2 - [B_h^{\texttt{SIN}}(\boldsymbol{x}_t)]^2}_{\text{Increase in bias}} \quad (57)$$

V. Bias and variance analysis by simulation

We carry out a Monte Carlo study to investigate the performance of the different multi-step-ahead forecasting strategies from the perspective of bias and variance.

Because we do not know the ground truth for the case of real-world time series, simulated data with a controlled noise component is the only way to accurately and effectively measure bias and variance effects.

A. Data generating processes

We consider three data generation processes (DGP's) in the presented simulation study, described as follows:

- The smooth transition autoregressive (STAR) process

$$\begin{aligned} y_t &= 0.3y_{t-1} + 0.6y_{t-2} \\ &\quad + (0.1 - 0.9y_{t-1} + 0.8y_{t-2})[1 + e^{-10y_{t-1}}]^{-1} + \varepsilon_t \end{aligned} \quad (58)$$

where ϵ_t is independently and identically distributed (i.i.d.) $\mathcal{N}(0, \sigma^2)$, with $\sigma = 0.01$. This particular STAR process has been used in several other published simulations studies notably for the purposes of model selection, model evaluation and model comparison. Theoretical background and applications of the STAR process are given in references [3] and [2].

- The nonlinear autoregressive (NLAR) process

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

with

$$G_1(\boldsymbol{y}_{t-1}) = (1 + \exp\{-0.46(0.29y_{t-1} - 0.87y_{t-2} + 0.40y_{t-7} - 6.68)\})^{-1}$$
$$G_2(\boldsymbol{y}_{t-1}) = (1 + \exp\{-1.17 \times 10^3(0.83y_{t-1} - 0.53y_{t-2} - 0.18y_{t-7} + 0.38)\})^{-1}$$

where ϵ_t is independently and identically distributed (i.i.d.) $\mathcal{N}(0, 1)$. We set the error variance to a value which assures a reasonable amount of predictability for the time series. [51] build this process by fitting an artificial neural network with two hidden units to the annual sunspot series. In [8], this process has been used to compare different forecasting methods in a nonlinear setting.

- The linear AR (LAR) process

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

where ϵ_t is independently and identically distributed (i.i.d.) $\mathcal{N}(0, 1)$. This process exhibits cyclic behavior 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. 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.

B. Forecasting strategies and learning algorithms

Table I gives a list of the different strategies that we will consider in the analysis. These are a variety of strategies that belong to the single-horizon or the multi-horizon categories, and are either of a recursive type or the direct type.

TABLE I: List of strategies considered in the simulations. Each strategy is either a single-horizon or a multi-horizon strategy and is either recursive or direct.

	Single-horizon	Multi-horizon
Recursive	`REC` `RECMULTI(RTI)`	`RECJOINT(RJT)` `RECJOINTL(RJTL)`
Direct	`DIR`	`DIRJOINT(DJT)` `DIRJOINTL(DJTL)`

For both `RECJOINTL` and `DIRJOINTL` note that we used $i = j = 1$ (see (II-C)). In other words, at each horizon h, we used data samples from the previous and the next horizon, thus using a total of 2 additional horizons. These two strategies will be denoted `RECJOINTL11(RJTL11)` and `DIRJOINTL11(DJTL11)`.

The goal of this research is not to make a comparison of machine learning algorithms for forecasting (which has been already conducted in [4]), but rather to analyze the behavior of bias and variance for different multi-step strategies using machine learning models. The machine learning models considered in this study are the K-Nearest Neighbor (KNN) and the multilayer perceptron (MLP).

The KNN model is frequently considered as a benchmark model in the machine learning community, and has proven to be an effective and robust forecasting model [7], [33], [45]. The KNN is a nonparametric model where the prediction for a given data point x_q is obtained by averaging the target outputs $y_{[i]}$ of the K training data points in the vicinity of the given point x_q [52]. We used a weighted KNN model, whereby a *weighted* average rather than a simple average is used. The used weights are a function of the Euclidean distance between the query point and the neighboring point (we used the biweight function [52]).[1]

MLP (also called neural networks) is one of the most successful machine learning algorithm in time series forecasting [4], [10], [34]. We considered the standard feedforward neural network with one-hidden layer. The MLP is given as follows

$$\hat{y} = \alpha_0 + \sum_{j=1}^{NH} \alpha_j g(\boldsymbol{w}_j^T \boldsymbol{x}') \quad (59)$$

where x' is the input vector $\boldsymbol{x}$, augmented with 1, i.e., $x' = (1, \boldsymbol{x}^T)^T$, $\boldsymbol{w}_j$ is the weight vector for jth hidden node, α_0, α_1,

$\ldots, \alpha_n$ are the weights for the output node, NH is the number of hidden nodes, and $\hat{y}$ is the prediction at the output of the network. We used the logistic function for the function g. For training the network we used the BFGS method implemented in the `optim` package which is used by the `nnet` package in the R programming language.

For each of the two models there are one or two parameters that controls the complexity of the model, and therefore it has to be selected with care. For the KNN model the number of neighbors K is this key parameter. A large K will lead to a smoother fit, and therefore a lower variance, of course at the expense of a higher bias, and vice versa for a small K. For MLP it is the number of hidden nodes NH that controls its complexity. Weight decay, another but less influential parameter, adds some kind of regularization to the network. In addition to these hyperparameters, the number of lagged values p is a critical parameter that should be carefully selected. We used a 5-fold validation approach with two nested loops, one for p and the other for the other hyperparameters. For MLP, we considered an additional loop for the weight decay parameter. For KNN, the tested values of K are in the range $[2, N]$ where N is the size of the dataset. For MLP, the tested possible values of NH are as follows $\{0, 1, 2, 3, 5\}$ (where 0 means no hidden neurons, i.e. effectively a linear network). The weight decay λ's possible values are from the following choices $\{0.005, 0.01, 0.05, 0.1, 0.2, 0.3\}$.

C. Bias and variance estimation

As given by expression (15), we have seen that the MSE can be decomposed into the three different components: noise, squared bias and variance. We will estimate, for each strategy, the bias and the variance terms, B_h and V_h, defined in (15), by replacing expectations with averages.

For a given DGP, we generate L independent time series $\boldsymbol{Y}_T^{(i)} = \{y_1^{(i)}, \ldots, y_T^{(i)}\}$, $i \in \{1, \ldots, L\}$, each composed of T observations using different randomly generated numbers for the noise terms. These generated time series represent samples of the DGP.

In addition, we generate another independent time series from the true DGP for the testing purpose. From this time series we extract R input/output pairs $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{R}$, where $\boldsymbol{x}_j$ is the temporal pattern of length d that will yield the lagged input variables, and the vector $\boldsymbol{y}_j$ is the subsequent pattern of length H that needs to be forecasted, e.g. if u_j is the test time series, then $\boldsymbol{x}_j = [u_j, \ldots, u_{j+d-1}]$ and $\boldsymbol{y}_j = [u_{j+d}, \ldots, u_{j+d+H-1}]$.

Let $g(\boldsymbol{x}_j; \boldsymbol{\theta}_{\boldsymbol{Y}^{(i)}}; h)$ denote the forecast of a given strategy for the input $\boldsymbol{x}_j$ at horizon h using dataset $\boldsymbol{Y}^{(i)}$ (T is omitted for simplicity) and let $\boldsymbol{y}_j(h)$ denote the hth element of the vector $\boldsymbol{y}_j$, then the MSE at horizon h, given in expression (15), can be estimated as

$$\widehat{MSE}_h = \frac{1}{R}\sum_{j=1}^{R} \underbrace{\frac{1}{L}\sum_{i=1}^{L} \left(\boldsymbol{y}_j(h) - g(\boldsymbol{x}_j; \boldsymbol{\theta}_{\boldsymbol{Y}^{(i)}}; h)\right)^2}_{\widehat{\text{MSE}}_h(\boldsymbol{x}_t)} \tag{60}$$

$$= \frac{1}{R}\sum_{j=1}^{R} \left[\hat{N}_h(\boldsymbol{x}_j) + \hat{B}_h(\boldsymbol{x}_j)^2 + \hat{V}_h(\boldsymbol{x}_j)^2\right] \tag{61}$$

$$= \underbrace{\frac{1}{R}\sum_{j=1}^{R} \hat{N}_h(\boldsymbol{x}_j)}_{\hat{N}_h} + \underbrace{\frac{1}{R}\sum_{j=1}^{R} \hat{B}_h(\boldsymbol{x}_j)^2}_{\hat{B}_h} + \underbrace{\frac{1}{R}\sum_{j=1}^{R} \hat{V}_h(\boldsymbol{x}_j)^2}_{\hat{V}_h} \tag{62}$$

with

$$\hat{N}_h(\boldsymbol{x}_j) = \left(\boldsymbol{y}_j(h) - \text{Avg}[\boldsymbol{y}_j(h) \mid \boldsymbol{x}_j]\right)^2,$$
$$\hat{B}_h(\boldsymbol{x}_j)^2 = \left(\text{Avg}[\boldsymbol{y}_j(h) \mid \boldsymbol{x}_j] - \bar{g}(\boldsymbol{x}_j; \boldsymbol{\theta}; h)\right)^2,$$
$$\hat{V}_h(\boldsymbol{x}_j)^2 = \frac{1}{L}\sum_{i=1}^{L} \left[g(\boldsymbol{x}_j; \boldsymbol{\theta}_{\boldsymbol{Y}^{(i)}}; h) - \bar{g}(\boldsymbol{x}_j; \boldsymbol{\theta}; h)\right]^2,$$

where $\bar{g}(\boldsymbol{x}_j; \boldsymbol{\theta}; h) = \frac{1}{L}\sum_{i=1}^{L} g(\boldsymbol{x}_j; \boldsymbol{\theta}_{\boldsymbol{Y}^{(i)}}; h)$.

We will explain here the term $\text{Avg}[\boldsymbol{y}_j(h) \mid \boldsymbol{x}_j]$. For every $\boldsymbol{x}_j$ there are different possible subsequent patterns $\boldsymbol{y}_j$ that depend on the realization of the error term. These variations account for the noise term $N_h(\boldsymbol{x}_j)$ in the bias and variance decomposition. To compute the aforementioned average we generate S different realizations of $y_s^{(h)}(\boldsymbol{x}_j)$ given a fixed starting vector $\boldsymbol{x}_j$. Then we evaluate $\text{Avg}[\boldsymbol{y}_j(h) \mid \boldsymbol{x}_j] = \frac{1}{S}\sum_{s=1}^{S} y_s^{(h)}(\boldsymbol{x}_j)$.

In the simulations, the first three hundred simulated values were discarded for each simulated series to stabilize the time series, as suggested by [53]. To investigate the effect of the size of the time series T for each strategy, we compare different sizes, namely $T = [50, 100, 400]$. We took $L = 2000$ time series for training. We used a long testing time series, and extracted from it $R = 2000$ testing pairs. The number of realizations for computing the noise term S is taken as 30,000.

VI. Results and discussion

Figures 2-4 gives the MSE decomposition for the NLAR DGP. Figures 5-7, for the STAR DGP. And Figures 8-10, for the LAR DGP.

Figures 2-10 have three rows and four columns. Each row corresponds to a length of time series $T = [50, 100, 400]$. For the four columns, we have the MSE (first column), the squared bias (second column), the variance (third column) and the squared bias plus variance (fourth column). In the first column, which represents the MSE, the bias and variance components are substantially masked by the noise term, making comparisons between the strategies difficult. Consequently, we consider the three other columns for the purpose of comparing the strategies, and use the MSE as a measure of the predictability of the time series relative to the mean (the red line). In the first column, the grey line represents the noise component $\hat{N}_h$ defined in (62).

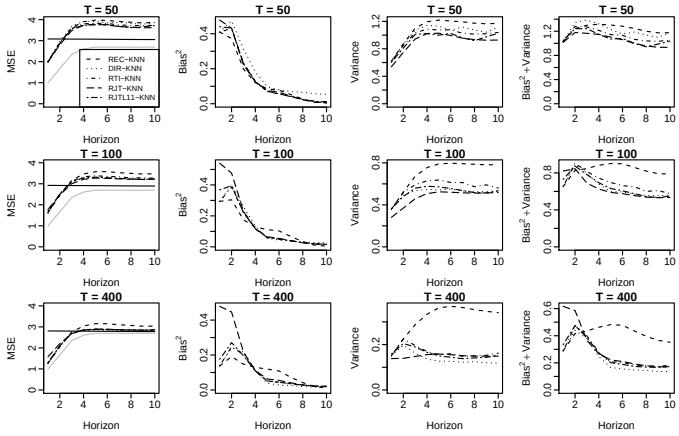

Fig. 2: MSE decomposition for the NLAR DGP with KNN (I/II).

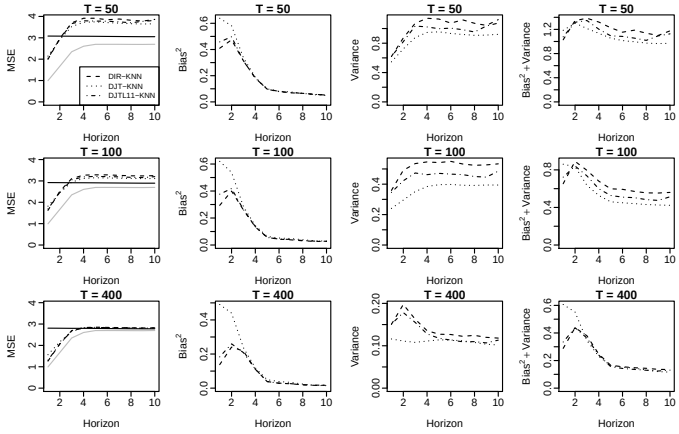

Fig. 3: MSE decomposition for the NLAR DGP with KNN (II/II).

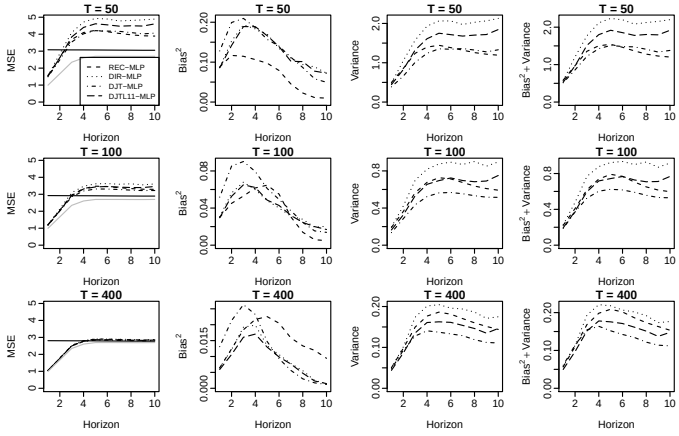

Fig. 4: MSE decomposition for the NLAR DGP with MLP.

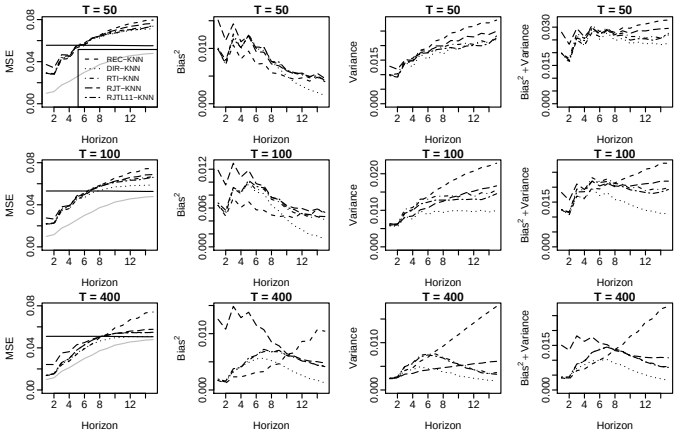

Fig. 5: MSE decomposition for the STAR DGP with KNN (I/II).

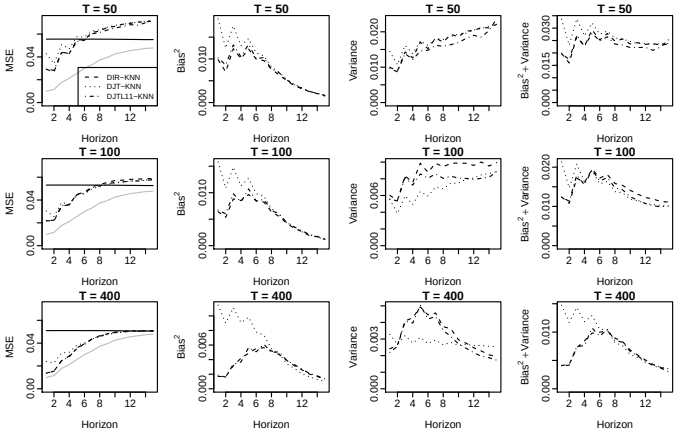

Fig. 6: MSE decomposition for the STAR DGP with KNN (II/II).

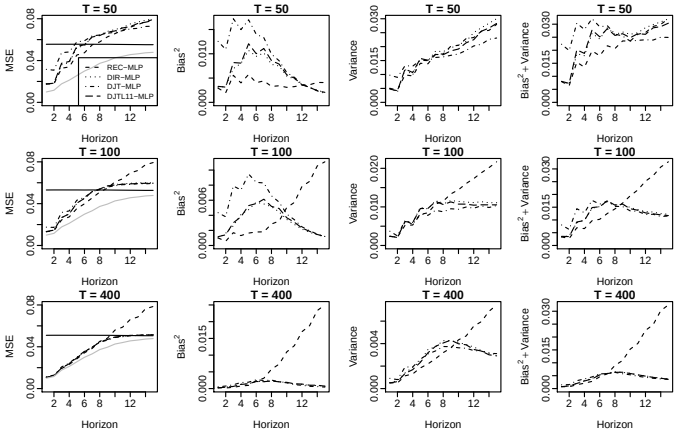

Fig. 7: MSE decomposition for the STAR DGP with MLP.

Let us analyze the performance of each of the methods. We will consider both the *absolute performance*, and the *relative performance* (i.e. the performance compared to the other strategies), with more focus on the latter. When we say "relative performance improves", we do not mean that in the absolute, but rather that the standing of the method compared to others becomes better. Here are our general observations concerning the following compared strategies.

The one-step recursive (`REC`) strategy: It has an erratic performance. Generally it gives the highest variance, while the results for the bias are mixed (sometimes it is bad and sometimes it is the best). For most of the DGP's the variance increases significantly with the horizon as can be seen in the third column of Figures 2, 5 and 8. This is due to the

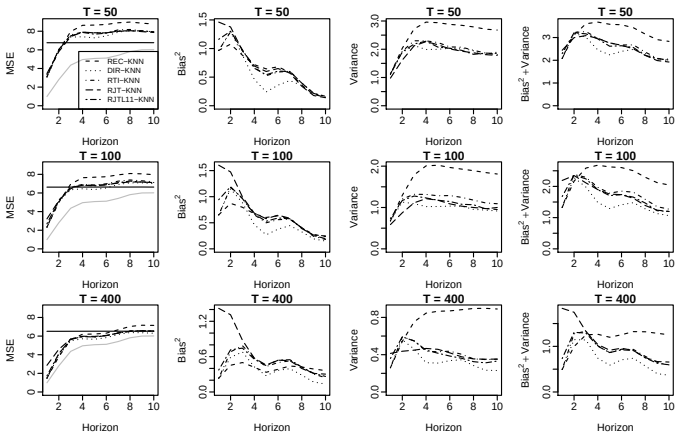

Fig. 8: MSE decomposition for the LAR DGP with KNN (I/II).

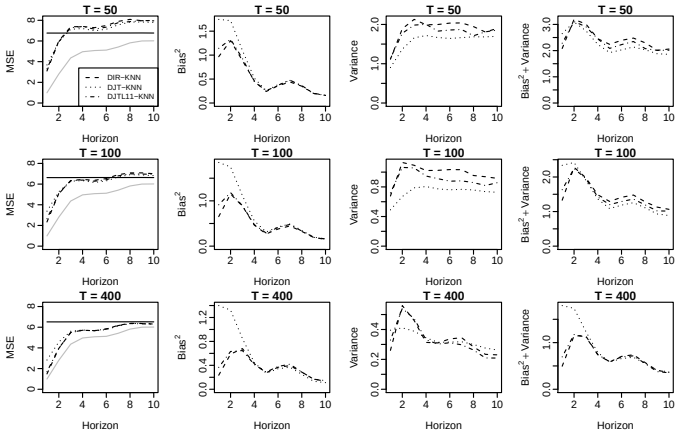

Fig. 9: MSE decomposition for the LAR DGP with KNN (II/II).

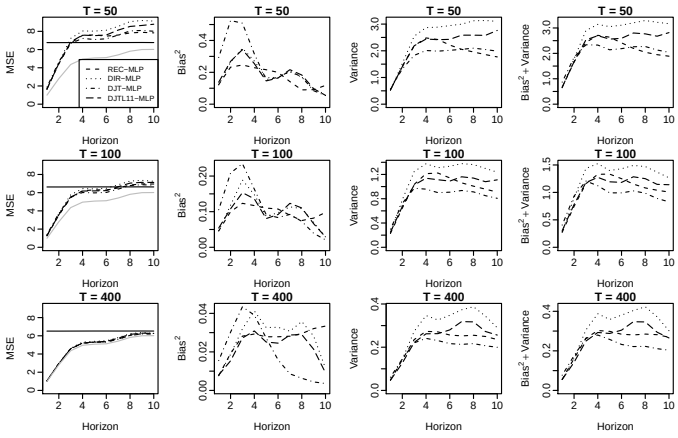

Fig. 10: MSE decomposition for the LAR DGP with MLP.

additive or accumulative effect of forecast error in the absence of a corrective mechanism. Also, the relative performance (compared to other strategies) for the bias and the variance almost always gets worse for longer time series (except for the LAR DGP) as can be seen, for example, in the last column of Figure 2. This illuminates the fact that REC 's advantage is in shorter time series. The reason is that for other strategies the effective training set size is smaller, typically by an amount $h-1$ when forecasting h-step ahead, due to the lack of use of the end portion of the time series.

In Figures 4 and 10, we can also see that REC which uses MLP as learning model has an advantage for the NLAR and LAR DGP's. This is because the former's DGP is equivalent to a one-layer MLP (so iterating the forecasts would still be consistent with the DGP), and because the latter is linear, so it is encompassed by the MLP model. This also lessens the effect we discussed in Subsection IV-A that nonlinearities worsen the performance of REC , because in that case we are effectively dealing with a linear model.

The direct (DIR) strategy: From the experiments, its distinctive feature is that it consistently obtains a better relative performance for the bias and the variance with increasing time series length. At $T = 400$ it generally becomes the leading strategy for the bias. The reason for underperformance for small time series is that DIR has a smaller number of effective training data points compared to REC , because of edge effects (the size of the training set is less by $h-1$ when forecasting h-step ahead), and this becomes a relatively bigger problem for small time series.

The results for the variance are not very good for the case of MLP model as can be seen in Figures 4 and 10. The reason why the variance of DIR often lags some other strategies, such as DIRJOINT is the following. Each horizon h is forecasted in isolation of the other horizons. So the strategy could produce completely unrelated forecasts over the whole horizon, possibly leading to unrealistic discontinuities. In reality, however, time series have some aspects of continuous behavior. It is this aspect that DIRJOINT exploits to some extent to make up for smaller time series lengths, and control overfitting. This is also accentuated for DIR with MLP, since MLP is a highly complex model, and therefore more vulnerable to a variance deterioration effect. This described fact also explains why for long time series DIR make a comeback: it makes use of the larger amount of data points, and that helps overcome the effect of the large variance. These arguments are also consistent with the analysis that we presented in Subsection IV-A.

The multi-step recursive (RECMULTI) strategy: As can be seen in Figures 2, 5 and 8, RECMULTI is a better alternative to the REC strategy for long-term horizons. At short-term horizons, its performance is very close to REC since they have a closer objective. Overall, the gain with RECMULTI compared to REC is in terms of variance. This can be explained by the multi-step-ahead objective used by RECMULTI which limits the propagation of errors. RECMULTI has close or better performance to DIR with short time series but as the length of the time series increases, DIR becomes the leader. Recall that one particularity of RECMULTI is that the same model (structure) is used for all horizons but different values of the model parameters are allowed at each horizon. Overall, RECMULTI is in the middle of the pack, for both the bias and the variance. It is never a prominent and leading strategy, but at the same time never gets very bad performance.

The multi-horizon strategies: let us start with DIRJOINT and RECJOINT . These two strategies have similar behavior, and have many common features. Generally, they obtain a smaller variance than other strategies (see, for example, Figures 5 and 6). However, their relative performance (also in terms of variance) suffers a little for longer time series. The negative aspect of these two strategies is that they possess a very high bias for small horizons (see, for example, Figures 4 and 7). This can be explained by observing the objective function of DIRJOINT in the expression (11). In fact, DIRJOINT and RECJOINT select the parameters that give minimum mean square forecast error over the whole horizon. But these errors are of a different scale. Errors at longer horizons are generally larger than errors at short term horizons, so they dominate the error function. So DIRJOINT and RECJOINT are implicitly putting more weight on these longer horizons. As a result, the final estimated set of parameters give a large bias for the first few horizons. For large horizons the bias improves considerably, and becomes comparable to the other strategies. If we compare DIRJOINT and RECJOINT , we observe that they generally give comparable performance for the variance, but DIRJOINT is generally better for the bias. The reason is that for longer horizons RECJOINT dictates an iterated function formulation for the fit. This may not be flexible enough, and would produce a bias. On the other hand, DIRJOINT does not have this constraint, and uses the full power of the forecasting model to directly fit the h-step ahead function.

If we now compare DIRJOINTL11 and RECJOINTL11 , we see that they have very good and robust performance. The effect of the high bias that we encounter for the DIRJOINT and the RECJOINT strategies is less pronounced, and quite acceptable. DIRJOINTL11 is somewhat better than RECJOINTL11 for the case of the bias, and comparable for the case of the variance. The reason is similar to the case of DIRJOINT and RECJOINT .

VII. Summary and conclusion

In this work we have presented a comprehensive investigation into the bias and variance behavior of multi-step-ahead forecasting strategies. The bias and variance are central quantities that can give considerable insight into model performance. We considered the three major multi-step strategies, the recursive strategy, the direct strategy, and the joint strategy, along with other variants and combination strategies. We applied some theoretical analysis that investigates the bias and variance behavior. We also applied a detailed simulation study, that analyzes some of the effects of different factors, such as time series length, forecasting horizon and learning model, on the bias and variance. We could observe that the simulation study has confirmed many of the findings of the theoretical study.

A short digest of the findings is that REC holds advantage for short time series and for cases when there is reason to believe we have a well-specified forecasting model. DIR is superior for the case of long time series or if we need a particularly small bias, like in some applications where there is a focus on directional forecasting. RECMULTI can be useful if we want to produce recursive forecasts with the same model but at the same time allow more flexibility for the forecast function over the horizon. DIRJOINT and RECJOINT are particularly attractive for short time series and long-term horizons because of their better variance behavior. DIRJOINTL and RECJOINTL give balanced and robust performance for both bias and variance. As such, they are some of the highly recommended models because of their good performance for most conditions.

We hope this work would lead to a much more reliable selection of which strategy to use, since it would be grounded on qualitative arguments, theory, and simulations. Also, this work could help identify how to overcome some of the weaknesses of the different strategies, possibly through some modifications, since the sources of the weaknesses are to some extent pinned down.

References

[1] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[2] T. Teräsvirta, D. Tjø stheim, and C. W. J. Granger, *Modelling Nonlinear Economic Time Series*, ser. Advanced Texts in Econometrics. OUP Oxford, 2010.

[3] J. Fan and Q. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*. New York: Springer, 2005.

[4] N. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5, pp. 594–621, Sep. 2010.

[5] L. Breiman, "Statistical modeling: The two cultures," *Statistical Science*, pp. 199–215, 2001.

[6] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16-18, pp. 2861–2869, Oct. 2007.

[7] J. D. Wichard, "Forecasting the NN5 time series with hybrid models," *International Journal of Forecasting*, vol. 27, no. 3, pp. 700–707, May 2010.

[8] A. Kock and T. Teräsvirta, "Forecasting with nonlinear time series models," in *Oxford Handbook of Economic Forecasting*, M. P. Clements and D. F. Hendry, Eds., 2011, pp. 61–87.

[9] R. R. Andrawis, A. F. A. Atiya, and H. El-Shishiny, "Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition," *International Journal of Forecasting*, vol. 27, no. 3, pp. 1–29, 2011.

[10] G. Zhang and D. Kline, "Quarterly time-series forecasting with neural networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1800–1814, 2007.

[11] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[12] S. Ben Taieb and R. Hyndman, "A gradient boosting approach to the Kaggle load forecasting competition," *International Journal of Forecasting*, vol. 30, no. 2, pp. 382–394, Aug. 2014.

[13] M. Pulido, P. Melin, and O. Castillo, "Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange," *Information Sciences*, vol. 280, pp. 188–204, Oct. 2014.

[14] F. Gaxiola, P. Melin, F. Valdez, and O. Castillo, "Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction," *Information Sciences*, vol. 260, pp. 1–14, Mar. 2014.

[15] O. Castillo, J. R. Castro, P. Melin, and A. Rodriguez-Diaz, "Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction," *Soft Computing*, vol. 18, no. 6, pp. 1213–1224, Oct. 2014.

[16] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*. New York: Cambridge University Press, 2004.

[17] A. Weiss, "Multi-step estimation and forecasting in dynamic models," *Journal of Econometrics*, vol. 48, pp. 135–149, 1991.

[18] G. C. Tiao and R. S. Tsay, "Some advances in non-linear and adaptive modelling in time-series," *Journal of forecasting*, vol. 13, no. 2, pp. 109–131, 1994.

[19] R. Bhansali, "MultiStep Forecasting," in *A Companion to Economic Forecasting*, M. P. Clements and D. F. Hendry, Eds., 2004, no. 1.

[20] C.-K. Ing, "Multistep prediction in autoregressive processes," *Econometric Theory*, pp. 254–279, 2003.

[21] ——, "Selecting optimal multistep predictors for autoregressive processes of unknown order," *The Annals of Statistics*, vol. 32, no. 2, pp. 693–722, Apr. 2004.

[22] G. Chevillon and D. F. Hendry, "Non-parametric direct multi-step estimation for forecasting economic processes," *International Journal of Forecasting*, vol. 21, no. 2, pp. 201–218, Apr. 2005.

[23] M. Marcellino, J. Stock, and M. Watson, "A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series," *Journal of Econometrics*, vol. 135, no. 1-2, pp. 499–526, Nov. 2006.

[24] G. Chevillon, "direct multi-step estimation and forecasting," *Journal of Economic Surveys*, vol. 21, no. 4, pp. 746–785, 2007.

[25] P. V. Kabaila, "Estimation based on one step ahead prediction versus estimation based on multi-step ahead prediction," *Stochastics*, vol. 6, no. 1, pp. 43–55, 1981.

[26] A. F. Atiya, S. M. El-shoura, S. I. Shaheen, and M. S. El-sherif, "A comparison between neural-network forecasting techniques–case study: river flow forecasting." *Neural Networks, IEEE Transactions on*, vol. 10, no. 2, pp. 402–409, 1999.

[27] T. Teräsvirta, "Forecasting economic variables with nonlinear models," *Handbook of economic forecasting*, pp. 413–457, 2006.

[28] Y. Bao, T. Xiong, and Z. Hu, "Multi-step-ahead time series prediction using multiple-output support vector regression," *Neurocomputing*, vol. 129, pp. 482–493, Apr. 2014.

[29] C. Hamzaçebi, D. Akay, and F. Kutay, "Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting," *Expert Systems with Applications*, vol. 36, no. 2,, pp. 3839–3844, 2009.

[30] R. Boné and M. Crucianu, "Multi-step-ahead prediction with neural networks: a review," in *In 9emes rencontres internationales: Approches Connexionnistes en Sciences*, 2002, pp. 97–106.

[31] F.-J. Chang, Y.-M. Chiang, and L.-C. Chang, "Multi-step-ahead neural networks for flood forecasting," *Hydrological Sciences Journal*, vol. 52, no. 1, pp. 114–130, Feb. 2007.

[32] E. Ghysels, R. I. Valkanov, and A. R. Serrano, "Multi-period forecasts of volatility: direct, iterated, and mixed-data approaches," in *EFA 2009 Bergen Meetings Paper*, 2009.

[33] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[34] V. L. Berardi and G. P. Zhang, "An empirical investigation of bias and variance in time series forecasting: modeling considerations and error evaluation." *Neural Networks, IEEE Transactions on*, vol. 14, no. 3, pp. 668–79, Jan. 2003.

[35] B. E. Hansen, "Multi-step forecast model selection," in *20th Annual Meetings of the Midwest Econometrics Group*, 2010.

[36] M. Clements and D. Hendry, *Forecasting Economic Time Series*, ser. Forecasting economic time series. Cambridge University Press, 1998.

[37] A. Sorjamaa and A. Lendasse, "Time series prediction using dirrec strategy," in *ESANN, European Symposium on Artificial Neural Networks, European Symposium on Artificial Neural Networks*. Citeseer, Apr. 2006, pp. 143–148.

[38] X. Zhang and J. Hutchinson, "Simple architectures on fast machines: practical issues in nonlinear time series prediction," in *Time Series Prediction Forecasting the Future and Understanding the Past*, A. S. Weigend and N. A. Gershenfeld, Eds., Santa Fe Institute. Addison-Wesley, 1994, pp. 219–241.

[39] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, "Iterated time series prediction with multiple support vector regression models," *Neurocomputing*, vol. 99, pp. 411–422, Jan. 2013.

[40] S. Ben Taieb and R. J. Hyndman, "Recursive and direct multi-step forecasting: the best of both worlds," 2012.

[41] M. Solnon, S. Arlot, and F. Bach, "Multi-task regression using minimal penalties," *Journal of Machine Learning Research*, vol. 13, pp. 2773–2812, 2012.

[42] L. Breiman and J. Friedman, "Predicting multivariate responses in multiple linear regression," *Journal of the Royal Statistical Society B*, vol. 59, no. 1, pp. 3–54, 1997.

[43] G. Bontempi and S. Ben Taieb, "Conditionally dependent strategies for multiple-step-ahead prediction in local learning," *International Journal of Forecasting*, vol. 27, no. 3, pp. 689–699, Jan. 2011.

[44] D. M. Kline, "Methods for multi-step time series forecasting with neural networks," in *Neural Networks in Business Forecasting*, G. P. Zhang, Ed. Information Science Publishing, 2004, pp. 226–250.

[45] S. Ben Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting," *Neurocomputing*, vol. 73, no. 10-12, pp. 1950–1957, 2010.

[46] F. Jin and S. Sun, "Neural network multitask learning for traffic flow forecasting," in *IEEE International Joint Conference on Neural Networks*, Jun. 2008, pp. 1897–1901.

[47] A. G. Parlos, O. T. Rais, and A. F. Atiya, "Multi-step-ahead prediction using dynamic recurrent neural networks." *Neural Networks*, vol. 13, no. 7, pp. 765–86, Sep. 2000.

[48] Y. Xia and H. Tong, "Feature Matching in Time Series Modeling," *Statistical Science*, vol. 26, no. 1, pp. 21–46, Feb. 2011.

[49] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[50] S. G. Makridakis and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.

[51] M. C. Medeiros, T. Teräsvirta, and G. Rech, "Building neural network models for time series: a statistical approach," *Journal of Forecasting*, vol. 25, no. 1, pp. 49–75, Jan. 2006.

[52] C. Atkeson, A. Moore, and S. Schaal, "Locally weighted learning," *Artificial intelligence review*, vol. 11, no. 1, pp. 11–73, 1997.

[53] A. M. Law and W. D. Kelton, *Simulation Modelling and Analysis*, 3rd ed. McGraw-Hill, 2000.