Long-Term Prediction of Time Series by combining Direct and MIMO Strategies

Souhaib Ben Taieb, Gianluca Bontempi, Antti Sorjamaa and Amaury Lendasse

***Abstract*— Reliable and accurate prediction of time series over large future horizons has become the new frontier of the forecasting discipline. Current approaches to long-term time series forecasting rely either on iterated predictors, direct predictors or, more recently, on the Multi-Input Multi-Output (MIMO) predictors. The iterated approach suffers from the accumulation of errors, the Direct strategy makes a conditional independence assumption, which does not necessarily preserve the stochastic properties of the time series, while the MIMO technique is limited by the reduced flexibility of the predictor. The paper compares the Direct and MIMO strategy and discusses their respective limitations to the problem of long-term time series prediction. It also proposes a new methodology that is a sort of intermediate way between the Direct and the MIMO technique. The paper presents the results obtained with the ESTSP 2007 competition dataset.**

I. Introduction

The prediction of the future behavior of an observed time series over a long horizon H is still an open problem in forecasting [1]. Currently, the most common approaches to long-term forecasting rely either on iterated [2] or direct prediction techniques [3].

Given a time series $\{\varphi_1, \ldots, \varphi_t\}$, in the first case, an H-step-ahead prediction problem is tackled by iterating, H times, a one-step-ahead predictor. Once the predictor has estimated the future series value, this value is fed back as an input to the following prediction. Hence, the predictor takes as inputs estimated values, instead of actual observations with evident negative consequences in terms of error propagation. Examples of iterated approaches are recurrent neural networks [4] or local learning iterated techniques [5], [6].

Another way to perform H-step-ahead forecasting consists in estimating a set of H prediction models, each returning a direct forecast of φ_{t+h} with $h \in \{1, \ldots, H\}$ [3]. Direct methods often require higher functional complexity than iterated ones in order to model the stochastic dependency between two series values at two distant instants.

In computational intelligence literature, we have also examples of research works, where the two approaches have been successfully combined [7].

In spite of their diversity, iterated and direct techniques for multi-step-ahead prediction share a common feature, in the sense that they model from data, a multi-input single-output mapping, whose output is the variable φ_{t+1} in the iterated case and the variable φ_{t+h} in the direct case.

In [8], the author proposed a MIMO approach for long-term time series prediction, where the predicted value is no more a scalar quantity but a vector of future values $\{\varphi_{t+1}, \ldots, \varphi_{t+H}\}$ of the time series φ. This approach replaces the H models of the Direct approach with one multi-output model, which aims to preserve, between the predicted values, the stochastic dependency characterizing the time series.

This paper goes one step further in that direction by proposing a new methodology for long-term prediction, aiming to preserve the most appealing aspects of both the Direct and MIMO approaches. The Direct and the MIMO approach can be indeed seen as two distinct instances of the same prediction approach, which decomposes long-term prediction into multi-output tasks. In the direct case, the number of prediction tasks is equal to the size of the horizon H and the size of the outputs is 1. In the MIMO case, the number of prediction tasks is equal to one and the size of the output is H. Intermediate configurations can be imagined by transforming the original task into $n = \frac{H}{s}$ prediction tasks, each with multiple outputs of size s, where $s \in \{1, \ldots, H\}$. This approach, called Multi-Input Several Multi-Output (MISMO), trades off the property of preserving the stochastic dependency between future values with a greater flexibility of the predictor. For instance, the fact of having $n > 1$ allows the selection of different inputs for different horizons. In other terms, s addresses the bias/variance trade-off of the predictor by constraining the degree of dependency between predictions (null in the case $s = 1$ and maximal for $s = H$).

This paper introduces and assesses the MISMO methodology by implementing each prediction model with a Lazy Learning model [9], [10]. Lazy Learning (LL) is a local modeling technique, which is *query-based* in the sense that the whole learning procedure (i.e. structural and parametric identification) is deferred until a prediction is required. [11], [12] presented a LL algorithm that selects on a query-by-query basis and by means of a local cross-validation scheme the optimal number of neighbors. Iterated versions of Lazy Learning were successfully applied to multi-step-ahead time series prediction [13], [14]. A LL-MIMO version was presented in [8]. This paper presents a LL method for the MISMO prediction of multiple and dependent outputs in the context of long-term time series prediction.

Section II introduces the new MISMO strategy with respect to existing approaches to long-term prediction. Sec-

Gianluca Bontempi and Souhaib Ben Taieb are with the Machine Learning Group, Computer Science Department, Faculty of Sciences, Université Libre de Bruxelles (email: {gbonte, sbentaie}@ulb.ac.be).

Antti Sorjamaa and Amaury Lendasse are with the Department of Information and Computer Science, Helsinki University of Technology, Finland (email: {lendasse, Antti.Sorjamaa}@hut.fi).

tion III details the MISMO methodology. First, the Lazy Learning model and the validation procedure is explained. Next, the way to choose the parameter s of the MISMO model is presented. Then, the input selection procedure for the MISMO model is explained. Section IV presents the MISMO prediction results obtained with a series proposed by the ESTSP 2007 competition [15], [16].

II. Long-term Prediction techniques

This section first introduces the Direct and the MIMO strategies and then discusses how the original MISMO approach situates with respect to the state of the art.

Let us consider a regular and univariate time series $\{\varphi_1, \varphi_2, \ldots, \varphi_t\}$ for which we intend to predict the continuation for the next H steps. Suppose that we can embed the time series into an input-output format and that an input selection strategy is adopted. Suppose also that the maximum embedding order is d and that the embedding order after the input selection is m $(m \leq d)$.

A. Direct strategy

The Direct strategy first embeds the original series into H datasets

$$D_1 = \{(\mathbf{x}_{i1}, y_{i1}) \in (\mathbb{R}^m \times \mathbb{R})\}_{i=1}^N, \quad (1)$$

$$\vdots$$

$$D_H = \{(\mathbf{x}_{iH}, y_{iH}) \in (\mathbb{R}^m \times \mathbb{R})\}_{i=1}^N. \quad (2)$$

with

$$\mathbf{x}_{ih} \subset \{\varphi_i, \ldots, \varphi_{i+d-1}\}, \quad (3)$$

$$y_{ih} = \{\varphi_{i+d-1+h}\}, \quad h \in \{1, \ldots, H\}. \quad (4)$$

and where $x_{.h}$ stands for the subset of inputs returned by the input selection procedure

Once the time series is embedded, the Direct prediction strategy learns H direct models $f_h(\cdot)$ with

$$y_{ih} = f_h(\mathbf{x}_{ih}) + w_{ih}\ , \quad (5)$$

where w_{ih} denotes the additive noise.

An attractive property of this method is that it is not prone to the accumulation of the prediction errors [17]. However, the conditional independence of the H trained models does not allow the technique to keep into consideration complex dependency patterns existing between the variables φ_{t+h} [8].

B. MIMO Strategy

In order to remove the conditional independence assumption which is implicit in the Direct approach, a MIMO strategy was introduced in [8].

Unlike the Direct strategy, MIMO considers a single dataset

$$D = \{(\mathbf{x}_i, \mathbf{y}_i) \in (\mathbb{R}^m \times \mathbb{R}^H)\}_{i=1}^N, \quad (6)$$

where

$$\mathbf{x}_i \subset \{\varphi_i, \ldots, \varphi_{i+d-1}\}, \quad (7)$$

$$\mathbf{y}_i = \{\varphi_{i+d}, \ldots, \varphi_{i+d+H-1}\}. \quad (8)$$

The MIMO strategy estimates a single multi-output model $f(\cdot)$ with

$$\{\mathbf{y}_i\} = f(\mathbf{x}_i) + \mathbf{w}_i, \quad (9)$$

which returns a vector of H predictions. The MIMO strategy constrains all the horizons to be predicted with the same model structure, for instance with the same set of inputs $\mathbf{x}$. This constraint greatly reduces the flexibility of the prediction approach and can excessively bias the returned model. This is not the case of the Direct strategy, where each model is allowed to use a different set of input variables.

C. MISMO strategy

A solution to the shortcomings of the previously discussed techniques comes from the adoption of an intermediate approach, where the constraint of MIMO is relaxed by tuning an integer parameter s, which calibrates the dimensionality of the output on the basis of a validation criterion. For a given s, the training set of the MISMO techniques is composed of $n = \frac{H}{s}$ portions

$$D_1 = \{(\mathbf{x}_{i1}, \mathbf{y}_{i1}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N, \quad (10)$$

$$\vdots$$

$$D_n = \{(\mathbf{x}_{in}, \mathbf{y}_{in}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N. \quad (11)$$

where

$$\mathbf{x}_{ip} \subset \{\varphi_i, \ldots, \varphi_{i+d-1}\}, \quad (12)$$

$$\mathbf{y}_{ip} = \{\varphi_{i+d+(p-1)s}, \ldots, \varphi_{i+d+ps-1}\}. \quad (13)$$

and $p \in \{1, \ldots, n\}$.

The MISMO technique trains $n = \frac{H}{s}$ models $f_p(\cdot)$ with

$$\{\mathbf{y}_{ip}\} = f_p(\mathbf{x}_{ip}) + \mathbf{w}_{ip}, \quad (14)$$

where $p \in \{1, \ldots, n\}$. We can see that for $s = 1$, the approach boils down to the direct approach, while for $s = H$ we have the MIMO predictor.

For instance, given an 100-steps-ahead prediction task, the dataset of the MISMO strategy with $s = 50$ is composed of 2 parts

$$D_1 = \{(\mathbf{x}_{i1}, \mathbf{y}_{i1}) \in (\mathbb{R}^m \times \mathbb{R}^{50})\}_{i=1}^N, \quad (15)$$

$$D_2 = \{(\mathbf{x}_{i2}, \mathbf{y}_{i2}) \in (\mathbb{R}^m \times \mathbb{R}^{50})\}_{i=1}^N. \quad (16)$$

and the MISMO strategy estimates 2 models, f_1 and f_2 with

$$\{\mathbf{y}_{i1}\} = f_1(\mathbf{x}_{i1}) + \mathbf{w}_{i1}, \quad (17)$$

$$\{\mathbf{y}_{i2}\} = f_2(\mathbf{x}_{i2}) + \mathbf{w}_{i2}. \quad (18)$$

Note that, throughout the paper, we assume that the value of the parameter s is a divisor of H. If it is not the case, it is sufficient to increase the horizon H of the quantity $[s - (H \bmod s)]$ where mod stands for the modulo operation.

III. GLOBAL METHODOLOGY

The general principle underlying the MISMO strategy can lead to distinct implementations according to the learner used to estimate the dependencies f_p and the procedure adopted to tune the value of the parameter s. In the following, we will detail the choices made in this paper in order to design a long-term predictor to be used in real forecasting tasks.

A. Learning and validation procedure

The estimation of the functions f_p in (14) relies on a nearest neighbor approach, where the problem of adjusting the size of the neighborhood is solved by a Lazy Learning strategy [9], [10]. Lazy Learning algorithms are query-based local learning algorithms, i.e. they defer the whole learning process until a specific query needs to be answered, and once the prediction is returned, they discard both the answer and the constructed model [18].

In this work, we use a multi-output extension of the Lazy Learning model called LL-MIMO previously discussed in [8].

Once we have embedded the time series, the forecasting problem boils down to $n = \frac{H}{s}$ multi-input multi-output supervised learning tasks

$$D_1 = \{(\mathbf{x}_{i1}, \mathbf{y}_{i1}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N, \tag{19}$$

$$\vdots$$

$$D_n = \{(\mathbf{x}_{in}, \mathbf{y}_{in}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N. \tag{20}$$

where H is the horizon of the prediction and s is the size of the output vectors $\mathbf{y}_{ip}$(see equation 11).

Given a query point $\mathbf{x}_q \in \mathbb{R}^m$ and a metric on the space $\mathbb{R}^m$, we estimate the n prediction vectors $\mathbf{y}_{qp}$ of $\mathbf{x}_q$, for each dataset D_p as follows :

- Given k, the number of neighbors
- Order increasingly the set of vectors $\{\mathbf{x}_{ip}\}_{i=1}^N$ with respect to the distance to $\mathbf{x}_q$
- Denote by $[j]$ the index of the jth closest neighbor of $\mathbf{x}_q$
- $\hat{\mathbf{y}}_{qp} = \frac{1}{k}\sum_{j=1}^k \mathbf{y}_{[j]p}$ where $\mathbf{y}_{[j]p}$ is the output of the jth closest neighbor of $\mathbf{x}_q$ in D_p

After the calculation of $\hat{\mathbf{y}}_{qp}$ for each $p \in \{1, \ldots, n\}$, the long term prediction is given by the concatenated vector $(\hat{\mathbf{y}}_{q1}, \ldots, \hat{\mathbf{y}}_{qn})$.

The adoption of local approach to solve a prediction task requires the choice of a set of model parameters (e.g. the number k of neighbors, the kernel function, the distance metric). In the following, we will present two criteria to assess and compare local models with different number of neighbors. The first criterion is a multi-ouput extension of the local Leave-One-Out (LOO) and the second is a measure of discrepancy between the training series and the forecasted sequence, which rely either on linear or nonlinear measures [19].

A computationally efficient way to perform LOO cross-validation and to assess the performance in generalization of local linear models is the PRESS statistic, proposed in 1974 by Allen [20]. By assessing the performance of each local model, alternative configurations can be tested and compared in order to select the best one in terms of expected prediction.

Let D denote a multi-input single-output dataset

$$D = \{(\mathbf{x}_i, y_i) \in (\mathbb{R}^m \times \mathbb{R})\}_{i=1}^N \tag{21}$$

and suppose we want to estimate the output for a query point $\mathbf{x}_q \in \mathbb{R}^m$. The idea consists of associating a LOO error $e_{LOO}(k)$ to the estimation

$$\hat{y}_k = \frac{1}{k}\sum_{j=1}^k y_{[j]} \tag{22}$$

returned by k neighbors. In case of constant model, the LOO term can be derived as follows [18]:

$$e_{LOO}(k) = \frac{1}{k}\sum_{j=1}^k (e_j(k))^2, \tag{23}$$

$$e_j(k) = y_{[j]} - \frac{\sum_{i=1(i\neq j)}^k y_{[i]}}{k-1} = k\frac{y_{[j]} - \hat{y}_k}{k-1}. \tag{24}$$

The optimal number of neighbors is then defined as the number

$$k^* = \arg\min e_{LOO}(k), \tag{25}$$

which minimizes the LOO error.

Now, in a multi-input multi-output setting with an output of size s, where

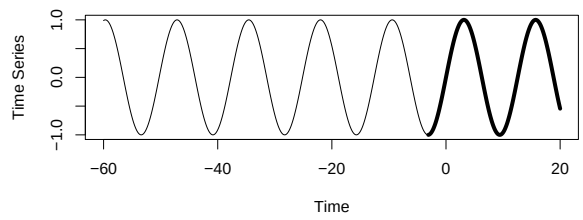
$$D = \{(\mathbf{x}_i, \mathbf{y}_i) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N, \tag{26}$$

we can define as multi-step LOO error the quantity

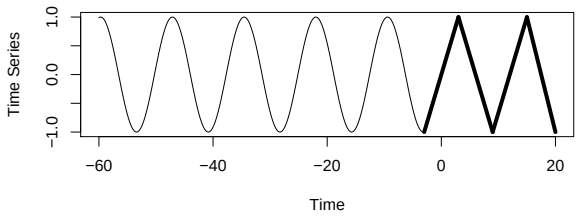
$$E_{LOO}(k) = \frac{1}{s}\sum_{h=1}^s (e_{LOO}^h(k))^2, \tag{27}$$

where $e_{LOO}^h(k)$ is the LOO error for the horizon h in (23). In the following, LL-MIMO-CV will refer to a Lazy Learner which selects the number of neighbors according to the multiple LOO assessment. Note that, in a MISMO setting, there are several datasets D_p and then, an optimal number of neighbors $k*_p$ has to be found for each dataset D_p.

The second criterion we propose, is not based on cross-validation but rather on a measure of stochastic discrepancy between the forecasted sequence and the training time series. The rationale is that, the lower the discrepancy between the descriptors of the prediction and the descriptors of the training series, the better is the quality of the returned forecast [8]. Suppose, for instance, that we have a training time series, which behaves like a sine wave and that we want to predict $H = 20$ steps ahead. Suppose, that we have to choose betwen two alternative models (e.g. two LL models with different number of neighbors) whose predictions (in bold on the Figures 1(a) and 1(b)) have not significantly different LOO errors. The question is "which one is the



(a) Training series (normal line) and the forecasted sequence (bold line)



(b) Training series (normal line) and the forecasted sequence (bold line)

Fig. 1. Training series and forecasted sequence of two models, which have not significantly different LOO errors.

best, knowing that they do not have significantly different LOO errors ?". Intuitively, we would choose for the model in Figure 1(a) since its predicted continuation has a "look" more similar to the training time series. Now, we have to define mathematically the term "look".

We can define several measures of discrepancy, both linear and non-linear . For example, we can use the autocorrelation or the partial autocorrelation for the linear case and the likelihood for the non-linear case. In this work, we will restrict to consider linear measures. Let φ denote the training time series and $\hat{\mathbf{y}}_k$ the estimation returned by k neighbors. The linear discrepancy measure Δ_k is

$$\begin{aligned}\Delta_k &= 1 - |cor[\rho(\varphi \cdot \hat{\mathbf{y}}_k), \rho(\varphi)]| \\ &+ 1 - |cor[\pi(\varphi \cdot \hat{\mathbf{y}}_k), \pi(\varphi)]|,\end{aligned}$$

where $\varphi \cdot \hat{\mathbf{y}}_k$ is the concatenation of the training time series φ and the forecasted sequence $\hat{\mathbf{y}}_k$, $\rho(x)$ is the autocorrelation of the series x, $\pi(x)$ is the partial autocorrelation of the series x and $cor[x, y]$ is the correlation of the series x and the series y.

So, we can associate to the long-term forecasting $\hat{\mathbf{y}}_k$, a measure of quality, which is not based on cross-validation but rather on the preservation of the stochastic properties of the series. The corresponding selection criterion is then

$$k^* = \text{arg min } \Delta_k, \tag{28}$$

which aims to find the number of neighbors k for which the predicted sequence is the closest, in terms of stochastic properties, to the training series. In the following, LL-MIMO-D will refer to a LL predictor where the neighbor selection relies on such criterion.

B. Choice of the parameter s

In the MISMO model (14) the parameter s addresses the bias/variance trade-off by constraining the degree of dependency between the predictions. The value of this parameter s is expected to play a major role in the accuracy of the method.

A possible way to choose the value of this parameter, is to analyze the performance of the MISMO model for different values of s on the learning set, and then use the best value obtained so far, to estimate the outputs. The following algorithm illustrates a cross-validated strategy to choose a good value of s.

Algorithm 1: Selection of the parameter s

Input : $\varphi = \{\varphi_1, \ldots, \varphi_t\}$, the time series
Input : $m =$ Embedding order
Input : $H =$ Horizon
Input : K= Max range of number of neighbors
Output: s^*, best value of the parameter s

1 **for** s *in* $\{1, \ldots, H\}$ **do**
  $n = \frac{H}{s}$
  $D_1^{(s)} = \{(\mathbf{x}_{i1}, \mathbf{y}_{i1}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N$
  $\vdots$
  $D_n^{(s)} = \{(\mathbf{x}_{in}, \mathbf{y}_{in}) \in (\mathbb{R}^m \times \mathbb{R}^s)\}_{i=1}^N$
2 **for** p *in* $\{1, \ldots, n\}$ **do**
    $D_p^{(s)} = \{(\mathbf{x}_{ip}, \mathbf{y}_{ip}) \in (\mathbb{R}^d \times \mathbb{R}^s)\}_{i=1}^N$
    $E_{learning}^{(p)} =$ vector of size K
    for nn *in* $\{2, \ldots, K\}$ **do**
      $E_{learning}^{(p)}[nn] \leftarrow$ Learn model on $D_p^{(s)}$ with range $\{2, \ldots, nn\}$
    end
  end
  $E_{learning}(s) =$ vector of size K
  for nn *in* $\{2, \ldots, K\}$ **do**
    $E_{learning}(s)[nn] \leftarrow \frac{1}{n} \sum_{p=1}^{n} E_{learning}^{(p)}[nn]$
  end
  $Error(s) \leftarrow \frac{1}{K} \sum_{nn=2}^{K} E_{learning}(s)[nn]$
end
$s^* =$ arg min $Error(s)$
return s^*

The first loop (line 1) ranges over all the values of the parameter s. Given a value of the parameter s, the output $\mathbf{y}$ is divided into $n = \frac{H}{s}$ portions $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$. The second loop (line 2) processes each portions separately by measuring the LOO error for a number $nn \in \{2, \ldots, K\}$ of neighbors. For a given s, we obtain as a result a vector $E_{learning}^{(p)}$ of size K for each ouput $\mathbf{y}_p$, $p \in \{1, \ldots, n\}$. Now, we average over p to obtain a vector $E_{learning}(s)$ of length K whose k term represents the LOO performance of the MISMO model with the parameter s and k neighbors.

Several methods could be used to derive a global (i.e. independent of the number of neighbors) estimate $Error(s)$ of the MISMO performance for a given s. For instance, we

could take the minimum or the mean value of the vector $E_{learning}(s)$.

In this work, we adopt an averaging approach and we estimate the performance of the MISMO model by taking the mean value of the vector $E_{learning}(s)$.

C. Input Selection Procedure

The flexibility of the MISMO approach allows a different input selection for each of the n prediction tasks.

Input selection [21],[22] consists of choosing a subset of $(x_{\cdot 1}, x_{\cdot 2}, \ldots, x_{\cdot d})$, that has the maximal predictive power ($x_{\cdot i}$ is the ith variable of the vector $\mathbf{x}$). Each input selection procedure relies on two elements: *a relevance criterion* and *a search procedure* [23].

The relevance criterion is a statistical measure of the relevance of the variable selected. Several relevance criteria like Mutual information, Gamma test,etc [23] have been proposed in litterature. In this work we adopted as *relevance criterion* the 2-fold cross-validation of a 1-NN approximator [24]. Since in the MIMO technique the size of the output $\mathbf{y}$ can be greater than one, the relevance criterion is adapted for multiple outputs by taking an average over the prediction horizon H, as shown in the following pseudo-code.

Calculation of the relevance criterion

- Given a dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i) \in (\mathbb{R}^d \times \mathbb{R}^s)\}_{i=1}^N$
- Given a set of variables V of size m ($m \leq d$), with $V \subset \{x_{\cdot 1}, \ldots, x_{\cdot d}\}$
- Given a metric on the space $\mathbb{R}^m$
- Divide the N input-outputs pairs in two parts $D1$ and $D2$
- For each point $(\mathbf{x}_i, \mathbf{y}_i)$ in $D1$
  - Find the nearest neighbor, say $\mathbf{x}_i^*$, of $\mathbf{x}_i$ in $D2$ according to the metric and the set V.
  - Calculate $err_{x_i} = \frac{1}{s}\sum_{j=1}^{s}(y_{ij} - y_{ij}^*)^2$ where y_{ij} is the jth component of $\mathbf{y}_i$ and y_{ij}^* is the jth component of $\mathbf{y}_i^*$
- For each point $(\mathbf{x}_i, \mathbf{y}_i)$ in $D2$
  - Find the nearest neighbor, say $\mathbf{x}_i^*$, of $\mathbf{x}_i$ in $D1$ according to the metric and the set V.
  - Calculate $err_{x_i} = \frac{1}{s}\sum_{j=1}^{s}(y_{ij} - y_{ij}^*)^2$ where y_{ij} is the jth component of $\mathbf{y}_i$ and y_{ij}^* is the jth component of $\mathbf{y}_i^*$
- Calculate $C(V) = \frac{1}{N}\sum_{i=1}^{N} err_{x_i}$ which is the statistical measure of the relevance of the set of variables V.

As far as the search is concerned we adopted a Forward-Backward Search procedure(FBS) [21],[22] which is a combination of conventional Forward and Backward search. FBS is flexible in the sense that a variable is able to return to the selected set once it has been dropped and vice versa, a previously selected variable can be discarded later. This method can start from any initial input variable set: empty set, full set, custom set or randomly initialized set.

In our experiments, we use the Forward-Backward method with four sets of initial variables, the first one is the empty set, and the three others are randomly initialized sets. After calculating the value of the criterion for these four sets, the final set of variables is the one which minimizes the relevance criterion. We are not starting the FBS from the full set, because our experiments have shown that it leads to far more selected variables and less accurate results. This is due to the local optima problem of the search.

IV. Experiments

In order to assess the performance of the MISMO strategy, we tested it on the ESTSP 2007 competition dataset [15], [16]. The training series, composed of 875 values, is shown in Figure 2.

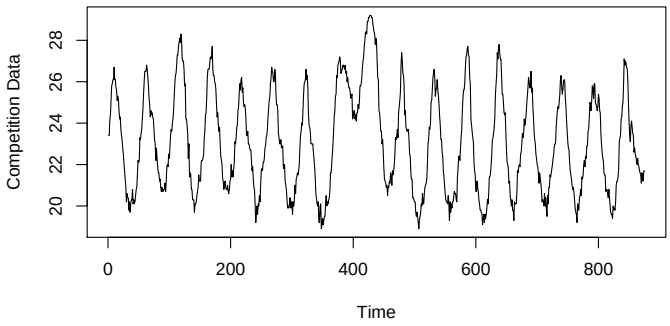


Fig. 2. ESTSP 2007 Competition dataset.

We set the maximum embedding order d to 55 [15], the maximum number of neighbors in LL to $K = 160$ and we consider prediction tasks with $H = 100$. For time reasons, the selection of the s value was restricted to the five values of the set $s = \{1, 5, 10, 50, 100\}$. Note that the value $s = 1$ corresponds to the Direct strategy, $s = 100$ to the MIMO strategy and the values $s = \{5, 10, 50\}$ to intermediate MISMO strategies.

Figure 3 plots the vector of the LOO errors obtained for the training series as a function of the number of neighbors for the five values of the parameter s.

This figure suggests that the MISMO model with $s = 50$ is on average better than the others.

The following figures compare the predicted continuation of the 5 MISMO models (both for LL-MIMO-CV and LL-MIMO-D) to the real continuation of the series.

Figure 4 shows the prediction returned by the MISMO with $s = 1$, i.e. by the Direct strategy. The Mean Square Error (MSE) is 1.68 for the LL-MIMO-CV model. Note

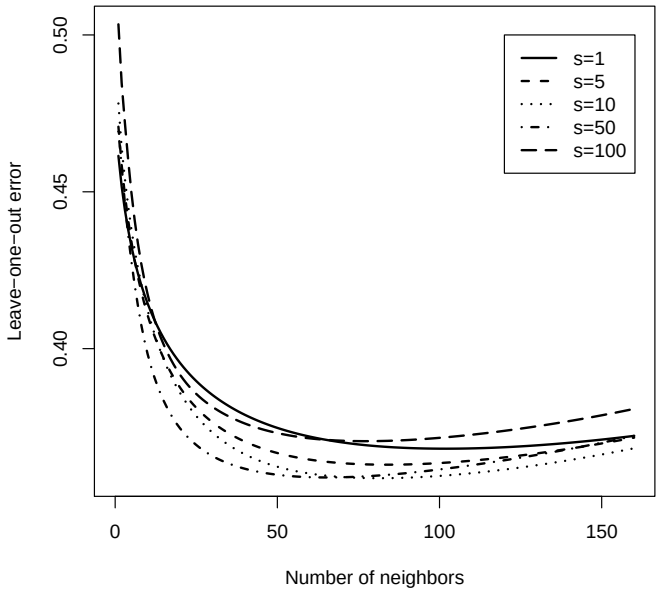


Fig. 3. Performance of MISMO on the 10-fold cross validation with $s = 1$ (solid line), $s = 5$ (dashed line), $s = 10$ (dotted line), $s = 50$ (dotdashed line) and $s = 100$ (large dashed line).

that the LL-MIMO-D prediction is not present here since the too small size of the output prevents the autocorrelation computation.

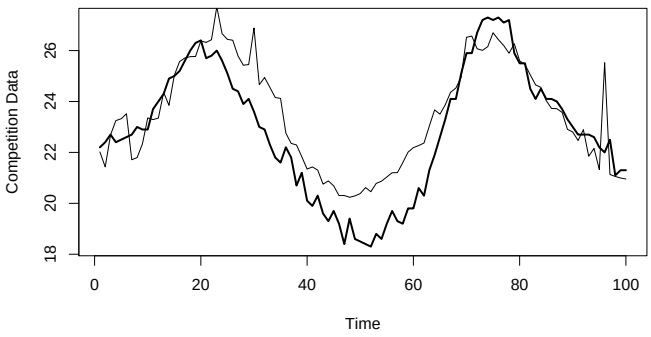


Fig. 4. ESTSP Competition dataset, prediction of 100 values by using the MISMO strategy with $\mathbf{s} = \mathbf{1}$, which corresponds to the Direct strategy. Solid thick line represents the real value and the solid thin line is the prediction with the LL-MIMO-CV model.

Figure 5 shows the $s = 5$ predictions. The Mean Square Error (MSE) amounts to 1.24 for the LL-MIMO-CV model and to 1.57 for the LL-MIMO-D model.

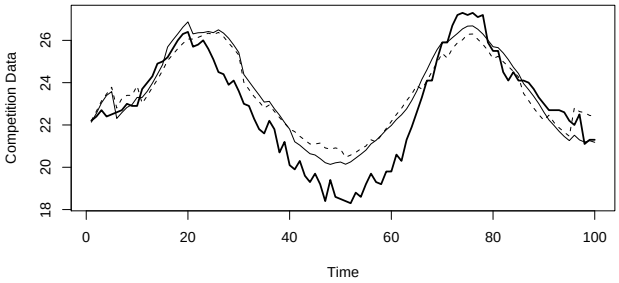


Fig. 5. ESTSP Competition dataset, prediction of 100 values by using the MISMO strategy with $\mathbf{s} = \mathbf{5}$. Solid thick line represents the real value, the solid thin line is the prediction with the LL-MIMO-CV model and the dotted one is the prediction with the LL-MIMO-D model.

The $s = 10$ case is illustrated by Figure 6. The accuracy is worse than in the $s = 5$ case. The MSE is 1.38 for the LL-MIMO-CV model and 1.05 for the LL-MIMO-D model.

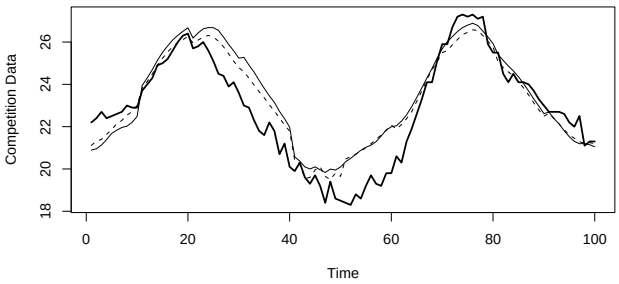


Fig. 6. ESTSP Competition dataset, prediction of 100 values by using the MISMO strategy with $\mathbf{s} = \mathbf{10}$. Solid thick line represents the real value, the solid thin line is the prediction with the LL-MIMO-CV model and the dotted one is the prediction with the LL-MIMO-D model.

As suggested by the results in cross-validation, the best performance is attained for $s = 50$ (Figure 7). The MSE drops to 0.82 for the LL-MIMO-CV model and to 0.67 for the LL-MIMO-D model.

The MIMO case ($s = 100$) is ilustrated by Figure 8. The MSE is 1.34 for the LL-MIMO-CV model and 0.95 for the LL-MIMO-D model.

In Table I, the errors for the learning and the test set are summarized for the different values of the parameter s.

The results of Table I and the related figures show clearly the impact of the parameter s on the accuracy of the prediction and justify the adoption of a MISMO strategy. The value of s controls effectively the trade-off between bias and

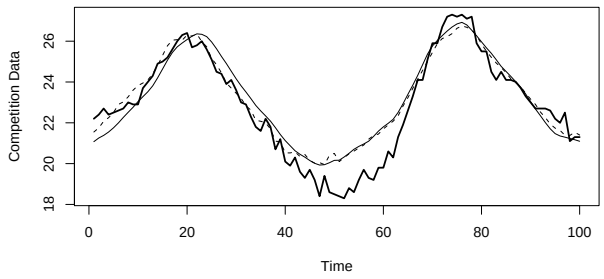


Fig. 7. ESTSP Competition dataset, prediction of 100 values by using the MISMO strategy with $\mathbf{s = 50}$. Solid thick line represents the real value, the solid thin line is the prediction with the LL-MIMO-CV model and the dotted one is the prediction with the LL-MIMO-D model.

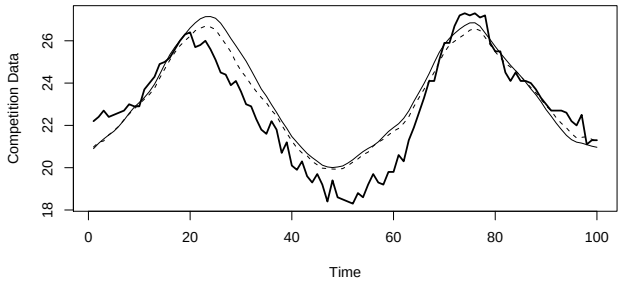


Fig. 8. ESTSP Competition dataset, prediction of 100 values by using the MISMO strategy with $\mathbf{s = 100}$. Solid thick line represents the real value, the solid thin line is the prediction with the LL-MIMO-CV model and the dotted one is the prediction with the LL-MIMO-D model.

	$s=1$	$s=5$	$s=10$	$s=50$	$s=100$
$E_{learning}$	0.3781	0.3733	0.3701	**0.3688**	0.3812
E_{test} LL-MIMO-CV	1.68	1.24	1.38	**0.82**	1.34
E_{test} LL-MIMO-D	-	1.57	1.05	**0.67**	0.95

TABLE I

LOO ERROR OF THE LEARNING AND THE TEST SET, FOR DIFFERENT VALUE OF THE PARAMETER s.

variance as made evident also by the nature of the predicted profile (strongly variant in the $s = 1$ case and progressively smoother for higher values of s). A careful choice of s should be then recommended in case of long-term forecasting.

The experimental results confirm also the important role of non cross-validated criteria in long term prediction tasks. Indeed it appears that LL-MIMO-D is competitive and sometimes better than the conventional cross-validated selection criterion. Once, more than a single prediction is required, criteria which resume the global behavior of the prediction become attractive.

Finally, it is worth adding that the MSE of the prediction returned by MISMO ($s = 50$) with the LL-MIMO-D model for an horizon $H = 50$ is 0.379. Note that this accuracy is better than the one obtained by the winner of the ESTSP 2007 competition (MSE$= 0.506$) [15], [16].

V. CONCLUSIONS

Predictive modelling for single output tasks is known to require careful procedures of assessment and selection. The extension of this procedure to long-term forecasting and consequently to multi-output modelling requires the tuning of additional parameters and the definition of specific calibration procedures. This paper discussed how the size of the multiple outputs play a major role in the generalization accuracy of a long-term forecasting predictor. We showed that Direct and MIMO strategies implicitly constrain the size of the output target with consequent impact on the bias/variance tradeoff and the consequent accuracy. We proposed a new strategy called MISMO and the associated calibration procedure which aims to decompose a long-term prediction task in the optimal number of subtasks. Preliminary results on the ESTSP 2007 competition show that the approach is promising.

REFERENCES

[1] A. Weigend and N. Gershenfeld, *Time Series Prediction: forecasting the future and understanding the past*. Harlow, UK: Addison Wesley, 1994.

[2] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse, "Direct and recursive prediction of time series using mutual information selection," in *Computational Intelligence and Bioinspired Systems: 8th International Workshop on Artificial Neural Networks, IWANN'05, Vilanova i la Geltra, Barcelona, Spain*, ser. Lecture Notes in Computer Science, vol. 3512. Springer-Verlag GmbH, June 8-10 2005, pp. 1010–1017.

[3] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, 2007.

[4] R. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.

[5] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series." *Physical Review Letters*, vol. 8, no. 59, pp. 845–848, 1987.

[6] J. McNames, "A nearest trajectory strategy for time series prediction," in *Proceedings of the InternationalWorkshop on Advanced Black-Box Techniques for Nonlinear Modeling*. Belgium: K.U. Leuven, 1998, pp. 112–128.

[7] T. Sauer, "Time series prediction by using delay coordinate embedding," in *Time Series Prediction: forecasting the future and understanding the past*, A. S. Weigend and N. A. Gershenfeld, Eds. Harlow, UK: Addison Wesley, 1994, pp. 175–193.

[8] G. Bontempi, "Long term time series prediction with multi-input multi-output local learning." in *Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP08*, 2008.

[9] B. Birattari and M. Bersini, "Lazy learning for local modeling and control design," 1997. [Online]. Available: http://citeseer.ist.psu.edu/bontempi97lazy.html

[10] M. Birattari, G. Bontempi, and H. Bersini, "Lazy learning meets the recursive least-squares algorithm," in *Advances in Neural Information Processing Systems 11*. MIT Press, 1999, pp. 375–381.

[11] G. Bontempi, M. Birattari, and H. Bersini, "Lazy learning for modeling and control design," *International Journal of Control*, vol. 72, no. 7/8, pp. 643–658, 1999.

[12] M. Birattari, G. Bontempi, and H. Bersini, "Lazy learning meets the recursive least-squares algorithm," in *NIPS 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge: MIT Press, 1999, pp. 375–381.

[13] G. Bontempi, M. Birattari, and H. Bersini, "Lazy learning for iterated time series prediction," in *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*, J. A. K. Suykens and J. Vandewalle, Eds. Katholieke Universiteit Leuven, Belgium, 1998, pp. 62–68.

[14] G. Bontempi, M. Birattari, and H. Bersin, "Local learning for iterated time-series prediction," in *Machine Learning: Proceedings of the Sixteenth International Conference*, I. Bratko and S. Dzeroski, Eds. San Francisco, CA: Morgan Kaufmann Publishers, 1999, pp. 32–38.

[15] A. Lendasse, Ed., *ESTSP 2007: Proceedings*, 2007.

[16] A. Lendasse. (2007) ESTSP'07 homepage. [Online]. Available: http://www.cis.hut.fi/projects/tsp/ESTSP/

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

[18] G. Bontempi, "Local learning techniques for modeling, prediction and control," Ph.D., IRIDIA-Universit Libre de Bruxelles, Louvain-la-Neuve, BELGIUM, 1999.

[19] T. W. Anderson, *The statistical analysis of time series [by] T. W. Anderson*. Wiley New York,, 1971.

[20] D. M. Allen, "The relationship between variable selection and data augmentation and a method for prediction." *Technometrics*, vol. 16, pp. 125–127, 1974.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=944968

[22] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition (Computer Science and Scientific Computing Series)*. Academic Press, September 1990. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0122698517

[23] D. Franois, "High-dimensional data analysis: optimal metrics and feature selection," Ph.D., Universit catholique de Louvain, Louvain-la-Neuve, BELGIUM, 2007.

[24] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, November 1995. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0198538642