Recursive multi-step time series forecasting by perturbing data

Souhaib Ben Taieb, Gianluca Bontempi
Machine Learning Group, Department of Computer Science
Faculty of Sciences, Université Libre de Bruxelles (ULB)
Brussels, Belgium
Email: sbentaie@ulb.ac.be, gbonte@ulb.ac.be

***Abstract*—The Recursive strategy is the oldest and most intuitive strategy to forecast a time series multiple steps ahead. At the same time, it is well-known that this strategy suffers from the accumulation of errors as long as the forecasting horizon increases. We propose a variant of the Recursive strategy, called RECNOISY, which perturbs the initial dataset at each step of the forecasting process in order to i) handle more properly the estimated values at each forecasting step and ii) decrease the accumulation of errors induced by the Recursive strategy. In addition to the RECNOISY strategy, we propose another strategy, called HYBRID, which for each horizon selects the most accurate approach among the REC and the RECNOISY strategies according to the estimated accuracy. In order to assess the effectiveness of the proposed strategies, we carry out an experimental session based on the 111 times series of the NN5 forecasting competition. Accuracy results are presented together with a paired comparison over the horizons and the time series. The preliminary results show that our proposed approaches are promising in terms of forecasting performance.**

***Keywords*-Time series; Multi-step forecasting; Machine Learning; Recursive forecasting; NN5 forecasting competition;**

I. Introduction

"Better predictions remain the foundation of all science..." [1].

Forecasting several future values of an observed time series plays an important role in nearly all fields of science and engineering, such as economics, finance, meteorology and telecommunication [2]. Unlike one-step ahead forecasting, multi-step ahead forecasting tasks are more difficult [3], since they have to deal with various additional complications, like accumulation of errors, reduced accuracy, and increased uncertainty [4], [5].

The forecasting domain has been influenced for a long time, by linear statistical methods such as ARIMA models [6]. However, in the last two decades, machine learning models have drawn attention and have established themselves as serious contenders to classical statistical models in the forecasting community [2], [7], [8].

These models, also called black-box or data-driven models [9], are examples of nonparametric nonlinear models which only use historical data to learn the stochastic dependency between the past and the future. Moreover, the empirical accuracy of several machine learning models has been explored in a number of forecasting competitions under different data conditions (e.g. the NN3, NN5, and the annual ESTSP competitions [10], [11]) creating interesting scientific debates in the area of data mining and forecasting [12]–[14].

In the forecasting community, researchers have paid attention to several aspects of the forecasting procedure such as model selection [15], effect of deseasonalization [16], forecasts combination [17] and many other critical topics [6]. However, methods for generating multi-step ahead forecasts for machine learning models did not receive as much attention, as pointed out by Kline: "*One issue that has had limited investigation is how to generate multiple-step-ahead forecasts*" [18].

Several strategies have been proposed in the literature to deal with an H-step ahead forecasting task such as Recursive (REC), Direct (DIR), DirRec (DIRREC), MIMO and MISMO [5], [19]. All these strategies learn the dependency between the past and the future in different manners by making specific assumptions. Among them, the oldest and most intuitive strategy is the REC strategy (also called Iterated or Multi-Stage). This strategy tackles an H-step forecasting task by iterating, H times, a one-step-ahead predictor. Once the predictor has estimated the future series value, this value is fed back as input to the following forecast.

It is well-know in the forecasting literature [5] that the REC strategy suffers from the accumulation of errors for two main reasons: the use of approximated values in the inputs in absence of real ones and the one-step ahead criterion adopted in the learning strategy. Different alternatives and adaptations have been proposed in the literature to avoid or minimize these shortcomings.

For example, one possible way to remedy to these shortcomings (but still using the REC strategy) is to move from one-step ahead to multi-step ahead criteria. This allows to take into account the temporal behavior of the multi-step forecasting problem and consequently improve the results of the REC strategy. Multi-step criteria are used in recurrent neural networks [20], [21] and local learning iterated techniques [22], [23]. However, the adoption of a multi-step criterion requires the adaptation of the learning algorithm, making their use problematic [24].

Another way to improve the REC strategy consists in

expanding the training dataset with a new one, made of forecasts obtained with a model trained on the original data. The goal of this method is to improve the accuracy by enlarging the dataset with additional (estimated) realisations of the stochastic process underlying the time series. The extension of the dataset can be seen as a time series denoising process since if the fitted model is accurate enough, it allows to partially get rid of the noise of the time series with positive effect on accuracy [25]. However, if the model is not accurate enough, it can alter the initial dataset negatively (negative transfer) and consequently deteriorate the performance.

The DirRec strategy [5] is another extension which combines the principles of REC and DIR strategies. DirRec adds a new approximated variable at each step which is either kept in case of improvement or removed. The advantage of this approach is that it ensures that the input vector always contain at least a fixed number of true values in addition to some approximated values. Nevertheless, a shortcoming resides in the fact that an input selection is required at each step. This means that, in terms of computational time and complexity, the strategy can become intractable for long horizons.

In this work, we go one step further by proposing a strategy, called RECNOISY, which perturbs the initial dataset at each step of the forecasting process to handle more properly the approximated values in the prediction process. This is done in a specific manner for each different horizon: for that reason in the following we will denote as forecasting task the forecasting related to a specific horizon.

The development of the RECNOISY method is notably motivated by the fact that the training examples used by the REC strategy, though observed, are not necessarily representative of the forecasting tasks which will be required later all along the forecasting process. In other terms, the training data contains no examples with approximated values while all the forecasting tasks will require predictions for inputs affected by model errors.

To remedy to this problem, the proposed strategy exploits the particular nature of the forecasting tasks induced by the REC strategy and incorporates it in the learning phase in order to improve the results. The incorporation of the knowledge about the structure in the data consists in having a specific dataset for each forecasting task. To do so, we start from the initial dataset and perturb the training examples according to a residual estimated in the previous forecasting task. It is then hoped that the learning algorithm will extract the particularities specific to that forecasting task from the perturbed data. Hence, a different model is learnt for each forecasting task (or equivalently for each horizon) using the specific dataset. By doing so, it is expected that each model will deal correctly with the model errors present in its inputs.

In addition to the RECNOISY strategy, we will propose a racing strategy, called HYBRID, which at each horizon compares the forecasting performance of both the REC and the RECNOISY strategies and selects the best one accordingly. This strategy can be effective in situation where the distribution of the residual is poorly estimated at some horizons.

The two proposed strategies are experimentally assessed against their REC counterparts on the 111 time series of the NN5 international forecasting competition benchmark. These time series pose some of the realistic problems that one usually encounters in a typical multi-step ahead forecasting task, for example the existence of several times series of possibly related dynamics, outliers, missing values, and multiple overlying seasonalities. Preliminary results for these new strategies appear to be promising.

The paper is organized as follows. The next section presents more precisely the RECNOISY method. Section III gives a detailed presentation of the datasets and the methodology applied for the experimental comparison. Section IV presents the results and discusses them. Finally, Section V gives a summary and concludes the work.

II. The RECNOISY strategy

Suppose we wish to forecast $H = 4$ steps ahead of a time series $y = [\varphi_1, \varphi_2, \ldots, \varphi_{20}]$ composed of $N = 20$ historical values. Suppose also that the REC strategy has been chosen to generate the forecasts.

Table I shows the dataset obtained after the time series has been embedded into a dataset D made of input-output pairs and defined as

$$D = \{(\mathbf{X}, y) | (\mathbf{x}_i, y_i) \in (\mathbb{R}^d \times \mathbb{R})\}, \tag{1}$$

where $\mathbf{x}_i$ is a temporal pattern of length $d = 3$, y_i is the next pattern and $i \in \{1, \ldots, N - d\}$.

Table I: Dataset after embedding the time series $[\varphi_1, \ldots, \varphi_{20}]$.

	$\mathbf{X}$			y
	v_3	v_2	v_1	
	φ_{t-2}	φ_{t-1}	φ_t	φ_{t+1}
Training	φ_1	φ_2	φ_3	φ_4
	φ_2	φ_3	φ_4	φ_5
	φ_3	φ_4	φ_5	φ_6
	φ_4	φ_5	φ_6	φ_7
	φ_5	φ_6	φ_7	φ_8
	φ_6	φ_7	φ_8	φ_9
	φ_7	φ_8	φ_9	φ_{10}
	φ_8	φ_9	φ_{10}	φ_{11}
	φ_9	φ_{10}	φ_{11}	φ_{12}
	φ_{10}	φ_{11}	φ_{12}	φ_{13}
	φ_{11}	φ_{12}	φ_{13}	φ_{14}
	φ_{12}	φ_{13}	φ_{14}	φ_{15}
Validation	φ_{13}	φ_{14}	φ_{15}	φ_{16}
	φ_{14}	φ_{15}	φ_{16}	φ_{17}
	φ_{15}	φ_{16}	φ_{17}	φ_{18}
	φ_{16}	φ_{17}	φ_{18}	φ_{19}
	φ_{17}	φ_{18}	φ_{19}	φ_{20}

The REC strategy estimates one model M which learns the dependency between past observations and one future observation from the dataset in Table I. In other words, the model M learns the dependency

$$\varphi_{t+1} = M(\varphi_t, \varphi_{t-1}, \varphi_{t-2}) + w \tag{2}$$

where w is an additive noise. After the model M has been learnt, it will be required to perform four different tasks as shown in Table II.

Table II: $H = 4$ forecasting tasks for the time series $[\varphi_1, \ldots, \varphi_{20}]$ when using the REC strategy.

	φ_{N-2}	φ_{N-1}	φ_N	$\hat{\varphi}_{N+1}$
Task 1	φ_{18}	φ_{19}	φ_{20}	$\hat{\varphi_{\mathbf{21}}}$
	φ_{N-1}	φ_N	$\hat{\varphi}_{N+1}$	$\hat{\varphi}_{N+2}$
Task 2	φ_{19}	φ_{20}	$\hat{\varphi_{\mathbf{21}}}$	$\hat{\varphi_{\mathbf{22}}}$
	φ_N	$\hat{\varphi}_{N+1}$	$\hat{\varphi}_{N+2}$	$\hat{\varphi}_{N+3}$
Task 3	φ_{20}	$\hat{\varphi_{\mathbf{21}}}$	$\hat{\varphi_{\mathbf{22}}}$	$\hat{\varphi_{\mathbf{23}}}$
	$\hat{\varphi}_{N+1}$	$\hat{\varphi}_{N+2}$	$\hat{\varphi}_{N+3}$	$\hat{\varphi}_{N+4}$
Task 4	$\hat{\varphi_{\mathbf{21}}}$	$\hat{\varphi_{\mathbf{22}}}$	$\hat{\varphi_{\mathbf{23}}}$	$\hat{\varphi_{\mathbf{24}}}$

We can see in Table II that the output (the forecasted value $\hat{\varphi}$) of each task is used as input in the subsequent tasks. Hence, each task takes as inputs some approximated values instead of actual observations with evident negative consequences in terms of error propagation.

Moreover, by analyzing the different forecasting tasks in Table II, we can notice (except for the first task) that the learnt model M will be required to perform forecasting tasks for which it has not been trained for. Indeed, by checking the dataset in Table I, we can see that the model was learnt on examples whose inputs do not contain any approximated values while each forecasting task (in Table II) gives inputs contaminated with model errors.

The proposed strategy exploits the particular nature of the forecasting tasks induced by the REC strategy. Indeed, we can see in Table II that the further we go with the horizon, the more inputs are affected by approximated values. In addition, each input of each forecasting task contains different types of approximated values. For example, by comparing tasks 2 and 3 in Table II, we can see that the task 2 contains one approximated value coming from a task which did not contain approximated values, while task 3 contains two approximated values, one of them from a task which contains approximated values.

So the REC strategy induces a certain structure which should be incorporated in the learning phase in order to improve the results. To incorporate the knowledge about the structure in the data, we propose to have a specific dataset for each forecasting task by starting with the initial dataset and then perturbing the training examples according to a residual estimated in the previous forecasting task. This idea of transforming the input examples has been already discussed in [26], where the authors considered the problem of transforming input examples in the context of training invariant support vector machines.

Require: $[\varphi_1, \ldots, \varphi_N]$, the time series to forecast.
Require: H, the forecasting horizon.
Ensure: $[\hat{\varphi}_{N+1}, \ldots, \hat{\varphi}_{N+H}]$, the forecasts.
// Create input-output dataset from time series and partition it.
$D = \{(\mathbf{X}, y) | (\mathbf{x}_i, y_i) \in (\mathbb{R}^d \times \mathbb{R})$ with $i \in \{1, \ldots, N - d\}\}$
Divide D in training, validation and residual sets: $D = \{D^{tr} \cup D^{val} \cup D^{res}\}$.
$\mathbf{x}_q^1 = \{\varphi_{N-d}, \ldots, \varphi_N\}$, query point.
for $h \in \{1, \ldots, H\}$ **do**
  // Create Dataset for the step h.
  $D_h = D$
  if $h > 1$ **then**
    Perturb $\mathbf{X}_h$ with $\{E_h, \ldots, E_1\}$: $\mathbf{X}_h = [\ldots, v_h + E_h, \ldots, v_1 + E_1]$
  end if
  // Identify the model
  Identify the parameters of the model M_h using D_h^{tr} and D_h^{val}.
  // Estimate the distribution of the residuals for step h.
  Predict each output $\hat{\varphi}$ of the dataset D_h^{res} using model M_h.
  Calculate $\epsilon^{M_h} = \{\varphi - \hat{\varphi}$ for each point in $D_h^{res}\}$.
  Create variable $\mathbf{E_h} \sim \mathcal{N}(E[\epsilon^{M_h}], Var^2[\epsilon^{M_h}])$
  // Identify the final model for step h.
  Create $D_h^{tr2} = \{D_h^{tr} \cup D_h^{val}\}$.
  Create $D_h^{val2} = D_h^{res}$.
  Identify the parameters of the final model M_h using D_{tr2} and D_{val2}.
  // Forecast $\hat{\varphi}_{N+h}$ using the final model M_h
  $\hat{\varphi}_{N+h} = M_h(\mathbf{x}_q^h)$.
  $\mathbf{x}_q^{h+1} = [\hat{\varphi}_{N+h} \cup \mathbf{x}_q^h]$
end for
return $[\hat{\varphi}_{N+1}, \ldots, \hat{\varphi}_{N+H}]$.

Figure 1: Algorithm of the RECNOISY strategy

Our aim is to have a predictive algorithm able to extract the particularities specific to each forecasting task from the perturbed data. For this reason, a different model is learnt for each forecasting task (or for each horizon) by using a specific perturbed dataset.

The pseudo-code of the RECNOISY strategy is given in Fig. 1.

The goal is to forecast the H future values $[\hat{\varphi}_{N+1}, \ldots, \hat{\varphi}_{N+H}]$ of a given time series $[\varphi_1, \ldots, \varphi_N]$ composed of N observations. After embedding the time

series in an input-output dataset (line 1), it is partitioned in three parts: the two first parts are dedicated to the identification of the model and third part is used to estimate the distribution of the residuals.

For each horizon (except for $h = 1$), a new dataset D_h is created from the initial dataset D by perturbing some variables according to the approximated values present in the forecasting task at step h (line 6-8). After the perturbation has been made, a model M_h is learnt for the horizon h using the training set D_h^{tr} and the validation set D_h^{val} (line 10).

Once the model M_h is available, it is used to predict the output of each points in the residual set D_h^{res}. This makes possible the computation of residuals $\epsilon = \varphi - \hat{\varphi}$ for each point and the consequent estimate of their distribution at step h (line 13). Note that we limit here to consider a parametric Gaussian estimator of the residual distribution, though non-parametric approaches could be taken into consideration by future work.

When the distribution of the residuals has been estimated, we concatenate the training and the validation set to form a new training set D_h^{tr2} and use the residual set as a new validation set D_h^{val2} (line 14-16). This dataset is used to learn a new model M_h and to generate the related forecast $\hat{\varphi}_{N+h}$, which will be added to the input of the next forecasting task $h + 1$ (line 17-18).

For the sake of clarity, a simplification has been made in Fig. 1. Instead of perturbing the residual set, we can take directly the forecasts of the previous steps. This allows us to have a better measure of the residuals since the inputs will contain the true forecasts of the previous steps and not simulated ones.

The application of the RECNOISY strategy to the examples of Tables I and II is shown in Table III. We can see that compared to the REC strategy, we have one dataset and one model M_h for each forecasting task. Concerning the perturbation of the datasets, we see that each variable is perturbed (in bold) according to the estimated residuals from the previous steps.

It is useful to make some considerations about the distinguishing features of RECNOISY with respect to the variants of REC available in literature.

First of all, we remark that the RECNOISY strategy deals only with one-step ahead predictors. So, other supervised algorithms (Neural Networks, SVM, K-NN, etc) can be used without specific adaptation. This represents an advantage with respect to the multi-step predictors.

If we compare with strategies expanding the dataset (noted EXPAND), the main difference is that RECNOISY uses true values in the output since the residuals are exclusively used to perturb the inputs. This guarantees reliable output values also when the distribution of the residuals is poorly estimated. This is not necessarily true for the EXPAND strategy which uses the forecasts obtained by a previously learnt model both as inputs and outputs. So, in case of poor model, the additional examples can have a negative impact on accuracy.

Ensure: Strategy to use at each horizon h.

1: **for** $h \in \{1, \ldots, H\}$ **do**
2: $\text{Err}^h_{\text{REC}} \leftarrow$ Average error for REC on test set D^{res} at horizon h.
3: $\text{Err}^h_{\text{RECNOISY}} \leftarrow$ Average error for RECNOISY on residual set D_h^{res} at horizon h.
4: **if** $\text{Err}^h_{\text{RECNOISY}}$ ¡ $\text{Err}^h_{\text{REC}}$ **then**
5: Use RECNOISY strategy at horizon h.
6: **else**
7: Use REC strategy at horizon h.
8: **end if**
9: **end for**

Figure 2: Algorithm of the HYBRID strategy

Finally, as far as the DirRec strategy is concerned, let us notice that the RECNOISY strategy preserves the original dimension of the training set. Instead of adding variables at each step, this strategy adds samples, therefore not exposing the learning to the curse of dimensionality.

At the same time it is worthy to stress two limitations of RECNOISY related to the sampling mechanism. The first is due to the fact that the variance of the predictor is increased by the sampling procedure performed to introduce the noise. The role of the experimental session will be to assess the impact of the sampling on the final accuracy. A second possible weakness is related to the fact that the strategy is vulnerable to wrong estimations of residuals which could deteriorate the accuracy wrt the REC approach.

For this reason, we propose a racing variant of the RECNOISY strategy. This new strategy, called HYBRID, is based on a racing mechanism to assess, compare and select the best strategy at each horizon. HYBRID uses either the REC or the RECNOISY strategy according to their respective forecasting performance on the test set (also called the residual test for the RECNOISY strategy). In brief, the idea behind the HYBRID strategy is to compare the quality of the predictions of each strategy on each horizon and choose the best one accordingly, as shown in Fig. 2.

III. Experimental setup

A. Data set and preprocessing

In these experiments, we use the time series from the NN5 forecasting competition which has been organized in 2007 to compare and evaluate the performance of computational intelligence methods in the context of multi-step time series forecasting.

Each of the 111 time series of this competition represents roughly two years of daily cash money withdrawl amounts (735 data points) at ATM machines at one of

Table III: RECNOISY strategy with an example for an $H = 4$ forecasting task.

Table IV: Horizon $h = 1$ - Dataset for model M_1

	X			y		
	v_3	v_2	v_1			
	φ_{t-2}	φ_{t-1}	φ_t	φ_{t+1}		
Training	φ_1	φ_2	φ_3	φ_4		
	φ_2	φ_3	φ_4	φ_5		
	φ_3	φ_4	φ_5	φ_6		
	φ_4	φ_5	φ_6	φ_7		
	φ_5	φ_6	φ_7	φ_8		
	φ_6	φ_7	φ_8	φ_9		
	φ_7	φ_8	φ_9	φ_{10}		
Validation	φ_8	φ_9	φ_{10}	φ_{11}		
	φ_9	φ_{10}	φ_{11}	φ_{12}		
	φ_{10}	φ_{11}	φ_{12}	φ_{13}		
	φ_{11}	φ_{12}	φ_{13}	φ_{14}		
					$\hat{\varphi}^{M_1}$	ϵ^{M_1}
Residual	φ_{12}	φ_{13}	φ_{14}	φ_{15}	$\hat{\varphi_{15}}$	$\varphi_{15} - \hat{\varphi_{15}}$
	φ_{13}	φ_{14}	φ_{15}	φ_{16}	$\hat{\varphi_{16}}$	$\varphi_{16} - \hat{\varphi_{16}}$
	φ_{14}	φ_{15}	φ_{16}	φ_{17}	$\hat{\varphi_{17}}$	$\varphi_{17} - \hat{\varphi_{17}}$
	φ_{15}	φ_{16}	φ_{17}	φ_{18}	$\hat{\varphi_{18}}$	$\varphi_{18} - \hat{\varphi_{18}}$
	φ_{16}	φ_{17}	φ_{18}	φ_{19}	$\hat{\varphi_{19}}$	$\varphi_{19} - \hat{\varphi_{19}}$
	φ_{17}	φ_{18}	φ_{19}	φ_{20}	$\hat{\varphi_{20}}$	$\varphi_{20} - \hat{\varphi_{20}}$
Task 1	φ_{18}	φ_{19}	φ_{20}	$\boxed{\hat{\varphi_{21}}}$		

Table V: Horizon $h = 2$ - Dataset for model M_2

	X			y		
	v_3	v_2	$\boldsymbol{v_1 + E_1}$			
	φ_{t-1}	φ_t	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+1}}$	φ_{t+2}		
Training	φ_1	φ_2	$\hat{\boldsymbol{\varphi_3}}$	φ_4		
	φ_2	φ_3	$\hat{\boldsymbol{\varphi_4}}$	φ_5		
	φ_3	φ_4	$\hat{\boldsymbol{\varphi_5}}$	φ_6		
	φ_4	φ_5	$\hat{\boldsymbol{\varphi_6}}$	φ_7		
	φ_5	φ_6	$\hat{\boldsymbol{\varphi_7}}$	φ_8		
	φ_6	φ_7	$\hat{\boldsymbol{\varphi_8}}$	φ_9		
	φ_7	φ_8	$\hat{\boldsymbol{\varphi_9}}$	φ_{10}		
Validation	φ_8	φ_9	$\hat{\boldsymbol{\varphi_{10}}}$	φ_{11}		
	φ_9	φ_{10}	$\hat{\boldsymbol{\varphi_{11}}}$	φ_{12}		
	φ_{10}	φ_{11}	$\hat{\boldsymbol{\varphi_{12}}}$	φ_{13}		
	φ_{11}	φ_{12}	$\hat{\boldsymbol{\varphi_{13}}}$	φ_{14}		
					$\hat{\varphi}^{M_2}$	ϵ^{M_2}
Residual	φ_{12}	φ_{13}	$\hat{\boldsymbol{\varphi_{14}}}$	φ_{15}	$\hat{\varphi_{15}}$	$\varphi_{15} - \hat{\varphi_{15}}$
	φ_{13}	φ_{14}	$\hat{\boldsymbol{\varphi_{15}}}$	φ_{16}	$\hat{\varphi_{16}}$	$\varphi_{16} - \hat{\varphi_{16}}$
	φ_{14}	φ_{15}	$\hat{\boldsymbol{\varphi_{16}}}$	φ_{17}	$\hat{\varphi_{17}}$	$\varphi_{17} - \hat{\varphi_{17}}$
	φ_{15}	φ_{16}	$\hat{\boldsymbol{\varphi_{17}}}$	φ_{18}	$\hat{\varphi_{18}}$	$\varphi_{18} - \hat{\varphi_{18}}$
	φ_{16}	φ_{17}	$\hat{\boldsymbol{\varphi_{18}}}$	φ_{19}	$\hat{\varphi_{19}}$	$\varphi_{19} - \hat{\varphi_{19}}$
	φ_{17}	φ_{18}	$\hat{\boldsymbol{\varphi_{19}}}$	φ_{20}	$\hat{\varphi_{20}}$	$\varphi_{20} - \hat{\varphi_{20}}$
Task 2	φ_{19}	φ_{20}	$\hat{\boldsymbol{\varphi_{21}}}$	$\boxed{\hat{\varphi_{22}}}$		

Table VI: Horizon $h = 3$ - Dataset for model M_3

	X			y		
	v_3	$\boldsymbol{v_2 + E_1}$	$\boldsymbol{v_1 + E_2}$			
	φ_t	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+1}}$	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+2}}$	φ_{t+3}		
Tr.	φ_1	$\hat{\boldsymbol{\varphi_2}}$	$\hat{\boldsymbol{\varphi_3}}$	φ_4		
	φ_2	$\hat{\boldsymbol{\varphi_3}}$	$\hat{\boldsymbol{\varphi_4}}$	φ_5		
	φ_3	$\hat{\boldsymbol{\varphi_4}}$	$\hat{\boldsymbol{\varphi_5}}$	φ_6		
	φ_4	$\hat{\boldsymbol{\varphi_5}}$	$\hat{\boldsymbol{\varphi_6}}$	φ_7		
	φ_5	$\hat{\boldsymbol{\varphi_6}}$	$\hat{\boldsymbol{\varphi_7}}$	φ_8		
	φ_6	$\hat{\boldsymbol{\varphi_7}}$	$\hat{\boldsymbol{\varphi_8}}$	φ_9		
	φ_7	$\hat{\boldsymbol{\varphi_8}}$	$\hat{\boldsymbol{\varphi_9}}$	φ_{10}		
Val.	φ_8	$\hat{\boldsymbol{\varphi_9}}$	$\hat{\boldsymbol{\varphi_{10}}}$	φ_{11}		
	φ_9	$\hat{\boldsymbol{\varphi_{10}}}$	$\hat{\boldsymbol{\varphi_{11}}}$	φ_{12}		
	φ_{10}	$\hat{\boldsymbol{\varphi_{11}}}$	$\hat{\boldsymbol{\varphi_{12}}}$	φ_{13}		
	φ_{11}	$\hat{\boldsymbol{\varphi_{12}}}$	$\hat{\boldsymbol{\varphi_{13}}}$	φ_{14}		
					$\hat{\varphi}^{M_3}$	ϵ^{M_3}
Res.	φ_{12}	$\hat{\boldsymbol{\varphi_{13}}}$	$\hat{\boldsymbol{\varphi_{14}}}$	φ_{15}	$\hat{\varphi_{15}}$	$\varphi_{15} - \hat{\varphi_{15}}$
	φ_{13}	$\hat{\boldsymbol{\varphi_{14}}}$	$\hat{\boldsymbol{\varphi_{15}}}$	φ_{16}	$\hat{\varphi_{16}}$	$\varphi_{16} - \hat{\varphi_{16}}$
	φ_{14}	$\hat{\boldsymbol{\varphi_{15}}}$	$\hat{\boldsymbol{\varphi_{16}}}$	φ_{17}	$\hat{\varphi_{17}}$	$\varphi_{17} - \hat{\varphi_{17}}$
	φ_{15}	$\hat{\boldsymbol{\varphi_{16}}}$	$\hat{\boldsymbol{\varphi_{17}}}$	φ_{18}	$\hat{\varphi_{18}}$	$\varphi_{18} - \hat{\varphi_{18}}$
	φ_{16}	$\hat{\boldsymbol{\varphi_{17}}}$	$\hat{\boldsymbol{\varphi_{18}}}$	φ_{19}	$\hat{\varphi_{19}}$	$\varphi_{19} - \hat{\varphi_{19}}$
	φ_{17}	$\hat{\boldsymbol{\varphi_{18}}}$	$\hat{\boldsymbol{\varphi_{19}}}$	φ_{20}	$\hat{\varphi_{20}}$	$\varphi_{20} - \hat{\varphi_{20}}$
Task 3	φ_{20}	$\hat{\boldsymbol{\varphi_{21}}}$	$\hat{\boldsymbol{\varphi_{22}}}$	$\boxed{\hat{\varphi_{23}}}$		

Table VII: Horizon $h = 4$ - Dataset for model M_4

	X			y		
	$\boldsymbol{v_3 + E_1}$	$\boldsymbol{v_2 + E_2}$	$\boldsymbol{v_1 + E_3}$			
	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+1}}$	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+2}}$	$\hat{\boldsymbol{\varphi}}_{\boldsymbol{t+3}}$	φ_{t+4}		
Tr.	$\hat{\boldsymbol{\varphi_1}}$	$\hat{\boldsymbol{\varphi_2}}$	$\hat{\boldsymbol{\varphi_3}}$	φ_4		
	$\hat{\boldsymbol{\varphi_2}}$	$\hat{\boldsymbol{\varphi_3}}$	$\hat{\boldsymbol{\varphi_4}}$	φ_5		
	$\hat{\boldsymbol{\varphi_3}}$	$\hat{\boldsymbol{\varphi_4}}$	$\hat{\boldsymbol{\varphi_5}}$	φ_6		
	$\hat{\boldsymbol{\varphi_4}}$	$\hat{\boldsymbol{\varphi_5}}$	$\hat{\boldsymbol{\varphi_6}}$	φ_7		
	$\hat{\boldsymbol{\varphi_5}}$	$\hat{\boldsymbol{\varphi_6}}$	$\hat{\boldsymbol{\varphi_7}}$	φ_8		
	$\hat{\boldsymbol{\varphi_6}}$	$\hat{\boldsymbol{\varphi_7}}$	$\hat{\boldsymbol{\varphi_8}}$	φ_9		
	$\hat{\boldsymbol{\varphi_7}}$	$\hat{\boldsymbol{\varphi_8}}$	$\hat{\boldsymbol{\varphi_9}}$	φ_{10}		
Val.	$\hat{\boldsymbol{\varphi_8}}$	$\hat{\boldsymbol{\varphi_9}}$	$\hat{\boldsymbol{\varphi_{10}}}$	φ_{11}		
	$\hat{\boldsymbol{\varphi_9}}$	$\hat{\boldsymbol{\varphi_{10}}}$	$\hat{\boldsymbol{\varphi_{11}}}$	φ_{12}		
	$\hat{\boldsymbol{\varphi_{10}}}$	$\hat{\boldsymbol{\varphi_{11}}}$	$\hat{\boldsymbol{\varphi_{12}}}$	φ_{13}		
	$\hat{\boldsymbol{\varphi_{11}}}$	$\hat{\boldsymbol{\varphi_{12}}}$	$\hat{\boldsymbol{\varphi_{13}}}$	φ_{14}		
					$\hat{\varphi}^{M_4}$	ϵ^{M_4}
Res.	$\hat{\boldsymbol{\varphi_{12}}}$	$\hat{\boldsymbol{\varphi_{13}}}$	$\hat{\boldsymbol{\varphi_{14}}}$	φ_{15}	$\hat{\varphi_{15}}$	$\varphi_{15} - \hat{\varphi_{15}}$
	$\hat{\boldsymbol{\varphi_{13}}}$	$\hat{\boldsymbol{\varphi_{14}}}$	$\hat{\boldsymbol{\varphi_{15}}}$	φ_{16}	$\hat{\varphi_{16}}$	$\varphi_{16} - \hat{\varphi_{16}}$
	$\hat{\boldsymbol{\varphi_{14}}}$	$\hat{\boldsymbol{\varphi_{15}}}$	$\hat{\boldsymbol{\varphi_{16}}}$	φ_{17}	$\hat{\varphi_{17}}$	$\varphi_{17} - \hat{\varphi_{17}}$
	$\hat{\boldsymbol{\varphi_{15}}}$	$\hat{\boldsymbol{\varphi_{16}}}$	$\hat{\boldsymbol{\varphi_{17}}}$	φ_{18}	$\hat{\varphi_{18}}$	$\varphi_{18} - \hat{\varphi_{18}}$
	$\hat{\boldsymbol{\varphi_{16}}}$	$\hat{\boldsymbol{\varphi_{17}}}$	$\hat{\boldsymbol{\varphi_{18}}}$	φ_{19}	$\hat{\varphi_{19}}$	$\varphi_{19} - \hat{\varphi_{19}}$
	$\hat{\boldsymbol{\varphi_{17}}}$	$\hat{\boldsymbol{\varphi_{18}}}$	$\hat{\boldsymbol{\varphi_{19}}}$	φ_{20}	$\hat{\varphi_{20}}$	$\varphi_{20} - \hat{\varphi_{20}}$
Task 4	$\hat{\boldsymbol{\varphi_{21}}}$	$\hat{\boldsymbol{\varphi_{22}}}$	$\hat{\boldsymbol{\varphi_{23}}}$	$\boxed{\hat{\varphi_{24}}}$		

the various cities in the UK. For each time series, the competition concerned the forecasting of the next 56 daily values.

The specificity of the NN5 series requires a preprocessing step called *gaps removal* where by gap we mean two types of anomalies: (i) zero values that indicate that no money withdrawal occurred and (ii) missing observations for which no value was recorded. About 2.5% of the data is corrupted by gaps. In our experiments we adopted the gap removal method proposed in [27]. If φ_m is the gap sample, this method replaces the gap with the median of the set $[\varphi_{m+365}, \varphi_{m-365}, \varphi_{m+7}$ and $\varphi_{m-7}]$ among which are available.

The adoption of deseasonalization may have a large impact on the forecasting strategies because the NN5 time series possess a variety of periodic patterns. For that reason we apply deseasonalization in order to better account for the role of the forecasting strategy. We adopt the deseasonaliza-

tion methodology discussed in [28] to remove the strong day of the week seasonality as well as the moderate day of the month seasonality. Note however that all the reported results are obtained after having restored back the seasonality in the predictions.

Every forecasting strategy requires the setting of the size d of the embedding dimension which represents the size of the input vector containing the past observations. Several approaches have been proposed in the literature to select this value [29]. Since this aspect is not a central theme in our paper we limited to consider, for each time series, the Delta test proposed in [30]. Note also that, for computational reasons, we did not apply any input selection procedure.

B. Methodology

The experimental session aims to benchmark the RECNOISY strategy and the HYBRID approach against the REC strategy. A paired comparison is done both by fixing the horizon and the time series. In other words, the first comparison aims to show for which horizons a strategy outperforms the other, while the second comparison compares the strategies for specific time series.

Concerning the learning algorithm, note that the goal of the paper is not to compare one learning algorithm with other state-of-the-art forecasting techniques but rather to show how for a specific learning algorithm (here the Nearest Neighbors) the introduction of the RECNOISY strategy can improve the performance in multi-step forecasting.

Each time series (containing 735 observations) is partitioned in three mutually exclusive sets namely training (605 values), validation (40 values) and residual (Day 646 to Day 735: 90 values). The training and the validation sets are used to build and tune a first model (see line 10 in Fig. 1). As we employ a Nearest Neighbor model, we need to select, for each dataset, the best number of neighbors to use in performing the forecasting task. The residual set (or test set) is used to measure the performance of each strategy (required by the HYBRID strategy) and to estimate the distribution of the residuals (line 16 of Fig. 1).

After that, the training and the validation sets are concatenated to form a new training set and the residual set forms a new validation set. The final model is then learnt using, in the one hand, the new training set made of 645 values and in the other hand, the new validation set made of 90 values (see lines 14-16). The resulting model is used to generate the 56 continuations required by the competition.

C. Performance evaluation and comparison

Given the ith ($i \in \{1, \ldots, 111\}$) time series and the hth ($h \in \{1, \ldots, H\}$) horizon, the forecasting error is computed as follows

$$\text{SMAPE}_h^{(i)} = \frac{|\hat{\varphi}_{N+h}^{(i)} - \varphi_{N+h}^{(i)}|}{(|\hat{\varphi}_{N+h}^{(i)} + \varphi_{N+h}^{(i)}|)/2}. \quad (3)$$

From the error defined in Equation 3, we derive two measures of performance, concerning the horizon and the time series, respectively. The first one gives a measure of the average performance (over the 111 time series) for a given horizon h and is defined as

$$\text{SMAPE}_h^{(*)} = \frac{1}{111} \sum_{i=1}^{111} SMAPE_h^{(i)}. \quad (4)$$

We can then associate to a given strategy a vector

$$\boxed{\text{SMAPE-H} = [\text{SMAPE}_1^{(*)}, \ldots, \text{SMAPE}_{56}^{(*)}]}, \quad (5)$$

returning the strategy's performance on the 56 horizons.

The second measure quantifies the average performance (over the 56 horizons) for a given time series i and is defined as

$$\text{SMAPE}_*^{(i)} = \frac{1}{56} \sum_{h=1}^{56} \text{SMAPE}_h^{(i)}. \quad (6)$$

We can then associate to a given strategy a vector

$$\boxed{\text{SMAPE-S} = [\text{SMAPE}_*^{(1)}, \ldots, \text{SMAPE}_*^{(111)}]}, \quad (7)$$

which returns a measure of the strategy's accuracy on the entire set of time series.

Finally, we can introduce a comprehensive measure of performance for a given strategy on all the series and all the horizons given by

$$\boxed{\text{SMAPE}^* = \frac{1}{111} \sum_{i=1}^{111} \text{SMAPE}_*^{(i)} = \frac{1}{56} \sum_{i=1}^{56} \text{SMAPE}_h^{(*)}}. \quad (8)$$

This measure was used as the main criterion for the NN5 forecasting competition.

In order to compare the performance of two different strategies, we will first consider their SMAPE* performance. Next, we will compare the performance of two strategies on the different horizons by taking the difference between their respective SMAPE-H. Eventually, the difference between their SMAPE-S is used to compare their performance on the different time series.

In order to assess the sensitivity of the approach to the sampling of the residual distribution we carried out 15 runs for each strategy. So, instead of having one vector with differences, we have 15 vectors of differences. Boxplot diagrams are used to display the entire set of differences.

IV. Results and discussion

A. Average results

Table VIII reports the SMAPE* measure of accuracy, defined in Equation 8, for the 15 runs.

We observe that the RECNOISY strategy outperforms REC over all the 15 runs. At the same time, the accuracy of

Table VIII: SMAPE* results of the REC, RECNOISY and HYBRID strategies for 15 runs. The numbers in bracket represent the improvement over the REC strategy.

Runs	REC	RECNOISY	HYBRID
1	22.30	21.42 (**-0.88**)	21.22 (**-1.08**)
2	22.30	21.33 (**-0.97**)	21.23 (**-1.07**)
3	22.30	21.49 (**-0.81**)	21.34 (**-0.96**)
4	22.30	21.33 (**-0.97**)	21.20 (**-1.10**)
5	22.30	21.21 (**-1.09**)	21.24 (**-1.06**)
6	22.30	21.36 (**-0.94**)	21.17 (**-1.13**)
7	22.30	21.33 (**-0.97**)	21.31 (**-0.99**)
8	22.30	21.32 (**-0.98**)	21.24 (**-1.06**)
9	22.30	21.30 (**-1.00**)	21.29 (**-1.01**)
10	22.30	21.31 (**-0.99**)	21.26 (**-1.04**)
11	22.30	21.31 (**-0.99**)	21.23 (**-1.07**)
12	22.30	21.25 (**-1.05**)	21.22 (**-1.08**)
13	22.30	21.27 (**-1.03**)	21.18 (**-1.12**)
14	22.30	21.24 (**-1.06**)	21.32 (**-0.98**)
15	22.30	21.25 (**-1.05**)	21.26 (**-1.04**)

the RECNOISY strategy appears to be low sensitive to the random sampling.

The other remarkable result is that the adoption of the HYBRID strategy leads to a further reduction of SMAPE* and this consistently over all the runs.

Finally, to test whether the observed differences are statistically significant, we employed a Wilcoxon signed rank test, which is a distribution free test for testing the significance of the difference between the performances of a pair of methods [31]. The outcome of this test is that the two proposed strategies are statistically better than the REC strategy in terms of SMAPE* measure.

B. Results over the horizons

In this section, we compare the different strategies over the 56 horizons using the SMAPE-H performance measure defined in Equation 5. Fig. 3 shows the boxplot of the differences between the SMAPE-H of the RECNOISY strategy and the REC strategy over the 15 runs, while Fig. 4 shows the same information for the HYBRID strategy.

Note that, a negative difference means that the proposed strategy outperforms the REC strategy according to the SMAPEH-H criterion.

In Fig. 3 , we can see that the RECNOISY strategy is particularly accurate and provides better results than the REC strategy in distant horizons. This is interesting given that the further we go with the horizon the more perturbations are applied to the dataset. So, it seems that, by exploiting the particular nature of the forecasting tasks and by incorporating it into the dataset, the RECNOISY strategy is able to extract the specificities of the forecasting task required at each horizon.

The results are further improved by adopting the HYBRID strategy as it can be observed by comparing Figures 3 and 4. These figures show the HYBRID strategy is able to improve the accuracy of the recursive approach also in short horizons.

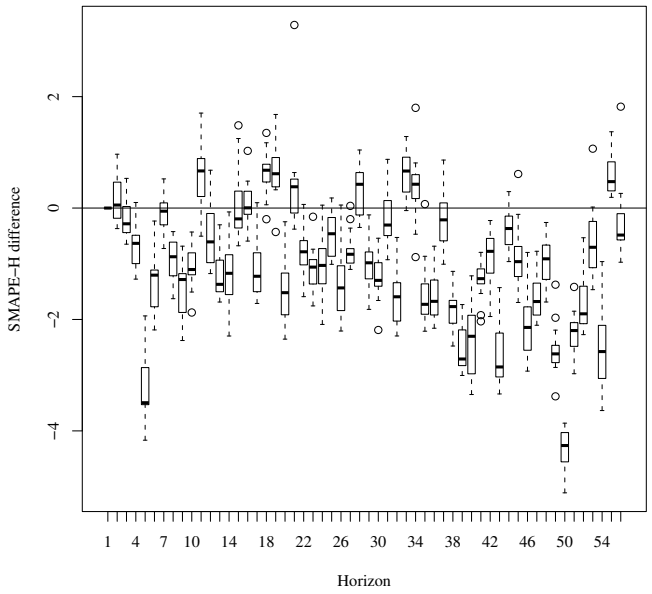

Figure 3: Boxplot of the difference between the SMAPE-H of the RECNOISY strategy and the REC strategy over 15 runs.

Finally, Fig. 5 shows, for each horizon, the average number of times (over the 15 runs) where the RECNOISY strategy performs better than the REC strategy on the residual set. Note that this figure shows also the number of times the RECNOISY strategy was selected by the HYBRID strategy at each horizon. We can see again that the RECNOISY strategy provides better results compared to the REC strategy particularly in distant horizons.

C. Results over the time series data

In this section, we compare the different strategies over the 111 time series using the SMAPE-S performance measure defined in Equation 7. Fig. 6 shows the boxplots of the difference between the SMAPE-S of the RECNOISY and the REC strategies over the 15 runs, while Fig. 7 shows the same information for the HYBRID strategy.

In Figures 6 and 7, since the order has no importance (in opposition to figures 3 and 4), the values have been sorted in order to have all the positive values on the left side and all the negative values on the right side.

The comparison of the RECNOISY strategy with the REC strategy in Fig. 6 shows that the RECNOISY strategy is more frequently better than the REC strategy (60 times out of 111). In addition, when the RECNOISY strategy is better (on the right side of the figure), the average improvement over REC is 3% with some cases going from 5% to 14%. In contrast, the REC strategy provides an average improvement

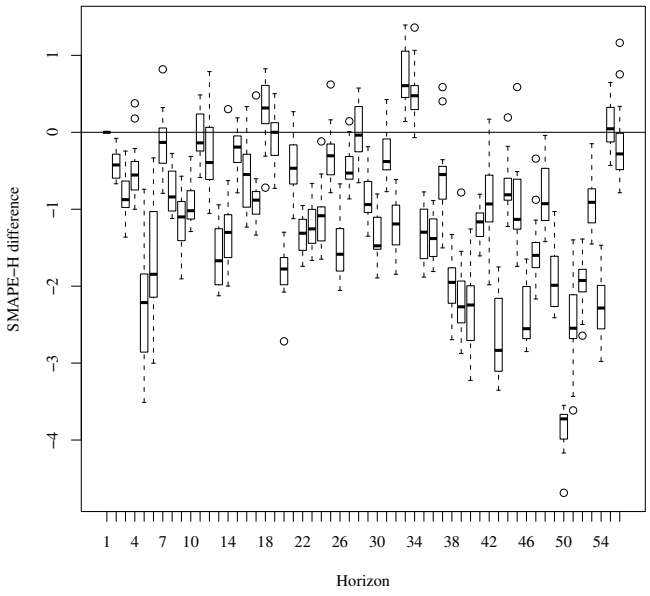

Figure 4: Boxplot of the difference between the SMAPE-H of the HYBRID strategy and the REC strategy over 15 runs.

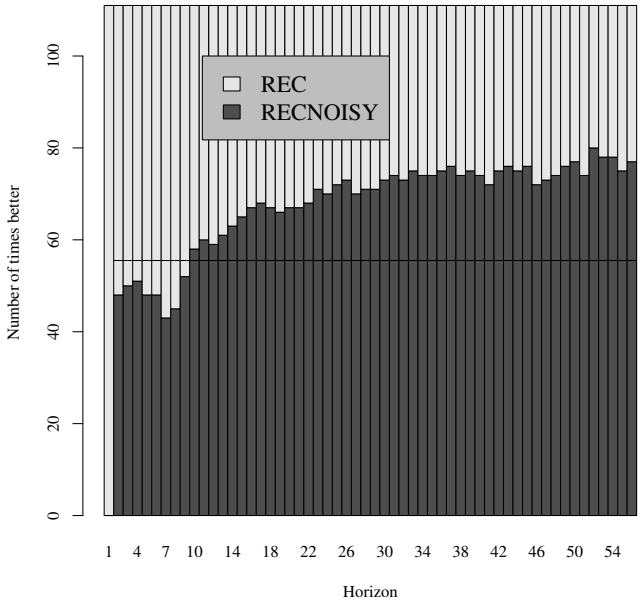

Figure 5: Average number of times the RECNOISY strategy performs better than the REC strategy (in black) and vice versa (in grey).

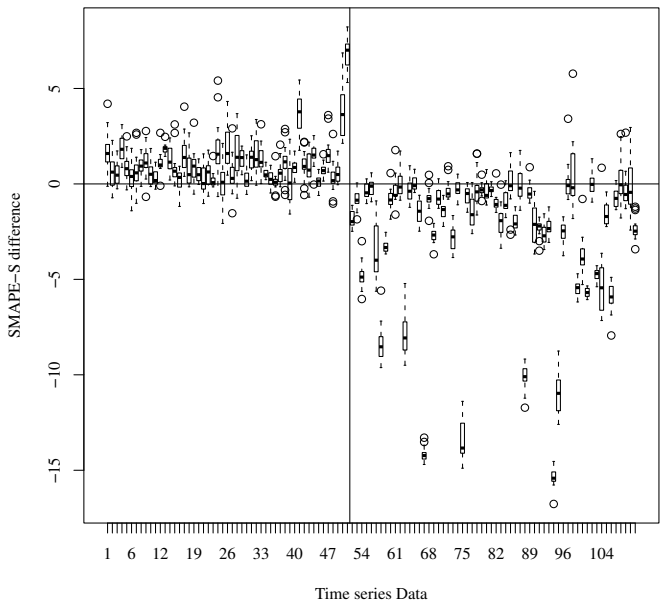

Figure 6: Boxplot of the difference between the SMAPE-S of the RECNOISY strategy and the REC strategy over 15 runs.

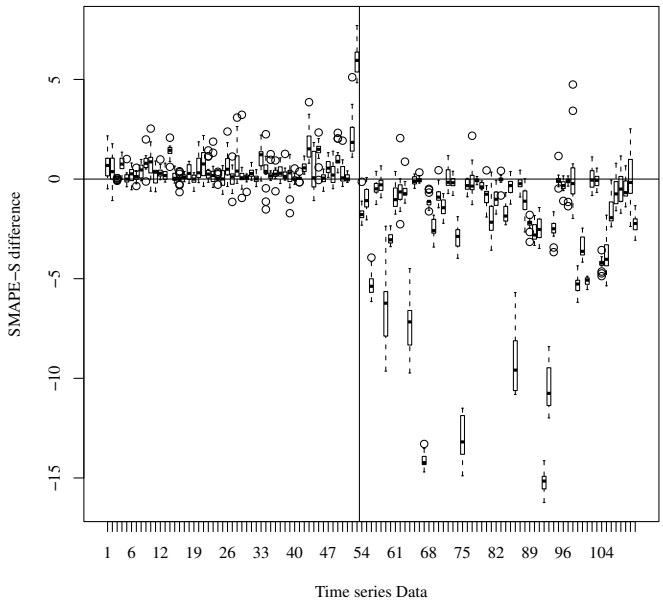

Figure 7: Boxplot of the difference between the SMAPE-S of the HYBRID strategy and the REC strategy over 15 runs.

of 1% over RECNOISY with some cases going from 2% to 4%. As far as the HYBRID strategy is concerned, we can see in Fig. 7 (on the left side of the figure) that the average improvement of REC over HYBRID is of 0.5 % compared to 1% in Fig. 6.

D. Forecast example

To illustrate the forecasting ability of the data-perturbing strategies, Fig. 8 (respectively Fig. 9) plots the forecasts obtained with the REC (resp. RECNOISY) strategy versus the true values.

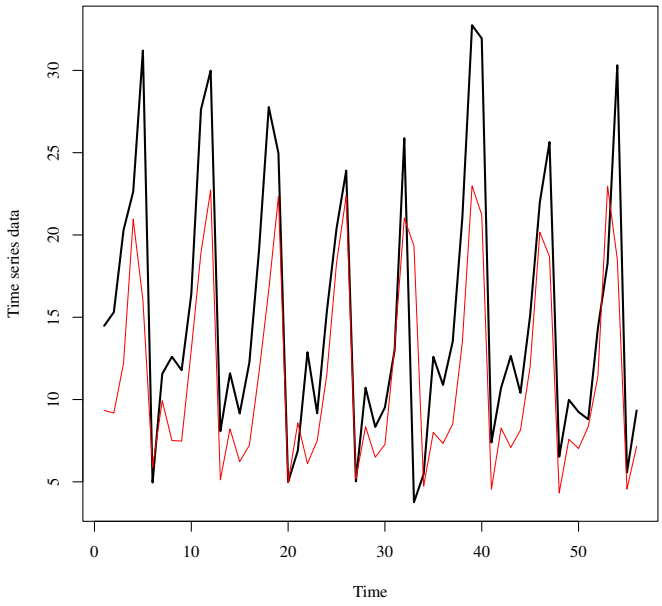

Figure 8: Forecasts of the REC strategy (normal line) versus the true values (bold line) for one of the NN5 time series.

V. Conclusion

This paper investigated the idea of perturbing a time series in order to handle more properly the approximated values and then remedy to a major limitation of the REC strategy in multi-step-ahead forecasting tasks.

A first strategy, called RECNOISY, which perturbs the dataset according to an estimate of residuals has been proposed. The experimental results, obtained on the 111 times series of the NN5 forecasting competition, show improvements in accuracy compared to the REC strategy, especially in distant horizons.

A second strategy, called HYBRID, based on selecting the best performing strategy for each horizon has been proposed and assessed. The experimental results showed that this strategy is more accurate than REC both for long and short-term forecasting.

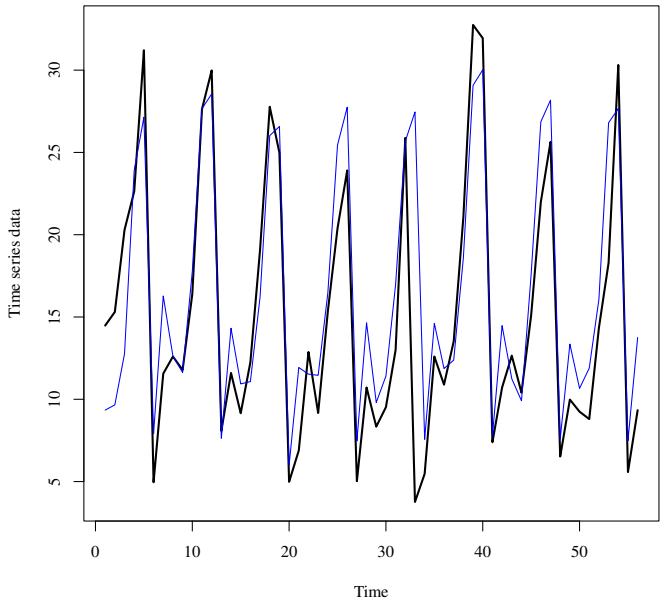

Figure 9: Forecasts of the RECNOISY strategy (normal line) versus the true values (bold line) for one of the NN5 time series.

Future work will focus on two main directions. First, we will investigate other ways of perturbing the dataset, for example by using non-parametric approaches. Second, we will handle the problem of prediction intervals to give an estimation of the uncertainty associated to the forecasts, for instance by taking advantage of the repeated sampling nature of the approach.

Acknowledgment

The authors would like to thank Yannaël Le Borgne, Catharina Olsen, Miguel Lopes and Shreyas Karnik for their comments.

References

[1] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.

[2] A. K. Palit and D. Popovic, *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[3] 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.

[4] A. S. Weigend and N. A. Gershenfeld, Eds., *Time series prediction: Forecasting the future and understanding the past*, 1994.

[5] 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, October 2007.

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

[7] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews (to appear)*, vol. 29, no. 5-6, 2010.

[8] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[9] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[10] S. Crone, "NN5 Forecasting Competition," http://www.neural-forecasting-competition.com/NN5/index.htm, last update 27/05/2009. Visited on 05/07/2010.

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

[12] D. Hand, "Mining the past to determine the future: Problems and possibilities," *International Journal of Forecasting*, October 2008.

[13] S. Price, "Mining the past to determine the future: Comments," *International Journal of Forecasting*, vol. 25, no. 3, pp. 452–455, July 2009. [Online]. Available: http://ideas.repec.org/a/eee/intfor/v25y2009i3p452-455.html

[14] S. F. Crone, "Mining the past to determine the future: Comments," *International Journal of Forecasting*, vol. 25, no. 3, pp. 456 – 460, 2009, special Section: Time Series Monitoring.

[15] D. W. Aha, Ed., *Lazy learning*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[16] M. Nelson, T. Hill, W. Remus, and M. O'Connor, "Time series forecasting using neural networks: should the data be deseasonalized first?" *Journal of Forecasting*, vol. 18, no. 5, pp. 359 – 367, 1999.

[17] A. Timmermann, "Forecast combinations," in *Handbook of Economic Forecasting*, G. Elliott, C. Granger, and A. Timmermann, Eds. Elsevier Pub., 2006, pp. 135–196.

[18] 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.

[19] 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, subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[20] R. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[21] I. Galván and P. Isasi, "Multi-step learning rule for recurrent neural models: an application to time series forecasting," *Neural processing letters*, vol. 13, no. 2, pp. 115–133, 2001.

[22] 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.

[23] G. Bontempi, M. Birattari, and H. Bersini, "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.

[24] H. Jaeger, "A tutorial on training recurrent neural networks , covering BPPT , RTRL , EKF and the " echo state network " approach," *ReVision*, vol. 2002, pp. 1–46, 2005.

[25] L. Herrera, H. Pomares, I. Rojas, a. Guillen, a. Prieto, and O. Valenzuela, "Recursive prediction for long term time series forecasting using advanced models," *Neurocomputing*, vol. 70, no. 16-18, pp. 2870–2880, Oct. 2007.

[26] D. Decoste and B. Schölkopf, "Training Invariant Support Vector Machines," *Machine Learning*, vol. 46, no. 1, pp. 161–190, 2002.

[27] "Forecasting the nn5 time series with hybrid models," *International Journal of Forecasting*, vol. In Press, Corrected Proof, pp. –, 2010.

[28] R. R. Andrawis, A. F. 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. 672–688, 2011.

[29] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. New York, NY, USA: Cambridge University Press, 2004.

[30] H. Pi and C. Peterson, "Finding the embedding dimension and variable dependencies in time series," *Neural Comput.*, vol. 6, pp. 509–520, May 1994.

[31] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*. Wiley-Interscience, 1999, vol. 2.