Ensemble Learning for Time Series Prediction

Abdelhamid Bouchachia
Dept. of Informatics-Systems
University of Klagenfurt, Austria
Email: hamid@isys.uni-klu.ac.at

Saliha Bouchachia
Group of Integrated Systems and Circuits Design
University of Applied Sciences, Villach, Austria
Email: sbouchachia@fh-kaernten.ac.at

Abstract— **This paper introduces a novel ensemble learning approach based on recurrent radial basis function networks (RRBFN) for time series prediction with the aim of increasing the prediction accuracy. Standing for the base learner in this ensemble, the adaptive recurrent network proposed is based on the nonlinear autoregressive with exogenous input model (NARX) and works according to a multi-step (MS) prediction regime. The ensemble learning technique combines various MS-NARX-based RRBFNs which differ in the set of controlling parameters. The evaluation of the approach which uses two known benchmark time series, Sunspot and chaotic Mackey-Glass, includes a discussion on the performance of the individual predictors and their combination.**

I. Introduction

A temporal sequence is a set of finite set of discrete items linked or correlated in time. Items may be scalars or vectors. Essentially time series can be classified into 3 types: deterministic, stochastic, chaotic [1]. In deterministic time series, the items and their order are specified with 100% certainty. Typically, stochastic time series result from measuring the behavior of some dynamic systems for which the observed variability has random noise as a basic component. On the other hand, while similar to stochastic time series, in chaotic time series the observed variability is not due to noise, but to nonlinear interaction among the variables of the underlying deterministic system.

In learning temporal (and spatial) sequences for predictions purposes, recurrent neural networks have attracted a lot of attention. There exist a number of studies showing different neural architectures and relying on different learning models, i.e., supervised, unsupervised and with reinforcement. While supervised learning recurrent algorithms seem to be the most popular ones [2], [3], unsupervised learning (i.e, clustering) has witnessed increasing attention especially with the advent of self-organized maps and vector quantization networks [4]. Reinforcement on the other hand has been applied for time series in a smaller number of studies [5].

Temporal relationship often are captured using feedback connection in neural networks. This has resulted in a number of architectures and have been classified in 3 main groups [6]. globally recurrent networks [2], locally recurrent networks [3] and nonlinear autoregressive with exogenous input networks (NARX networks) [7]. In the first class, hidden nodes provide a context (hidden states) and are globally fed back as new input. In locally recurrent networks, the feedback connections are allowed only from neurons to themselves (looped neurons). In the NARX architecture, the output of the network are fed back to the input layer.

In this paper we use a new adaptive NARX recurrent radial basis function network as a prototypical base learner for an ensemble learning approach. This neural network is called a MS-NARX-RRBFN standing for multi-step NARX-based recurrent radial basis function network. As will be described in Sec. II, the aim is to construct parsimonious and flexible radial basis function networks. To achieve such a gaol, the proposed neural network is equipped with multi-step-ahead prediction mechanisms and is totally self-adaptive in the sense that most of the parameters defining its architecture are learned too.

In addition to the multi-step-ahead prediction strategy endowing the radial basis function network during training, to further enhance the prediction accuracy, a neural ensemble (committee) predictor is devised. This ensemble is generated by varying the MS-NARX-RRBFN allotting different settings.

Ensemble learning has recently attracted much attention due to its ability to perform better than single learning model and to discover regularities in dynamic and non-stationary data. Ensemble methods aim at leveraging the performance of a set of models to achieve better prediction accuracy than that of the individual models. While in some literature sources, authors refer to individual models as weak learners, it is however necessary to have them as competent as possible [8]. This is the approach taken in this paper. We aim at obtaining a set of competent complementary decision makers. It is worth stressing here to note that due to the non-stationarity characterizing time series, prediction by means of committee learners is indeed a very appealing approach.

The rest of this paper is organized as follows. Section II describes the base learner used in this ensemble predictor. Section. III introduces the ensemble predictor. In Sec. IV, the evaluation of the proposed approach is discussed.

II. Recurrent Multi-Step RBFN

Like multilayer perceptron, RBF neural networks are function approximators [9] able of learning to map a given input set to its corresponding output set. In a RBF network, the hidden units form a set of functions that compose a random basis for the input patterns, hence the name of radial basis functions [10]. They serve to perform a nonlinear transformation of the patterns into a high-dimensional space in order to tackle the problem of pattern separability. An interesting development stage of RBFNs is regularization that allows to enhance the generalization of the network via interpolation mechanisms in the high-dimensional space [11].

This generalization capability, however, depends largely on the appropriateness of the model's parameters, i.e. centers, number, form, and width of the radial basis functions and the learning algorithm used to train the network. As to this latter aspect, several RBFN training schemes have been developed. The known ones include gradient descent [10] and orthogonal least square optimization [12]. Such training schemes may involve learning the RBF parameters also. In fact, the centers of the radial basis functions can be determined either by clustering (and vector quantization) [13] or can along with the radial basis widths be part of the training stage [10].

The number of radial basis functions depends on the data and should be carefully selected. It can either a priori fixed and remains static or dynamically set (i.e., centers are added or deleted) in the course of training. To avoid such a problematic, a criterion that defines the optimum number of basis functions for the RBF networks has been introduced in [14]. Such a criterion relies on Steins unbiased risk estimator to derive an analytical criterion for assigning the appropriate number of basis functions.

Moreover, there exists a set of basis functions that can be used and for which the interpolation can be achieved. These include multi-quadratic, Gaussian, inverse multi-quadratic, thin-plate spline, cubic and linear. These functions have been compared on time series in [15]. The authors recommend to try various basis functions with their range of widths to find an optimal solution.

Motivated by these considerations about the network's architecture and the diversity of heuristics used to estimate the network's parameters as discussed earlier, it seems very appealing to use ensemble learning to face such diversity and tuning problems. The ensemble method we propose in this paper relies on the NARX architecture of recurrent neural networks. Compared to globally recurrent networks, in NARX-based recurrent networks the states of the network are obtained from the output layer not from the hidden layer. In terms of complexity, NARX models are less dense since the size of hidden layer is larger than that of the output layer. In the case of time series, the output layer consists of only one neuron.

NARX-RRBFN relies on the nonlinear autoregressive model with exogenous inputs that is described by:

$$\hat{y}(t+1) = F(x(t+1), \cdots, x(t-D_x), y(t), \cdots, y(t-D_y)) \quad (1)$$

where $x(t)$ and $y(t)$ are the input and output of the nonlinear system at time t, F is a nonlinear function, D_y and D_x represent the order of the model. For time series, this model is reduced to:

$$\hat{y}(t+1) = F(y(t), y(t-1), \cdots, y(t-D)) \quad (2)$$

where D is the size of a time window. In other terms, the time series behavior can be captured by expressing the value $y(t+1)$ as a function of the D previous values of the time series, $(y(t) \cdots (y(t-D))$. Syntactically such behavior corresponds to one-step prediction which "fits" the last D samples to estimate the current value at time t. However, such a prediction scheme may not provide enough information especially if one wants to anticipate the behavior of the time series evolution.

To overcome this, NARX-RRBFN can be enhanced by embedding a multi-step predictive model that offers the possibility to handle complex dynamics over a long period of time. The idea underlying multi-step predictive model, as a generalization of the one-step model, is that predicting at time $t+1$ requires to perform p prediction steps ahead into the future, i.e. $\hat{y}(t+1), \cdots, \hat{y}(t+p+1)$. Hence, the goal is to approximate the function F such that the model given by Eq. 2 can be used as a multi-step prediction scheme.

The mathematical formulation of multi-step prediction is as follows:

$$\begin{aligned}\hat{y}(t+p+1) = F(&\hat{y}(t+p), \cdots, \hat{y}(t+1), y(t), \cdots, \\ &y(t-D+p))\end{aligned} \quad (3)$$

where p is called prediction horizon. Basically this formulation can be unfolded as follows:

$$\begin{cases}\hat{y}(t+1) = F(y(t), \cdots, y(t-D)) \\ \hat{y}(t+2) = F(\hat{y}(t+1), y(t), \cdots, y(t-D+1)) \\ \quad\cdots = \cdots \\ \hat{y}(t+p+1) = F(\hat{y}(t+p), \cdots, \hat{y}(t+1), \\ \qquad\qquad y(t), \cdots, y(t-D+p))\end{cases} \quad (4)$$

which suggests that at any time t, predictions have to be made based on the time interval $[t+1, t+p+1]$ taking D samples as input. Such input is split into two parts: context input and external input. The context input stands for the internal states of the network which are obtained from the delayed network output. They memorize the context of the current input by recalling information about the past. This context provides the network the ability to handle long-term predictions. Indeed for the sake of long-term predictions, the multi-step approach adopted here allows the network to take future sample change over a prediction horizon into account. The external input represents the last samples seen by the network preceding the current time t. Initially these samples act as a window that represents the historical trend obtained directly from the data. This means that the predicted network output $\hat{y}(t+1)$ at instant $t+1$ is sent back as input for the next step prediction. The remaining input part corresponds to the input values shifted ahead by one sample.

Graphically the multi-step NARX-RBFN is portrayed in Fig. 1 and its unfolding architecture is portrayed as a cascade of RBFNs in Fig. 2.

Basically the function F in Eq. 3 has the form of

$$\begin{aligned}\hat{y}(t+p+1) = F(&\hat{y}(t+p), \cdots, \hat{y}(t+1), y(t), \cdots, \\ &y(t-D+p), \Theta)\end{aligned} \quad (5)$$

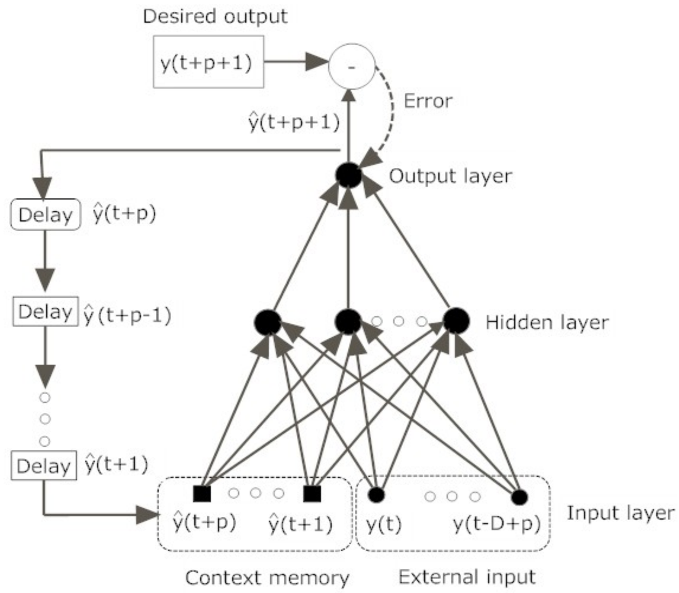

Fig. 1. Recurrent radial basis function with multi-step learning

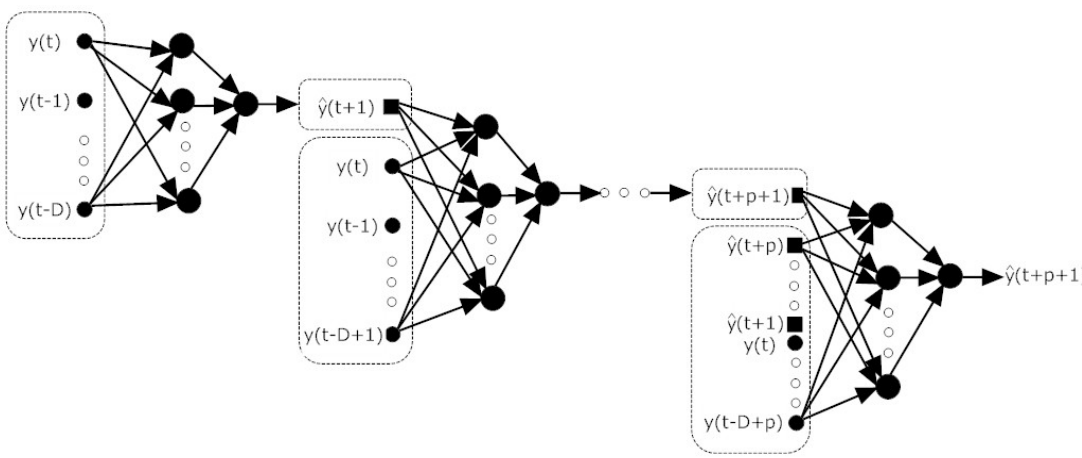

Fig. 2. Unfolding the multi-step NARX-RRBFN

where Θ is the parameter set of the model (C_i, Σ_i, W) which stands for the centers and widths of the radial basis functions and the weights between the hidden and the output layers.

To define $\Theta = \{v_j, \sigma_j, w_j\}$, the following performance index has to be minimized:

$$Q(t+1) = \frac{1}{2}\sum_{i=1}^{p}\left(y(t+i+1) - \hat{y}(t+i+1)\right)^2 \qquad (6)$$

standing for the multi-step prediction error.

Since, the present work is about adaptive recurrent radial basis function network, the weights, centers, and widths of the radial basis functions are updated using the following gradient descent rules:

$$\frac{\partial Q}{\partial \Theta} : \frac{\partial Q}{\partial v_j}, \frac{\partial Q}{\partial \sigma_j}, \frac{\partial Q}{\partial w_j} \qquad (7)$$

Recall that the output node is a linear combination of a set of basis functions:

$$\hat{y}(x_i) = \sum_{j=1}^{H} w_j \phi_j(x_i) \qquad (8)$$

where x_i is the input vector with elements x_{im} (where m is the dimension of the input vector); v_j is the center vector of the basis function $\phi_j(.)$ with elements v_{ji}; w_j are the output layer's weights. The hidden nodes equipped with the basis function $\phi_j(x_i)$ are nonlinear, while those of the output are linear.

The radial basis function takes different forms which we will use in ensemble learning approach intended in this study. These forms are shown in Tab. I. For the sake of simplicity we omit the index (t+1) from Eq. 7. Therefore, $\hat{y}(t+1+i)$ (resp.

TABLE I

RADIAL BASIS FUNCTIONS

Function	Form
Gaussian	$e^{(-\frac{\|\|y-v_j\|\|^2}{2\sigma_j^2})}$
Multiquadratic	$(\|\|y-v_j\|\|^2+\sigma_j^2)^{\frac{1}{2}}$
Inverse multiquadratic	$(\|\|y-v_j\|\|^2+\sigma_j^2)^{-\frac{1}{2}}$
Cubic	$\|\|y-v_i\|\|^3$

$y(t+1+i))$ will be written $\hat{y}_i$ (resp. y_i). Then the following holds:

$$\frac{\partial Q}{\partial w_j} = \frac{\partial Q}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial w_j} = -(y_i - \hat{y}_i)\phi_j \tag{9}$$

Hence, the weight can be updated as follows:

$$w_j = w_j - \eta_1 \frac{\partial Q}{\partial w_j} = w_j + \eta_1 (y_i - \hat{y}_i)\phi_j \tag{10}$$

To update the centers we need to compute:

$$\frac{\partial Q}{\partial v_j} = \frac{\partial Q}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial \phi_j}\frac{\partial \phi_j}{\partial v_j} = -w_j(y_i - \hat{y}_i)\frac{\partial \phi_j}{\partial v_j} \tag{11}$$

leading to the following update rule:

$$v_j = v_j - \eta_2 \frac{\partial Q}{\partial v_j} = v_j + \eta_2 w_j (y_i - \hat{y}_i)\frac{\partial \phi_j}{\partial v_j} \tag{12}$$

The last update operation is that of the width which requires:

$$\frac{\partial Q}{\partial \sigma_j} = \frac{\partial Q}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial \phi_j}\frac{\partial \phi_j}{\partial \sigma_j} = -w_j(y_i - \hat{y}_i)\frac{\partial \phi_j}{\partial \sigma_j} \tag{13}$$

leading to the following update rule:

$$\sigma_j = \sigma_j - \eta_3 \frac{\partial Q}{\partial \sigma_j} = \sigma_j + \eta_3 w_j (y_i - \hat{y}_i)\frac{\partial \phi_j}{\partial \sigma_j} \tag{14}$$

The MS-NARX-RRBF learning algorithm consists of the steps shown in Alg. 1. For a particular base learner, the architecture of the RRBF is kept fixed but the input layer is dynamic. Just recall that at time t, the algorithm must predict the time series values at instants $t+1,\cdots,t+p+1$ during which the number of external input nodes decreases from $D+1$ to $D+1-p$ while the number of context neurones increases from 0 to p. Thus, initially the number of context nodes is 0 and the external nodes correspond to the input indexed by $t,\cdots,t-D$. Generally, at time $t+i$ to predict the future $t+i+1^{th}$ time series sample, the context nodes receive output corresponding to the predictions realized in interval $[t+1, t+1+i-1 = t+i]$ (i.e, the number of context nodes is $(t+i)-(t+1) = i-1$, whereas the external input nodes correspond to the time series samples indexed by time interval $[t-D+i-1, t]$ (since the window is of length $D+1$). Note that to learn the last p training samples of the time series, the number of context nodes will not exceed $T-i$ (since $t+i \leq T$ must hold, where T is the size of the training data). On the other hand, in this study p is set to value less $D+1$, so that we have at least one external input, otherwise all neurons in the input layer will be context nodes.

Algorithm 1 : Training the multi-step NARX-RRBFN

- Initialize η_1, η_2, η_3, the size of the hidden layer H, the prediction horizon p, the window D (such that $p <= D$ to avoid learning exclusively from future predictions).

repeat

- Set the initial input window $[y_1,\cdots,y_D]$
- Initialize the network: radial basis function type, initial centers (V), width (σ) and the weights (W)

for $t = D+1$ to $T-1$ **do**

- Compute $\hat{y}(t+1) = F(y(t),\cdots,y(t-D))$

for $i = 1$ to p **do**

- Recursively predict the output $\hat{y}(t+i+1)$ of the current configuration of the NARX-RRBFN using:

$$\begin{aligned}\hat{y}(t+i+1) &= F(\hat{y}(t+i),\cdots,\hat{y}(t+1),\\ &\quad y(t),\cdots,y(t-D+i-1))\end{aligned}$$

- Update the parameter set $\Theta = \{V, \Sigma, W\}$ according to Eqs. 12,14, 10 respectively
- Update the input sequence at the input layer

end for

end for

until Stopping criterion is met

III. ENSEMBLE LEARNING

Radial basis function neural networks are universal nonlinear function approximators with a controllable complexity. They are known for their prediction power. However, due to the diversity and the definition range of their parameters, the performance of these neural networks may vary strongly. To alleviate the effect of parameter setting, it seems appealing to combine in a symbiotic way several predictors. The idea is that even if the performance of one or few neural networks may not be that much satisfactory, the ensemble of the algorithms can still predict the correct output. Usually, when the task is relatively hard, multiple predictors are used following the conquer-and-divide principle [16].

Ensemble learning has been mostly applied for classification problems. However, recently a certain number of studies propose their application for time series forecasting problems. For instance, an ensemble learning based on feedforward neural networks has been proposed in [17] for time series forecasting. In [18], the ensemble combines radial basis function networks and the Box-Jenkins models. In [19], a combination genetic classifiers is proposed for predicting stock indexes, while in [20], a hybrid combination of neural networks and the ARIMA model is applied for time series forecasting. In [21], an ensemble of Elman networks combined by Adaboost is proposed for predicting drug dissolution profiles. Similar work has been investigated in [22] relying on Adaboost and its variants such as Adaboost.R proposed in [23].

It is important to note that most of the studies rely on one scheme that is classifier combination trained on different data sets. In this scheme several classifiers, each trained on randomly generated sets (re-sampling from a larger training set)

TABLE II

DIVERSITY CRITERIA

	Parameter	Value set
1-	Learning	{Gradient descent}
2-	Number of RBFs	{Fitting mixture of Gaussians using EM)}
3-	Type of RBFs	{Multi-quadratic, Gaussian, inverse multi-quadratic, cubic}
4-	Width of RBFs	{Gradient descent}
5-	Center of RBFs	{Gradient descent}
6-	RBFN Architecture	{Globally recurrent}

TABLE III

COMBINATION RULES

Rule	Expression
Average Rule	$O^j(x) = \frac{1}{N}\sum_{i=1}^{N} O_i^j(x)$
Max Rule (optimistic)	$O^j(x) = max_{i=1}^{N} O_i^j(x)$
Min Rule (pessimistic)	$O^j(x) = \min_{i=1}^{N} O_i^j(x)$

are combined to perform the classification or regression task. These include stacking [24], bagging [25] and boosting [23].

In this study we rather focus on a different scheme that is *combination of different classifiers* [16], [26]. According to this scheme, the classifier ensemble contains several classifiers of different types (neural networks, decision trees, etc.), of different parameters (e.g. in multi-layer neural networks: different number of hidden layers, different number of hidden neurons, etc.), or trained using different initial conditions (e.g. weight initialization in neural networks, etc.) The application of such scheme is not well studied in the context of time series. For instance, in [26], an ensemble learning model using the fuzzy k-nearest neighbor classifier as a base classifier is proposed. K-nearest neighbor classifier is also used in [27] but combined with multi-layer perceptron, nearest trajectory models and some polynomial models.

In our study we focus on recurrent radial basis functions adapted to the NARX architecture and working in a multi-step prediction regime shown in Alg. 1. This ensemble of recurrent neural networks are mainly diversified according to the architecture, type of radial basis function, the initialization of the weights, and the initial position of the radial basis functions (see Tab. II). As already mentioned, there exist many ways the individual NARX-RBFFN can be combined. In the present study, we consider the rules shown in Tab. III.

IV. NUMERICAL SIMULATIONS

A. Benchmarks

In this study, we will rely on two major data sets to evaluate the proposed approach. These are the sunspots and the chaotic Mackey-Glass time-series datasets. The former contains the yearly number of dark spots on the sun from 1700 to 1979. The time series has a pseudo-period of 10 to 11 years. In many studies, the training set includes the time series from

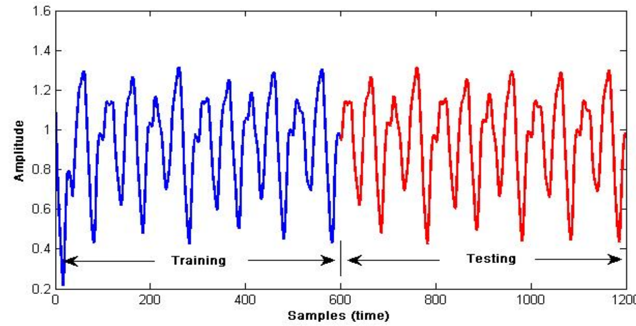

Fig. 3. Mackey-Glass time series

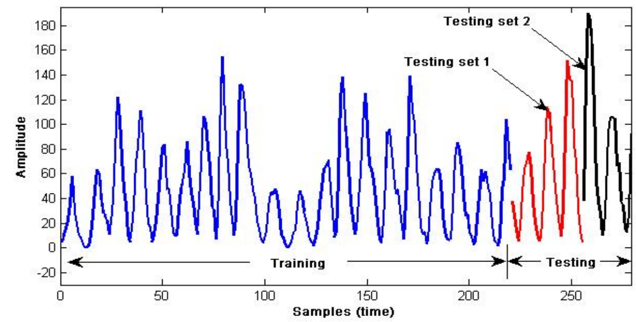

Fig. 4. Sunspot time series

1700 to 1920, while the testing set consists of two subsets, 1921-1955 (test1) to be used here and 1956-1979 (test2). The chaotic Mackey-Glass time-series are generated by the following nonlinear differential equation:

$$\frac{dx(t)}{dt} = -0.1 * x(t) + \frac{0.2 * x(t-\tau)}{1 + x^{10}(t-\tau)} \tag{15}$$

The initial conditions used in our test bench are set as $x(0) = 0.8$ and $t = 17$. These are set so in order to conduct a comparative study against other approaches using the same benchmarks.

B. Experiments

To assess the proposed approach, we study two aspects: (i) the prediction accuracy of the individual networks, (ii) the accuracy of their combination following the the four types of radial basis functions on both data sets: the sunspots and the chaotic Mackey-Glass time-series. For the sake of the evaluation, the root mean squared error (RMSE) measure is used to quantify the goodness-of-fit. It is given by:

$$RMSE = \sqrt{\frac{\sum_i^N (y(i) - \hat{y}(i))^2}{N}} \tag{16}$$

Before starting the evaluation of each of the individual NARX-RBFNN, it is important to check the effect the key parameters that characterize the proposed NARX architecture. Basically, these parameters include the time window size (D)

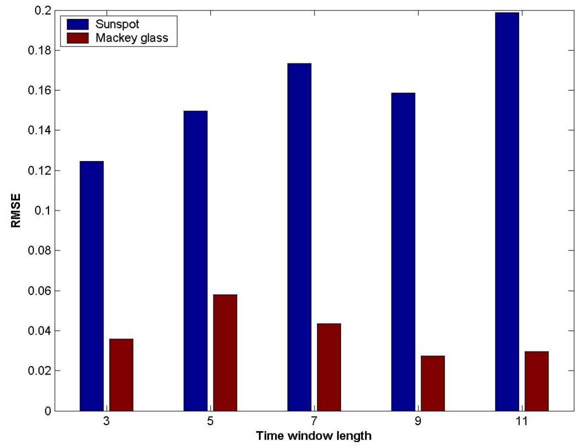

Fig. 5. The effect of window size

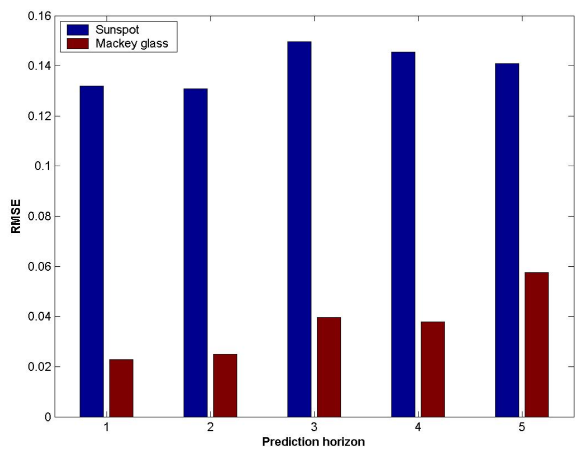

Fig. 6. The effect of the prediction horizon

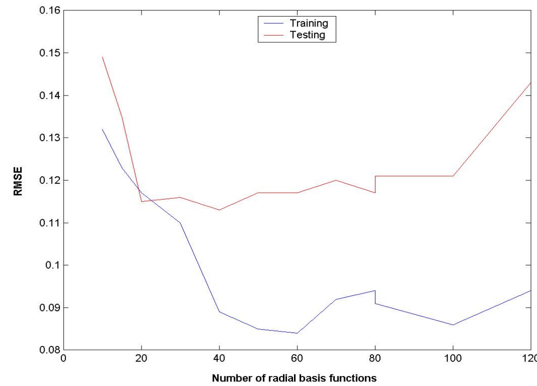

Fig. 7. The effect of cluster number - sunspot

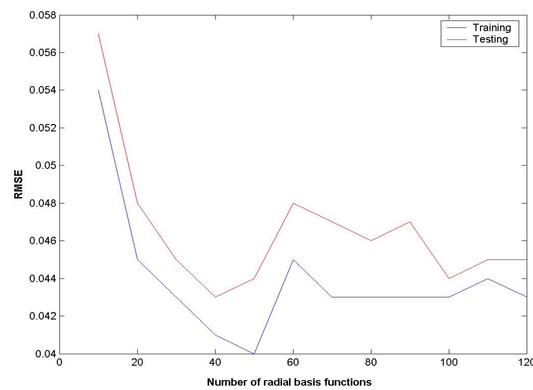

Fig. 8. The effect of the cluster number - Mackey glass

used for training the networks and the prediction horizon (p), and the number of radial basis functions. Figures 5 and 6 show the effect of the two parameters on the prediction root mean square error for both data sets.

The observed trend from these figures is that for the sunspot time series, small window size is preferred. In fact, when $D = 3$, a least root mean square error is obtained. In the case of Mackey glass data set, there is no clear trend window size, though larger sizes seem to be preferred. On the other hand, small size of the prediction horizon is preferred for both data sets. These results are consistent with the variability of time series. Indeed, the variability of Mackey-glass, that is 0.0556, is far smaller than that of sunspot (1554.20). Therefore, it is important to set the window time in the case of sunspot also smaller than that applied to the Mackey-glass time series.

This experiment has allowed to estimate the near-optimum D and p(in a certain range of values). In the next experiments, we will consider $D_{sunpot} = 3$, $D_{Mackey} = 9$, $p_{sunpot} = 2$, and $p_{Mackey} = 2$.

On the other hand, to find the optimal number of radial basis functions used by the neural networks, we have used the Bayesian Information Criterion (BIC) to judge the statistical significance of a number of inspected finite mixture models. We do that by relying on the expectation maximization algorithm. The highest BIC value corresponds to the optimal number of radial basis function.

As expected, the BIC increases as the number of clusters increases. However, a significant increase is obtained after setting the number of clusters to 48 in the case of Mackey glass data and 56 for the sunspot data. This has also been noticed when computing the root mean square error as shown in Figs. 7 and 8 for the case of Gaussian radial basis function as an illustrative example. Therefore, we have considered 60 and 50 clusters for sunspot and Mackey glass respectively.

Applying the set of NARX-RBFNNs under the optimal conditions (i.e., optimal number of radial basis functions, size of the time window, size of the prediction horizon) on the training and testing data of both data sets, we obtain Figs. 9, 10, 11, and 12. These show how good the fit is.

TABLE IV
COMBINING THE NARX-RBFNNS (RMSE)

		Sunspot		Mackey Glass	
		Train	Test	Train	test
Radial basis functions	Gaussian	1.2613	0.2603	0.5119	0.6782
	Multiquadratic	1.3660	0.2653	0.6636	1.0302
	Inverse multiquadratic	1.3350	0.2643	0.7696	1.1985
	Cubic	1.4385	0.2654	0.6590	1.2293
Combination rules	Max rule	1.3551	0.2399	0.7277	1.1631
	Min Rule	1.4010	0.2784	0.7240	1.1866
	Average	**1.2401**	**0.2231**	**0.5040**	**0.5999**

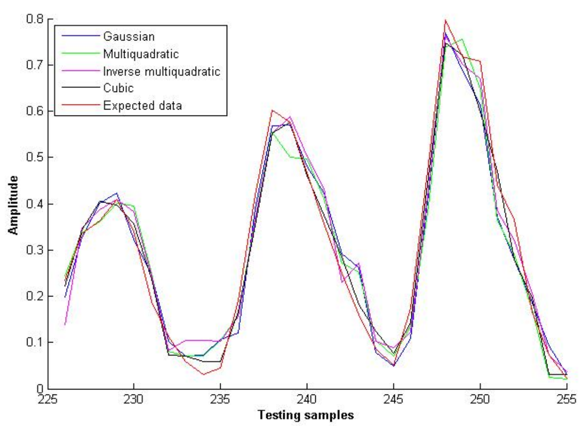

Fig. 10. Testing the individual NARX-RBFNNs - sunspot

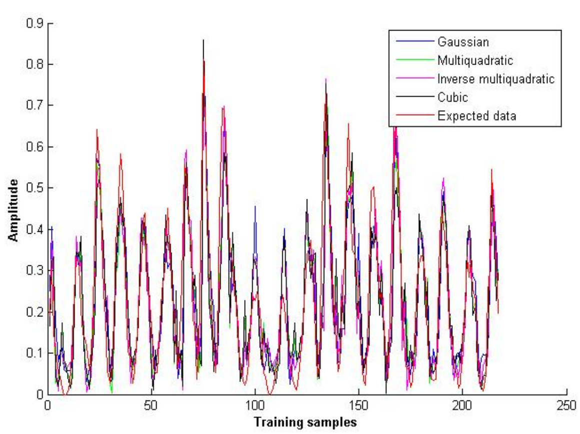

Fig. 9. Training the individual NARX-RBFNNs - sunspot

The combination of the four NARX-RBFNNs according to the combination rules portrayed in Tab. III yields the results shown in Tab. IV with respect to both data sets. The accuracy of the ensemble is much higher compared to each of the individual predictor. The average rule is the best combination rule. Comparing the individual NARX-RBFNNS on these two particular time series, it seems that the Gaussian radial basis

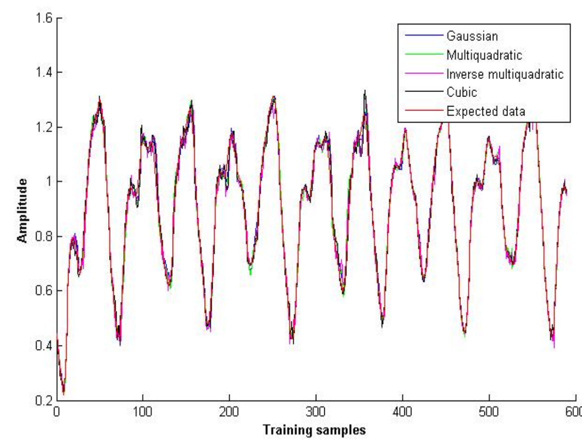

Fig. 11. Training the individual NARX-RBFNNs - Mackey glass

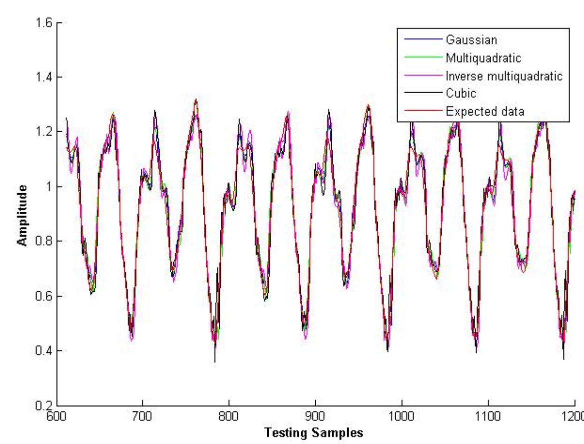

Fig. 12. Testing the individual NARX-RBFNNs - sunspot

function produces more accurate fitting.

V. Conclusion

The present paper deals with a new method of time series predictions based on multiple predictors. Each of these is a multi-step nonlinear autoregressive with exogenous input model (NARX) radial basis function network. Relying n two time series, the experiments have shown that the combination improves the prediction accuracy. However, it is still necessary to conduct further experiments with a larger sets of time series in order to thoroughly assess the approach suggested in this study.

References

[1] A. Barreto, A. A. Araujo, and S. Kremer, "A taxonomy for spatiotemporal connectionist networks revisited: the unsupervised case," *Neural Computation*, vol. 15, pp. 1255–1320, 2003.

[2] S. Kremer, "Spatiotemporal connectionist networks: A taxonomy and review," *Neural Comput.*, vol. 13, no. 2, pp. 249–306, 2001.

[3] P. Frasconi, M. Gori, and G. Soda, "Local feedback multilayered networks," *Neural Comput.*, vol. 4, no. 1, pp. 120–130, 1992.

[4] A. Araújo and G. Barreto, "Context in temporal sequence processing: A self-organizing approach and its application to robotics," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 45–57, 2002.

[5] F. Liu, G. Ng, and C. Quek, "Rldde: A novel reinforcement learning-based dimension and delay estimator for neural networks in time series prediction," *Neurocomput.*, vol. 70, no. 7-9, pp. 1331–1341, 2007.

[6] T. Lin, B. Horne, P. Tiño, and C. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996.

[7] J. Menezes and G. A. Barreto, "A new look at nonlinear time series prediction with narx recurrent neural network," in *SBRN '06: Proceedings of the Ninth Brazilian Symposium on Neural Networks*, 2006, p. 28.

[8] A. Bouchachia, "Learning with partly labeled data," *Neural Computing and Applications*, vol. 16, no. 3, pp. 267–293, 2007.

[9] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.

[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[11] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, 1990.

[12] G. Fasshauer, *Mathematical Methods for Curves and Surfaces*. Vanderbilt Univ. Press, 1995, ch. Adaptive least squares fitting with radial basis functions on the sphere, pp. 141–150.

[13] J. Moody and C. Darken, "Fast learning in networks of locally tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[14] A. Ali Ghodsi and D. Schuurmans, "Automatic basis selection techniques for rbf networks," *Neural Networks*, vol. 16, no. 5-6, pp. 809–816, 2003.

[15] C. Harpham and C. Dawson, "The effect of different basis functions on a radial basis function network for time series prediction: A comparative study," *Neurocomputing*, vol. 69, no. 16–18, pp. 2161–2170, 2006.

[16] L. Kuncheva, J. Bezdek, and R. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[17] D. Karunasinghea and S. Liongb, "Chaotic time series prediction with a global model: Artificial neural network," *Journal of Hydrology*, vol. 323, no. 1-4, pp. 92–105, 2006.

[18] D. Wedding II and K. Cios, "Time series forecasting by combining rbf networks, certainty factors, and the box-jenkins model," *Neurocomputing*, vol. 10, no. 2, pp. 149–168, 1996.

[19] G. Armano, M. Marchesi, and A. Murru, "A hybrid genetic-neural architecture for stock indexes forecasting," *Inf. Sci.*, vol. 170, no. 1, pp. 3–33, 2005.

[20] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L. Herrera, A. Guillen, L. Marquez, and M. Pasadas, "Hybridization of intelligent techniques and arima models for time series prediction," *Fuzzy Sets Syst.*, vol. 159, no. 7, pp. 821–845, 2008.

[21] W. Goh, C. Lim, and K. Peh, "Predicting drug dissolution profiles with an ensemble of boosted neural networks: a time series approach," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 459–463, 2003.

[22] M. Assaad, R. Boné, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Inf. Fusion*, vol. 9, no. 1, pp. 41–55, 2008.

[23] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, 1995, pp. 23–37.

[24] D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[25] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[26] C. Christian Dietrich, G. Palm, and F. Schwenker, "Decision templates for the classification of bioacoustic time series," *Information Fusion*, vol. 4, no. 2, pp. 101–109, 2003.

[27] J. Wichard and M. Ogorzalek, "Time series prediction with ensemble models applied to the cats benchmark," *Neurocomputing*, vol. 70, no. 13-15, pp. 2371–2378, 2007.

Abdelhamid Bouchachia is affiliated with the group of Software Engineering and Soft Computing, Dept. of Informatics, University of Klagenfurt, Austria, as Associate Prof. His current research interests cover a range of soft computing and machine learning techniques encompassing data and information processing, incremental learning, semi-supervised learning, prediction systems, nature-inspired computing, processing and modeling of uncertainty in pattern recognition and rule based systems.

Saliha Bouchachia, is affiliated as a research assistant with the Group of Integrated Systems and Circuits Design, University of Applied Sciences, Villach, Austria. Her research interests cover mobile multicast receivers, microwaves, transmission lines and analog design.