On Recurrent Neural Networks for Auto-Similar Traffic Prediction: A Performance Evaluation

José M. P. Menezes and Guilherme A. Barreto

Abstract— **The NARX network is a recurrent neural architecture commonly used for input-output modelling of nonlinear systems. The input of the NARX network is formed by two tapped-delay lines, one sliding over the input signal and the other one over the output signal. Currently, when applied to nonlinear time series prediction, the NARX architecture is designed as a plain Focused Time Delay Neural Network (FTDNN); thus, limiting its predictive abilities. In this paper, we propose a strategy that allows the original NARX architecture to fully exploit its computational resources to improve prediction performance. We use real-world VBR video traffic time series to evaluate the proposed approach in multi-step-ahead prediction tasks. The results show that the proposed approach consistently outperforms standard neural network based predictors, such as the FTDNN and Elman architectures.**

Index Terms— **Recurrent neural networks, traffic prediction, auto-similar processes, VBR video traffic, multi-step-ahead prediction.**

I. Introduction

Artificial neural networks (ANNs) have been successfully used as a tool for time series prediction and modeling in a variety of application domains, including financial time series prediction [1], river flow forecasting [2], biomedical time series modeling [3] and network traffic prediction [4], [5], [6], just to mention a few. Usually, ANN models outperform traditional linear techniques, such as the well-known Box-Jenkins models [7], when the time series are noisy and nonlinear. In such cases, the universal approximation and generalization abilities of ANN models seems to justify their better prediction performance.

In nonlinear time series prediction, ANN models are commonly used as one-step-ahead predictors, estimating only the next value of a time series without feeding the predicted value back to the model's input regressor. In other words, the input regressor contains only actual sample points of the time series. If the user is interested in a wider prediction horizon, a procedure known as multi-step-ahead prediction, the model's output should be fed back to the input regressor for a fixed but finite number of time steps. In this case, the input regressor's components, previously composed of actual sample points of the time series, are gradually replaced by predicted values as time goes by.

If the prediction horizon tends to infinity, from some moment in time on, the input regressor will start to be composed only of previous estimated values of the time series. In this case, the multi-step-ahead prediction task becomes a *dynamic modeling* task, in which the ANN model acts as an autonomous system, trying to recursively emulate the dynamic behavior of the system that generated the nonlinear time series [8]. Multi-step ahead prediction and dynamic modelling are much more complex to deal with than one-step-ahead prediction, and it is believed that these are complex tasks in which ANN models play an important role, in particular those related to recurrent neural architectures [9].

Recurrent ANNs have local and/or global feedback loops in their structure. Even though feedforward MLP-like networks can be easily adapted to process time series through an input tapped delay line, giving rise to the well-known Focused Time Delay Neural Network (FTDNN), they can also be easily converted to simple recurrent architectures by feeding back the neuronal outputs of the hidden or output layers, giving rise to Elman and Jordan networks, respectively [10]. Recurrent neural networks (RNNs) are capable to represent arbitrary nonlinear dynamical mappings [11], such as those commonly found in nonlinear time series prediction tasks.

The previously described neural architectures are usually trained through the standard backpropagation algorithm. However, learning to perform tasks in which the temporal dependencies present in the input/output signals span long time intervals can be quite difficult using gradient descent [12]. In [13], the authors reported that learning such long-term temporal dependencies with gradient-descent techniques is more effective in a class of recurrent ANN architecture called *Nonlinear Autoregressive with eXogenous input* (NARX) [14] than in simple MLP-based recurrent models. This occurs in part because the NARX model's input vector is cleverly built by means of a tapped-delay line sliding over the input signal together with another tapped-delay line over the network's output.

Despite the aforementioned advantages of the NARX network, its application to univariate time series prediction has been misdirected. In this type of application, the tapped-delay line over the output signal is eliminated, thus reducing the NARX network to a plain FTDNN architecture. Considering this under-utilization of the NARX network, we propose a simple strategy based on Takens' embedding theorem to allow the computational abilities of the original NARX network to be fully exploited in computer network traffic modelling and prediction.

The remainder of the paper is organized as follows. In Section II, we briefly describe the NARX recurrent network model and its main characteristics. In Section III we describe the basics of the nonlinear time series prediction problem and introduce our approach. The simulations and discussion of results are presented in Section IV. The paper is concluded

The authors are with the Department of Teleinformatics Engineering, Federal University of Ceará, C.P. 6007, CEP: 60555-760, Fortaleza-CE, Brazil. email: guilherme@deti.ufc.br

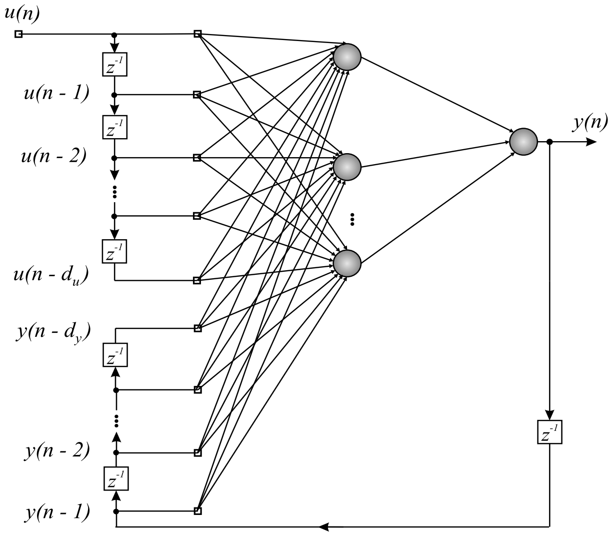


Fig. 1. A NARX network with d_u input and d_y output delays.

in Section V

II. The NARX Network

The Nonlinear Autoregressive model with Exogenous inputs (NARX) model is an important class of discrete-time nonlinear systems that can be mathematically represented as follows [15], [16]:

$$\begin{aligned} y(n+1) &= f[y(n), \ldots, y(n-d_y+1); \\ &\qquad u(n), u(n-1), \ldots, u(n-d_u+1)] \end{aligned} \tag{1}$$

or, in a compact form:

$$y(n+1) = f[\mathbf{y}(n); \mathbf{u}(n)] \tag{2}$$

where $u(n) \in \mathbb{R}$ and $y(n) \in \mathbb{R}$ denote, respectively, the input and output of the model at discrete time n, while $d_u \geq 1$ and $d_y \geq 1$, $d_u \leq d_y$, are the input-memory and output-memory orders. The vectors $\mathbf{y}(n)$ and $\mathbf{u}(n)$ denote the output and input regressors, respectively. Figure 1 shows the topology of an one-hidden-layer NARX network.

The function $f(\cdot)$ is a (generally unknown) nonlinear function which should be approximated. When this is done by a multilayer Perceptron (MLP), the resulting topology is called a *NARX recurrent neural network* [17], [11]. This is a powerful class of dynamical models which has been shown to be computationally equivalent to Turing machines [18]. The NARX network is trained basically under one out of two modes:

- **Series-Parallel (SP) Mode** - In this case, the output's regressor is formed only by actual values of the system's output:

$$\begin{aligned} \widehat{y}(n+1) &= \widehat{f}[\mathbf{y}_{sp}(n); \mathbf{u}(n)] \\ &= \widehat{f}[y(n), \ldots, y(n-d_y+1); \\ &\qquad u(n), u(n-1), \ldots, u(n-d_u+1)] \end{aligned} \tag{3}$$

- **Parallel (P) Mode** - In this case, estimated outputs are fed back and included in the output's regressor:

$$\begin{aligned} \widehat{y}(n+1) &= \widehat{f}[\mathbf{y}_{p}(n); \mathbf{u}(n)] \\ &= \widehat{f}[\widehat{y}(n), \ldots, \widehat{y}(n-d_y+1); \\ &\qquad u(n), u(n-1), \ldots, u(n-d_u+1)] \end{aligned} \tag{4}$$

It is worth noting that the feedback pathway shown in Figure 1 is present only in the Parallel Identification Mode. As a tool for nonlinear system identification, the NARX network has been successfully applied to a number of real-world input-output modelling problems, such as heat exchangers, waste water treatment plants, catalytic reforming systems in a petroleum refinery and nonlinear time series prediction.

Of particular interest for this paper is the issue of nonlinear time series prediction with the NARX network. In this type of application, the output-memory order is set $d_y = 0$, thus reducing the NARX network to a plain FTDNN architecture [19]:

$$\begin{aligned} y(n+1) &= f[\mathbf{u}(n)] \\ &= f[u(n), u(n-1), \ldots, u(n-d_u+1)] \end{aligned} \tag{5}$$

where $\mathbf{u}(n) \in \mathbb{R}^{d_u}$ is the input regressor. This simplified formulation of the NARX network eliminates a considerable portion of its representational capabilities as a recurrent network; that is, all the dynamic information that could be learned from the past memories of the output (feedback) path is discarded. For many practical applications, such as self-similar traffic modelling [20], the network must be able to robustly store information for a long period of time in the presence of noise.

It is worth emphasizing that the original formulation of the NARX network does not circumvent the problem of long-term dependencies, but it has been demonstrated that it often performs much better than standard recurrent ANNs in such a class of problems, achieving much faster convergence and better generalization performance [14]. However, if the output memory is fully discarded as in Equation (5) these properties may no longer be observed. Considering this limited use of the potentialities of the NARX network, we propose a simple strategy to allow the computational abilities of the NARX network to be fully exploited in nonlinear time series prediction tasks.

III. Nonlinear Time Series Prediction with NARX

For a better understanding of the proposed approach for nonlinear time series prediction using the NARX network, we give a brief description of the theory of embedding. For further details on this theory, the interested reader are referred to [21], [22], [23].

The state of a deterministic dynamical system is the information necessary to determine the entire future evolution of the system. In discrete time, this evolution can be described by the following system of difference equations:

$$\mathbf{x}(n+1) = \mathbf{F}[\mathbf{x}(n)] \tag{6}$$

where $\mathbf{x}(n) \in \mathbb{R}^d$ is the state of the system at time n, and $\mathbf{F}[\cdot]$ is a nonlinear vector valued function. A time series is a set of measures $\{x(n)\}$, $n = 1, \ldots, N$, of a scalar quantity observed at the output of the system over time. This observable quantity is defined in terms of the state $\mathbf{x}(n)$ of the underlying system as follows:

$$x(n) = h[\mathbf{x}(n)] + \varepsilon(t) \tag{7}$$

where $h(\cdot)$ is a nonlinear scalar-valued function, ε is a random variable which accounts for modelling uncertainties and/or measurement noise. It is commonly assumed that $\varepsilon(t)$ is drawn from a Gaussian white noise process. It can be inferred immediately from Equation (7) that the observations $\{x(n)\}$ are seen as a projection of the multivariate state space of the system onto the one-dimensional space. Equations (6) and (7) describe together the state-space behavior of the dynamical system.

In order to perform prediction, one needs to reconstruct (estimate) as well as possible the state space of the system using the information provided by $\{x(n)\}$. Takens [24] has shown that, under very general conditions, the state of a deterministic dynamic system can be accurately reconstructed by a time window of finite length sliding over the observed time series as follows:

$$\mathbf{x}_1(n) \triangleq [x(n), x(n-\tau), \ldots, x(n-(d_E-1)\tau)] \tag{8}$$

where $x(n)$ is the value of the time series at time n, d_E is the embedding dimension and τ is the embedding delay. Equation (7) implements the delay embedding theorem. This theorem motivates the technique of using time-delay coordinate reconstruction in reproducing the phase space of an observed dynamical system; that is, a collection of time-lagged values in a d_E-dimensional vector space will provide sufficient information to reconstruct the states of the dynamical system. Thus, the purpose of time-delay embedding is to unfold the projection back to a multivariate state space that is representative of the original system.

The embedding theorem provides a sufficient condition for choosing the embedding dimension d_E large enough so that the projection is theoretically able to reconstruct the original state space. This theorem also provides a theoretical framework for nonlinear time series prediction, where the predictive relationship between the current state $\mathbf{x}_1(t)$ and the next value of the time series is given by the following equation:

$$x(n+1) = g[\mathbf{x}_1(n)] \tag{9}$$

Once the embedding dimension d_E and delay τ are chosen, one remaining task is to approximate the mapping function $g(\cdot)$. It has been shown that a feedforward neural network with enough neurons is capable of approximating any nonlinear function to an arbitrary degree of accuracy. Thus, it can provide a good approximation to the function $g(\cdot)$ by implementing the following mapping:

$$\widehat{x}(n+1) = \widehat{g}[\mathbf{x}_1(n)] \tag{10}$$

where $\widehat{x}(n+1)$ is an estimate of $x(n+1)$ and $\widehat{g}(\cdot)$ is the corresponding approximation of $g(\cdot)$. The estimation error, $e(n+1) = x(n+1) - \widehat{x}(n+1)$, is commonly used to evaluate the quality of the approximation.

If we assume $\mathbf{u}(n) = \mathbf{x}_1(n)$ and $y(n+1) = x(n+1)$ in Equation (5), then it leads to an intuitive interpretation of the nonlinear state-space reconstruction procedure as equivalent to the time series prediction problem whose the goal is to compute an estimate of $x(n+1)$. Thus, the only thing we have to do is to train a FTDNN model [9]. Once training is completed, the FTDNN can be used for predicting the next samples of the time series.

Despite the correctness of the FTDNN approach, recall that it is derived from a simplified version of the NARX network by eliminating the output memory. In order to use the full computational abilities of the NARX network for nonlinear time series prediction, we propose novel definitions for its input and output regressors. Firstly, the input signal regressor, denoted by $\mathbf{u}(n)$, is defined by the delay embedding coordinates of Equation (8):

$$\begin{aligned}\mathbf{u}(n) &= \mathbf{x}_1(n) \qquad (11)\\ &= [x(n), x(n-\tau), \ldots, x(n-(d_E-1)\tau)]\end{aligned}$$

where we set $d_u = d_E$. In words, the input signal regressor $\mathbf{u}(n)$ is composed of d_E actual values of the observed time series, separated from each other of τ time steps.

Secondly, since the NARX network can be trained in two different modes, the output signal regressor $\mathbf{y}(n)$ can be written as follows:

$$\mathbf{y}_{sp}(n) = [x(n), \ldots, x(n-d_y+1)] \tag{12}$$

$$\mathbf{y}_p(n) = [\widehat{x}(n), \ldots, \widehat{x}(n-d_y+1)] \tag{13}$$

where the output regressor for the SP mode in Equation (12) contains d_y past values of the actual time series, while the output regressor the P mode in Equation (13) contains d_y past values of the estimated time series. For a suitably trained network, these outputs are estimates of previous values of $x(n+1)$, and should obey the following predictive relationships implemented by the NARX network:

$$\widehat{x}(n+1) = \widehat{f}[\mathbf{y}_{sp}(n), \mathbf{u}(n)] \tag{14}$$

$$\widehat{x}(n+1) = \widehat{f}[\mathbf{y}_p(n), \mathbf{u}(n)] \tag{15}$$

where the nonlinear function $\widehat{f}(\cdot)$ be readily implemented through a MLP trained with backpropagation. The NARX networks trained according to Equations (14) and (15) are denoted onwards by NARX-SP and NARX-P networks, respectively.

Note that, unlike the FTDNN-based approach for the nonlinear time series prediction problem, the proposed approach makes full use of the output signal regressor $\mathbf{y}_{sp}(n)$ (or $\mathbf{y}_p(n)$). Equations (11) and (12) are valid only for one-step-ahead prediction tasks. If one is interested in multi-step-ahead or recursive prediction tasks, the estimates $\widehat{x}$ should also be inserted into the regressors in a recursive fashion.

The proposed approach is summarized as follows. A recurrent NARX network is defined so that its input regressor $\mathbf{u}(n)$ contains samples of the measured variable $x(n)$ separated $\tau > 0$ time steps from each other, while the output regressor $\mathbf{y}(n)$ contains actual or estimated values of the same variable, but sampled at consecutive time steps. As training proceeds, these estimates should become more and more similar to the actual values of the time series, indicating convergence of the training process. Thus, it is interesting to note that the input signal regressor supplies medium- to long-term information about the dynamical behavior of the time series, since the delay τ is always much larger than unity, while the output regressor, once the network has converged, supplies short-term information about the same time series.

IV. Simulations

Since in Internet and other packet/cell switching broad-band networks (such as ATM), Variable bit rate (VBR) video traffic will certainly be a major part of the traffic produced by multimedia sources, many researches have focused on VBR video traffic prediction to devising network management strategies that satisfy QoS requirements. Another motivation for studies on network traffic prediction comes from the important discovery of self-similarity and long-range dependence (LRD) in broad-band network traffic [25]. Researchers have also found that VBR video traffic typically exhibits burstiness over multiple time scales [26], [27].

In this paper, we evaluate the NARX-P and NARX-SP models using VBR video traffic time series (trace), extracted from Jurassic Park [28]. This video traffic trace was encoded at University of Würzburg with MPEG-I. The frame rates of video sequence coded Jurassic Park have been used. The MPEG algorithm uses three different types of frames: Intraframe (I), Predictive (P) and Bidirectionally-Predictive (B). These three types of frames are organized as a group (Group of Picture, GoP) defined by the distance L between I frames and the distance M between P frames. If the cyclic frame pattern is {IBBPBBPBBPBBI}, then L=12 and M=3. These values for L and M are used in this paper.

The resulting time series has 2000 sample points which have been rescaled to the range $[-1, 1]$. The rescaled time series was further split into two sets for cross-validation purposes: 1500 samples for training and 500 samples for testing.

For the sake of completeness, a performance comparison with the FTDNN and Elman recurrent networks is also carried out. All the networks evaluated in this paper have two-hidden layers and one output neuron. All neurons in both hidden layers and the output neuron use the hyperbolic tangent activation function. The standard backpropagation algorithm is used to train the networks with learning rate equal to 0.001. No momentum term is used. In what concerns the Elman network, only the neuronal outputs of the first hidden layer are fed back to the input layer.

The number of neurons, $N_{h,1}$ and $N_{h,2}$, in the first and second hidden layers, respectively, are chosen according to the following rules:

$$N_{h,1} = 2d_E + 1 \quad \text{and} \quad N_{h,2} = \sqrt{N_{h,1}} \tag{16}$$

where $N_{h,2}$ is rounded up towards the next integer number. The parameter d_y is chosen according to the following rule:

$$d_y = 2\tau d_E \tag{17}$$

where τ is selected as the value occurring at the first minimum of the mutual information function of the time series [29].

The networks are evaluated in terms of the *Normalized Mean Squared Error* (NMSE), defined as follows:

$$NMSE(N) = \frac{1}{N \cdot \sigma_x^2} \sum_{n=1}^{N} e^2(n) = \frac{\widehat{\sigma}_e^2}{\widehat{\sigma}_x^2} \tag{18}$$

where N is the horizon prediction (i.e., how many steps into the future a given network has to predict), $\widehat{\sigma}_x^2$ is the sample variance of the actual time series, and $\widehat{\sigma}_e^2$ is the sample variance of the sequence of estimation errors[1]. All the reported values of NMSE are mean values averaged over 10 training/testing runs.

The simulations aim to evaluate, in qualitative and quantitative terms, the predictive ability of all networks of interest. Once they have been trained, the networks are required to provide estimates of the future sample values of the laser time series for a certain prediction horizon N. The predictions are executed in a recursive fashion until desired prediction horizon is reached, i.e., during N time steps the predicted values are fed back in order to take part in the composition of the regressors. In this sense, the NMSE quantity in Equation (18) is better understood as a multi-step-ahead NMSE. For the NARX-SP network in particular, the predicted values, during multi-step ahead predictions, should be fed back to both the input regressor $\mathbf{u}(n)$ and output regressor $\mathbf{y}_{sp}(n)$.

Evaluation of the multi-step-ahead predictive performances of all networks can also help assessing the sensitivity of the neural models to important training parameters, such as the number of training epochs and the embedding dimension (d_E), as shown in Figure 2.

Figure 2(a) shows the NMSE curves for all neural networks versus the value of the embedding dimension, d_E, which varies from 3 to 24. For this simulation we trained all the networks for 300 epochs, $\tau = 1$ and $d_y = 24$. One can easily note that the NARX-P and NARX-SP performed better than the FTDNN and Elman networks. In particular, the performance of the NARX-SP was rather impressive, in the sense that it remains constant throughout the studied range. From $d_E \geq 12$ onwards, the performances of the NARX-P and NARX-SP are practically the same. It is worth noting that the performances of the FTDNN and Elman networks approaches those of the NARX-P and NARX-SP networks when d_E is of the same order of magnitude of d_y. This suggests that, for NARX-SP (or NARX-P) networks, we can select a small value for d_E and still have a very good performance.

Figure 2(b) shows the NMSE curves obtained from the simulated neural networks versus the number of training epochs, ranging from 90 to 600. For this simulation we trained all the networks with $\tau = 1$, $d_e = 12$ and $d_y = 2\tau d_E = 24$. Again, better performances were achieved by the the NARX-P and NARX-SP. The performance of the NARX-SP is practically the same from 100 epochs on. The same behavior is observed for the NARX-P network from 200 epochs on. This can be explained by recalling that the NARX-P uses estimated values to compose the output regressor $\mathbf{y}_p(n)$ and, because of that, it learns slower than the NARX-SP network.

[1] Assuming that the sequence of estimation errors has zero mean.

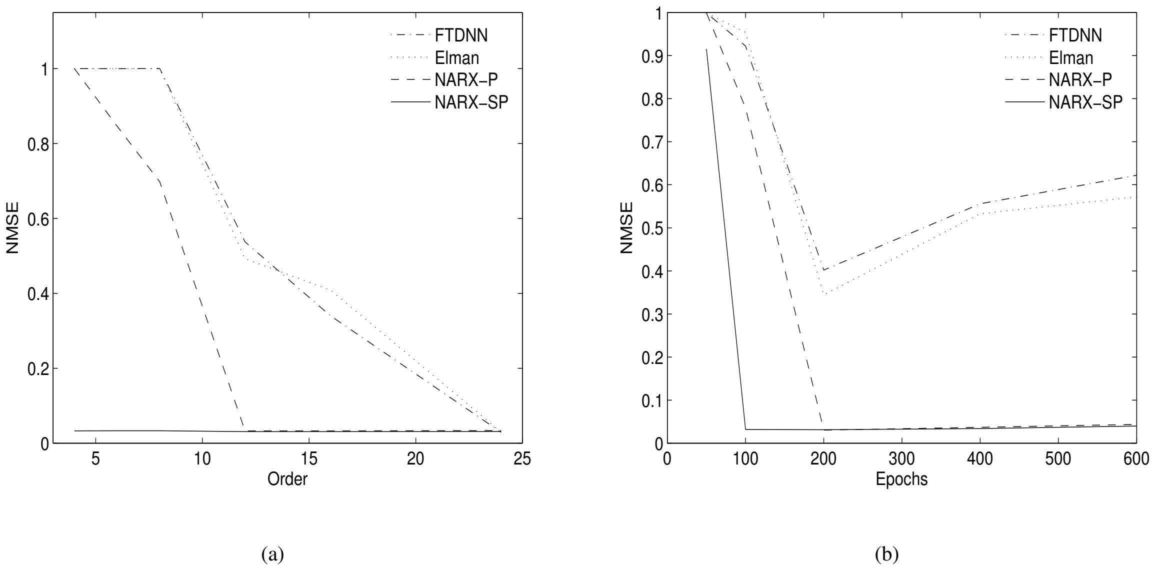


Fig. 2. Evaluation of the sensitivity of the neural networks with respect to (a) the embedding dimension and (b) the number of training epochs.

Another important behavior can be observed for the FTDNN and Elman networks. From 200 epochs onwards, these networks increase their NMSE values instead of decreasing them. This can be an evidence of overfitting, a phenomenon observed when powerful nonlinear models, with excessive degrees of freedom (too much weights), are trained for a long period with a finite size data set. In this sense, the results of Figure 2(b) strongly suggest that the NARX-SP and NARX-P networks are much more robust than the FTDNN and Elman networks.

Finally, we show in Figures 3(a), 3(b) and 3(c) typical estimated VBR video traffic traces generated by the FTDNN, Elman and NARX-SP networks, respectively. For this simulation, all the neural networks are required to predict recursively the sample values of the VBR video traffic trace for 300 steps ahead in time. For all networks, we have set $d_E = 12$, $\tau = 1$, $d_y = 24$ and trained the neural models for 300 epochs. For these training parameters, the NARX-SP predicted the video traffic trace much better than the FTDNN and Elman networks.

It is worth noting that the results reported in Figure 3 did not mean to say that the FTDNN and Elman networks cannot ever predict the video traffic trace as well as the NARX-SP. They only mean that, for the same training and configuration parameters, the NARX-SP has greater computational power provided by the output regressor. Recall that the MLP is an universal function approximation; and so, any MLP-based neural model, such as the FTDNN and Elman networks, are in principle able to approximate complex function with arbitrary accuracy, once enough training epochs and data are provided.

V. Conclusions

In this paper, we proposed a strategy that allows the original architecture of the NARX network to fully explore its computational power to improve performance in complex time series modelling and prediction tasks. We used real-world VBR video traffic time series to evaluate the proposed approach in multi-step-ahead prediction tasks. The results have shown that the proposed approach consistently outperforms standard neural network based predictors, such as the FTDNN and Elman architectures.

Acknowledgment

The authors would like to thank CNPq (grants #305275/2002-0 and #506979/2004-0) and FUNCAP for their financial support.

References

[1] S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, and M. Verleysen, "Time series forecasting with SOM and local non-linear models - Application to the DAX30 index prediction," in *Proceedings of the 4th Workshop on Self-Organizing Maps, (WSOM)'03*, 2003, pp. 340–345.

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

[3] D. Coyle, G. Prasad, and T. M. McGinnity, "A time-series prediction approach for feature extraction in a brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 4, pp. 461–467, 2005.

[4] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "An adaptable neural network model for recursive nonlinear traffic prediction and modelling of MPEG video sources," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 150–166, 2003.

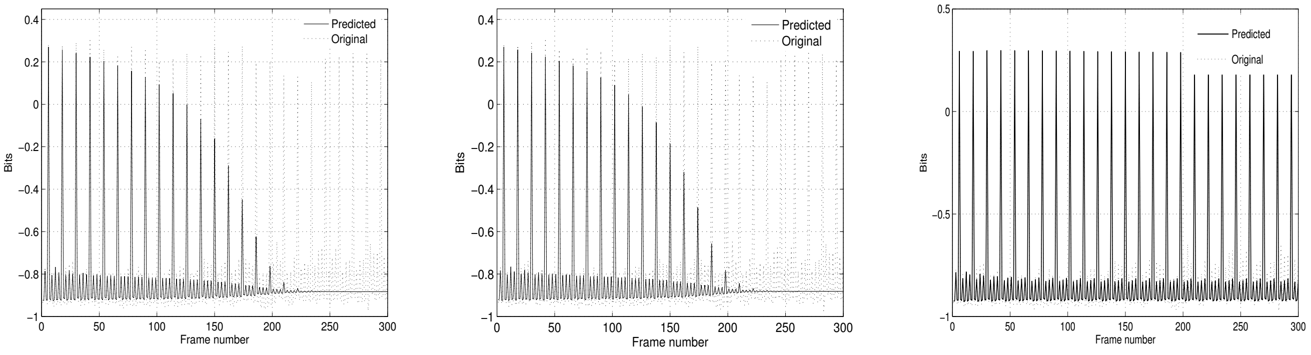
[5] Y. Liang, "Real-time VBR video traffic prediction for dynamic bandwidth allocation," *IEEE Transactions on Systems, Man and Cybernetics*, vol. C-34, no. 1, pp. 32–47, 2004.

[6] A. F. Atiya, M. A. Aly, and A. G. Parlos, "Sparse basis selection: New results and application to adaptive prediction of video source traffic," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1136–1146, 2005.

[7] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, Prentice Hall, 3rd edition, 1994.

[8] S. Haykin and J. C. Principe, "Making sense of a complex world," *IEEE Signal Processing Magazine*, vol. 15, no. 3, pp. 66–81, 1998.

[9] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural Adaptive Systems: Fundamentals Through Simulations*, John Willey and Sons, 2000.



(a) (b) (c)

Fig. 3. Recursive predictions obtained by (a) FTDNN, (b) Elman, and (c) NARX-SP networks.

[10] J. F. Kolen and S. C. Kremer, *A Field Guide to Dynamical Recurrent Networks*, Wiley-IEEE Press, 2001.

[11] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[12] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[13] T. Lin, B. G. Horne, and C. L. Giles, "How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies," *Neural Networks*, vol. 11, no. 5, pp. 861–868, 1998.

[14] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1424–1438, 1996.

[15] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for nonlinear systems - Part I: deterministic nonlinear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328, 1985.

[16] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, Springer, 2000.

[17] S. Chen, S. A. Billings, and P. M. Grant, "Nonlinear system identification using neural networks," *International Journal of Control*, vol. 11, no. 6, pp. 1191–1214, 1990.

[18] H. T. Siegelmann, B. G. Horne, and C. L. Giles, "Computational capabilities of recurrent NARX neural networks," *IEEE Transactions On Systems, Man, and Cybernetics*, vol. B-27, no. 2, pp. 208–215, 1997.

[19] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "A delay damage model selection algorithm for NARX neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2719–2730, 1997.

[20] M. Grossglauser and J. C. Bolot, "On the relevance of long-range dependence in network traffic," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 329–640, 1998.

[21] T. Sauer, J.A. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, pp. 579–616, 1991.

[22] H. D. I. Abarbanel, R. Brown, John J. Sidorowich, and L. Tsimring, "The analysis of observed chaotic data in physical systems," *Reviews of Modern Physics*, vol. 65, no. 4, pp. 1331–1392, 1993.

[23] H. Kantz and T. Schreiber, *Nonlinear time series analysis*, Cambridge University Press, Cambridge, 1997.

[24] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*, D. A. Rand and L.-S. Young, Eds. 1981, vol. 898 of *Lecture Notes in Mathematics*, pp. 366–381, Springer.

[25] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[26] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1566–1579, 1995.

[27] D. Heyman and T. Lakshman, "What are the implications of long-range dependence for VBR video traffic engineering," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 301–317, 1996.

[28] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proceedings of the 20th Annual IEEE Conference on Local Computer Networks (LCN'95)*. 1995, p. 397, IEEE Computer Society.

[29] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, pp. 1134–40, 1986.