ICCC '95, Seoul, Korea

ON THE PREDICTION OF THE STOCHASTIC BEHAVIOUR OF TIME SERIES BY USE OF NEURAL NETWORKS - AN APPLICATION TO SOURCE MODELLING

MARKUS D. EBERSPAECHER

University of Stuttgart, Institute of Communications Switching and Data Technics,
Seidenstrasse 36, 70174 Stuttgart, Germany

Phone: +49 711 1212482, Fax: +49 711 1212477, E-mail: eberspaecher@ind.uni-stuttgart.de

Abstract

In time series theory, the prediction of future values is a widely discussed subject. There are manyfold methods to derive models from data. One of the main objectives is to obtain the model parameters. Some proposals use self adapting techniques like Neural Networks for the parameters. Most of these approaches predict one future value of a time series. Some simulation tasks require models for traffic sources that are closely related to time series prediction though there exist different requirements. One of them is that a simulated traffic source should show the same stochastic behavior as a reference source. In this paper a procedure is presented that automatically adapts to a given reference source in the sense described above.

1 Introduction

General

When dealing with time series or, more generally, with sequences of data, there are mainly two areas of interest: the understanding of the underlying stochastic process and the prediction of future data. For the prediction of future data, a model of the stochastic process has to be built, based on a former analysis.

The analysis of time series is an extensively developed area of mathematics. There are many approaches to model different kinds of sequences of data. A simple method consists in adapting an ARMA model (Auto Regressive Moving Average) to the sequence. This works fine for most sequences without seasonal effects and trends. For other sequences, more sophisticated methods are required ([5], [6], [9]). However, many problems need their own specific solutions ([1], [2], [17]). Most of these approaches share the following characteristics: firstly, as much information as possible about the characteristics of the underlying data is collected and, secondly, a model that covers the essential features is deduced. Most procedures that deal with forecasting future values predict one (the next) value based on a set of past values. In this way these procedures lead to a deterministic behavior of the model.

All methods mentioned above share one disadvantage: they are inflexible in terms of changing parameters of the underlying data. These parameter changes have to be taken into account by the model and increase its complexity very much.

Time series models that are able to deal with changing parameters should be based on an architecture that is inherently able to automatically adapt to these changes. This architecture could be based on a Neural Network. Until today there are not many approaches that use a Neural Network (NN) architecture. Their advantage primarily consists in their ability to learn a given behavior without the exact knowledge of the underlying data and without difficult analysis needed for modelling. One disadvantage is that no detailed and understandable model of the underlying time series is built. These models are most often used for prediction ([12], [14], [15], [16]).

Source modelling is an area where the generation of future values is frequently used.

Source modelling

In source modelling the generation of deterministic new values from given data such as forecasting exchange rates is often of no particular interest. In contrast to that, a random data sequence is generated. Every new value is randomly chosen from a given distribution depending on the state of the model. The state is sometimes defined by some recently generated values (like autoregressive models). Source modelling is frequently used for traffic generation in the simulation of communication networks, in the simulation of manufacturing plants or in measurement technology. Since this paper concentrates on communications all the examples will relate to this area.

Multiple traffic sources built from the same model must be statistically independent from each other. A simple reproduction of measured data from a file (play back) is not sufficient. A shifted play back from a file where one traffic sequence starts at one point and another traffic sequence starts at another

point of the file is not sufficient, too, because of the strong correlations (especially when the file is short or when many sources are needed). Even in the case of very long files problems might occur in large simulations. This leads to the conclusion that reproduction from files is inflexible.

Another demand is that the number of events generated by a traffic source should be randomly choosable.

Like in conventional time series analysis ARMA models can be used for source modelling. There are some approaches employing Neural Networks but most of them don't fulfill all the stated requirements for source modelling. In [14], [15] NNs are used for modelling an ATM cell stream (Asynchronous Transfer Mode) without adding a random component. This kind of modelling is sufficiently exact, but not very useful for simulation.

How can traffic models be obtained? This is a simple task when only distributions are of interest and no correlations. Distributions might stem from measurement. The task can become very difficult when correlations must be taken into account. In this case an ARMA process might be profitably used, but how to obtain the model parameters? Another problem is that ARMA models are normally driven by white noise with the consequence that negative values are possible in any case, even if they are not allowed. [4] shows an example for source modelling using an ARMA model. Here the parameters are partly estimated from the measured data and partly calculated via the frequency domain.

Now suppose that the process to obtain a source model from a given sequence of data is transferred to a computer with an automatic adaptation of the source model to the data. This would save time and manpower needed to analyze the underlying data and to build a model.

In the following parts of this paper a new method is presented that adapts a source model to any random process or time series. It uses Neural Networks to learn the stochastic behavior of the underlying data.

Automatic source identification

The advantage of automatic source identification is a gain in productivity and saving of money since computational power is much cheaper than man power today. The disadvantage is the reduced possibility of interpretation of the generated source model. Only very few topological parameters have to be estimated from the data. All random processes that are weakly stationary and that have some seasonal effects can be modelled. Trends can be modelled only if they are very slow.

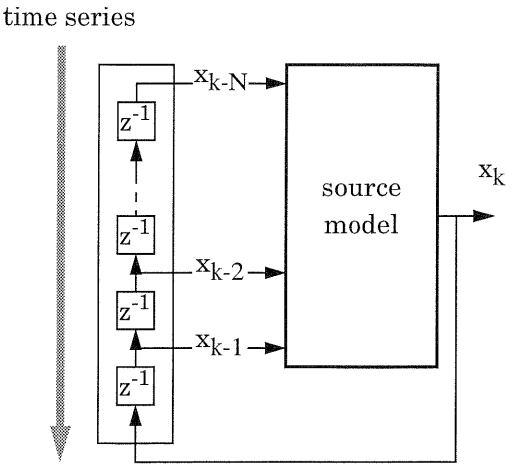

Figure 1: Principle of source model

Figure 1 shows one principle of this approach. Suppose that the content of the box named "source model" is already adapted to the data. The scalar value x_k is the output value of the modelled traffic source. N time delayed output values form the input vector $\mathbf{i}_k$

$$\mathbf{i}_k = (x_{k-1}, x_{k-2}, \ldots, x_{k-N}) \; . \qquad (1)$$

The shift register forms a memory for the past. The vector $\mathbf{i}_k$ is fed into the "source model," the output of which is the newly generated value. This output is fed back to the shift register, delayed by one step. So the number generating loop is closed.

Figure 1 provides a more detailed view. The source model is now divided into two parts, the distribution prediction (DP) and the random number generation (RNG). The RNG simply draws a new number according to the distribution at its input. The general inverse-transform method is used as RNG algorithm, see [10]. For every input vector the DP block computes the distribution the next output value is drawn from.

In Section 2 the components of the distribution prediction including the learning process are described.

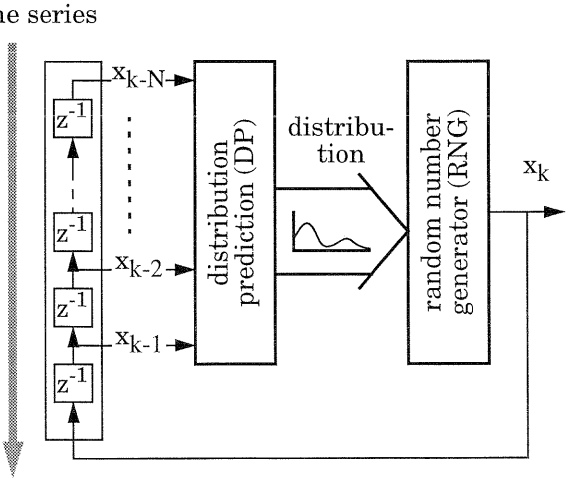

Figure 2: Extended principle of source model

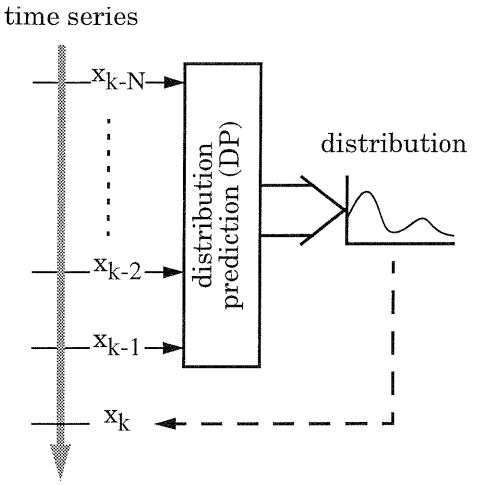

Figure 3: Principle of distribution prediction

Section 3 adds some examples of time series that are learned and predicted by applying the new source model.

2 Distribution Prediction

At first the scenario shown in Figure 3 is examined. N values of a time series preceding the current value x_k are fed to a black box named "distribution prediction." Inside this box the prediction for the distribution of the actual value is computed. For each vector $\mathbf{i}_k$ at the input there is a distinct distribution at the output.

The actual value x_k and the input vector $\mathbf{i}_k$ are now treated as one (N+1)-dimensional vector $\mathbf{x}_k = (x_k, \mathbf{i}_k)$ with a (N+1)-dimensional distribution function

$$F(x_k, x_{k-1}, \ldots, x_{k-N}) = P\{X_k \le x_k, \ldots, X_{k-N} \le x_{k-N}\} \tag{2}$$

and density function

$$f(x_k, x_{k-1}, \ldots, x_{k-N}) = \frac{\partial^N}{\partial x_k \ldots \partial x_{k-N}} F(x_k, \ldots, x_{k-N}) \tag{3}$$

Without loss of generality k is now set to 0 for simplicity. Then the density function becomes $f(x_0, x_{-1}, \ldots, x_{-N})$.

For distribution prediction the conditional distribution of x_0 for distinct values of the input vector $\mathbf{i}_k$ is needed. To obtain this the N-dimensional space that belongs to the input vector is quantized into M discrete vectors

$$\hat{\mathbf{p}}_i = (\hat{x}^i_{-1}, \ldots, \hat{x}^i_{-N}), \qquad i = 1, \ldots, M \tag{4}$$

This task is carried out by a vector quantizer (VQ).

Each vector $\hat{\mathbf{p}}_i$ points to the center of a region I_i of $\Re^N$. In these regions the density function of x_0, (3), is approximated by the function $\hat{f}_i(x_0)$. The union of all regions I_i forms the N-dimensional space $\Re^N$.

The approximation in region I_i is defined by the mean value of density (3) in this region:

$$\hat{f}_i(x_0) = \frac{1}{Vol(I_i)} \cdot \int_{I_i} \ldots \int f(x_0, t_1, \ldots, t_N) dt_1 \ldots dt_N \tag{5}$$

where

$$Vol(I_i) = \int_{I_i} \ldots \int dt_1 \ldots dt_N \tag{6}$$

This leads to the following error inside region I_i (the error measure is the squared difference between $f(\ldots)$ and $\hat{f}_i(\ldots)$):

$$E_i = \int_{-\infty}^{\infty} \left[\int_{I_i} \ldots \int f^2(x_0, t_1, \ldots, t_N) dt_1 \ldots dt_N \right] dx_0 - Vol(I_i) \cdot \int_{-\infty}^{\infty} \hat{f}_i^{\,2}(x_0) dx_0 \tag{7}$$

To compute the total error that results from quantization some features of the vector quantizer need to be known. This VQ adapts to the x_k in a way that all regions I_i are equally probable for the given time series. The algorithm used is partly taken from literature and is described in detail in [3]. The resulting total error is:

$$E_G = \sum_{i=1}^{M} E_i \tag{8}$$

E_G decreases for an increasing number M of regions.

The approximated density $\hat{f}_i(x_0)$ is obtained from the given time series, too. A new Neural Network

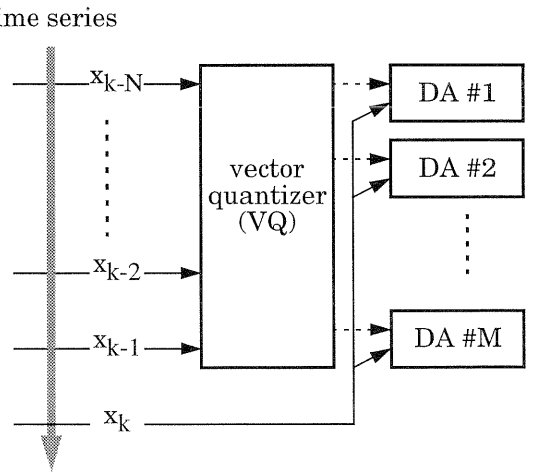

Figure 4: Learning of DP

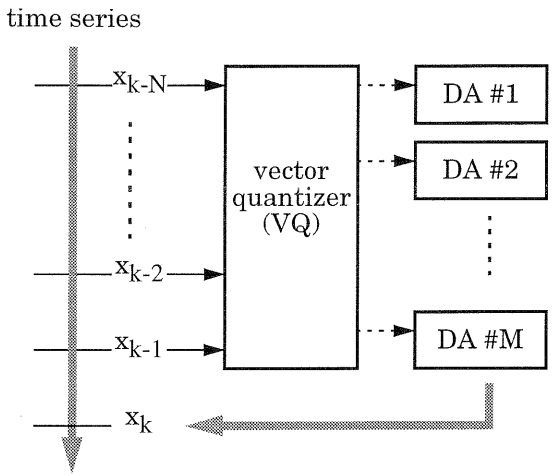

Figure 5: Prediction with VQ and DA's

algorithm was investigated that is able to form a distribution when a sequence of data is offered to it. This algorithm is presented in the next subsection.

Figures 4 and 5 show the connection between vector quantizer and distribution approximation (DA). Whenever an input vector falls into a region of the VQ, the corresponding DA is chosen for learning or prediction. In other words, the VQ is responsible to detect all correlations, the DA's are responsible for representing distributions.

Figure 4 shows the learning case. The actual value x_k is needed here as input for the actual DA to adapt the distribution. In the case of prediction (Figure 5), when learning is completed, one DA is chosen by the VQ for prediction of x_k.

Distribution adaptation

Figure 6 shows how the distribution adaptation module works. A time series that obeys to a distinct distribution is given. Minimum and maximum values are not known a priori. The algorithm inside the black box shall form an approximation of the distribution of the given data. The values of the time series are fed to the input of the adaptation module. The density function at the output is represented by a piecewise constant function, see Figure 7.

Usually, a distribution is measured by dividing the whole interesting region into small regions of equal width. The local density inside these regions is calculated from the frequency. The approach presented here has two advantages compared to normal distribution measurement: firstly, there is no a priori knowledge needed concerning the minimum and maximum values of the time series, the algorithm adapts automatically to them. Secondly, the regions are not of equal width and are adapted in order to obtain an optimal split. To achieve a high approximation quality the density function is approximated finer where it is high and coarser where it is low. This is achieved by equal probable regions. The region probability is approximated by the frequency f_i of each region. Figure 7 shows a sample approximation function with L regions. The values w_i denote the width of the regions and d_i the densities, respectively. The correspondence between width, density and region probability is $f_i = d_i \cdot w_i$ for i in 1..L.

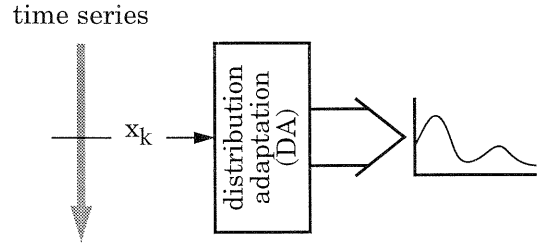

Figure 6: Principle of distribution adaptation

The algorithm is implemented as a non supervised Neural Network with L computing elements. There are no other inputs than the events from the time series for learning. The structure of the NN and the learning rule are described in more detail in [3]. In the recall phase, which is needed for distribution prediction, the output of the trained NN is used as the approximation of a density function.

3 Source Modelling - Examples

In this section some examples of modelling traffic sources are presented. The comparison between the original time series (reference) and the modelled time series is done by calculating the correlogram for both of them.

Markov modulated poisson process - MMPP

The first example is a MMPP process (Markov Modulated Poisson Process) with two states. The MMPP is a frequently used traffic model in telecommunications and represents a source with two activity states. Figure 8 shows a state-transition diagram. The process switches between the two states with probabilities $1-p_1$ and $1-p_2$, respectively. The expectation of event values in the states are m_1 and m_2. The parameters for the example are $p_1 = p_2 = 0.8$, $m_1 = 0.1$ and $m_2 = 15$.

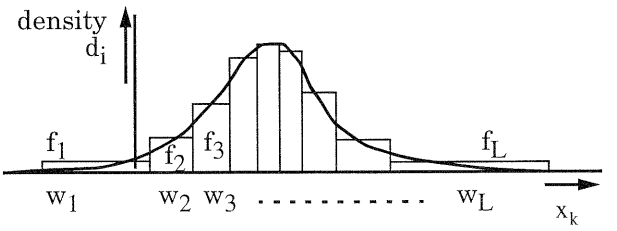

Figure 7: Example of approximated distribution

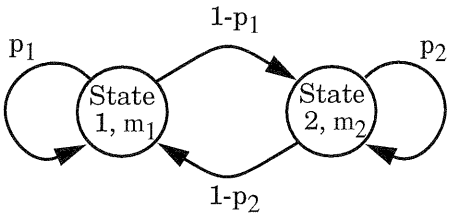

Figure 8: MMPP with 2 states

For distribution prediction N=5 VQ inputs and M=2 VQ units were used, thus having 2 distinct distributions approximated. L=10 segments were used for the approximation of each distribution. Figure 9 shows the resulting distribution functions of the approximation. Note that the mean values of the two curves are equal to the expectation mentioned above.

The correlograms in Figure 10 differ to a slight extent, because only two distribution approximation units have been used in this example.

Second order moving average process - MA(2)

The reference data for this test was produced from $y_k = \varepsilon_k + \Theta \cdot \varepsilon_{k-2}$ with $\Theta = 0.2$ and ε being white noise with mean 0 and variance 1.

The parameters of the source model are: N=3, M=25, L=10.

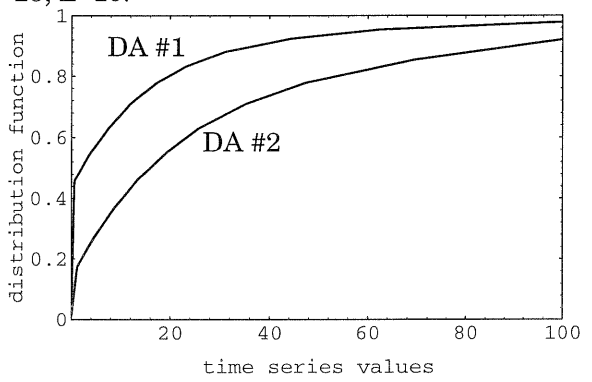

Figure 9: Distribution functions

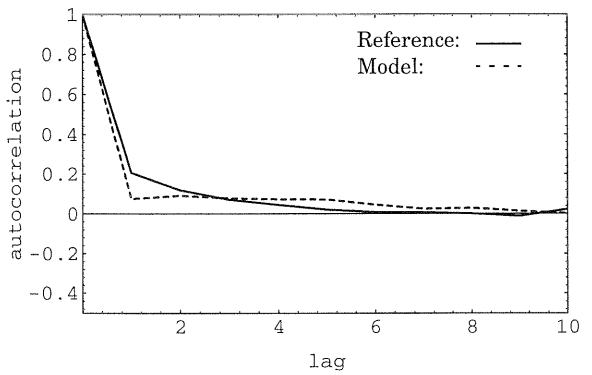

Figure 10: Correlogram

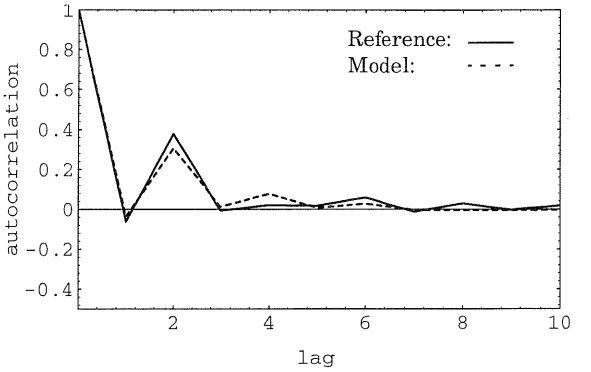

Figure 11: Correlogram

In Figure 11 the good correspondence of the autocorrelations can be seen.

MPEG coded video frames

This example is a real world case. A MPEG coded video sequence from the movie "Star Wars" is used. The sequence consists of the amount of data per video frame after compression.

Figure 12 shows the sequence generated by the MPEG scheme in principle. The large frames are so-called I-frames and comprise a whole picture. The medium-sized frames are P-frames, the small frames B-frames. The sequence I-B-B-P-B-B-P-B-B- is repeated cyclically and is defined by the MPEG parameters (more detailed information can be found in [8]). The size of the I-, B-, and P-frames is distributed according to the underlying video scenes. Due to this cyclical behavior the correlogram of MPEG-coded video data has a characteristic form as shown in Figure 13.

Figure 14 shows the resulting distribution approximations that are learned by the VQ and the DA's during the adaptation process (parameters N=9, M=15, L=10). It can be seen that there are different groups of distributions. Each group represents a different frame type (I, B or P) and the occurrence of the different distribution types obeys the same rule than the I-B-P-sequence above.

In Figure 13 it can be seen that the model fits the reference quite well. Figure 14 shows the set of 15 distribution functions belonging to this experiment.

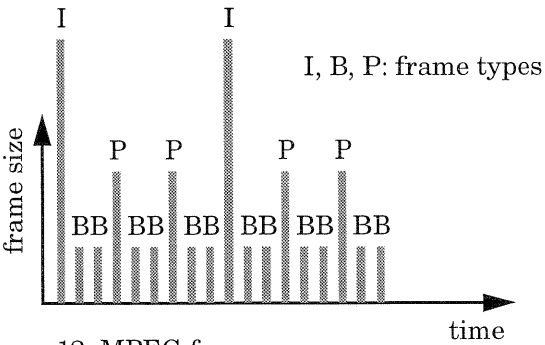

Figure 12: MPEG frames

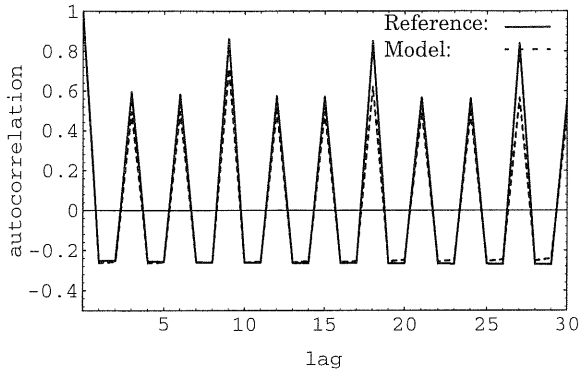

Figure 13: Correlation between frames

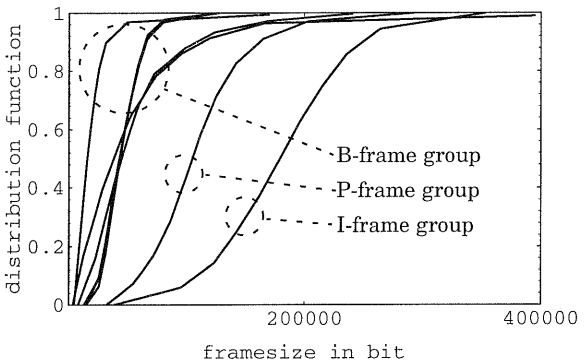

Figure 14: Distribution functions

In this figure the distribution groups belonging to different frame types are marked.

4 Summary

In this paper a new algorithm is presented that identifies arbitrary time series that may contain seasonalities and slow trends. It consists of a vector quantizer that reduces the complexity of the input data and a special Neural Network type that is able to learn distributions.

The model parameters for both the vector quantizer and the Neural Network are automatically derived by the learning process if an adequate topology is chosen.

Some examples demonstrate the usability of the method.

A more detailed description including a performance analysis of the presented approach and the algorithms can be found in [3].

5 References

[1] D. BRILLINGER, *New Directions in Time Series Analysis*, Part I, Springer, 1992.

[2] D. BRILLINGER, *New Directions in Time Series Analysis*, Part II, Springer, 1992.

[3] M. D. EBERSPAECHER, "On the Prediction of the Stochastic Behavior of Time Series by Use of Neural Networks - Performance Analysis and Results," accepted for presentation at PCN'95, Turkey.

[4] R. GRUENENFELDER, J. P. COSMAS, AND A. ODINMA-OKAFOR, "Characterization of Video Codecs as Autoregressive Moving Average Processes and Related Queueing System Performance," *IEEE Journal on Select. Areas in Commun.*, vol. 9, no. 3, April 1991, pp. 284-293.

[5] G. JANACEK, *Time Series*, Ellis Horwood, 1993.

[6] A. C. HARVEY, *Time Series Models*, Harvester Wheatsheaf, 1993.

[7] R. HECHT-NIELSEN, *Neurocomputing*, Addison-Wesley, 1989.

[8] D. LE GALL, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, vol. 34, no. 4, April 1991, pp. 46-58.

[9] J. D. HAMILTON, *Time Series Analysis*, Princeton University Press, 1994.

[10] AVERILL M. LAW, W. DAVID KELTON, *Simulation Modelling & Analysis,* McGraw-Hill, 1991.

[11] B. MAGLARIS, D. ANASTASSIOU, P. SEN, G. KARLSSON, AND J. D. ROBBINS, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Trans. on Comm.*, vol. 36, no. 7, July 1988, pp. 834-844.

[12] M. C. MOZER, "Neural Net Architectures for Temporal Sequence Processing," In: *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend, ed., Addison-Wesley, 1993, pp. 243-264.

[13] G. D. STAMOULIS, M. E. ANAGNOSTOU, AND A. D. GEORGANTAS, "Traffic Source Models for ATM Networks: A Survey," *Computer Communications*, vol. 17, no. 6, June 1994, pp. 428-438.

[14] A. A. TARRAF, I. W. HABIB, AND T. N. SAADAWI, "A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks," *IEEE Journal on Select. Areas in Commun.*, vol. 12, no. 6, August 1994, pp. 1088-1096.

[15] A. A. TARRAF, I. W. HABIB, AND T. N. SAADAWI, "Neural Networks for ATM Multimedia Traffic Prediction," Proceedings of IWANNT '93, pp. 85-91.

[16] E. A. WAN, "Time Series Prediction Using a Connectionist Network with internal Delay Lines," In: *Time Series Prediction: Forecasting the Future and Understanding the Past*, A. S. Weigend, ed., Addison-Wesley, 1993, pp. 195-217.

[17] A. S. WEIGEND, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.