Economics & Complexity

an

Interdisciplinary Journal

on Public, Financial, Globalisation and Social Economics

Contents

N. 1 Vol. 2 - Spec. NEU 99a

Published by
Economic & Complexity
www.ecople.org
April 2000

This number has been printed by financial support of University of Salerno

Re-Printed by Fondazione IDIS – CITTÀ DELLA SCIENZA-*ONLUS*
with DocuTech
for CUEN srl, Napoli
January 2004

A software toolbox for time series prediction and its application to daily rainfall forecasting in a geographic basin

Giovambattista Cicioni*, Francesco Masulli**, Daniela Baratta**

*Istituto di Ricerca sulle Acque, Consiglio Nazionale delle Ricerche,
Via Reno 1, 00198 Roma, Italy
fax: [+39] 06 8417861; e-mail: cicioni@irsa.rm.cnr.it

** INFM and DISI - Dipartimento di Informatica e Scienze dell'Informazione,
Università di Genova, Via Dodecaneso 35, 16146 Genova, Italy
fax: [+39] 010 3536699; e-mail: masulli@disi.unige.it

Abstract: *In this paper, we present the main components of a software toolbox supporting a constructive framework for the design of efficient time series forecasters proposed by our group that integrates many contributions from the fields of neural networks, dynamical system theory and signal analysis. In this approach, the original signal is decomposed in reconstructed waves, using the Singular Spectrum Analysis and then, for each reconstructed wave we design and train a Multi-Layer Perceptron predictor using suggestions from dynamical systems theory. In the operational phase the prediction of the original signal is obtained as the sum of the predictions of individual reconstructed waves. This forecasting approach has been successfully applied to daily rainfall forecasting in the Tiber river basin, and in this paper we choose this test-case for elucidate how to use the software toolbox.*

Keywords: *time series forecasting, singular spectrum analysis, delay-embedding theorem, multi-layer perceptrons*

1. Introduction

In the last few years, some relevant contributions from the fields of neural networks, dynamical system theory, signal analysis have been integrated by our group in a powerful constructive framework for the design of efficient time series predictors (Studer and Masulli 1997; Parenti et al. 1997; Masulli et al. 1999, 2000).

In particular, the *universal functions approximation property* of *neural networks* and *fuzzy systems* (Cybenko 1989; Wang and Mendel 1992) is the theoretical support to the usage of such kind of learning machines as time series predictors. However, from theory no suggestions can be obtained in order to size the structure of those systems. The neural network theory gives only general suggestions in order to choose their parameters, such as, in the case of multi-layer perceptrons (MLPs), the number of units in the input layer, the sampling time of the series, and the number and dimensions of hidden layers.

The constructive approach proposed by our group to shaping a supervised neural model of a non-linear process is based on theoretical results and heuristics related to the *delay-embedding theorem* (Takens 1981; Mãné 1981) and on an unsupervised preprocessing based on the *Singular-Spectrum Analysis* (SSA) (Kumaresan and Tuffs 1980; Pike et al. 1984; Vautard et al. 1992). This approach has been successfully applied in the design of multi-layer perceptrons and Neuro-Fuzzy systems for simulated non-linear and chaotic signal prediction (Studer and Masulli 1997; Parenti et al. 1997), system identification (Masulli et al. 1999), and daily rainfall forecasting (Masulli et al. 2000, Cicioni and Masulli, 2000). The application to daily rainfall forecasting, where our approach obtains a mean square error (MSE) of 1-2 mm of rain, is particularly interesting, as the signal of interest is discontinuous and intermittent, and classical approaches obtained no useful results before.

In this paper we present the main components of a software toolbox we have developed in the MATLAB © environment to support the constructive approach to complex signal prediction, and we show how we applied this software to daily rainfall forecasting in the Tiber river basin.

In the next section we draw a synthetic description of our approach to complex signal prediction. In Sec. 3 we present the software toolbox. An Application to the Tiber river basin rainfall data set is presented in Sect. 4. In Sec. 5 we give the conclusions of this work.

2. Constructive approach to complex signal prediction

In (Studer and Masulli 1997; Parenti et al. 1997; Masulli et al. 1999, 2000), we have proposed a constructive approach to shaping a supervised neural model of a non-linear process based on two main steps:

Unsupervised decomposition: The original signal is decomposed in *reconstructed waves* (RWs), using the Singular Spectrum Analysis (SSA) (Kumaresan and Tuffs 1980; Pike et al. 1984; Vautard et al. 1992);

Supervised learning: For each RW we design and train a MLP predictor using suggestions from dynamical systems theory (Takens 1981; Mãné 1981; Abarbanel 1994).

In the operational phase the prediction of the original signal is obtained as the sum of the predictions of individual RWs.

Note that the unsupervised decomposition step is need only for complex signals. In facts, the application to prediction of neural networks or fuzzy systems is supported on their universal function approximation properties (Cybenko 1989, Wang and Mendel 1992) that hold only for continuous maps. Then, in the case of discontinuous or intermittent signals is useful to apply to them the unsupervised decomposition step (Masulli et al. 2000), based on the Singular Spectrum Analysis method. The cornerstone of SSA is the Karhunen-Loève expansion or Principal Component Analysis (Therrien 1989) that is based on the eigenvalues problem of the lagged covariance matrix. Concerning the application of SSA to prediction, it is supported by the following argument. Since the *principal component* (PC) is filtered version of the signal and typically band-limited, their behavior is more regular than that of the raw series, and hence more predictable. Vautard and Ghil (1989), Vautard et al. (1992) fit an auto-regressive (AR) model for each individual PC using the AR coefficient estimate of Burg (1978), while Lisi et al. (1995) use multi-layer perceptrons in order to predict the PCs.

In our approach (Masulli et al. 2000), in order to reduce the computational costs, we decompose the raw series in *reconstructed waves* (RWs) corresponding to SSA subspaces equivalent to similar explained variance and we predict the RWs using multi-layer perceptrons. Often, the *reconstructed waves* (RWs) are enough regular and then can be predicted separately. The prediction of the original series can be then recovered as the sum of those of all the individual series components.

The supervised learning step is supported on results and prescriptions related to the delay-embedding theorem (Takens 1981; Mãné 1981) that allow us to size the input layer of the MLP predictor of a *reconstructed wave* as the embedding dimension of the dynamical system that we can estimate using, e.g., the *Global False Nearest Neighbors* (FNN) method (Abarbanel 1996), while the time lag of the input can be selected as the first minimum of the average mutual information of the signal (Fraser and Swinney 1986; Vastano and Rahman 1989; Abarbanel 1996).

It is worth noting that the estimation for the time lag based on mutual information is not supported from theory, and must be validated experimentally (Studer and Masulli 1997). On the other hand, as shown by the Statistical Learning Theory (Vapnik 1996), for limited data sets, under the same performances on the training set, the best generalization can be obtained with learning machines of limited complexity. This theoretical result leads, sometimes, to select MLP with input layers smaller than the embedding dimension of the dynamical system. Anyway, the FNN technique gives a reasonable starting point for the search of the optimal structure of the predictor.

3. The software toolbox

We have developed a MATLAB toolbox supporting the constructive approach to complex signal prediction described in the previous Section. The most computational intensive modules have been written in C language and integrated in MATLAB as *.mex* files. In this section we present the principal modules included in the software toolbox.

3.1 SSA module

This module decomposes a discontinuous or intermittent raw series in a set of regular series using the Singular-Spectrum Analysis (SSA) (Kumaresan and Tuffs 1980; Pike et al. 1984; Vautard et al. 1992). In SSA, the state vectors $S_i = (s_i, s_{i+1}, ..., s_{i+M-1})$ are M-dimensional windows (augmented vectors) of the time series s. The original series can be expanded with respect to the orthonormal basis corresponding to the eigenvectors of the lagged covariance matrix Z_s, i.e., as the Karhunen-Loève expansion or Principal Component Analysis (Therrien 1989):

$$s_{i+j} = \sum_{k=1}^{M} p_i^k u_j^k \quad , \quad 1 \le j \le M \quad , \quad 0 \le i \le N-M \qquad (1)$$

where p_i^k are called *principal components* (PCs) and the eigenvalues u_j^k are called the *empirical orthogonal functions* (EOFs), and the orthonormality property

$$\sum_{k=1}^{M} u_j^k u_l^k = \delta_{jl} \quad , \quad 1 \le j \le M \quad , \quad 1 \le l \le M \qquad (2)$$

holds. It is worth noting that SSA does not resolve periods longer than the window length M. Hence, if we want to reconstruct a strange attractor whose spectrum includes periods of arbitrary length, the large M the better, avoiding to exceeding $M = N/3$ as, otherwise, statistical errors could dominate the last values of the auto-covariance function (Vautard et al. 1992).

Following Vautard and Ghil (Vautard et al. 1992), suppose we want to reconstruct the original signal s starting from a SSA subspace Θ of k eigenvectors. This problem can be formalized as the search for a series $\hat{s}$ of length N that is the

closest, in the least-squares sense, to the projection of the augmented series S onto EOFs with indices belonging to Θ. The solution of this problem is given by

$$\hat{s}_i = \begin{cases} \dfrac{1}{M} \displaystyle\sum_{j=1}^{M} \sum_{k \in \Theta} p_{i-j}^k u_j^k & for \quad M \le i \le N-M+1 \\ \dfrac{1}{i} \displaystyle\sum_{j=1}^{i} \sum_{k \in \Theta} p_{i-j}^k u_j^k & for \quad 1 \le i \le M-1 \\ \dfrac{1}{N-i+1} \displaystyle\sum_{j=i-N+M}^{M} \sum_{k \in \Theta} p_{i-j}^k u_j^k & for \quad N-M+2 \le i \le N \end{cases} \qquad (3)$$

When Θ consists on a single index k, the series $\hat{s}$ is called the k-th RC, and will be denoted by $\hat{s}^k$. RCs have additive properties, i.e.

$$\hat{s} = \sum_{k \in \Theta} \hat{s}^k \qquad (4)$$

In particular the series s can be expanded as the sum of its RCs:

$$s = \sum_{k \in \Theta} \hat{s}^k \qquad (5)$$

If we truncate this sum to an assigned number of RCs, the explained variance of the related augmented vector $\hat{S}$ is the sum of the eigenvalues associated to those RCs, while the estimation of the resulting reconstruction error is the sum of the eigenvalues corresponding to the remaining RCs. As a consequence, it is suitable to order the RCs following the value of the eigenvalues.

Let be L disjoint subspaces $\Theta_1, \Theta_2, ..., \Theta_L$, then a *reconstructed wave* (RW) is defined as

$$RW_l = \sum_{k \in \Theta_l} \hat{s}^k, \quad 1 \le l \le L \qquad (6)$$

Then, from Eq.s 5 and 6, we obtain:

$$s = \sum_{l=1}^{L} RW_l \qquad (7)$$

that says that the original series s can be recovered as the sum of all the individual RWs.

3.2 MI module

This module estimates the normalized embedding delay τ that is a candidate for the interval between the components of the state vectors that will be input to the neural network model of the non-linear dynamical process.

The delay-embedding theorem (Takens 1981; Mãné 1981) do not give any support in order to choose the normalized embedding delay τ, and permits, in principle, to use of any τ so long as the available series is infinitely long. In practice, if the normalized embedding delay τ is too long, the samples $s_1, s_{i+\tau}, ..., s_{i+(d-1)\tau}$ are not correlated (this happens in particular for chaotic systems, for which even two initially close chaotic trajectories diverge exponentially in time) and then, in general, the dynamical system can not be reconstructed. If τ is too short, every sample is essentially a copy of the previous one, bringing very little information on the dynamical system. Then τ should be large enough for the samples to be essentially independent of each other so as to serve as coordinates of the reconstruction space, but not so independent as to have no correlation with each other. This requirement is best satisfied by using the particular τ corresponding to the first minimum of the *average mutual information* between s_n and $s_{n+\tau}$ (Fraser and Swinney 1986; Vastano and Rahman 1989; Abarbanel 1996) that is defined as

$$I(\tau) = \sum_i P(s_i, s_{i+\tau}) \log_2 \frac{P(s_i, s_{i+\tau})}{P(s_i)P(s_{i+\tau})} \qquad (8)$$

where $P(.)$ are probabilities distributions based on frequency observations. As this criteria is not theoretical derived, it can fail in many practical applications, see, e.g., (Studer and Masulli 1997), and the optimal value for τ should be evaluated experimentally.

3.3 FNN module

This module estimates the *embedding dimension* d_E of the attractor that generates the time series. This value will be used as the sufficient dimension of then input

layer of the multi-layer perceptron to be used as predictor, as it is high enough that the deterministic part of the dynamics of the system is unfold.

The *embedding dimension* d_E is defined as the lowest (integer) dimension which unfolds the attractor, i.e., as the minimal dimension for which foldings due to the projection of the attractor in a lower dimensional space are avoided. The delay-embedding theorem (Takens 1981; Mãné 1981) guarantees that if the counting box dimension of the attractor is d_0, then we can unfold the attractor in a space of dimension $d_E \geq 2d_0 + 1$. In principle, d_0 can be estimated directly from the time series itself, but this task is very sensitive to the noise and needs large set of data points (order of 10^{d_0} data points) (Abarbanel 1996). Moreover, it is worth of noting that $d_E \geq 2d_0 + 1$ is not a necessary condition for unfolding, but is sufficient.

In the FNN module we estimate d_E using the *method of false nearest neighbors* proposed by Abarbanel (1996) that we describe hereby.

Given a data space reconstruction in dimension d, with data vectors $S_i = (s_i, s_{i+\tau}, ..., s_{i+(d-1)\tau})$, where the time delay τ is the first minimum of average mutual information, and $S_i^{NN} = (s_i^{NN}, s_{i+\tau}^{NN}, ..., s_{i+(d-1)\tau}^{NN})$, the nearest neighbor vector in phase space. If the vector S_i^{NN} is a *false neighbor* (FNN) of S_i, having arrived in its neighborhood by projection from a higher dimension because the present dimension d does not unfold the attractor, then by going to the next dimension $d+1$ we may move this point out of the neighborhood of S_i. Let be $R_d^2(.)$ and $R_{d+1}^2(.)$ the square of the Euclidean distances between the nearest neighbor points as seen in dimensions d and $d+1$. We define the distance ξ between points when seen in dimension $d+1$ relative to the distance in dimension d, as

$$\xi_i = \sqrt{\frac{R_{d-1}^2(i) - R_d^2(i)}{R_d^2(i)}}, \qquad (9)$$

then

$$\xi_i = \frac{|s_{i+d\tau} - s_{i+d\tau}^{NN}|}{R_d(i)} \quad (10)$$

S_i^{NN} and S_i can be classified as a false neighbor if ξ_i is a number greater than a threshold θ ($\xi_i \geq \theta$) (Abarbanel 1996). In many applications a good value for θ is 15. In case of clean data from a dynamical system, we expect that the percentage of FNNs will drop from nearly 100% in dimension one close to zero when d_E is reached.

It is worth noting that, as we go to higher dimensional spaces the volume available for data grows as the distance to the power of dimension, and no near neighbor will be classified close neighbor (*curse of dimensionality problem*). In this case we can modify the Eq. 10 as

$$\xi_i = \frac{|s_{i+d\tau} - s_{i+d\tau}^{NN}|}{R_A} \quad (11)$$

where A is the nominal *"radius"* of the attractor defined as the root mean-square (RMS) error value of data about its mean, i.e.:

$$R_A = \frac{1}{N}\sum_{i=1}^{N}|s_i - s_{av}| \quad , \quad s_{av} = \frac{1}{N}\sum_{i=1}^{N} s_i \quad . \quad (12)$$

Using *k-d tree* search scheme (Friedman et al. 1977; Sproull 1991), establishing the neighbor relationships among N points takes order $N(\log N)$ operations.

In our implementation we have used the faster algorithm for nearest neighbor search proposed by Nene and Nayar (1997). This algorithm finds in the *point set* the point that is closest to a novel query point Q of coordinates $Q_1, Q_2, ..., Q_d$ and within a distance ε, taking order $N\varepsilon$ operation for small ε. The algorithm first finds all the points that lie inside a hypercube cube of side 2ε centered at Q (*candidate list*). Since ε is typically small, the number of points inside the cube is also small. The closest point can then be found by performing an exhaustive search on these points. If there are not points inside the cube, we know that there are not points within ε.

More in details, the candidate list can be found as follows: We first find points in the point set that lie between a pair of parallel hyperplanes separated by a distance 2ε, perpendicular to the first coordinate axis, and centered at Q_1; then we trim the candidate list using dimensions $2,3,...,d$. The algorithm uses only $1D$ binary searches (Aho et al. 1974) on an *ordered set* that is a simple pre-compiled data structure obtained from the point set made up by d $1D$ sorted arrays, where the j th array contains the j th coordinate of the points. The algorithm uses also two integer arrays: A *backward map* that maps a coordinate in the ordered set to the corresponding coordinate in the point set and the *forward map* that maps a point in the point set to a point in the ordered set.

In our implementation we use an adaptive ε, as we must find at least one nearest point to the novel point. We start with a small value of ε. In case of failure we increase ε, until, for $\varepsilon = \varepsilon^*$, we find at least one point in the candidate list. We assign then $\varepsilon = \sqrt{2}\varepsilon^*$, and we study the FNNs in the associate candidate list.

3.4 MLP module

This module implements a standard multi-layer perceptron with sigmoidal activation function (Haykin 1998) with back-propagation learning algorithm (Rumhelart et al. 1986) with momentum (Vogl et al. 1988) and early stopping criteria based on the best *generalization* obtained on a *validation set* disjoint on the *training* set used for training. Another disjoint set of data (*test set*) is used after training for evaluate the network's performances.

4. Putting it all together: An Application to the Tiber river basin rainfall data set

In this section we illustrate in a specific case the application of our software toolbox to the prediction of rainfall in a river basin. For details see Cicioni and Masulli (2000) and Masulli et al. (2000).

The data series is the average rainfall series (that we call *Mean Station* or MS) obtained from the rainfall intensities series collected by 135 stations distributed in the Tiber river basin for a period of 10 years (01/01/1958 - 12/31/1967). In figure 1 we illustrate a window on the period 07/01/66 - 12/30/66 that enlightens the discontinuity and intermittence of this signal.
From the MI module, we obtained the average mutual information of the MS series. Its first minimum gives an estimation of the time lag τ equal to 7. This time

lag was then used for the computation of Global False Nearest Neighbors, using the FNN module. The FNN ratio untill a dimension equal to 6 decreases with the growing of the dimension, and then reaches a plateau of 20%. The estimated embedding dimension is then $d_E = 6$.

The results obtained from raw MS data using the neural of the MLP module sized with the results of previous processing, were very poor, due to the discontinuity of the hydrological variable. Then, we applied the Singular-Spectrum Analysis to the first 3000 samples of the MS series using a window length M = 182 days (SSA module). In this way, from the original MS series we obtain 10 waves $(RW_1, ..., RW_{10})$ reconstructed from 10 disjoint sub-spaces, each of them representing 10% of the explained variance.

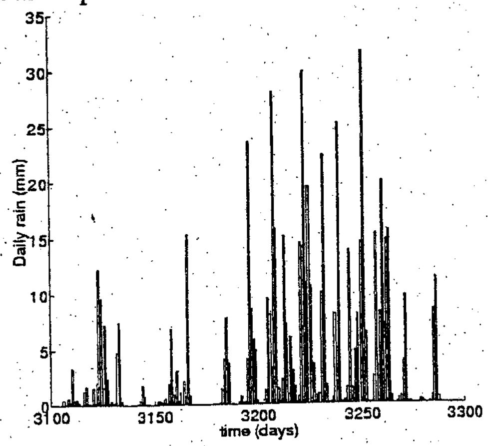

Fig.1 Mean Station: Daily rain millimeters. Period 07/01/66 - 12/30/66

The best results for each RW have been obtained using as inputs windows of 5 consecutive elements and two hidden layers. As each wave contains 3652 daily samples, in our case for each wave we obtained a data set of 3645 associative couples, each of them consisting of a window of 5 consecutive elements, as input, and the next day rainfall intensity, as output. Each MLP was trained using a training set corresponding to the first 2000 associative. The following 1000 associative couples constitute the validation sets, while the remaining 645 constitute the test sets.

The sum of the predictions of the 10 waves at 1 day ahead is very satisfactory, as for the resulting MS prediction the Mean Square Error (MSE) on the test set is .95 mm of rain, while the Maximum Absolute Error is 6.47 mm, i.e., the predicted signal is substantially coincident with the measured MS rainfall intensity signal (see fig. 2).

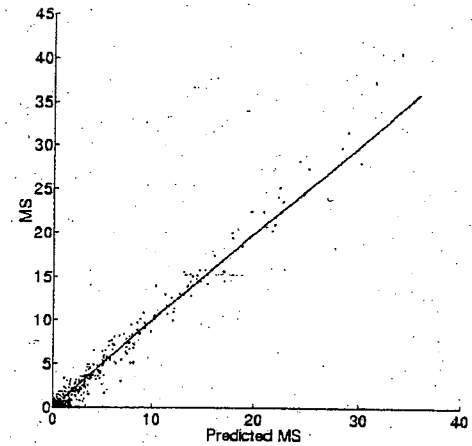

Fig.2 Mean Station: scatter plot for one-day ahead forecasting

Moreover, we notice that preliminary results for the forecasting of the rainfall series of individual stations are also in good agreement with data.

5. Conclusions

In this paper, we have presented the main components of a software toolbox integrated in the MATLAB environment supporting a constructive framework for the design of efficient time series forecasters proposed by our group that integrates many contributions from the fields of neural networks, dynamical system theory and signal analysis (Studer and Masulli 1997; Parenti et al. 1997; Masulli et al. 1999, 2000). The steps of the constructive approach to shaping a supervised neural model of a non-linear process based are:

Unsupervised decomposition: The original signal is decomposed in *reconstructed waves* (RWs), using the Singular Spectrum Analysis (SSA) (Kumaresan and Tuffs 1980; Pike et al. 1984; Vautard et al. 1992);

Supervised learning: For each RW we design and train a MLP predictor using suggestions from dynamical systems theory (Takens 1981; Mãné 1981; Abarbanel 1994).

In the operational phase the prediction of the original signal is obtained as the sum of the predictions of individual RWs.

Details on the implementation of the main software modules are presented. In particular the speed-up obtained by the application of the algorithm by Nene and Nayar (1997) in the implementation of the method of false nearest neighbors is noticeable.

This forecasting approach has been successfully applied to daily rainfall forecasting in the Tiber river basin (Cicioni and Masulli 2000, Masulli et al. 2000), and in this paper we chosen this test-case for elucidate how to use the software toolbox.

Acknowledgements

This works was partially supported by Progetto Coordinato CNR: "Neuro-fuzzy systems to forecast hydrological variables", IRSA-CNR, INFM, and University of Genoa. We thank Leonard Studer for useful discussions and Fabio Montarsolo for developing the C code.

References

Abarbanel, H.D.I. (1994), "Analysis of Observed Chaotic Data", New York, USA, Springer.

Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1974), "The Design and Analysis of Computer Algorithms", Addison-Wesley.

Cicioni, Gb. and Masulli, F. (2000), "Rainfall Forecasting in the Tiber Basin using a Predictor based on Singular Spectrum Analysis and Neural Networks"; Proceedings of Hydroinformatics'2000 (in press).

Cybenko, G. (1989), "Approximation by superpositions of a sigmoidal function", Mathematics of Control, Signals and Systems, 2, 303-314.

Burg, J.P. (1978), "Maximum entropy spectral analysis", In D.G. Childers, (ed.), Modern Spectrum Analysis, New York, IEEE Press, 34.

Fraser, A. and Swinney, L. (1986), "Independent coordinates for strange attractors from mutual information", Physical Review, 33:1134-1140.

Friedman, J.H., Bentley, J.L. and Finkel, R.A. (1977), "K-d tree", ACM Trans. Math. Software, 3:209-226.

Haykin, S. (1999), "Neural Networks, A Comprehensive Foundation" (Second Edition). Prentice Hall, Upper Saddle River, NJ.

Kumaresan, R. and Tuffs, D.W. (1980), "Data-adaptive principal component signal processing", In IEEE Proc. Conf. on Decision and Control, Albuquerque, USA, IEEE, 949.

Lisi, F., Nicolis, O. and Sandri, M. (1995), "Combining Singular-Spectrum Analysis and neural networks for time series forecasting", Neural Computation, 2:6-10.

Mãné, R. (1981), "On the dimension of the compact invariant sets of certain non-linear maps", In D.A. Rand and L.-S. Young, editors, Dynamical Systems and Turbulence, volume 898 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 230-242.

Masulli, F., Parenti, R. and Studer, L. (1999), "Neural modeling of non-linear processes: Relevance of the Takens-Mãné theorem", International Journal on Chaos Theory and Applications.

Masulli, F., Cicioni, Gb. and Studer, L. (2000), "Discontinuous and Intermittent Signal Forecasting: A Hybrid Approach", DISI-TR-00-4, Department of Computer and Information Sciences, University of Genoa, Genova (Italy).

Nene S.A. and Nayar, S.K. (1997), "A simple algorithm for nearest neighbor search in high dimensions", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:989-1003.

Parenti, R., Masulli, F. and Studer, L. (1997), Control of non-linear process by neural networks: Benefits using the Takens-Mãné theorem, In *Proceedings of the ICSC Symposium on Intelligent Industrial Automation*, IIA'97, Millet, Canada, ICSC, 44-50.

Pike, E.R, MCWhirter, J.C., Bertero, M. and deMol, C. (1984), Generalized information theory for inverse problems in signal processing, *IEE Proceedings*, 59:660–667.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), Learning internal representations by error propagation, In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, vol. 1, 318-362.

Sproull, R.F. (1991) Fast k-d tree, *Algorithmica*, 6:579-589.

Studer, L. and Masulli, F. (1997), Building a neuro-fuzzy system to efficiently forecast chaotic time series, Nuclear Instruments and Methods in *Physics Research, Section A*, 389:264-667.

Takens, F. (1981), Detecting strange attractors in turbulence, In D.A. Rand and L.-S. Young, editors, Dynamical Systems and Turbulence, volume 898 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 366-381.

Therrien, C.W. (1989), *Decision, Estimation, and Classification: An Introduction to Pattern Recognition and Related Topics*, Wiley, New York.

Vastano, J. and Rahman, L. (1989), Information transport in spatio-temporal chaos, *Physical Review Letters*, 72:241-275.

Vautard, R. and Ghil, M. (1989), Singular-spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series, *Physica D*, 35:395-424.

Vautard, R., You, P. and Ghil, M. (1992), Singular-Spectrum analysis: A toolkit for short, noisy chaotic signals, *Physica D*, 58:95–126.

Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.Y. and Alkon, D.L. (1988), Accelerating the convergence of the back-propagation method, *Biological Cybernetics*, 59:257-263.

Wang, L. and Mendel, J.M. (1992), Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, *IEEE Trans. on Neural Networks*, 5:807-814.

Vapnik, V.N. (1995), *The Nature of Statistical Learning Theory*, Springer, New York