Reprinted from: Engineering Diagnosis and Application
CD ROM Tutorial track 11 - H.Szu, F.C. Morabito, Eds..
IJCNN'99, Int. Joint Conf. on Neural Networks - Washington, D.C., USA, 1999

Time Series Forecasting and Neural Networks

Francesco Masulli
Istituto Nazionale per la Fisica della Materia &
DISI-Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova, via Dodecaneso 35, I-16146 Genova, Italy
E-mail: masulli@ge.INFM.it

Léonard Studer
Institut de Physique des Hautes Énergies,
Université de Lausanne, CH-1015 Dorigny, Switzerland
E-mail: leonard.studer@iphe.unil.ch

Abstract

In this tutorial, we present a constructive methodology for shaping neural networks models of non-linear dynamical systems on the basis of their output signals. The method is supported by results and prescriptions related to the Takens-Mañé theorem, including the evaluation of the time delay using the measurement of the first minimum of the mutual information of the signal, and in the estimation of the embedding dimension using the method of global false nearest neighbors. We present some numerical experiments to assess this constructive approach to the identification of the Mackey–Glass chaotic system and a non-linear dynamic system, and its application to the design of a neural network to forecasting a time series generated by an accelerometer coupled to a 150 MW steam turbine. We present also an extension of this approach to discontinuous or intermittent signals prediction. As the universal function approximation theorem for neural networks and fuzzy systems requires the continuity of the function to be approximate, we apply the Singular-Spectrum Analysis (SSA) to the original raw signal, in order to obtain a family of time series components that are more regular than the original signal and can be, in principle, predicted one at a time using the mentioned methodology. On the basis of the properties of SSA, the prediction of the original series can be recovered as the sum of those of all the individual series components. We show then an application of this prediction approach to a hydrology problem concerning the forecasting of daily rainfall intensity series, using a database collected for 10 years from 135 stations distributed in the Tiber river (Italy) basin.

Keywords: Neuro-Fuzzy Systems, Multi-Layer Perceptrons, Time Series Forecasting, Embedding Method, Singular Spectrum Analysis, Chaotic Signals Forecasting, Steam Turbines Identification, Rainfall Forecasting.

1 Introduction

In the last decade, neural networks have been widely tested on non-linear dynamic systems modeling and forecasting. Existence theorems, stating that Multi-Layer Perceptrons (MLPs) and Neuro-Fuzzy Systems are universal approximators of any arbitrary continuous function, have been demonstrated [4, 12, 15, 36]. However, from theory no information can be obtained in order to define the structure of the approximator based on neural network.

Applying Multi-Layer Perceptrons or Neuro-Fuzzy Systems to the problem of time series forecasting implies the setting of the number of units in the input layer, the structure and dimension of the hidden layers, the size of the training set. The neural network theory gives only general suggestions in order to choose these numbers. The specificities of data set have to be taken into account at this level to tailor the neural network to the time series which have to be forecasted.

Results achieved in the theory of chaotic systems point out very relevant elements which can be extracted from the measurement of one-variable time series of the non-linear dynamic system. One of these results is given by the Takens-Mañé theorem about the sufficient dimension of an Euclidean space to guarantee a fair representation of the true strange attractor of the underlying system.

In this tutorial, that is based on a series of recent results obtained our group [28, 22, 25, 21, 3], we present a constructive methodology for shaping a neural model of a non-linear process, supported by results and prescriptions related to the Takens-Mañé theorem: they are based on the measurement of the first minimum of the mutual information of the output signal and on the application of the method of global false nearest neighbors to estimating the embedding dimension.

Even if many other neural networks have been applied to time series forecasting, we shall consider only Neuro-Fuzzy Systems and Multi-Layer Perceptrons because, although their applications are simple, they are powerful enough as time series forecasters.

Some examples of application of the constructive methodology for time series forecasting will be presented:

- the modeling of a chaotic system;
- the identification of a non-linear dynamic system;
- the forecasting of the vibration dynamic of a steam turbine.

We will present also an extension of this approach to the prediction of discontinuous or intermittent signals. In this case the application of the Singular-Spectrum Analysis (SSA) [16, 26, 33] to the original raw signal permits to deal with a family of series components (*reconstructed waves*) that are more regular signals. Each reconstructed wave can be, in principle, predicted separately either by a neural network or by another forecasting method. Thanks to the properties of SSA, the prediction of the original series can be recovered as the sum of prediction of the reconstructed waves.

We show an application of all presented methods in the hydrology field, consisting in the neural forecasting of rainfall intensity series collected by 135 stations distributed in the Tiber basin for a period of 10 years.

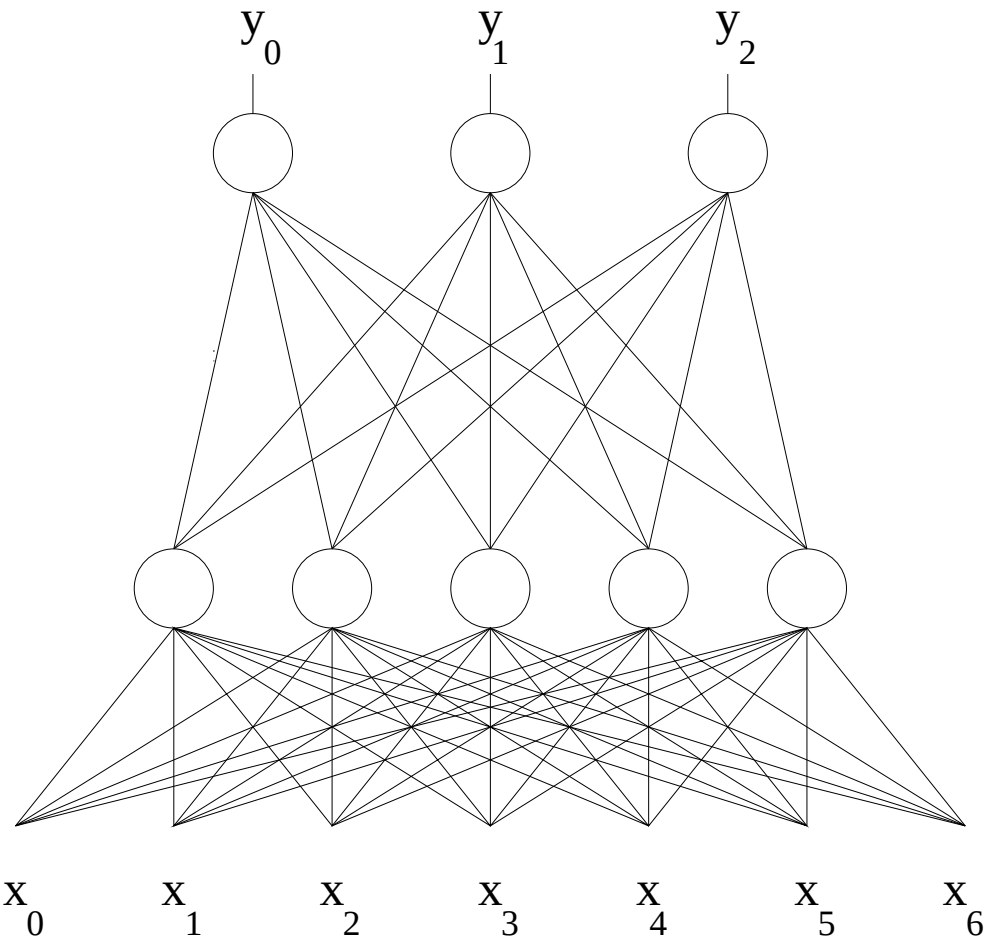

Figure 1: A MLP with 7 nodes in the input layer, 3 nodes in the output layer and one hidden layer with 5 nodes.

In the next Sect. we present the Multi-Layer Perceptron and the Neuro-Fuzzy System that we will apply to Time Series Forecasting. The constructive methodology for time series forecasting based on Dynamical Systems Theory is shown in in Sect. 3, while in Sect. 4 we illustrate its applications to some simple cases, namely a chaotic dynamics, a non-linear system and a steam turbine. In Sect. 5, the extension of the methodology to discontinuous and intermittent signals based on Singular-Spectrum Analysis is presented, and in the following Sect. we show an application to rainfall intensity series. In Sect. 7, we draw the conclusions.

2 Neural Networks for Time Series Forecasting

2.1 Multi-Layer Perceptrons

Artificial neural networks are made up of simple *nodes* or *neurons* interconnected to one another. Generally speaking, a node of a neural network can be regarded as a block that measures the

similarity between the input vector and the parameter vector, or *weight vector*, associated to the node, followed by another block that computes an activation function, normally not linear [20, 11]. The transfer function of an artificial neuron is given by the equation:

$$y = H(\sum_i w_i x_i - \theta) \tag{1}$$

where y is the output of the neuron, H is an activation function, w_i are weights, x_i are the inputs, and θ is the threshold.

The most used neural network is the Multi-Layer Perceptron (MLP) that is a feed-forward model based by layers of neurons (see Fig. 1). Nodes of each layer are interconnected with all nodes of the following layer. In this way Multi-Layer Perceptrons perform non-linear maps from an input space to an output space. Moreover, as demonstrated by the *Universal Approximation Theorem* [4, 12], an MLP with a single hidden layer [1], and using sigmoid activation functions $H(x) = 1/(1 + exp(-ax))$, with slope parameter a, is sufficient to uniformly approximate any continuous function with support in a unit hypercube.

The non-linear map can be automatically learned from data by a MLP thought supervised learning techniques based on the minimization of a cost function, such as the Mean Square Error. The most diffused learning technique is the *Error Back-Propagation* that is an efficient application of the Gradient Descent method [27, 11].

2.2 Neuro-Fuzzy Systems

Fuzzy set theory [37, 14] is an extension of the conventional (crisp) set theory. A *fuzzy set* A is defined via a *membership function* $\mu_A(x)$ which gives the membership grades of elements x to the fuzzy set A. By construction, fuzzy sets are very convenient to numerically capture linguistic concepts such as "large", "warm", or "cold". The form of the membership function is arbitrary and has to be determined in function of the problem. Methods have been devised for this determination, inspired from knowledge engineering or statistics.

Fuzzy logic relates these linguistic variables (i.e. fuzzy sets) through operations. Fuzzy logical operations between fuzzy sets are extensions of classical connectives such as intersection, union, set-complement, AND, OR, THEN etc.

Fuzzy logical operations are not uniquely definite. In fact, for one given fuzzy logical operation (let say intersection) there is an infinite family of usable operations. Fuzzy intersection can be represented by *min* or *product*, fuzzy union can be *max*, or *sum* operations (among others). Fuzzy complement is appropriately represented by complement to 1 of the membership function. As in classical logic, logical conjunctive operation (AND) is implemented by an intersection between sets; disjunction by union and negation by set-complement.

Fuzzy Inferential Systems (or Fuzzy Systems) are constituted by four components:

- The *fuzzification module* that transforms the *crisp* (i.e. not-fuzzy) input data coming from the real world into membership values.

[1] The output nodes constitute the output of the MLP. The remaining nodes constitute *hidden layers* of the network.

- The *fuzzy rule base* with a bank of *fuzzy if-then rules* or *fuzzy conditional statements*, of the type: **IF** A **AND** B **THEN** C, where A, B and C are fuzzy sets.
- The *decision making unit* or *fuzzy inference engine* performing the inferences on the rules following the selected approximate reasoning method.
- The *defuzzification module* that transforms the fuzzy sets resulting from fuzzy inference into crisp outputs.

One interesting fuzzy system is the neuro-fuzzy system (NFS)[36] that is a fuzzy system based on the following assumptions: height defuzzification, sum composition, product inference rules, Gaussian membership functions, and singleton fuzzifier. The NFS can be associated to a feedforward connectionist system with only one hidden layer. More specifically, if there are K units in the input layer, J fuzzy inference rules and I outputs, the rule activations can be written as:

$$r_j \;=\; \prod_k \mu_{jk}(x_k). \tag{2}$$

The quantity $\mu_{jk}(x_k)$ is the value of the membership function of the component x_k of the input vector for the j-th rule, and is defined as:

$$\mu_{jk}(x_k) = \exp\left(-\frac{(x_k - m_{jk})^2}{2\sigma_{jk}^2}\right), \tag{3}$$

where m_{jk} and σ_{jk}^2 are the means and the variances. The values of the output units are:

$$y_i = \frac{\sum_j r_j s_{ij}}{\sum_j r_j} \tag{4}$$

and s_{ij} is the fuzzy singleton of the j-th rule associated with the output y_i.

We notice that this implementation mapping a Fuzzy Inference System in a Radial Basis Functions Neural Network (RBFNN) [11] is a very interesting case of *Computing with Words* [38]. In fact, the NFS has been proven to be an *Universal Approximator*[36] on any real continuous function on a compact support to an arbitrary precision. Moreover, some other advantages of using a NFS are:

1. The possibility to easily input some *a priori* knowledge (from an expert) to bias the NFS towards the problem to be solved.
2. By training, NFS can learn internal relations in numerical data sets.
3. Extraction of learned rules by a trained NFS is possible in form of meaningful linguistic (fuzzy) relations.

In the experiments shown in Sect. 4.1, no *a priori* knowledge has been used to enforce (before the learning phase) some fuzzy if-then rules in the NFS. We obtained the NFS parameters (namely m_{jk}, σ_{jk} and s_{ij}) by performing a gradient descent across the training set with respect to the Mean Square Error (MSE). The formula for the gradient descent is shown, e.g., in [36].

2.3 Universal Approximation Property

The universal approximation property, that holds for MLPs, NFSs and other neural networks, implies that, if the non-linear dynamical process can be represented by a continuous function, an efficient non-linear model can be built from data using one of those neural networks. In this way the costly detailed design step of the first principles model usually implemented in the non-linear system identification is transformed in a more simpler structuring step of the neural network plus an optional pre-processing (eventually driven by any understanding of the physical model of the process) of the raw data coming from the field.

Even if, in principle, the function approximation property guarantees the feasibility of data-based models of non-linear dynamical systems, the neural network theory doesn't give any suggestion about many details. For example no general prescriptions are available concerning the dimension of the data window (i.e. input layer of the MLP), the sampling rate of the input data, the dimension of the hidden layer, and the dimension of the training set, and then most of time those fundamental design parameters have to be obtained by experiments and heuristics [11].

3 A Methodology based on Dynamical Systems Theory

3.1 Dynamical Systems and Chaos Theory

3.1.1 State Space

A deterministic dynamical system is described by a set of differential equations. Its evolution is represented by the trajectory in state space (of dimension n) of the vector $\mathbf{Q} = (x, \dot{x}, y, \dot{y}, z, \dot{z}, \ldots)^\top$ where $x, \dot{x}, y, \dot{y}, z, \dot{z}, \ldots$ are the variables of the system and their derivatives. The figure made in state space by $\mathbf{Q}$ is the attractor of the system.

For non-linear systems, the dynamical variables $(x, y, z, \ldots)$ are coupled. The evolution of one variable (let say x) is not independent of all the other ones $(y, z, \ldots)$. Except for few simple phenomena, the set of differential equations is unknown. Even, often the whole set of relevant effective dynamical variables is not always well defined. But, as the variables are interdependent, the observation of only one of those brings information, even if in an implicit way, on the other ones and consequently on the complete dynamical system. This is the reason why time series of non-linear dynamic systems are so useful.

3.1.2 Embedding Theorem

The question can be put now as: "How to reconstruct the complete dynamical system with only the one-variable time series $(s_1, s_2, s_3, \ldots)$?" Here the *Embedding Theorem*, proposed independently in 1981 by Takens and Mañé [29, 18], gives an answer.

In the Takens-Mañé theorem we consider an augmented vector $\mathbf{S}$ built with d elements of the time series. The dimension of the vector d has to be greater than two times the box-counting dimension D_0 of the attractor of the system:

$$d > 2D_0 \tag{5}$$

A vector **S** satisfying this bound will evolve in a reconstructed state space, and its evolution will be in a diffeomorphic relation with the original **Q** state space point (a diffeomorphism is a smooth one-to-one relation). In other words, for every practical purposes the evolution of **S** is a fair copy of the evolution of **Q**.

It is worth noting that there is a distinction between the order n of the differential equation which is the dimension of the state space where live the true state vector **Q** and the sufficient dimension of a reconstructed state space d where the reconstructed vector **S** lives.

3.1.3 The Method of Embedding

In order to reconstruct the reconstruct the dynamical system we can use the *time delay embedding method* [1]. This method consists in building d-dimensional state vectors $\mathbf{S}_i = (s_i, s_{i+T}, \ldots, s_{i+(d-1)T})$. In principle, it suffices that $d \geq n$. But, the *effective* dimension d is not directly related to the dynamical dimension n − as in the case of weak coupled variables.

3.2 Choosing the time delay

The time delay T (or *time lag*) used in the embedding has to be chosen carefully. If it is too long, the samples $s_i, s_{i+T}, \ldots, s_{i+(d-1)T}$ are not correlated[2] and then, in general, the dynamical system can not be reconstructed. If it is too short, every sample is essentially a copy of the previous one, bringing very little information on the dynamical system.

We use the Shannon's mutual information concept to quantify the amount of information shared by two samples in order to get an useful estimation of the time lag T. Let's defined the *average mutual information* between measurements a_i drawn from the set A and measurements b_i drawn from set B. The set of measurements A is made of the values of the observable s_i and the set B is made of the values s_{i+t} (t is a time interval). Average mutual information is then :

$$I(t) = \sum_{s_i \in A, s_{i+t} \in B} P(s_i, s_{i+t}) \times log_2 \frac{P(s_i, s_{i+t})}{P(s_i)P(s_{i+t})}, \tag{6}$$

where $P(\ldots)$ are probabilities distributions based on frequency observations.

It has been suggested [8, 7, 31, 1] to take the time where the first minimum of I(t) occurs, as the value to use at the time delay T in the phase space reconstruction. In this way the values of s_i and s_{i+T} are the most independent of each other in an information-theoretic sense.

Moreover the first minimum of average mutual information is a good 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.

3.3 Evaluating the Global Embedding Dimension

From the Embedding Theorem, the box counting dimension D_0 should be evaluated. In principle, it 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) [1].

[2] This happens in particular for chaotic systems, for which even two initially close chaotic trajectories will diverge exponentially in time.

In order to avoid those problems, we can estimate the *embedding dimension* d_E, defined as the lowest (integer) dimension which unfolds the attractor, i.e. the minimal dimension for which foldings due to the projection of the attractor in a lower dimensional space are avoided. The embedding dimension is a *global* dimension and in general is different from the local dimension of the underlying dynamics.

The Embedding Theorem guarantees that if the dimension of the attractor is D_0, then we can unfold the attractor in a space of dimension d_E ($d_E > 2D_o$). It is worth noting that d_E is not a necessary condition for unfolding, but is sufficient.

The dimension of input layer of the Multi-Layer Perceptron will be then of dimension high enough in order that the deterministic part of the dynamics of the system is unfold.

3.3.1 Global False Nearest Neighbors

In practice, the method of *Global False Nearest Neighbors* proposed by Abarbanel [1], can be used to evaluate the embedding dimension d_E. Given a data space reconstruction in dimension d, with data vectors $\mathbf{S}_i = (s_i, s_{i+T},\ \ldots\ , s_{i+(d-1)T})$, where the time delay T is the first minimum of average mutual information (Eq. 6).

Let be $\mathbf{S}_i^{NN} = (s_i^{NN}, s_{i+T}^{NN},\ \ldots\ , s_{i+(d-1)T}^{NN})$, the nearest neighbor vector in phase space. If the vector $\mathbf{S}_i^{NN}$ is a *false neighbor* (FNN) of $\mathbf{S}_i$, 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 $\mathbf{S}_i$.

We define the distance ξ between points when seen in dimension $d+1$ relative to the distance in dimension d as

$$\xi_i \equiv \sqrt{\frac{R_{d-1}^2(i) - R_d^2(i)}{R_d^2(i)}}, \tag{7}$$

then

$$\xi_i = \frac{|s_{i+dT} - s_{i+dT}^{NN}|}{R_d(i)}. \tag{8}$$

As suggested by Abarbanel [1], $\mathbf{S}_i^{NN}$ and $\mathbf{S}_i$ can be classified as a false neighbor if ξ_i is a number greater than a threshold θ ($\xi_i \geq \theta$). 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. In this case we can modify the Eq. 8 as

$$\xi_i = \frac{|s_{i+dT} - s_{i+dT}^{NN}|}{R_A}, \tag{9}$$

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

$$R_A = \frac{1}{N}\sum_{i=1}^{N} |s_i - s_{av}|, \tag{10}$$

$$s_{av} = \frac{1}{N} \sum_{i=1}^{N} s_i. \tag{11}$$

We can list now some bells whistles and pitfalls of FNN method:

- The global FNN calculation is simple and fast [3].
- The FNN calculation applied to signals coming from two different outputs of the same dynamical system gives, in general, two different values of d_E. Then from each signal we will obtain different reconstructed coordinate systems, but both consistent with the original dynamical system.
- FNN method is valid even if the signal of interest results from a filtered output of a dynamical system [1, 5].
- If the signal is contamined by noise (assumed to be generated by an high dimensional system), it may be that the contamination will dominate the signal of interest and FNN will show the dimension required to unfold the contamination. Here, a simple byproduct of FNN calculation is an indication of noise level in a signal.

4 Test of the Constructive Methodology

4.1 Application to a Chaotic System

In [28], to shed more lights on the optimal values for m and T, we have chosen to use a synthetic chaotic time series based on the Mackey–Glass (MG) equation: $\dot{x}(t) = ax(t-\Delta)/\left(1 + x(t-\Delta)^c\right) - bx(t)$ with $a = 0.2$, $b = 0.1$ and $c = 10$. Varying Δ from 17 to 100 let vary D from 2.1 to 10 ([6]). We used MG time series with $\Delta = 17$ ($D_{17} \approx 2.1$ and $m_{17} \geq 5$) and $\Delta = 30$ ($D_{30} \approx 3.6$ and $m_{30} \geq 8$).

We have trained our NFS with a training set of 6000 patterns $\mathbf{v}_t$ and we have used a test set of 1000 vectors. Initial values of the parameters to be adjusted were set randomly. We noticed a fast convergence in the learning phase. For instance, the normalized generalization error is less than 1% after only 10 training epochs, for a 4-8-1 network, which is quite good. The training procedure was left running for 500 epochs, where usually the MSE was still decreasing but in small proportion. No sign of overfitting was observed.

The configuration of the NFS was d input nodes, $2d$ hidden nodes and 1 output node, (d-$2d$-1) while d scanned the range 2 to 15. Time delay T between measurements was also tested for values between 1 and 24.

In Fig. 2, a sharp decreasing of $\log(error)$ can be noticed, as soon as d is big enough. About T, no clear value seems optimal for low forecasting errors: apparently, best results are achieved for short T. Indication from dynamical systems literature based on the evaluation of the first minimum of the average mutual information of the time series would have suggested a value of T of about 11.

[3] We notice a very efficient implementation of FNN algorithm developed by by F. Montarsolo [23], on the basis of the work by Nene and Nayar [24].

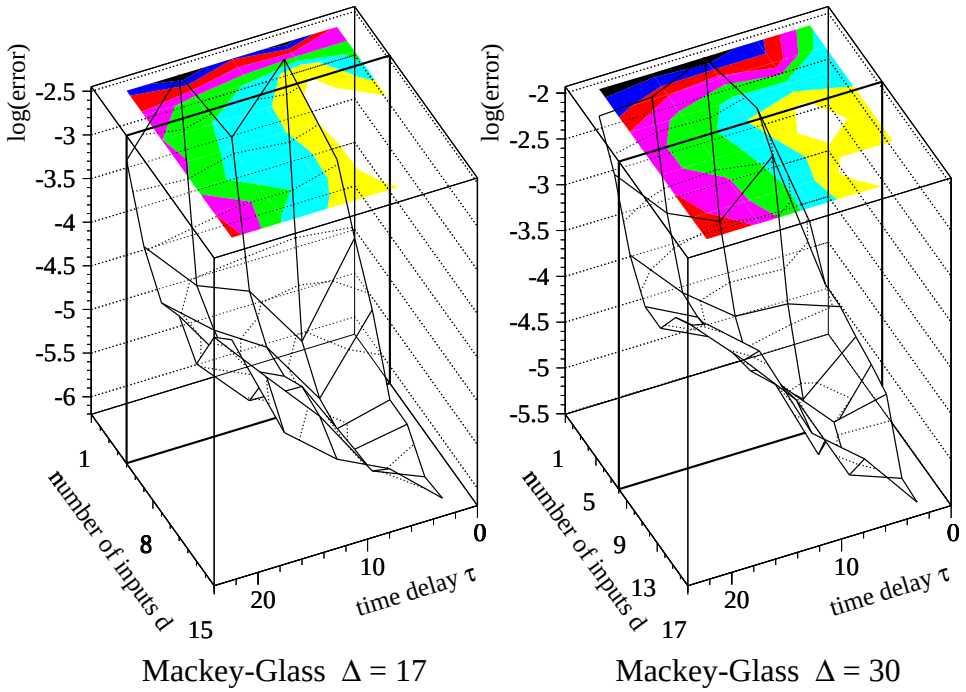

Figure 2: Dependence of the forecasting error in function of the number d of inputs and of the time delay T. Takens–Mañé theorem requires $d \geq 5$ (left) or $d \geq 8$ (right).

4.2 Application to a Non-Linear System

4.2.1 MATLAB/Simulink model

In [25], another dedicated computer experiment has been developed in order to test the constructive methodology for time series prediction. A non-linear dynamic system able to show different dynamic behavior for different amplitude of its input has been implemented using a Matlab/Simulink environment (Fig. 3).

The target of this experiment was to build-up a non parametric model using only knowledge extracted from output signals of the simulated system.

In preliminary experiments, the application of classical linear identification methods lead to poor results, such as a strong dependency of the model from the working point, and impredictable results in connection with changes on time scale.

4.2.2 Data Set

A possible approach to study the input/output relations of a system is to input it a random series (random stimulation input - RSI), and then to study its inputs and outputs. This approach is mimicking a blind acquisition on a real plant where it is very easy to collect data but it is very difficult to have any control of the system inputs shape and amplitude.

In the numerical experiment, the input of the system was a train of steps with random

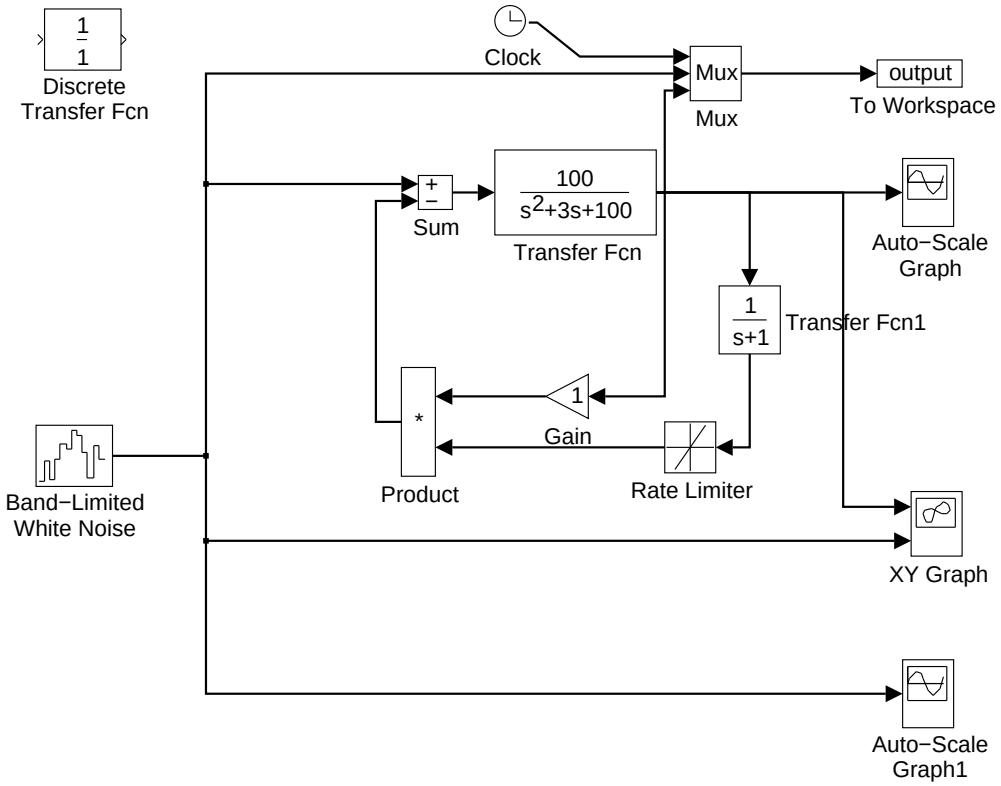

Figure 3: MATLAB/SIMULINK representation of the non-linear system.

amplitudes. The length of the plateau was 3 times the period of the fundamental frequency of the dynamical system.

4.2.3 Data Analysis

If we stimulate the circuit with RSI changing every 1/5 Hz, the power spectrum shows a peak at the left end, in correspondence of the frequency of changing of the RSI.

In Fig. 4, we present the plot of average mutual information $I(T)$ of the output signal. The first minimum of $I(T)$ is for $T = .16$ sec.

As shown in Fig. 5, the FNN ratio goes to a minimum for $d \geq 4$. Also, as the input signal is a train of steps, two past inputs are sufficient to describe the external excitation signal. $d \geq 4$ and two past inputs imply then that we should use two past outputs to reconstruct the dynamical system.

4.2.4 Neural Network Design

The model of the dynamical system was made up by a Multi-Layer Perceptron with d inputs $2d$ sigmoidal nodes in a first hidden layer, d sigmoidal nodes in a second hidden layer, and 1 output

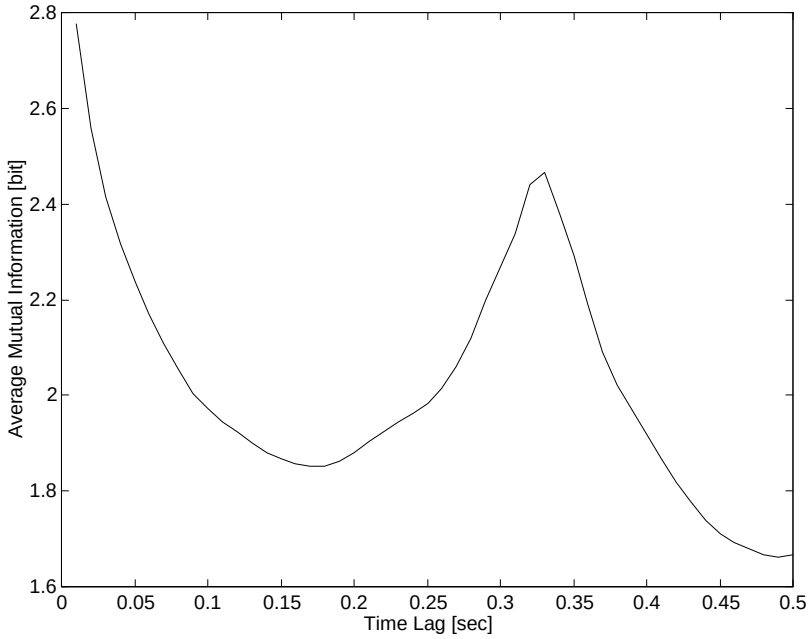

Figure 4: Average Mutual Information of the output of the simulated non-linear system.

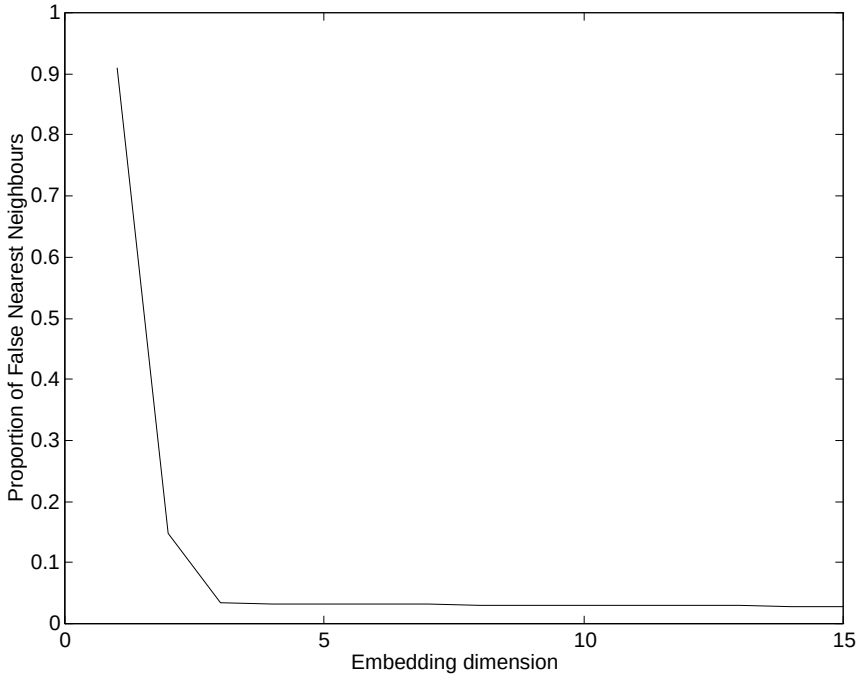

Figure 5: False Nearest Neighbors of the output of the simulated non-linear system.

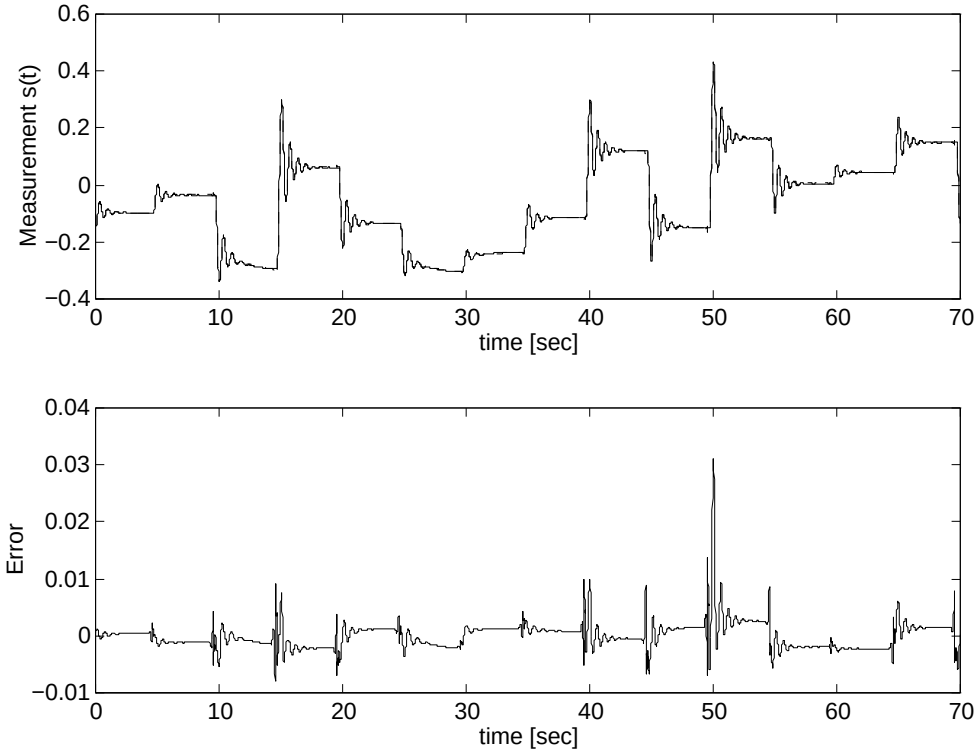

Figure 6: *Above:* Output signal from the simulated non-linear system AND its estimation by a 4-8-1 MLP. The training set was the 5000 first steps (i.e. up to 50 sec). *Below:* Difference between the system output and its MLP approximation. *Notice the different vertical scales.*

linear node (d–2d-d-1). The dimension of the input layer d of the MLP was set equal to the dimension of the reconstructed space of the dynamical system, so $l = 4$.

4.2.5 Results

A data base of labeled patterns $P_k = [(u_k, u_{k+T}, s_k, s_{k+T});\ s_{k+2T}]$ was obtained by stimulating the non-linear circuit with a RSI of 2 sec. $T = .16$ sec (u are the inputs and s the outputs of the system). The data base was subdivided in a learning, a test and a validation sets of 5000, 1000, and 1000 patterns. Then, the learning set was shuffled and a 4-8-4-1 MLP was trained on it.

In Fig. 6, the very good quality of approximation of the behavior of the dynamical system obtained by the MLP is shown.

4.3 Application to a Steam Turbine

In [21], the constructive methodology was applied to the design of a Neural Network for forecasting the vibration corresponding to the correct working state of 150 MW Siemens steam turbine at a given running point.

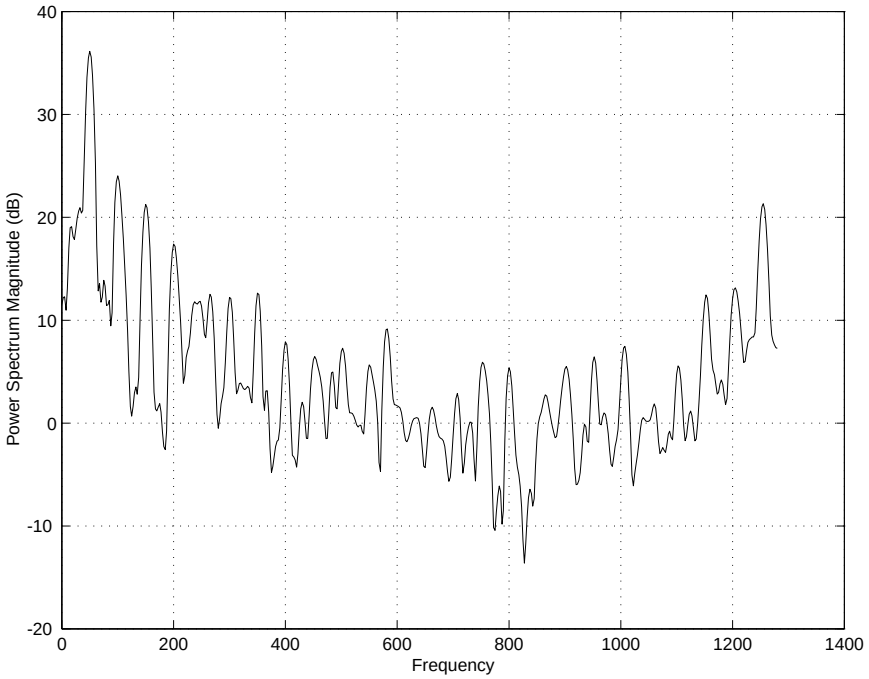

Figure 7: Power Spectrum of the time series of 1024 points used for training the MLP. The series has been recorded in 400 ms by an accelerometer coupled to a 150 MW steam turbine.

Information of the working state of the turbine are obtained through the measurements of piezoelectric accelerometers coupled to part of the turbine and collecting the various vibrations.

The considered data were two time series of 400 ms (1024 points each) recorded with an interval of 8 hours.

4.3.1 Data Analysis

The power spectrum of the first time series is displayed in Fig. 7. The main period of the time series is clearly visible, but a lot of secondary frequency peaks should be noticed, apart of the harmonics of the main period. The multiplicity of the secondary peaks and the decay of the power spectrum is a sign of the nonlinearity of the system [1].

The time lag corresponding to the first minimum of the average mutual information is $T = 4.68ms$ (Fig. 8). This interval is a candidate for the time lag between the components of the state vectors to be input to the neural network forecaster.

The False Nearest Neighbors algorithm leads to an estimate of the embedding dimension of $d_E = 5$.

4.3.2 Neural Networks Forecaster Design and Results

The Multi-Layer Perceptron forecaster was designed using the previous evaluations of the optimal time lag T and the minimal embedding dimension d_E, we can design now The number of inputs

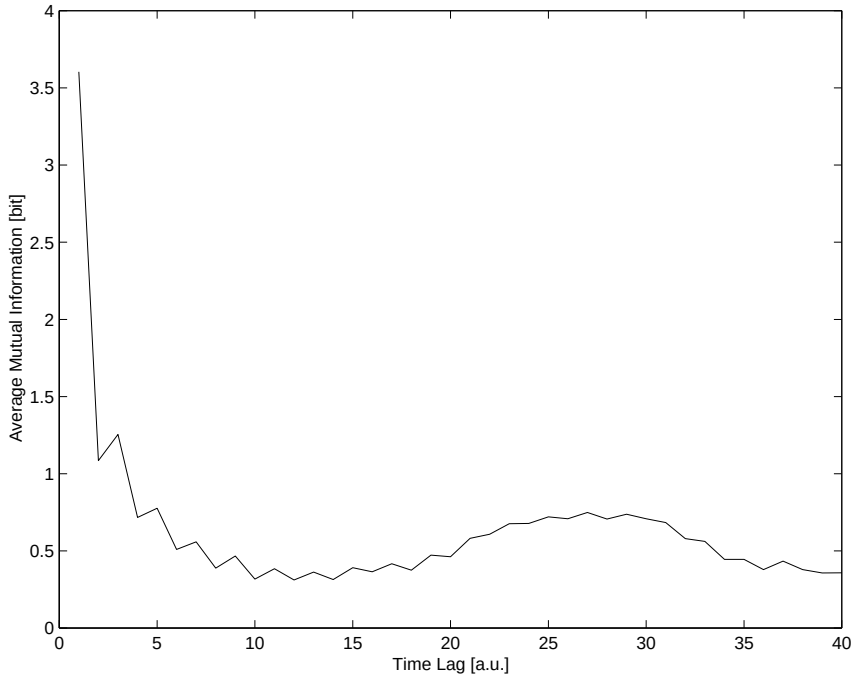

Figure 8: Average Mutual Information Spectrum for the steam turbine.

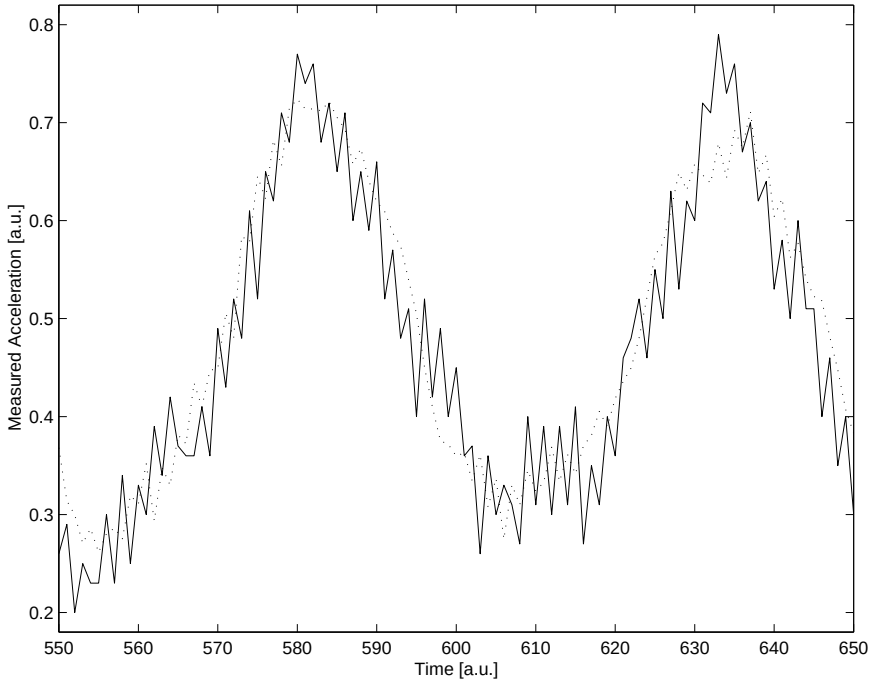

Figure 9: Steam turbine time series (continuous line) and its forecast (dotted line) by a 5-7-1 sigmoidal Multi-Layer Perceptron.

of the network was set to 5. The training patterns were the 5-dimensional state vectors with components separated by 4.68 ms. The requested output was the value of the time series 1.56 ms later than the most recent component of the state vector. The best results were obtained using a 5-7-1 MLP.

The training set of the network consisted of 600 vectors. A test set of 379 vectors was used in order to stop before over-training. These two sets were built with the first time series.

The obtained forecaster was then validated with 977 vectors built on the second time series (which was recorded 8 hours after the first one, and shows a slightly different power spectrum). In Fig. 9 we can seen that the forecaster is able to follow the second time series.

5 Singular Spectrum Analysis and Time Series Forecasting

5.1 Singular Spectrum Analysis

The proposed constructive methodology can not be directly applied to forecasting discontinuous or intermittent signals, as the universal function approximation theorems for neural networks [4] and fuzzy systems [35] require the continuity of the function to be approximate.

In [3], it has been proposed an extension of the proposed approach to the design of neural networks base time series forecaster to prediction of avoid the effect of discontinuous or intermittent signals using a pre-processing of the raw time series based on the Singular-Spectrum Analysis (SSA) [16, 26, 33, 17].

In SSA the state vector $\mathbf{S}_i = (s_i, s_{i+1}, \ \ldots \ , s_{i+M-1})$ is an augmented vector of the series s, made up by a given number of samples M.

The cornerstone of SSA is the Karhunen-Loève expansion or Principal Component Analysis (PCA) [30] that is based on the eigenvalues problem of the lagged covariance matrix Z_s.

The original series can be expanded with respect to the orthonormal basis corresponding to the eigenvectors of Z_s

$$s_{i+j} = \sum_{k=1}^{M} p_i^k u_j^k, \quad 1 \leq j \leq M, \quad 0 \leq i \leq N - M \tag{12}$$

where p_i^k are called *principal components* (PCs) and the eigenvalues u_j^k are called the *empirical orthogonal functions* (EOFs) [4].

In [32, 10, 33, 13, 17, 9] many applications of Singular Spectrum Analysis have been presented, including noise reduction, detrending, spectral estimate, and prediction.

5.2 Reconstructed components and reconstructed waves

Following Vautard and Ghil [33], suppose we want to reconstruct the original signal s_i starting from a SSA subspace $\mathcal{A}$ of k eigenvectors. By analogy with Eq. 12, the problem can be formalized as the search for a series $\hat{s}$ of length N, such that the quantity

[4] The EOFs constitute an orthonormal basis.

$$H_{\mathcal{A}}(\hat{s}) = \sum_{i=0}^{N-M} \sum_{j=1}^{M} (\hat{s}_{i+j} - \sum_{k \in \mathcal{A}} p_i^k u_j^k)^2 \tag{13}$$

is minimized. In other words, the optimal series $\hat{s}$ is the one whose augmented version $\hat{S}$ is the closest, in the least-squares sense, to the projection of the augmented series S onto EOFs with indices belonging to $\mathcal{A}$. The solution of the least-squares problem of Eq. 13 is given by

$$\hat{s}_i = \begin{cases} \frac{1}{M} \sum_{j=1}^{M} \sum_{k \in \mathcal{A}} p_{i-j}^k u_j^k & \text{for} \quad M \leq i \leq N - M + 1 \\ \frac{1}{i} \sum_{j=1}^{i} \sum_{k \in \mathcal{A}} p_{i-j}^k u_j^k & \text{for} \quad 1 \leq i \leq M - 1 \\ \frac{1}{N-i+1} \sum_{j=i-N+M}^{M} \sum_{k \in \mathcal{A}} p_{i-j}^k u_j^k & \text{for} \quad N - M + 2 \leq i \leq N. \end{cases} \tag{14}$$

When $\mathcal{A}$ consists on a single index k, the series $\hat{s}$ is called the kth RC, and is denoted by $\hat{s}^k$. RCs have additive properties, i.e.

$$\hat{s} = \sum_{k \in \mathcal{A}} \hat{s}^k \tag{15}$$

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

$$s = \sum_{k=1}^{M} \hat{s}^k \tag{16}$$

Note that, despite its linear aspect, the transform changing the series s into $\hat{s}^k$ is, in fact, non-linear, since the eigenvectors u^k depend non-linearly on s.

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 $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_L$ disjoint subspaces, then a *reconstructed wave* (RW) Ω_l ($l = 1, ..., L$) is defined as [3]:

$$\Omega_l = \sum_{k \in \mathcal{A}_l} \hat{s}^k, \quad 1 \leq l \leq L, \tag{17}$$

and, from Eq.s 16 and 17, one can obtain:

$$s = \sum_{l=1}^{L} \Omega_l, \tag{18}$$

id., the original series s can be recovered as the sum of all the individual RWs.

5.3 Reconstructed Waves Forecasting

Concerning the application of SSA to time series prediction, that is the main interest of the present tutorial, it is supported by the following argument [33]: Since the PCs are filtered version

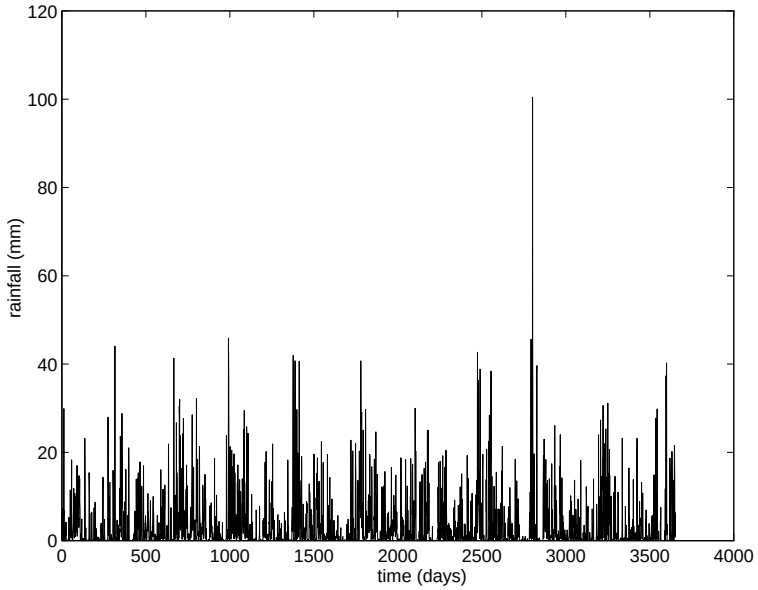

Figure 10: Mean Station: Daily rain millimeters. Period 01/01/1958 - 12/31/1967.

of the signal and typically band-limited, their behavior is more regular than that of the raw series s, and hence more predictable.

Vautard and Ghil in [33] fit an autoregressive (AR) model for each individual PC using the AR coefficient estimate of Burg [2], while Lisi, Nicolis and Sandri [17] used Multi-Layer Perceptrons in order to estimate the PCs.

In order to reduce the computational costs, in [3] it has been suggested:

- to decompose the raw series s in RWs corresponding to SSA subspaces with equivalent explained variance, and then
- to predict each RW using Multi-Layer Perceptrons designed following the constructive approach described in Sect. 3; and finally
- to obtain the prediction of the raw signal by addition of the forecasts of the RWs.

6 Application to Rainfall Forecasting

6.1 Data Set and Methods

In [3] the application of the previous described forecasting approach concerns the forecasting of daily rainfall intensities series was presented. The series were collected by 135 stations located in the Tiber river (Italy) basin in the period 01/01/1958 - 12/31/1967.

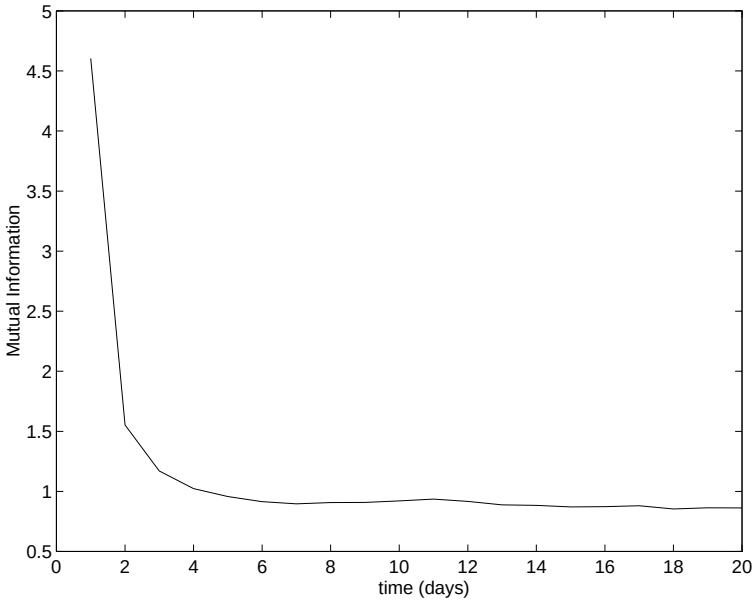

Figure 11: Mean Station: Mutual Information. The first minimum of is for $t = 7$.

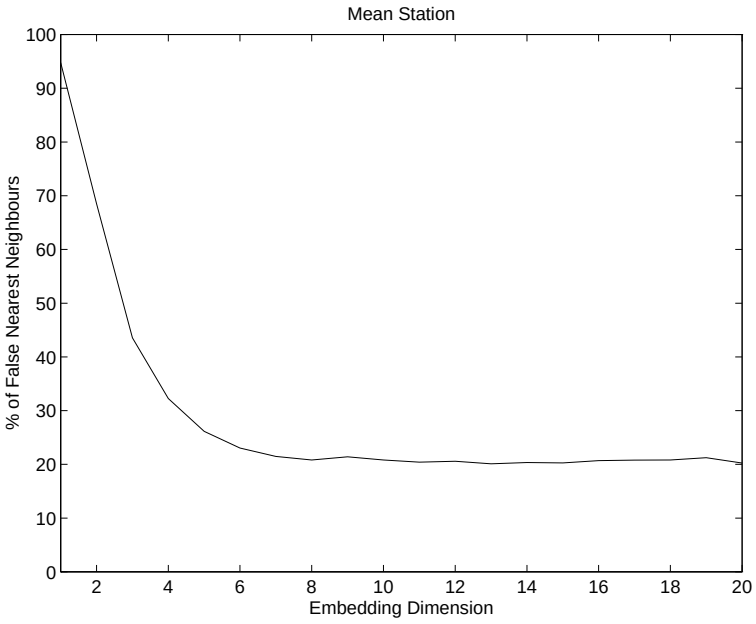

Figure 12: Mean Station: Global False Nearest Neighbors.

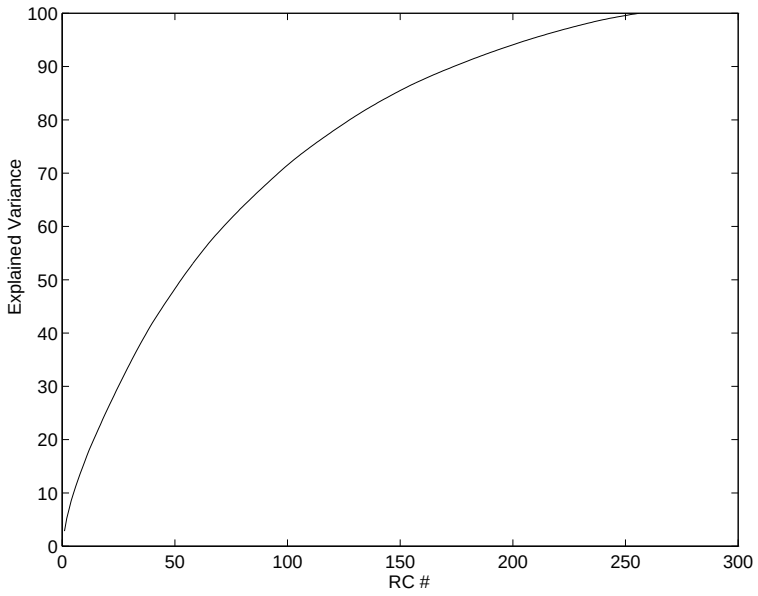

Figure 13: Mean Station: Explained variance of the augmented vectors related to an increasing number of RCs.

The data processing started by considering the series of the Mean Station (MS), defined as the average of all 135 rainfall intensity series (Fig. 10). We notice the high discontinuity of the obtained signal.

Fig. 11 shows the graph of the mutual information of the MS's time series. Its first minimum is for $T = 7$. This value has been used as the time lag for the computation of Global False Nearest Neighbors. The graph of FNN is shown in Fig. 12. Till $d = 6$ the curve decreases with the growing of dimension, and then reaches a plateau of 20%. The plateau is the symptom of the presence of high dimensional noise. The evaluation of the embedding dimension is $d_E = 6$.

Following the constructive approach described in Sec. 3, predictor based on a Multi-Layer Perceptron has been designed. The MLP was made up by two hidden layers of 5 units, an input layer of 6 inputs spaced by a time lag of 7 days. The obtained results were very poor, due to the discontinuity of the hydrological variable.

In order to reduce the effects of the discontinuities, the Singular-Spectrum Analysis to MS series was then applied. A window length $M = 256$ was select. Fig. 13 presents the explained variance of the reconstructed signal using an increasing number of RCs.

Then, using the method shown in Sec. 5.2, from the raw MS series we obtained 10 waves $\Omega_1, ..., \Omega_{10}$ reconstructed from 10 disjoint sub-spaces, each of them representing a 10% of the explained variance (see Tab 1). Waves $\Omega_1, ..., \Omega_6$ (corresponding to the first 76 RCs), are enough regular, while the remaining waves (corresponding to subspaces with low eigenvalues) are more complex.

We designed a neural predictor based on a MLP for each individual wave of the MS, following the constructive approach described in Sect 3. For each wave we obtained $T = 7$ and $d_E = 6$, as

Table 1: Reconstructed waves (RWs) from disjoint SSA subspaces (each of them corresponding to 10% of the explained variance) and corresponding reconstructed components (RCs).

RWs	RCs
Ω_1	1-6
Ω_2	7-16
Ω_3	17-27
Ω_4	28-40
Ω_5	41-56
Ω_6	57-76
Ω_7	77-97
Ω_8	98-137
Ω_9	138-181
Ω_{10}	182-256

for the raw MS, and the best results were obtained by MLPs with two hidden layers of 5 neurons, and a size of the input layer of five neurons [5].

For each wave (corresponding to 3652 daily samples), we obtained 3645 associative couples, each of them consisting of a window of 6 elements delayed 7 days, as input, and the next-day rainfall intensity, as output.

Each MLP was trained using the first 2000 associative couples (*training set*), using the error back-propagation algorithm with momentum [34], and batch presentation of samples. The following 1000 associative couples (*validation set*) were used in order to implement an early stopping of the training procedure. The remaining 645 were used for measure the quality of the forecasting of the reconstructed wave (*test set*).

6.2 Results

The prediction results on waves $\Omega_1, .., \Omega_6$, corresponding to 60% of the explained variance (first 76 RCs), are good. Fig.s 14 and 15 show the results obtained for wave Ω_5, while for waves $\Omega_7, .., \Omega_{10}$, corresponding to subspaces with low eigenvalues, the predictions are unsatisfactory.

Following the criteria of the *best prediction* [17] in the Eq. 18 $\Omega_7, .., \Omega_{10}$ were excluded, as if enclosed in the addition, made worse the overall prediction.

The sum of the prediction of the 6 waves at 1 day ahead gives a signal well correlated with the original rainfall intensity of the MS, as shown in Fig. 16 and Fig. 17. Note that in the comparison shown in Fig. 16 the predicted signal is set to 0 when negative.

Moreover, preliminary results for the forecasting of the rainfall series of individual stations are also in good agreement with data [23]. We notice that in some cases the best forecasting results have been obtained using the SSA space generated by the MS instead of the one obtained from data of the individual station.

[5] d_E is an upper bound for the size of the input layer of the MLP.

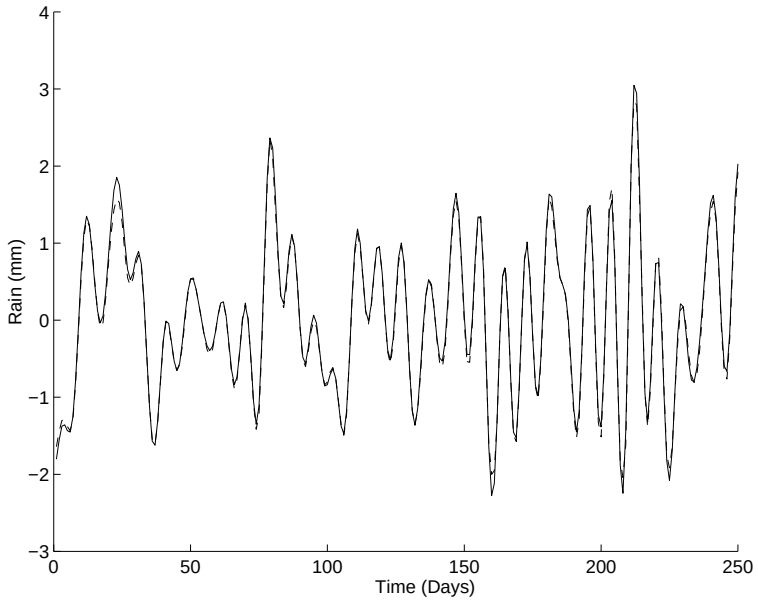

Figure 14: Mean Station: 1 day ahead forecasting for wave Ω_5. Period 3/19/66 - 12/4/66.

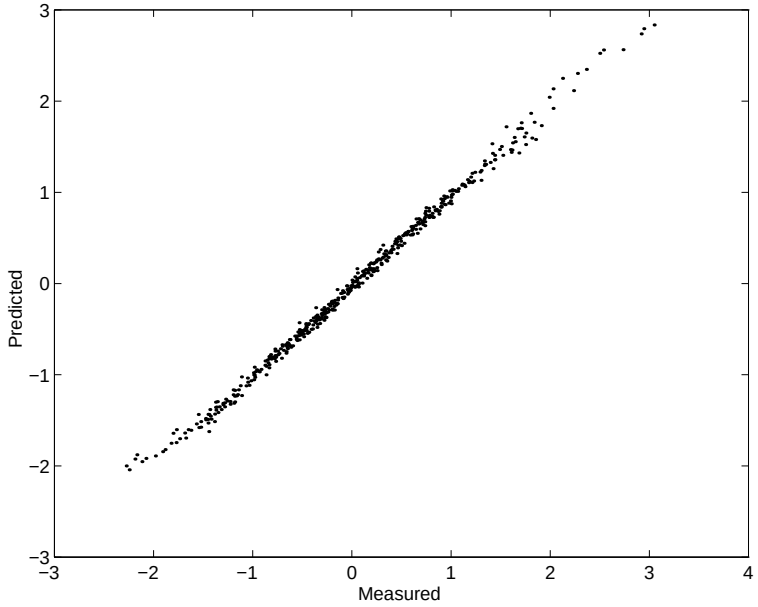

Figure 15: Mean Station: scatter plot - 1 day ahead forecasting wave Ω_5 on the test set.

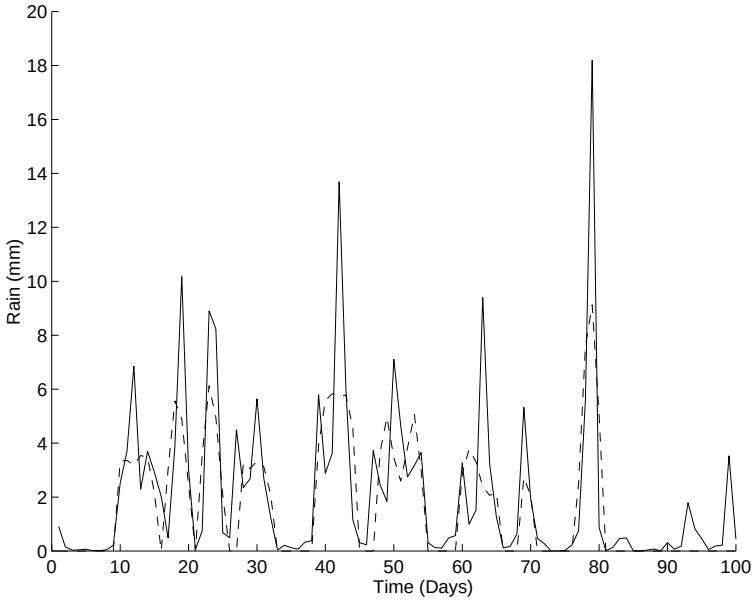

Figure 16: Mean Station: 1 day ahead forecasting using RCs 1-76. Period 3/19/66 - 12/4/66.

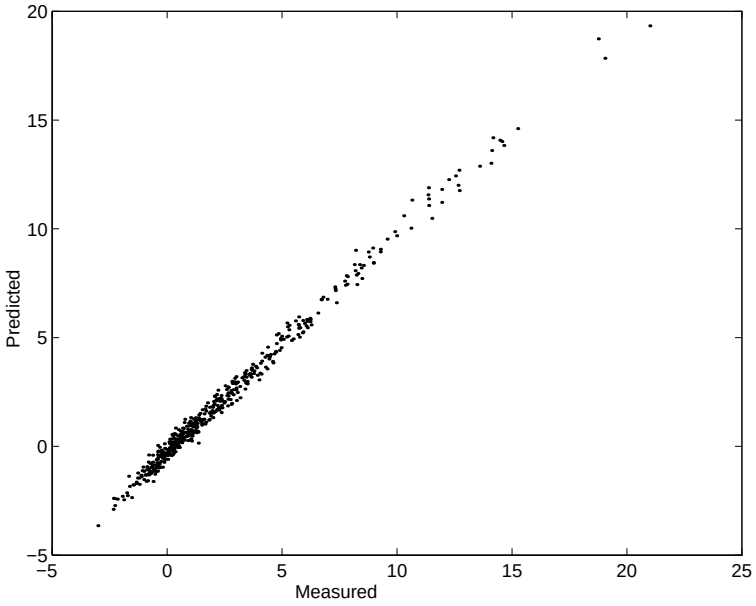

Figure 17: Mean Station: scatter plot - 1 day forecasting using RWs 1-6 (i.e. RCs 1-76) on the test set.

7 Conclusions

In the last years, neural networks have been extensively tested on non-linear dynamic systems modeling and forecasting [11]. Those applications are supported by the *universal approximation theorems* [4, 12, 35, 36], that, unfortunately, are not constructive: In facts, no information can be extracted from the theory in order to define the structure of the neural network based approximator. In other words, the neural network theory doesn't give any general suggestion about: dimension of the data window (i.e. input layer of the MLP), sampling rate of the input data, dimension of the hidden layer, dimension of the training set.

On the other hand, results achieved in the theory of chaotic systems point out very relevant elements which can be extracted from the measurement of time series of one variable of the non-linear dynamic system. One of these results is given by the Takens-Mañé theorem [29, 18] about the sufficient dimension of an Euclidean space to secure a fair representation of the true strange attractor of the underlying system.

In this tutorial, we have examined pragmatically the sensitivity of the method to the exact value of the embedding dimension for the case of the chaotic system obtained by the Mackey-Glass equation[19, 28]. As expected, this dimension is set by the lower bound in the number of components of the state vector as given by the Takens-Mañé theorem. The relevant observable for neural models of non-linear dynamic systems is a sharp transition in the quality of the forecaster in function of the number of components of the temporal window

On the contrary, the quality of the forecaster is less sensitive to the time lag between the components of the state vector. This apparent lack of sensitivity is directly related to the Takens-Mañé theorem which, by hypotheses, is valid for any time lag. In practice, it can be expected that the time lag depends of the noise affecting the dynamical system.

Supported by results and prescriptions related to the Takens-Mañé theorem [29, 18], our constructive methodology for shaping a neural model of a synthetic non-linear process has been applied to the design neural model of the vibration dynamic of a Siemens steam turbine [21]. The proposed constructive methodology has been shown to be very easy to use, leading to useful results.

We extended our methodology for signal forecasting to the case of discontinuous and intermittent signals [3]. In order to avoid the effect of the discontinuities, we have proposed the application of the Singular-Spectrum Analysis (SSA) [16, 26, 33, 17] that permits to decompose the original signal in a family of more regular temporal series (reconstructed waves). This extended methodology has been successfully applied to the forecasting of rainfall intensities series collected by 135 stations distributed in the Tiber river basin for a period of 10 years.

The integration of the Neural Network estimation and the Chaos Theory proposed in the present work should be very useful in order to develop the new generation of the predictive state estimator for non linear dynamic systems.

Acknowledgments

This works was partially supported by INFM and MURST. Part of the work of Léonard Studer has been supported by a grant from the Swiss National Science Foundation. The original source

code of the Neuro-Fuzzy System used here was written by Franco Casalino and Renato Caviglia from the University of Genoa (Italy). Fabio Montarsolo and Daniela Baratta from the University of Genoa (Italy) have contributed to the development of a Matlab (r) Toolbox supporting all the algorithms described in this tutorial. André Perrenoud at Vibro-Meter S.A. (Fribourg, Switzerland) has kindly supplied us with data recorded from the steam turbine. We especially thank Riccardo Parenti at Ansaldo Ricerche (Genoa, Italy) and Giovambattista Cicioni at IRSA-CNR (Roma, Italy) for the continuous collaborations and useful discussions.

References

[1] H.D.I. Abarbanel. *Analysis of Observed Chaotic Data*. Springer, New York, USA, 1996.

[2] J.P. Burg. Maximum entropy spectral analysis. In D.G Childers, editor, *Modern Spectrum Analysis*, page 34. IEEE Press, New York, 1978.

[3] Gb. Cicioni, F. Masulli, and F. Montarsolo. Forecasting discontinuous signals using singular spectrum analysis and neural networks. In *Proceedings of NEU99 - Financial Applications of Neural Networks and Fuzzy Systems*, Venice (Italy), 1999. *(in press)*.

[4] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals, and Systems*, 2:303–314, 1989.

[5] M.E. Dave. Reconstruction of attractors from filtered time series. *Physica D*, 101:195–206, 1997.

[6] J.D Farmer. Chaotic attractors of an infinite-dimensional system. *Physica D*, pages 366–393, 1987.

[7] A. Fraser. Information theory and strange attractors. Technical Report PhD thesis, University of Texas, Austin, 1989.

[8] A. Fraser and L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review*, 33:1134–1140, 1986.

[9] M Ghil. The SSA-MTM toolkit: Applications to analysis and prediction of time series. In B. Bosacchi, J.C. Bezdek, and D.B. Fogel, editors, *Application of Soft Computing*, volume 3165 of *Proceedings of SPIE*, pages 216–230, Bellingham, WA, 1997.

[10] M. Ghil and R. Vautard. Rapid disintegration of the wordie ice shelf in response to atmospheric warning. *Nature*, 350:324, 1991.

[11] S. Haykin. *Neural Networks. A Comprehensive Foundation*. Macmillan College Publishing, New York, 1994.

[12] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[13] C.L. Keppenne and M. Ghil. Adaptive filtering and prediction of noisy multivariate signals: an application to subannual variability in atmospheric angular momentum. *International Journal of Bifurcation and Chaos*, 3:625–634, 1993.

[14] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1995.

[15] B. Kosko. Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, 43:1329–33, 1994.

[16] R. Kumaresan and D.W. Tuffs. Data-adaptive principal component signal processing. In *IEEE Proc. Conf. on Decision and Control*, page 949, Albuquerque, USA, 1980. IEEE.

[17] F. Lisi, O. Nicolis, and M. Sandri. Combining Singular-Spectrum Analysis and neural networks for time series forecasting. *Neural Computation*, 2:6–10, 1995.

[18] R. Mañé. 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*, pages 230–242, Warwick 1980, 1981. Springer-Verlag, Berlin.

[19] M.C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197:287, 1977.

[20] F. Masulli. Bayesian classification by feedforward connectionist systems. In F. Masulli, P. G. Morasso, and A. Schenone, editors, *Neural Networks in Biomedicine - Proceedings of the Advanced School of the Italian Biomedical Physics Association - Como (Italy) 1993*, pages 145–162, Singapore, 1994. World Scientific. *(invited)*.

[21] F. Masulli, R. Parenti, and L. Studer. Neural modeling of non-linear processes: Relevance of the Takens-Mañé theorem. *Internal Report of DISI, University of Genova (Italy). International Journal on Chaos Theory and Applications, (submitted)* .

[22] F Masulli and L. Studer. Neuro-fuzzy system for chaotic time series forecasting. In B. Bosacchi, J.C. Bezdek, and D.B. Fogel, editors, *Applications of Soft Computing - SPIE Proceedings Series*, volume 3165, pages 205–215, San Diego, CA, 1997. SPIE - Bellingam, WA, USA. *(invited paper)*.

[23] F. Montarsolo. A toolkit for discontinuous series forecasting. Laurea thesis in computer science (in italian), DISI - Department of Computer and Information Sciences, University of Genova - Genova, (Italy), 1998.

[24] S.A. Nene and S.K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.

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

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

[27] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–362. MIT Press, Cambridge, 1986.

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

[29] F. Takens. 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*, pages 366–381, Warwick, 1981. Springer-Verlag, Berlin.

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

[31] J. Vastano and L. Rahman. Information transport in spatio-temporal chaos. *Physical Review Letters*, 72:241–275, 1989.

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

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

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

[35] L. Wang and J.M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Trans. on Neural Networks*, 5:807–14, 1992.

[36] L. X. Wang. *Adaptive Fuzzy Systems and Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[37] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–352, 1965.

[38] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transaction on Fuzzy Systems*, 4:103, 1996.