# Nonlinear Time Series Forecasting Using Neural Networks and Local Modelling

José Luis Guillén Aguirre

J. Jesús Rico Melgoza

División de Estudios de Posgrado

Facultad de Ingeniería Eléctrica

Universidad Michoacana de San Nicolás de Hidalgo

**Abstract:** Forecasting nonlinear time series is a problem of great importance in many areas of science and technology, such as electrical markets, weather forecasting, financial markets, generation companies and others.

In this paper, nonlinear time series are characterized by a set of local linear models that are identified using historical data. Local models represent different operation regimes. These regimes are obtained through a classification of historical patterns revealed from the data.

An advantage of local modeling is that highly nonlinear systems can be represented by a set of simpler systems. Thus, analysis and identification are substantially simplified. In our study cases, local modeling has proven very fast for forecasting and the learning stage.

To prove our forecasting approach several classic chaotic systems are considered, and also one-step ahead and multiple-step ahead predictions are carried out with satisfactory results.

**Keywords:** Forecasting, Nonlinear Time Series, Neural-Networks, Local Modelling.

# 1 Introduction

An important number of physical phenomena, whether natural or those that arise from the interactions of man-made systems, are subjected to nonlinear behaviors, and can generate complex dynamics like chaos. Chaos Theory has achieved great transcendence in many areas of engineering since it was identified when trying to recreate mathematical models to predict weather. Currently, Chaos Theory enclosed a set of tools aimed to the study and comprehension of nonlinear systems. In this paper a tool useful for predicting the response of nonlinear systems in the short term is presented.

Mathematical complexity of nonlinear systems is such that to generate deterministic mathematical models it is impossible in practice. For instance weather forecasts with a prescribed accuracy cannot be done beyond a few days or weeks. For long time it was believed that nonlinear systems behavior was a phenomenon of stochastic nature, however, in the light of Lorenz studies [1] over its climate-model, was concluded that such behavior was far from being random, additionally, it showed patterns in its evolution in time, which opens the possibility to predict nonlinear behavior in the short run.

An important step in nonlinear systems understanding was given by Takens, [2] with a theorem that takes his name and that establishes the possibility of recreating global behavior in a systems given just one observable state variable of the system. Takens’ theorem is indeed a remarkable result that tells us that any deterministic nonlinear dynamical system with a *d*-dimensional state space, the state can be effectively reconstructed by observing 2*d*+1 time lags of any one of its inputs. That is, if you suspect that the system is nonlinear and that it has *d* state dimension, then instead of building a state-space model, you can do away with representing states and just build an autoregressive model directly on the observations that nonlinearly relates previous outputs and the current output.

Based on the aforementioned studies, we are in possibility to create tools capable of predicting the behavior in the short and that show chaotic characteristics. In this paper a tool based on Takens’ theorem, Linear Local Models and Self-organizing maps is presented. For validation of our models, several classical non-linear differential equations are considered; one-step ahead and multiple-step ahead- predictions are performed for these equations, showing accuracy of our results.

# 2 Some salient features of Chaotic Systems

Chaotic systems are best defined by listing their principal characteristics:

* It is a deterministic system
* It is a dense set of points with periodic orbits.
* Has high sensitivity to initial conditions.

The first point implies the fact that the system could be represented by a differential equation system. Density requires that the set of system’s states generate an infinity set of orbits and such orbits don’t intersect themselves. Finally, third point settles that, by modifying infinitesimally the initial conditions, generated trajectories will be completely different at a later time.

A common example of a chaotic system is the Lorenz’s System:



Graphically, the system’s trajectory is shown in Figure 1. The geometric space that occupies is denominated *attractor*.

Figure1

**Figure 1**. Lorenz system plot

Initial conditions for this system are (0, 0, 0), and the relevant parameters values will be =10, *r*=28 and *b*=8/3. Variable  is known as *Prandtl's number* and *r* as *Rayleigh's number*. This set of equations is a simplification of a system describing a three-dimensional flow of a fluid in a rectangular box, heated from the bottom. At first, it was used to simulate atmospheric convection’s effect in a medium scale. This system tends to exponential divergence in the face of minimal changes in its initial conditions, which can be interpreted as evidence for the difficulty to predict the evolution of flows in turbulent regimes [3].

Figure2

**Figure 2**. Variable *x* versus time for the Lorenz’ System.

The behavior of variable *x*, plotted versus time, with the same initial conditions is shown in figure 2.

This features make long term predictions impossible, although forecast for the short term may be carried out with good accuracy.

# 3 Measuring in large nonlinear systems

In practice, the nonlinear systems of interest are very large and make measurements of variables a very hard and costly task. Although the advance of digital measuring is making devices cheaper, in actual systems it is difficult to justify the deployment of a large number of measuring devices. In fact, full observation of complex systems is intrinsically impossible.

It is common to have measured time series for a system, and extract dynamics patterns in order to predict future behavior; in the case of climate a measured variable can be temperature; in an electrical system, voltages or currents. In this context Takens’ theorem has profound practical consequences. In principle, it establishes that a time series corresponding to one state variable of an *n*-order dynamical system will be enough to determinate if the system is chaotic and to reproduce its global behavior. The theorem is stated in detail later in this paper.

# 4 Recurrence Plots

Chaotic behavior can be proved using several methods, as *Poincaré Maps*, *Lyapunov Exponents*, *Recurrence Plots*, among others. In this paper Recurrence Plots are employed. Recurrence Plots reveal no stationary characteristics and aperiodicity degree in time series. With Recurrence Plots it is possible to detect hidden patterns and similarity among them, and structural changes in data.

The fundamental assumption in a Recurrence Plot is that a time series is just one realization of some dynamical process. Highly complex processes have a big number of factors affecting his dynamics. In a particular case we may have the realization of just a single variable: in climate, for example, temperature measurements; in stock exchange, average closing share prices. In both cases we have multidimensional dynamical systems with a one-dimensional output.

It has been proved mathematically that it is possible to recreate a topologically equivalent portrait of a multidimensional system using one observable time series. The basic idea underlying this statement is that the effect of all other system's variables (non-observable) is reflected in the observed output time series. Furthermore, rules governing the original system’s behavior can be recovered from its output.

A Recurrence Plot of a sequence of *n* point *x*1, *x*2,..., *x*n, is the visualization of a recurrence matrix of that sequence. Color's element (*i*,*j*) and (*j*,*i*) in the recurrence plot is black if the distance between the *i*-th and *j*-th point in time series lies in some threshold corridor



for some adequate norm, where  is the minimum value and the maximum value of the threshold, and is white for any other distance. This technique, applied to a time series, enable us to visualize temporal correlations existing in the series. Figure 3 shows recurrence plots for several time series.

figure3

**Figure 3**. Recurrence plot for different time series.

Recurrence plots display structures and patterns that indicate time series nature. Figure 3(a) is a recurrence plot of a time series of random data with a uniform distribution and shows no defined pattern, as expected, since no correlation exists between a given value and previous or subsequent values. Figure 3(b) shows data from a sine function. This plot shows a perfect defined pattern, given that is a periodic function. Figure 3(c) show the plot for a time series for the Lorenz's System. Is in this last series where Recurrence Plot gets valuable, since it allows visualizing the patterns that arise in a chaotic time series. Even if is not a perfect defined pattern, like sine function in Figure 3(b), is much more defined than that for random time series in figure 3(a). This also shows that a correlation exists between values of the Lorenz's time series.

After proving that a time series show chaotic behavior, we appeal to Taken Theorem to determine additional parameters, needed to reconstruct the system.

# 5 Takens’ Theorem

Takens’ theorem establishes that for practically any scalar function *h*(*k*), that is to say, some observations’ set, a wide range of questions can be answered about original system dynamics, through observation of a space defined by delayed values of *h*(*x*). System parameters can be calculated and, even, interpolating the delay embedding space, forecast its behavior.

The theorem’s objective is to obtain a generic result, therefore, provides an approximate model that can be useful, justifying the use of delay embedding.

From the aforementioned, arise two central concepts: Delay values and embedding space or dimensionality *m*. Through use of these concepts it is possible to reconstruct an *n*-dimensional system from a one-dimensional observation. Application is simple and consists of the following steps:

A time series of *n* elements it is defined by (*x*1, *x*2,...,*x*n). If for this series is obtained a delay value , and its dimensionality is , then, to recreate the system we generate the array:



the calculation of this parameters are carried out using two additional methods: *Average Mutual Information* and *False Nearest Neighbors Algorithm*.

Mutual Average Information *I*, provides a general measurement of dependency between two dynamic variable sets. Proposed on [4], as an estimated to determine dimension of**s, the delay coordinate for attractor's plotting. Algorithm for calculations of *I* evaluate the normalized histogram of *x*(*t*), its probability *P*(*x*(*t*)), and the normalized histogram of *x*(*t*+**) and its probability *P*(*x*(*t*+**)), as well as the joined normalized histogram of variables *x*(*t*), *x*(*t*+**)and *P*(*x*(*t*),*x*(*t*+**)).

The False Nearest Neighbors [8] algorithm provides the embedding dimension, *m*, of the system's attractor. Embedding dimension is the state space dimension of a system. A two variable system has, for example, an embedding dimension *m*=2. Takens’ Theorem establishes that an attractor unfold completely, with no flow intersections, if embedding dimensions is chosen so it is greater than 2*m*. In [5] an algorithm that gives us an adequate value of *m* is proposed based on the fact that if two points are nearer in a dimension *m*, could be too far in a dimension *m*+1. This happens when the smaller dimension is just a projection of bigger dimensions. Thus, the algorithm establishes firstly, candidate neighbors, increments dimension *m* and inspects then for false neighbors. When no false neighbors are identified, it is possible to say that the attractor is completely unfolded, and *m* is taken as the embedding dimension.

Once established both parameters, it is said that we have parameterized the time series and it is possible to define a forecasting scheme for the system.

# 6 Forecasting scheme

The proposed forecasting scheme can be decomposed in the next steps.

* Time series preprocessing
* Parameter estimation
* Models’ classification
* System’s reconstruction

The first step consists in determining if the time series presents chaotic behavior using Recurrence Plots and, if necessary, conditioning (noise filtering, normalization, etc.). In the next step  y *m* are obtained by means of the mentioned average mutual information and false nearest neighbor algorithms. The third step consists in the classification of several patterns of behavior in the time series through a neural network, for this work, *Self-organizing maps* are employed. Each pattern is associated to a simple linear regressive model, like *least squares method*; therefore we have a set of models or Multi-models. Fourth step is to reconstruct the system.

The use of a neural network makes necessary a stage of training, using data from the series that can act as training data.

Figure 4 shows a schematic diagram of the steps.

Figure4

**Figure 4**. Schematic diagram of the process.

# 7 Case of study

One-step-ahead predictions are very useful in control systems engineering and many schemes have being proposed for that purpose. However, several-steps-ahead predictions are becoming of great interest in many areas. Purchasing electricity in liberalized electrical markets to maximize economic agents’ benefits requires negotiating energy for several day and weeks ahead, for instance [6].

We define two modalities of forecasting. The method proposed in this paper can be used for:

* One-step forecasting
* Iterated forecasting

One-step forecasting means that every new point in the time series is calculated exclusively from a dataset of measured points, a new point can be calculated provided a new measurement is achieved. The dataset is conformed from a fixed number of points and when a new point is acquired the oldest point is discarded.

In iterated forecasting, each point is calculated using, as in the one-step forecasting method, a dataset with a fixed number of points. However, it is possible to calculate multiple points ahead. The dataset will be conformed from a combination of measured and forecasted data, as new forecasted points are calculated.

The Lorenz’s system is employed as first example of test for the scheme of prediction proposed.

Validations of results are carried out using Cross-Validation method:

* Dataset are separated in a training set and a test set.
* Errors between forecasting data and test data are measured with Normalized Mean Squared Error (NMSE) and the square root of the NMSE.

For Lorenz’s system we obtain the results showed in Table 1:

**Table 1**. Scheme parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Value | Variable | Value |
| ** | 10 | *ne* | 10,000 |
| *m* | 3 | *np* | 500 |
| *nc* | 100 | *ni* | 200 |

Where ** is the delay value obtained from the Average Mutual Information algorithm, *m* is the embedding space from the False Nearest Neighbors Algorithm, *nc* is the number of neurons employed by the self-organizing map, *ne* the size of the data set used for training of the network, *np* is the size of set of data forecasted through one step forecasting and *ni* the size of set of data forecasted through the iterated modality.

Achieved errors are presented in Table 2.

**Table 2**. Error estimator of the forecasting in Lorenz System

|  |  |  |
| --- | --- | --- |
| Modality | NMSE | Sqrt(NMSE) |
| One step | 1.2553x10-6 | 0.0011 |
| Iterated | 0.0221 | 0.1487 |

Figures 5 and 6 show the forecasting obtained, as well as the error, point by point, associated with the forecasting.

C:\Users\jguillen\Desktop\Paper\Figure5.emf

**Figure 5**. Plots for the forecasting and its associated error for one step forecasting modality.

The same process was applied to a several other well known chaotic systems, (Chua, Chen, and Rössler), with similar results.

C:\Users\jguillen\Desktop\Paper\Figure6.emf

**Figure 6**. Plots for the forecasting and its associated error for iterated forecasting modality.

# 8 Conclusions

Takens’ theorem has profound and practical consequences, it is a well known result among mathematicians and systems community. Nevertheless, it is not generally known among engineers responsible for monitoring complex systems such as electric networks or circuits. It may contribute greatly to reduce monitoring costs and price services for large infrastructure facilities.

As it has been demonstrated in this work, it is possible to generate a reliable method of forecasting time series with chaotic behavior with a simple and easily implementable scheme. The neural network employed is an excellent classifier and its training requires just a couple of minutes in a personal computer, giving rise to possibilities of implementation in real time.

Given the modularity of the proposed scheme, it is possible to adequate the different parts (neural network, regressive models) to particular needs of forecasting.

# References

1 Lorenz, E.: Deterministic Nonperiodic Flow. Journal of Atmospheric Sciences. 130—141 (1963)

2 Takens, F.: Detecting Strange Attractors in Turbulence. 898 ed., Berlin, Springer. 366—381 (1981)

3 Strogatz, S. H.: Nonlinear Dynamics and Chaos. First ed., Westview Press,(2000)

4 Fraser, A. M. and Swinney, H. L.: Independent Coordinates for Strange Attractors from Mutual Information. Physical Review A. 1134—1140 (1986)

5 Kennel, M. B., Brown, R. and Abarbanel, H. D. I.: Determining Minimum Embedding Dimension using a Geometrical Construction. Phys. Rev. A. 3403—3411 (1992)

6 Simon Hayki: Kalman Filtering and Neural Networks. Wiley Iterscience. (2001)

7 E. N. Sanchez, A. Y. Alanis and J.J. Rico, “Electric Load Demand and Electricity Prices Forecasting Using Higher Order Neural Networks Trained by Kalman Filtering, In artificial Higher Order Neural Networks for Economic and Business. Information Science Reference. (2008)

8 Rainer Hegger and Holger Kantz: False Nearest Neighbors, Improved false nearest neighbor method to detect determinism in time series data. Phys. Rev. E 60, 4970 -- 4973 (1999).