A Genetic Programming System for Time Series Prediction and Its Application to El Niño Forecast

Chapter · May 2006

DOI: 10.1007/3-540-32400-3_12

3 authors:

Ivanoe De Falco
Italian National Research Council

Antonio Della Cioppa
Università degli Studi di Salerno

Ernesto Tarantino
Italian National Research Council

A Genetic Programming System for Time Series Prediction and Its Application to El Niño Forecast

I. De Falco[1], A. Della Cioppa[2], and E. Tarantino[1]

[1] Institute of High Performance Computing and Networking – CNR, Via P. Castellino 111, 80131 Naples, Italy
ivanoe.defalco@na.icar.cnr.it
[2] Dept. of Computer Science and Electrical Engineering, University of Salerno, Via Ponte don Melillo 1, 84084 Fisciano (SA), Italy
adellacioppa@unisa.it

Summary. In this paper a system based on Genetic Programming for forecasting nonlinear time series is outlined. Our system is endowed with two features. Firstly, at any given time t, it performs a τ-steps ahead prediction (i.e. it forecasts the value at time $t+\tau$) based on the set of input values for the n time steps preceding t. Secondly, the system automatically finds among the past n input variables the most useful ones to estimate future values. The effectiveness of our approach is evaluated on El Niño 3.4 time series on the basis of a 12-month-ahead forecast.

Key words: Genetic Programming, time series forecasting

1 Introduction

The prediction of spatio-temporal patterns is substantial to a variety of applications in fields ranging from engineering to economics. Different methods have been investigated over the years [1]. The first approaches were devoted to linear models for which the theory is known and consequently many algorithms for model building are available. The most used linear regression methods have been the autoregressive (AR) and the autoregressive moving average (ARMA) models [2, 3]. An example of more complex regression method is the multivariate adaptive regression splines [4]. These stochastic methods are fast but of limited applicability. In fact, time series produced by systems with nonlinear dynamical behavior have proven notoriously resistant to prediction by conventional techniques.

This is the reason why nonlinear methods have become widely utilized. Among these Artificial Neural Networks (ANNs) represent an attractive approach for time series prediction problems [5, 6, 7, 8]. In fact, in addition to their ability to representing complex nonlinear functions, many different types of ANNs, such as Multi-Layer Perceptron and Radial Basis Function networks, can effectively

I. De Falco et al.: *A Genetic Programming System for Time Series Prediction and Its Application to El Niño Forecast*, Advances in Soft Computing **1**, 151–162 (2005)
www.springerlink.com

construct approximations for unknown functions by learning from examples [9]. In time series prediction with ANNs the main problems are the network structure design and the effectiveness of the learning phase. These limitations have been mitigated exploiting the training adaptability of evolutionary computations [10, 11, 12]. Though these approaches have revealed to be more powerful than stochastic methods, they present drawbacks in the neural model design and in the capability of providing explicit forecasting models.

As an alternative, the flexibility of evolutionary search methods has been exploited. In the last years the Genetic Programming (GP) framework [13] has been applied in various fields like engineering, medicine, economics and so forth with excellent results [10, 14, 15, 16, 17, 18, 19, 20]. The objective is to exploit GP's flexible tree structure for building a time series prediction model. The motivation of this choice is that GP has many advantages which allow to mitigate some of the limitations typical of neural models. These advantages can be summarized as follows: (i) generation of explicit model representations amenable to easy human comprehension, (ii) automatic discovering of the model structure from the given data, (iii) adaptive evolutionary search that allows to escape trapping in suboptimal, unsatisfactory local solutions, (iv) absence of specific knowledge. Moreover, GP is more attractive than traditional Genetic Algorithms for problems that require the construction of explicit models. In fact, with the variable length program tree representation, GP evolves program contents and structures at the same time.

Our GP-based system is able to modelise and forecast time series. In particular, the modeling and the forecasting of El Niño is carried on. Its prediction is of great importance as the global influence of this phenomenon on climate determines strategic planning in areas such as agriculture, management of water resources and reserves of grain and fuel oil before all in the more directly involved countries.

Our system, at any given time t, performs a τ-steps ahead prediction (i.e. it forecasts the value at time $t+\tau$) based on the set of input values for the n time steps preceding t. Furthermore, it automatically finds among the past n input variables the most useful ones to estimate future values.

It is known that a τ-steps ahead prediction can be performed in two different ways. A first approach consists in carrying out prevision recursively, by evaluating the forecast at $t+1$, then using it to forecast $t+2$, and so on, until $t+\tau$ is obtained as a function of these intermediate values. Another approach, instead, consists in directly forecasting the value at $t+\tau$ by using the values known at time t, i.e. $t-1, t-2$ and so on. Our system is based on this latter approach.

The paper is organized as follows. In Sect. 2 our GP-based forecasting system is outlined together with implementation details. In Sect. 3 the description of El Niño time series and the related previous studies are reported. In Sect. 4 the experimental results achieved are shown and discussed. Section 5 contains final comments and prospects of future work.

2 The Genetic Programming System

The aim is the implementation of a genetic system able to automatically provide the modeling of a nonlinear time series and its prediction.

Given a fitness function, the forecasting problem becomes the search of the model which best describes the essential characteristics of the time series. It is evident that an exhaustive search by enumerating all the possible descriptions is computationally impracticable. Hence we appeal to GP which is a powerful and flexible search method inspired by natural selection. It does not guarantee to find the global optimum, nonetheless it usually allows to retrieve a suboptimal solution in a reasonable computation time.

The evolving population is constituted by "programs" representing the potential forecasting models in the form of trees with variable depth. Each model is composed by elementary functions and numerical operators. Moreover, at each time t the prediction function $\tilde{x}(t+\tau)$ for time $t+\tau$ is simply searched as a function of the n past states of $x(t)$ in the n time steps preceding t [22]; this means that:

$$\tilde{x}(t+\tau) = f(x(t-1), \quad x(t-2), \ldots, x(t-n))$$

where n and τ are model parameters to be set.

A population of these candidate models is maintained and gradually improved by constructing new fitter ones until a model of sufficient precision is found or other stopping criteria are satisfied.

To construct the prediction model, data is partitioned into four sets: the processing, the training, the validation and the prediction sets. The processing set contains a number of n time series known elements, the training set contains the values to be approximated during the learning phase, while the validation set is used to evaluate the generalization ability of the found model, and real "blind" forecasting will be performed on the prediction set.

2.1 Encoding

The individuals are composite functions encoded as tree structures for which limits on the tree depth can be specified. The tree nodes are either functions or terminals. The function set consists of elementary functions and numerical operators as reported in Table 1. This set can be easily extended.

The terminal set has a fixed number of elements: the set of n time series past values in input and an Ephemeral Random Constant (ERC). ERC is a special terminal with a defined value. When a terminal ERC is generated a constant value is associated to that terminal. A range of variation $[ERC_{\min}, ERC_{\max}]$ can be specified for this value, depending on the problem at hand. The terminal set is reported in Table 2.

2.2 Genetic Operators

The new elements in the population are generated by means of three operators: *crossover*, *reproduction* and *mutation*:

Table 1. Set of numerical operators and elementary functions with the related arity

Symbol	Arity	Description
$+$	2	Addition
$-$	2	Subtraction
$*$	2	Multiplication
$/$	2	Protected division (returns 1 if the denominator is 0)
sin	1	Sine
cos	1	Cosine
exp	1	Exponential
rlog	1	Protected logarithm (rlog(0) is 0)

Table 2. Set of terminal symbols

Symbol	Arity	Description
x_1	0	input value of time series at time $(t-1)$
		...
x_i	0	input value of time series at time $(t-i)$
		...
x_n	0	input value of time series at time $(t-n)$
R	0	ERC in the range $[ERC_{\min}, ERC_{\max}]$

- **Crossover.** Two parent individuals are selected and a subtree is picked on each one. Then crossover swaps the nodes and their relative subtrees from one parent to the other. This operator must ensure the respect of the depth limits. If a condition is violated the too-deep offspring is simply replaced by one of the parents. There are other parameters that specify the frequency with which functions or terminals are selected as crossover points.
- **Reproduction.** The reproduction operator simply chooses an individual in the current population and copies it without changes into the new population.
- **Mutation.** The mutation operator can be applied to either a function node or a terminal node. A node in the tree is randomly selected. If the chosen node is a terminal it is simply replaced by another terminal. If it is a function and *point mutation* is to be performed, it is replaced by a new function with the same arity. If, instead, *tree mutation* is to be carried out, a new function node (not necessarily with the same arity) is chosen, and the original node together with its relative subtree is substituted by a new randomly generated subtree. A depth ramp is used to set bounds on size when generating the replacement subtree. Obviously it is to check that this replacement does not violate the depth limit. If this happens mutation just reproduces the original tree into the new generation. Further parameters specify the probability with which functions or terminals are selected as mutation points.

2.3 Fitness Function

To evaluate the accuracy of a forecasting model, researchers make normally reference to statistical indices. One of the most used is the Mean Squared Error (*MSE*). In formula this can be devised as follows:

$$MSE = \frac{\sum_{i=1}^{l}(x(i) - \tilde{x}(i))^2}{l} \tag{1}$$

where l is the number of the discrete points in the set under examination.

For our GP tool we have chosen as fitness function for any proposed model the value of MSE computed on the training set (MSE_t). With this choice the problem becomes a minimization task, with the global optimum equal to 0.

It is important to note here that when we perform an evolutionary search for a model, we hope to find a model with good generalization capability, i.e. it can predict unseen values with same quality as values on which training is being carried out.

Usually an arbitrarily good prediction performance can be achieved on training data, yet it often happens that the system specializes itself on those data, and may poorly predict other data of the series not shown (the so-called *overfitting* problem).

To overcome this drawback, we have decided to divide the set of data as mentioned earlier and to follow the approach proposed in [23]. Thus, we evaluate the model on the training set, and we monitor its performance over the validation set. It would be highly desirable to find by evolution models showing an error on the validation set lower than or equal to (or even slightly higher than) that on the training set. Therefore at the end of each generation we take into account its best individual and evaluate the errors on the training set (MSE_t), on the validation set (MSE_v) and on the set composed by points in previous two sets (MSE_0). This latter error is a weighted average of MSE_t and MSE_v. Intuitively, overfitting takes place when MSE_v increases and is higher than MSE_t. Whenever a proposed model shows errors such that $MSE_t \cong MSE_0$, this means that $MSE_t \cong MSE_v$, and we can save it as the current best model. Since it usually happens that MSE_t gets lower and lower as long as evolution takes place, every time we save a new model this will show better quality on validation set than the previous one. At the end of the evolution the lastly saved model will be used to perform forecasting.

2.4 The Genetic Time Series Algorithm

Our Genetic Time Series algorithm (GTS) is based on a freeware *lil-gp* Genetic Programming software [21]. The software has been adapted to our purposes. Once defined the fitness function and the operators, GTS operates as follows:

1. load the time series values;
2. generate at random an initial population of composite functions representing potential forecasting models;
3. evaluate the models by using the fitness;
4. at each generation

- select the functions to undergo the mechanism of reproduction;
- apply the genetic operators *crossover*, *reproduction* and *mutation* to produce new composite functions;
- insert these offspring in the new population;
- evaluate the models by means of fitness;
- evaluate MSE_t, MSE_v and MSE_0 for the best individual and, if is the case, save it as best-so-far model;

5. repeat step 4 until the specified maximum number of generations has been reached.

Even though the individuals in the initial population are randomly created by selecting from the function and terminal sets, during the tree construction some restrictions have been imposed.

Moreover, a so-called half-and-half method has been chosen to create these random initial structures. This is a compromise between the full and the grow methods which are chosen 50% of times each. The full method generates only full trees, that is the tree path length from any terminal node to the root of the tree is the same. The grow, starting from a node chosen as root, recursively calls itself to produce child trees for any node which needs them. When the initial structures reach the maximum allowed depth all further nodes are restricted to be terminals. This method has been used because it allows to create unbalanced trees with a variety of shapes and sizes.

3 El Niño Anomalies Time Series

The Sea Surface Temperature (SST), monitored from ship reports, buoys and satellite imagery, is an important physical quantity in understanding earth's ocean-atmosphere interactions. The so-called El Niño phenomenon is closely connected with and partially defined by changes in SST. El Niño is characterized by the "*Pacific basin-wide increase in both SSTs in the central and/or eastern equatorial Pacific Ocean and in sea level atmospheric pressure in the western Pacific*" which occurs periodically [24]. Indeed, Rasmusson and Carpenter [25] point out that El Niño is the leading factor in interannual variability of SST off the coast of Peru and Ecuador. Predicting this phenomenon is essential, as El Niño is linked to reduction of fish population in coastal waters and changes in the climate of South America and other global regions, often seriously impacting the economy.

Very often, rather than making reference to actual SST series, researchers deal with SST anomalies series. Any such value represents the departure of a given SST from the long term average temperature.

A rigorous definition [26] states that an El Niño occurs in a region when a positive five-month running mean SST anomaly in the investigated region, exceeding 0.4 K, lasts at least six months. Similarly, there are cold events when the SSTs become unusually cold for at least six months: such events are termed La Niña.

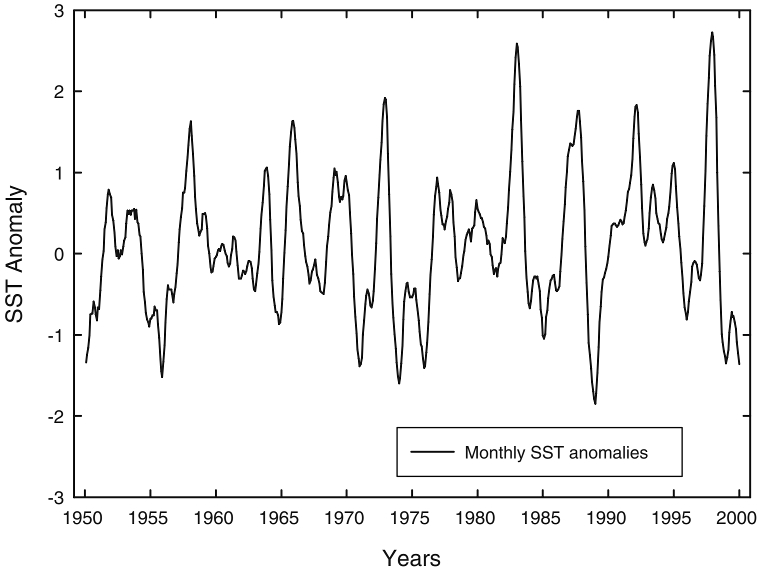

Fig. 1. The El Niño 3.4 SST anomalies original time series

For data sampling of SSTs the equatorial Pacific (between 5°S–5°N) has been divided into a number of regions named El Niño 1, 2, 3, 4, and 3.4 (which encompasses parts of both regions 3 and 4).

In this paper we have decided to take into account the El Niño 3.4 area. This is defined by 120W°–170°W and 5°N–5°S, so it runs along the Equator and is broadly bounded to west by Kiritimati atoll (Kiribati) and to east by Galapagos Islands (Ecuador). For this area literature provides researchers with the monthly series of SST anomalies for the period January 1950–December 1999. The series is publicly available, downloadable at [28], contains 600 values and is reported in Fig. 1.

Forecast of this series has been faced by many researchers with different approaches.

Classical multivariate statistical methods, like for example Multiple Linear Regression (MLR), Principal Component Analysis (PCA), Canonical Correlation Analysis (CCA) and many others, are widely used for data analysis in meteorology and oceanography [27].

Several dynamical models have been applied to predict this series. They are too numerous to be described here. There is an excellent review on them in [29]. Among them we can cite here at least Lamont Simple Coupled Model (LDEO), Australian Bureau of Meteorology Research Center (BMRC) low-order coupled model, University of Oxford intermediate coupled model, Scripps/MPI Hybrid Coupled Model (HMC–3), The Center for Ocean Land Atmosphere Studies (COLA) comprehensive coupled model, NCEP comprehensive coupled model.

Unger and others [30] have approached the problem by combining the forecasts of three input models into a single forecast based on the past behavior of each contributing model. One of the selected models is dynamical (CMP12 NCEP) while the other two are statistical (Constructed Analogue model and Canonical Correlation Analysis model).

Recent advances in Artificial Neural Network (ANN) modeling have led to the nonlinear generalization of CCA. Non-Linear Canonical Correlation Analysis (NLCCA) using an NN approach was introduced by Hsieh and applied to the tropical Pacific SST fields [31].

Tang and others [32] have used a Feedforward Neural Network model. An ensemble of 20 neural networks are trained, each with a different weight initialization. The final model output is the average of the outputs from the 20 members of the ensemble. The advantage of the ensemble model is to reduce variance, or instability of the neural network. By doing so, they have been able to perform several forecasts with respectively 3, 6, 9 and 12 months steps ahead.

As far as we know there is no paper in literature dealing with Evolutionary Algorithms applied to El Niño 3.4 forecast.

4 The Experimental Results

4.1 Parameter Setup and Performance Measures

The El Niño 3.4 SST anomalies series has been divided as it follows: due to the above choice, the former 100 points are used as processing set. Points in the range [101–400] are used to perform training, those in the range [401–588] are used as validation. Finally, the points [589–600] are used to perform prediction. Firstly the value of n has been set equal to 100 and the value of τ has been set equal to 12, so that our system performs 12-month-ahead forecasting. Due to this choice, the first value forecast by our system is that at time $t = 100 + 12 = 112$.

A preliminary set of runs has been carried out to find a good parameter set. The best resulting values are listed in the following. The population size chosen is $3,000$, the maximum number of generations allowed is 350, crossover rate is 0.40, mutation rate is 0.59 and reproduction rate is 0.01. Initial tree depth is in the range [2–6], and maximum tree depth is 30. Initial population is generated by half-and-half method. Tournament selection with size equal to 200 is carried out. Crossover is internal with probability 90%, and external for the remaining 10%. Mutation is equiprobabilistically internal or external. Depth of new subtrees generated by mutation is in the range [0–10].

4.2 Findings

Experiments have been performed on a Sun workstation and require few hours. We have carried out a set of ten runs with the best found parameter values. We consider as the best among those trials the one that gives rise to the best model in terms of lower MSE_v. Results for this best run are reported in Fig. 2 where the training and the validation are shown. It should be noted that values shown have been

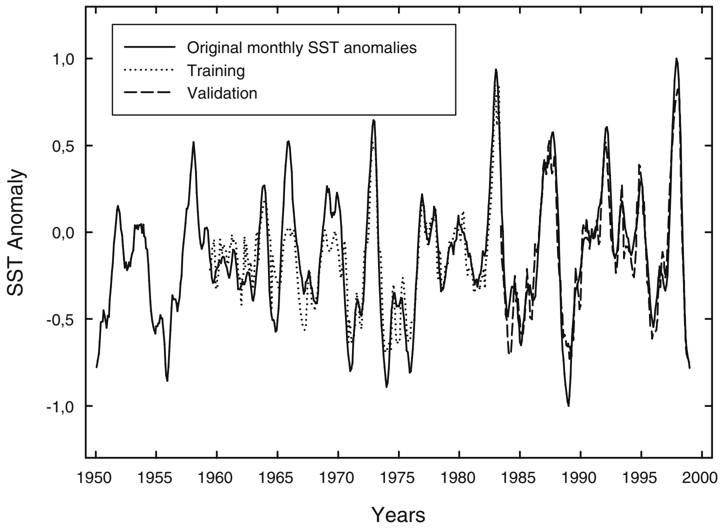

Fig. 2. Training and validation

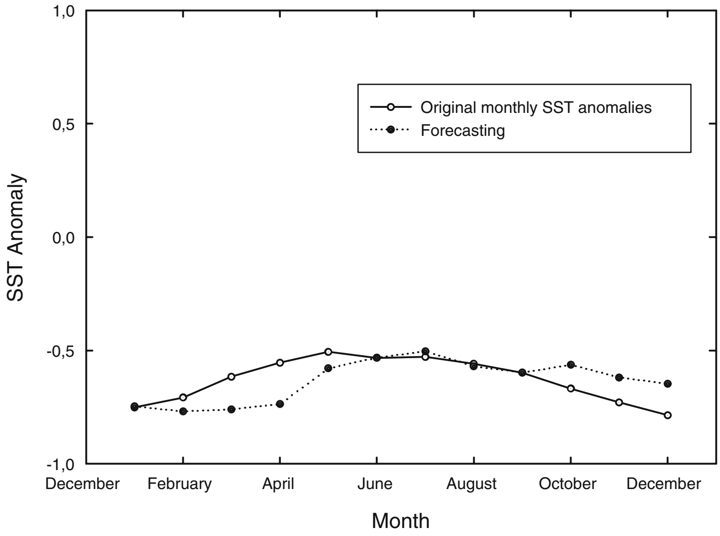

Fig. 3. Forecasting for the twelve months of year 1999

normalized in $[-1.0, 1.0]$. It can be seen that our system models the phenomenon in good accordance with historical data, both hot and cold phases are caught correctly, and all peaks are forecast, though in some cases computed peak heights are lower than real ones. Figure 3 reports the forecasting for the twelve months making up year 1999. Also in this case values are normalized in $[-1.0, 1.0]$. As it can be easily seen,

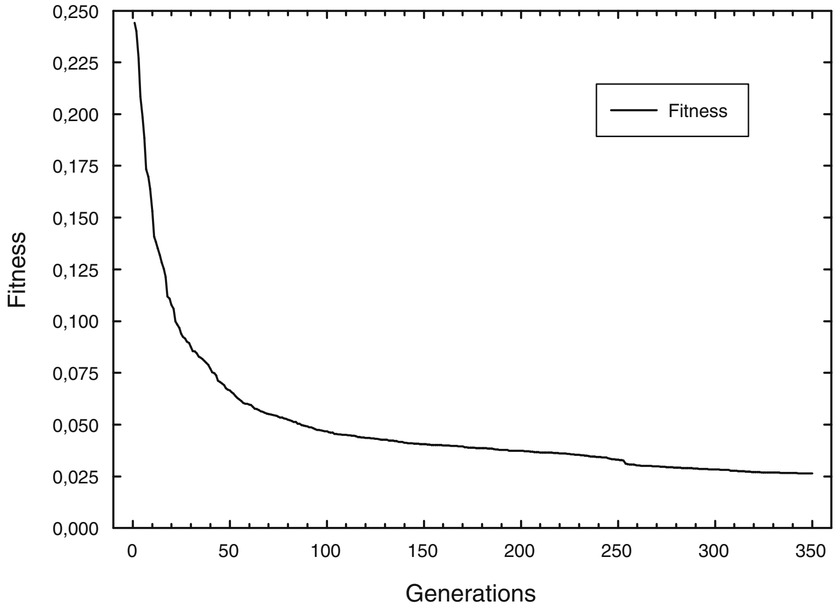

Fig. 4. Evolution of best fitness as a function of number of generations

our forecasting is quite close to actual data: the general shape of this twelve-month time interval is reproduced with good approximation, and can be interpreted as a La Niña phase. Forecast values show a first increasing trend (though initially slower than the real series) followed by a decreasing phase. Values for months from June to September are in excellent agreement with real data. Figure 4 shows the evolution of the best fitness value as a function of the number of generations. The improvement in solution quality is very high during a first phase lasting about 50 generations, though decrease continues satisfactorily in the next phases as well. The values of the errors related to the best model found in this evolution are $MSE_t = 0.026442$, $MSE_v = 0.016818$ and $MSE_p = 0.008829$ for the prediction set.

5 Conclusions and Future Works

In this paper we have presented a GP tool for time series forecasting based on a τ-steps ahead approach. The system has been tested on a publicly available series, that of El Niño 3.4 SST anomalies. Experimental results have demonstrated the effectiveness of the proposed approach in providing good-quality resemblance of the computed series to the real one. Furthermore, good forecasting has been obtained, which guesses the next phase as a La Niña one. This has been confirmed by the real data.

Future work will include the application of the proposed system to other real-world time series in order to further validate the promising results reported in the present paper. Furthermore, a parallel implementation of the GP tool on Multiple Instruction Multiple Data (MIMD) parallel machines will be carried out with the aim to reduce computational times.

References

1. Weigend A, Gershenfeld N (eds) (1994). Time series prediction: forecasting the future and understanding the past, Addison-Wesley, Reading, Massachussetts.
2. Tong H (1990). Nonlinear time series: a dynamical system approach. Oxford University Press.
3. Box G, Jenkins G, Reinsel G (1994). Time series analysis: forecasting and control. Prentice Hall, Englewood Cliffs, New Jersey.
4. Friedman J (1991). Multivariate adaptive regression splines. Annals of Statistics 19:1–142.
5. Mozer N (1993). Neural net for temporal sequence processing. In: Weigend A, Gershenfeld N (eds) Time series prediction: forecasting the future and understanding the past. Addison–Wesley, Reading, Massachussetts.
6. Tsoi A, Back A (1994). Locally recurrent globally feedforward networks: a critical review of architectures. IEEE Trans. on Neural Networks 2:229–239.
7. Koskela T, Lehtokangas M, Saarinen J, Kaski K (1996). Time series prediction with multilayer perceptron, FIR and Elman neural networks. In: World Congress on Neural Networks, INNS Press, 491–496.
8. De Falco I, Iazzetta A, Luongo G, Mazzarella A, Tarantino E (2000). The seismicity in the southern tyrrhenian area and its neural forecasting. Pure and Applied Geophysics 157:343–355.
9. Cybenko G (1989). Approximations by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2:303–314.
10. Angeline P J (1998). Evolving predictors for chaotic time series. In: Rogers S, Fogel D, Bezdek J, Bosacchi B (eds) Proceedings of SPIE: Application and Science of Computational Intelligence, Vol. 3390. Bellingham, WA, USA, 170–180.
11. De Falco I, Della Cioppa A, Iazzetta A, Natale P, Tarantino E (1999). Optimizing neural networks for time series prediction. In: Advances in Soft Computing in Engineering Design and Manufacturing. Springer–Verlag, London.
12. Lee Ki–Youl, Lee Dong–Wook, Sim Kwee–Bo (2000). Evolutionary neural networks for time series prediction based on L–system and DNA coding method. In: Congress on Evolutionary Computation, La Jolla, California, USA, July 16–19. IEEE Press, 1467–1474.
13. Koza J (1992). Genetic programming. On the programming of computers by means of natural selection. MIT Press, Cambridge, MA, USA.
14. Iba H (1993) System identification using structured genetic algorithms. In: Forrest S. (Ed.) Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California. Morgan Kaufmann, 279–286.
15. Oakley H (1994) Two scientific applications of genetic programming: stack filters and nonlinear equation fitting to chaotic data. In: Kinnear K (ed.) Advances in Genetic Programming. MIT Press, Cambridge, MA, USA, 369–389.
16. Mulloy B S, Riolo R L, Savit R S (1996). Dynamics of genetic programming and chaotic time series Prediction. In: Koza J R, Goldberg D E, Fogel D B, Riolo R L (eds), Genetic Programming 1996: Proceedings of First Annual Conference, Stanford University, CA, USA, July 28–31, 166–174.

17. Numata M, Yoshihara I, Yoshizawa M, Abe K (1999). GP-based heart rate prediction for artificial heart control. In: Brave S. and Wu A. S. (Eds) Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference, Orlando, Florida, USA, July 13–17, 193–197.
18. Iba H, Nikolaev N I (2000). Genetic programming polynomial models of financial data series. In: Congress on Evolutionary Computation, La Jolla, California, USA, July 16–19. IEEE Press, 1459–1466.
19. Nikolaev N I, Iba H (2001) Regularization approach to inductive genetic programming. IEEE Trans. on Evolutionary Computation, 4:359–375.
20. Duan M, Povinelli R J (2001). Estimating stock price predictability using genetic programming. In: Spector L, Goodman E D, Wu A, Langdon W B, Voigt H M, Gen M, Sen S, Dorigo M, Pezeshk S, Garzon M H, Burke E (eds) Genetic and Evolutionary Computation Conference (GECCO-2001), S. Francisco, California, USA, July 7–11. Morgan Kaufmann, 193–197.
21. Zongker D, Punch W (1995). lilgp version 1.0, Lansing, Michigan State University, GA Research and Application Group. Available at: http://isl.cps.msu.edu/GA/software/lil–gp.
22. Takens F (1981). Detecting strange attractors in turbulence. In: Lecture Notes in Mathematics. Springer–Verlag, Berlin.
23. Jäske H (1996). Prediction of sunspots by genetic programming and their applications. In: Alander J T (ed), Proceedings of Second Nordic Workshp on Genetic Algorithms, Vaasa, Finland, 19–23 August, 79–88.
24. Trenberth K E, (1996) El Niño definition. CLIVAR – Exchanges, 3:6-8.
25. Rasmusson E M, Carpenter T H (1982) Variations in tropical sea surface temperature and surface wind fields. Associated with the Southern Oscillation/El Niño, Monthly Weather Review, Vol. 110, 354-384.
26. Trenberth K E (1997), The definition of El Niño. Bullettin of the American Meteorological Soc., Vol. 78, 2771–2777.
27. von Storch H, Zwiers F W (1999) Statistical Analysis in Climate Research. Cambridge University Press, 484.
28. El Niño 3.4 SST index series.
Available at: http://www.cgd.ucar.edu/cas/catalog/climind/Nino_3_3.4_indices.html.
29. Barnston A G, Glantz M H, He Y (1998) Predictive skill of statistical and dynamical climate models in SST forecasts during the 1997-98 El Niño episode and the 1998 La Niña onset. Bulletin of the American Meteorological Soc., 217–243.
30. Unger D, Barnston A, van den Dool H, Kousky V (1996) Consolidated forecasts of tropical pacific SST in Niño 3.4 using one dynamical model and two statistical models. Experimental Long-Lead Bulletin, Vol. 5:1 of the Hardcopy Version.
31. Hsieh W (2000) Nonlinear canonical correlation analysis of the tropical pacific climate variability using a neural network approach. Journal of Climate, 14:2528-2539.
32. Tang B, Hsieh W W, Tangang F T (1998) Neural network model forecasts of the Niño 3.4 sea surface temperature. Experimental Long-Lead Forecast Bulletin, Vol. 7:1.