Comparison of Artificial Intelligence Methods for Predicting the Time Series Problem

S.SP. PAPPAS
University of the Aegean
Information & Communications Systems Engineering
26 Deligeorgi Str.
17456 Athens

L. EKONOMOU
Public Power Corporation S.A.
22 Chalcocondyli Str.
104 32 Athens

GREECE

Abstract: This paper studies the time series prediction problem. Artificial intelligence methods are applied to two different time series in order to compare their effectiveness and their producing results. The applied methods are based on the Group Method of Data Handling (GMDH) algorithms and the hybrid method of GMDH and Genetic Algorithms, i.e. Genetics-Based Self-Organising Network (GBSON). Finally useful conclusions and the advantages and disadvantages of each method are stated.

Keywords: Time series prediction, Group Method of Data Handling (GMDH) algorithms, Genetics-Based Self-Organising Network (GBSON) method, Artificial neural networks, Genetic algorithms, Evolutionary algorithms

1. Introduction

The prediction of time series signals is based on their past values. Therefore, it is necessary to obtain a data record. When obtaining a data record, the objective is to have data that are maximally informative and an adequate number of records for prediction purposes [1, 2]. Hence, future values of a time series $x(t)$ can be predicted as a function of past values $x(t\text{-}1), x(t\text{-}2), \ldots, x(t\text{-}\varphi)$.

$$x(t+\tau) = f(x(t\text{-}1), x(t\text{-}2), \ldots, x(t\text{-}\varphi))$$

The problem of time series prediction now becomes a problem of system identification. The unknown system to be identified is the function $f(\cdot)$ with inputs the past values of the time series.

The search for the most suitable model for a system is guided by an assessment criterion of the goodness of a model. In the prediction of time series, the assessment of the goodness of a model is based upon the prediction error of the specific model. When the most suitable model of a system has been determined then it has to be validated. The validation step in the system identification procedure is very important because the most suitable model obtained was chosen among the predefined candidate models set. This step will certify that the model obtained describes the true system. Usually, a different set of data than the one used during the identification of the model, the *validation set*, is used during this step [3, 4].

In this paper the time series prediction problem is studied. Two different methods are applied to two different time series in order to compare their effectiveness, their producing results and to derive useful conclusions.

2. *Group Method of Data Handling (GMDH)*

The Group Method Data Handling (GMDH) [5] is a self-organising method that was initially proposed by Ivakhnenko to produce mathematical models of complex systems by handling data samples of observations. It is based on the sorting-out procedure, i.e. consequent testing of increasingly complex models, chosen from a set of models-candidates, in accordance with a given external criterion on a separate part of data samples. Thus, GMDH algorithms solve the argument:

$$\tilde{g} = \arg\min_{g \subset G} CR(g)$$

where G is the set of candidate models and $CR(g)$ is an external criterion of model's g quality.

Most GMDH algorithms use polynomial reference functions to form the set of candidate models. The Kolmogorov-Gabor theorem shows that any function $y = f(\vec{x})$ can be represented as:

$$y = \alpha_0 + \sum_i \alpha_i x_i + \sum_i \sum_j \alpha_{ij} x_i x_j + \sum_i \sum_j \sum_k \alpha_{ijk} x_i x_j x_k + \ldots$$

where x_i is the independent variable in the input variable vector $\vec{x}$ and $\vec{\alpha}$ is the coefficient vector. Other non-linear reference functions such as difference, logistic and harmonic can also be used. GMDH algorithms are used to determine the coefficients and terms of the reference functions used to partially describe a system. GMDH algorithms are multi-layered, and at each layer the partial description is simple and it is conveyed to the next layers to gradually obtain the final model of a complex system.

It has been proven that GMDH algorithms converge and that a non-physical model obtained by GMDH is better than a full physical model on error criterion [6]. A special feature of the GMDH algorithms is that the model to be selected is evaluated on a new data set, different from the one used to estimate its parameters.

2.1 Combinatorial GMDH algorithm (COMBI)

This is the simplest GMDH algorithm. First *n* observations of regression-type data are taken. These observations are divided into two sets: the training set and the validating set.

The COMBI algorithm is multi-layered; at each layer, it obtains a candidate model of the system and once the models of each layer are obtained, the best one is chosen to be the output model.

The first layer model is obtained by using the information contained in every column of the training sample of observations. The candidate models for the first layer have the form:

$$y = a_0 + a_1 x_i,\ i = 1, 2, \ldots, m$$

To obtain the values of the coefficients *a*0 and *a*1 for each of the *m* models, a system of Gauss normal equations is solved. In the case of the first layer, the system of Gauss normal equation for the *i*th model will be:

$$\begin{bmatrix} nt & \sum_{k=1}^{nt} x_{ki} \\ \sum_{k=1}^{nt} x_{ki} & \sum_{k=1}^{nt} x_{ki}^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{nt} y_k \\ \sum_{k=1}^{nt} x_{ki} y_k \end{bmatrix}$$

where *nt* is the number of observations in the training set.

After all possible models from this layer have been formed, the one with the minimum *regularity criterion AR*(*s*) [7] is chosen. The regularity criterion is defined by the formula:

$$AR(s) = \frac{1}{nv} \sum_{i=nt+1}^{n} \left(y_i - \hat{y}_i\right)^2$$

where *nv* is the number of observations in the validation set, *n* is the total number of observations, *y*ˆis the estimated output value and *s* is the model whose fitness is evaluated.

A small number of variables that give the best results in the first layer, are allowed to form second layer candidate models of the form:

$$y = a_0 + a_1 x_i + a_2 x_j,\ i, j = 1, 2, \ldots, m$$

Models of the second layer are evaluated for compliance with the criterion, and again the variables that give best results will proceed to form third layer candidate models. This procedure is carried out as long as the criterion decreases in value, and candidate models at the m^{th} layer will have the form

$$y = a_0 + a_1 x_i + a_2 x_j + \ldots + a_m x_l$$

$$i, j, l = 1, 2, \ldots, m$$

After the best models of each layer have been selected, the output model is selected by the *discriminating criterion* termed as δ^2. A possible discriminating criterion is the *variation criterion RR*(*s*) defined by [8]:

$$RR(s) = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

where *y* is the mean output value and *s* is the model whose fitness is evaluated.

The model with the minimum value of the variation criterion *RR*(*s*) is selected as the output model. Other discriminating criteria can be used that make a compromise between the accuracy and complexity of a model.

2.2 Genetics-Based Self-Organising Network (GBSON)

The method introduced by Kargupta and Smith [9], i.e. the Genetics-Based Self-Organising Network (GBSON), is a hybrid method of the GMDH and Genetic Algorithms [6]. The GBSON method was introduced to overcome the drawbacks of the original GMDH algorithms, since they use local search techniques to obtain an optimal solution [10, 11].

The GBSON uses polynomial neural networks to represent the model of the system to be identified. Each layer of the polynomial neural network is regarded as a separate optimisation problem. The input to the first layer of the network is the independent variables of the data sample. The output of each layer is the peak nodes obtained by the use of a multi-modal Genetic Algorithm

[12]. The peak nodes selected to be the output of a layer are also the inputs for the next layer.

The population members of the GA are network nodes represented by an eightfield bit string. The two first fields are used to represent the nodes from the previous layer connected to the present node. The other six fields are used to represent the coefficients of a quadratic function that determines the output of the node y:

$$y = a + bz_1 + cz_2 + dz_1z_2 + ez_1^2 + fz_2^2$$

where z_1 and z_2 are the outputs of the connected nodes in the previous layer.

The fitness measure of a node is given by calculating its description length. The description length gives a trade off between the accuracy of the prediction and the complexity of the network. The equation used by Kargupta and Smith for calculating the description length is:

$$I = 0.5n \log D_n^2 + 0.5m \log n$$

where D_n^2 is the mean-square error, m is the number of coefficients in the model selected and n is the number of observations used to determine the mean-square error.

The multi-modal GA used in GBSON incorporates the fitness-sharing scheme, where the shared fitness is given by:

$$f_i' = \frac{f_i}{m_i}$$

f_i is the original fitness of the node and m_i is the niche count defined by:

$$m_i = \sum_{j=1}^{N} sh(d_{ij})$$

where

$$sh(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_s}\right)^a & if\ d_{ij} < \sigma_s \\ 0 & otherwise \end{cases}$$

N is the population size and d_{ij} is the Hamming distance between the members of the population i and j. The niche radius σ_s is determined by the equation:

$$\frac{1}{2^l} \sum_{i=0}^{\sigma_s} \binom{l}{i} = \frac{l}{q}$$

where l is the string length and q is the number of nodes in the previous network layer.

New populations are obtained after applying the genetic operators of tournament selection, single-point crossover and point mutation. A mating restriction is also applied to the members to be crossed. If a member i is to be crossed, its mate j is selected such that $d_{ij} < \sigma_s$. If no such mate can be found then j is selected randomly.

The GBSON procedure continues until the GA converges to a layer with a single node.

3. Case Studies

3.1 Thunderstorm Days Series

The first set of experiments was conducted on monthly thunderstorm days numbers, recorded by the National Meteorological Authority of Hellas [13], from January 1980 to December 2005. These numbers are indicative of the average relative number of thunderstorm days observed every month of the year.

The thunderstorm days are strongly related to the lightning. In result, the thunderstorm days can determine the lightning level of an area, i.e. the number of lightning flashes to earth. The prediction of the thunderstorm days is therefore essential to the studies of transmission and distribution lines' designers, since the knowledge of the future lightning level of an area can result in a better design and consequently to the reduction of the lightning faults in lines.

The thunderstorm days time series has been classified as quasiperiodic, and it has been found that the period varies between 8 to 12 years with irregular amplitudes, making the time series hard to predict.

3.2 Lorentz Attractor Series

Edward Lorentz obtained the Lorentz attractor system, in his attempt to model how an air current rises and falls while it is heated by the sun. The Lorentz attractor system is defined by the following three ordinary differential equations.

$$\frac{dx(t)}{dt} = \sigma\, x(t) - \sigma\, y(t)$$

$$\frac{dy(t)}{dt} = -y(t) + r\, x(t) - x(t)\, y(t)$$

$$\frac{dz(t)}{dt} = -bz(t) + x(t)\, y(t)$$

The Lorentz attractor system has also been used to model a far-infrared NH_3 laser that generates chaotic intensity fluctuations [14]. The far-infrared NH_3 laser is described by exactly the same equations, only the variables and constants have different physical meaning [15-18].

The time series used in this experiment, is the *x-component* in the Lorentz equations. The data were generated by solving the system of differential equations, that describe the Lorenz

attractor, with the initial conditions of σ = 10, r = 50 and b = 8/3. The data were again normalised to take values from zero to one, before they were used as inputs to the polynomial neural networks.

4. Simulation Results

The two different methods, i.e. GMDH COMBI and GBSON, have been applied to both thunderstorm days time series and Lorenz attractor time series. To allow better comparison of the results obtained using the GBSON and COMBI algorithms, the same number of data was used for training and validation.

Therefore, in the thunderstorm days time series the first 208 points are used for training the following 52 points are used for validation and the last 52 points are used for testing the model, while in the Lorenz attractor time series, the first 2000 points are used for training, the following 500 points are used for validation and the last 500 points are used for testing the model obtained in data that have not been used in any part of the modelling process. The model's fitness is based on the percent square error as in the GBSON method.

Thunderstorm Days Series

The input pattern was assigned as ($x(t$-1), $x(t$-2), $x(t$-3)) and thus the output pattern is:

$$x(t) = f((x(t-1), x(t-2), x(t-3))$$

as in the GBSON method.

The algorithm resulted to a network with two layers. The percent square error (PSE) and the root mean square error (RMSE) over the whole data set are 0.059128 and 0.007860, respectively. The PSE and RMSE for each of the data sets for the COMBI and GBSON algorithms, are summarised in Table 1. The actual time series as well as the output generated by the network constructed by the COMBI algorithm is shown in Figure 1. The actual error for each point in the data set is shown in Figure 2.

The network obtained with the COMBI algorithm is less complex than the one obtained with the GBSON method. Nevertheless, the results obtained with the GBSON method are better for all the data points from the results obtained with the COMBI. In addition, the prediction over the new data set is approximately 50% better. The only data set that the COMBI predicted with a smaller error is the training set. This set though, is a new data set for the GBSON method, since the parameters are determined with a GA, and there is no training set for the GBSON. The GBSON method used only the points in the validation set to determine the fitness of solutions obtained by the GA. As a result, the GBSON algorithm generalises better than the COMBI algorithm.

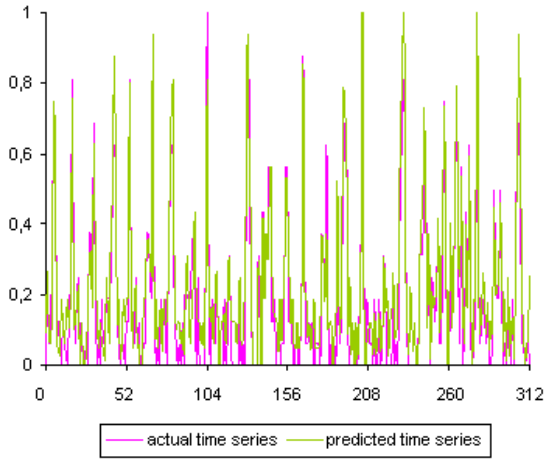

Figure 1: The actual and predicted thunderstorm days time series with COMBI.

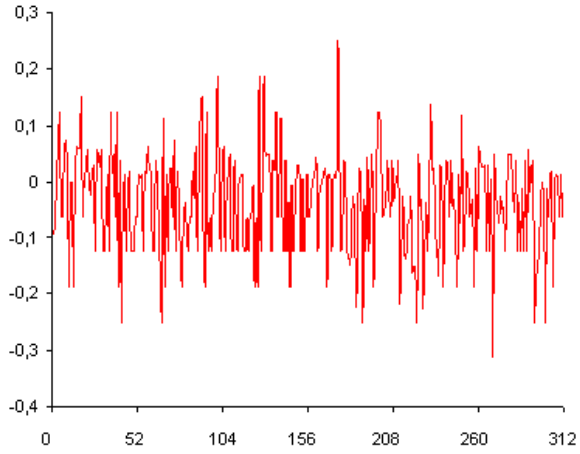

Figure 2: The actual error for each point of the thunderstorm days series predicted with COMBI.

Lorenz Attractor Series

The input pattern was assigned as ($x(t$-1), $x(t$-2), $x(t$-3), $x(t$-4)) and thus the output pattern is:

$$x(t) = f((x(t-1), x(t-2), x(t-3), x(t-4))$$

as in the GBSON method.

The COMBI algorithm converged to a network with two layers. This network predicted the Lorentz attractor system with a PSE over the whole data set of 0.006652. The RMSE for the whole data set again was 0.001377. The actual system and its prediction are shown in Figure 3. The actual error of the prediction can be seen in Figure 4.

Table 1: Comparison of the results for the Thunderstorm Days Series

	PSE whole set	PSE training set	PSE validation set	PSE new data set	RMSE whole set
COMBI	0.059128	0.031255	0.074548	0.059301	0.007860
GBSON	0.043825	0.043211	0.067253	0.029435	0.007243

Table 2: Comparison of the results for the Lorenz Attractor Series

	PSE whole set	PSE training set	PSE validation set	PSE new data set	RMSE whole set
COMBI	0.006652	0.006535	0.007448	0.006471	0.001377
GBSON	0.000244	0.000255	0.000231	0.000207	0.000050

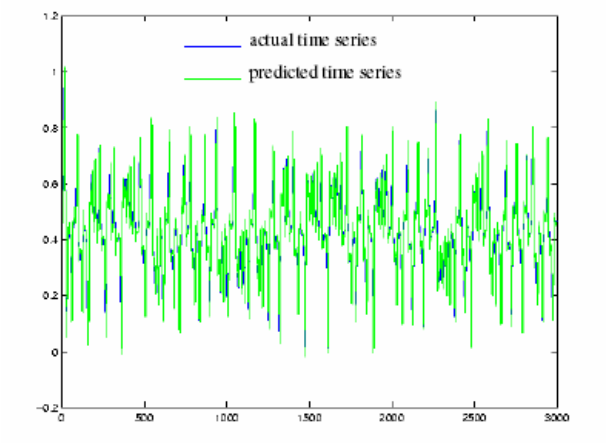

Figure 3: The actual and predicted Lorentz attractor system time series with COMBI.

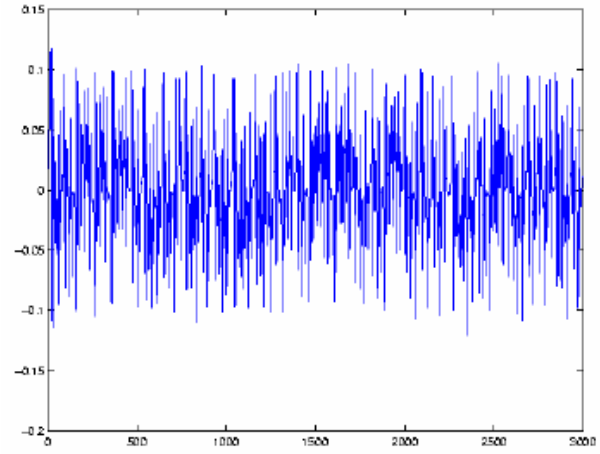

Figure 4: The actual error for each point of the Lorentz attractor predicted with COMBI.

The comparison between the prediction achieved with the COMBI and GBSON algorithms is summarised in Table 2. The complexity of the model obtained with the GBSON method has increased considerably, it has six more layers in the network, but the results obtained are approximately 95% better for all data sets compared to the ones obtained with COMBI.

5. Conclusions

The paper has presented the use of artificial intelligence and more specifically artificial neural networks, genetic algorithms and evolutionary algorithms in the solution of the time series prediction problem. The time series prediction problem has been formulated as a system identification problem, where the input to the system was the past values of a time series, and its desired output is the future values of a time series. Two different methods (GMDH COMBI and GBSON) have been applied to two different sets of significant time series data. Their producing results have been compared deriving useful conclusions on their effectiveness.

References:

[1] T. Izumi, Y. Iiguni, "Data compression of nonlinear time series using a hybrid linear/nonlinear predictor", Signal Processing, vol. 86, no. 9, 2006, pp. 2439-2446.

[2] Z. Lu, "A regularized minimum cross-entropy algorithm on mixtures of experts for time series prediction and curve detection", Pattern Recognition Letters, vol. 27, no. 9, 2006, pp. 947-955.

[3] J.M. Matías, W. González-Manteiga, J. Taboada, C. Ordóñez, "Managing distribution changes in time series prediction", Journal of Computational and Applied Mathematics, vol. 191, no. 2, 2006, pp. 206-215.

[4] A. Yadav, D. Mishra, R.N. Yadav, S. Ray, P.K. Kalra, "Time-series prediction with single integrate-and-fire neuron", Applied Soft Computing, In Press, Corrected Proof, available online 18 April 2006.

[5] S.J. Farlow, "The GMDH algorithm, Self-Organizing Methods in Modelling", pp. 1-24, 1984.
[6] http://www.inf.kiev.ua/GMDH-home/GMDH_res.htm
[7] V.S. Stepashko, "Asymptotic properties of external criteria for model selection", Soviet Journal of Automation and Information Sciences, vol. 21, no. 6, 1988, pp. 24-32.
[8] V.P. Belogurov, "A criterion of model suitability for forecasting quantitative processes", Soviet Journal of Automation and Information Sciences, vol. 23, no. 3, 1990, pp. 21-25.
[9] H. Kargupta, R.E. Smith, "System identification with evolving polynomial networks", Proceedings of the 4th International Conference on Genetic Algorithms, 1991, pp. 370-376.
[10] X. Yao, Y. Liu, "Making use of population information in evolutionary artificial neural networks", IEEE Trans on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 28, no. 3, 1998, pp. 417-425.
[11] D. Quagliarella, J. Periaux, C. Poloni, G. Winter, "Generic algorithms and evolution strategies in engineering and computer science", Wiley, 1998.
[12] O.V. Pictet, M.M. Dacorogna, R.D. Dave, B. Chopard, R. Schirru, M. Tomassini, "Genetic algorithms with collective sharing for robust optimization in financial applications", Neural Network World, vol. 5, no. 4, 1995, pp. 573-587.
[13] Data supplied from the National Meteorological Authority of Hellas, 2005.
[14] H. Kantz, T. Schreiber, "Nonlinear time series analysis", Cambridge University Press, 1997.
[15] N.E. Mastorakis, "Solving differential equations via genetic algorithms", Proc. of the Circuits, Systems and Computers '96 (CSC'96), Piraeus, Greece, July 15-17, 1996, 3rd Volume: Appendix, pp.733-737.
[16] N.E. Mastorakis, "Genetic Algorithms and Nelder-Mead Method for the solution of boundary value problems with the collocation method", 5th WSEAS Int. Conf. on Simulation, Modeling & Optimization, Corfu, Greece, August 17-19, 2005 (pp690-694).
[17] N.E. Mastorakis, "Unstable ordinary differential equations: solution via genetic algorithms and the method of Nelder-Mead", 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation (ISTASC'06), Crete, Greece, August 18-20, 2006.
[18] N.E. Mastorakis, "The singular value decomposition (SVD) in tensors (multidimensional arrays) as an optimization problem. solution via genetic algorithms and method of Nelder-Mead", 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation (ISTASC'06), Crete, Greece, August 18-20, 2006.

Vitae:

S.Sp. Pappas received a Bachelor of Engineering (Hons) in Electrical and Electronic Engineering (1997) and a Master of Science in Advanced Control (1998) from University of Manchester Institute of Science and Technology (U.M.I.S.T). He is currently a PhD student at the Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos, Greece. His research interest is in the area of Partitioning Theory & their applications, Evolutionary Algorithms and Modern Control Methods. He is an I.E.E.E member.

Lambros Ekonomou was born on January 9, 1976 in Athens, Greece. He received a Bachelor of Engineering (Hons) in Electrical Engineering and Electronics in 1997 and a Master of Science in Advanced Control in 1998 from University of Manchester Institute of Science and Technology (U.M.I.S.T.) in United Kingdom. In 2006 he received a Ph.D. from the National Technical University of Athens (N.T.U.A.) in Greece. Currently he is working in the Hellenic Public Power Corporation S.A. as an electrical engineer and he is also a research assistant at the High Voltage Laboratory of N.T.U.A. His research interests concern high voltage transmission lines, lightning performance, lightning protection stability analysis, control design and artificial neural networks.