E-tsRBF: preliminary results on the simultaneous determination of time-lags and parameters of Radial Basis Function Neural Networks for time series forecasting

E. Parras-Gutierrez, V. Rivas, and M.J. del Jesus
Department of Computer Sciences
Campus Las Lagunillas s/n
23071, Jaen, Spain
eparrasg,vrivas@vrivas.es, mjjesus@ujaen.es

***Abstract*—Radial basis function neural networks have been successfully applied to time series prediction in literature. Frequently, methods to build and train these networks must be given the past periods or lags to be used in order to create patterns and forecast any time series. This paper introduces E-tsRBF, a meta-evolutionary algorithm that evolves both the neural networks and the set of lags needed to forecast time series at the same time. Up to twenty-one time series are evaluated in this work, showing the behavior of the new method.**

***Index Terms*—Neural Network; evolutionary algorithms; time series;**

I. INTRODUCTION

Time series are present in many activities in a lot of fields like engineering, biological, economic or social, since they represent a chronological sequence of observed data. Time series forecasting is the power to predict future values based on past and present values through the time line in order to achieve the information of the underlying model.

Diverse technologies have been arising, fundamentally from the Mathematics and the Statistics fields, to shape, explain, and predict the values of the time series. This way, in a generic form, it can be distinguished three kind of techniques: a) the descriptive traditional technologies, which are those previous to the appearance of the Autoregressive Integrated Moving Average (ARIMA, [1]) models; b) the ARIMA models themselves, which supposed a qualitative jump in the study of this kind of data, being widely used at present; and c) the technologies arisen inside the area of data mining, which have experienced a notable interest throughout last decade [9].

On the other hand, independently of the model carried out, one of the main problems that emerge working with time series is the choice of the time periods (or lags) that must be used in order to forecast future values. In this way, the own selection of the input variables for the model to build turns itself into a problem that can be faced using data mining.

One of the most widely used methods in time series forecasting is the model described by Box and Jenkins [1], the univariant Box-Jenkins method combines moving average and autoregressive models into one unified approach. This approach is both simple and yields accurate results which explains its wide use. However, when conditions are not favorable the Box and Jenkins' methodology can lead to manage forecasts out of reasonable limits, becoming indispensable new previous transformations that guarantee the conditions to perform the estimation of the model and the prediction of future values. In this cases Artificial Neural Network are an alternative more precise in order to predict the future behavior of the time series.

Artificial Neural Networks (ANN) emphasize for their capacity of learning from the information provided [5]. This makes obvious the need to do suppositions over the models and relations of the time series [20], and allows them to calculate forecasts of any time series without having to assure before the conditions of stationary and invertible. After the publication of some initial works related to the prediction of time series ([3], [8]), the interest on the part of the scientific community has been increasing throughout last decade.

In this work, an approach to automatically choose the best lags in time series using an evolutionary algorithm and artificial neural networks (namely, E-tsRBF) is proposed. The method is based on EvRBF [12], [13], which was previously developed to design asymmetric Radial Basis Function Networks (RBFNs) [2].

Radial Basis Function Networks are two-layer, fully-connected, feed-forward networks, in which hidden neuron activation functions are Radial Basis Functions (RBF), usually Gaussian.

RBFNs output is given by eq. 1.

$$s_j(\vec{x_k}) = \lambda_{0j} + \sum_{i=1}^{p'} \lambda_{ij}\phi_i(\vec{x}, \vec{c_i}, \vec{r_i}) \quad (1)$$

where $k = 1..p, j = 1..n', s_j \in R, \vec{x_k} \in R^n$, and ϕ_i is the RBF assigned to hidden neuron i; λ_{0j} is a bias term; λ_{ij} represents the weight between hidden neuron i and output

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.234

neuron j; $\vec{c_i}$ and $\vec{r_i}$ are called, respectively, the *center* and *radii* (or *widths*) of the RBF; n and n' are the input and output space dimensions, respectively; p' is the number of hidden neurons, and p is the number of patterns to which s_j is going to be applied.

RBFNs' main advantage is that optimal biases and weights (i.e, λ_{0j} and λ_{ij}) can be efficiently computed for a certain set of desired output, once the number of hidden neurons, centers and radii have been set. This is shown in eqs. 2 to 4.

$$\begin{pmatrix} f_{11}..f_{1n'} \\ \vdots \\ f_{p1}..f_{pn'} \end{pmatrix} = \begin{pmatrix} 1 & A_{11}..\ A_{1p'} \\ 1 & \vdots \\ 1 & A_{p1}..A_{pp'} \end{pmatrix} \begin{pmatrix} \lambda_{01}..\lambda_{0n'} \\ \vdots \\ \lambda_{p'1}..\lambda_{p'n'} \end{pmatrix} \quad (2)$$

which can also be expressed in matricial form as in eq. 3:

$$F = A\lambda \quad (3)$$

whose solution is given by eq. 4:

$$\lambda = A^{-1^*}F \quad (4)$$

where F is the set of desired outputs; A is the so-called *design matrix* (where A_{ij} represents the output of hidden neuron j when input pattern i is applied to the net); λ is the set of weights and biases (being λ_{jk} the weight of the connection between hidden neuron j and output neuron k); and A^{-1^*} represents the *pseudo-inverse* of matrix A.

Biases and weights calculated using A^{-1^*} yield the minimum mean square error (MSE). Using less hidden neurons than values to be approximated (i.e., $p' << p$), singular value decomposition (SVD) [11] or any gradient descent method can be used to compute A^{-1^*}.

The rest of the paper is organized as follows: section II briefly introduces some of the papers found in literature closely related to this research; section III describes the method developed for this work, while section IV presents the experimentation carried out and the results obtained.

II. State of the art

The ARIMA models are a result of the work of G.E. Stall and G.M.Jenkins [1] realized in the decade of the 70, known as models Box-Jenkins. The initial problem consisted of determining the evolution of the pollution in the bay of San Francisco, with the intention of improving it prediction and control. The procedures used by both researchers had a wide diffusion and it has spread to different branches of the science.

However, the main disadvantage of the method is that it gives simplistic models that only use several previous values to forecast the future. The method is, therefore, unable to find subtle patterns in the time series data.

On the other hand, many successful applications suggest that Artificial Neural Networks (ANNs) can be a promising alternative tool for both forecasting researchers and practitioners. Zhang et al. [20] presented a review of the current status in applications of neural networks for forecasting. The popularity of ANNs is derived from the fact that they are generalized nonlinear forecasting models. Forecasting has been dominated by linear statistical methods for several decades. Linear models have many advantages in implementation and interpretation, although they have serious limitations because they cannot capture nonlinear associations in the data which are common in many complex real world problems [7].

Inside the wide range of neural networks existing in the literature, the Multilayer Perceptron (MLP) and Radial Basis Function Networks (RBFNs) enhance. The use of Radial Basis Functions (RBF) as functions of activation for neural networks and its application to time series forecasting was realized by the first time by Broomhead and Lowe in 1988 [2]. After these, new works by Carse and Fogarty [4], and Whitehead and Choate [17] focused on the prediction of time series. In all these works, the time series used to evaluate the algorithms were synthetic problems.

There also exist works in which time series taken from the real world have been forecasted by means of genetic algorithms and RBFNs. One of the best examples is Sheta and De Jong's work [16], where the data used described the rate of exchange between British pounds and American dollars during 3 years, from 1980 to 1983. The same time-series would be used later by Rivas et al. in [13].

Among the most recent papers found in literature, Rivera et al. [14] achieved to build RBFNs by means of fuzzy-rule tables. These tables indicate to the algorithm whether new neurons had to be created or destroyed, and also the direction to which the centers of the RBF should move.

III. Description of the method

This section describes E-tsRBF, a meta-evolutionary algorithm based on EvRBF [13], which is an evolutionary method developed to automatically design asymmetric Radial Basis Functions. The main goal of this new method is to forecast the time series using RBFNs building at the same time the neural network and the set of past values that must be used to predict new ones.

Every individual in this evolutionary algorithm represents a set of lags. Every chromosome is a binary string that indicates whether the specific lag will be used or not. Thus, a chromosome as 10010001 means that lags t, $t-3$, $t-7$ will be used to forecast the value in time $t + horizon$. By default, $horizon = 1$, although this is one of the parameters fixed by the user when running the algorithm. Both values of the genes and length of the chromosomes can change along the execution of the algorithm, using an adaptive probability for the evolutionary operators it includes. This probability is bigger at the begining and decreases as the number generations grows, in order to get a trade-off between exploration (at first generations) and explotation (at the end of the execution). As in EvRBF, the maximum length of the individuals of the first generation is computed as a percentage of the available values used to train the nets.

The specific characteristics of E-tsRBF are:

- Individual selection: E-tsRBF implements tournament selection to choose the individuals which are going to form the population.
- Evolutionary operators
  - Crossover operator: the method uses a multipoint crossover operator. Couples of points are randomly generated and the fragment of chromosome between them is exchanged among the two parents.
  - Mutation operator: for every gene of the chromosome a random probability is generated, if this value results lower than the probability of given mutation, the value of the gene will mutate, in other case it will remain equal.
  - AdderDeleter operator: this operator is used in order to vary the length of the chromosome through the execution. An adaptive probability is used in order to keep a balance between diversity and convergence. This adaptive probability depends on the generation number, thus first generations will try to increase diversity, while the last ones try to converge to the optimal solution. The probability is calculated using eq. 5.

$$p(i,j) = 1 - (gen(x_i)/gen(x_n)) * L_j \tag{5}$$

    where $gen(x_i)$ represents the current generation number, $gen(x_n)$ the total number of generation and L_j the chromosome length of the current individual. Independently, the length of chromosome has the same possibility of increasing like of decreasing, for that a random value is produced and it will decide that the operator increases or decreases the length.
- Function evaluation: in order to set the fitness of an individual the chromosome is decoded into the lags it represents. Then, the set of selected lags are used build a training file with which to completely run the EvRBF method. The inverse of the root mean square error computed by EvRBF will be set as the fitness of the individual.

IV. Experimentation and Results

Seventeen different data sets[1] are used in this work to test the effectiveness of E-tsRBF. These time series come from different areas and have different statistical characteristics. Next, a brief description of every one is given:

- Double map: We consider this time series as a sorted sequence of the equation represented in eq. 6.

$$x_{n+1} = 2x_n(modulo1) \tag{6}$$

  The time series is taken from the Lowe y Broomhead's work [2]. Data are composed of 500 observations, 250 of which are used to train and 250 for test.

[1]Data can be accessed at https://sites.google.com/site/presetemp/datos

- Quadratic map: This time series is considered as a organized sequence of the equation showed in eq. 7

$$x_{n+1} = 4x_n(1 - x_n) \tag{7}$$

  The information is taken from the Lowe y Broomhead's work [2]. The time series is composed of 500 observations, from which 250 were used to train and 250 for test.
- Exchange: This time-series is composed of real data representing the exchange rates between British pound and US dollar during the period going from 31 December 1979 to 26 December 1983, available from http://pacific.commerce.ubc.ca/xr/data.html, thanks to the work done by Prof. Werner Antweiler, from the University of British Columbia, Vancouver, Canada. Data are composed of 208 observations, 156 for training and 52 for test.
- MackeyGlass: This represents the time series created by Mackey and Glass. The terms of this problem are described in [10] and it has been employed in a lot of works by Whitehead y Choate [17], Carse y Fogarty [4], and Gonzalez [6] among others. The graphic representation of this time series is show in eq. 8.

$$\frac{dx(t)}{dt} = -bx(t) + a\frac{x(t-T)}{1 + x(t-T)^{10}} \tag{8}$$

  Data are composed of 1170 observations, 585 for training and 585 for test.
- Accidents: It represents the accident number during a working day. The observations express the average of accidents over a month and they cover from January 1979 until December 1998. The data are taken from the statistic national institute. Data are composed of 240 observations, 180 of which are used to train and 60 for test.
- Airline: It represents the airplane passengers of international flies. The data are the average of a month between January 1949 and December 1960. The time series have been got from *Time series analysis forecasting and control*, Box and Jenkins [1]. Data are composed of 144 observations, 108 for training and 32 for test.
- WorldMarket: It represents the month values about seven different world markets. The observations were extracted from January 1988 until December 2000. The source of the information is Eurostat. The seven world markets are the following:
  1) Paris, CAC 40 Index, France
  2) Frankfort, DAx-EXtra, Germany
  3) Milan, Italian Commercial Banc Index, Italy
  4) London, FT-SE 100 Index, United Kingdom
  5) New York, Dow Jones, United States of America
  6) Tokyo, Nikkei 225 Index, Japan
  7) Madrid, General Index, Spain

  Every time series are composed of 156 observations, 117 to train and 39 for test.
- CrestColgate: These are four time series of the market quota of toothpaste Crest and Colgate, and price of both.

The data are taken weekly among January 1958 and April 1963. The source is *Assessing the impact of market disturbances using intervention analysis* [18]. The four time series are the following:

1) Colgtems
2) Colgtepr
3) Crestms
4) Crestpr

Every time series are composed of 276 observations, 207 for training and 69 to test.

Both EvRBF and E-tsRBF models have been applied to these data sets in order to compare. The value to be forecasted was $t+1$ in all the examples but MackeyGlass, in which $t+85$ is used (according to previous works). The lags to be used for EvRBF were estimated computing the partial autocorrelation function, while E-tsRBF estimates the lags to be used by itself.

Any of the considered problems have been forecasted 30 times, using the same training and test sets in any execution. Table I shows the results yielded by both E-tsRBF and EvRBF methods. In that table, three kinds of results are showed: the mean squared error (MSE) obtained when forecasting the test file, the number of nodes composing the best net found by every method, and the time (in seconds) needed by each algorithm to get the results. Best results, i.e., lower MSE, number of nodes and seconds, are in bold.

Table I shows that the results yielded by E-tsRBF are comparable (in many cases better) than those obtained by EvRFB using only the lags provided by the partial autocorrelation function. In this sense, E-tsRBF makes easier the forecasting of the time series since no "a priori" knowledge has to be extracted from them.

The figure 1 can help to show in which problems every method outperforms the other. Nevertheless, in order to study the statistical significance of the results obtained, two kind of statistical test have been carried out. First one is ANOVA [15], which has been computed per every problem, showing that in 6 cases (Accidents, BM-Franforct, BM-Madrid, BM-Tokio, and CC-Crestpr) E-stRBF improves the results obtained by EvRBF, in 1 case (CC-Crestms) EvRBF is better than the new algorithm, and in the others the results are statistically equivalents.

The Wilcoxon test [19] has also been used in two ways. Firstly, every problem has been considered independently, as above. In this case, E-stRBF shows better results than EvRBF for 9 problems (the same than ANOVA, and also Exchange, BM-Milan y CC-ColgTems); once more, in CC-Crestms EvRBF turns to yield better results than E-stRBF, and there exist no significant differences for the others 7 problems.

Finally, the Wilcoxon test has been used to estimate the differences between both algorithms computing the mean of every algorithm on every problem, and then carrying out the test. In this case, the resulting value shows that currently it is not possible to state that E-stRBF definitively outperforms EvRBF

Thus, future work will focus on improving the new algorithm, mainly reducing the number of times the evolutionary algorithm has to run the underlying EvRBF algorithm to set the fitness of the individuals. New experiments are also being designed to study the effect of the horizon of the forecasting. Currently, the horizon is set to 1 and, in most cases, the partial autocorrelation function showed that the most important lag to be used was just $t-1$. For this reason, we shall study the way the forecasting degrades as the horizon grows.

ACKNOWLEDGEMENT

This work has been supported by the University of Jaen (Spain) UJA_08_16_30 project, the regional project TIC-3928, and the Spanish project TIN 2008-06681-C06-02.

REFERENCES

[1] G. Box and G. Jenkins. *Time series analysis, forecasting and control*. Holden-Day, Incorporated, 1976.
[2] D. Broomhead and D. Lowe. *Multivariable Functional Interpolation and Adaptative Networks*. Complex Systems, 11:321355, 1988.
[3] P. Caire, G. Hatabian, and C. Muller. *Progress in forecasting by neural networks*. In Proceedings of the International Joint Conference on Neural Networks, pages 540545, 1992.
[4] B. Carse and T. Fogarty. *Fast evolutionary learning of minimal radial basis function neural networks using a genetic algorithm*. In Proceedings of the Evolutionary Computing, Springer-Verlag, pages 122, 1996.
[5] R. Frank, N. Davey, and S. Hunt. *Time series prediction and neural networks*. Journal of Intelligent and Robotic Systems, 31(2):91103, 2001.
[6] J. Gonzalez. *Identificacion y optimizacion de redes de funciones base radiales para aprimacion funcional*. University of Granada, 2001.
[7] C. Granger and T. Terasvirta. *Modelling nonlinear economic relationships*. Oxford University Press, 1993.
[8] R. Jones, Y. Lee, and C. B. et al. *Function approximation and time series prediction with neural networks*. Proceedings of the IEEE International Joint Conference on Neural Networks, 1:649665, 1990.
[9] E. Keogh. *On the need for time series data mining benchmarks: A survey and empirical*. Data Mining and Knowledge Discovery, 7(4):349371, 2003.
[10] J. Platt. *Learning by combining memorization and gradient descent*. Advances in Neural Information Processing Systems, 3, 1991.
[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.
[12] V. Rivas, I. Garcia-Arenas, J. Merelo, and A. Prieto. *Evrbf: Evolving rbf neural networks for classification problems*. Proceedings of the International Conference on Applied Informatics and Communications (AIC07), pages 100106, 2007.
[13] V. Rivas, J. Merelo, P. Castillo, M. Arenas, and J. Castellanos. *Evolving rbf neural networks for time-series forecasting with evrbf*. Information Sciences, 165(3-4):207220, 2004.
[14] A. Rivera, I. Rojas, J. Ortega, and M. del Jesus. *A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks*. Soft Computing, 2006.
[15] G. Rupert and J. Miller. *Basics of applied statistics*. Statistical Science Series, page 336, 1997.
[16] A. Sheta and K. D. Jong. *Time-series forecasting using ga-tuned radial basis functions*. Information Sciences, 133(3-4):221228, 2001.
[17] B. Whitehead and T. Choate. *Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction*. IEEE Transactions on Neural Networks, 7(4):869880, July 1996.
[18] D. Wichern and R. Jones. *Assessing the impact of market disturbances using intervention analysis*. Management Science, 24:329337, 1977.
[19] F. Wilcoxon. *Individual comparisons by ranking methods*. Biometrics, 1:8083, 1945.
[20] G. Zhang, B. Patuwo, and M. Hu. *Forecasting with artificial neural networks: The state of the art*. International Journal of Forecasting, 14(1):3562, 1998.

TABLE I
RESULTS OF E-TSRBF AND EVRBF. COLUMNS SHOW THE MSE OVER THE TEST SET, THE NUMBER OF NODES AND TIME TOOK TO YIELD THE RESULTS.

Data set	EvRBF			E-tsRBF		
	MSE	Nodes	Time (secs.)	MSE	Nodes	Time (secs.)
Double map	3.05E-5 ±6.81E-5	25.43±10.17	**5.73**±0.88	**2.53E-6**±4.82E-6	**20.3**±9.24	23.09±13.41
Quadratic map	**3.94E-6**±2.61-6	**19.23**±8.12	**5.26**±0.66	1.58E-4±3.35E-4	20.14±11.59	22.86±0.88
Exchange	**3.07E-3**±1.80E-3	**5.57**±3.39	**1.34**±0.16	6.72E-3±1.20E-2	6.59±3.02	8.85±4.64
MackeyGlass	2.29E-3±6.61E-4	57.6±22.81	**24.64**±4.53	**1.48E-3**±1.22E-4	**31.8**±16.79	215.57±116.71
Accidents	**1.36E8**±8.87E7	11.87±3.97	**2.77**±0.24	1.61E8±5.55E7	**11.83**±3.2	15.76±0.34
Airline	**3.52E3**±4.61E3	**5.03**±1.96	**1.39**±0.08	4.61E3±1.02E3	6.6±2.3	9.89±5.33
Wm-Frankfort	1.94E4±7.58E3	6.27±2.53	**1.62**±0.10	**8.18E2**±1.10E3	**5.87**±3.49	11.34±5.46
Wm-London	**1.00E3**±8.49E2	**5.1**±2.98	**1.83**±0.12	4.64E3±4.21E3	5.23±2.62	11.64±5.97
Wm-Madrid	6.88E3±2.71E3	**3.4**±1.89	**1.59**±0.15	**3.30E3**±3.27E3	4.9±3.01	10.73±5.79
Wm-Milan	**7.36E3**±4.63E3	4.8±2.47	**1.67**±0.14	1.14E4±8.25E3	**4.47**±2.5	11.70±6.22
Wm-NewYork	1.35E2±1.29E2	7.67±3.33	**1.87**±0.18	**7.62E1**±6.61E1	**7.0**±4.88	10.76±5.99
Wm-Paris	**2.17E3**±2.04E3	**4.13**±1.76	**1.53**±0.12	4.91E3±5.26E3	6.1±3.44	10.87±5.38
Wm-Tokyo	3.97E1±7.55E0	**6.73**±2.24	**1.80**±0.09	**2.63E1**±6.36E1	8.0±3.14	11.31±4.97
Cc-colgtems	1.49E-3±7.16E-5	**3.67**±2.17	**2.75**±0.10	**1.44E-3**±7.94E-5	11.1±6.19	20.93±0.50
Cc-colgtepr	3.88E-3±2.50E-3	**5.23**±3.45	**2.70**±0.18	**1.87E-3**±1.50E-3	7.17±6.42	20.65±0.55
Cc-crestms	**1.59E-3**±1.38E-4	**7.6**±6.2	**2.66**±0.38	2.66E-3±7.03E-4	10.87±3.85	20.08±0.31
Cc-crestpr	5.12E-3±1.98E-3	**5.67**±2.47	**2.69**±0.19	**4.24E-3**±1.15E-3	10.27±5.64	20.82±0.47

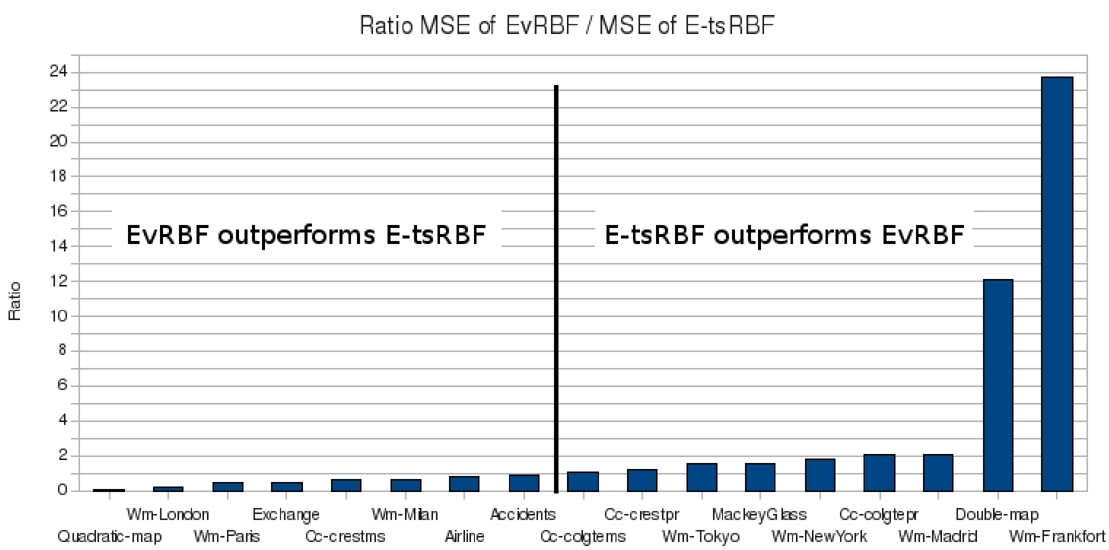

Fig. 1. Simulation Results