Evolving Time Series Forecasting Neural Network Models

Paulo Cortez
Departamento de Informática
Universidade do Minho
4750-057 Braga, Portugal
pcortez@di.uminho.pt

Miguel Rocha
Departamento de Informática
Universidade do Minho
4750-057 Braga, Portugal
mrocha@di.uminho.pt

José Neves
Departamento de Informática
Universidade do Minho
4750-057 Braga, Portugal
jneves@di.uminho.pt

Abstract

In the last decade, bio-inspired methods have gained an increasing acceptation as alternative approaches for *Time Series Forecasting*. Indeed, the use of tools such as *Artificial Neural Networks (ANNs)* and *Genetic and Evolutionary Algorithms (GEAs)*, introduced important features to forecasting models, taking advantage of nonlinear learning and adaptive search. In the present approach, a combination of both paradigms is proposed, where the *GEA's* searching engine will be used to evolve candidate *ANNs* topologies, enhancing forecasting models that show good generalization capabilities. A comparison was performed, contrasting bio-inspired and conventional methods, which revealed better forecasting performances, specially when more difficult series were taken into consideration; i.e., nonlinear and chaotic ones.

Keywords: Artificial Neural Networks, Genetic and Evolutionary Algorithms, Time Series Forecasting, Model Selection

1 Introduction

Nowadays, the ability to forecast the future, based only on past data, leads to strategic advantages, which may be the key to success in organizations. *Time Series Forecasting (TSF)*, the forecast of a time ordered variable, allows one to model complex systems, where the goal is to predict the system's behavior and not how the system works. Indeed, in the last few decades an increasing focus as been put over this field. Contributions from the arenas of *Operational Research*, *Statistics*, and *Computer Science* has lead to solid *TSF* methods that replaced the old fashioned ones. Although these methods give accurate forecasts on linear *Time Series (TS)*, they carry an handicap with noisy or nonlinear components, which are common in real world situations (e.g., in financial daily *TSs*) [22].

An alternative approach for *TSF* arises from the *Artificial Intelligence (AI)* field, where one has observed a trend to look at *Nature* for inspiration, when building problem solving models. In particular, studies on the nervous system and biological evolution influenced the loom of powerful tools, such as *Artificial Neural Networks (ANNs)* and *Genetic and Evolutionary Algorithms (GEAs)*, that enriched the potential use of *AI* in a broad set of scientific and engineering problems, such as the ones of *Combinatorial* and *Numerical Optimization*, *Pattern Recognition* or *Computer Vision* [12].

ANNs are connectionist models that mimic the central nervous system, being innate candidates for *TSF* due to capabilities such as nonlinear learning, input-output mapping and noise tolerance. Indeed, comparative studies have shown that *ANNs* can perform as well or even better than conventional methods (Sharda & Patil, 1990; Tang & Fishwick, 1993). On the other hand, *GEAs* are suited for combinatorial problems, where the exhaustion of all possible solutions requires huge computation. *GEAs* perform a global multi-point search, being able to escape from undesired local minima.

The present work aims at combining both bio-inspired approaches, over a broad range of real and artificial *TSs*. The paper is organized as follows: firstly, the basic concepts for *TS* analysis are defined; then, *ANNs* are briefly introduced and the *ANNs* forecasting models are presented. Next, a description of the different experiments performed on *ANNs* models is given and the results are analyzed; the *GEAs* are then defined, as well as the combination of both paradigms under the proposed approach. Finally, the results obtained are shown and discussed, being compared with other conventional *TSF* methods (e.g., *Exponential Smoothing* or *ARIMA*).

2 Time Series Forecasting

A *Time Series (TS)* is a collection of time ordered observations x_t, each one being recorded at a specific time t (period). *TS* can uprise in a wide set of domains such as *Finance*, *Production* or *Control*, just to name a few. A *TS* model ($\widehat{x}_t$) assumes that past patterns will occur in the future. The *error* of a forecast is given by the difference between actual values and those predicted by the model:

$$e_t = x_t - \widehat{x}_t \tag{1}$$

The overall performance of a model is evaluated by a forecasting accuracy measure, namely the *Sum Squared Error (SSE)*, *Root Mean Squared (RMSE)* and *Normalized Mean Square Error (NMSE)*, given in the form:

$$\begin{aligned} SSE &= \textstyle\sum_{i=1}^{l} e_i^2 \\ RMSE &= \sqrt{\frac{SSE}{l}} \\ NMSE &= \frac{SSE}{\sum_{i=1}^{l} (x_t - \overline{x})^2} \end{aligned} \tag{2}$$

where l denotes the number of forecasts and $\overline{x}$ the mean of the *TS*.

Table 1: *Time Series* data

Series	Type	Domain	Description
passengers	*Seasonal*	Tourism	Monthly international airline passengers
paper	*& Trended*	Sales	Monthly sales of French paper
deaths	*Seasonal*	Traffic	Monthly deaths & injuries in UK roads
maxtemp		Meteorology	Maximum temperature in Melbourne
chemical	*Trended*	Chemical	Chemical concentration readings
prices		Economy	Daily *IBM* common stock closing prices
sunspots	*Stationary*	Physics	Annual Wolf's Sunspot Numbers
kobe		Geology	Seismograph of the Kobe earthquake
quadratic	*Chaotic*	Artificial	Quadratic map
henon		Artificial	Henon map

A common statistical instrument for *TS* analysis is the *autocorrelation* coefficient, defined by:

$$r_k = \frac{\sum_{t=1}^{s-k}(x_t - \overline{x})(x_{t+k} - \overline{x})}{\sum_{t=1}^{s}(x_t - \overline{x})} \quad (3)$$

in terms of the k's lag, where s denotes the *TS*' size. Autocorrelations can be useful for decomposition of the *TS* main components (e.g., trend and seasonal effects).

One popular *TSF* method is *Exponential Smoothing (ES)*, also known as *Holt-Winters*, which is based on underlying patterns (e.g., trend and seasonal ones) that are distinguished from random noise by averaging the historical values [14]. This popularity is due to advantages such as the simplicity of use, the reduced computational demand, and the accuracy on short-term forecasts, specially with seasonal series.

The *AutoRegressive Integrated Moving-Average (ARIMA)* is another important *TSF* methodology, going through model identification, parameter estimation, and model validation [3]. The main advantage relies on the accuracy over a wider range of series domains, despite being more complex, in terms of usability and computational effort, than *ES*. The *ARIMA* model is based on a linear combination of past values and errors, being its parameters estimated using statistical approaches (e.g., least squares methods).

To the experiments carried out in this work, a set of ten series was selected (Table 1), ranging from financial markets to natural processes [3][14][10] (Figure 4). The last two series were artificially created, using the chaotic formulas: $x_t = ax_{t-1}(1 - x_{t-1}), x_0 = 0.2, a = 4$ for the **quadratic** series [15]; and $x_t = 1 - ax_{t-1}^2 + bx_{t-2}$, $a = 1.4$, $b = 0.3$, $x_0 = 0.11$ for the **henon** one [2]. A gaussian noise was also added each value of the last series, with a standard deviation equal to 0.1. All *TS* were classified into five main categories, that encompass the majority of the *TS's* types, namely: *Seasonal and Trended*, *Seasonal*, *Trended*, *Stationary* and *Chaotic*.

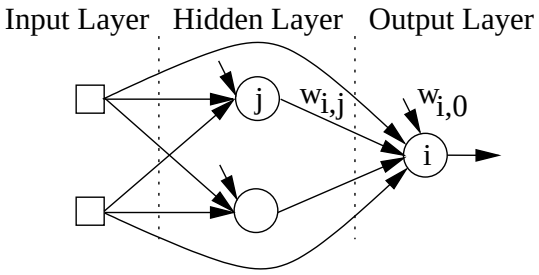

Figure 1: A fully connected *FNN* with 2 inputs, 2 hidden nodes, 1 output, bias and shortcut connections.

3 Connectionist Forecasting Models

An *Artificial Neural Network (ANN)* is made up by simple processing units, the *neurons*, which are connected in a network by synaptic strengths, where the acquired knowledge is stored. One can find a kaleidescope of different *ANNs*, that diverge on several features, such as the learning paradigm or the internal architecture [8]. In a *Feedforward Neural Network (FNN)*, neurons are grouped in layers and only forward connections exist (Figure 1). This provides a powerful architecture, capable of learning any kind of continuous nonlinear mapping, with successful applications ranging from *Computer Vision*, *Data Analysis* or *Expert Systems*, just to name a few. *FFNs* are usually trained by gradient descent algorithms, such as the popular *Backpropagation*, or fast variants like *RPROP* [17].

The use of *ANNs* for *TSF* began in the late eighties, with the work of Lapedes and Farber [11], where *FFNs* were used to predict chaotic deterministic *TS*. Forecasting competitions, confronting *ANNs* with traditional *TSF* methods, have reported either poor (e.g., M competition [13]) or favorable results (e.g., Santa Fé competition [25]). Results seem contradictory for the latter competition, where *FFNs* got the best and the worst results, showing that a great care is needed when fitting *ANNs* as forecasting models. Other studies also showed that *ANNs* can forecast as well or even better than the *ARIMA* methodology [20][23].

Under this context, there are several *ANN* candidates, such as *Radial Basis Functions* [21] or *Recurrent Neural Networks*

[24], although most studies use *FNNs* [23, 4, 5]. This last architecture will be adopted, using fully connected networks, with *bias* and shortcut connections (from input to output nodes), since these links add a linear component to the model (Figure 1). To enhance nonlinearity, the *logistic* activation function was applied on the hidden nodes, while in the output node, the *linear* function was used instead, to scale the range of the outputs (the logistic function has a [0,1] co-domain). This solution avoids the need of filtering procedures, which may give rise to loose information (e.g., *rescaling*).

A *Sliding Time Window (STW)* defines the set of time lags used to build a forecast, being denoted by the sequence $STW =< k_1, k_2, ..., k_n >$, for a network with n inputs and k_i time lags. For one-step ahead forecasts, one output is trained to map present values with past ones (given by the *STW*). The general model provided by the *ANN* is given in the form:

$$\widehat{x}_t = w_{o,0} + \sum_{i=1}^{n} x_{t-k_i} w_{o,i} + \sum_{j=n+1}^{o-1} f(\sum_{i=1}^{n} x_{t-k_i} w_{j,i} + w_{j,0}) \quad (4)$$

where o denotes the output node, f the logistic function ($f(x) = \frac{1}{1+e^{-x}}$), and n the number of input nodes (or the size of the *STW*).

Generalization is influenced by three factors: the size of the training set (a function of the size of the *TS*), the number of parameters of the model (number of weights), and the complexity of the problem at hands (*TS* patterns). A model is said to overfit when it correctly handles the training data but fails to generalize. The usual statistical approach to *model selection* is to estimate the generalization errors of different candidate models, being selected the model with the minimum of such estimate. Several complex estimators have been developed (e.g., Bootstrapping), which are computationally burdensome [18]. A reasonable alternative is the use of simple statistics that add a penalty that is a function of model complexity, such as the *Akaike's Information Criterion (AIC)* [1] or the *Bayesian Information Criterion (BIC)* [19], which are depicted bellow:

$$\begin{aligned} AIC &= N \times ln(\tfrac{SSE}{N}) + 2p \\ BIC &= N \times ln(\tfrac{SSE}{N}) + p \times ln(N) \end{aligned} \quad (5)$$

where N denotes the number of training examples and p the number of parameters or weights, in this case:

$$p = n(n_h + 1) + 2n_h + 1 \quad (6)$$

for a network with n_h hidden nodes.

4 Model Selection

What is the best sliding window for a given *TS*? A large sliding window can increase the system complexity, diminishing the learning capabilities of the model, while small windows may contain insufficient information. The selection of the relevant time lags can improve forecasting (e.g., *ARIMA* models often use the 1, 12 and 13 lags for monthly seasonal trended series).

An empirical approach to the problem is to use information based on the *TS* analysis. Four heuristic strategies will be tested for time lag selection, based on the series' characteristics, namely:

A - a full *STW* with all lags in a given range: $STW =< 1, 2, ..., m >$ (m was set to 13, a value that was considered sufficient to encompass monthly seasonal and trended effects);

B - a *STW* with all time lags with autocorrelations above a given threshold (which was set to 0.2);

C - a *STW* with the time lags with the four highest autocorrelations (in the case of the seasonal trended series, these were taken after differencing, since trend effects may prevail in the original form); and

D - the use of decomposable information; i.e.,

- $STW =< 1, K, K+1 >$ if the series is seasonal (period K) and trended;
- $STW =< 1, K >$ if the series is seasonal ; and
- $STW =< 1 >$ and $STW =< 1, 2 >$ if the series trended.

Several *FNNs*, with a number of hidden nodes (n_h) ranging from 0 to 13, were used to explore all sliding windows for each *TS* of Table 1. Each model was trained with 90% of the series elements, being the rest 10% used for the forecasts. The initial weights were randomly generated within the range $[\frac{-2}{i}; \frac{2}{i}]$, for a node with i inputs. The *RPROP* algorithm was used since it allows a faster convergence, being also more stable in terms of its parameters adjustment [17]. The *ANN* learning proceeds until the *training progress* drops; i.e, when the training error slope is approaching zero [16]. This criteria can not warranty in itself termination; but making a disjunctive pair with a well defined number of training epochs (here set to 1000), a termination criteria was enforced. Since different random starting weights can generate different results, thirty independent runs were applied to each model.

As an example, the methodology used will be explained in detail for the **sunspots** series (Table 2). Here, only the first three sliding window strategies were applied, since this is a stationary series. To simplify the explanation, only some relevant results are visible in this table. The results of the last four columns are given in terms of the mean of the thirty runs. The 95% confidence intervals are also shown for the forecasting errors [6]. The best training error ($RMSE_t$) is achieved for window **A** and 12 hidden nodes. This model has an high number of parameters and overfits. In fact, more hidden nodes result in lower training *RMSE* values; indeed, when a network specializes, its performance usually degrades. The model suggest by the *AIC* values still has too many parameters and also overfits. In contrast, the *BIC* criterion seems to

Table 2: *ANNs*' results of the empirical approach to the **sunspots** series

STW	n_h	p	$RMSE_t$	AIC	BIC	$RMSE_f$
	0	14	14.7	1355	1404	18.2±0.1
A	6	104	11.6	1419	1784	20.5±0.7
	12	194	**9.0**	1474	2155	20.2±0.8
	0	5	14.8	1345	1369	17.9±0.1
B	5	47	11.5	1302	1467	**17.0**±0.6
	13	111	9.4	1328	1717	19.0±0.8
	0	5	15.1	1352	1369	18.1±0.0
C	1	11	14.1	1329	**1368**	17.8±0.3
	8	53	10.7	**1278**	1464	19.4±0.5

Table 3: Model selection criteria performance

Series	RMSE	AIC	BIC
passengers	13.5%	**5.58**%	**5.58**%
paper	14.7%	**0.65**%	**0.65**%
deaths	4.00%	**0.66**%	**0.66**%
maxtemp	11.7%	**6.76**%	**6.76**%
chemical	**0.63**%	6.91%	15.17%
prices	**0.00**%	**0.00**%	**0.00**%
sunspots	18.7%	14.42%	**4.88**%
kobe	12.8%	**0.00**%	**0.00**%
quadratic	27.3%	**0.00**%	**0.00**%
henon	19.3%	20.84%	**0.00**%
Mean	12.3%	5.58%	**3.37**%

work better, selecting a network that provides one of the best short term forecasts, measured in terms of the *RMSE* (column $RMSE_f$). However, the best forecasting model (the one with the lower *RMSE*) seems unattainable.

Faraday and Chatfi eld [5] recommended the *BIC* criterion for the comparison of different *ANN* topologies, although the unique tested series were the **passenger** one. A more detailed test for the performance of the model selection criteria with *ANNs* is given in Table 3. The quality of each *Selected Model (SM)* will be measured in terms of how far it is from the best model (in percentage), being given by:

$$100 \times \frac{RMSE_{SM} - RMSE_{best}}{RMSE_{best}} \qquad (7)$$

Results confi rm that the *RMSE* measures lead to overfi tting with *ANNs*. The few exceptions occur with the trended series, although the feeling is that it would ultimately overfi t for these kind of series, with an higher number of hidden nodes. The *AIC* and *BIC* criteria suggest identical models for seven of the series. However, the overall mean shows that *BIC* provides better short term forecasts, failing only 3.37% in average to get the best *ANN* model.

Table 4 shows the best *ANNs*, when adopting the *BIC* values for model selection. As expected, this criterion suggests small sliding windows and linear models (with $n_h = 0$ hidden nodes) for all linear series. However, the *BIC* statistic also favors simple models (with zero or one hidden nodes) for the nonlinear series (**kobe** and **sunspots**). For instance, the best forecasts for series **sunspots** are given for a network with 5 hidden nodes, although resulting in a higher *BIC* value, due to an excessive weights' number (Table 2).

Table 4: *ANNs* forecasting models with lower *BIC* values.

Series	STW	n_h	p	$RMSE_f$
passengers	$<1,12,13>$	0	4	18.4±0.2
paper	$<1,12,13>$	0	4	51.6±0.2
deaths	$<1,11,12,13>$	0	5	134±1
maxtemp	$<1,11,12,13>$	0	5	0.90±0.02
chemical	$<1,2>$	0	3	0.40±0.01
prices	$<1>$	0	2	7.49±0.00
sunspots	$<1,2,9,...,12>$	1	11	17.8±0.3
kobe	$<1,...,13>$	0	14	557±4
quadratic	$<1,...,13>$	9	149	0.06±0.02
henon	$<1,...,13>$	2	44	0.35±0.05

In terms of the *BIC* criterion, adopting fully connected networks seems to prejudice networks with hidden nodes, due to the number of extra weights implied by adding a single hidden node. Moreover, the sliding window heuristics are ultimately based on autocorrelation values, which only measure linear interactions that are not adequate for nonlinear series.

5 Evolutionary Neural Network Approach

The term *Genetic and Evolutionary Algorithm (GEA)* is used to name a family of computational procedures where a number of potential solutions to a problem makes the way to an evolving population. Each individual codes a solution in a string (*chromosome*) of symbols (*genes*), being assigned a numerical value (*fitness*), that stands for a solution's quality measure. New solutions are created through the application of genetic operators (typically *crossover* or *mutation*). The whole process evolves via a process of stochastic selection biased to favor individuals with higher fi tnesses.

The original model, due to Holland (1975), uses a binary alphabet to encode the problem's solutions. In each generation, all individuals are replaced by the offspring, which are generated by the application of the two genetic operators, *crossover* and *mutation*. In this work, the two-point crossover and bit mutation operators were considered [7].

GEAs and *ANNs* have been combined in three major ways: to set the weights in fi xed architectures, to learn neural network topologies, and to select training data for *ANNs* [26]. The attention will be directed at the last two tasks in order to tackle the issues raised in Section 4; i.e, what is the best sliding time window and *ANN* topology for a given *TS*? These issues can be addressed by adopting specifi c heuristics with trial-and-error procedures (such as the ones applied in Section 4), which tend to be unsuitable due to the huge size of the search spaces involved.

A *GEA* is proposed, where an individual codes a different *ANN* topology, each gene representing a possible connection. If its value is 1, then the corresponding connection exists, otherwise it is not considered. The connections between the hidden nodes and the output one always exist, since this strategy enhances the creation of valid networks.

Assuming a fully connected network with n inputs nodes, n_h hidden nodes, bias and shortcut connections, the size of the chromosome will be given by $(n+1) \times (n_h+1)$. Under this scenario, hidden node pruning will occur when there are no connections from input nodes, and time lag pruning will occur when an input node has no outputs. This allows the *GEA* to explore a search space containing *ANN* topologies from the simplest linear network (with no hidden nodes) to the fully connected one, and also to reach any subset of time lags.

The fitness of each individual is obtained by decoding its chromosome into the *ANN*, training it and, finally, calculating the *BIC* values. The aim of the computation is, therefore, to find the *ANN* topology and sliding time window that minimizes the *BIC* value. The overall system is depicted in Figure 2.

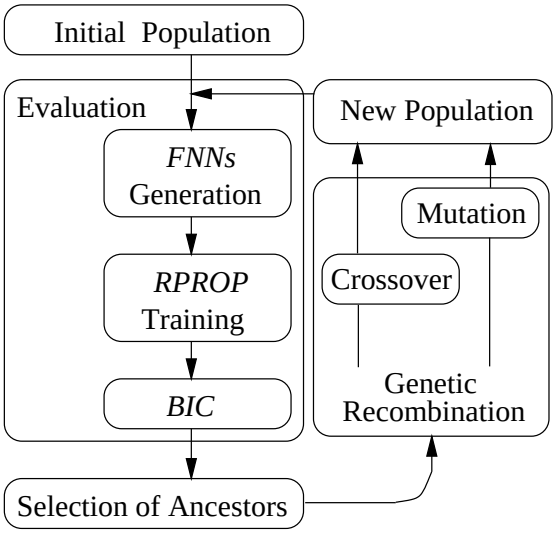

Figure 2: The evolutionary approach

6 Experiments on the Evolutionary Approach

The initial populations' genes were randomly assigned within the alphabet $\{0, 1\}$. The maximum number of input (n) and hidden nodes (n_h) were set to 13 and 6. Previous experiments have already favored small networks and larger values would enlarge the search space with no added benefits. The population size was set to 100 individuals. The *crossover* operation is responsible for breeding 80% of the offspring and the *mutation* operation is accountable for the remaining ones. The *selection* procedure is done by converting the fitness value into its ranking in the population and then applying a roulette wheel scheme. The *GEA* is stopped after a convenient number of generations, here set to 500, once the best individuals were found in earlier generations. Thirty independent *ANN* trainings are applied to the best topology obtained during the evolutionary process, being the final result presented as the average of the runs.

Table 5: Evolutionary forecasting models.

Series	**STW**	n_h	p
passengers	$< 1, 12, 13 >$	0	3
paper	$< 12 >$	1	4
deaths	$< 1, 2, 11, 12, 13 >$	1	6
maxtemp	$< 1, 2, 3, 6, 10, ..., 13 >$	3	13
chemical	$< 1, 2, 7, 12, 13 >$	2	7
prices	$< 1, 7, 13 >$	1	4
sunspots	$< 1, 2, 3, 8, 11, 12, 13 >$	3	17
kobe	$< 1, ..., 7, 9, ..., 13 >$	3	17
quadratic	$< 1, 2, 4, 6, 7, 8, 9, 12, 13 >$	6	34
henon	$< 1, ..., 5, 8, 9, 11, 12, 13 >$	4	23

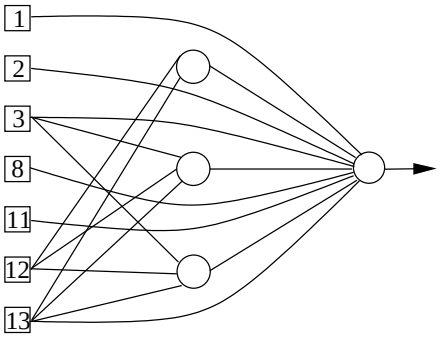

Figure 3: The best model for the **sunspots** series.

Table 5 shows the best models achieved by the *GEA*, for all series of Table 1. As an example, Figure 3 plots the best *ANN* topology for the **sunspots** series. In comparison with the previous results (Table 4), the ones obtained by the evolutionary approach (Table 6) show better forecasts (the exception is series **paper**), specially for the nonlinear series. Here, the higher flexibility of the *GEA* allows to select models with low *BIC* values for *ANN's* topologies with a small number of weights, despite having some hidden nodes.

A comparison throughout bio-inspired and conventional models is given in Table 7. The error values in the table are given in terms of two measures, namely the *RMSE* and the *NMSE* (in brackets). This last measure is included since it makes easier the comparison among the different series and methods. Each model was optimized using all known values from the *TS*, excluding the last 10% values, which will be used for forecasting. The *ES* parameters (α, β and γ) were optimized using a 0.01 grid search for the best *RMSE*, while the *ARIMA* models were achieved using a forecasting package (*FORECAST PRO*).

ES gives a better overall performance on the seasonal

Table 6: Evolutionary forecasting results.

Series	$RMSE_f$
passengers	18.2±0.3
paper	52.5±0.6
deaths	132±1
maxtemp	0.87±0.02
chemical	0.36±0.01
prices	7.49±0.01
sunspots	17.4±0.5
kobe	498±8
quadratic	0.01±0.00
henon	0.24±0.02

Table 7: Comparison between different *TSF* approaches

Series	**ES**	**ARIMA**	**ENN**
passengers	**16.7** (0.71%)	17.8 (0.81%)	18.2 (0.84%)
paper	**41.0** (3.1%)	61.0 (6.8%)	52.5 (5.0%)
deaths	145 (43%)	144 (42%)	**132** (36%)
maxtemp	0.91 (4.1%)	1.07 (5.6%)	**0.87** (3.8%)
chemical	**0.35** (51%)	0.36 (53%)	0.36 (54%)
prices	7.50 (0.39%)	7.72 (0.41%)	**7.49** (0.38%)
sunspots	28.4 (35%)	21.4 (20%)	**17.4** (13%)
kobe	3199 (105%)	582 (3.5%)	**498** (2.6%)
quadratic	0.38 (101%)	0.35 (101%)	**0.01** (0.07%)
henon	0.76 (106%)	0.63 (83%)	**0.24** (13%)

trended series. This is not surprising, since *ES* was developed specifically for these kind of series. In the case of the trended series (**chemical** and **prices**), both *ES* and the proposed *Evolutionary Neural Network (ENN)* produce comparable results. However, this scenario differs when considering other series, where the evolutionary approach outperforms both conventional *TSF* methods, specially for the last four nonlinear series. For the chaotic series, conventional approaches fail, while *ENN* captures perfectly the nonlinear effects, producing accurate forecasts.

7 Conclusions and Future Work

The surge of new bio-inspired optimization techniques, such as *ANNs* and *GEAs*, has created new exciting possibilities for the field of forecasting. Currently, the application of these bio-inspired methods requires some effort from an analyst, in processes such as data analysis (e.g., preprocessing and feature analysis) and model selection. In this work, a systematic approach is applied, assuming no prior knowledge over each series (e.g., the use of specific known transformations). Furthermore, the system works autonomously and does not require any kind of statistical preprocessing or analysis. The drawback is the increase of computational effort required.

Comparative experiments, among conventional and bio-inspired approaches, with several real and artificial series from diverse domains, were held. These show that *ES*, although very simple, gives a good overall performance on linear *TS* (seasonal and trended ones), being also a method that requires few computational resources. However, when the domain gets more complex, with nonlinear behavior, this kind of methods is clearly not appropriate. The proposed approach shows its strength exactly in these scenarios. The results so far obtained prevail both on nonlinear series or on the linear ones, specially on those with seasonal components.

In the experiments conducted, it was possible to verify that *ANNs* are powerful methods, yet rely heavily on the network design. Poor structures provide insufficient learning capabilities while too complex ones lead to overfitting. There is the need of methods to select which topologies provide the best generalizations for a given task. The proposed *ENN*, based on the *BIC* criterion, has revealed itself as an adequate solution for the forecasting domain, although it is potentially useful for any other applications.

In future research it is intend to explore different *ANNs* topologies, such as *Recurrent Neural Networks* or *Radial Basis Functions*. In terms of the *GEA* it is possible that its performance could be enhanced by more appropriate control of genetic diversity in order to avoid local minima. Finally, one promising field is the application of similar approaches to long term and multivariate forecasting.

Acknowledgements

The work of Paulo Cortez was supported by the portuguese *Foundation of Science & Technology* through the PRAXIS XXI/BD/13793/97 grant. The work of José Neves was supported by the PRAXIS' project PRAXIS/P/EEI/13096/98.

Bibliography

[1] H. Akaike. Information Theory and an Extension of the Maximum Likehood Principle. In B. N. Petrov and F. Csaki, editors, *2nd International Symposium on Information Theory*, pages 267–281, Budapest, 1973.

[2] E. Azoff. *Neural Network Time Series Forecasting of Financial Markets*. John Wiley & Sons, 1994.

[3] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco, USA, 1976.

[4] P. Cortez, M. Rocha, J. Machado, and J. Neves. A Neural Network Based Forecasting System. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Western Australia, November 1995.

[5] J. Faraday and C. Chatfield. Times Series Forecasting with Neural Networks: A Case Study. *Applied Statistics*, 47:231–250, 1998.

[6] A. Flexer. Statistical evaluation of neural networks experiments: Minimum requirements and current practice.

In *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, volume 2, pages 1005–1008, Vienna, Austria, 1996.

[7] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc., NY, USA, 1989.

[8] S. Haykin. *Neural Networks - A Compreensive Foundation*. Prentice-Hall, New Jersey, 2nd edition, 1999.

[9] John Holland. *Adaptation in Natural and Artificial Systems*. PhD thesis, University of Michigan, Ann Arbor, 1975.

[10] R. Hyndman. *Time Series Data Library*. (http://www-personal.buseco.monash.edu.au/˜hyndman/TSDL/), 2000.

[11] A. Lapedes and R. Farber. Non-Linear Signal Processing Using Neural Networks: Prediction and System Modelling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, USA, 1987.

[12] G. Luger and W. Stubblefi eld. *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*. Addison Wesley Longman, Inc., USA, 1998.

[13] S. Makridakis. The acuracy of extrapolation (times series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1:111–153, 1982.

[14] S. Makridakis and S. Wheelwright. *Forecasting Methods for Management*. John Wiley & Sons, New York, fi fth edition, 1989.

[15] H. Peitgen, H. Jürgens, and D. Saupe. *Chaos and Fractals - New Frontiers of Science*. Springer-Verlag, New York, 1992.

[16] L. Prechelt. *Early Stopping – but when?* In: *Neural Networks: Tricks of the trade*, Springer Verlag, Heidelberg, 1998.

[17] M. Riedmiller. Supervised Learning in Multilayer Perceptrons - from Backpropagation to Adaptive Learning Techniques . *Computer Standards and Interfaces*, 16, 1994.

[18] W. Sarle. Stopped Training and Other Remedies for Overfi tting. In *Proceedings of the 27th Symposium on the Interface of Computer Science and Statistics*, pages 352–360, 1995.

[19] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–4, 1978.

[20] R. Sharda and R. Patil. Neural Networks as forecasting experts: An empirical test. In *Proc. of the International Joint Conference on Neural Networks*, volume 1, pages 491–494, IEEE,Washington, DC, USA, 1990.

[21] Z. Shi, Y. Tamura, and T. Ozaki. Nonlinear time series modelling with the radial basis function-based state-dependent autoregressive model. *International Journal of Systems Science*, 30(7):717–727, 1999.

[22] E. Shoneburg. Price Prediction using Neural Networks: A Project Report. *Neurocomputing*, 2:17–27, 1990.

[23] Z. Tang and F. Fishwick. Feed-forward Neural Nets as Models for Time Series Forecasting. *ORSA Journal of Computing*, 5(4):374–386, 1993.

[24] C. Ulbricht. Multi-recurrent Networks for Traffi c Forecasting. In *Proceedings of AAAI'94 Conference*, Seatle, Washington, USA, July 1994.

[25] A. Weigend and N. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, USA, 1994.

[26] D. Whitley. Genetic Algorithms and Neural Networks. In J. Perioux, G. Winter, M.Galan, and P.Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*. John Willey & Sons Ltd., 1995.

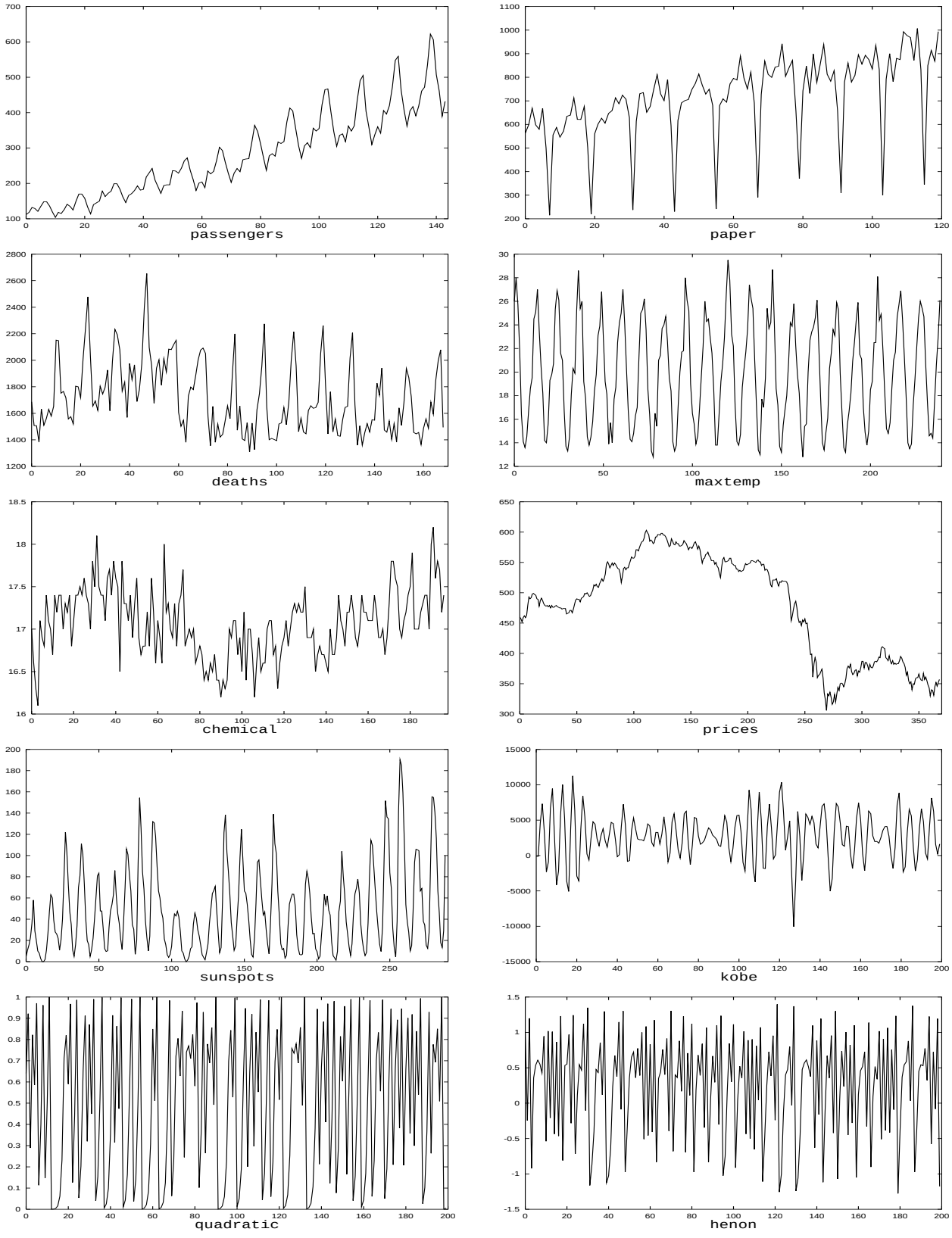

Figure 4: The series of Table 1