Soft Comput
DOI 10.1007/s00500-011-0784-2

ORIGINAL PAPER

Coevolution of lags and RBFNs for time series forecasting: L-Co-R algorithm

E. Parras-Gutierrez · M. Garcia-Arenas · V. M. Rivas · M. J. del Jesus

Abstract This paper introduces Lags COevolving with Rbfns (L-Co-R), a coevolutionary method developed to face time-series forecasting problems. L-Co-R simultaneously evolves the model that provides the forecasted values and the set of time lags the model must use in the prediction process. Coevolution takes place by means of two populations that evolve at the same time, cooperating between them; the first population is composed of radial basis function neural networks; the second one contains the individuals representing the sets of lags. Thus, the final solution provided by the method comprises both the neural net and the set of lags that better approximate the time series. The method has been tested across 34 different time series datasets, and the results compared to 6 different methods referenced in literature, and with respect to 4 different error measures. The results show that L-Co-R outperforms the rest of methods, as the statistical analysis carried out indicates.

Keywords Neural networks · Coevolutionary algorithms · Time series forecasting · Significant lags

E. Parras-Gutierrez · V. M. Rivas (✉) · M. J. del Jesus
Department of Computer Sciences,
Campus Las Lagunillas s/n, 23071 Jaen, Spain
e-mail: vrivas@vrivas.es

E. Parras-Gutierrez
e-mail: eparrasg@vrivas.es

M. J. del Jesus
e-mail: mjjesus@ujaen.es

M. Garcia-Arenas
Department of Computers, Architecture and Technology,
C/ Periodista Daniel Saucedo s/n, 18071 Granada, Spain
e-mail: mgarenas@atc.ugr.es

1 Introduction

Time series forecasting can be described as the task of predicting values of the series based on past and present values through the time line in order to achieve the information of the underlying model. Thus, time series represent chronological sequences of observed data, and they are present in many activities in different fields like Engineering, Biology, Economy, or Social Sciences, among many others.

Techniques developed to face with time series forecasting can be grouped in descriptive traditional technologies, linear and nonlinear modern models, and technologies arisen inside the area of soft computing.

Among the available modern methods (see Crone et al. 2011 for an update review), the model described by Box and Jenkins (1976), ARIMA, is probably the most widely used in time series forecasting. This method forecasts future time series data with a combination of autoregressive and moving average data. The autoregressive part relates the future value to past and present values in a linear fashion. The moving average component relates the future value to the errors of previous forecasts. The disadvantage of the method is that it gives simplistic linear models, being unable to find subtle patterns in the time series data.

Regarding the soft computing area, time series forecasting has been tackled by means of diverse techniques, such as fuzzy logic, expert systems, evolutionary algorithms (EAs) and, especially, artificial neural networks (ANNs). Many successful applications have shown that ANNs are a suitable alternative tool for both forecasting researchers and practitioners, due to their learning and generalization capabilities.

Independently of the model being used, one of the main problems that emerge working with time series is the

Published online: 08 November 2011

choice of the time periods (or lags) that must be used in order to forecast values. This way, the own selection of the input variables to build the model turns itself into a problem that can be faced using data mining techniques.

In this work, the combined use of radial basis function networks (RBFNs) and EAs is proposed in order to find both: (a) the neural network that models the time series, and (b) the set of significant lags the net needs to forecast future values. This kind of learning processes in which two complementary and dependent objectives exist is suited to be tackled by coevolutionary algorithms (Paredis 1995). Therefore, we propose Lags COevolving with Rbfns (L-Co-R), a coevolutionary algorithm able to jointly solve the two problems.

L-Co-R is a coevolutionary method developed to find an optimum minimization of the error obtained for time series forecasting. It evolves two populations in which any member of one of the population can cooperate with individuals from the other one in order to progressively generate better solutions. So, the first population evolves sets of significant lags for the time series that are used to forecast future values. The second population evolves a set of RBFNs obtaining a suitable design for time series prediction. The last implies the establishment of the architecture and parameters associated with the nets: number of layers, connection between neurons, optimum set of weights, and radii and centers for neurons.

The rest of the paper is organized as follows: Sect. 2 introduces some preliminary topics related to this research; Sect. 3 describes the method L-Co-R; Sect. 4 presents the experimentation carried out and the results obtained, and finally Sect. 5 presents some conclusions of the work.

2 Preliminaries

2.1 Time series

A time series is a set of observations from a variable along the time in regular intervals (every day, every month, every year, and so on) (Pena 2005). Thus, time series forecasting consists of predicting future values based on present and past values, and external factors, when available. The main objective is to analyze the evolution of the variable taking into account the past behavior, and predict next values with accurate forecasting. Furthermore, forecasting can be divided into short-term, medium-term, and long-term. Generally, forecasting is trended to short-term prediction such as one-step ahead prediction, since longer period prediction (medium-term or long-term) is more difficult, and sometimes may not be reliable because of the error propagation (Chatterjee and Siarry 2006).

To determine the accuracy of the forecast method applied to time series data, many measures have been proposed. Most textbooks recommended the use of the Mean Absolute Percentage Error (MAPE) (Bowerman et al. 2004) and it was the primary measure in the M-competition (Makridakis et al. 1982). Other works recommended other measures such as Geometric Mean Relative Absolute Error (GMRAE), Median Relative Absolute Error (MdRAE), and Median Absolute Percentage Error (MdAPE) (Armstrong and Collopy 1992; Fildes 1992). Later, the MdRAE, sMAPE (Symmetric Mean Absolute Percentage Error), and sMdAPE (Symmetric Median Absolute Percentage Error) were proposed (Makridakis and Hibon 2000).

Nevertheless, Hyndman and Koehler in their work (Hyndman and Koehler 2006) determined that all measures mentioned before were not generally applicable since they can be infinite or undefined and can produce misleading results. Therefore, they proposed a new measure suitable for all situations: the Mean Absolute Scaled Error (MASE), which is less sensitive to outliers, less variable on small samples, and more easily interpreted.

In Gooijer and Hyndman (2006) and Hyndman and Koehler (2006), a description of different error measures can be found. Among all of them, in this work we use the following:

- Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \mathrm{mean}(\mid p_t \mid) \tag{1}$$

- Median Absolute Percentage Error (MdAPE):

$$\mathrm{MdAPE} = \mathrm{median}(\mid p_t \mid) \tag{2}$$

- Symmetric Median Absolute Percentage Error (sMdAPE):

$$\mathrm{sMdAPE} = \mathrm{median}(200 \mid Y_t - F_t \mid (Y_t + F_t)) \tag{3}$$

- Mean Absolute Scaled Error (MASE):

$$\mathrm{MASE} = \mathrm{mean}(\mid q_t \mid) \tag{4}$$

where Y_t is the observation at time $t = 1, \ldots, n; F_t$ is the forecast of Y_t; e_t is the forecast error (i.e. $e_t = Y_t - F_t$); $p_t = 100 e_t / Y_t$ is the percentage error, and $q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n} |Y_i - Y_{i-1}|}$.

2.1.1 Statistical methods for time series forecasting

Time series forecasting has been a major field of research in the area of statistics (Gooijer and Hyndman 2006) as well as operational research (Fildes et al. 2008). The arrival of the ARIMA methods (Box and Jenkins 1976) in the 1970s established a border line between traditional and modern methods. These latest 40 years have seen the rapid growth of many new methods intended to face the tasks of modeling and/or forecasting time series by means of linear and nonlinear models.

Linear methods have been widely used in order to model time series. Exponential smoothing methods (Brown 1959; Winters 1960) were the standard in the 1950s and 1960s. They provide a useful classification of trend and seasonal patterns depending on whether they could be modeled in an additive or multiplicative way. The best-known methods are simple exponential smoothing (SES), Holt's linear methods, and some variations of the Holt-Winter's methods.

Exponential smoothing methods received a boost thanks to the work of Gardner (1985) and Snyder (1985). A useful variation of the original exponential smoothing methods is SES with drift (Hyndman and Billah 2003), which has been proved to be equivalent to Theta method. Theta method (Assimakopoulos and Nikolopoulos 2000) performed extremely well in the M3-competition, although why its particular choice of model and parameters was so good could not be precisely established (Gooijer and Hyndman 2006).

ARIMA also belongs to the class of linear methods. ARIMA methods integrate autoregressive (AR) and moving average (MA) models in a three-stage iterative cycle consisting of: identification of the time series, estimation of the parameters of the model, and verification of the model. The need for an expert's judgement in order to identify and validate the model is considered the main drawback of this method. For this reason, many techniques and methods have been developed in order to add mathematical rigor to the search process. Theoretically all of them lead to equivalent models, but real application on finite sample data show that there exist large differences in the values provided by these methods for the models' parameters, and these can lead to large differences in the forecasted values.

State space models (Snyder 1985), as well as dynamic linear models (Harrison and Stevens 1976) and structural models (Harvey 1984), have been used since the 1980s by statisticians for time series forecasting. These methods provide a unified framework in which any linear time series model can be written, and in fact, they bear many similarities with exponential smoothing methods. Fildes (1983) concluded that the additional complexity added by state spaces did not lead to better forecasting performance, when comparing these methods with simpler ones, as simpler exponential smoothing.

In the early 1980s it became clear that linear time series forecasting was insufficient in many real applications. Nevertheless, nowadays, the development of nonlinear time series forecasting is just starting, at least compared with linear forecasting. Clements et al. (2004) pointed out current nonlinear method's main problems: they used to develop very complex models; they do not perform in a robust way; and, worst of all, they are difficult to use. Gooijer and Hyndman (2006) also concludes that future research on nonlinear models should include, among others, the search for easy to use software.

Nonlinear models include regime-switching models, an these include the wide variety of existing threshold autoregressive (TAR) models (Tong 1978). TAR models are built using a linear autoregressive model in which parameters change according to the value of a given observable variable. Some of the currently available, most used TAR models are self-exciting (SETAR) models (Tong 1983), in which the threshold variable is a lagged value of the time series; smooth transition (STAR) models (Chan and Tong 1986), that define two different regimes and incorporate a smooth transition between them; and continuous-time (CTAR) models (Brockwell and Hyndman 1992), in which the observable variable is time itself. There exist contradictory papers related to the power of TAR models to forecast time series, as shown by Sarantis (2001) and Bradley and Jansen (2004).

2.1.2 *Soft computing methods for time series forecasting*

Among other soft computing methods, ANNs have been recognized as an important tool for forecasting. Zhang et al. (1998) presented a review of the status in applications of neural networks for forecasting. The popularity of ANNs is derived from the fact they are generalized nonlinear forecasting models. Forecasting has been dominated by linear statistical methods for several decades, as mentioned before, and linear models have many advantages with respect to implementation and interpretation; nevertheless, they have serious limitations because they cannot capture nonlinear associations in the data which are common in many complex real-world problems (Granger and Tersvirta 1993).

The work by Tang et al. (1991) concluded that neural networks not only could provide better long-term forecasting but also did a better job than ARIMA models with short series of input data. Furthermore, contrary to the traditional linear and nonlinear time series models, ANNs are nonlinear data-driven approaches with more flexibility and effectiveness in modeling for forecasting (Zhang et al. 1998). Jain and Kumar (2007) determined in their work that the ANN models were able to produce more accurate forecasts than traditional models because they do not presuppose any functional form of the model to be developed and they do not depend on the assumptions of linearity.

There exist numerous works of different application areas where ANNs are used to forecast. The work by Arizmendi et al. (1993) obtained accurate predictions of the airborne pollen concentrations using ANNs. Zhang and Hu (1998) employed ANNs, and Rivas et al. (2004)

RBFNs, for forecasting British pound and US dollar exchange rates. Bezerianos et al. (1999) employed RBFNs for the assessment and prediction of the heart rate variability. Valenzuela et al. (2008) used a combination of ANNs and other techniques to propose a hybrid ARIMA-ANN model; they used an expert fuzzy logic-based system whose rules were weighted in an automatic way applied to problems like the number of users logged on to an Internet server each minute over 100 min, pollution, or Mackey-Glass chaotic time series, among others.

On the other hand, time series forecasting is faced with some other soft computing approaches. Samanta (2011) and Zhu et al. (2011) developed methods based on cooperative particle swarm optimization. Works like Qiu et al. (2011) and Wang (2011) proposed fuzzy time series models for forecasting, and Yu and Huarng (2010) applied ANNs for training and forecasting in their fuzzy time series model. Models such as support vector regression (Kavaklioglu 2011) and fuzzy expert system (Dash et al. 1995) were proposed for the electricity demand forecasting, among others.

In Sect. 2.1.3, another soft computing approaches for lag selection process are described. Moreover, different soft computing methods for the type of ANNs used in this paper, that is RBFN, are revised in Sect. 2.2.

2.1.3 Lags selection

Another problem that emerges working with time series is the correct choice of the lags considered for representing the series.

The relationship involving time series historical data defines a d-dimensional space where d is the minimum dimension capable of representing such a relationship. Takens' (1980) theorem establishes that if d is sufficiently large is possible to build a state space using the correct time lags and if this space is correctly rebuilt also guarantees that the dynamics of this space is topologically identical to the dynamics of the real systems state space. Many methods can be found in the literature for the correct definition of the variable d, that is, the correct choice of the important time lags of the system dynamics, sometimes called as active dimension of the dynamics generating a time series from the observed series Tanaka et al. (2001).

For this problem an evolutionary method that performs a search for the minimum number of dimensions, Time-delay Added Evolutionary Forecasting (TAEF), is presented in Ferreira et al. (2008). The methodology is inspired in Takens' theorem and consists of an iterative hybrid model composed of an ANN combined with a genetic algorithm (GA).

This method iterates the GA increasing the possible number of lags to obtain a solution with a minimum fitness. Once determined d and the lags, the ANN is tuned in a second stage. In Lukoseviciute and Ragulskis (2010) the evolutionary selection of lags is divided into two stages: first, the optimal dimension of the reconstructed phase space is determined by the false-nearest-neighbor algorithm and then a near-optimal set of time lags is found with a genetic algorithm for a fuzzy inference system.

In general, these proposals are based on the primary dependences among the variables, do not consider any possible induced dependences, and discard any possible correlation that can exist among the time series parameters, even higher order correlations. There are some methods that carry out an automatic search for solving the problem of finding the relevant time lags. QIEHI algorithm (de A. Araújo 2010b), for instance, is a quantum-inspired evolutionary hybrid intelligent method which is composed of an ANN and a modified quantum-inspired evolutionary algorithm to search the minimum dimension to determine the characteristic phase for time series. The model is built in two stages as in Ferreira et al. (2008). Another hybrid methodology composed of a modular morphological neural network with a quantum-inspired evolutionary algorithm that searches for the best time lags is is described in de A. Araújo (2010a). With the same modular morphological neural network, the time lags are obtained by means of a particle swarm optimizer in de A. Araújo (2010c) and by means of a modified GA in de A. Araújo (2011).

In García Pajares et al. (2008) a study on the selection not only of the lags but also of the exogenous features with classical feature selection algorithms as pre-processing stage is performed. The authors show the utility of a feature selection pre-processing stage for time series forecasting with different models.

The lag selection is performed as a postprocessing stage in Maus and Sprott (2011) with a sensitivity computation of the output to each time lag. The initial stage trains a single-layer, feed-forward ANN based on d time lags, with d chosen large enough to capture the relevant dynamics of the time series. TDSEP (Sun et al. 2006) uses a GA for the optimal selection of time lags for a previously obtained and diagonalized second-order correlation matrices.

As can be observed, the approaches of the literature consider the lags selection as a pre- or post-processing or as a part of the learning process but, instead of together, in hybrid processes with two or three stages. On the contrary, our goal is to address the selection of the lags which represent the series (with any type of correlation) jointly with the design process.

2.2 Radial basis function networks

RBFNs are two-layer, fully-connected, feed-forward networks, in which hidden neuron activation functions are Radial Basis Functions (RBFs), usually Gaussian functions. RBFs were introduced by Broomhead and Lowe (1988), being their main applications function approximation and time-series forecasting, as well as classification or clustering tasks.

An RBF can be characterized by a point of the input space, c, and a radius or width, such that the RBF reaches its optimum value (maximun/minimun) when applied to c, and decreases/increases to its opposite optimum value when applied to points far from c. The radius controls how distance affects that increment or decrement. For this reason, experts have used groups of RBFs to successfully interpolate data. In this work we use the Gaussian function, one of the most common RBFs given by Eq. 5.

$$\text{Gaussian} = e^{\frac{-z^2}{2r^2}} \tag{5}$$

where z represents the distance from the point evaluated to the center, which in this case is $x = 0$.

On the whole, the output of an RBFN is given by Eq. 6.

$$s_j(\mathbf{x_k}) = \lambda_{0j} + \sum_{i=1}^{p'} \lambda_{ij}\phi_i(\mathbf{x}, \mathbf{c_i}, \mathbf{r_i}) \tag{6}$$

where $k = 1..p$, $j = 1..n'$, $s_j \in R$, $\mathbf{x_k} \in R^n$, 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 neuron j; $\mathbf{c_i}$ and $\mathbf{r_i}$ 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.

The main problem in RBFNs design concerns establishing the number of hidden neurons to use and their centers and radii. The need for automatic mechanisms to build RBFNs is already present in Broomhead and Lowe (1988)'s work, where they showed that one of the parameters that critically affects the performance of RBFNs is the number of hidden neurons.

One of the advantages of the RBFNs is that once the structure has been fixed, the optimal set of weights linking hidden to output neurons can be analytically computed. For that reason, scientists have applied data mining techniques to the tasks of finding the optimal RBFNs that solves a given problem. Selecting good, or even the best, parameter setting is a very time-consuming task and could be studied as a combinatorial problem. Thus, these type of problems have been faced with EAs (Holland 1975). The main areas where the EAs have been applied to RBFN design are the following ones (Harpham et al. 2004):

- Evolving network architecture. In the RBFN design, the determination of the network architecture implies the obtaining of the number of hidden nodes. This problem is usually addressed with evolutionary proposals together with the RBFN parameters in Pittsburgh approaches (Xue and Watton 1998; Rivas et al. 2007).
- Evolving RBFN parameters (centers, widths and weights of the RBFs). The use of EAs to optimize the connection weights could eliminates the possibility of converging to a local minimum but usually this problem is not addressed with EAs in an independent way but with other parameters of the net (Sheta and Jong 2001). Other approaches evolve only the basis center and width (Dawson et al. 2000).
- Optimizing the dataset. The dimensionality of the learning problem can be drastically reduced by selecting an optimal subset of training data, which is used for training the RBFN (Sergeev et al. 1998). On the other hand, the selection of the most relevant attributes for the RBFN design is not deeply studied in the specialized bibliography (Fu and Wang 2003) and it can be tackled by means of EAs (Perez-Godoy et al. 2008).

In the evolutionary design of RBFN most of the proposals face the different problems by means of hybrid algorithms:

- In a first stage, the EA optimizes the basis centers and widths, as well as the net architecture.
- The second stage uses a supervised learning method in order to obtain the weights.

Harpham et al. (2004) reviewed some of the best-known methods that apply evolutionary algorithms to RBFNs design. They concluded that, in general, methods tend to concentrate in part of the RBFNs components when designing RBFNs, as in Perez-Godoy et al. (2010b) and Rivera et al. (2007). Nevertheless, there also exist methods intended to optimize the whole RBFN, such as Rivas et al. (2007), or other kind of ANNs as Learning Vector Quantization (LVQ) nets (Merelo and Prieto 1995), or multilayer perceptrons (Castillo et al. 2000).

The use of RBFs as activation functions for neural networks and its application to time series forecasting were first considered by Broomhead and Lowe (1988). After these, new works by Carse and Fogarty (1996), and Whitehead and Choate (1996) focused on the prediction of time series.

In later works, Harpham and Dawson (2006) studied the effect of different basis functions on an RBFN for time series prediction. Moreover, Du and Zhang (2008) used time series with an encoding scheme for training RBFNs by GAs. Both the architecture (numbers and selections of nodes and inputs) and the parameters (centers and widths) of the RBFNs are represented in one chromosome and evolved simultaneously by GAs so that the selection of nodes and inputs can be achieved automatically.

Neural network models have been traditionally applied in short-term forecasting (Hippert and Taylor 2010; Lee and Ko 2009). For instance, the work by Perez-Godoy et al. (2010a) applied a hybrid evolutionary cooperative-competitive algorithm for the design of RBFNs to the short-term and even medium-term forecasting of the extra-virgin olive oil price.

Time series prediction problems can be easily decomposed into two subproblems, the selection of the significant lags and the RBFN learning. For this reason, coevolutionary algorithms can be considered a good way to simultaneously solve these problems.

2.3 Coevolutionary algorithms

Cooperative coevolution was introduced by Potter and Jong (2000) and is a research field which has grown in an important way during last years. The Potter's approach consists of identifying the natural decomposition of a problem into subcomponents. A species represents a subcomponent of the potential solution and is assigned to a subpopulation, then each component is evolved isolated from the rest. The fitness of each member of each subpopulation is evaluated by forming collaborations with individuals from other species or populations. The individuals will ultimately be judged on how well they work together to solve the target problem. Finally, a complete solution to the problem is assembled by combining representative members from each of the species.

There are many possible methods for choosing representatives with which to collaborate: random collaboration (Wiegand et al. 2001), best collaboration (Potter and Jong 1994) (the most widely used in the methods of the literature), complete collaboration, and mixed collaboration (Panait et al. 2003). The number of collaborators possibly plays the most important role on the success of a cooperative coevolutionary algorithm and can significantly increase overall computation time, a problem which is combinatorial with respect to the number of subpopulations. Wiegand et al. (2001)'s work suggests that a relatively conservative adjustment from one to two collaborators will frequently yield substantial benefits.

Apart from the collaboration schema and the number of collaborators, another important point to take into account is the collaboration credit assignment method, i.e., the way an individual is being set a fitness when multiple collaborators are selected. There are three common methods: maximum, average, and minimum, although it is always significantly better using maximum method than using minimum or average (Wiegand et al. 2001).

Cooperative coevolution has been employed for tasks like function optimization (Au and Leung 2007), multi-objective evolutionary optimization (Tan et al. 2006), instance selection (García-Pedrajas et al. 2010), and feature selection (Derrac et al. 2010), among others. Cooperative coevolution has also been used in order to train ANNs, such as the cooperative coevolutive approach for designing neural network ensembles (Garcia-Pedrajas et al. 2005) and RBFNs (Li et al. 2008).

It is possible to find coevolution applied to forecasting tasks as in Ma and Wu (2010) where coevolution with immune network, evolving the structure and parametres of the neural network, is applied for predicting short-term load of a city in eastern China. The work by Qian-Li et al. (2008) proposes a coevolutionary recurrent neural network for the multi-step-prediction of chaotic time series estimating the proper parameters of phase space reconstruction and optimizing the structure of recurrent neural networks by coevolutionary strategy.

3 L-Co-R: Lags COevolving with Rbfns

This section describes the method L-Co-R: Lags COevolving with Rbfns. As mentioned above, an algorithm which designs RBFNs for time series forecasting must obtain an appropriate number of RBFs, a radius and a center for every RBF, the weights for the whole network, a suitable set of time lags, and in addition, it should be able to remove the trend of the time series (Zhang and Qi 2005). Our proposal solves the trend problem with an automatic data pre- and post-processing, and the learning of the rest by means of an EA. Since the main goal of the algorithm implies building at the same time both RBFNs and sets of significant lags that will be used to predict future values, L-Co-R is based on a coevolutionary approach. Thus, the main problem can be decomposed into two subproblems which depend on each other.

For this task, L-Co-R simultaneously evolves two populations of different individual species, in which any member of each population can cooperate with the best individual from the other one in every generation, in order to generate good solutions. Therefore, the new algorithm is composed of the following two populations:

- A set of RBFNs which evolves designing the architecture of the net.
- Sets of lags that are used to forecast values of the times series.

In both populations every individual is itself a possible solution to the problem. In the population of lags, an individual represents a set of significant lags, and in the population of RBFNs, a radial basis function network.

The objective of L-Co-R is to forecast any given time series, reducing any hand-made preprocessing step, and building suitable RBFNs designed with appropriate sets of lags, for what it is optimized a quality measure.

In the following subsections it is described the general scheme of the proposal, each process which takes part in the coevolution, the process of collaboration between them, and the trend removal mechanism.

3.1 General scheme

Figure 1 shows the general skeleton of L-Co-R, specifying any of the 17 steps it is composed of.

First, the method performs a preliminary stage of preprocessing which removes the trend of the time series (step 0). Then, the L-Co-R algorithm creates the two initial populations (P_lags and P_RBFNs) and evaluates every individual of each population, as it is explained in Sects. 3.2.2 and 3.3.2, respectively (Step 1).

Once the initial populations have been created, the coevolutionary process starts (Steps 2 and 3). First, the population of lags selects the individuals which are going to take part of the subpopulation (Step 4). HUX crossover (see Sect. 3.3.2) is applied to these individuals of the subpopulation and then, they are evaluated by choosing the collaborators from the population of RBFNs, assigning the result as fitness to the individual that was being evaluated (Steps 5, 6, and 7). Afterwards, parents from population and children from subpopulation are joined in a single and bigger population and they are ranked regarding their fitness (Step 8). Finally, the worst individuals are deleted from this population until it reaches the original size, becoming the new parent population, and eventually the population can be reinitialized (Step 9).

```
Trend preprocessing                                                    (Step 0)
t = 0;
initialize P_lags(t);                                                  (Step 1)
initialize P_RBFNs(t);
evaluate individuals in P_lags(t);
evaluate individuals in P_RBFNs(t);
while termination condition not satisfied do                           (Step 2)
begin
  t = t+1;
  /* Evolve population of lags */
  for i=0 to max_gen_lags do                                           (Step 3)
  begin
    set threshold;
    select P_lags'(t) from P_lags(t);                                  (Step 4)
    apply genetic operators in P_lags'(t);                             (Step 5)
    /* Evaluate P_lags'(t) */
      choose collaborators from P_RBFNs(t);                            (Step 6)
      evaluate individuals in P_lags'(t);                              (Step 7)
    replace individuals from P_lags(t) with P_lags'(t);                (Step 8)
    if threshold < 0
    begin
      diverge P_lags(t);                                               (Step 9)
    end
  end
  /* Evolve population of RBFNs */
  for i=0 to max_gen_RBFNs do                                          (Step 10)
  begin
    select P_RBFNs'(t) from P_RBFNs(t);                                (Step 11)
    apply genetic operators in P_RBFNs'(t);                            (Step 12)
    /* Evaluate P_RBFNs'(t) */
      choose collaborators from P_lags(t);                             (Step 13)
      evaluate individuals in P_RBFNs'(t);                             (Step 14)
    replace individuals from P_RBFNs(t) with P_RBFNs'(t);              (Step 15)
  end
end
train models and select the best one                                   (Step 16)
forecast test values with the final model                              (Step 17)
Trend postprocessing                                                   (Step 18)
```

Fig. 1 General scheme of method L-Co-R

On the other hand, the population of RBFNs begins to evolve when the population of lags has been evolved during a pre-specified number of generations (Step 10). Then, the individuals of the subpopulation are selected, the operators (precisely explained in Sects. 3.2.2 and 3.3.2) are applied, and a collaborator from population of lags is designated in order to establish the fitness of every individual (Steps 11 to 15).

At the end of the coevolutionary process, two models formed by a neural network and a set of lags are obtained. The first one is composed of the best net and its best collaborator, and the second one is formed by the best set of lags and its best collaborator. Next, they are training again and the one with the best fitness will be the final model (Step 16). Then, the forecasted values for the data test are obtained (Step 17), and at this point, the postprocessing phase takes place so that the final test error can be computed (Step 18).

L-Co-R has been implemented following a sequential scheme, so the two populations take turns in evolving. During each generation only one of the two populations is active. Contrary to other algorithms, which at the end of the generation the population that was evolving communicates its best individual to the population that was waiting, in L-Co-R, the collaborator is given only when a member of population needs it.

3.2 Evolution of the population of RBFNs

3.2.1 *Codification*

The population of RBFNs uses a real codification. Every individual is represented by a set of neurons (RBFs) composing the network, as Fig. 2 shows. The number of neurons is variable since it can increase or decrease during the evolutionary process. Every neuron (*a* in the Fig. 2) is defined by a center (*b*) and a radius (*c*). The center (*b*) is a vector with the same dimension as the inputs. The exact dimension of the input space is given by an individual of the population of lags (the one chosen to evaluate the net).

3.2.2 *Evolutionary process*

For the first generation each network in this population is randomly generated, considering that every individual will have a number of neurons chosen at random which may not exceed a maximum number previously fixed only for this first generation. Subsequently, the number of neurons may

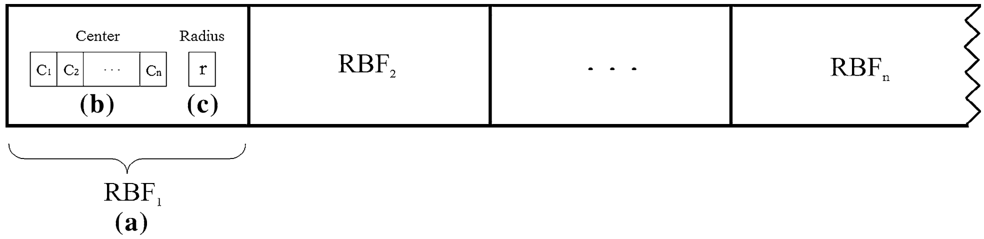

Fig. 2 Example of an individual in the population of RBFNs

be growing or shrinking as the algorithm evolves. The vector of weights is initialized to zero, the center is determined choosing patterns from the training set at random, and the radius is estimated calculating the half of the average distance from centers.

The population of RBFNs incorporates evolutionary operators specifically designed to work with the individuals of this population. Thus, the operators have been designed trying to cover the search space in an effective way, maximizing the success probability.

The operators used for L-Co-R are the following:

- Selection: P_RBFNs population implements tournament selection. Therefore, a group of *TournamentSize* individuals is randomly chosen from the parent population. This group takes part in a *tournament* and a winning individual is determined depending on its fitness value. Finally, the best individual (the winner) is inserted in the subpopulation and the process is repeated to obtain the whole child population.
- X_fix crossover operator: it replaces a sequence of neurons (*a*) in the hidden layer of a network by an equal size sequence of neurons in the hidden layer of other network. To do this, an individual and a number of neurons are randomly selected. Then, the current and random individual exchange as many neurons as the random number indicates. This operator enables sharing information between the networks without affecting the hidden layer size.
- Mutation: there are four operators to mutate the individuals. The choice of one of this mutation operators is carried out randomly, giving to the deleter operator double possibility of being selected.
  - C_random: the application of this operator can modify the point where each RBF of hidden neurons of the net is centered. The number of neurons affected is determined by an internal application factor. The operator performs an exploration of the solution space replacing the center of the neuron (*b*) by a new random center. Each of the components of the new center is chosen following an uniform probability distribution in the range [*min*, *max*]. *Min* and *max* are obtained from input patterns.
  - R_random: in the same way, this operator modifies the radius (*c*) value of hidden neurons. The operator assigns a random value to the radius following an internal probability.
  - Adder: it adds new neurons (*a*) to the hidden layer. The values for the center and radius vectors of a new neuron are randomly set, within the range for each dimension of input space.
  - Deleter: this operator does the opposite of adder operator; it deletes neurons (*a*) from the hidden layer. The exact number of neurons varies from one net to another, since the operator is applied to each neuron with a probability. The deleter operator has a twofold objective. The first one is to reduce the complexity of the network without losing their ability to approximate the training dataset. The second one is to prevent overtraining networks, since a high capacity of generalization is desirable.
- Replacement: the new individuals and the parent ones are joined in an unique population. Then, the worst individuals are eliminated keeping the best ones until the population reaches the original population size. Therefore, the best individuals remain in the next generation.

3.3 Evolution of the population of lags

3.3.1 Codification

The population of lags uses a binary codification scheme where each gene indicates whether the specific lag in the time series will be used to predict the values or not. Figure 3 shows an example of an individual of this population. The length of the chromosome is set at the beginning

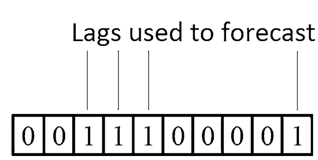

Fig. 3 Example of an individual in the population of lags

corresponding with the specific parameter, so that it cannot vary its size during the execution of the algorithm.

3.3.2 Evolutionary process

The set of lags is evolved by means of CHC (Eshelman 1991) algorithm. Like the population of RBFNs, for the first generation all individuals in this population are randomly generated, taking into account that at least one gene of the chromosome must be set to one, since at least one input has to be given to the net to obtain the forecasted value.

The genetic operators used are the following:

- Selection: in order to select individuals for the child population, the individuals of the parent population are randomly organized to form the current population. Then, they are coupled and the crossover operator will be applied to breed. Since the algorithm uses elitism, the best individuals found up to the moment will remain in the current population.
- Crossover: the HUX crossover operator is used by this population for breeding. It needs two parents; if both parents are not very similar, couples of points are randomly generated and the fragment of the chromosome between them is exchanged, bearing in mind the incest prevention. This application way guarantees the two offspring are always at the maximum Hamming distance from their parents.
- Replacement: the population follows the same process of replacement as described previously. The new individuals and the parent ones are joined in an unique population. Then, the worst individuals are eliminated keeping the best ones until the population reaches the original population size. Therefore, the best individuals remain in the next generation.
- Diverge: when the population is stagnated a restart is produced. The best individual is kept and the rest of the population is generated again in a random way.

3.4 Collaboration process

Since an individual of a species represents only a subcomponent of the problem, collaborations are needed in order to form complete solutions. Then, a collaborator will be selected from each population to assess the fitness according to a collaboration schema.

As mentioned in Sect. 2.3, there are four main collaboration schemas in the literature (Au and Leung 2007):

- Best collaboration (Potter and Jong 1994): choosing as collaborator the best individual from the complementary subpopulation, that is, the individual which fitness value is the best in the previous generations.
- Random collaboration (Wiegand et al. 2001): one or more collaborators are randomly chosen from the other population.
- Complete collaboration (Panait et al. 2003): an individual collaborates with all individuals from the other population.
- Mixed collaboration (Panait et al. 2003): an individual may collaborate with a fixed number of collaborators according to different collaboration schemas, e.g., an individual chooses the best collaborator and some random collaborators. Mixed collaboration might turn into best collaboration, if the number of collaborators is set to 1.

There are three main and common collaboration credit assignment methods used only when multiple collaborators are selected:

- Optimistic: this method consists of assigning to an individual the value of its best collaboration as fitness.
- Hedge: assigning a fitness equal to the average value of its collaborations.
- Pessimistic: assigning as fitness the value of its worst collaboration.

Wiegand et al. (2001) shows that using the optimistic method is always significantly better than using hedge or pessimistic, so that the optimistic is the one we use.

L-Co-R is implemented to use a best collaboration scheme and optimistic approach for credit assignment. More precisely, for every individual in the first population the algorithm chooses the best collaborator of the other population. Exceptionally, at the beginning of the evolutionary process, since the population has not been evaluated, individuals are evaluated by a random collaborator.

Once every individual has selected its collaborator (the best one), the population asks for the collaborator to the other population. Thus, the communication is not produced at the end of a generation, but when a population asks for the specific collaborator it needs. On the other hand, the other population has been keeping the best representative in every generation. So, the individual who is going to be evaluated is coupled with the collaborator and the result obtained is set as its fitness. Fitness function is calculated using the Eq. 7.

$$F = \frac{1}{\sqrt{\frac{1}{n}\sum_{t=0}^{n}(Y_t - F_t)^2}} \quad (7)$$

This fitness is strictly assigned to the individual being evaluated and it is not shared with the representative from the other subpopulation that participates in the collaboration. Figure 4 shows in detail the general process of collaboration and evaluation fitness for the lags individuals. The process is the same for RBFNs.

```
for i = 0 to max_individuals from P_lags do
begin
  c = 0;
  cols(c) = the best collaborator from P_RBFNs;
  for c = 1 to num_collaborators do
  begin
    cols(c) = choose random collaborator from P_RBFNs;
  end
  ask for cols to the P_RBFNs;
  error = 0;
  bestCollaborator = cols(0);
  For c = 1 to cols_size do
  begin
    evaluate i with cols(c);
    if result(i,c) > error
    begin
      error = result(i,c);
      bestCollaborator = c;
    end
  end
  fitness(i) = error;
  bestCollaborator(i) = bestCollaborator;
end
```

Fig. 4 Collaboration process and fitness evaluation for P_lags population. For the P_RBFNs population the process is the same interchanging the roles of the populations

3.5 Process of trend removing

L-Co-R automatically removes the trend of the time series that need it. The process is clearly divided into two phases: pre and postprocessing.

The preprocessing stage takes place before the evolutionary process begins. In this phase, first of all, it is necessary to know whether a time series includes trend or not. For that, a least-square regression line is estimated by the Eq. 8.

$$y = ax + b \tag{8}$$

where a is the slope of the line, and b is the interception of the line with the y-axis. On the other hand, a and b are given by Eqs. 9 and 10, respectively.

$$a = \frac{\sum F_t - b \sum Y_t}{n} \tag{9}$$

$$b = \frac{n \sum Y_t F_t - \sum Y_t \sum F_t}{n \sum Y_t^2 - \left(\sum Y_t\right)^2} \tag{10}$$

Once the slope (a) is known, if it is significant, a preliminary transformation of the data is used to make the transformed data more compatible with the model. The process of data transformation is done by means of differentiation. This involves subtracting each data with the previous one, obtaining a new series.

Once differentiation has been performed, the values $|t_{\mathrm{exp}}|$ and $|t_{\mathrm{teo}}|$ are calculated. $|t_{\mathrm{exp}}|$ is given by the Eq. 11 while $|t_{\mathrm{teo}}|$ is usually approximated to 2.043 value.

$$t_{\mathrm{exp}} = \frac{\beta_1^1 - \beta_1^2}{\frac{S_r}{\sqrt{n} S_x}} \tag{11}$$

where β_1^1 is the slope of the line fitted to the data, β_1^2 is the value to compare the slope (value 0), S_r is the square root of residual variance, n the number of data, and S_x is the square root of x variance.

Since the objective is to detect if the slope is close enough to 0:

- If $|t_{\mathrm{exp}}| > |t_{\mathrm{teo}}|$: the hypothesis that the slope is equal to 0 is rejected and therefore, another differentiation could be made.
- If $|t_{\mathrm{exp}}| \leq |t_{\mathrm{teo}}|$: there is no evidence to say that the slope is not 0, or in other words, the trend has been eliminated.

The postprocessing phase begins at the end of coevolutionary process, when it is obtained the final RBFN+lag model. Then, the differentiation process is undone in order to provide the final forecasting of the test dataset.

4 Experiments

In this section we describe the experiments performed by L-Co-R, the results obtained, a comparison with other six methods, and a statistical study.

The experimental design takes into account the following elements to consider:

- The experimentation has been realized with 34 bases of examples (described in Sect. 4.1). They have different characteristics with respect to number of data, period of time and topic they represent. Most of them are extracted from the Spanish National Statistics Institute.[1]
- We compared the proposed method with other six different methods (described in Sect. 4.3) found in the literature: EvRBF, Fuzzy-WM, NNEP, PolCuadraticLMS, RBFN, and ARIMA. Nevertheless, as described in Sect. 4.3, we have use two different configurations for NNEP and RBFN. For this reason the tables and figures show the results containing two more columns or bars.
- In order to compare with the other methods in the same conditions, they were given the data without trend obtained following the process of trend removing (Sect. 3.5), and then, the postprocessing phase were done to get the final results, like in L-Co-R algorithm.

[1] National Statistics Institute (http://www.ine.es/).

- It has been used four quality measures: MAPE, MdAPE, sMdAPE, and MASE in order show the results obtained (see Sect. 4.4).
- These quality measures have been estimated by means of forecasting 30 times, using the same training and test sets in any execution, and with a horizon of forecasting equal to 1.

4.1 Time series used

In order to test the effectiveness of the L-Co-R method 34 datasets have been used. These time series come from different areas and have different statistical characteristics. Next, a brief description of every one is given:

- Accidents: it represents the number of accidents 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 INE[2] and there are 240 observations.
- AccDeath: it represents the number of deaths on the roads since 1990 until 2007. The observations were extracted from DGT[3] and it is composed of 216 observations.
- AccVictims: this time series represents the number of road accident casualties, since 1990 until 2007. There are 216 observations which were extracted from DGT (see footnote 2).
- 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 Box and Jenkins' *Time series analysis forecasting and control* (Box and Jenkins 1976).
- WorldMarket: is a set of seven time series, any of them representing the monthly values about seven different world markets. The observations were extracted from January 1988 until December 2000. The source of the information is Eurostat. Every time series is composed of 156 observations.
- 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* (Wichern and Jones 1977). Every time series is composed of 276 observations.
- Deceases: this time series represents the number of monthly deceases since 1980 until 1998. There are 228 observations, they were taken from the INE (see footnote 1).
- Gasoline: it represents the finished motor gasoline production (1,000 barrels) from 1993 to 2005. The series is composed of 618 observations expressed daily and it was taken from the NN3-competition.[4]
- Spectators: it corresponds to the number of 1,000 spectators who were in the cinema since 1990 until 2009. The observations are monthly expressed and this time series is composed of 235 observations. The data were taken from the MCU[5] and INE (see footnote 1).
- SpaMovSpec: it represents the number of thousand spectators who were watching a Spanish movie in the cinema from 1990 until 2009. There are 235 observations expressed monthly extracted from the MCU (see footnote 3) and INE (see footnote 1).
- ForMovSpec: this time series represents the number of thousand spectators who were watching a foreign movie in the cinema from 1990 until 2009. It is composed of 235 observations expressed monthly and extracted the MCU[3] and INE (see footnote 1).
- Exchange: this time-series is composed of data representing the exchange rates between British Pound and US Dollar during the period going from 31 December 1979 to 26 December 1983.[6] Data are composed of 208 observations.
- MortCanc: it is the number of canceled mortgages from 2006 to 2009 in 43 observations. The data are taken from the INE (see footnote 1).
- MortMade: it represents the number of made mortgages since 2003 until 2009 in 79 observations. The data are taken from the INE (see footnote 1).
- Books: it is the editorial production of books from 1998 to 2008. The data are taken from the INE[1] and they are composed of 132 observations.
- Motorcycles: this time series represents the manufacture of motorcycles since 1990 until 2009. The data were taken from the INE (see footnote 1), and ANFAC[7] and Mityc.[8]
- Unemployed: it is the number of Spanish unemployed people from 1996 to 2009. It is composed of 164 observations expressed monthly and they were taken from INE (see footnote 1) and MTIN.[9]

[2] National Statistics Institute (http://www.ine.es/).

[3] The General Direction of Traffic (http://www.dgt.es/).

[4] http://www.neural-forecasting-competition.com/NN3/datasets.htm.

[5] The Ministry of Culture (http://www.mcu.es/).

[6] 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.

[7] Spanish Association of Automobile and Truck Manufacturers (http://www.anfac.com/).

[8] The Ministry of Industry, Tourism and Trade (http://www.mityc.es/).

[9] The Ministry of Labour and Immigration (http://www.mtin.es/).

- FreeHousingPrize: it represents the price per m^2 of private housing collected quarterly. The time series is composed of 58 observations from 1995 to 2009 and it was taken from INE (see footnote 1) and MVIV.[10]
- Prisoners: it is the number of prisoners per month since 1990 until 2009. This time series is composed of 233 observations and they were taken from INE (see footnote 1).
- Takings: it represents the average spending per spectator since 1990 until 2009. There are 235 observations expressed in euros and they were extracted from MCU (see footnote 3) and INE (see footnote 1).
- TurIn: it represents the internal air traffic from 1990 to 2009 in 234 observations. The observations were extracted from General Direction of Civil Aviation of the Ministry of Public Works.
- TurOut: it represents the external air traffic from 1990 to 2009 in 234 observations. The observations were extracted from General Direction of Civil Aviation of the Ministry of Public Works.
- TUrban: it is the number of passengers transported by urban transport. The data are taken from the INE (see footnote 1) and they are composed of 164 observations.
- Cars: it represents the vehicle manufacture (cars) from 1990 to August 2009. The data are taken from ANFAC (see footnote 5), Mityc (see footnote 6), and INE (see footnote 1). The data are composed of 236 observations.
- HouseFin: it is the number of finished houses from 1992 to 2009. They are composed of 211 observations extracted from INE (see footnote 1) and Ministry of Public Works.[11]

The time series can be accessed at https://sites.google.com/site/presetemp/datos. For the experimentation, the first 75% of the observations have been considered to form the training data and the other 25% the test, for the 34 datasets.

4.2 Results obtained with L-Co-R

L-Co-R has been applied to the 34 time series with the specific parameter values shown in Table 1.

Table 2 shows the results yielded by L-Co-R method. In this table, from left to right, six kinds of results are shown: Mean Absolute Percentage Error (MAPE), Median Absolute Percentage Error (MdAPE), Symmetric Median Absolute Percentage Error (sMdAPE), and Mean Absolute Scaled Error (MASE) obtained when forecasting the data test, the number of nodes composing the best net found by the method, and the number of lags used by the net to forecast. All these results are the average values calculated after the 30 executions.

Table 1 Parameters used in L-Co-R

Parameter	Value	Description
PopSizeLag	50	Size of P_lags population
MaxGenerationLag	5	Maximum number of generations of P_lags population
MaxLongCrom	10%	Size of chromosome for lags
PopSizeRbfn	50	Size of P_RBFNs population
MaxGenerationsRbfn	10	Maximum number of generations of P_RBFNs population
ValidationRate	0.25	Validation rate
NeuronsRate	0.05	Rate of neurons which indicates the maximum number of neurons in the first generation
TournamentSize	3	Tournament size
ReplacementRate	0.5	Replacement rate
XOverRate	0.8	Crossover rate
MutatorRate	0.2	Mutation rate
MaxGenerations	20	Total number of generations of L-Co-R
Horizon	1	Horizon of the forecasting

As can be seen in the Table 2, the RBFNs built by L-Co-R turn to be non-complex models with respect to the number of hidden neurons, while keeping a high degree of generalization. This good property of the method is achieved without imposing a limit to the number of neurons neither penalizing the complexity when evaluating the nets.

4.3 Comparison with other methods

The performance of the algorithm L-Co-R has been compared with six methods found in the literature. All of them, except ARIMA, are extracted from Keel (Alcalá-Fdez et al. 2009) which is a software tool developed to assess evolutionary algorithms for Data Mining problems. It contains a big collection of classical knowledge extraction algorithms, preprocessing techniques, Computational Intelligence based learning algorithms, including evolutionary rule learning algorithms based on different approaches, and hybrid models such as genetic fuzzy systems, or evolutionary neural networks, among others. The methods considered in this work can be briefly described as follows:

- EvRBF (Evolutionary Radial Basis Function Neural Networks, Rivas et al. 2004). This method is focused on determining the parameters of RBFNs (number of neurons, and their respective centers and radii) automatically. While this task is often done by hand, or

[10] The Ministry of Housing (http://www.mviv.es/).

[11] http://www.fomento.es/.

Table 2 Results of the L-Co-R method

Dataset	MAPE	MdAPE	sMdAPE	MASE	Nodes	Average lags
Accidents	0.432	1.905	7.743	0.909	8.07	7.23
AccDeath	3.768	0.915	10.224	1.023	6.60	7.43
AccVictims	0.117	0.293	5.417	1.029	7.30	6.27
Airline	19.281	1.333	10.553	0.987	4.73	2.57
WmFrancfort	0.540	2.319	5.064	1.248	2.43	5.17
WmLondres	0.078	0.286	3.515	1.321	2.17	5.70
WmMadrid	0.309	0.521	3.699	1.300	2.40	5.80
WmMilan	1.223	0.546	4.572	1.182	2.40	5.37
WmNuevayork	0.094	0.520	3.096	1.231	2.93	4.90
WmParis	1.364	2.048	4.065	1.103	2.57	5.60
WmTokio	0.006	1.457	4.537	1.518	2.60	5.53
Colgtems	1.505	1.838	12.966	0.970	5.73	9.60
Colgtepr	0.254	0.115	2.152	0.858	8.50	7.43
Crestms	1.016	0.546	9.810	0.991	4.47	9.07
Crestpr	0.012	0.480	3.373	0.980	4.37	10.07
Deceases	0.057	0.033	7.409	0.999	7.10	7.20
Gasoline	4.292	2.806	15.949	0.977	13.67	22.43
Spectators	25.663	9.287	40.860	0.958	7.03	7.17
SpaMovSpec	3.698	0.027	17.766	0.910	2.63	9.67
ForMovSpec	0.323	0.013	0.932	1.145	6.20	7.63
Exchange	0.129	0.001	1.618	0.409	2.23	4.30
MortCanc	0.507	0.020	8.286	0.942	1.93	1.77
MortMade	1.218	0.586	10.533	1.133	1.93	1.77
Books	6.634	1.615	16.829	1.024	2.40	4.23
Motorcycles	86.173	4.604	25.549	1.025	3.03	8.53
Unemployed	9.206	3.318	3.579	3.224	10.50	3.40
FreeHousingPrize	1.415	1.088	1.102	0.873	2.40	1.20
Prisoners	0.214	0.192	0.371	0.462	3.67	7.77
Takings	3.064	2.512	15.146	0.965	5.83	7.03
TurIn	0.533	1.055	3.231	1.056	7.07	6.53
TurOut	1.140	0.693	4.541	0.999	8.33	7.57
TUrban	1.825	2.560	9.046	0.993	6.27	5.30
Cars	18.661	1.122	26.023	1.250	11.03	7.77
HouseFin	5.636	3.331	13.223	1.054	6.03	6.03

based in hill-climbing methods which are highly dependent on initial values, in this method, evolutionary algorithms are used to automatically build an RBFN that solves a specified problem.

- Fuzzy-WM (Fuzzy Rule Learning, Wang–Mendel Algorithm, Wang and Mendel 2002). This is a general method to generate fuzzy rules from numerical data. This method can be used as a general way to combine both numerical and linguistic information into a common framework, a fuzzy rule base. This fuzzy rule base consists of two kinds of fuzzy rules: some obtained form experts, and others generated from measured numerical data. It is proved that the generated fuzzy system is able to approximate any nonlinear continuous function on a compact set to arbitrary accuracy.
- NNEP (Neural Network Evolutionary Programming, Martínez-Estudillo et al. 2006). This method is based on the evolution of a type of feed-forward neural networks whose basis function units are products of the inputs raised to real number power. These nodes are usually called product units and the main advantage of product units is their capacity to implement higher order functions. Thus, NNEP evolves weights and structure of product unit-based neural networks and it has been applied to the regression function problem.
- PolCuadraticLMS (LMS Quadratic Regression) (Rustagi 1994). This deterministic model uses the quadratic

Table 3 Parameters used by the methods

Method	Parameter	Value
EvRBF	Population size	100
	Generations	10
	Validation rate	0.25
	Neurons rate	0.1
	Tournament size	30
	Replacement rate	0.75
	Crossover rate	0.9
	Mutator rate	0.1
NNEP	Number of neurons in hidden layer	4
	Transfer function in each neuron	Product_Unit
	Number of generations	1,000
PolCuadraticLMS	–	–
RBFN	Number of hidden neurons	50
Fuzzy-WM	Number of labels	5
	KB Output File Format with Weight values to 1?	0
NNEP2	Number of neurons in hidden layer	Depends on the dataset
	Transfer function in each neuron	Product_Unit
	Number of generations	1,000
RBFN2	Number of hidden neurons	Depends on the dataset

regression to find values from other known. This type of regression offers a higher adjustment of the points than the lineal regression. The approach is performed using the parameters of a parabola, which is given by the Eq. 12:

$$y = ax^2 + bx + c \tag{12}$$

- RBFN. This method refers to the simple RBFN, as described in Broomhead and Lowe (1988).
- ARIMA. The models proposed by G.E. Box and G.M. Jenkins and better known as Box–Jenkins models (Box and Jenkins 1976).

The NNEP2 and RBFN2 columns derive from a specific adaptation of NNEP and RBFN methods, respectively. They are the result of a study about the complexity of the nets found by L-Co-R. Once the study was made, the average number of neurons was used as parameter for the algorithms. Then, NNEP and RBFN are equaled to L-Co-R having the same initial complexity of the networks, which is resulting in NNEP2 and RBFN2. Finally, NNEP2 and RBFN2 have been executed with the nearest integer average number to the number of neurons obtained by L-Co-R with every dataset. These average numbers can be seen in Table 2.

Table 4 ARIMA models obtained for every dataset

Dataset	Model
Accidents	ARIMA (3, 1, 2) with drift
AccDeath	ARIMA (1, 1, 2) with drift
AccVictims	ARIMA (2, 1, 2)
Airline	ARIMA (4, 1, 2)
WmFrancfort	ARIMA (1, 2, 2)
WmLondres	ARIMA (0, 1, 0) with drift
WmMadrid	ARIMA (1, 2, 2)
WmMilan	ARIMA (0, 1, 0)
WmNuevayork	ARIMA (1, 2, 1)
WmParis	ARIMA (1, 1, 2) with drift
WmTokio	ARIMA (0, 1, 0)
Colgtems	ARIMA (0, 1, 1)
Colgtepr	ARIMA (2, 1, 1) with drift
Crestms	ARIMA (0, 1, 1)
Crestpr	ARIMA (0, 1, 1)
Deceases	ARIMA (0, 1, 0)
Spectators	ARIMA (2, 1, 2) with drift
SpaMovSpec	ARIMA (0, 1, 3)
ForMovSpec	ARIMA (0, 1, 2) with drift
Exchange	ARIMA (2, 1, 0) with drift
Gasoline	ARIMA (2, 1, 2)
MortCanc	ARIMA (0, 1, 1) with drift
MortMade	ARIMA (3, 1, 3) with drift
Books	ARIMA (0, 1, 1)
Motorcycles	ARIMA (3, 1, 2)
Unemployed	ARIMA (2, 1, 2) with drift
FreeHousingPrize	ARIMA (3, 2, 1)
Prisoners	ARIMA (0, 1, 2) with drift
Takings	ARIMA (2, 1, 1) with drift
TurIn	ARIMA (5, 1, 2) with drift
TurOut	ARIMA (3, 1, 5) with drift
TUrban	ARIMA (0, 1, 2) with drift
Cars	ARIMA (0, 1, 1) with drift
HouseFin	ARIMA (2, 1, 2) with drift

Every method has been executed using the default parameters, that is, those which have been considered suitable by the authors of the algorithms. These parameters are shown in Table 3 for every method. After that, Table 4 shows the ARIMA models obtained for every dataset.

The Estimated Partial Autocorrelation Function (EPAF) has been used to work with the eight methods mentioned before. It indicates which intervals of time from datasets are considered more important to be taken into account when patterns of data are going to be formed. One of the main advantages of L-Co-R is that it is not necessary to

Table 5 Lags selected by EPAF and L-Co-R for each dataset

Dataset	EPAF	L-Co-R
Accidents	1,3,4,9,11,12,13,15,25,57	1,2,3,7,9,12
AccDeath	1,7,8,10,11,12,13,15,23,40,44	1,4,6,8,12
AccVictims	1,7,8,9,12,13,18,22,25,26,27,49	1,3,6,12
Airline	1,4,11,12,13,23	1,2,4
WmFrancfort	1,3,30	1,3,6,10
WmLondres	1	4,8,10
WmMadrid	1,12,30	1,3,4,5,6,9,10
WmMilan	1,30	2,8,10
WmNuevayork	1,30	3,6,7,10,11
WmParis	1,30	1,2,4,5,7,8,10,11
WmTokio	1,6	5,7,9,10,11
Colgtems	1,2,3,5,30	1,2,4,6,9,10,12,15,16
Colgtepr	1,2,3,4	1,2,4,12,15,17
Crestms	1,2,3	1,2,3,4,7,10
Crestpr	1,2,3,4,5	1,4,7,9,11,12,13,14,19,20
Deceases	1,3,9,10,11,12,13,23,24,25,37	1,3,4,5,7,8,9,12,13
Spectators	1,3,7,9,11,12,13,14,23,24,26,29,36	4,6,8,10,11,12,14,15,17
SpaMovSpec	1,5,10,11,24,50	1,3,4,6,9,11,13,14,15,17
ForMovSpec	1,3,4,5,7,11,12,13,24,29,39	1,2,4,6,8,9,10,12,13
Exchange	1	1,3,5,6,7
Gasoline	1,2,3,4,5,6,7,8,14,27	1,3,5,13,14,16,17,19,24,25,30,31,33,42
MortCanc	1	1,2,3
MortMade	1,2,4,6,7,8	1,3,4
Books	1,12	3,5,6,8,9
Motorcycles	1,2,4,5,6,7,8,10,11,12,13,14,25,48	1,4,8,9,10,11,12,15
Unemployed	1	1,8
FreeHousingPrize	1	4
Prisoners	1	2,4,6,12,13,14,15,17
Takings	1,2,3,6,7,8,9,11,12,13,14,23,25,26,31	1,2,3,5,9,10,12
TurIn	1,2,6,7,8,9,13,25	1,2,6,11,12,15,17
TurOut	1,3,5,6,7,9,13,16,21	1,2,3,4,7,8,10,12
TUrban	1,3,4,6,9,11,12,13,14,15	1,4,5,7,8,9,12
Cars	1,4,8,9,12,13,14,15,22	1,5,6,8,12,16
HouseFin	1,2,3,4,8,11,12,13,14,15,17,22,35	3,4,6,7,8,9,12,14

apply any a priori preprocessing in this sense, since the algorithm is able to automatically find the most suitable lags during the evolution of the algorithm by itself. So, a previous study of the significant lags was made to test every method used to compare. Table 5 shows a comparison between lags selected by the EPAF and the selected ones by L-Co-R. As can be seen, L-Co-R selects less lags than EPAF in 15 from the 34 datasets, more lags in 18 cases, and for 1 dataset the number of lags is the same in both methods.

Tables 6, 7, 8, and 9 show the results yielded by the methods EvRBF, NNEP, PolCuadraticsLMS, RBFN, Fuzzy-WM, NNEP2, RBFN2, and ARIMA compared to L-Co-R. They show the MAPE, MdAPE, sMdAPE, and MASE, respectively, yielded when forecasting the test data by every method. As can be seen in the Tables 6, 7, 8, and 9, L-Co-R yields good results in general with regard to all error measures. Best results, that is, lower errors, are emphasized in bold.

L-Co-R obtains the best result in 31 of the 34 datasets tested for MAPE, in 33 of 34 for MdAPE, in 21 of 34 for sMdAPE, and in 19 of the 34 for MASE.

4.4 Statistical study and conclusions of the experimentation

A statistical study has been done in order to check if the differences among methods are significant for each one of

Table 6 Comparison among the methods with respect to MAPE

Dataset	L-Co-R	EvRBF	NNEP	Fuzzy-WM	PolCuadLMS	RBFN	NNEP2	RBFN2	ARIMA
Accidents	**0.432**	300.736	538.062	43.177	56.398	35.416	510.974	35.865	15.341
AccDeath	**3.768**	846.347	519.683	606.858	432.049	335.364	450.568	356.938	19.393
AccVictims	**0.117**	423.550	891.292	214.475	238.433	100.048	788.280	100.589	9.935
Airline	19.281	**4.175**	5.667	18.178	7.770	10.070	5.800	10.270	53.636
WmFrancfort	**0.540**	50.235	115.114	24.925	26.578	26.911	162.299	26.917	12.136
WmLondres	**0.078**	34.552	14.238	6.811	7.394	7.466	14.306	7.296	5.212
WmMadrid	**0.309**	42.431	1.012E+18	19.893	21.027	21.062	3.917E+16	21.444	12.930
WmMilan	**1.223**	560.832	198.560	38.706	40.261	40.204	144.676	40.379	34.823
WmNuevayork	**0.094**	62.291	33.615	22.256	22.532	22.427	31.572	22.710	7.536
WmParis	**1.364**	56.778	46.800	42.211	42.137	41.993	45.500	41.772	25.880
WmTokio	**0.006**	466.303	35.085	27.058	34.530	33.659	35.107	35.509	12.591
Colgtems	**1.505**	241.615	122.352	118.746	125.129	125.601	121.616	119.684	18.865
Colgtepr	**0.254**	299.596	20.243	21.702	21.651	21.565	21.864	21.193	7.358
Crestms	**1.016**	174.082	10.151	12.084	9.569	10.628	10.016	11.137	13.688
Crestpr	**0.012**	49.625	3.789	3.969	3.572	4.039	3.964	3.815	4.262
Deceases	**0.057**	124.988	4,171.248	85.826	94.629	44.201	7,134.274	46.034	8.040
Spectators	**4.292**	761.847	330.549	463.646	281.150	338.340	334.979	333.512	40.240
SpaMovSpec	**25.663**	464.728	8.309E+08	1320.780	804.564	566.631	2.992E+09	534.917	88.197
ForMovSpec	**3.698**	713.870	656.366	828.488	708.733	711.377	744.107	727.076	41.054
Exchange	**0.323**	31.625	22.161	47.044	47.078	47.148	22.382	47.116	45.254
Gasoline	**0.129**	45.528	65.298	35.932	49.630	47.231	490.115	47.010	9.359
MortCanc	**0.507**	23,583.926	50,982.436	15,229.556	22,601.441	21,041.056	38,664.348	22,496.528	5.440
MortMade	**1.218**	107.899	89.851	116.751	115.773	93.734	89.663	89.876	31.000
Books	**6.634**	239.405	49.294	30.827	31.782	30.788	42.025	29.892	23.476
Motorcycles	86.173	205.594	858.605	51.876	256.689	40.211	668.072	**40.022**	89.111
Unemployed	9.206	**2.434**	10.916	9.496	5.336	5.965	10.760	9.200	18.726
FreeHousingPrize	**1.415**	27.389	4.930	3.850	3.512	3.544	4.702	3.617	10.227
Prisoners	**0.214**	70.011	57.038	47.328	48.004	48.422	58.481	49.319	3.150
Takings	**3.064**	516.609	98.938	144.502	121.822	67.581	84.574	64.239	28.456
TurIn	**0.533**	34.375	8.095	5.343	5.435	4.756	9.202	4.696	6.377
TurOut	**1.140**	54.858	5.079	6.651	6.027	5.782	15.812	5.136	9.634
TUrban	**1.825**	601.135	1.076E+04	107.623	108.978	45.094	5.666E+04	49.427	9.291
Cars	**18.661**	756.159	668.864	1350.170	626.907	579.962	942.617	564.535	45.940
HouseFin	**5.636**	106.733	2,316.286	80.554	107.796	47.899	3,026.778	44.603	19.555

Table 7 Comparison among the methods with respect to MdAPE

Dataset	L-Co-R	EvRBF	NNEP	Fuzzy-WM	PolCuadLMS	RBFN	NNEP2	RBFN2	ARIMA
Accidents	**1.905**	47.412	36.375	36.243	44.996	33.954	38.149	35.492	13.683
AccDeath	**0.915**	98.673	190.886	266.875	221.828	186.757	199.549	184.601	16.380
AccVictims	**0.293**	90.021	31.935	138.363	120.674	33.935	37.160	36.344	9.504
Airline	**1.333**	16.750	3.524	15.802	6.224	7.941	3.834	9.103	15.212
WmFrancfort	**2.319**	26.487	30.587	24.032	25.142	26.494	30.718	26.721	11.026
WmLondres	**0.286**	16.598	14.150	6.395	6.834	7.035	14.378	7.043	5.099
WmMadrid	**0.521**	20.763	22.195	19.046	20.638	20.771	22.498	20.782	11.446
WmMilan	**0.546**	41.636	40.131	37.717	39.213	39.073	39.453	39.487	34.643
WmNuevayork	**0.520**	28.271	24.030	22.539	22.034	22.345	24.341	22.838	5.712
WmParis	**2.048**	31.337	40.691	41.657	40.450	40.328	40.784	40.369	23.773
WmTokio	**1.457**	53.528	34.135	25.949	33.329	32.644	34.238	34.392	9.556
Colgtems	**1.838**	105.204	107.239	104.643	108.514	108.867	107.077	105.630	13.806
Colgtepr	**0.115**	35.676	20.120	20.730	21.254	20.835	20.001	20.587	6.364
Crestms	**0.546**	32.681	7.318	10.865	7.988	8.767	7.760	9.347	12.697
Crestpr	**0.480**	11.255	2.921	3.235	2.458	3.192	2.914	2.979	3.414
Deceases	**0.033**	71.158	32.146	65.168	51.209	24.012	40.327	25.993	5.458
Spectators	**2.806**	170.139	189.701	244.679	181.648	200.650	194.851	191.784	39.955
SpaMovSpec	**9.287**	104.833	113.331	345.402	143.447	112.690	169.294	113.692	54.033
ForMovSpec	**0.027**	101.443	278.349	434.823	272.276	291.237	284.306	312.477	32.322
Exchange	**0.013**	16.422	17.009	45.882	45.677	45.799	17.008	45.770	45.961
Gasoline	**0.001**	14.589	48.975	47.156	50.773	48.895	49.424	48.170	8.923
MortCanc	**0.020**	84.455	19.699	25.595	11.564	10.092	19.336	12.039	5.116
MortMade	**0.586**	17.668	19.650	24.128	15.838	13.239	18.006	23.200	28.374
Books	**1.615**	110.920	21.143	17.446	20.432	22.652	20.336	19.216	18.093
Motorcycles	**4.604**	104.070	37.822	30.900	157.470	36.581	42.096	36.201	43.058
Unemployed	3.318	8.489	2.331	8.023	**2.308**	2.669	2.421	4.945	10.698
FreeHousingPrize	**1.088**	6.610	4.071	4.145	3.657	3.735	4.027	3.831	6.572
Prisoners	**0.192**	19.603	25.763	20.521	21.552	26.298	25.769	26.952	1.621
Takings	**2.512**	109.377	55.852	135.472	93.241	59.132	57.722	55.492	24.928
TurIn	**1.055**	10.668	6.919	4.524	4.902	4.351	6.816	4.415	4.605
TurOut	**0.693**	12.715	3.624	3.984	4.443	4.232	4.078	3.477	7.689
TUrban	**2.560**	129.223	66.730	74.341	57.560	19.885	127.745	22.819	6.374
Cars	**1.122**	103.049	278.362	528.130	283.314	282.586	295.020	272.297	14.808
HouseFin	**3.331**	35.493	14.757	41.579	40.426	18.632	18.461	17.816	17.297

the quality measures considered, MAPE, MdAPE, sMdAPE, and MASE.

The use of parametric statistical techniques over the sample of results is only adequate when they fulfill three necessary conditions: independency, normality, and homoscedasticity (Sheskin 2006; Zar 1999). With respect to normality condition, we applied Shapiro–Wilk test as it is used in work by García et al. (2009). This test confirmed that the condition was not fulfilled; therefore, a non-parametric test should be used.

Then, in order to test whether significant differences exist among all methods, Friedman and Iman–Davenport tests have been applied. Tables 10 and 11 show the results: the Friedman and Iman–Davenport values (χ^2 and F_F, respectively), the corresponding critical values for each distribution by using a level of significance $\alpha = 0.05$, and the p value obtained for all measures used. The statistics of Friedman and Iman–Davenport are clearly greater than their associated critical values, so it can be concluded that there are significant differences among the observed results with a level of significance $\alpha \leq 0.05$, in all cases. According to these results, a post-hoc statistical analysis is needed.

Tables 12 and 13, and graphically Figs. 5, 6, 7, and 8, show a ranking of the methods obtained by Friedman

Table 8 Comparison among the methods with respect to sMdAPE

Dataset	L-Co-R	EvRBF	NNEP	Fuzzy-WM	PolCuadLMS	RBFN	NNEP2	RBFN2	ARIMA
Accidents	**7.743**	33.318	40.987	38.315	45.334	40.577	41.318	42.910	14.688
AccDeath	10.224	**−115.993**	101.641	114.807	83.938	97.627	102.772	97.126	15.140
AccVictims	**5.417**	25.769	24.763	85.137	60.897	29.364	28.545	30.836	9.073
Airline	10.553	17.372	**3.484**	14.640	6.109	7.815	3.796	8.822	16.297
WmFrancfort	**5.064**	30.531	36.027	27.314	28.756	30.541	36.151	30.843	11.308
WmLondres	**3.515**	18.101	15.227	6.606	6.976	6.983	15.493	7.050	5.233
WmMadrid	**3.699**	23.069	24.275	21.050	23.013	23.180	24.754	23.193	10.827
WmMilan	**4.572**	50.803	48.057	46.482	48.777	48.562	48.175	49.202	41.900
WmNuevayork	**3.096**	32.925	27.202	25.402	24.762	25.157	27.627	25.785	5.553
WmParis	**4.065**	37.160	50.854	52.617	50.705	50.515	50.776	50.578	26.980
WmTokio	**4.537**	35.434	29.158	22.969	28.568	28.046	29.233	29.325	9.120
Colgtems	**12.966**	13.273	69.800	68.699	70.346	70.182	69.733	68.528	13.315
Colgtepr	**2.152**	16.305	18.281	18.783	19.212	18.863	18.182	18.661	6.168
Crestms	9.810	**2.781**	7.527	10.437	8.320	8.976	7.993	9.586	13.557
Crestpr	3.373	10.664	2.899	3.184	**2.429**	3.172	2.905	2.977	3.474
Deceases	7.409	93.774	33.424	51.428	42.748	27.195	34.306	29.752	**5.453**
Spectators	**15.949**	84.231	97.391	110.043	95.185	100.121	98.457	97.811	33.302
SpaMovSpec	40.860	**12.974**	67.632	129.367	90.029	81.827	74.277	82.023	59.943
ForMovSpec	**17.766**	107.665	116.614	136.960	115.299	118.502	116.155	121.792	27.822
Exchange	**0.932**	17.892	18.590	59.541	59.197	59.401	18.589	59.353	59.674
Gasoline	**1.618**	15.374	64.703	61.705	67.624	64.838	64.732	63.551	9.036
MortCanc	8.286	96.567	16.505	29.354	11.218	9.935	17.349	11.796	**5.256**
MortMade	**10.533**	17.518	21.203	27.439	15.876	13.871	19.433	27.224	24.849
Books	16.829	**8.183**	19.584	17.440	18.751	22.199	19.002	19.356	19.574
Motorcycles	25.549	**14.159**	37.437	28.024	47.660	44.287	36.474	43.379	50.360
Unemployed	3.579	8.429	2.338	8.031	**2.305**	2.685	2.432	4.874	11.340
FreeHousingPrize	**1.102**	6.601	4.154	4.233	3.725	3.807	4.107	3.906	6.362
Prisoners	**0.371**	21.616	29.460	22.867	24.155	30.310	29.460	31.216	1.634
Takings	**15.146**	52.149	43.116	80.763	68.738	45.604	44.487	43.387	22.165
TurIn	**3.231**	10.331	6.687	4.424	4.785	4.258	6.578	4.324	4.714
TurOut	4.541	12.007	3.583	3.985	4.434	4.143	4.012	**3.425**	7.600
TUrban	9.046	19.083	27.125	54.196	51.593	18.860	41.279	21.445	**6.584**
Cars	26.023	**−180.476**	143.135	175.122	132.439	147.887	142.626	144.447	13.784
HouseFin	**13.223**	42.745	13.771	39.028	44.051	20.366	15.925	19.193	17.278

method. The best method is stressed in bold for every measure.

As can be seen in Tables 12 and 13, the new method L-Co-R achieves the best ranking with a result that is lower than the rest for all measures, so it is taken as the control algorithm.

After this, a post-hoc test is used to find whether the control algorithm presents statistical differences with regard to the remaining methods in the comparison. As it is recommend in García et al. (2009), we apply the Holm (1979) procedure. Tables 14 and 15 show all the adjusted p values for each comparison which involves the control algorithm, for MAPE, MdAPE, sMdAPE, and MASE, respectively. The p value is indicated in each comparison considering a level of significance $\alpha = 0.05$.

As it is shown in Tables 14 and 15, there are significant differences between L-Co-R and the remaining methods for all measures used. Therefore, we can conclude that the new algorithm really shows a better behavior with respect to test error comparing to other methods. Even with the methods NNEP2 and RBFN2, in which the complexity of the initial networks are the same than in L-Co-R, the new algorithm yielded better results with significant differences.

Taking everything into account, L-Co-R stands out for its accurate over a large set of sample data, which has

Table 9 Comparison among the methods with respect to MASE

Dataset	L-Co-R	EvRBF	NNEP	Fuzzy-WM	PolCuadLMS	RBFN	NNEP2	RBFN2	ARIMA
Accidents	0.909	34.960	79.328	**0.739**	3.081	5.029	65.152	5.086	2.237
AccDeath	1.023	136.922	4.380	8.183	4.517	5.803	5.153	5.833	**0.972**
AccVictims	**1.029**	30.513	69.999	13.270	3.607	4.651	25.986	4.913	1.578
Airline	0.987	0.191	**0.092**	0.870	0.365	0.292	0.103	0.315	1.441
WmFrancfort	**1.248**	14.313	32.036	6.414	6.913	7.005	45.540	7.011	7.988
WmLondres	**1.321**	13.958	4.927	1.707	2.039	2.111	4.950	1.982	3.484
WmMadrid	**1.300**	2.863	3.066E+17	5.081	5.387	5.394	1.187E+16	5.491	8.625
WmMilan	**1.182**	11.545	39.441	10.885	11.294	11.283	24.751	11.338	19.327
WmNuevayork	**1.231**	29.834	9.139	7.258	7.367	7.342	8.623	7.436	6.228
WmParis	**1.103**	19.547	8.980	14.759	14.778	14.728	9.228	14.652	15.744
WmTokio	**1.518**	29.756	5.194	3.993	5.108	4.984	5.198	5.254	1.628
Colgtems	0.970	70.262	2.986	2.893	3.061	3.064	2.977	2.899	**0.869**
Colgtepr	**0.858**	101.366	5.726	6.144	6.127	6.100	6.139	5.996	2.307
Crestms	0.991	65.133	0.561	0.631	**0.534**	0.548	0.554	0.660	1.418
Crestpr	0.980	26.111	0.553	0.488	0.548	0.583	0.514	**0.486**	1.330
Deceases	**0.999**	84.219	460.625	6.838	4.162	2.238	998.835	2.300	1.144
Spectators	**0.977**	9.684	3.817	5.137	3.663	3.849	3.838	3.794	1.831
SpaMovSpec	0.958	17.614	6.244E+06	2.959	1.470	0.886	2.291E+07	**0.806**	1.933
ForMovSpec	**0.910**	1.374	3.550	4.672	3.931	3.769	2.917	3.849	1.626
Exchange	**1.145**	31.420	15.009	54.223	54.266	54.346	14.739	54.309	70.734
Gasoline	**0.409**	21.430	16.742	12.722	17.974	17.166	101.233	17.059	1.698
MortCanc	0.942	7.834	7.926	0.387	3.062	2.508	8.590	2.669	**0.277**
MortMade	1.133	0.874	0.703	1.075	1.753	**0.269**	0.600	1.570	1.712
Books	1.024	40.588	1.055	0.822	0.508	**0.385**	0.596	0.654	1.147
Motorcycles	**1.025**	21.548	28.792	1.123	4.316	1.242	19.197	1.215	2.670
Unemployed	3.224	5.774	6.191	**1.886**	2.680	3.005	6.087	3.899	15.809
FreeHousingPrize	**0.873**	8.559	1.951	2.397	2.382	2.292	1.885	2.377	6.805
Prisoners	**0.462**	72.334	22.037	33.725	22.131	39.834	24.705	35.566	4.031
Takings	**0.965**	8.315	2.184	4.842	1.676	2.254	2.257	2.142	1.978
TurIn	**1.056**	8.825	1.974	1.180	1.241	1.132	1.865	1.089	1.950
TurOut	0.999	19.311	**0.586**	0.798	0.758	0.792	1.944	0.672	2.241
TUrban	0.993	58.326	1,565.053	10.555	**0.241**	1.597	8,465.803	2.343	0.897
Cars	1.250	77.501	3.260	6.620	3.186	3.409	3.637	3.420	**1.048**
HouseFin	1.054	8.390	155.469	2.147	1.678	**0.216**	189.479	0.225	1.502

Table 10 Results of the Friedman test (α = 0.05)

Measure	Friedman value	Value in χ^2	p value
MAPE	122.925	8	1.010E−10
MdAPE	101.051	8	6.431E−11
sMdAPE	58.168	8	1.104E−9
MASE	68.996	8	7.000E−11

Table 11 Results of the Iman–Davenport test (α = 0.05)

Measure	Iman–Davenport value	Value in F_F	p value
MAPE	27.212	8 and 264	1.278E−30
MdAPE	19.507	8 and 264	5.070E−23
sMdAPE	8.976	8 and 264	6.875E−11
MASE	11.216	8 and 264	1.181E−13

different characteristics and nature; for instance, *AccDeath* describes the number of deaths on the roads monthly, whereas *FreeHousingPrize* represents the price per m^2 of private housing collected quarterly. Although L-Co-R considers in fitness function neither the simplicity of the networks nor the size of the set of lags, it designs simple and appropriate RBFNs and it is able to automatically determine a suitable set of lags for the specified network, finding small sets of lags that can be smaller than the ones found by EPAF. In addition, L-Co-R includes an automatic process to remove the trend of the time series with which it works.

Table 12 Average Rankings of the algorithms (Friedman) for MAPE and MdAPE. The method on the top of the table corresponds to the best classified

MAPE		MdAPE	
Method	Ranking	Method	Ranking
L-Co-R	**1.412**	**L-Co-R**	**1.118**
ARIMA	3.382	ARIMA	3.765
RBFN2	4.529	RBFN	5.088
RBFN	4.558	RBFN2	5.265
PolCuadLMS	5.118	NNEP	5.382
Fuzzy-WM	5.324	PolCuadLMS	5.500
NNEP	6.441	NNEP2	6.059
NNEP2	6.971	Fuzzy-WM	6.176
EvRBF	7.265	EvRBF	6.647

Table 13 Average Rankings of the algorithms (Friedman) for sMdAPE and MASE. The method on the top of the table corresponds to the best classified

sMdAPE		MASE	
Method	Ranking	Method	Ranking
L-Co-R	**2.059**	**L-Co-R**	**2.441**
ARIMA	3.971	RBFN	4.441
EvRBF	5.147	PolCuadLMS	4.529
NNEP	5.206	RBFN2	4.706
PolCuadLMS	5.441	ARIMA	4.735
RBFN	5.500	Fuzzy-WM	4.794
NNEP2	5.618	NNEP	5.794
RBFN2	5.974	NNEP2	6.147
Fuzzy-WM	6.265	EvRBF	7.441

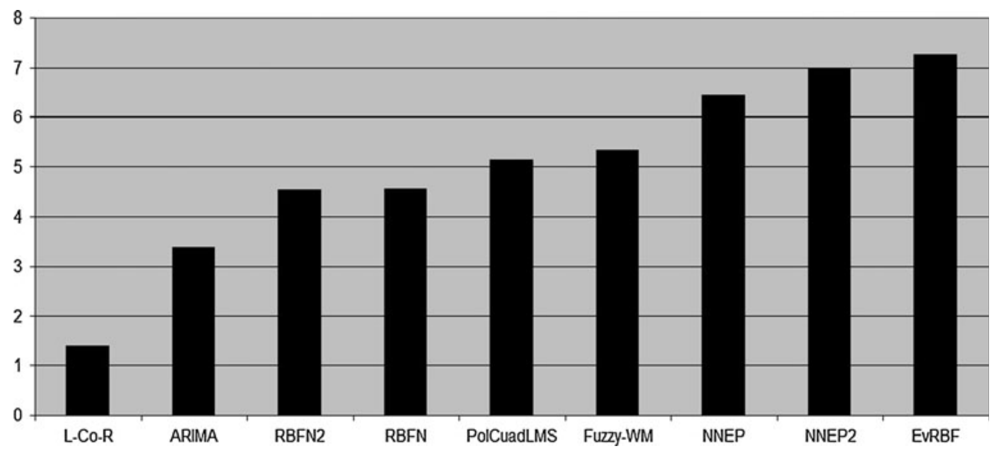

Fig. 5 Graphic of the ranking of the methods for MAPE. *Lower values* represent better predictions of the test dataset

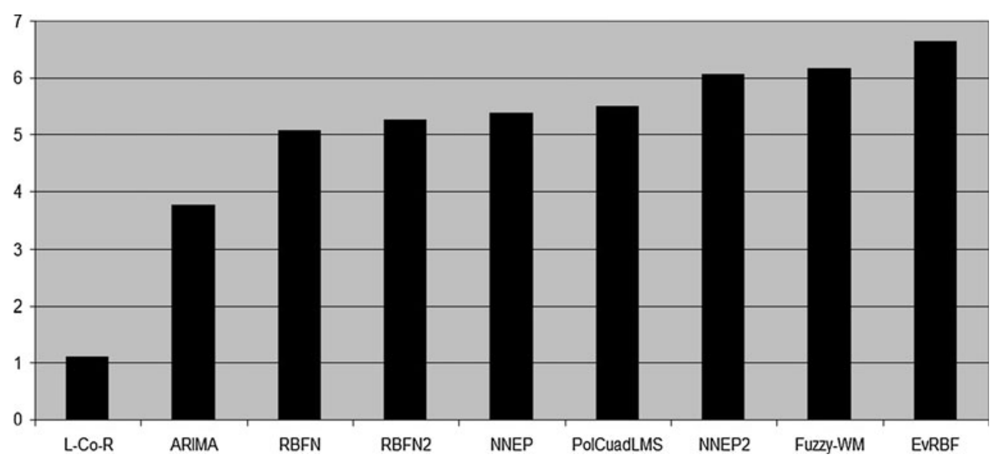

Fig. 6 Graphic of the ranking of the methods for MdAPE. *Lower values* represent better predictions of the test dataset

5 Conclusions and future research

This work describes the L-Co-R method, a coevolutionary algorithm for time series forecasting. L-Co-R simultaneously evolves two populations or different individual species, RBFNs and sets of lags that will be used to predict future values. In the population of RBFNs, a set of neural networks evolves trying to obtain a suitable network architecture. The population of lags is formed by sets of significant lags that are used to forecast future values. Any member of a population has to cooperate with individuals from the other populations (collaborators) to generate good solutions.

L-Co-R has been tested across 34 datasets from different areas and with different statistical characteristics. In

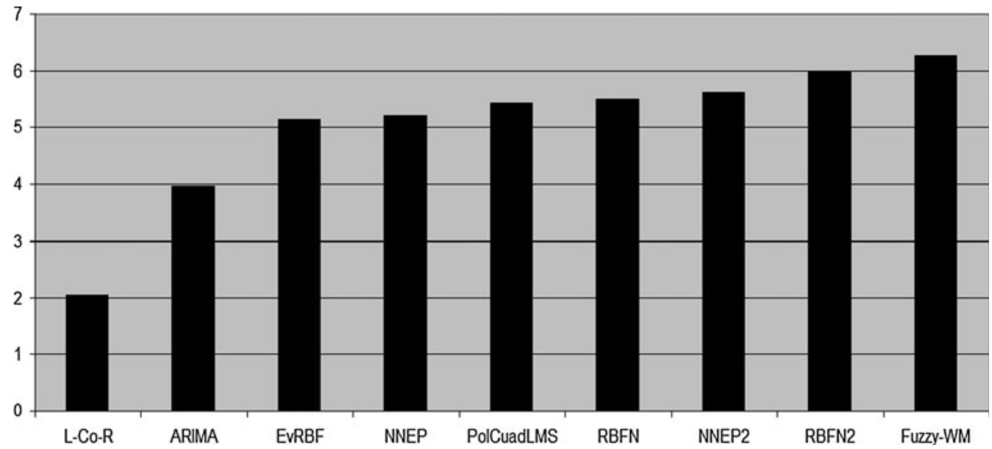

Fig. 7 Graphic of the ranking of the methods for sMdAPE. *Lower values* represent better predictions of the test dataset

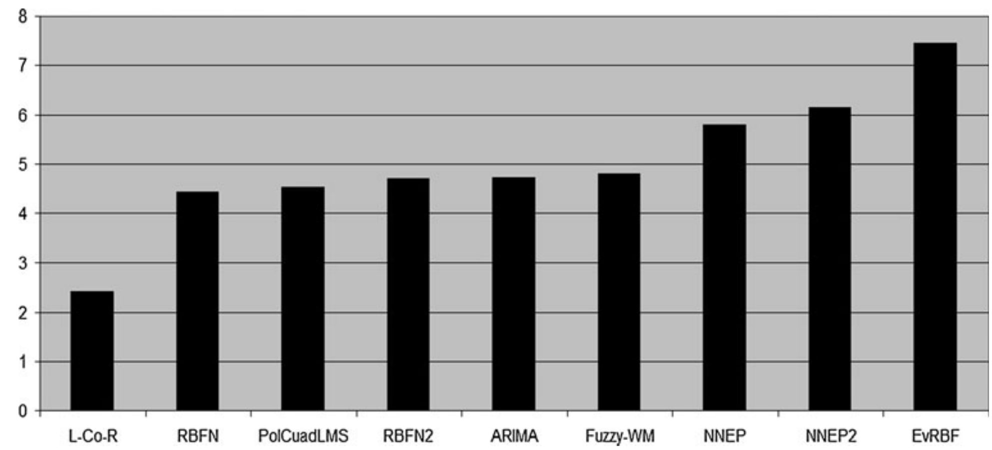

Fig. 8 Graphic of the ranking of the methods for MASE. *Lower values* represent better predictions of the test dataset

Table 14 Adjusted p valuesof Holm's procedure for the comparison of the control algorithm (L-Co-R) with the remaining algorithms with respect to MAPE and MdAPE

MAPE		MdAPE	
Method	p Holm	Method	p Holm
EvRBF	9.85E−18	EvRBF	6.76E−16
NNEP2	4.07E−16	Fuzzy-WM	1.83E−13
NNEP	2.21E−13	NNEP2	6.08E−13
Fuzzy-WM	1.93E−08	PolCuadLMS	2.09E−10
PolCuadLMS	9.66E−08	NNEP	5.43E−10
RBFN	6.47E−06	RBFN2	1.28E−09
RBFN2	6.47E−06	RBFN	4.52E−09
ARIMA	3.00E−03	ARIMA	6.74E−05

Table 15 Adjusted p values of Holm's procedure for the comparison of the control algorithm (L-Co-R) with the remaining algorithms with respect to sMdAPE and MASE

sMdAPE		MASE	
Method	p Holm	Method	p Holm
Fuzzy-WM	1.93E−09	EvRBF	4.13E−13
RBFN2	1.31E−07	NNEP2	1.69E−07
NNEP2	5.04E−07	NNEP	2.67E−06
RBFN	1.10E−06	Fuzzy-WM	1.98E−03
PolCuadLMS	1.41E−06	ARIMA	2.21E−03
NNEP	6.47E−06	RBFN2	2.21E−03
EvRBF	6.65E−06	PolCuadLMS	3.33E−03
ARIMA	4.00E−03	RBFN	3.33E−03

addition, the new algorithm has been compared with six different methods found in the literature and with four different quality measures.

A statistic study has been carried out in order to confirm the first observations in the results. First, Friedman and Iman–Davenport tests have been applied to test whether significant differences exist among all methods, and then Holm procedure to find whether the control algorithm (L-Co-R) presents statistical differences with respect to the remaining methods.

Then, it can be affirmed that L-Co-R yields good results with respect to the other methods, taking into account the large set of sample data, which have different characteristics and nature. L-Co-R designs simple and appropriate

RBFNs, and finds small and suitable set of lags. Furthermore, the method proposed is able to automatically remove the trend of the time series.

In any case, further study must be performed in order to assess the robustness of the new method as well as the effectiveness of their predictions for long-time forecasting and with a changing horizon environment. Future works should also include the study of the effect of the credit assignment and the number of collaborators in the process of collaboration.

Acknowledgments This work has been supported by the Caja Rural de Jaen and the University of Jaen (Spain) UJA-08-16-30 project, the regional project TIC-3928 (Feder Founds), and the Spanish project TIN 2008-06681-C06-02 (Feder Founds).

References

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput Fusion Found Methodol Appl 13:307–318. doi:10.1007/s00500-008-0323-y

Arizmendi CM, Sanchez J, Ramos NE, Ramos GI (1993) Time series predictions with neural nets: Application to airborne pollen forecasting. Int J Biometeorol 37:139–144. doi:10.1007/BF01212623

Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting methods: empirical comparisons. Int J Forecast 8:69–80

Assimakopoulos V, Nikolopoulos K (2000) The theta model: a decomposition approach to forecasting. Int J Forecast 16(4): 521–530

Au CK, Leung HF (2007) Biasing mutations in cooperative coevolution. In: Proceedings of IEEE Congress on evolutionary computation, CEC 2007, pp 828–835

Bezerianos A, Papadimitriou S, Alexopoulos D (1999) Radial basis function neural networks for the characterization of heart rate variability dynamics. Artif Intell Med 15(3):215–234. doi:10.1016/S0933-3657(98)00055-4

Bowerman BL, O'Connell RT, Koehler AB (2004) Forecasting: methods and applications. Thomson Brooks/Cole: Belmont, CA

Box GEP, Jenkins GM (1976) Time series analysis: forecasting and control. Holden Day, San Francisco

Bradley MD, Jansen DW (2004) Forecasting with a nonlinear dynamic model of stock returns and industrial production. Int J Forecast 20(2):321–342. doi:10.1016/j.ijforecast.2003.09.007

Brockwell P, Hyndman R (1992) On continuous-time threshold autoregression. Int J Forecast 8(2):157–173. doi:10.1016/0169-2070(92)90116-Q

Broomhead D, Lowe D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 2:321–355

Brown R (1959) Statistical forecasting for inventory control. McGraw-Hill, New York

Carse B, Fogarty T (1996) Fast evolutionary learning of minimal radial basis function neural networks using a genetic algorithm. In: Proceedings of evolutionary computing. Lecture dois in Computer Science, vol 1143, pp 1–22. Springer, Berlin. doi:10.1007/BFb0032769

Castillo PA, Merelo JJ, Prieto A, Rivas VM, Romero G (2000) G-prop: global optimization of multilayer perceptrons using gas. Neurocomputing 35:149–163. doi:10.1016/S0925-2312(00)00302-7

Chan KS, Tong H (1986) On estimnating thresholds in autoregressive models. J Time Ser Anal 7(3):179–190. doi:10.1111/j.1467-9892.1986.tb00501.x

Chatterjee A, Siarry P (2006) Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput Oper Res 33:859–871. doi:10.1016/j.cor.2004.08.012

Clements MP, Franses PH, Swanson NR (2004) Forecasting economic and financial time-series with non-linear models. Int J Forecast 20(2):169–183. doi:10.1016/j.ijforecast.2003.10.004

Crone S, Hibon M, Nikolopoulos K (2011) Advances in forecasting with neural networks? Empirical evidence from the nn3 competition on time series prediction. Int J Forecast 27(3):635–660

Dash PK, Liew AC, Rahman S, Ramakrishna G (1995) Building a fuzzy expert system for electric load forecasting using a hybrid neural network. Exp Syst Appl 9(3):407–421. doi:10.1016/0957-4174(95)00013-Y

Dawson CW, Wilby RL, Harpham C, Brown MR, Cranston E, Darby EJ (2000) Modelling ranunculus presence in the rivers test and itchen using artificial neural networks. In: Proceedings of international conference on geocomputation

de A Araújo R (2010a) Hybrid intelligent methodology to design translation invariant morphological operators for brazilian stock market prediction. Neural Netw 23:1238–1251

de A Araújo R (2010b) A quantum-inspired evolutionary hybrid intelligent approach for stock market prediction. Int J Intell Comput Cybern 3(10):24–54

de A Araújo R (2010c) Swarm-based translation-invariant morphological prediction method for financial time series forecasting. Inform Sci 180:4784–4805

de A Araújo R (2011) Translation invariant morphological time-lag added evolutionary forecasting method for stock market prediction. Exp Syst Appl 38:2835–2848. doi:10.1016/j.eswa.2010.08.076

Derrac J, García S, Herrera F (2010) Ifs-coco: instance and feature selection based on cooperative coevolution with nearest neighbor rule. Pattern Recogn 43(6):2082–2105. doi:10.1016/j.patcog.2009.12.012

Du H, Zhang N (2008) Time series prediction using evolving radial basis function networks with new encoding scheme. Neurocomputing 71:1388–1400. doi:10.1016/j.neucom.2007.06.004

Eshelman LJ (1991) The chc adptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Proceedings of first workshop on foundations of genetic algorithms, Morgan Kaufmann, Menlo Park, pp 265–283

Ferreira T, Vasconcelos G, Adeodato P (2008) A new intelligent system methodology for time series forecasting with artificial neural networks. Neural Process Lett 28(2):113–129

Fildes R (1983) An evaluation of bayesian forecasting. J Forecast 2(2):137–150. doi:10.1002/for.3980020205

Fildes R (1992) The evaluation of extrapolative forecasting methods. Int J Forecast 8(1):81–98. doi:10.1016/0169-2070(92)90009-X

Fildes R, Nikolopoulos K, Crone SF, Syntetos AA (2008) Forecasting and operational research: a review. J Oper Res Soc 59:1150–1172

Fu X, Wang L (2003) Data dimensionality reduction with application to simplifying rbf network structure and improving classification performance. IEEE Trans Syst Man Cybern Part B 33:399–409. doi:10.1109/TSMCB.2003.810911

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13:959–977. doi:10.1007/s00500-008-0392-y

García Pajares R, Benitez J, Sainz Palmero G (2008) Feature selection form time series forecasting: a case study. In: Eighth international conference on hybrid intelligent systems, pp 555–560

Garcia-Pedrajas N, Hervas-Martinez C, Ortiz-Boyer D (2005) Cooperative coevolution of artificial neural network ensembles for pattern classification. IEEE Trans Evol Comput 9:271–302. doi:10.1109/TEVC.2005.844158

García-Pedrajas N, del Castillo JR, Ortiz-Boyer D (2010) A cooperative coevolutionary algorithm for instance selection for instance-based learning. Mach Learn 78:381–420. doi:10.1007/s10994-009-5161-3

Gardner ES (1985) Exponential smoothing: the state of the art. J Forecast 4(1):1–28. doi:10.1002/for.3980040103

Gooijer JGD, Hyndman RJ (2006) 25 years of time series forecasting. Int J Forecast 22(3):443–473

Granger C, Tersvirta T (1993) Modelling non-linear economic relationships. Oxford University Press, Oxford

Harpham C, Dawson CW (2006) The effect of different basis functions on a radial basis function network for time series prediction: a comparative study. Neurocomputing 69:2161–2170. doi:10.1016/j.neucom.2005.07.010

Harpham C, Dawson CW, Brown MR (2004) A review of genetic algorithms applied to training radial basis function networks. Neural Comput Appl 13:193–201. doi:10.1007/s00521-004-0404-5

Harrison PJ, Stevens CF (1976) Bayesian forecasting. J Royal Stat Soc Ser B (Methodological) 38(3):205–247

Harvey AC (1984) A unified view of statistical forecasting procedures. J Forecast 3(3):245–275. doi:10.1002/for.3980030302

Hippert HS, Taylor JW (2010) An evaluation of bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. Neural Netw 23(3):386–395. doi:10.1016/j.neunet.2009.11.016

Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor

Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6:65–70

Hyndman RJ, Billah B (2003) Unmasking the theta method. Int J Forecast 19(2):287–290

Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688. doi:10.1016/j.ijforecast.2006.03.001

Jain A, Kumar AM (2007) Hybrid neural network models for hydrologic time series forecasting. Appl Soft Comput 7(2):585–592. doi:10.1016/j.asoc.2006.03.002

Kavaklioglu K (2011) Modeling and prediction of turkey's electricity consumption using support vector regression. Appl Energ 88(1):368–375. doi:10.1016/j.apenergy.2010.07.021

Lee CM, Ko CN (2009) Time series prediction using rbf neural networks with a nonlinear time-varying evolution pso algorithm. Neurocomputing 73(1–3):449–460. doi:10.1016/j.neucom.2009.07.005

Li M, Tian J, Chen F (2008) Improving multiclass pattern recognition with a co-evolutionary rbfnn. Pattern Recogn Lett 29(4):392–406. doi:10.1016/j.patrec.2007.10.019

Lukoseviciute K, Ragulskis M (2010) Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems. Neurocomputing 73:2077–2088

Ma X, Wu HX (2010) Power system short-term load forecasting based on cooperative co-evolutionary immune network model. In: Proceedings of 2nd international conference on education technology and computer (ICETC), pp 582–585

Makridakis SG, Hibon M (2000) The m3-competition: results, conclusions and implications. Int J Forecast 16(4):451–476

Makridakis SG, Andersen A, Carbone R, Fildes R, Hibon M, Lewandowski R, Newton J, Parzen E, Winkler R (1982) The accuracy of extrapolation (time series) methods: results of a forecasting competition. J Forecast 1(2):111–153. doi:10.1002/for.3980010202

Martínez-Estudillo A, Martínez-Estudillo F, Hervás-Martínez C, García-Pedrajas N (2006) Evolutionary product unit based neural networks for regression. Neural Netw 19(4):477–486. doi:10.1016/j.neunet.2005.11.001

Maus A, Sprott JC (2011) Neural network method for determining embedding dimension of a time series. Commun Nonlinear Sci Numer Simul 16(8):3294–3302

Merelo JJ, Prieto A (1995) G-lvq, a combination of genetic algorithms and lvq. In: Proceedings of artificial neural nets and genetic algorithms, Springer, Berlin, pp 92–95

Panait L, Wiegand RP, Luke S (2003) Improving coevolutionary search for optimal multiagent behaviors. In: Proceedings of the eighteenth international joint conference on artificial intelligence, Morgan Kaufmann, Menlo Park, pp 653–658

Paredis J (1995) Coevolutionary computation. Artif Life 2(4):355–375. doi:10.1162/artl.1995.2.4.355

Pena D (2005) Análisis de Series Temporales. Alianza Editorial

Perez-Godoy MD, Aguilera JJ, Berlanga FJ, Rivas VM, Rivera AJ (2008) A preliminary study of the effect of feature selection in evolutionary rbfn design. In: Proceedings of information processing and management of uncertainty in knowledge-based system, pp 1151–1158

Perez-Godoy MD, Pérez-Recuerda P, Frías M, Rivera AJ, Carmona C, Parras M (2010a) Co^{2}rbfn for short and medium term forecasting of the extra-virgin olive oil price. In: González J, Pelta D, Cruz C, Terrazas G, Krasnogor N (eds) Proceedings of nature inspired cooperative strategies for optimization (NICSO 2010), Studies in Computational Intelligence, vol 284, pp 113–125. Springer, Berlin. doi:10.1007/978-3-642-12538-6_10

Perez-Godoy MD, Rivera A, Berlanga FJ, del Jesus MJ (2010b) Co^{2}rbfn: an evolutionary cooperative–competitive rbfn design algorithm for classification problems. Soft Comput Fusion Found Methodol Appl 14:953–971. doi:10.1007/s00500-009-0488-z

Potter M, Jong KD (1994) A cooperative coevolutionary approach to function optimization. In: Davidor Y, Schwefel HP, Mnner R (eds) Proceedings of parallel problem solving from nature PPSN III, Lecture Notes in Computer Science, vol 866, pp 249–257. Springer, Berlin. doi:10.1007/3-540-58484-6_269

Potter M, Jong KD (2000) Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evol Comput 8(1):1–29. doi:10.1162/106365600568086

Qian-Li M, Qi-Lun Z, Hong P, Tan-Wei Z, Jiang-Wei Q (2008) Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network. Chin Phys B 17(2). doi:10.1088/1674-1056/17/2/031

Qiu W, Liu X, Li H (2011) A generalized method for forecasting based on fuzzy time series. Exp Syst Appl 38(8):10446–10453. doi:10.1016/j.eswa.2011.02.096

Rivas VM, Merelo JJ, Castillo PA, Arenas MG, Castellano JG (2004) Evolving rbf neural networks for time-series forecasting with evrbf. Inform Sci 165(3–4):207–220. doi:10.1016/j.ins.2003.09.025

Rivas VM, Arenas MG, Merelo JJ, Prieto A (2007) Evrbf: evolving rbf neural networks for classification problems. In: Proceedings of the 7th conference on 7th WSEAS international conference on applied informatics and communications, Stevens Point, Wisconsin, USA, vol 7, pp 98–103

Rivera AJ, Rojas I, Ortega J, del Jesus MJ (2007) A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks. Soft Comput Fusion Found Methodol Appl 11:655–668. doi:10.1007/s00500-006-0128-9

Rustagi JS (1994) Optimization techniques in statistics. Academic Press, Boston

Samanta B (2011) Prediction of chaotic time series using computational intelligence. Exp Syst Appl 38(9):11406–11411. doi:10.1016/j.eswa.2011.03.013

Sarantis N (2001) Nonlinearities, cyclical behaviour and predictability in stock markets: international evidence. Int J Forecast 17(3):459–482. doi:10.1016/S0169-2070(01)00093-0

Sergeev S, Mahotilo K, Voronovsky G, Petrashev S (1998) Genetic algorithm for training dynamical object emulator based on rbf neural network. Int J Appl Electromagn Mech 9:65–74

Sheskin D (2006) Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC, London

Sheta AF, Jong KD (2001) Time-series forecasting using ga-tuned radial basis functions. Inform Sci 133(3-4):221–228. doi:10.1016/S0020-0255(01)00086-X

Snyder RD (1985) Recursive estimation of dynamic linear models. J Royal Stat Soc Ser B (Methodological) 47:272–276. http://www.jstor.org/stable/2345570

Sun ZL, Huang D, Zheng CH, Shang L (2006) Optimal selection of time lags for tdsep based on genetic algorithm. Neurocomputing 69(7–9):884–887

Takens F (1980) Detecting strange attractor in turbulence. In: Dynamical systems and turbulence. Lecture notes in mathematics, vol 898. Springer, New York, NY, pp 366–381

Tan KC, Yang YJ, Goh CK (2006) A distributed cooperative co-evolutionary algorithm for multi-objective optimization. IEEE Trans Evol Comput 10:527–549. doi:10.1109/TEVC.2005.860762

Tanaka N, Okamoto H, Naito M (2001) Estimating the active dimension of the dynamics in a time series based on a information criterion. Phys D 158:19–31

Tang Z, de Almeida C, Fishwick PA (1991) Time series forecasting using neural networks vs. boxjenkins methodology. Simulation 57:303–310

Tong H (1978) On a threshold model. Pattern Recogn Signal Process NATO ASI Ser E Appl Sci 29:575–586

Tong H (1983) Threshold models in non-linear time series analysis. In: Lecture notes in statistics, vol 21. Springer, Berlin

Valenzuela O, Rojas I, Rojas F, Pomares H, Herrera LJ, Guillen A, Marquez ML, Pasadas M (2008) Hybridization of intelligent techniques and arima models for time series prediction. Fuzzy Sets Syst 159(7):821–845. doi:10.1016/j.fss.2007.11.003

Wang CC (2011) A comparison study between fuzzy time series model and arima model for forecasting taiwan export. Exp Syst Appl 38(8):9296–9304

Wang LX, Mendel JM (2002) Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 22:1414–1427. doi:10.1109/21.199466

Whitehead BA, Choate TD (1996) Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. IEEE Trans Neural Netw 7:869–880. doi:10.1109/72.508930

Wichern DW, Jones RH (1977) Assessing the impact of market disturbances using intervention analysis. Manag Sci 24:329–337

Wiegand RP, Liles WC, De Jong K (2001) An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In: Proceedings of the genetic and evolutionary computation conference, Morgan Kaufmann, Menlo Park, pp 1235–1242

Winters PR (1960) Forecasting sales by exponentially weighted moving averages. Manag Sci 6:324–342. http://www.jstor.org/stable/2627346

Xue Y, Watton J (1998) Dynamics modelling of fluid power systems applying a global error descent algorithm to a self-organising radial basis function network. Mechatronics 8(7):727–745. doi:10.1016/S0957-4158(98)00024-5

Yu THK, Huarng KH (2010) A neural network-based fuzzy time series model to improve forecasting. Exp Syst Appl 37(4):3366–3372. doi:10.1016/j.eswa.2009.10.013

Zar J (1999) Biostatistical analysis. Prentice Hall, Englewood Cliffs

Zhang G, Hu MY (1998) Neural network forecasting of the british pound/us dollar exchange rate. Omega Int J Manag Sci 26(4):495–506. doi:10.1016/S0305-0483(98)00003-6

Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14(1):35–62. doi:10.1016/S0169-2070(97)00044-7

Zhang GP, Qi M (2005) Neural network forecasting for seasonal and trend time series. Eur J Oper Res 160(2):501–514. doi:10.1016/j.ejor.2003.08.037

Zhu S, Wang J, Zhao W, Wang J (2011) A seasonal hybrid procedure for electricity demand forecasting in china. Appl Energ 88(11):3807–3815. doi:10.1016/j.apenergy.2011.05.005