Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 00 (2012) 1–16

Neurocomp.

Short, medium and long term time series forecasting using the L-Co-R algorithm

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

Abstract

This paper describes the coevolutionary algorithm L-Co-R (Lags COevolving with Rbfns), an analyzes a study about its performance in the short, medium and long term time series forecasting tasks. The method allows the coevolution, in a single process, of the time series model as well as the set of lags to be used for predictions. Radial basis functions neural networks are used as the model, being the individuals of one population, while set of candidate lags are also evaluated as individuals of the second population. In order to test the behaviour of the algorithm in a new context of a variable horizon, 5 different measures have been analyzed, for more than 30 different databases, comparing against 6 existing algorithms and, all of this, for 7 different prediction horizons. The statistical analysis of the results shows that L-Co-R outperforms the rest of methods, regardless of the horizon.

Keywords:
Neural Networks, Coevolutionary algorithms, Time series forecasting, Significant lags, Variable term horizon

1. Introduction

A time series can be defined as a chronological sequence of observed data from any daily task or behavior, or any activity in fields like Engineering, Biology, Economy, or Social Sciences, among many others. Therefore, the task of predicting values of the series based on past and present values in order to achieve the information of the underlying model can be understood under the concept of time series forecasting.

Dealing with time series forecasting implies to consider three important aspects, where the first one is the choice of the time periods (or lags) that must be used in order to forecast values. This way, the own selection of these lags to be used as input variables for the model turns itself into a problem that can be faced using data mining techniques. The second aspect to have into account is the trend, i.e, whether the time series tends to grow or decrease considering a long period of time. Finally, it has to be considered the prediction period or horizon. Usually, there exists a tendency to forecast using short horizons due to the difficulty of utilizing longer periods and, therefore, the results of the first ones tend to be more reliable.

There exist a wide number of techniques that have been developed to model and forecast time series. These techniques can be coursely grouped into descriptive traditional technologies, linear and nonlinear modern models, and technologies coming from the soft computing area. Among all of these technologies, ARIMA, by Box and Jenkins [1], is probably the most well known and widely used method. The method combines autoregressive and moving average terms into an equation in order to build a linear model to forecast new values. The autoregressive part of the equation relates the future value to past and present ones, while the moving average component relates the future value to the errors of previous forecasts. Nevertheless, the models provided by the ARIMA method are simplistic linear models, being unable to find complex subtle patterns in the time series data.

There also exist diverse techniques in the the soft computing area developed to tackle time series forecasting, such as artificial neural networks (ANNs), evolutionary algorithms, fuzzy logic or expert systems. The learning and generalization capabilities of ANNs have shown, by means of many successful applications, that they are a suitable alternative tool for both forecasting researchers and practitioners.

In the herein presented work, it has been utilized the coevolutionary algorithm L-Co-R (Lags COevolving with Rbfns) [70] which makes use of Radial Basis Function Networks (RBFNs) and Evolutionary Algorithms (EAs). The objective is to get neural networks capable of modeling time series in one hand, and finding out the specific lags of the time series to predict future values on the other hand. This double goal is carried out by a coevolutionary process that divides the main problem into two subproblems which depend on each other. This way, one population evolves sets of time series lags to forecast future values, and the second population evolves a set of RBFNs to obtain an appropriate design for the mentioned forecasting. The last one determines the architecture of the net and parameters like number of layers, connection between neurons, weights and radii for neurons, among others.

Apart from minimizing the error obtained when predicting time series, L-Co-R is also developed to be used with variable horizons of prediction. In addition, it employs a collaboration process in which the individuals of one population can cooperate with individuals form the other population to favor better solutions.

In order to determine the effectiveness of the L-Co-R with short, medium and long term, we have set 7 diverse horizons to predict with, and 34 different time series. The results obtained have been compared with the results of others 6 algorithms found in the literature, and finally, we have applied 5 quality measures to get a conclusion.

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

2. State of the art

Time series forecasting has been a major field of research in the area of statistics [2] as well as operational research [3]. In the latest years numerous methods have been emerged with the objective of modeling and/or forecasting time series by means of linear and nonlinear models. Linear methods have been widely used to model time series, among them stand out the exponential smoothing methods [4, 5], simple exponential smoothing (SES), Holt's linear methods, some variations of the Holt-Winter's methods, and State space models [6]. However, the ARIMA methods [1], which also belongs to linear methods, established a border line between traditional and modern 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.

Nevertheless, these linear time series forecasting were insufficient in many real applications, leading to the development of nonlinear time series forecasting. Nonlinear models include regime-switching models, which comprise the wide variety of existing threshold autoregressive (TAR) models [7] as: self-exciting (SETAR) models [8], smooth transition (STAR) models [9], and continuous-time (CTAR) models [10]. Nevertheless, as pointed out by Clements [11], current nonlinear method's main problems are the followings: they use to develop very complex models; they do not perform in a robust way; and, they are difficult to use. De Gooijer [2] also concludes that future research on nonlinear models should include, among others, the search for easy to use software.

On the other hand, time series forecasting has been faced with soft computing approaches, as the ones by Samanta [12] and Zhu [13], who developed methods based on cooperative particle swarm optimization. Works like [14] and [15] proposed fuzzy time series models for forecasting, and Yu and Huarng [16] applied ANNs for training and forecasting in their fuzzy time series model. Models such as support vector regression [17] and fuzzy expert system [18] were proposed for the electricity demand forecasting, among others.

ANNs have been also successfully applied to time series and recognized as an important tool for forecasting. The work by Tang [19] 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 [20]. Jain and Kumar determined in their work [21] 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 time series. The work by Arizmendi [22] obtained accurate predictions of the airborne pollen concentrations using ANNs. Zhang and Hu [20] employed ANNs, and Rivas et al. [23] RBFNs, for forecasting British pound and US dollar exchange rates. Bezerianos et al. [24] employed RBFNs for the assessment and prediction of the heart rate variability.

Specifically inside the ANNs, the use of RBFs as activation functions for them and its application to time series forecasting were firstly considered by Broomhead and Lowe in 1988 [25]. Afterwards, new works by Carse and Fogarty [26], and Whitehead and Choate [27] focused on the prediction of time series.

In later works, Harpham and Dawson [28] studied the effect of different basis functions on an RBFN for time series prediction. Moreover, Du [29] 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 were represented in one chromosome and evolved simultaneously by GAs so that the selection of nodes and inputs could be automatically achieved.

Previous works found in literature can also be classified according to the prediction horizon. Thus, 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 reliant because of the error propagation [30]. Thus, neural network models have been traditionally applied in short-term forecasting [31, 32]. For instance, the work by Perez-Godoy [33] 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.

As mentioned, 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' theorem [34] 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.

In order to tackle the lags selection problem, an evolutionary method that performs a search for the minimum number of dimensions, Time-delay Added Evolutionary Forecasting (TAEF), is presented in [35]. The methodology is inspired in Takens' theorem and consists of an iterative hybrid model composed of an ANN combined with a genetic algorithm (GA). In [36] 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 neighbors algorithm and then a near-optimal set of time lags is found with a genetic algorithm for a fuzzy inference system.

There are some methods that carry out an automatic search of the relevant lags. QIEHI algorithm [37], for instance, is an evolutionary hybrid intelligent method which is composed of an ANN and a modified evolutionary algorithm to search the minimum dimension to determine the characteristic phase for time series. Another hybrid methodology composed of a modular morphological ANN with an evolutionary algorithm that searches for the best time lags is described in [38]. In [39] 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 lag selection is performed as a postprocessing stage in [40] with a sensitivity computation of the output to each time lag.

As can be observed, the approaches in 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.

For this reason, we consider coevolutionary algorithms a good mechanism to simultaneously solve these problems.

Cooperative coevolution, introduced by Potter [41, 42], consists of identifying the natural decomposition of a problem into subcomponents. Then, a population of individuals per subproblem is created and evolved by means of collaboration with individuals from others populations. There are many possible methods for choosing representatives with which to collaborate: random collaboration [43], best collaboration [41] (the most widely used in the methods of the literature), complete collaboration, and mixed collaboration [44]. Another important point 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 has been proved to be significantly better using maximum method than using minimum or average [43].

Cooperative coevolution has been employed for tasks like function optimization [45], multi-objective evolutionary optimization [46], instance selection [47], and feature selection [48], among others. Cooperative coevolution has

Error measures		
MAE	Mean Absolute Error	$mean(\mid e_t \mid)$
MAPE	Mean Absolute Percentage Error	$mean(\mid p_t \mid)$
MdAPE	Median Absolute Percentage Error	$median(\mid p_t \mid)$
sMdAPE	Symmetric Median Absolute Percentage Error	$median(200 \mid Y_t - F_t \mid (Y_t + F_t))$
MASE	Mean Absolute Scaled Error	$mean(\mid q_t \mid)$

Table 1. Used forecast accuracy measures.

also been used in order to train ANNs, such as the cooperative coevolutive approach for designing neural network ensembles [49] and RBFNs [50].

It is possible to find coevolution applied to forecasting tasks as in [51] 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 [52] 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.

Finally, in order 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) [53] and it was the primary measure in the M-competition [54]. Other works recommended other measures such as Geometric Mean Relative Absolute Error (GMRAE), Median Relative Absolute Error (MdRAE), and Median Absolute Percentage Error (MdAPE) [55, 56]. Later, the MdRAE, sMAPE (Symmetric Mean Absolute Percentage Error), and sMdAPE (Symmetric Median Absolute Percentage Error) were proposed [57].

Nevertheless, Hyndman and Koehler in their work [58] 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 [2] and [58], it can be found a description of different error measures. Among all of them, those used in this work are shown in table 1, taking into account that Y_t is the observation at time $t = 1, ..., n$; F_t is the forecast of Y_t; e_t is the forecast error (i.e. $e_t = Y_t - F_t$); $p_t = 100e_t/Y_t$ is the percentage error, and $q_t = \dfrac{e_t}{\dfrac{1}{n-1}\sum_{i=2}^{n} \mid Y_i - Y_{i-1} \mid}$.

3. L-Co-R: Lags COevolving with Rbfns

L-Co-R, Lags COevolving with Rbfns [70], is an algorithm which designs RBFNs for time series forecasting. It obtains 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 [59]. 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.

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. Therefore, the new algorithm is composed of the following two populations:

- Population of RBFNs: a set of RBFNs evolves to design an appropriate architecture of the net. The population uses a real codification in which every individual represent a set of neurons (RBFs) that composes the network. The number of neurons is variable since it can increase or decrease during the evolutionary process. Every neuron is defined by a center and a radius. The center 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).

- Population of lags: sets of lags evolves to forecast future values of the time series. This population utilizes 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. The length of the chromosome is set at the beginning corresponding with the specific parameter, so that it cannot vary its size during the execution of the algorithm.

In both populations every individual is itself a possible solution to the subproblem.

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

The method performs a preliminary stage of preprocessing which removes the trend of the time series. Then, the L-Co-R algorithm creates the two initial populations and evaluates every individual of each population. Once the initial populations have been created, the coevolutionary process starts.

Firstly, the population of lags selects the individuals which are going to take part of the subpopulation. HUX crossover is applied to these selected individuals 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. Subsequently, parents from population and children from subpopulation are joined in a single and bigger population and they are ranked regarding their fitness. Finally, the worst individuals are deleted from this population until it reaches the original size, becoming the new parent population, and, as the CHC scheme is followed, eventually the population can be reinizialized.

And secondly, the population of RBFNs begins to evolve when the population of lags has been evolved during a pre-specified number of generations. Then, the individuals of the subpopulation are selected, the operators are applied, and a collaborator from population of lags is designated in order to establish the fitness of every individual.

Finally, at the end of the coevolutionary process, two models formed by a neural network and a set of lags are obtained. The first model 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 trained again and the one with the best fitness will be the final model. Then, the forecasted values for the data test are obtained, and at this point, the postprocessing phase takes place so that the final test error can be computed.

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. A collaborator will be selected from each population to assess the fitness.

L-Co-R is implemented to use a best collaboration scheme [41] and optimistic approach [43] 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 equation 1.

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

3.1. Evolutionary process

Both populations are randomly generated for the first generation. The population of RBFNs considers 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 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 lags takes 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 set of lags is evolved by means of CHC [60] algorithm.

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

The operators used by L-Co-R for every population are the followings:

- Population of RBFNs:
  - Selection: this 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 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 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 value of hidden neurons. The operator assigns a random value to the radius following an internal probability.
    * Adder: it adds new neurons 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 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 data set. 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.
- Population of lags:
  - 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.

Time series	Data	Units	Period	Description
Accidents	240	Units	Monthly (Jan. 1979 - Dec. 1998)	Number of accidents during a working day.
AccDeath	216	Units	Monthly (Jan. 1990 - Dec. 2007)	Number of road accident casualties.
AccVictims	216	Units	Monthly (Jan. 1990 - Dec. 2007)	Number of road accident casualties.
Airline	144	Thousands	Monthly (Jan. 1949 - Dec. 1960)	Airplane passengers of international flies.
WmFrancfort	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of Frankfort.
WmLondon	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of London.
WmMadrid	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of Madrid.
WmMilan	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of Milan.
WmNewYork	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of New York.
WmParis	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of Paris.
WmTokyo	156	Index	Monthly (Jan. 1988 - Dec. 2000)	Monthly values about market of Tokyo.
Colgtems	276	Market quota	Weekly (Jan. 1958 - Apr. 1963)	Market quota of Colgate toothpaste.
Colgtepr	276	Price	Weekly (Jan. 1958 - Apr. 1963)	Price of Colgate toothpaste.
Crestms	276	Market quota	Weekly (Jan. 1958 - Apr. 1963)	Market quota of Crest toothpaste.
Crestpr	276	Price	Weekly (Jan. 1958 - Apr. 1963)	Price of Crest toothpaste.
Deceases	228	Units	Monthly (Jan. 1980 - Dec. 1998)	Number of monthly deceases.
Spectators	233	Thousands	Monthly (Jan. 1990 - May 2009)	Number of thousand spectators who were in the cinema.
SpaMovSpec	233	Thousands	Monthly (Jan. 1990 - May 2009)	Spectators who watched a Spanish movie in the cinema.
ForMovSpec	233	Thousands	Monthly (Jan. 1990 - May 2009)	Spectators who watched a foreign movie in the cinema.
Exchange	208	Price	Weekly (Dec. 1979 - Dec. 1983)	Exchange rates between British Pound and US Dollar.
Gasoline	618	Thousands	Weekly (Jul. 1993 - May 2005)	Finished motor gasoline production (thousand barrels).
MortCanc	43	Units	Monthly (Jan. 2006 - Jul. 2009)	Number of canceled mortgages.
MortMade	79	Units	Monthly (Jan. 2003 - Jul. 2009)	Number of made mortgages.
Books	132	Thousands	Monthly (Jan. 1998 - Dec. 2008)	Editorial production of books.
Motorcycles	234	Units	Monthly (Jan. 1990 - Jun. 2009)	Manufacture of motorcycles.
Unemployed	164	Units	Monthly (Jan. 1996 - Aug. 2009)	Number of Spanish unemployed people.
FreeHouPrize	58	Euros	Quarterly (Q1 1995 - Q2 2009)	Price per m^2 of private housing.
Prisoners	235	Units	Monthly (Jan. 1990 - Jul. 2009)	Number of prisoners.
Takings	233	Euros	Monthly (Jan. 1990 - mayo 2009)	Average spending per spectator.
TurIn	234	Thousands	Monthly (Jan. 1990 - Jun. 2009)	Internal air traffic.
TurOut	234	Thousands	Monthly (Jan. 1990 - Jun. 2009)	External air traffic .
TUrban	164	Thousands	Monthly (Jan. 1996 - Aug. 2009)	Number of passengers transported by urban transport.
Cars	236	Units	Monthly (Jan. 1990 - Aug. 2009)	Vehicle manufacture (cars).
HouseFin	211	Units	Monthly (Jan. 1992 - Jul. 2009)	Number of finished houses.

Table 2. Characteristics of used time series.

– 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.

4. Experimental study

This section describes the experiments carried out to test the behavior of L-Co-R predicting time series as forecasting horizon grows. For that, a new context has been created in which L-Co-R has been applied to long prediction periods. The effectiveness of the algorithm has been compared with other methods, and a statistical study is included in order to draw a conclusion.

The experimentation has been realized for 7 different horizons using 34 public bases of examples which have different characteristics with respect to number of data, period of time and topic they represent. Most data bases have been extracted from the Spanish National Statistics Institute[1]. A brief description of every one is given in table 2.

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

Method	Parameter	Value	Method	Parameter	Value
L-Co-R	PopSizeLag	50	EvRBF	Population size	100
	MaxGenerationLag	5		Generations	10
	MaxLongCrom	10%		Validation rate	0.25
	PopSizeRbfn	50		Neurons rate	0.1
	MaxGenerationsRbfn	10		Tournament size	30
	ValidationRate	0.25		Replacement rate	0.75
	NeuronsRate	0.05		Crossover rate	0.9
	TournamentSize	3		Mutator rate	0.1
	ReplacementRate	0.5	Fuzzy-WM	Number of labels	5
	XOverRate	0.8		KB Output File Format with	
	MutatorRate	0.2		weight values to 1?	0
	MaxGenerations	20	NNEP2	Number of neurons in hidden layer	Depends on
NNEP	Number of neurons in hidden layer	4			the data set
	Transfer function in each neuron	Product_Unit		Transfer function in each neuron	Product_Unit
	Number of generations	1000		Number of generations	1000
PolCuadraticLMS	-	-	RBFN2	Number of hidden neurons	Depends on
RBFN	Number of hidden neurons	50			the data set

Table 3. Parameters used by the methods.

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

On the other hand, the proposed method is compared with other 6 different methods found in the literature: EvRBF (Evolutionary Radial Basis Function Neural Networks) proposed by Rivas et al. [23], Fuzzy-WM (Fuzzy Rule Learning) by Wang and Mendel [63], NNEP (Neural Network Evolutionary Programming) proposed by Martinez-Estudillo et al. [64], PolCuadraticLMS (LMS Quadratic Regression) by Rustagi [65], RBFN by Broomhead and Lowe [25], and ARIMA proposed by Box and Jenkins and better known as Box-Jenkins models [1]. Nevertheless, as described next, we have used 2 different configurations for NNEP and RBFN, showed in the tables of results with two additional columns.

These methods, except ARIMA, are extracted from Keel[62] 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 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 data set.

Table 3 shows the specific parameter values employed by every method utilized.

In order to work with the eight methods mentioned before, it has been used the Estimated Partial Autocorrelation Function (EPAF). It indicates which intervals of time from data sets are more considered 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 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. A comparison between lags selected by the EPAF and the selected ones by L-Co-R can be found in XXXXXX.

To compare all methods in the same conditions, they were given the data without trend obtained following the process of trend removing, and then, the postprocessing phase were done to get the final results, like in L-Co-R algorithm.

It has been used 5 quality measures: MAE, MAPE, MdAPE, sMdAPE, and MASE in order to show the results obtained (see section 4.1). These quality measures have been estimated by means of forecasting 30 times, using the same training and test sets in any execution.

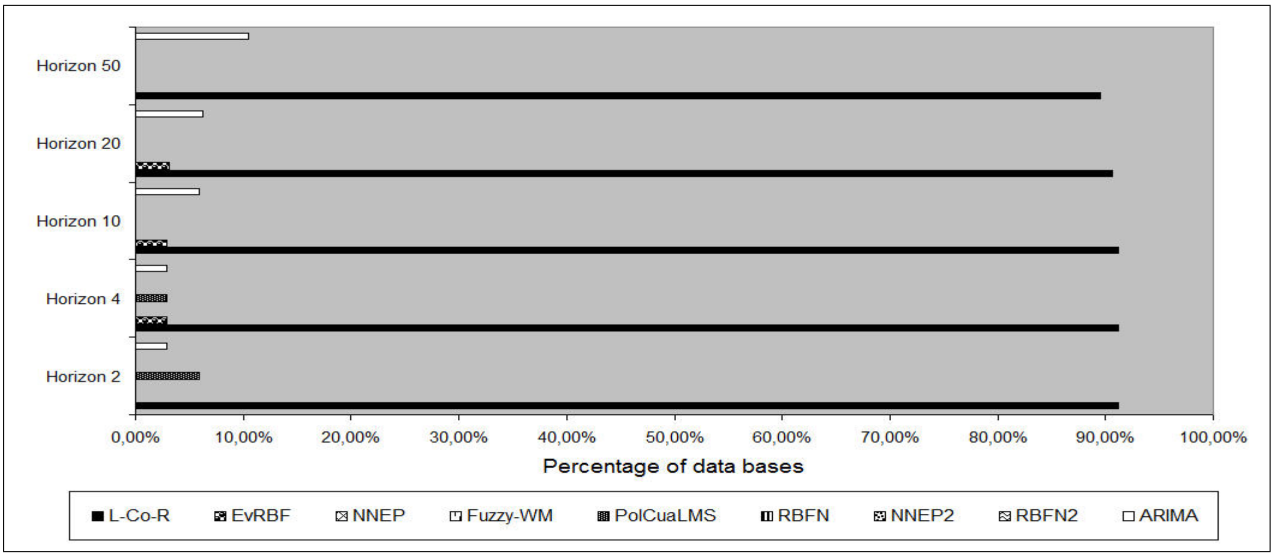

Figure 1. Graphic of the number of time series (expressed as a percentage) in which every algorithm obtains better results than the other methods with respect to MAPE, for horizons 2, 4, 10, 20, and 50.

Because of the limited space, the completed results are available at XXXXXX. It includes the results of all quality measures and the 7 horizons considered in this work.

Figures 1 and 2, graphically show the results of the methods for MAPE and MASE, respectively. Every figure represents the number of data bases (expressed as a percentage) in which every algorithm obtains the best result, distinguishing each horizon individually[2]. As can be observed, L-Co-R achieves the best percentage with every measure and with respect to all horizons. More precisely, L-Co-R gets the best result in more than 90% of time series predicting with horizon 1, 2, 4, 8, 10, and 20, and more than 89% with horizon 50, respecting the quality measure MAPE. With regard to MASE, L-Co-R obtains better results than the other methods in more than 52% of data bases, using any horizon considered.

4.1. Statistical study and conclusions of the experimentation

A statistic study has been done in order to check if the differences among methods are significant for each one of the horizon and quality measures considered, MAE, MAPE, MASE, MdAPE, and sMdAPE[3]. The procedure carried out to do this is the following:

1. Firstly, it is tested if is possible the use of parametric statistical techniques over the sample of results checking the three necessary conditions: independency, normality and homoscedasticity [66, 67]. With respect to normality condition, we applied Shapiro-Wilk test as it is used in work by García et. al [68]. This test confirmed that the condition was not fulfilled therefore a non-parametric test should be used.
2. Friedman and Iman-Davenport tests have been applied in order to study whether significant differences exist among all methods. For all horizons, the statistics of Friedman and Iman-Davenport were 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. A ranking of the method obtained from Friedman test will determine the algorithm which achieves the best classification with a result that is lower than the rest for all measures, so it will taken as the control algorithm.
3. In order to find whether the control algorithm presents statistical differences with regards to the remaining methods in the comparison, we apply the Holm procedure [69], as it is recommend in [68].

[2]The results for the rest horizons can be found in XXXXXX.
[3]The results respecting all quality measures is available at XXXXXX

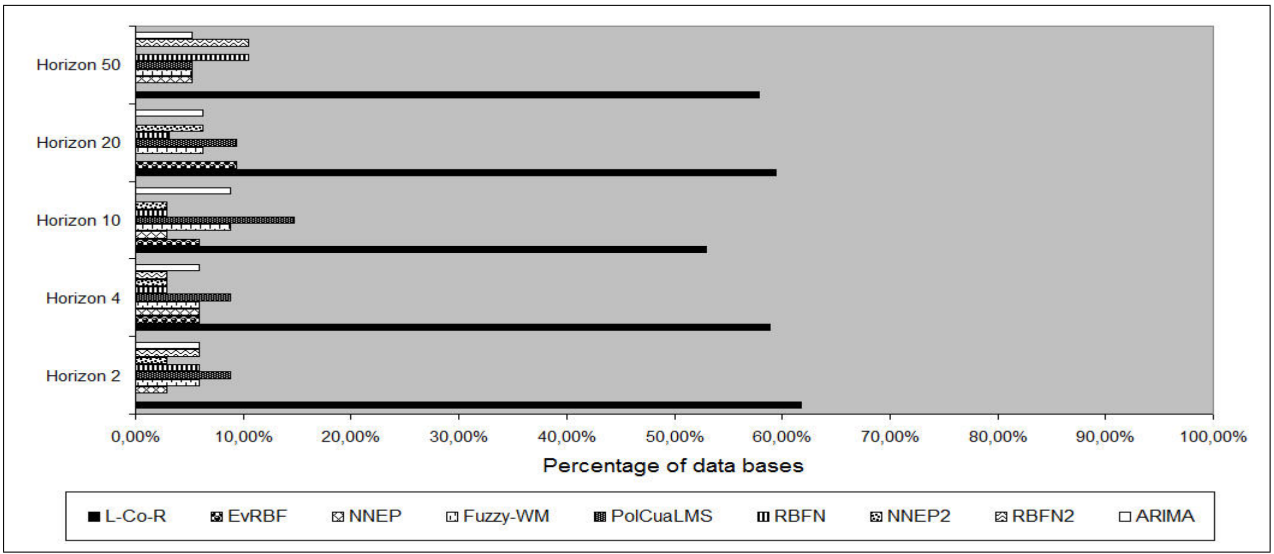

Figure 2. Graphic of the number of time series (expressed as a percentage) in which every algorithm obtains better results than the other methods with respect to MASE, for horizons 2, 4, 10, 20, and 50.

MAE		MAPE		MASE		MdAPE		sMdAPE	
Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking
L-Co-R	**1.647**	**L-Co-R**	**1.441**	**L-Co-R**	**2.412**	**L-Co-R**	**1.265**	**L-Co-R**	**1.918**
ARIMA	3.411	ARIMA	3.029	RBFN	4.235	ARIMA	3.450	ARIMA	3.353
RBFN2	4.264	RBFN	4.412	RBFN2	4.294	RBFN2	4.912	EvRBF	5.147
RBFN	4.382	RBFN2	4.412	PolCua.	4.559	RBFN	5.088	RBFN2	5.412
Fuzzy-WM	5.118	PolCua.	5.412	ARIMA	4.823	NNEP	5.235	RBFN	5.418
PolCua.	5.147	Fuzzy-WM	5.618	Fuzzy-WM	4.882	NNEP2	6.000	NNEP	5.441
NNEP	6.294	NNEP	6.382	NNEP	5.941	Fuzzy-WM	6.147	NNEP2	5.765
NNEP2	6.794	NNEP2	6.676	NNEP2	6.382	PolCua.	6.294	Fuzzy-WM	6.206
EvRBF	7.941	EvRBF	7.618	EvRBF	7.471	EvRBF	6.559	PolCua.	6.353

Table 4. Average rankings of the algorithms (Friedman) for horizon 2.

Tables 4 to 8 show the ranking of the methods obtained by Friedman method. The best method for every error measure is stressed in bold on the top. As can be seen in these tables, the method L-Co-R achieves the best ranking with a result that is lower than the rest for all measures, and for every horizon, so it is taken as the control algorithm.

The results of the Holm procedure, to see whether the control algorithm presents statistical differences with the others algorithms, are shown in tables 9 to 13 with regard to horizon 2, 4, 10, 20, and 50[4], respectively. These tables present all the adjusted *p-values* for each comparison which involves the control algorithm, for MAPE and MASE[5], respectively. The *p-value* is indicated in each comparison considering a level of significance $\alpha = 0.05$, and z as the statistic that compares the i-th and the j-th method.

The Holm procedure compares each *p-value* with α/i, so every α/i indicates if the hypothesis of equal means is rejected. Thus, if *p-value* is less than α/i is possible to assure that there are significant differences.

As it is shown in tables 9, 10, 11, and 12, and in those shown in XXXXXX, there are significant differences between L-Co-R and the remaining methods for all measures used. Therefore, L-Co-R really shows a better behavior with respect to test error comparing to other methods regarding horizons 1, 2, 4, 8, 10, and 20 (see complete results

[4]XXXXXX shows the results for the rest horizons.
[5]The results of all quality measures can be seen at XXXXXX.

MAE		MAPE		MASE		MdAPE		sMdAPE	
Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking
L-Co-R	**1.529**	**L-Co-R**	**1.294**	**L-Co-R**	**2.441**	**L-Co-R**	**1.088**	**L-Co-R**	**1.853**
ARIMA	3.500	ARIMA	3.147	PolCua.	4.265	ARIMA	3.235	ARIMA	3.206
RBFN	4.059	RBFN	4.147	RBFN	4.294	RBFN	4.912	EvRBF	5.029
RBFN2	4.471	RBFN2	4.618	ARIMA	4.824	RBFN2	5.353	NNEP	5.353
PolCua.	5.029	PolCua.	5.471	RBFN2	4.882	NNEP	5.647	RBFN	5.412
Fuzzy-WM	5.059	Fuzzy-WM	5.529	Fuzzy-WM	4.912	Fuzzy-WM	5.941	NNEP2	5.618
NNEP	6.647	NNEP	6.675	NNEP	6.059	NNEP2	6.176	RBFN2	5.765
NNEP2	6.971	NNEP2	6.912	NNEP2	6.118	PolCua.	6.265	Fuzzy-WM	6.176
EvRBF	7.735	EvRBF	7.118	EvRBF	7.206	EvRBF	6.382	PolCua.	6.588

Table 5. Average rankings of the algorithms (Friedman) for horizon 4.

MAE		MAPE		MASE		MdAPE		sMdAPE	
Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking
L-Co-R	**1.559**	**L-Co-R**	**1.206**	**L-Co-R**	**2.353**	**L-Co-R**	**1.029**	**L-Co-R**	**1.706**
ARIMA	3.382	ARIMA	2.918	Fuzzy-WM	4.559	ARIMA	3.265	ARIMA	3.206
RBFN2	4.382	RBFN	4.706	PolCua.	4.588	RBFN	5.235	EvRBF	4.941
RBFN	4.559	RBFN2	4.794	RBFN	4.647	RBFN2	5.235	NNEP	5.294
Fuzzy-WM	5.176	Fuzzy-WM	5.265	RBFN2	4.765	NNEP	5.382	RBFN2	5.529
PolCua.	5.265	PolCua.	5.559	ARIMA	4.765	NNEP2	5.971	NNEP2	5.618
NNEP	6.471	NNEP	6.471	NNEP2	5.824	PolCua.	6.088	RBFN	5.618
NNEP2	6.500	NNEP2	6.765	NNEP	6.265	Fuzzy-WM	6.353	PolCua.	6.412
EvRBF	7.706	EvRBF	7.324	EvRBF	7.235	EvRBF	6.441	Fuzzy-WM	6.441

Table 6. Average rankings of the algorithms (Friedman) for horizon 10.

MAE		MAPE		MASE		MdAPE		sMdAPE	
Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking
L-Co-R	**1.469**	**L-Co-R**	**1.250**	**L-Co-R**	**2.218**	**L-Co-R**	**1.063**	**L-Co-R**	**1.594**
ARIMA	2.906	ARIMA	2.781	RBFN2	4.531	ARIMA	3.156	ARIMA	3.063
RBFN2	4.469	RBFN2	4.531	ARIMA	4.563	RBFN	5.156	EvRBF	4.750
RBFN	4.531	RBFN	4.781	PolCua.	4.718	RBFN2	5.531	NNEP	5.436
Fuzzy-WM	5.406	Fuzzy-WM	5.438	RBFN	4.750	NNEP	5.531	RBFN	5.500
PolCua.	5.656	PolCua.	5.844	Fuzzy-WM	4.969	NNEP2	5.750	NNEP2	5.844
NNEP	6.375	NNEP	6.406	NNEP2	5.906	Fuzzy-WM	5.844	RBFN2	6.031
NNEP2	6.563	NNEP2	6.719	NNEP	6.188	EvRBF	6.313	Fuzzy-WM	6.156
EvRBF	7.625	EvRBF	7.250	EvRBF	7.156	PolCua.	6.656	PolCua.	6.625

Table 7. Average rankings of the algorithms (Friedman) for horizon 20.

MAE		MAPE		MASE		MdAPE		sMdAPE	
Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking	Method	Ranking
L-Co-R	**1.421**	**L-Co-R**	**1.263**	**L-Co-R**	**2.368**	**L-Co-R**	**1.000**	**L-Co-R**	**2.000**
ARIMA	3.158	ARIMA	2.474	PolCua.	3.737	ARIMA	3.053	ARIMA	3.105
RBFN	4.053	RBFN2	4.316	RBFN	4.263	RBFN	4.789	EvRBF	4.105
RBFN2	4.105	RBFN	4.474	RBFN2	4.474	RBFN2	5.263	RBFN	5.263
PolCua.	4.842	PolCua.	4.895	ARIMA	4.579	NNEP	5.526	NNEP	5.474
Fuzzy-WM	5.579	Fuzzy-WM	6.105	Fuzzy-WM	5.579	PolCua.	6.105	RBFN2	5.737
NNEP	6.526	NNEP	6.895	NNEP	6.053	NNEP2	6.211	NNEP2	5.789
NNEP2	7.105	EvRBF	7.105	NNEP2	7.053	EvRBF	6.368	PolCua.	6.421
EvRBF	8.211	NNEP2	7.474	EvRBF	7.895	Fuzzy-WM	6.684	Fuzzy-WM	7.105

Table 8. Average rankings of the algorithms (Friedman) for horizon 50.

Measure	**Alg$_{\text{Control}}$**	***i***	**Algorithm**	***z***	***p***	**α/i**	**Hypothesis**
MAPE	L-Co-R	8	EvRBF	9.299	1.418E-20	0.006	**Rejected**
		7	NNEP2	7.882	3.223E-15	0.007	**Rejected**
		6	NNEP	7.439	1.013E-13	0.008	**Rejected**
		5	Fuzzy-WM	6.288	3.219E-10	0.01	**Rejected**
		4	PolCuaLMS	5.978	2.260E-09	0.013	**Rejected**
		3	RBFN2	4.472	7.736E-06	0.017	**Rejected**
		2	RBFN	4.472	7.736E-06	0.025	**Rejected**
		1	ARIMA	2.391	1.680E-02	0.05	**Rejected**
MASE	L-Co-R	8	EvRBF	7.616	2.611E-14	0.006	**Rejected**
		7	NNEP2	5.978	2.260E-09	0.007	**Rejected**
		6	NNEP	5.314	1.074E-07	0.008	**Rejected**
		5	Fuzzy-WM	3.720	1.996E-04	0.01	**Rejected**
		4	ARIMA	3.631	2.823E-04	0.013	**Rejected**
		3	PolCuaLMS	3.232	1.227E-03	0.017	**Rejected**
		2	RBFN2	2.834	4.597E-03	0.025	**Rejected**
		1	RBFN	2.745	6.044E-03	0.05	**Rejected**

Table 9. Results of Holm's procedure in every comparison between control algorithm and the rest of algorithms for horizon 2.

Measure	$Alg_{Control}$	i	Algorithm	z	p	α/i	Hypothesis
MAPE	L-Co-R	8	EvRBF	8.768	1.825E-18	0.006	**Rejected**
		7	NNEP2	8.458	2.729E-17	0.007	**Rejected**
		6	NNEP	8.236	1.777E-16	0.008	**Rejected**
		5	Fuzzy-WM	6.376	1.813E-10	0.01	**Rejected**
		4	PolCuaLMS	6.288	3.219E-10	0.013	**Rejected**
		3	RBFN2	5.004	5.623E-07	0.017	**Rejected**
		2	RBFN	4.295	1.745E-05	0.025	**Rejected**
		1	ARIMA	2.790	5.276E-03	0.05	**Rejected**
MASE	L-Co-R	8	EvRBF	7.173	7.311E-13	0.006	**Rejected**
		7	NNEP2	5.535	3.111E-08	0.007	**Rejected**
		6	NNEP	5.447	5.136E-08	0.008	**Rejected**
		5	Fuzzy-WM	3.720	1.996E-04	0.01	**Rejected**
		4	RBFN2	3.675	2.376E-04	0.013	**Rejected**
		3	ARIMA	3.587	3.348E-04	0.017	**Rejected**
		2	RBFN	2.790	5.276E-03	0.025	**Rejected**
		1	PolCuaLMS	2.745	6.044E-03	0.05	**Rejected**

Table 10. Results of Holm's procedure in every comparison between control algorithm and the rest of algorithms for horizon 4.

Measure	$Alg_{Control}$	i	Algorithm	z	p	α/i	Hypothesis
MAPE	L-Co-R	8	EvRBF	9.210	3.249E-20	0.006	**Rejected**
		7	NNEP2	8.369	5.808E-17	0.007	**Rejected**
		6	NNEP	7.926	2.259E-15	0.008	**Rejected**
		5	PolCuaLMS	6.554	5.619E-11	0.01	**Rejected**
		4	Fuzzy-WM	6.111	9.917E-10	0.013	**Rejected**
		3	RBFN2	5.402	6.581E-08	0.017	**Rejected**
		2	RBFN	5.269	1.369E-07	0.025	**Rejected**
		1	ARIMA	2.568	1.022E-02	0.05	**Rejected**
MASE	L-Co-R	8	EvRBF	7.351	1.973E-13	0.006	**Rejected**
		7	NNEP	5.889	3.877E-09	0.007	**Rejected**
		6	NNEP2	5.225	1.740E-07	0.008	**Rejected**
		5	ARIMA	3.631	2.823E-04	0.01	**Rejected**
		4	RBFN2	3.631	2.823E-04	0.013	**Rejected**
		3	RBFN	3.454	5.525E-04	0.017	**Rejected**
		2	PolCuaLMS	3.365	7.645E-04	0.025	**Rejected**
		1	Fuzzy-WM	3.321	8.968E-04	0.05	**Rejected**

Table 11. Results of Holm's procedure in every comparison between control algorithm and the rest of algorithms for horizon 10.

Measure	$Alg_{Control}$	i	Algorithm	z	p	α/i	Hypothesis
MAPE	L-Co-R	8	EvRBF	8.764	1.892E-18	0.006	**Rejected**
		7	NNEP2	7.988	1.376E-15	0.007	**Rejected**
		6	NNEP	7.531	5.028E-14	0.008	**Rejected**
		5	PolCuaLMS	6.710	1.952E-11	0.01	**Rejected**
		4	Fuzzy-WM	6.116	9.581E-10	0.013	**Rejected**
		3	RBFN	5.158	2.500E-07	0.017	**Rejected**
		2	RBFN2	4.793	1.647E-06	0.025	**Rejected**
		1	ARIMA	2.237	2.532E-02	0.05	**Rejected**
MASE	L-Co-R	8	EvRBF	7.212	5.527E-13	0.006	**Rejected**
		7	NNEP	5.797	6.762E-09	0.007	**Rejected**
		6	NNEP2	5.386	7.207E-08	0.008	**Rejected**
		5	Fuzzy-WM	4.017	5.904E-05	0.01	**Rejected**
		4	RBFN	3.697	2.181E-04	0.013	**Rejected**
		3	PolCuaLMS	3.651	2.607E-04	0.017	**Rejected**
		2	ARIMA	3.423	6.187E-04	0.025	**Rejected**
		1	RBFN2	3.378	7.312E-04	0.05	**Rejected**

Table 12. Results of Holm's procedure in every comparison between control algorithm and the rest of algorithms for horizon 20.

Measure	$Alg_{Control}$	i	Algorithm	z	p	α/i	Hypothesis
MAPE	L-Co-R	8	NNEP2	6.990	2.754E-12	0.006	**Rejected**
		7	EvRBF	6.575	4.863E-11	0.007	**Rejected**
		6	NNEP	6.338	2.326E-10	0.008	**Rejected**
		5	Fuzzy-WM	5.450	5.048E-08	0.01	**Rejected**
		4	PolCuaLMS	4.087	4.366E-05	0.013	**Rejected**
		3	RBFN	3.613	3.023E-04	0.017	**Rejected**
		2	RBFN2	3.436	5.912E-04	0.025	**Rejected**
		1	ARIMA	1.362	1.731E-01	0.05	**Non reject.**
MASE	L-Co-R	8	EvRBF	6.220	4.982E-10	0.006	**Rejected**
		7	NNEP2	5.272	1.350E-07	0.007	**Rejected**
		6	NNEP	4.146	3.377E-05	0.008	**Rejected**
		5	Fuzzy-WM	3.613	3.023E-04	0.01	**Rejected**
		4	RBFN2	2.369	1.782E-02	0.013	**Rejected**
		3	RBFN	2.132	3.297E-02	0.017	**Rejected**
		2	PolCuaLMS	1.540	1.235E-01	0.025	**Non reject.**
		1	ARIMA	1.362	1.731E-01	0.05	**Non reject.**

Table 13. Results of Holm's procedure in every comparison between control algorithm and the rest of algorithms for horizon 50.

at XXXXXX). Even with the methods NNEP2 and RBFN2, in which the complexity of the initial networks are the same than in L-Co-R, it yielded better results with significant differences.

Regarding horizon 50, only 3 of the 5 quality measures (MAPE, MASE, and sMdAPE) argue that L-Co-R cannot be considered better, significantly, than ARIMA method. And only 1 measure, MASE, says that the results of the L-Co-R method are not significantly better than the PolCuadraticLMS ones.

Finally, taking all statistic studies carried out into account, we can conclude that L-Co-R has a good behavior in time series forecasting with short, medium and long-term horizons, being better in most cases than the rest of the considered algorithms. L-Co-R stands out for its accurate over a large set of sample data, which has 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.

5. Conclusions and Future research

In this paper the effectiveness of the L-Co-R method, a coevolutionary algorithm for time series forecasting, for long-time forecasting and with a changing horizon environment is tested. Two different populations coevolve to obtain future values predictions whatever the period given: short, medium or long term. On one hand, a population of RBFNs evolves sets of neural networks to get an appropriate network architecture. On the other hand, a population of time lags evolves sets of important lags, which will be utilized to make future predictions. In order to implement the coevolution, both individuals of lag population and individuals of RBFNs population can cooperate together to produce a global solutions.

To test the performance of L-Co-R forecasting with a variable horizon, 34 different time series has been used. The results of L-Co-R has been compared, regarding 5 different quality measures (MAE, MAPE, MASE, MdAPE, and sMdAPE) and for 7 considered horizons (1, 2, 4, 8, 10, 20 and 50), with other 6 methods found in the literature.

In order to conclude about the obtained results, a statistic study has been carried out. First of all, it has been used Friedman and Iman-Davenport tests to see if the observed differences between methods are significant, and next, to find whether the control algorithm (L-Co-R in all cases) presents statistical differences regarding the rest of the methods, Holm procedure is applied.

Thus, we can conclude that L-Co-R achieves better results than the other methods, taking into account the large set of time series and the context of variable horizon.

Future works should include the estimation of the time interval which provides the future value. In this way, a couple of values, minimum and maximum, could be indicated for each prediction, accompanied by a confidence value. Further study could also includes exogenous features and selection processes of them.

Acknowledgements

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

[1] G. Box, G. Jenkins, Time series analysis: forecasting and control, San Francisco: Holden Day, 1976.
[2] J. De Gooijer, R. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (3) (2006) 443–473.
[3] R. Fildes, K. Nikolopoulos, S. Crone, A. Syntetos, Forecasting and operational research: a review, Journal of the Operational Research Society 59 (2008) 1150–1172.
[4] R. Brown, Statistical forecasting for inventory control, McGraw-Hill, 1959.
[5] P. Winters, Forecasting sales by exponentially weighted moving averages, Management Science 6 (3) (1960) 324–342.
[6] R. Snyder, Recursive estimation of dynamic linear models, Journal of the Royal Statistical Society. Series B (Methodological) 47 (2) (1985) 272–276.
[7] H. Tong, On a threshold model, Pattern recognition and signal processing, NATO ASI Series E: Applied Sc. 29 (1978) 575–586.
[8] H. Tong, Threshold models in non-linear time series analysis, Springer-Verlag, 1983.
[9] K. Chan, H. Tong, On estimating thresholds in autoregressive models, Journal of Time Series Analysis 7 (3) (1986) 179–190.
[10] P. Brockwell, R. Hyndman, On continuous-time threshold autoregression, International Journal of Forecasting 8 (2) (1992) 157–173.
[11] M. Clements, P. Franses, N. Swanson, Forecasting economic and financial time-series with non-linear models, International Journal of Forecasting 20 (2) (2004) 169–183.

[12] B. Samanta, Prediction of chaotic time series using computational intelligence, Expert Systems with Applications 38 (9) (2011) 11406–11411.
[13] S. Zhu, J. Wang, W. Zhao, J. Wang, A seasonal hybrid procedure for electricity demand forecasting in china, Applied Energy 88 (11) (2011) 3807–3815.
[14] W. Qiu, X. Liu, H. Li, A generalized method for forecasting based on fuzzy time series, Expert Systems with Applications 38 (8) (2011) 10446–10453.
[15] C. Wang, A comparison study between fuzzy time series model and arima model for forecasting taiwan export, Expert Systems with Applications 38 (8) (2011) 9296–9304.
[16] T. Yu, K. Huarng, A neural network-based fuzzy time series model to improve forecasting, Expert Systems with Applications 37 (4) (2010) 3366–3372.
[17] K. Kavaklioglu, Modeling and prediction of turkey's electricity consumption using support vector regression, Applied Energy 88 (1) (2011) 368–375.
[18] P. K. Dash, A. C. Liew, S. Rahman, G. Ramakrishna, Building a fuzzy expert system for electric load forecasting using a hybrid neural network, Expert Systems with Applications 9 (3) (1995) 407–421.
[19] Z. Tang, C. de Almeida, P. Fishwick, Time series forecasting using neural networks vs. box-jenkins methodology, Simulation 57 (5) (1991) 303–310.
[20] G. Zhang, B. Patuwo, M. Hu, Forecasting with artificial neural networks: The state of the art, International Journal of Forecasting 14 (1) (1998) 35–62.
[21] A. Jain, A. Kumar, Hybrid neural network models for hydrologic time series forecasting, Applied Soft Computing 7 (2) (2007) 585–592.
[22] C. M. Arizmendi, J. Sanchez, N. E. Ramos, G. I. Ramos, Time series predictions with neural nets: Application to airborne pollen forecasting, International Journal of Biometeorology 37 (3) (1993) 139–144.
[23] V. Rivas, J. Merelo, P. Castillo, M. Arenas, J. Castellano, Evolving rbf neural networks for time-series forecasting with evrbf, Information Sciences 165 (3-4) (2004) 207 – 220.
[24] A. Bezerianos, S. Papadimitriou, D. Alexopoulos, Radial basis function neural networks for the characterization of heart rate variability dynamics, Artificial Intelligence in Medicine 15 (3) (1999) 215–234.
[25] D. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321–355.
[26] B. Carse, T. Fogarty, Fast evolutionary learning of minimal radial basis function neural networks using a genetic algorithm, in: Proceedings of Evolutionary Computing, Vol. 1143 of Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 1996, pp. 1–22.
[27] B. Whitehead, T. Choate, Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction, IEEE Transactions on Neural Networks 7 (4) (1996) 869–880.
[28] C. Harpham, C. Dawson, The effect of different basis functions on a radial basis function network for time series prediction: A comparative study, Neurocomputing 69 (16-18) (2006) 2161–2170.
[29] H. Du, N. Zhang, Time series prediction using evolving radial basis function networks with new encoding scheme, Neurocomputing 71 (7-9) (2008) 1388–1400.
[30] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Computers & Operations Research 33 (3) (2006) 859–871.
[31] H. Hippert, J. Taylor, An evaluation of bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting, Neural Networks 23 (3) (2010) 386–395.
[32] C. Lee, C. Ko, Time series prediction using rbf neural networks with a nonlinear time-varying evolution pso algorithm, Neurocomputing 73 (1-3) (2009) 449–460.
[33] M. Perez-Godoy, P. Pérez-Recuerda, M. Frías, A. Rivera, C. Carmona, M. Parras, Co^2rbfn for short and medium term forecasting of the extra-virgin olive oil price, in: J. González, D. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), Proceedings of Nature Inspired Cooperative Strategies for Optimization, Vol. 284, Springer Berlin/Heidelberg, 2010, pp. 113–125.
[34] F. Takens, Dynamical Systems and Turbulence, Lecture Notes In Mathematics, Vol. 898, Springer, New York, NY, 1980, Ch. Detecting strange attractor in turbulence, pp. 366–381.
[35] T. Ferreira, G. Vasconcelos, P. Adeodato, A new intelligent system methodology for time series forecasting with artificial neural networks, Neural Processing Letters 28 (2) (2008) 113–129.
[36] K. Lukoseviciute, M. Ragulskis, Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems, Neurocomputing 73 (10-12) (2010) 2077–2088.
[37] R. Araújo, A quantum-inspired evolutionary hybrid intelligent apporach fo stock market prediction, International Jorunal of Intelligent Computing and Cybernetics 3 (10) (2010) 24–54.
[38] R. Araújo, Hybrid intelligent methodology to design translation invariant morphological operators for brazilian stock market prediction, Neural Networks 23 (10) (2010) 1238–1251.
[39] R. García-Pajares, J. Benitez, G. Sainz Palmero, Feature selection form time series forecasting: a case study, in: Proceedings of 8th International Conference on Hybrid Intelligent Systems, 2008, pp. 555–560.
[40] A. Maus, J. C. Sprott, Neural network method for determining embedding dimension of a time series, Communications in Nonlinear Science and Numerical Simulation 16 (8) (2011) 3294–3302.
[41] M. Potter, K. De Jong, A cooperative coevolutionary approach to function optimization, in: Proceedings of Parallel Problem Solving from Nature, Vol. 866 of Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 1994, pp. 249–257.
[42] M. Potter, K. De Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, Evolutionary Computation 8 (1) (2000) 1–29.
[43] R. Wiegand, W. Liles, K. De Jong, An empirical analysis of collaboration methods in cooperative coevolutionary algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2001, pp. 1235–1242.
[44] L. Panait, R. Wiegand, S. Luke, Improving coevolutionary search for optimal multiagent behaviors, in: Proceedings of the International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2003, pp. 653–658.
[45] C. Au, H. Leung, Biasing mutations in cooperative coevolution, in: Proceedings of IEEE Congress on Evolutionary Computation, 2007, pp.

828–835.
[46] K. Tan, Y. Yang, C. Goh, A distributed cooperative co-evolutionary algorithm for multi-objective optimization, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 527–549.
[47] N. García-Pedrajas, J. R. del Castillo, D. Ortiz-Boyer, A cooperative coevolutionary algorithm for instance selection for instance-based learning, Machine Learning 78 (3) (2010) 381–420.
[48] J. Derrac, S. García, F. Herrera, Ifs-coco: Instance and feature selection based on cooperative coevolution with nearest neighbor rule, Pattern Recognition 43 (6) (2010) 2082–2105.
[49] N. García-Pedrajas, C. Hervas-Martínez, D. Ortiz-Boyer, Cooperative coevolution of artificial neural network ensembles for pattern classification, IEEE Transactions on Evolutionary Computation 9 (3) (2005) 271–302.
[50] M. Li, J. Tian, F. Chen, Improving multiclass pattern recognition with a co-evolutionary rbfnn, Pattern Recognition Letters 29 (4) (2008) 392–406.
[51] X. Ma, H. Wu, 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, 2010, pp. 582–585.
[52] M. Qian-Li, Z. Qi-Lun, P. Hong, Z. Tan-Wei, Q. Jiang-Wei, Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network, Chinese Physics B 17 (2).
[53] B. Bowerman, R. O'Connell, A. Koehler, Forecasting: methods and applications, Thomson Brooks/Cole: Belmont, CA., 2004.
[54] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler, The accuracy of extrapolation (time series) methods: Results of a forecasting competition, Journal of Forecasting 1 (2) (1982) 111–153.
[55] J. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, International Journal of Forecasting 8 (1) (1992) 69–80.
[56] R. Fildes, The evaluation of extrapolative forecasting methods, International Journal of Forecasting 8 (1) (1992) 81–98.
[57] S. Makridakis, M. Hibon, The m3-competition: results, conclusions and implications, International Journal of Forecasting 16 (4) (2000) 451–476.
[58] R. Hyndman, A. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting 22 (4) (2006) 679–688.
[59] G. Zhang, M. Qi, Neural network forecasting for seasonal and trend time series, European Journal of Operational Research 160 (2) (2005) 501–514.
[60] L. Eshelman, The chc adptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, in: Proceedings of 1st Workshop on Foundations of Genetic Algorithms, 1991, pp. 265–283.
[61] D. Wichern, R. Jones, Assessing the impact of market disturbances using intervention analysis, Management Science 24 (3) (1977) 329–337.
[62] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera, Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing - A Fusion of Foundations, Methodologies and Applications 13 (3) (2009) 307–318.
[63] L. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man and Cybernetics 22 (6) (2002) 1414–1427.
[64] A. Martínez-Estudillo, F. Martínez-Estudillo, C. Hervás-Martínez, N. García-Pedrajas, Evolutionary product unit based neural networks for regression, Neural Networks 19 (4) (2006) 477–486.
[65] J. Rustagi, Optimization techniques in statistics, Academic Press (Boston), 1994.
[66] D. Sheskin, Handbook of parametric and nonparametric statistical procedures, Chapman & Hall/CRC, 2004.
[67] J. Zar, Biostatistical analysis, Prentice Hall, Englewood Cliffs, 1999.
[68] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.
[69] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (2) (1979) 65–70.
[70] E. Parras-Gutierrez, M. Garcia-Arenas, V.M. Rivas, and M.J. del Jesus, Coevolution of lags and RBFNs for time series forecasting: L-Co-R algorithm, Soft Computing, 16 (6) (2012) 919–942.