Neurocomputing 218 (2016) 37–50

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Dynamic selection of forecast combiners

Anderson T. Sergio*, Tiago P.F. de Lima, Teresa B. Ludermir

Informatics Center, Federal University of Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 17 February 2016
Received in revised form
29 June 2016
Accepted 14 August 2016
Communicated by W.S. Hong
Available online 28 August 2016

Keywords:
Time series forecasting
Dynamic selection
Chaotic time series
Time series forecasting ensemble

ABSTRACT

Time series forecasting is an important research field in machine learning. Since the literature shows several techniques for the solution of this problem, combining outputs of different models is a simple and robust strategy. However, even when using combiners, the experimenter may face the following dilemma: which technique should one use to combine the individual predictors? Inspired by classification and pattern recognition algorithms, this work presents a dynamic selection method of forecast combiners. In the dynamic selection, each test pattern is submitted to a certain combiner according to a nearest neighbor rule. The proposed method was used to forecast eight time series with chaotic behavior in short and long term. In general, the dynamic selection presented satisfactory results for all datasets.

1. Introduction

Time Series Forecasting (TSF) is one of the most traditional problems in statistics and machine learning. In this kind of problem, past values of a given measurement are collected and subsequently used in forecasting future values. Over the years, various machine learning techniques have been used for this purpose, e.g. Artificial Neural Networks (ANN) [1], Support Vector Machines (SVM) [2], fuzzy logic techniques [3] and hybrid systems of some of these techniques with evolutionary computation [4] or swarm intelligence [5]. Although older, statistical models are still studied and used with satisfactory results. Among these predictors, one can find linear models such as ARMA (Autoregressive Moving Average) and ARIMA (Autoregressive Integrated Moving Average) [6], and non-linear, as ARCH (Autoregressive Conditional Heteroscedasticity) and GARCH (Generalized ARCH) [7]. Literature shows various real-world applications of TSF in several areas of human activity: energy [8], financial market [9], meteorology [10], epidemiology [11], space weather [12], traffic control [13], seismic activity [14] and so on.

An important class of time series is the so-called chaotic time series. Chaos theory is the mathematical research field that studies the behavior of chaotic dynamical systems [15]. Dynamic systems are highly sensitive to initial conditions, being this property popularly known as butterfly effect. Time series based on chaotic dynamic systems are an important dataset for benchmark models and practical applications such as signal processing [16], astronomy [17] and biomedicine [18].

Given to the wide variety of techniques for time series forecasting, the question of which one to use naturally emerges in the early stages of the problem resolution. In addition, taking into account the no-free-lunch theorem [19], there is no guarantee that a particular model has good performance in all or even more than one dataset category. A possible solution to this scenario is the forecast combination. Forecast combination has been used effectively since the seminal work of Bates and Granger [20]. Simple statistical measures can be used to combine forecasts generated by an ensemble of different models, such as average, median or trimming average [21]. More sophisticated models can be used to combine the ensemble outputs, since statistical combiners do not work when the predictors have similar performances [22]. Moreover, the combination may also be performed in a non-linear fashion as seen in the work of Adhikari and Agrawal [23] and Gheyas and Smith [24]. The foundation for the use of non-linear combiners is the fact that linear combiners only consider individual contributions from each predictor, but not their relationship.

Although the use of combiners normally improves substantially the performance of individual predictors, the great amount of available methods raises a fundamental question: which combiner should one use? There is no formal indication of what methods are best in what situations. The answer to this question may lie in the dynamic selection. The dynamic selection is an emerging area in machine learning, with many papers published in recent years (a survey of the main contributions can be seen in [25]). In most cases, dynamic selection is used in classification and pattern recognition problems. In general, the procedure for dynamic

* Corresponding author.
E-mail addresses: ats3@cin.ufpe.br (A.T. Sergio), tpfl2@cin.ufpe.br (T.P.F. de Lima), tbl@cin.ufpe.br (T.B. Ludermir).

http://dx.doi.org/10.1016/j.neucom.2016.08.072
0925-2312/© 2016 Elsevier B.V. All rights reserved.

selection can be shortened in three stages: generation, selection and integration. In generation, a set (or ensemble) of classifiers is generated. In selection, a subset of these classifiers is dynamically selected according to some criteria. Finally, in the integration phase, a final decision is made in respect to which selected classifiers will be used for the classification of a particular input pattern.

This paper proposes a method of dynamic selection of forecast combiners. Due to their importance in benchmarking and in human activities, time series with chaotic behavior were used as dataset in the experiments, for short and long term prediction. Four models were used as predictors in the generation phase of dynamic selection: a feedforward neural network with one hidden layer, a feedforward neural network with two hidden layers, a DBN (Deep Belief Network) [26] and an SVR (Support Vector Regression) [27]. Average, median and the softmax function were used as combiners. The details of the proposed dynamic selection will be explained in the following sections. It is important to note that there is no work in the literature that deals with the dynamic selection of forecast combiners in a similar way. Another important point is that the built method is independent of the individual models and selected combiners, being a framework for dynamic selection.

This work is organized as follows: Section 2 is a review of the main and the latest methods of combining predictors. Section 3 shows how dynamic selection can improve the performance of individual models. Section 4 presents the proposed method of dynamic selection of forecast combiners. Section 5 describes the experiments. Section 6 shows the results of the experiments, followed by discussion. Finally, Section 7 presents the conclusions and proposals for future work.

2. Time series forecast combination

The instability of a given predictor can be mitigated when an ensemble is used to generate the final prediction, since mistakes can be smoothed. A theoretical justification for the forecast combination from a Bayesian model can be seen in [28].

The simplest way to combine predictions is the linear combination of the predictors. Let $Y = \{y_1, y_2, \ldots, y_N\}$ and $\hat{Y}^{(i)} = \left\{ \hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \ldots, \hat{y}_N^{(i)} \right\}$ be the actual time series and the forecasts from the ith method, respectively. According to [22], the time series obtained from a linear combination of these n series is provided by the Eq. (1):

$$\begin{cases} \hat{Y}^{(c)} = \left\{ \hat{y}_1^{(c)}, \hat{y}_2^{(c)}, \ldots, \hat{y}_N^{(c)} \right\} \\ \hat{y}_k^{(c)} = w_1\hat{y}_k^{(1)} + w_2\hat{y}_k^{(2)} + \cdots + w_n\hat{y}_k^{(n)} = \sum_{i=1}^{N} w_i\hat{y}_k^{(i)} \\ \forall\, k = 1, 2, \ldots, N \end{cases} \tag{1}$$

where w_i is the weight associated to the ith forecasting method. The weights usually add up to unity, avoiding bias.

In general, the existing methods of linear combination in literature vary in how these weights are calculated. Some of these combinations are performed with a simple arithmetic calculation on all or some of the individual forecasts. This applies to the simple average in which equal weights are assigned to all models, i.e., $w_i = 1/n(i = 1, 2, \ldots, n)$. Although simple, this combination has proven to be a very robust method, sometimes used as a minimum performance measure for more sophisticated combiners. Similarly, the combination can be made with other statistical measures, such as median, maximum or minimum value.

Andrews et al. used the average to combine the predictions of an ensemble consisting of neural networks and Gaussian and linear regression models, in a competition time series [29]. Also, Lian et al. combined the outputs of an ensemble of ELM (Extreme Learning Machines) with an average, trying to predict a landslide index [30].

The trimming average differs from the simple average in a way that the arithmetic mean is calculated excluding $k\%$ of the worst performing models. According to [21], the recommended value of k ranges from 10% to 30%. The use of the trimming average can be seen in [31]. This combination requires the performance of the individual models in some validation dataset. This concept is followed in the methods known as error-based in which the weights are chosen inversely proportional to past performance [32] and outperformance when the weights are calculated according to the number of times a particular method has been better in the past [33]. Other combination following this direction is the softmax, calculated according to Eqs. (2) and (3):

$$f_i'' = \frac{f_i' - \min(f')}{\max(f') - \min(f')} \tag{2}$$

$$w_{smi} = \frac{e^{(f''_i)}}{\sum_{k=1}^{N} e^{(f''_k)}} \tag{3}$$

where f' is the reverse of the forecast error in a validation dataset, $f_i' = 1/f_i$ and $\min(f')$ and $\max(f')$ are the minimum and maximum values of all f'. The use of softmax and median as forecast combiners can be seen in [4].

According to [22], statistical combiners do not work when the predictors have similar performance. Along these combinations with relatively little computational effort, literature shows some other methods built with more sophistication. Adhikari, for example, proposed a new linear combination that sets the weights by the analysis of patterns in successive forecasts in a validation dataset [22]. Nonlinear combinations, although more uncommon, may also be found. Gheyas and Smith built an ensemble of hybrid models of neural networks and linear regression called GRNN [24]. The output of several GRNN for each subfeature of the time series is presented for a second GRNN training. The work of Adhikari and Agrawal [23] extends the linear combination model of Rodrigues and Freitas [34] to calculate the weights in a non-linear manner.

An ensemble tends to achieve good results when the models that comprise it have a good degree of diversity, providing guarantees against a limited range [29]. There are several ways to generate this diversity, such as using different models, different specifications of the same model, different types of data pre-processing and different input variables. The input data, for example, may undergo a bootstrapping or bagging and cross-validation, as seen in [35]. Zhang added noise to the input data and formed distinct training sets [36]. Andrawis et al. pre-processed the time series, removing the trend [29]. Of course, a combination of these techniques is possible. That is the case of the work presented in this article.

It is important to note that combining forecasts is not the only way to increase the performance of the models. Other approaches can be considered. One of them can be seen in the work of Crone e Kourentzes [37]. In this paper, the authors propose a feature selection method in order to automatically set the best configuration of feedforward neural networks. In the context of time series forecasting, features selection implies the use of a technique to choose which lags must be taken into account in the model training.

The use of multiple datasets can also increase the performance of the forecasts. In this case, the selection of a particular set can coevolve with the construction of the predictor. Mirmomeni e Punch [38] propose an evolutionary approach to model the dynamics of chaotic time series. The population of solutions comprises both the candidate models and the dataset.

3. Dynamic selection

Consider a classification or prediction problem. Let $C = \{h_1, h_2, \ldots, h_L\}$ be a set of L experts and $E = \{e_1, e_2, \ldots, e_M\}$ be a set of M ensembles formed from C. The dynamic selection can be seen as a division of the feature space in $K > 1$ competence regions, denoted by $R_1, R_2, \ldots, R_K$. So, for each region R_j, $j = 1, 2, \ldots, K$, an ensemble from E which has the highest accuracy in R_j is designated. Fig. 1 presents an illustration of the feature space division into four competence areas.

Let $e^* \in E$ be the ensemble with the highest average accuracy over the whole feature space. Denote by $P(e_i|R_j)$ the probability of correct classification by ensemble e_i in region R_j. Consider $e_{i(j)}$ the ensemble designated for region R_j. The overall probability of correct classification of the system is described in Eq. (4), where $p(R_j)$ is the probability of a pattern $\vec{x}$ belonging to R_j. To maximize p_c, $e_{i(j)}$ must be assigned according Eq. (5)

$$p_c = \sum_{j=1}^{K} p(R_j)p_c(R_j) = \sum_{j=1}^{K} p(R_j)P(e_{i(j)}|R_j) \tag{4}$$

$$p(e_{i(j)}|R_j) \ge p(e_t|R_j),\ t = 1, 2, \ldots, M \tag{5}$$

$$\sum_{j=1}^{K} p(R_j)P(e_{i(j)}|R_j) \ge \sum_{j=1}^{K} p(R_j)p(e^*|R_j) \tag{6}$$

From Eqs. (4) and (5), (6) shows that the combined scheme performs equally or better than the best ensemble e^*, regardless of the way the feature space has been partitioned.

As explained above, the dynamic selection process can be summarized in three steps. The first one is responsible for the generation of the base experts set, and this set can be formed by models of the same kind or heterogeneous ones. The diversity of the experts is important in both cases.

The second phase (selection) is carried out by estimating the competence of the available models in the set generated in the first phase, with respect to local regions of the feature space. In the case of dynamic selection, the model selection is performed for each test pattern, instead of using the same selection for all of them (static selection). It is common, for example, to use an NN-rule based schema to define the neighborhood of an unknown pattern in the test phase. Britto et al. [25] propose a taxonomy for the various competence measurements found in literature. According to this taxonomy, the model selection is divided between the use of measures based on the individual and measures that combine the accuracy of base experts with any information related to the interaction between them (group-based). In the first case, the measures may be based on ranking, accuracy, probability, behavior or oracle. Concerning the group-based measurements, it may be based on diversity, complexity and ambiguity.

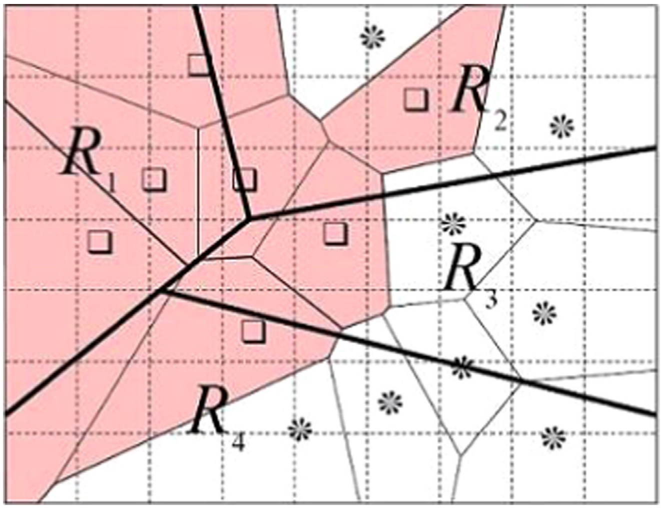


Fig. 1. Partitioning of the feature space in competence regions.

The third phase is the integration of the selected models. Literature shows several ways to accomplish this step. A proposed taxonomy can be seen in [39].

Literature presented in this section is predominantly applied to classification and pattern recognition problems. Despite the lack of work on dynamic selection in time series forecasting, some research with similar bias can be found in literature. Some of these researches are dealing with model selection. Model selection means selecting, from the data, a specific model for task completion. However, in time series forecasting problem, this process is usually accomplished in a static manner [40].

4. Dynamic selection of forecast combiners

This article proposes a method of dynamic selection of forecast combiners. In general, the goal of the method is to generate individual predictors, combine them and select which combination is most promising for each of the test patterns. At this point, one must define what a test pattern is.

As explained in previous sections, a time series can be formalized as a sequence of random scalar observations $Y = \{y_1, y_2, \ldots, y_N\}$. The lag of the series is given for the delay used to form the training and testing patterns. Predicting a time series involves discovering a future value for the sequence, given by $\hat{Y}_{1+1} = F[y_i, y_{i-k}, \ldots, y_{i-(d-1)k}]$. d is the lag, k is the step lag and F the used model. Thus, the dimension of the training and testing patterns are directly related to the delay used. The dynamic selection is, therefore, to select the best combination for each of these patterns. The proposed method will be explained below, according to each phase of the dynamic selection, whenever applicable.

4.1. Generation

In the generation of individual predictors, a considerable degree of diversity was obtained through two manners. The first way was the use of heterogeneous models. The models: a feedforward neural network with one hidden layer (FANN-1), a feedforward network with two hidden layers (FANN-2), a Deep Learning neural network called Deep Belief Network (DBN) with two hidden layers and a support vector machine for regression (SVR). Four predictors were used because there is an indication in literature to use up to a maximum of five models to reduce prediction errors, since more predictors may decrease dramatically the combination accuracy [41].

To generate diversity in ensembles composed of neural networks, various steps can be taken [42]. One of these mensures is to use models with different architectures. This is the rationale for using a neural network with different hidden layers. Training neural networks with more than one hidden layer came to be of great interest to the machine learning community after the appearance of the field known as Deep Learning [43].

Most of the models known as Deep Learning neural networks share the following characteristics: unsupervised learning of the data representations to pre-train each of the layers; unsupervised training for each layer, where the input to the next layer is the representation learned at each previous level; using supervised training for fine tuning of all of the pre-trained layers and one or more additional layers dedicated to produce predictions. These characteristics can also generate diversity in the ensemble, since Deep Learning models have a particular way to initialize the network weights, in contrast to the use of random weights in conventional architectures. Thus, the model learning known as Deep Belief Network (DBN) was used as an individual predictor. The use of Deep Belief Networks in time series forecasting can be seen in [44].

The Support Vector Machines (SVM) was also used as an individual predictor. The version of SVM for regression problems, named SVR, is an important solution for time series forecasting. An example of its use can be seen in [2].

The use of some strategy for generation of the individual predictors was necessary, since setting the free parameters of neural networks and support vector machines before training is difficult and usually problem oriented. A known solution to this issue is to use some optimization algorithm. In the dynamic selection described in this section, there was used the PSO (Particle Swarm Optimization) for this purpose. An example of using PSO to optimize a neural network to time series forecasting can be seen in [5].

In addition to using heterogeneous individual predictors, diversity was also reached with the use of cross-validation in the training data. Thus, the individual predictors were generated with different training sets.

4.2. Selection

In the proposed method, the selection phase is responsible for choosing which combination of individual predictors is most promising for each test pattern. Three statistical combiners were selected, namely: average, median and softmax. Such combiners were selected over other more sophisticated combiners because of their simplicity and low computational cost.

To estimate the competence of each of the combiners, we have developed an algorithm inspired in dynamic selection of classifiers. Giacinto and Roli proposed an algorithm called DS-MCB (Dynamic Selection - Multiple Classifier Behavior) [45]. DS-MCB uses a function to measure the similarity between the test pattern and the available classifiers after selection. Each test pattern presented to the model is associated with an array of labels calculated for each of the base classifiers. After this phase, k nearest neighbors of the test pattern in the training dataset are found. Then, a similarity function is used to find samples of this local region that have similar behavior to the test pattern. The classifier that has better performance in these filtered training patterns calculates the model output for the test pattern, thus characterizing a dynamic selection. Inspired by DS-MCB, we proposed the DS-FC (Dynamic Selection of Forecast Combiners). DS-FC is described by Algorithm 1.

The DS-FC algorithm input is composed by the set of prediction combiners, the training and test dataset and neighborhood size k, being the output the most promising combiner for each test pattern presented to the model. For each test pattern t, the vector $PRED_t$ is computed as the prediction t for each of the combiners. Then, the set of k nearest training patterns is defined. A subset of this cluster is calculated from the similarity degree among $PRED_t$ and the combiners forecasts to these training patterns ($PRED_{\psi_j}$). The similarity function Sim is given by the Euclidean distance between $PRED_t$ and $PRED_{\psi_j}$. Taking into account this patterns subset, the combiners produce their forecasts. The combiner with better performance for the test pattern is selected.

Algorithm 1. DS-FC.

```
INPUT: the set of combiners $C$; the datasets $T_r$ (training) and $T_e$ (testing); the neighborhood size $K$;
OUTPUT: $c_t^*$, the most promising combiner for each unknown pattern $t$ in $T_e$;
for each test pattern $t \in T_e$ do
    Compute the vector $PRED_t$ as the forecast to $t$ by all combiners in $C$;
    Find $\Psi$ as the $K$ nearest neighboors of the test pattern $t$ in $T_r$;
    for each pattern $\psi_j \in \Psi$ do
        Compute $PRED_{\psi_j}$ as the forecast assigned to $\psi_j$ by all combiners in $C$;
        Compute $Sim$ as the similarity between $PRED_t$ and $PRED_{\psi_j}$;
    if $(Sim > SimilarityThreshold)$ then
            $\Psi' = \Psi' \cup \psi_j$;
        end if
    end for
    for each combiner $c_i \in C$ do
        Calculate $PRED_i$ as the forecast assigned to $c_i$ in $\Psi'$;
    end for
    Select the best combiner $c_t^* = argmax_i\{PRED_i\}$;
end for
```

The integration phase does not apply to the proposed method. This is due to the fact that the final prediction is given only through the single combiner dynamically selected for each test pattern.

5. Experiments

This section describes the experiments. First, the datasets used will be presented. Then, the methodology of the experiments is shown as well as the parameters used.

5.1. Datasets

The dynamic selection proposed in this paper was mostly applied to the problem of predicting chaotic time series. As shown above, the importance of the chaotic series study undergoes fields such as astronomy and signal processing, being an important benchmark for forecasting models. Five artificial time series were used for evaluating the proposed method: Laser, Lorenz, Mackey-Glass, Henon, and Rossler, each one with 1000 points. To test the method in real time series, three datasets were used: EEG1 e EEG2 (each one with 1000 points, also) e NOAA (with 1680 points). The datasets were constructed from a sampling rate equals to 1.

All series were normalized to lie within the interval $[0, 1]$ and divided into three sets: training (70% of the points), validation (20% of the points), and testing (10% of the points). Next, the tested datasets are described:

5.1.1. Mackey-glass

The Mackey-Glass series, continuous, one-dimensional and standard benchmark for time series forecasting test is formed by the Eq. (7):

$$\frac{dx}{dt} = \beta\frac{x_\tau}{1 + x_\tau^n} - \gamma x, \gamma, \beta, n > 0 \tag{7}$$

where β, τ, γ and n are real numbers and x_τ represents the value of the variable x in time $(t-\tau)$. The chaotic dynamic appears when $\tau > 16.8$. The following parameters were used: $\tau=17$, $\beta=2$, $\gamma=0.1$ and $n=10$.

5.1.2. Lorenz

The Lorenz time series, introduced in [15], is given by Eq. (8):

$$\frac{dx}{dt} = \sigma[y - x]$$
$$\frac{dy}{dt} = rx - y - xz$$
$$\frac{dz}{dt} = xy - bz \tag{8}$$

where the following parameters were used in the experiments: $\sigma=10$, $r=28$ and $b=8/3$.

5.1.3. Rossler

The Rossler time series, introduced in [46], is given by Eq. (9):

$$\frac{dx}{dt} = -z - y$$
$$\frac{dy}{dt} = x + ay$$
$$\frac{dz}{dt} = b + z(x - c) \tag{9}$$

where the following parameters were used in the experiments: $a=0.15$, $b=0.2$ and $c=10$.

5.1.4. Henon

The Henon map is given by Eq. (10) [47]:

$$x_{n+1} = y_n + 1 - \alpha x_n^2$$
$$y_{n+1} = \beta x_n \tag{10}$$

where the following parameters were used in the experiments: $\alpha=1.4$e $\beta=0.3$.

5.1.5. Laser

Laser is an univariate time series obtained from measurements collected in a physics lab. The data is a cross section of a regular intensity laser, where the pulses generated follow a pattern similar to the theoretical model of Lorenz. The series is used as a time series forecasting benchmark due to its simplicity and well documented and understandable standards. The data were obtained in [48].

5.1.6. EEG1 e EEG2

The electroencephalogram, known as EEG, is a medical test that analyzes the instant brain activity of individuals, usually captured by electrodes. The relationship between electroencephalogram time series with dynamic systems can be seen in many works, as in [18]. The series were obtained from tests in laboratory mice [49].

5.1.7. NOAA

NOAA is an acronym for National Oceanic and Atmospheric Administration, an organization that is part of the United States Department of Commerce. Several climatic data have been collected by ESRL (NOAA Earth System Research Laboratory. The dataset used in this word, so-called 20th Century Reanalysis, contains objectively-analyzed 4-dimensional weather maps and their uncertainty from the late 19th century to 21st century [50].

5.2. Experimental setup

The Algorithm 2 shows the methodology used in this work to reach the results, which are presented and discussed in the next section. For each dataset, the algorithm is executed three times: one for performing short-term forecast (one step ahead) and two for long-term forecast, by direct prediction (ten and twenty steps ahead).

Table 1
Parameters of the proposed method.

Name	Description	Value
n	Time series lag for dataset generation	5
RunsNumber	Number of method executions	30
TrainPercent	Percentage of the training dataset	70%
ValPercent	Percentage of the validation dataset	20%
TestPercent	Percentage of the test dataset	10%
IterMax	Maximum iterations of PSO	10
SwarmSize	Size of PSO population	20
FitnessFunction	Fitness function of PSO	MSE (Eq. (11))
SimilarityThreshold	Threshold of the dynamic selection algorithm	0.15

The Table 1 shows the description and values of the parameters used. The parameters were defined empirically, after conducting exhaustive initial tests.

Algorithm 2. Dynamic Selection of Forecast Combiners for Chaotic Time Series

```
INPUT: Time series
OUTPUT: Mean and standard deviation of the method prediction errors
for each run $r \in runsNumber$ do
    Build the dataset from the time series with lag $n$ (embedding dimension=5);
    Based in trainPercent, valPercent and testPercent, split the dataset in training ($DB_{tr}$), validation ($DB_{val}$) and test ($DB_{te}$);
    Do 4-fold cross-validation in $DB_{tr}$, generating different training datasets for each individual predictor;
    According to iterMax, swarmSize and fitnessFunction, run standard PSO[51] to obtain the best individual predictors from each model: FANN-1, FANN-2, DBN and SVR;
    Using $DB_{val}$, calculate the weights for softmax combination (Eqs. (2) and (3));
    Using $DB_{te}$, calculate the forecasts of the individual predictors and combiners;
    Using $DB_{te}$ and the Algorithm 1, calculate the prediction of the proposed method from the dynamic selection of combiners;
    Calculate the forecast errors of the individual models, the combiners and the dynamic selection;
end for
```

The MSE (Mean Square Error, Eq. (11)) was used to evaluate the method. The MSE was selected for discussion and used as a performance measure because it is sensitive to the scale of the time series, incorporating both the variance predictor as well as a possible bias. In the Eq. (11), P is the total number of patterns in the set, T_{ij} and L_{ij} are respectively the actual values and the values calculated by the model and $var(t)$ is the variance of the values in the set of desired outputs.

$$\text{MSE} = \frac{\sum_{i=1}^{P}(T_i - L_i)^2}{P} \tag{11}$$

Others forecast error measures were calculated to compare with studies in literature. They are: NMSE (Normalized Mean Square Error, Eq. (12)), NRMSE (Normalized Root Mean Square

Table 2
PSO coding schema.

Model	Parameter	Set of Values
FANN1	Number of units in hidden layer	[5 25]
	Training epochs	[100 5000]
	Initial mu (Levenberg-Marquardt algorithm)	[0.0001 0.1]
FANN2	Number of units in first hidden layer	[5 25]
	Number of units in second hidden layer	[5 25]
	Training epochs	[100 5000]
	Initial mu (Levenberg-Marquardt algorithm)	[0.0001 0.1]
DBN	Number of units in first hidden layer	[5 25]
	Number of units in second hidden layer	[5 25]
	Training epochs	[100 5000]
	Initial mu (Levenberg-Marquardt algorithm)	[0.0001 0.1]
	Pretraining epochs	[50 500]
SVR	SVR type [52]	Epsilon-SVR, nu-SVR
	Kernel type [52]	Linear, polynomial, radial basis, sigmoid
	Cost [52]	[0.1 100]
	Nu [52]	[0.1 1]
	Epsilon [52]	[0.1 1]
	Shrinking [52]	[0 1]

Error, Eq. (13)) and RMSE (Root Mean Square Error, Eq. (14)).

$$NMSE = \frac{\sum_{i=1}^{P} \frac{(T_i - L_i)^2}{var(t)}}{P} \tag{12}$$

$$NRMSE = \frac{\sum_{i=1}^{P} sqrt\left(\frac{(T_i - L_i)^2}{var(t)}\right)}{P} \tag{13}$$

$$RMSE = sqrt\left(\frac{\sum_{i=1}^{P} (T_i - L_i)^2}{P}\right) \tag{14}$$

As PSO was used for algorithm optimization, it was necessary to define the solution code scheme for each of the individual predictors. Table 2 shows the parameters and set of values used to build the candidate solutions.

6. Results and discussion

Tables 3 shows the average performance and standard deviation of the individual predictors, combiners and the proposed method of dynamic selection for each dataset, in the one step ahead forecast, after 30 executions. Highlighted is the best performance for each dataset.

Statistical techniques for comparison of set measurements should be used to determine whether there are significant differences between the results with different methods. The Wilcoxon signed rank test is a statistical nonparametric hypothesis test used to compare two paired samples from the same population, each pair being independent, randomly selected. The efficacy of the Wilcoxon test compared with other tests in the machine learning models is discussed in [53].

Table 4 shows the comparison of the proposed method (using MSE) with the best individual predictor and the best combiner, for one step ahead forecast. The Wilcoxon function tests the null hypothesis in which the data come from a distribution whose median is zero with 5% confidence level, returning $p - value$ probability. When $p - value$ is low enough, then one can assume that the null hypothesis is false (the difference between the distributions is significant). In Tables e 10, the = sign indicates that the null hypothesis was not rejected (the difference between the error averages is not statistically significant) and the models present the same performance. The > sign indicates that the null hypothesis was rejected and the proposed method has superior performance compared with the method used for comparison, while the sign < indicates otherwise.

In the one step ahead forecast, one can see that among the individual predictors, the DBN model performed better in six of the eight datasets: Mackey-Glass, Lorenz, Rossler, Henon, EEG2 and NOAA. In the Laser and EEG1 datasets, the best individual performance was obtained by the feedforward neural network with two hidden layers, without any kind of pre-training. Both models can be considered Deep Learning, turning this neural network category into an important solution for chaotic time series forecasting problem. The difference in performance of DBN in relation to the SVR, for instance, reaches distinct orders of magnitude. This can be seen, for example, in the Lorenz dataset, where the MSE varies from e-05 to e-9.

Still in respect to the individual predictors, for one step ahead forecast, Table 5 shows the average and standard deviation of the parameters calculated by PSO to generate the used neural networks. One can see that the model topologies were relatively close.

Table 3
MSE for one step ahead - 30 runs.

Dataset	Individual models				Combination methods			Proposed
	FANN-1	FANN-2	DBN	SVR	Average	Median	Softmax	DS-FC
Mackey-Glass	1.54e-06 (2.84e-07)	1.25e-06 (3.33e-07)	1.19e-06 (2.32e-07)	2.26e-05 (1.08e-06)	2.31e-06 (1.24e-07)	1.02e-06 (1.13e-07)	1.13e-06 (2.69e-07)	**9.23e-07** (1.60e-07)
Lorenz	6.35e-09 (8.39e-09)	2.67e-09 (2.94e-09)	1.81e-09 (1.58e-09)	1.27e-05 (4.37e-06)	7.84e-07 (2.73e-07)	1.23e-09 (9.71e-10)	1.14e-09 (1.19e-09)	**7.74e-10** **(6.61e-10)**
Rossler	4.26e-08 (3.90e-08)	4.45e-08 (4.94e-08)	1.23e-08 (9.57e-09)	1.51e-04 (6.82e-05)	9.64e-06 (4.34e-06)	2.37e-08 (1.69e-08)	9.26e-09 (8.36e-09)	**8.35e-09** **(6.82e-09)**
Henon	2.39e-10 (2.29e-10)	1.81e-10 (3.20e-10)	4.45e-11 (1.53e-11)	2.70e-05 (3.10e-06)	1.69e-06 (1.94e-07)	9.47e-11 (7.47e-11)	4.37e-11 (1.90e-11)	**3.72e-11** **(1.14e-11)**
Laser	6.85 (1.81)	4.86 (1.83)	5.26 (2.42)	23.2 (52.5)	4.83 (6.97e-01)	3.98 (7.50e-01)	4.13 (1.11)	**3.72** **(8.20e-01)**
EEG1	7.09e-02 (1.02e-02)	6.07e-02 (1.11e-02)	6.16e-02 (8.37e-03)	8.88e-02 (2.68e-03)	5.80e-02 (4.84e-03)	5.72e-02 (4.70e-03)	5.62e-02 (5.55e-03)	**5.52e-02** **(5.14e-03)**
EEG2	9.51e-03 (7.33e-04)	8.94e-03 (9.13e-04)	8.86e-03 (8.35e-04)	1.29e-02 (4.93e-04)	8.53e-03 (3.72e-04)	8.25e-03 (4.31e-04)	8.33e-03 (4.01e-04)	**8.17e-03** **(4.33e-04)**
NOAA	55.74 (2.59)	55.01 (2.33)	53.14 (2.64)	58.25 (9.12e – 01)	52.77 (1.07)	53.13 (1.06)	**52.72** **(1.08)**	52.93 (1.07)

Table 4
Wilcoxon test in respect to MSE, one step ahead.

Dataset	Best Individual	Proposed x Best Individual (*p*-value)	Best Combiner	Proposed x Best Combiner (*p*-value)
Mackey-Glass	DBN	1.3601e-05 >	Median	1.2506e-04 >
Lorenz	DBN	4.3544e-04 >	Softmax	0.0191 >
Rossler	DBN	0.0072 >	Softmax	0.1658 =
Henon	DBN	8.9187e-05 >	Softmax	3.8811e-04 >
Laser	FANN-2	9.6266e-04 >	Median	0.0015 >
EEG1	FANN-2	4.1955e-04 >	Softmax	0.0098 >
EGG2	DBN	8.1878e-05 >	Median	0.0196 >
NOAA	DBN	0.5857 =	Softmax	0.0111 <

Table 5
Mean and std of generated models by PSO.

Model	Parameter	Values mean and std
FANN1	Number of units in hidden layer	18.9 (2.7)
	Training epochs	2830.2 (1190.0)
FANN2	Number of units in first hidden layer	13.4 (4.3)
	Number of units in second hidden layer	14.4 (4.5)
	Training epochs	1928.6 (1240.3)
DBN	Number of units in first hidden layer	15.6 (4.8)
	Number of units in second hidden layer	16.0 (4.5)
	Training epochs	2115.4 (1507.1)
	Pretraining epochs	272.8 (141.4)

The FANN-1 model required on average more units in the hidden layer, and this amount seems to be compensated in the FANN-2 and DBN models by using a second hidden layer. On average, the simplest model of neural network also needed more training epochs. Although the DBN model is more expensive, because it has two hidden layers and needs an average of 272.8 epochs of pretraining, its performance in the final prediction seems to outweigh this increased processing cost.

As discussed in several works in literature, combining forecasts tends to generate responses with better performance. In the experiments, both the median and the softmax had generally superior performance than individual predictors in the one step ahead forecast. Softmax was the best combiner in five datasets (Lorenz, Rossler, Henon, EEG1 and NOAA) and the median was the best combiner in the remaining series (Mackey-Glass, Laser and EEG1). Though the average is a widely used combination in literature, and generally performs better than individual models, this did not occur in the experiments. One explanation for this behavior is the average nature, that is a statistical measure very susceptible to outliers. That was precisely what happened in the experiments: in most of the tested datasets the performance of the SVR model considerably disagreed when compared with neural networks, pulling down the average performance. According to the results, the median managed to overcome this scenario, making itself useful in situations with this characteristic, when at least one of the individual models has a lower performance compared with the others. The results of the combiners support the need for a dynamic selection method, since it was not possible to predict which combiner would have the best performance. Rely on average, due to its extensive use, would be a mistake in the forecast of these time series. Although softmax had the best combiner performance, this is the only one that needs a validation dataset. Generating the weights of the weighted sum, as softmax does, stands for a small additional computational cost in the method.

For one step ahead forecast, the dynamic selection of combiners proposed in this paper showed satisfactory results in all tested datasets. When compared with individual predictors and performed combinations, the performance of the method was higher than the one in all calculated error measure, except in NOAA dataset. With the Wilcoxon test, it was found that the differences in average MSE were significant for all time series both for individual predictors as for the combiners, except for Rossler and NOAA dataset compared with softmax. The results confirm the initial hypothesis that, from the moment it is not known which combiner produces the best predictions, it is necessary to have a method to dynamically select the best combination from each test pattern. For example, the best combination for Mackey-Glass base was the median, while for Henon the softmax was the best combiner. In both cases, the dynamic selection was statistically superior to these combinations.

Figs. 2, 3, 4, 5, 6, 7, 8 and 9 show forecasts and absolute errors of an execution round of the proposed method for each time series, in one step ahead forecast. In forecast, the blue line with dots is the desired output and the red line the produced output. In

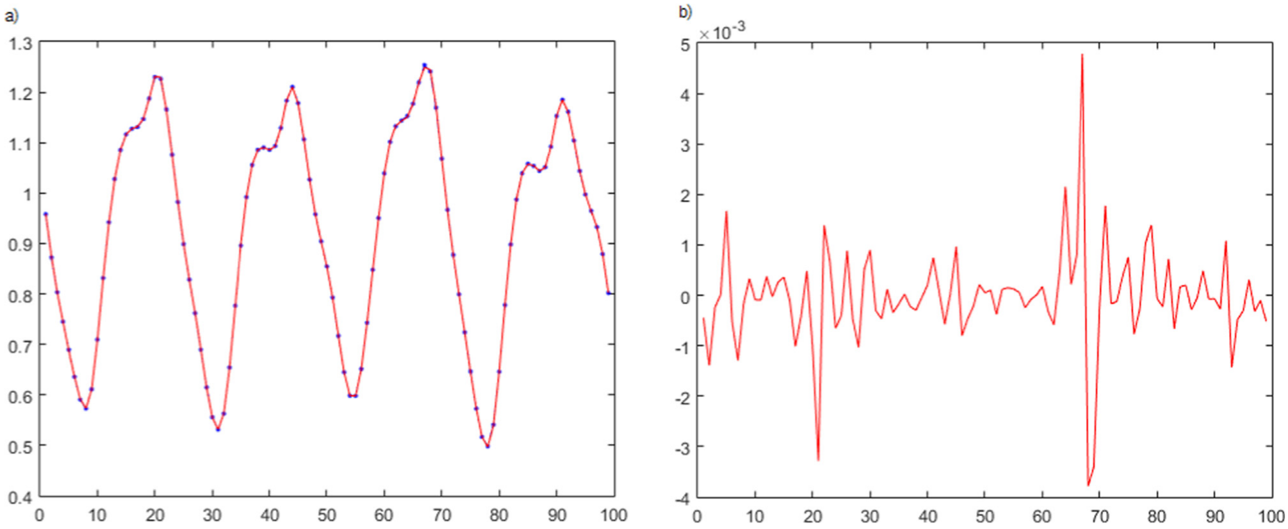


Fig. 2. Forecast (a) and error prediction (b) of proposed method for Mackey-Glass.

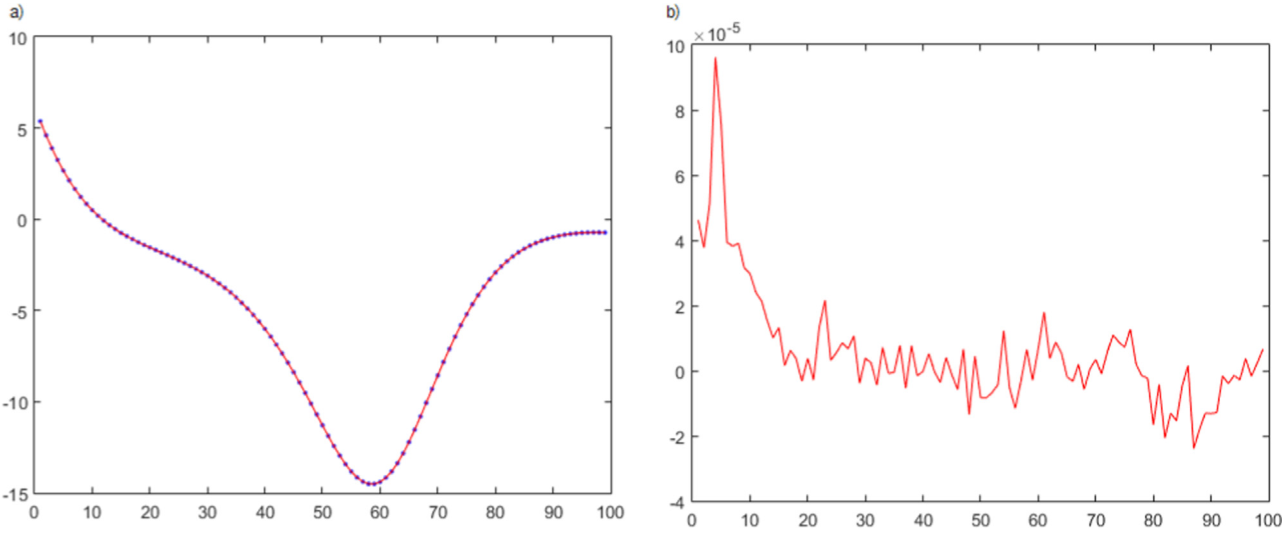


Fig. 3. Forecast (a) and error prediction (b) of proposed method for Lorenz.

each figure, the vertical axis of the first graph is given by the desired values and produced predictions. In the second graph, the vertical axis marks the absolute difference between the desired and produced output. In both cases, the horizontal axis is given by the points of the test set. In all datasets, the predictions were very close to the desired output. Absolute errors present uniform behavior, with exceptions in some curve points. Interestingly, the less close predictions of the ideal curve occurred in datasets with noise. However, that would be the expected behavior, since the prediction of natural time series is inherently a more difficult problem. In NOAA dataset, the combiners were not able to substantially improve the performance of individual predictors, influencing the result of the dynamic selection.

The dynamic selection method had satisfactory results when compared to individual models and combiners of the outputs of these models. However, it is also necessary to compare the performance of the proposed method with models with lower computational cost, such as statistical methods. Table 6 shows the comparison between the average DS-FC's performance and the performance of one execution round of AR, ARMA, ARIMA models ($n=5$ is the number of autoregressive terms and moving averages, where applicable) and the naive predictor. The best performance is highlighted in tables. It can be seen that the performance of the dynamic selection exceeds the tested statistical models, although these are still used successfully on some problems. Interestingly, the difference in performance is lower in natural datasets, with noise. In such cases, it is possible to use one or more statistical models as individual predictors in dynamic selection.

In real problems, provide only the immediately posterior value of a time series may be insufficient to ensure the relevance of the model. In this case, the predictors should be constructed to enable long-term forecasting. For long-term forecast, two strategies are commonly used. The multi-stage prediction consists in iteratively using the short-term model predictions, up to the desired horizon. In this case, it is shown empirically that the errors can accumulate and be propagated in future predictions. Another approach is called direct prediction, where the dataset is constructed to include the desired output prediction horizon. Tables 7,8 show, according to the same disposition of the above results, the performance of the proposed method for long-term prediction of ten and twenty steps. The Tables 9,10 show the results of the Wilcoxon test for these two sets of experiments.

In the ten steps ahead forecast, once again the DBN stood out as the best single predictor, getting better results in five of the eight datasets. Regarding combiners, softmax and median were better in most datasets. The average was better in Henon dataset. The

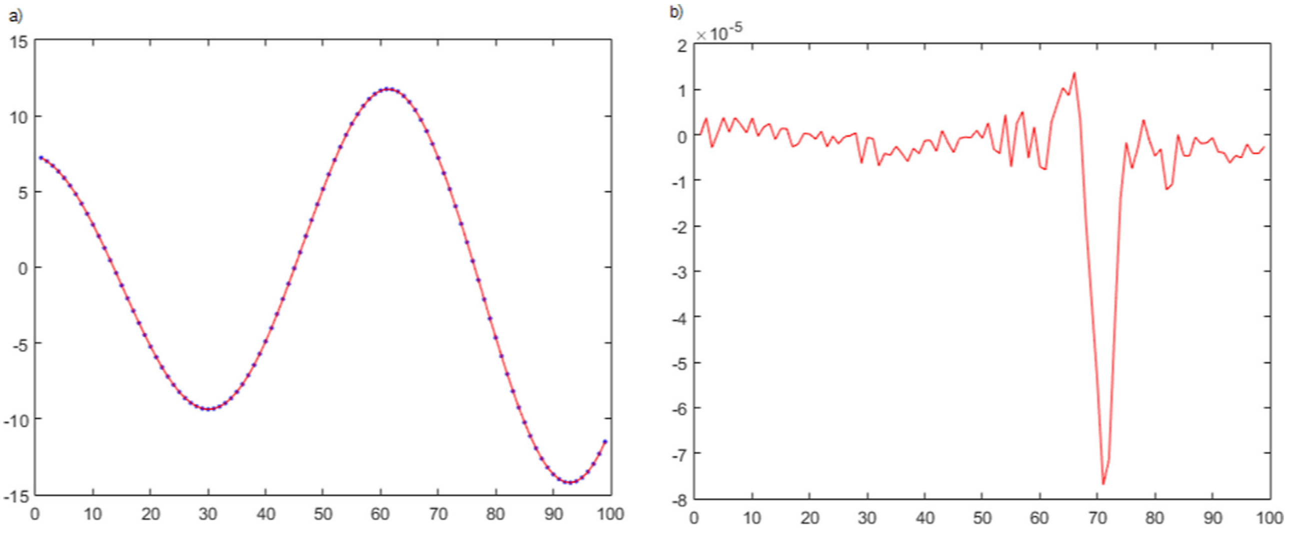


Fig. 4. Forecast (a) and error prediction (b) of proposed method for Rossler.

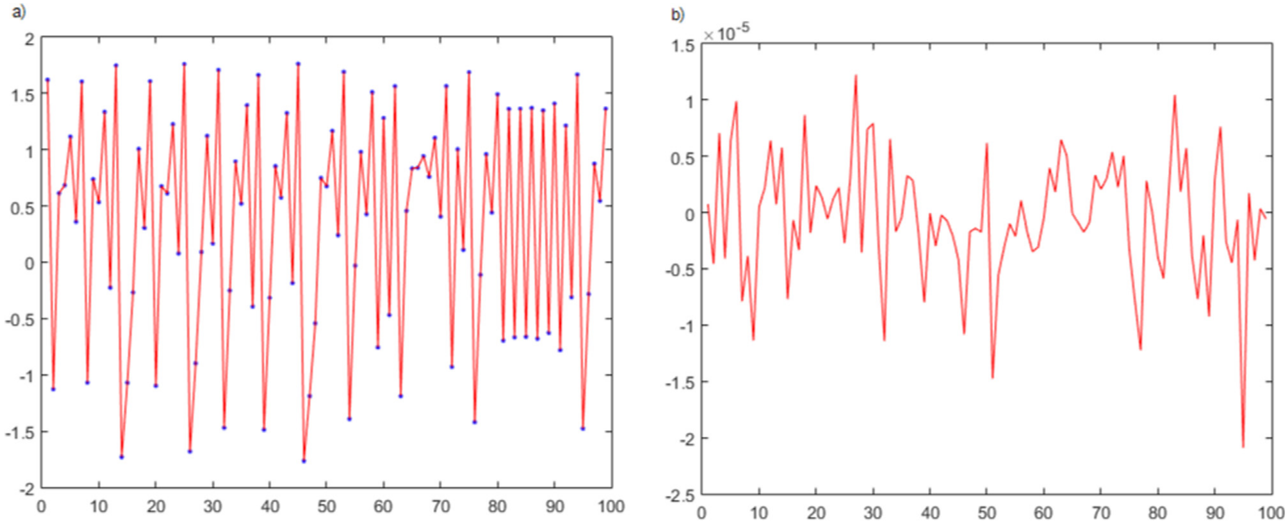


Fig. 5. Forecast (a) and error prediction (b) of proposed method for Henon.

proposed dynamic selection method achieved superior performance in most of the tested datasets (Mackey-Glass, Lorenz, Laser and EEG2). In EEG1 and Rossler, the performance of the dynamic selection was statistically similar to the best combiner, despite having reached the absolute best results. In NOAA dataset, again, the DS-FC had underperformed the best combiner, despite having overcome the single best predictor (SVR, in its only appearance among the best predictors). As in the case of short-term forecasting, dynamic selection in ten steps prediction is justified because, except for one dataset, its performance was at least equal to the best combiner.

Regarding the twenty steps ahead forecast, the performance of the proposed method was more affected. In three of the tested datasets, the DS-FC did not statistically outperform the best individual predictor, although always got an absolute best result. When compared to the best combiners, the dynamic selection was statistically better in two datasets. In the remaining time series, except for Henon, where the DS-FC was statistically worse than the median, dynamic selection achieved similar performance. However, as it happened in the previous experiments, the DS-FC ensured, in most scenarios, at least the performance of the best individual predictor and also the best combiner.

Normally, the longer the forecast horizon, more difficult the problem is. In most tested scenarios, the performance of all predictors, combiners and dynamic selection decreased when predictions with greater horizon were made. However, the decay curve showed variability in the datasets, and three behaviors could be identified. The first behavior can be observed in Fig. 10, which shows the average of MSE for predictions of one, ten and twenty steps in the EEG2 dataset. In this case, with a similar behavior in EEG1 and Henon time series, the performance difference between the one and ten step forecasts was greater than the difference between the ten and twenty steps forecasts. Opposite behavior occurred in Mackey-Glass dataset (Fig. 11), Lorenz and Laser. In these time series the performance decreased more abruptly according the forecast horizon. Two unusual situations occurred. The first one can be seen in Fig. 12, which shows the performance variation of the NOAA dataset. In NOAA, the short-term forecast was worse than both long-term predictions. This behavior may have influenced the poor performance of the proposed method in this time series. Another odd situation occurred in Henon dataset, where the long-term forecasts were much worse than the short-term predictions in several orders of magnitude. This behavior may be explained by the presence of more seasonal cycles in this series, as shown in Fig. 5.

The results achieved by the proposed method were compared

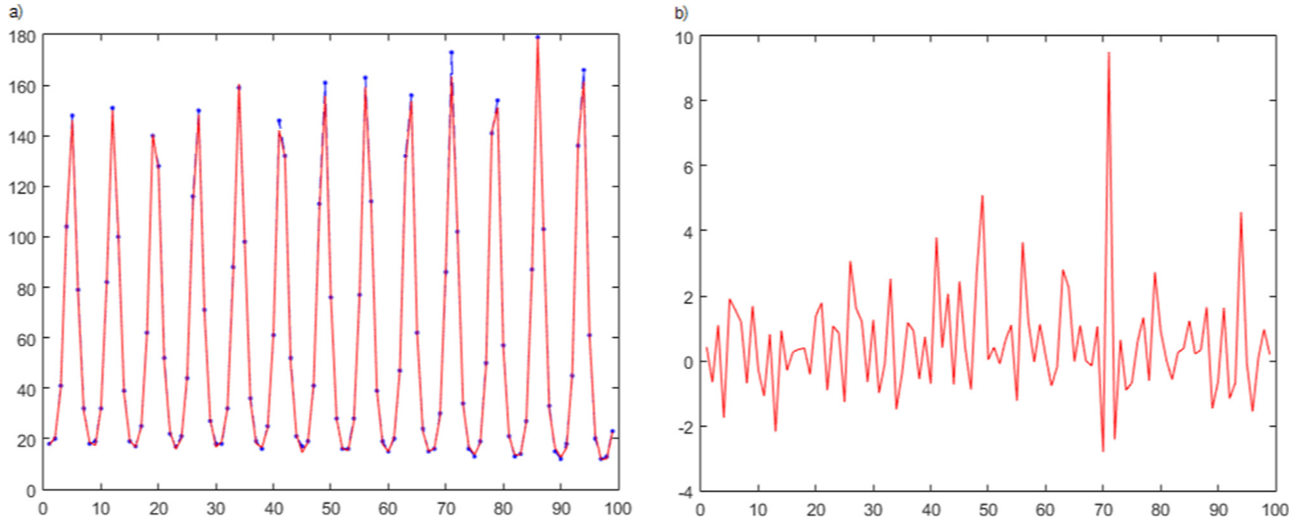


Fig. 6. Forecast (a) and error prediction (b) of proposed method for Laser.

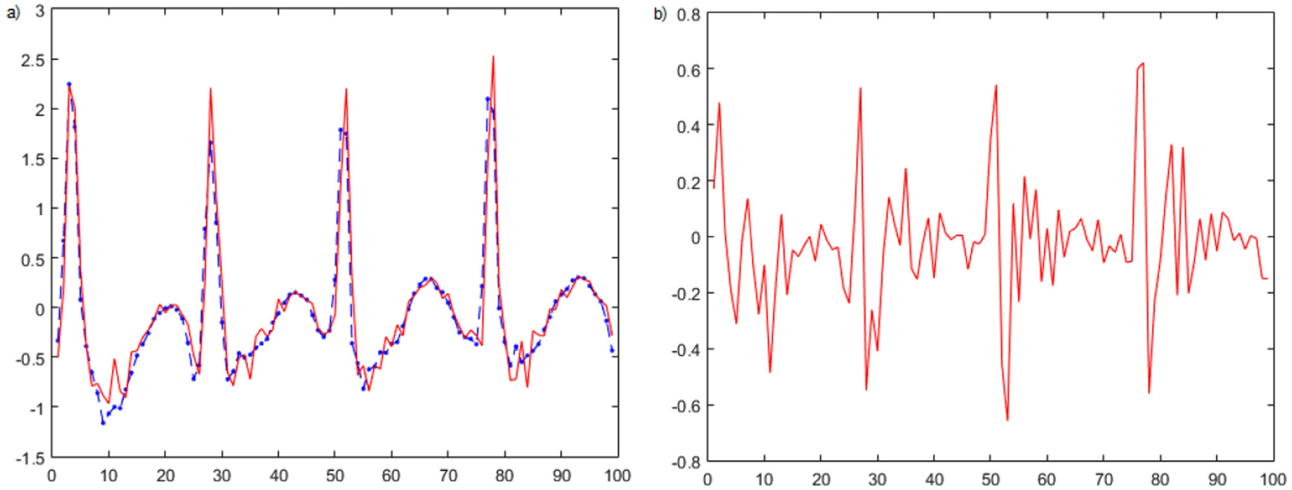


Fig. 7. Forecast (a) and error prediction (b) of proposed method for EEG1.

with some published studies that had used the tested datasets. The works related in this analysis are of different natures, predicting chaotic series with techniques like swarm intelligence [54], fuzzy logic [55] and hybrid models [3]. Table 11 shows a comparison of the average of three forecasting error measures achieved by the proposed method (NMSE, NRMSE and RMSE) with published results, in the one step forecast. In absolute values, the dynamic selection achieved better performance than all the works shown in four of the tested time series (Mackey-Glass, Lorenz, Rossler and Henon). In Rossler and Henon series, the performance of the proposed method in relation to NMSE had a very significant gain. In Laser series, the proposed method had a lower performance compared to two techniques: a recursive Bayesian neural network and a SVR model with fuzzy logic. Nevertheless, the results obtained in this work proved to be competitive. It is important to remember that this comparison should be made with reservations, since the other experiments have not been reproduced, and most of them are the results of a single execution round.

The literature also contains works that investigate the long-term forecasting of chaotic time series. One category of techniques widely used in this task are the so-called neurofuzzy models. An example can be seen in the work of Gholipour et al. [59]. In this paper, the authors provide long-term prediction of three time series, including Lorenz in ten steps ahead forecasts. The neurofuzzy prediction achieved NMSE of 3.9e-03, while the DS-FC obtained 1.5140e-08 as an average of the same error measure. Another example of using neurofuzzy approaches to time series forecasting can be seen in [60], where the authors propose a model applied to space weather prediction.

Since the proposed method is not limited to the use of individual predictors and combinations tested, any of these neurofuzzy models can be used. The dynamic selection presented in this study acts more like a framework. It is possible to modify both predictors as combiners. Thus the dynamic selection can be made with a combination of several techniques that have already been tested or that may be used in the future and tend to show better results than individual models, as discussed in this analysis.

7. Conclusions and future works

Based on classification and pattern recognition problems, this paper proposed a method of dynamic selection of forecast combiners for chaotic time series. Initially, individual predictors with a good degree of diversity produce their respective outputs. Diversity is achieved by the use of heterogeneous models and cross-validation. Second, combinations are performed on individual

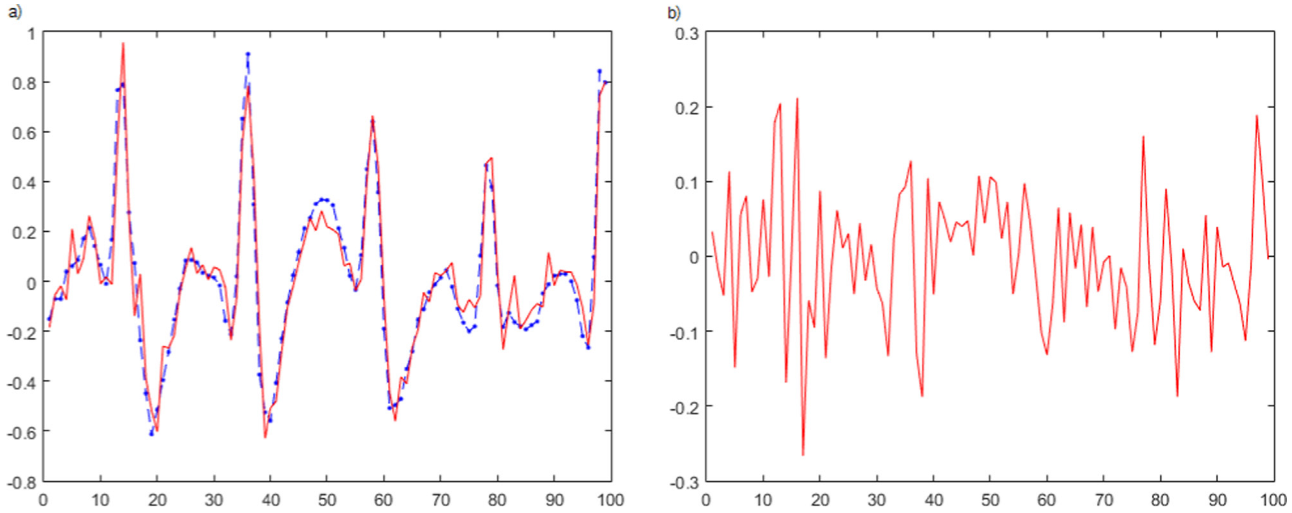


Fig. 8. Forecast (a) and error prediction (b) of proposed method for EEG2.

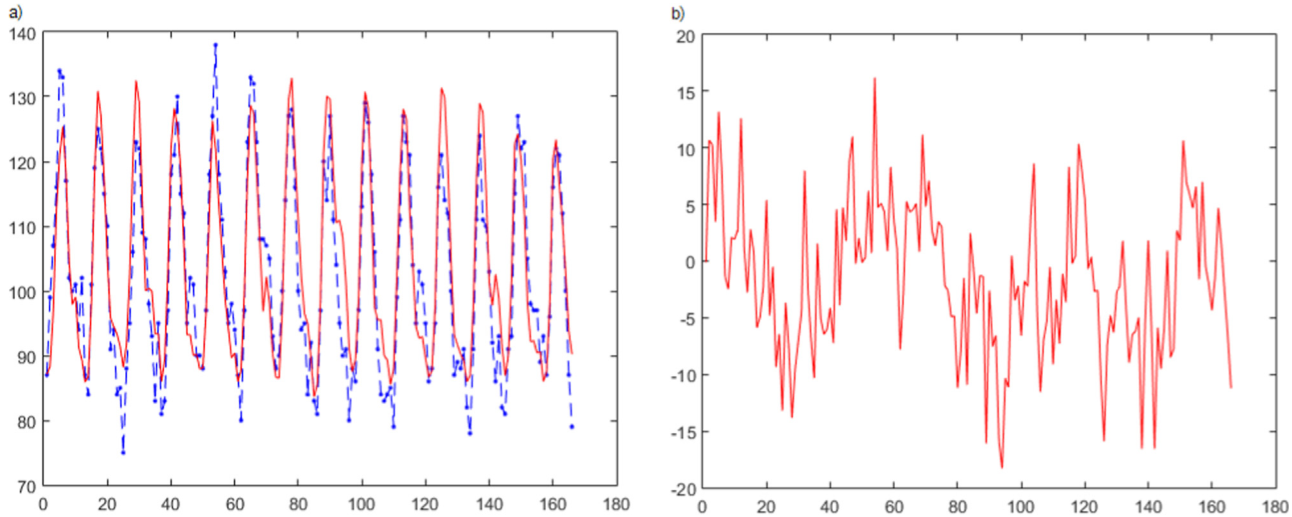


Fig. 9. Forecast (a) and error prediction (b) of proposed method for NOAA.

Table 6
Comparison with statical models - MSE.

Dataset	DS-FC	AR	ARMA	ARIMA	Naive
Mackey-Glass	**9.23e-07**	2.78e-02	9.60e-02	5.09e-02	7.86e-02
Lorenz	**7.74e-10**	7.09e-04	7.07e-04	8.99e-04	4.59
Rossler	**8.35e-09**	1.72e-01	2.77e-03	6.00e-01	13.78
Henon	**3.72e-11**	1.09	1.97	1.09	2.46
Laser	**3.72**	2443.53	734.77	568.22	7506.55
EEG1	**5.52e-02**	7.55e-01	4.39	3.60	1.03
EEG2	**8.17e-03**	1.76e-01	3.53	9.13e-01	2.29e-01
NOAA	**52.93**	484.75	969.607	764.72	7.86e-02

predictions. Finally, a dynamic selection algorithm is used to choose the most promising combination for each test pattern. Neural networks and support vector machine were used as individual predictors, while the selected combiners were the average, median and softmax. Simple statistical measures were used as combiners due to their easy implementation and relative robustness compared with more sophisticated methods. The dynamic selection was performed with the algorithm called DS-FC (Dynamic Selection - Forecast Combiners). It is important to state that the method acts as a framework, working similarly with other predictors and combiners.

To test the proposed method, eight time series with chaotic behavior were used: Mackey-Glass, Lorenz, Rossler, Henon, Laser, EEG1, EEG2 and NOAA. The prediction of chaotic series is important for many areas of human activity such as astronomy and signal processing, and those that were tested also are used as benchmark in several works. In the experiments, the DBN model stood out as best individual predictor in four datasets. As in the other dataset the best model was a neural network with two hidden layers, Deep Learning seems to be a good technique for chaotic time series forecasting. Median and softmax were the best combiners. Apparently, the average could not overlap outliers, since the SVR model presented discrepant results compared with other individual predictors.

The proposed dynamic selection achieved satisfactory results in all datasets. For one step ahead forecasts, after conducting statistical tests, it was proven that the method was superior to the best combiners in six of the eight time series. Compared with many works of literature, the dynamic selection presented in this study had better results than most of them and, when it had no better performance, it showed competitive error measures.

In the case of long-term forecasts for ten and twenty steps, the performance of the proposed method is less significant than in the case of the simpler prediction. But yet, in most scenarios tested, the DS-FC reached at least the best performance of the individual

Table 7
MSE for ten steps ahead - 30 runs.

Dataset	Individual models				Combination methods			Proposed
	FANN-1	FANN-2	DBN	SVR	Average	Median	Softmax	DS-FC
Mackey-Glass	1.83e-06	1.39e-06	1.09e-06	2.13e-04	1.46e-05	1.22e-06	1.07e-06	**9.46e-07**
	(3.13e-07)	(3.30e-07)	(2.12e-07)	(1.24e-05)	(8.30e-07)	(1.50e-07)	(1.89e-07)	**(1.74e-07)**
Lorenz	1.13e-06	5.89e-07	2.86e-07	2.35e-01	1.46e-02	2.85e-07	6.84e-07	**2.72e-07**
	(8.64e-07)	(2.71e-07)	(1.33e-07)	(6.54e-02)	(4.07e-03)	(1.02e-07)	(8.87e-07)	**1.14e-07)**
Rossler	8.42e-04	9.44e-05	1.65e-04	1.09e+00	6.84e-02	1.47e-04	8.16e-05	**7.56e-05**
	(9.87e-04)	(9.36e-05)	(1.54e-04)	(3.90e-01)	(2.43e-02)	(1.56e-04)	(8.91e-05)	**(8.17e-05)**
Henon	1.04	1.05	1.05	1.05	**1.02**	1.02	1.02	1.03
	(2.55e-02)	(5.23e-02)	(2.57e-02)	(2.26e-02)	**(1.42e-02)**	(1.55e-02)	(1.42e-02)	(1.43e-02)
Laser	47.58	34.59	27.74	257.45	28.93	20.13	18.26	**13.38**
	(18.72)	(28.62)	(10.91)	(100.98)	(12.24)	(8.81)	(8.82)	**(5.14)**
EEG1	3.87e-01	3.85e-01	3.89e-01	4.39e-01	3.79e-01	3.80e-01	3.79e-01	**3.79e-01**
	(1.47e-02)	(1.28e-02)	(1.79e-02)	(1.09e-02)	(7.11e-03)	(8.14e-03)	(7.05e-03)	**(6.73e-03)**
EEG2	6.51e-02	6.33e-02	6.64e-02	8.34e-02	6.51e-02	6.44e-02	6.46e-02	**6.40e-02**
	(3.70e-03)	(3.84e-03)	(5.84e-03)	(3.13e-03)	(2.38e-03)	(2.70e-03)	(2.46e-03)	**(2.54e-03)**
NOAA	46.03	46.63	45.91	44.27	43.36	43.94	**43.23**	43.61
	(2.03e)	(2.15e)	(2.28)	(7.31e-01)	(7.39e-01)	(8.59e-01)	**(7.35e-01)**	(7.90e-01)

Table 8
MSE for twenty steps ahead - 30 runs.

Dataset	Individual models				Combination methods			Proposed
	FANN-1	FANN-2	DBN	SVR	Average	Median	Softmax	DS-FC
Mackey-Glass	1.83e-05	1.05e-05	9.70e-06	2.89e-03	1.88e-04	9.70e-06	9.21e-06	**7.65e-06**
	(4.93e-06)	(2.25e-06)	(1.76e-06)	(1.81e-04)	(1.20e-05)	(1.17e-06)	(1.66e-06)	**(1.18e-06)**
Lorenz	2.84e-05	9.91e-06	8.13e-06	8.52	5.32e-01	9.45e-06	1.25e-05	**8.12e-06**
	(1.93e-05)	(6.43e-06)	(5.19e-06)	(1.87)	(1.17e-01)	(3.69e-06)	(1.26e-05)	**(4.52e-06)**
Rossler	3.22e-03	2.06e-04	1.73e-04	3.83e+00	2.38e-01	4.09e-04	1.14e-04	**1.08e-04**
	(2.02e-03)	(1.62e-04)	(3.93e-04)	(1.94e+00)	(1.21e-01)	(2.65e-04)	(9.20e-05)	**(8.07e-05)**
Henon	1.01	1.03	9.95e-01	9.68e-01	9.85e-01	**9.83e-01**	9.85e-01	9.86e-01
	(4.14e-02)	(4.36e-02)	(2.86e-02)	(2.95e-02)	(1.63e-02)	**(1.59e-02)**	(1.62e-02)	(1.49e-02)
Laser	138.35	65.41	84.72	505.38	78.63	67.49	47.13	**40.32**
	(75.68)	(32.85)	(30.82)	(143.21)	(25.82)	(25.97)	(18.67)	**(14.38)**
EEG1	4.03e-01	4.14e-01	4.06e-01	4.05e-01	3.93e-01	3.94e-01	3.93e-01	**3.93e-01**
	(1.78e-02)	(2.01e-02)	(1.75e-02)	(1.44e-02)	(7.86e-03)	(8.70e-03)	(7.73e-03)	**(7.98e-03)**
EEG2	6.30e-02	6.28e-02	6.64e-02	6.45e-02	6.15e-02	6.18e-02	6.14e-02	**6.13e-02**
	(4.53e-03)	(5.03e-03)	(5.72e-03)	(5.36e-03)	(2.49e-03)	(2.23e-03)	(2.46e-03)	**(2.34e-03)**
EEG2	49.42	49.63	48.13	49.17	46.67	46.83	46.66	**46.61**
	(2.03)	(2.09)	(2.14e)	(6.87e-01)	(6.80e-01)	(8.90e-01)	(6.63e-01)	**(6.52e-01)**

Table 9
Wilcoxon test in respect to MSE, ten steps ahead.

Dataset	Best Individual	Proposed x Best Individual (*p*-value)	Best Combiner	Proposed x Best Combiner (*p*-value)
Mackey-Glass	DBN	1.7988e-05 >	Softmax	3.1817e-06 >
Lorenz	DBN	0.5377 =	Median	0.0496 >
Rossler	FANN-2	0.0185 >	Softmax	0.3820 =
Henon	FANN-1	0.0175 >	Mean	3.8811e-04 <
Laser	DBN	1.9209e-06 >	Softmax	9.3157e-06 >
EEG1	DBN	0.0041 >	Median	0.6143 =
EGG2	DBN	0.0157 >	Median	0.0053 >
NOAA	SVR	0.0044 >	Softmax	5.2165e-06 <

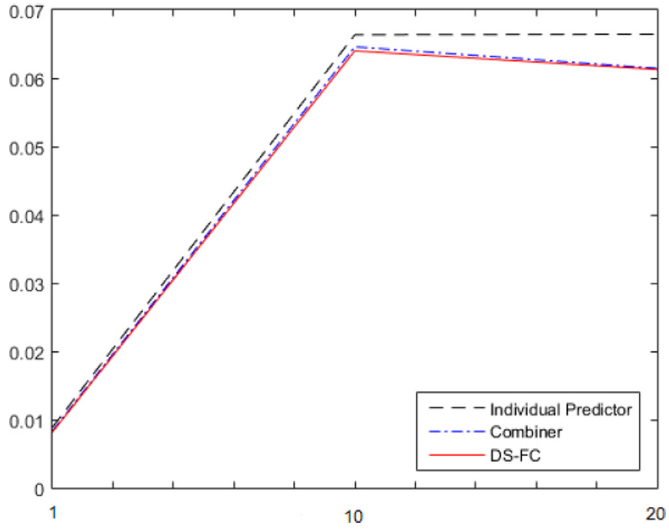


Fig. 10. Multi-step forecasting decay curve - EEG2.

Table 10
Wilcoxon test in respect to MSE, twenty steps ahead.

Dataset	Best Individual	Proposed x Best Individual (*p*-value)	Best Combiner	Proposed x Best Combiner (*p*-value)
Mackey-Glass	DBN	4.2857e-06 >	Softmax	1.7344e-06 >
Lorenz	DBN	0.8936 =	Median	0.0830 =
Rossler	DBN	0.8612 =	Softmax	0.6612 =
Henon	DBN	0.0545 =	Median	0.0021 <
Laser	FANN-2	5.7517e-06 >	Softmax	3.7243e-05 >
EEG1	FANN-1	0.0060 >	Softmax	0.6733 =
EGG2	FANN-2	0.0407 >	Softmax	0.0627 =
NOAA	DBN	3.8811e-04 >	Softmax	0.9099 =

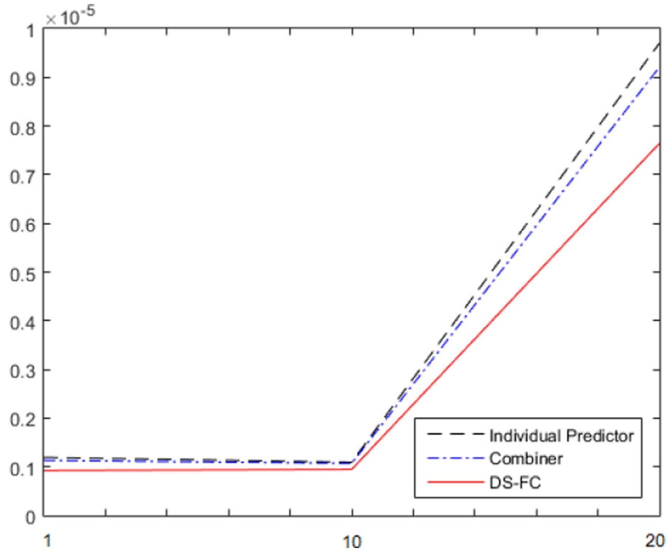


Fig. 11. Multi-step forecasting decay curve - Mackey-Glass.

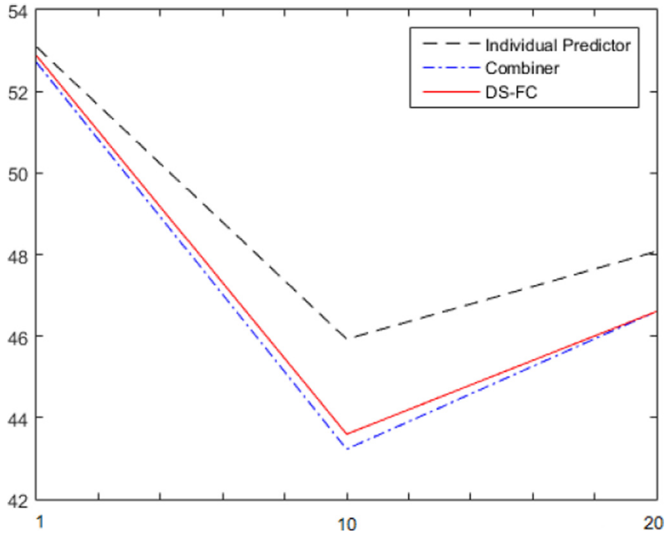


Fig. 12. Multi-step forecasting decay curve - NOAA.

Table 11
Comparison with literature.

Dataset	Method	NMSE	NRMSE	RMSE
Mackey-Glass	Proposed	1.98e-05	2.95e-03	9.57e-04
	Ardalani-Farsa et al. (2011) [56]			1.30e-03
	Chandra et al. (2012) [17]	2.79e-04		6.33e-03
	Li et al. (2012) [1]		1.94e-01	
	Miranian et al. (2013) [55]			7.90e-04
	Yilmaz et al. (2010) [57]			1.09e-03
Lorenz	Proposed	2.97e-11	3.28e-06	2.51e-05
	Ardalani-Farsa et al. (2011) [56]			2.96e-02
	Chandra et al. (2012) [17]			6.36e-03
	Li et al. (2012) [1]			2.23e-01
	Miranian et al. (2013) [55]	6.40e-05		
	Bodyanskiy et al. (2013) [3]			1.89e-01
Rossler	Proposed	1.27e-10	7.10e-06	8.50e-05
	Mirikitani et al. 1 (2010) [58]	1.01e-03		
	Mirikitani et al. 2 (2010) [58]	8.10e-04		
	Miranian et al. (2013) [55,58]	1.50e-05		
Henon	Proposed	3.58e-11	4.41e-06	6.04e-06
	Mirikitani et al. 1 (2010) [58]	7.20e-04		
	Mirikitani et al. 2 (2010) [58]	6.80e-04		
	Miranian et al. (2013) [55]	4.40e-04		
Laser	Proposed	1.47e-03	2.70e-02	1.92
	Mirikitani et al. 1 (2010) [58]	4.36e-03		
	Mirikitani et al. 2 (2010) [58]	6.00e-04		
	Miranian et al. (2013) [55]	5.30e-04		

predictors and combiners.

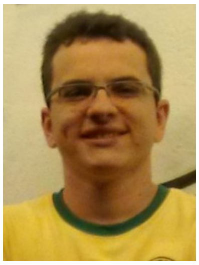
The proposed method selects the combiner with better performance on a subset of the training dataset similar to a given test pattern. The similarity in the implemented algorithm is given by the Euclidean distance between the predictions of combiners to the test pattern and the selected competence area. Selection of the competence area is determined by the k nearest neighbors of the test pattern. The method tends to beat the performance of the combiners because it is unlikely that any of them is superior to others in all areas of competence. By extending the reasoning described by Eqs. (4), (5) and (6), the dynamic selection has a disposition to get at least the performance of the best model in the ensemble. For each test pattern, the algorithm searches the best combiner in the most appropriate competence region, defined by the performance experience in a known dataset. This case is precisely what occurred in the experiments where the proposed method reached at least the performance of the best combiner.

As future work, several paths can be followed: use of other dynamic selection methods rather than nearest neighbor techniques; test the method with other individual predictors and combiners; use unsupervised techniques to support dynamic selection.

References

[1] D. Li, M. Han, J. Wang, Chaotic time series prediction based on a novel robust echo state network, IEEE Trans. Neural Netw. Learn. Syst. 23 (5) (2012) 787–799.
[2] Y. Bao, T. Xiong, Z. Hu, Multi-step-ahead time series prediction using multiple-output support vector regression, Neurocomputing 129 (2014) 482–493.
[3] Y. Bodyanskiy, O. Vynokurova, Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification, Inf. Sci. 220 (2013) 170–179.
[4] J.P. Donate, X. Li, G.G. Sánchez, A.S. De Miguel, Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm, Neural Comput. Appl. 22 (1) (2013) 11–20.
[5] A.T. Sergio, T.B. Ludermir, Reservoir computing optimization with a hybrid method, in: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 2653–2660.
[6] I. Rojas, O. Valenzuela, F. Rojas, A. Guillén, L.J. Herrera, H. Pomares, L. Marquez, M. Pasadas, Soft-computing techniques and arma model for time series prediction, Neurocomputing 71 (4) (2008) 519–537.
[7] P.R.A. Firmino, P.S. De Mattos Neto, T.A. Ferreira, Error modeling approach to improve time series forecasters, Neurocomputing 153 (2015) 242–254.
[8] H. Liu, H.-q. Tian, D.-f. Pan, Y.-f. Li, Forecasting models for wind speed using wavelet, wavelet packet, time series and artificial neural networks, Appl. Energy 107 (2013) 191–208.
[9] U. Yolcu, C.H. Aladag, E. Egrioglu, V.R. Uslu, Time-series forecasting with a novel fuzzy time-series approach: an example for Istanbul stock market, J. Stat. Comput. Simul. 83 (4) (2013) 599–612.
[10] C. Voyant, M.L. Nivet, C. Paoli, M. Muselli, G. Notton, Meteorological time series forecasting based on mlp modelling using heterogeneous transfer functions, in: Journal of Physics: Conference Series, 574, IOP Publishing, 2015, p. 012064.
[11] J. Shaman, W. Yang, S. Kandula, Inference and Forecast of the Current West African Ebola Outbreak in Guinea, Sierra Leone and Liberia, PLoS Currents 6.
[12] G. Reikard, Forecasting space weather: can new econometric methods improve accuracy? Adv. Space Res. 47 (12) (2011) 2073–2080.
[13] F. Moretti, S. Pizzuti, S. Panzieri, M. Annunziato, Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling, Neurocomputing 167 (2015) 3–7.
[14] K.F. Tiampo, R. Shcherbakov, Seismicity-based earthquake forecasting techniques: ten years of progress, Tectonophysics 522 (2012) 89–121.
[15] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[16] M.B. Kennel, S. Isabelle, Method to distinguish possible chaos from colored noise and to determine embedding parameters, Phys. Rev. A 46 (6) (1992) 3111.
[17] R. Chandra, M. Zhang, Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction, Neurocomputing 86 (2012) 116–123.
[18] B. Samanta, Prediction of chaotic time series using computational intelligence, Expert Syst. Appl. 38 (9) (2011) 11406–11411.
[19] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.
[20] J.M. Bates, C.W. Granger, The combination of forecasts, Or (1969) 451–468.
[21] V.R.R. Jose, R.L. Winkler, Simple robust averages of forecasts: some empirical results, Int. J. Forecast. 24 (1) (2008) 163–169.
[22] R. Adhikari, A neural network based linear ensemble framework for time series forecasting, Neurocomputing 157 (2015) 231–242.
[23] R. Adhikari, R. Agrawal, A novel weighted ensemble technique for time series forecasting, in: Advances in Knowledge Discovery and Data Mining, Springer, 2012, pp. 38–49.
[24] I.A. Gheyas, L.S. Smith, A novel neural network ensemble architecture for time series forecasting, Neurocomputing 74 (18) (2011) 3855–3864.
[25] A.S. Britto, R. Sabourin, L.E. Oliveira, Dynamic selection of classifiersa comprehensive review, Pattern Recognit. 47 (11) (2014) 3665–3680.
[26] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[27] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[28] J.A. Hoeting, D. Madigan, A.E. Raftery, C.T. Volinsky, Bayesian model averaging: a tutorial, Stat. Sci. (1999) 382–401.
[29] R.R. Andrawis, A.F. Atiya, H. El-Shishiny, Forecast combinations of computational intelligence and linear models for the nn5 time series forecasting competition, Int. J. Forecast. 27 (3) (2011) 672–688.
[30] C. Lian, Z. Zeng, W. Yao, H. Tang, Ensemble of extreme learning machine for landslide displacement prediction based on time series analysis, Neural Comput. Appl. 24 (1) (2014) 99–107.
[31] R. Adhikari, R. Agrawal, Combining multiple time series models through a robust weighted mechanism, in: 2012 1st International Conference on Recent

Advances in Information Technology (RAIT), IEEE, 2012, pp. 455–460.
[32] J.S. Armstrong, Combining forecasts, in: Principles of forecasting, Springer, 2001, pp. 417–439.
[33] D.W. Bunn, A bayesian approach to the linear combination of forecasts, Oper. Res. Q. (1975) 325–329.
[34] P.S. Freitas, A.J. Rodrigues, Model combination in neural-based forecasting, Eur. J. Oper. Res. 173 (3) (2006) 801–814.
[35] M. Oliveira, L. Torgo, Ensembles for time series forecasting, in: Proceedings of the Sixth Asian Conference on Machine Learning, 2014, pp. 360–370.
[36] G.P. Zhang, A neural network ensemble method with jittered training data for time series forecasting, Inf. Sci. 177 (23) (2007) 5329–5346.
[37] S.F. Crone, N. Kourentzes, Feature selection for time series prediction-a combined filter and wrapper approach for neural networks, Neurocomputing 73 (10) (2010) 1923–1936.
[38] M. Mirmomeni, W. Punch, Co-evolving data driven models and test data sets with the application to forecast chaotic time series, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 14–20.
[39] J. Kittler, M. Hatef, R.P. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[40] M. Qi, G.P. Zhang, An investigation of model selection criteria for neural network time series forecasting, Eur. J. Oper. Res. 132 (3) (2001) 666–680.
[41] S. Makridakis, R.L. Winkler, Averages of forecasts: Some empirical results, Manag. Sci. 29 (9) (1983) 987–996.
[42] A.J. Sharkey, N.E. Sharkey, Combining diverse neural nets, Knowl. Eng. Rev. 12 (03) (1997) 231–247.
[43] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[44] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using a deep belief network with restricted boltzmann machines, Neurocomputing 137 (2014) 47–56.
[45] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: International Conference on Image Analysis and Processing, 1999, IEEE, 1999, pp. 659–664.
[46] O.E. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.
[47] G.W. Flake, The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation, MIT Press, 1998.
[48] Laser generated time series, 〈http://www-psych.stanford.edu/andreas/Time-Series/SantaFe.html〉, accessed: 2015-12-23.
[49] Eeg time series database, 〈https://vis.caltech.edu/rodri/data.htm〉, accessed: 2016-05-23.
[50] Noaa, 〈http://www.esrl.noaa.gov/psd/data/timeseries/〉, accessed: 2016-06-01.
[51] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, 2010, pp. 760–766.
[52] Lib svm, 〈https://www.csie.ntu.edu.tw/cjlin/libsvm/〉, accessed: 2015-12-23.
[53] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[54] C.-J. Lin, C.-H. Chen, C.-T. Lin, A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications, IEEE Trans. Syst., Man Cybern. C: Appl. Rev. 39 (1) (2009) 55–68.
[55] A. Miranian, M. Abdollahzade, Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction, IEEE Trans. Neural Netw. Learn. Syst. 24 (2) (2013) 207–218.
[56] M. Ardalani-Farsa, S. Zolfaghari, Residual analysis and combination of embedding theorem and artificial intelligence in chaotic time series forecasting, Appl. Artif. Intell. 25 (1) (2011) 45–73.
[57] S. Yilmaz, Y. Oysal, Fuzzy wavelet neural network models for prediction and identification of dynamical systems, IEEE Trans. Neural Netw. 21 (10) (2010) 1599–1609.
[58] D.T. Mirikitani, N. Nikolaev, Recursive bayesian recurrent neural networks for time-series modeling, IEEE Trans. Neural Netw. 21 (2) (2010) 262–274.
[59] A. Gholipour, C. Lucas, B.N. Araabi, M. Mirmomeni, M. Shafiee, Extracting the main patterns of natural time series for long-term neurofuzzy prediction, Neural Comput. Appl. 16 (4–5) (2007) 383–393.
[60] M. Mirmomeni, C. Lucas, B. Moshiri, B.N. Araabi, Introducing adaptive neurofuzzy modeling with online learning method for prediction of time-varying solar and geomagnetic activity indices, Expert Syst. Appl. 37 (12) (2010) 8267–8277.



Anderson Sergio received the M.Sc in Computer Science in 2013 from Federal University of Pernambuco and the B.Sc in Cumputer Engineering in 2008 from University of Pernambuco, both in Brazil. He is currently working towards the pH. D degree in Computational Intelligence at the Federal University of Pernambuco. His current research interests include neural networks, evolutionary computation, swarm intelligence, hybrid systems and time series forecasting.

Tiago Lima received the B.Sc (2010) and M.Sc (2013) in Computer Science. He is currently working towards the pH. D degree in Computer Science at the Universidade Federal de Pernambuco, Brazil. His current research interests include neural networks, evolutionary computation, hybrid systems and applications of neural networks.

Teresa Ludermir received the pH.D. degree in Artificial Neural Networks in 1990 from Imperial College, University of London, UK. From 1991 to 1992, she was a lecturer at Kings College London. She joined the Center of Informatics at Federal University of Pernambuco, Brazil, in September 1992, where she is currently a Professor and head of the Computational Intelligence Group. She has published over 300 articles in scientific journals and conferences, three books in Neural Networks and organized two of the Brazilian Symposium on Neural Networks. Her research interests include weightless Neural Networks, hybrid neural systems and applications of Neural Networks.