2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Selecting and Combining Models with Self-Organizing Maps for Long-Term Forecasting of Chaotic Time Series

Rigoberto Fonseca-Delgado
Department of Computer Science
National Institute of Astrophisics, Optics and Electronics
Tonantzintla, Puebla, México
Email: rfonseca@inaoep.mx

Pilar Gómez-Gil
Department of Computer Science
National Institute of Astrophisics, Optics and Electronics
Tonantzintla, Puebla, México
Email: pgomez@acm.org

***Abstract*—When time series are generated by chaotic systems, a reasonable estimation of large prediction horizons is hard to obtain, but this may be required by some applications. Over the last years, some researchers have focused on the use of ensembles and meta-learning as a strategy for improving prediction accuracy. This paper addresses the problem of selecting and combining models for the design of efficient long-term predictors of chaotic time series based on meta-learning and self-organization. We propose and evaluate the use of four heuristic rules for selecting models using a self-organizing map (SOM) neural network and meta-features. The meta-features are extracted from the performances of each involved model when applied to the training time series. A trained SOM map, which was generated using these meta-features, allows the selection of models with diverse behaviors. Two strategies for the combination of models are compared; one is based on the average and a second is based on the median of the forecasts of the selected models. The experiments were executed using four types of series: the time series dataset provided by the NN5 tournament and time series generated from the Mackey-Glass equation, from an ARIMA model and from a sine function. In most cases, the best results were obtained using a percentage of the models belonging to the group that contained the best model. Our results also showed that a combination using a median strategy obtained better results that using an average strategy.**

I. Introduction

In the last years several researchers have found that the selection and combination of an appropriate set of models for time series forecasting achieves better results than other techniques [1]–[3]. However, to find the right models and combinations is a very complex problem. In general, there are two main strategies for solving this problem [4]: (1) analyzing different approaches and then using expert knowledge to provide guidelines to select forecasting models and (2) using the results of previous studies to estimate the relationship between data features and performance of the models. Such analysis involves models, parameters and data, and is repeated from the beginning with each new time series to be forecasted. An alternative is that an algorithm learns from the results of previous analysis, which is known as meta-learning; the knowledge generated from this is called meta-data [5]. This selection and combination could be done in different ways; in this work, we evaluated four rules for selecting models based on a self-organizing map (SOM) neural network [6]. In addition, we evaluated two ways of combining the selected models: first by averaging and second using the median of the predictions returned from the selected models.

Chaotic time series are cataloged as unpredictable, due their high sensibility to initial conditions [7]. Despite that, many applications deal with chaotic systems and require a reasonable estimation of future values. Many domains are looking for improvement in the accuracy obtained by current forecasting models such as financial applications, load forecasting or wind speed [8]. Nevertheless, the problem of predicting multi-step-ahead based on data captured from the chaotic system is still an open problem [8].

Recent works show a relation between chaos theory and the self-organization theory [9]–[11]. According to Helbing, the right approach to influence complex systems is to support and strengthen the self-organization and self-control of the system by a specific design of this mechanism [9]. There are several ways to induce self-organization using adaptive networks; some samples in one-step ahead prediction are [12]–[14]. Another successful model based on self-organization is the SOM neural network, proposed by Kohonen [6], which follows two well defined rules: compete and cooperate [15]. Based on these concepts, our goal is to build a system able to select and combine models using a SOM along with meta-features extracted from the performance of the different models with the time series. The SOM divides the models into groups, based on the representative error obtained by each involved model. The models to be combined are chosen from these groups. The group with the model that obtained the best expected accuracy are selected first. Then, with the aim of increasing the diversity between models, a second group are selected from the farthest neuron with respect to the first group. Once these two groups are defined, a percentage of models from each group is chosen. Four rules for choosing the models are used: first, always use the models coming from the two groups. Second, only use the first group. Third, use the first group and the second, if the expected accuracy of the best model in the second group is less than twice the best model in the first group. Fourth, use the first group and the second, if the expected accuracy of the best model in the second group is less than three times the best model in the first group. Once models are selected, their outputs are combined to calculate the output of the prediction. In our experiments, outputs are

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

combined in two ways: by the average of the output and by the median.

The involved models were built changing the main parameters of two base models: "Autoregressive Integrated Moving Average" (ARIMA) [16] and "Non-linear Autoregressive with eXogenous inputs" (NARX) [17]. The experiments were executed using four types of time series: the reduced set of the international forecasting tournament NN5 [18], time series generated using an ARMA model described in [16], a Mackey-Glass equation [19] and a sine function. Our experiments found that the combination of models using the median significantly improved the accuracy achieved by combining them with average. In addition, the rule using only the group which had the model with the best expected accuracy gave the best results.

This paper is organized as follows: section 2 describes fundamental concepts. Section 3 explains the proposed method and rules for selecting groups. Section 4 presents experiments comparing the different rules and different methods for combining. Finally, section 5 presents some conclusions and future directions for this research.

II. Fundamental Concepts

A. Time Series Forecasting

A time series is a set of observations y_t, each one being recorded at a specified time t at fixed intervals [20]. A sequence of h future values of a time series Y can be estimated based on a set of observations of Y, where h is known as the prediction horizon. A general expression of multi-step prediction may be described as:

$$\{y_{t+1}, y_{t+2}, \ldots, y_{t+h}\} = F\left(y_t, y_{t-1}, \ldots, y_{t-m+1}\right) \quad (1)$$

where F predicts h future values using m values of the past. Mainly, there are two forms to calculate h values of Y: one is estimating the complete horizon in a single iteration; the second strategy, known as iterative prediction and used in this research, consists of estimating one value each time, using the previous predicted value for calculating the next prediction [2].

B. Chaotic Time Series

The theory of chaos deals with complex non-linear systems; this theory had its breakthrough in the late 1800s, when Poincaré [21] addressed the stability of the solar system and the position of the planets. Abarbanel and Gollub [22] proposed that: "chaos is the deterministic evolution of a non-linear system which is between regular behavior and stochastic behavior or 'noise.'" Dhanya and Kumar [23] summarize the features of a chaotic system as: (i) they are deterministic, i.e., there are some determining equations ruling their behavior; (ii) they are sensitive to initial conditions, that is, a slight change in the starting point can lead to significantly different outcomes; (iii) they are neither random nor disorderly. Chaotic systems have a sense of order and patterns, even though they do not repeat. In this work, a chaotic time series is the outcome of a chaotic system.

The best approach to influence complex systems is to support and strengthen the self-organization and self-control of the system. This basically means that coordination and cooperation in a complex system will appear by itself, if the interactions among the system elements are well chosen. That is, regulations should not specify what exactly the system elements should do, but set bounds to actions, that is, to define "the rules of the game." This strategy provides the system with enough degrees of freedom to self-organize good solutions. If the interaction rules are suitable, this approach will usually lead to a much more flexible and adaptive system behavior. However, everything depends on the interactions of the system elements; for example, unsuitable interactions can cause that the system behaves dynamically unstable or that it gets trapped in a suboptimal state [9].

C. Self-Organizing Maps

Self-organizing maps (SOM), originally defined by T. Kohonen [6], consist of an input layer and one or several output layers. Input neurons connect to all output neurons; output neurons have some influences over their neighbors. In this research, we use a SOM with one output layer organized in a two-dimensional array. The relations between neurons have a hexagonal pattern, as shown at Figure 1. SOM is trained using a non-supervised algorithm that is executed a specific number of iterations. For each input pattern, a competition process among all neurons in the output layer takes place, and then the weights of the winning neuron and their neighbors are adapted. The winning neuron $i\left(x\right)$ is the one whose weights have the minimum Euclidean distance to the input pattern x, that is:

$$i\left(x\right) = \underset{i}{\operatorname{argmin}} \left\|x - w_j\right\|, j = 1, 2, \ldots, l \quad (2)$$

where l is the number of neurons. Weight updating is defined as:

$$w_j\left(n+1\right) = w_j\left(n\right) + \eta\left(n\right) h_{j,i(x)}\left(n\right)\left(x - w_j\left(n\right)\right) \quad (3)$$

where: n represents the current iteration, x is the input pattern, $\eta\left(n\right)$ is the learning rate parameter, starting with a value η_0, and then decreasing gradually as time n increases [15]. This requirement can be satisfied by an exponential decay for $\eta\left(n\right)$, as shown by:

$$\eta\left(n+1\right) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), n = 0, 1, 2, \ldots \quad (4)$$

τ_2 is a time constant. The function $h_{j,i(x)}$ defines the neighborhood of the winning neuron $i\left(x\right)$. For the results presented in this paper, the neighborhood was defined with a link distance, which is calculated as the number of links or steps that must be taken to get to the neuron under consideration. The neighborhood of a winning neuron with a link distance of 1 is shown in the Figure 1.

SOM networks have been used in the problem of long-term forecasting of time series in different ways [26], [27].

D. Accuracy Measures of Forecasting Models

Unless the expected prediction ability of a model is known, such model is hardly put into use in practice [28]. There are several metrics for measuring the expected ability of prediction of a model. In this paper, results are evaluated using three different functions. Let: Y be a time series, $y_t \in Y$ be the expected output at time t, $\hat{Y}$ be the sequence estimated by the model, $\hat{y}_t \in \hat{Y}$ be the model prediction at time t, and h the prediction horizon. The accuracy functions are:

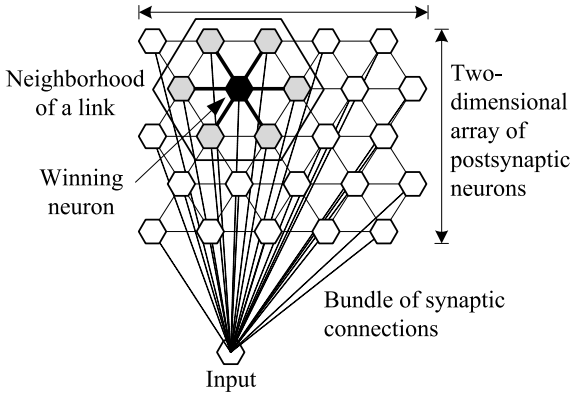

Fig. 1. Architecture of a Self-Organizing Map (SOM) based on [24], [25].

1) Root Mean Square Error (RMSE): It is a quadratic scoring rule, which measures the average magnitude of the error. RMSE is calculated as [29]:

$$\text{RMSE}\left(Y,\hat{Y}\right)=\sqrt{\frac{\sum_{t=1}^{h}\left(y_t-\hat{y}_t\right)^2}{h}} \tag{5}$$

Since the errors are squared before they are averaged, RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable [29]. The range of possible values obtained by RMSE depends of the magnitudes of the time series. To compare RMSEs among different time series requires to normalize the time series data.

2) Mean Absolute Error (MAE): This function measures the average magnitude of the errors in a set of forecasts, without considering their directions. It does not distinguish between variance and bias and it is appropriate when the cost function is linear [29]. MAE is defined as:

$$\text{MAE}\left(Y,\hat{Y}\right)=\frac{\sum_{t=1}^{h}\left|y_t-\hat{y}_t\right|}{h} \tag{6}$$

3) Symmetric Mean Absolute Percentage Error (SMAPE): SMAPE allows comparing different models with different time series. The tournament NN5 and other works [30] use the SMAPE function, whose results are values from 0 to 200. The value 0 means that the obtained prediction matches exactly with the expected output; the worst possible prediction implies a value of 200. The SMAPE function is given by [29]:

$$\text{SMAPE}\left(Y,\hat{Y}\right)=\frac{\sum_{t=1}^{h}\frac{\left|y_t-\hat{y}_t\right|}{\frac{1}{2}\left(y_t+\hat{y}_t\right)}}{h}\cdot 100 \tag{7}$$

E. Monte Carlo Cross-Validation (MCCV)

Cross-validation (CV) is a method commonly used to check the accuracy of models. It is more attractive than other methods, since it gives a statistical estimation of the expected prediction ability of the model [31]. CV requires splitting the samples into two parts: a training set and a validation set. When CV is applied in a model of time series forecasting, this division must maintain the order of observations. In other words, training and validating sets must contain consecutive values. CV follows these steps: first, fitting the model using the training set; second, obtaining a prediction of h steps ahead; and third, evaluating the prediction obtained $\hat{Y}$ comparing it with the known expected values of Y.

A particular method of CV is Monte Carlo cross-validation, (MCCV) [32] considered for some authors a simple and effective procedure [33]. The training process uses a set of observations r plus a random number of them, where the maximum number of random observations is $v=n-r-h$, being n the total number of observations. After training, the model predicts h values and this process is repeated k times. Figure 2 depicts this distribution of training and testing data. Finally, the SMAPE average of all iterations composes the MCCV. In a previous work [34], we found that the MCCV obtained good results in the task of selecting the best NARX model for time series forecasting.

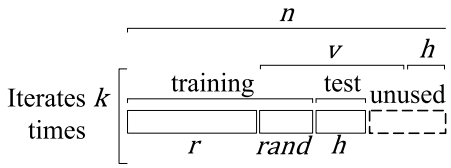

Fig. 2. Monte Carlo cross-validation takes a window of random size for training. The process iterates k times [34].

F. Base Models

The base models trained with different parameters generate several new models. For the experiments reported here, we used an ARIMA model [16] and the NARX neural networks [17] as base models, which are described next.

1) ARIMA Model: An autoregressive moving average model (ARMA) expresses the conditional mean of y_t as a function of p past observations $y_{t-1},\ldots,y_{t-p}$ and q past errors $\varepsilon_{t-1},\ldots,\varepsilon_{t-q}$. The general form of the ARMA (p,q) is [16]:

$$\begin{aligned} y_t=\phi_1 y_{t-1}+\ldots+\phi_p y_{t-p}+\\ \varepsilon_t+\theta_1\varepsilon_{t-1}+\ldots+\theta_q\varepsilon_{t-q}\end{aligned} \tag{8}$$

where ε_t is an uncorrelated innovation process with zero mean; ϕ_i and θ_i are determined from the data.

An autoregressive integrated moving average (ARIMA) model is generated adding or integrating an ARMA model d times. ARIMA models have shown an ability to obtain good results in the forecasting area. A detailed description of ARIMA model can be found in [16].

2) NARX Networks: An important class of discrete-time non-linear systems is the Non-linear Auto-Regressive with eXogenous Inputs (NARX) model: [17], [35]:

$$y_t=f\left(u_t,u_{t-1},\ldots,u_{t-n_u},y_{t-1},y_{t-2},\ldots,y_{t-n_y}\right) \tag{9}$$

where u_t and y_t represent the input and output of a non-linear function f at time t. In this work, exogenous variables are not involved. When f is approximated by a feed-forward neural network (FFNN) [15], the system is called a NARX network; Figure 3 shows its structure [25]. The first component is a tapped delay line (TDL); the input time series enters from the left and passes through m delays. The output of the TDL is an m-dimensional vector, made up of the input time series at the current time, and m previous observations [25]. The next

block is the hidden layer with several neurons. The right block is the output layer with only one neuron. As usual in FNNN, weights associated are adjusted during the training phase. After the NARX network is trained, its output is fed back to the input of the FFNN. The model iterates until it reaches h predictions. NARX had been evaluated in a theoretical and empirical way with good results that guarantee its effectiveness in time series forecasting [2], [36]. However, key components for this success are the selection of the learning algorithm and the determination of the right NARXs architectural elements, which is a difficult selection problem [36].

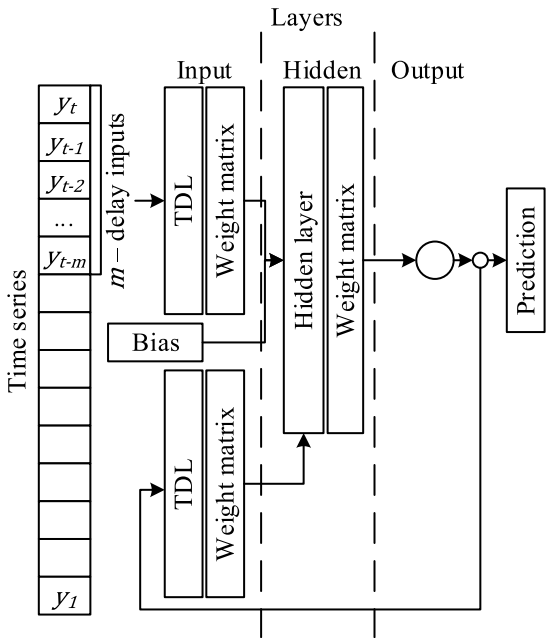

Fig. 3. NARX network structure with a tapped delay line (TDL) of m neurons and one neuron in the output layer. Image based on [25].

III. Proposed method

Our goal is to select and combine models with diverse behavior in the prediction horizon. Diversity has been recognized as a very important characteristic in combination of models [37], [38], therefore, it is a key component in our method. Figure 4 shows the proposed method. It contains two parts: extraction of meta-features and selection of models. Two types of meta-features are computed: performance and representative error. The performance of each model is estimated using a Monte Carlo cross-validation (MCCV) and the training time series.

For the second meta-feature, models are trained using $n-h$ values of the training time series, where n is the size of this series. Representative error is calculated as: $E = Y - \hat{Y}$, where Y contains the last h values of the training set and $\hat{Y}$ the last h predictions. When the involved model is a NARX network, the calculation of representative error is repeated k times, each time with a new training of the network. In these cases, the error is calculated as the average of these k errors. This is done to eliminate the randomness induced in the behavior of the performance of the model, originated by the fact that final values of the weights of a network depend upon the initial random values assigned to them during the training.

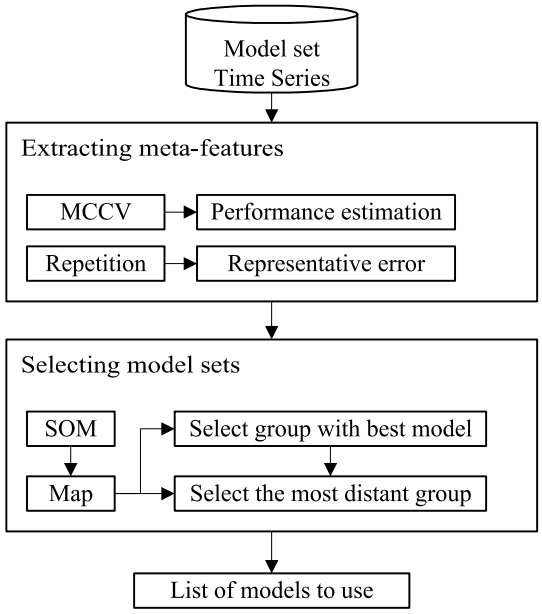

Fig. 4. General process for selecting prediction models. First to extract meta-features and then, using the meta-features, to select a list of models to use.

In the selection processes, a self-organizing map (SOM) clusters the models, using the vector of representative error of each model as an input pattern. The SOM creates a map where each neuron corresponds to a group of models. Two groups (neurons) are selected from this map: the one that contains the model with the best expected accuracy and the farthest neuron from that one. This distance is calculated using the positions of neurons in the map, as follows: let the neuron A be in position (A_r, A_c) and neuron B be in position (B_r, B_c). The Euclidean distance between neurons is given by:

$$\Delta(A, B) = \sqrt{(A_r - B_r)^2 + (A_c - B_c)^2} \qquad (10)$$

Once the two groups are selected, a list of models to use is built, which will contain a percentage p of models with the best expected SMAPE from each group. In this work, we define four kinds of rules for building the list of models to use:

1) The list always contains $p\%$ models from each group.
2) The list only contains $p\%$ models from the first group, which has the best model.
3) The list contains $p\%$ models from the first group and $p\%$ models from the second group, if the expected SMAPE of the second group is not larger than twice the expected SMAPE of the first group (Max. 2).
4) The list contains $p\%$ models from the first group and $p\%$ models from the second group, if the expected SMAPE of the second group is not larger than three times the expected SMAPE of the first group (Max. 3).

The model with the smallest expected SMAPE in the group defines the expected SMAPE of that group.

IV. Experimental analysis

In this section, we compared the four different rules for selecting models, which were previously described. The percentage of selected models from each group varied from 10% to 100%. The predictions of the selected models were combined in two ways: one with average and other with median. A combination was evaluated from its predictions with three functions: SMAPE, RMSE and MAE.

A. Data Description

Four types of time series were used to test our strategy; all the time series were normalized between 0 and 1. The first subset of time series was the reduced set provided by the NN5 prediction tournament [18], which consisted of 11 time series. Each training sequence contained 735 observations; the prediction horizon was composed of 56 future values for all series, which was referenced as test set. In a previous work [39], we found that this time series had a chaotic behavior.

The second time series resulted from integrating the Mackey-Glass differential equation [19]:

$$\frac{dx(t)}{dt} = \frac{ax\left(t-\tau\right)}{1+x\left(t-\tau\right)^{10}} - bx(t) \tag{11}$$

This function has a chaotic behavior with: $a = 0.2$, $b = 0.1$, $\tau = 17$, $x_0 = 1.2$; the time step was set to 0.1. The first 750 samples were used for training and the last 250 for testing. The third time series was generated using an ARMA$(2, 1)$ model defined as:

$$y_t = 0.5y_{t-1} - 0.3y_{t-2} + \varepsilon_t + 0.2\varepsilon_{t-1} \tag{12}$$

where ε_t follows a Gaussian distribution with mean 0 and variance 0.1. The first 500 values formed the training set and the last 50 the test set. A fourth series was generated using a sine function with a time step size of $2\pi/64$, the first 750 observations were used for training and the next 250 values for testing.

B. Prediction Models

Different models with the same NARX base were trained using different parameters. Notice that when a NARX is trained using different algorithms, its weight values would change as does its performance. For that reason, the experiments considered the training algorithm as a parameter. Three functions, available in the Matlab Neural network toolbox [25], were used for training the networks:

- Bayesian regulation back propagation (BP), "`trainbr`"
- Conjugate gradient BP with Fletcher-Reeves updates, "`traincgf`"
- Leveberg-Marquardt BP, "`trainlm`"

Two other parameters were used to generate the NARX models: the number of delay neurons $m = \{25, 30, 35\}$ and the number of neurons in the hidden layer $\{20, 25, 30\}$. In total, we generated 27 models with NARX form. The ARIMA models were generated changing four parameters: the number of autoregressive terms $p = \{0, 1, 2\}$, the number of non-seasonal difference $d = \{1, 2\}$, the number of lagged forecast errors $q = \{0, 1, 2\}$ and the seasonality $\{0, 7, 12\}$. In total, we generated 54 ARIMA models from a total of 81 models used in these experiments.

C. Results

The experiments were executed using a SOM with 5 rows and 5 columns, trained with $12,500$ iterations. This number was chosen following the recommendation given by Haykin [15], who suggests that the number of iterations would be 500 times the number of neurons.

Figure 5, 6 and 7, show the SOM maps generated using the time series No. 1 of the NN5 reduced set, Mackey-Glass and ARMA, respectively. Each neuron has an identification number, which is the first number inside the hexagon. The size of the hexagons in the figures represents the number of models in the group, which is also shown by the second number in brackets. The groups marked in dark gray with a letter "A" contain the models with the best expected SMAPE (referenced as subset A). The farthest group to subset A is referenced as subset B. Neurons marked in dark gray with a letter "W" contain the model with the worst expected SMAPE.

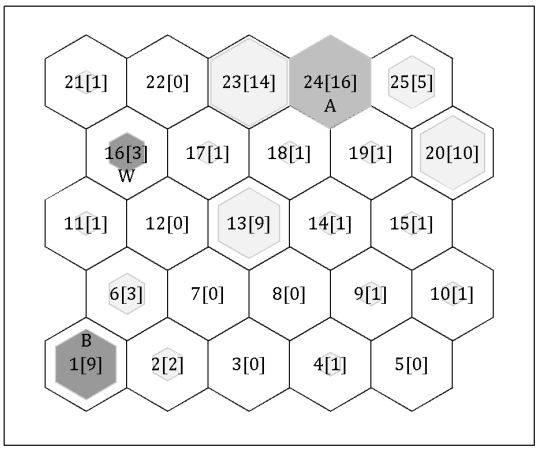

Fig. 5. SOM map with the time series 1 of the NN5 reduced set.

In Figure 5 for the time series 1 of the NN5 reduced set, subset A corresponds to group number 24 with 16 models. The best model was a NARX trained with "traincgf", $m = 25$ feedback delays and 20 neurons in the hidden layer; its expected SMAPE was 34.69. In the same case, B corresponds to group number 1 with 9 models. The best model in the subset B was an ARIMA with 0 auto-regressive terms, 1 lagged forecast error, 1 non-seasonal difference, and a seasonality of 12; its expected SMAPE was 106.46. Also, the subset B was far from the group with the worst expected SMAPE.

In Figure 6 for the time series Mackey-Glass, subset A was group number 9 with 31 models. The best model was a NARX with $m = 25$ feedback delays, 30 neurons in the hidden layer and trained with the "trainbr" algorithm; its expected SMAPE was 0.04. Subset B was group number 21 with 1 model. The model in subset B was an ARIMA with 1 auto-regressive term, 0 lagged forecast errors, 2 non seasonal differences, and a seasonality of 12; its expected SMAPE was 173.60. Also,

subset B was far from the group with the worst expected SMAPE.

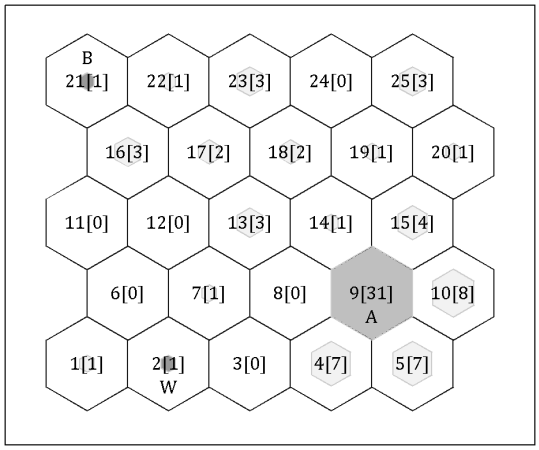

Fig. 6. SOM map with the time series Mackey-Glass.

In Figure 7 for the time series generated by a defined ARMA, subset A corresponds to group number 24 with 4 models. The best model was an ARIMA with 1 auto-regressive term, 2 lagged forecast errors, 0 non seasonal differences, and a seasonality of 12; its expected SMAPE was 141.39. Subset B corresponds to group number 1 with 2 models. The best model in subset B was an ARIMA with 1 auto-regressive term, 0 lagged forecast errors, 2 non-seasonal differences, and a seasonality of 7; its expected SMAPE was 191.37. Subset B contains the model with the worst expected SMAPE.

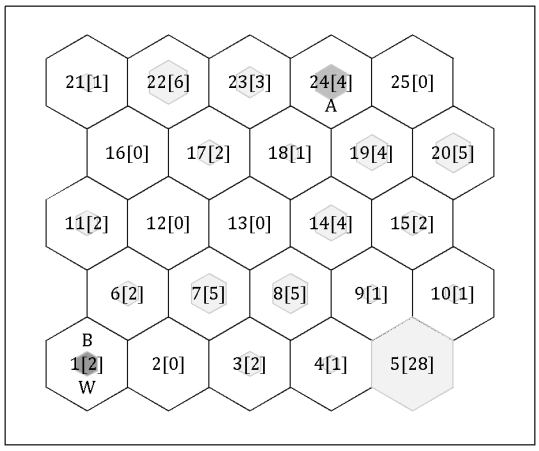

Fig. 7. SOM map with the time series generated with an ARMA model.

Figure 8 shows the metrics SMAPE, RMSE and MAE obtained by the proposed method with different percentages of selected models to be combined. The forecast values of selected models were combined by average. The results plotted are the averages obtained with all the time series for every single rule. Figure 9 shows the same for results obtained by a combination based on the median.

It can be seen in Figures 8 and 9 that the model combination based on the median has a better performance than a model combination based on the average. These figures also show that, when the median is used, fewer models are needed to achieve good results. This is because the median is not affected by the extreme values in the predictions. With the aim of evaluating whether the difference in results (dr) between combining models with median and combining models with average has a statistical significance, we defined a null hypothesis assuming that the dr is equal to 0. The alternative hypothesis is that there exists a difference among combination methods in average. A t-test [40] was executed for every single rule and evaluation function. Table I shows the results of these tests. The columns are the evaluation functions and the rows are the rules. A value 1 indicates that the t-test rejected the null hypothesis; a value 0 shows that the null hypothesis was not rejected. These tests confirmed that the difference between the two methods is statistically significant with most of the evaluation functions and rules. In other words, combining models with median obtained results statistically better than combining with average in most cases.

TABLE I. STATISTICAL SIGNIFICANCE ANALYSIS FOR DIFFERENT RULES AND EVALUATION FUNCTIONS. A VALUE "1" MEANS THAT THE DIFFERENCE AMONG COMBINATION STRATEGIES AVERAGE AND MEDIAN IS STATISTICALLY SIGNIFICANT.

	Evaluation functions		
Rules	RMSE	MAE	SMAPE
Rule 1	1	1	1
Rule 2	0	0	1
Rule 3	1	1	1
Rule 4	1	1	1

It is also clear that, even when the SOM separated the models based on the representative error, in most cases, the farthest group did not have the model with the worst expected SMAPE. However, the models of the farthest group sometimes did not have a good expected SMAPE compared with the model with the best SMAPE. Then, the selection of groups based on the rules of excluding groups with more than two or three times the error had the same performance. In general, the best results were obtained with the second rule, which is only to combine the models of the subset A.

V. CONCLUSIONS AND FUTURE DIRECTIONS

In this work, we presented four rules for selecting models based on a self-organizing map and two ways of combining the model predictions. Combining the models using the median obtained the best results, because the median is not affected by extreme values in the predictions. The best rule for selecting models was to choose only the group with the model with the best expected SMAPE. The farthest group, based on the representative error, had a different behavior but its models did not have a good expected SMAPE. In future work we will analyze the results of selecting models considering the neighborhood of the group with the best expected SMAPE. In addition, the behavior of the models in different prediction windows will be analyzed.

ACKNOWLEDGMENT

R. Fonseca thanks the National Council of Science and Technology (CONACYT), México, for the scholarship granted to him, No. 234540. This research has been partially supported by CONACYT, project grant No. CB-2010-155250. R. Fonseca also wishes to thank Rebekah Clark for her valuable assistance.

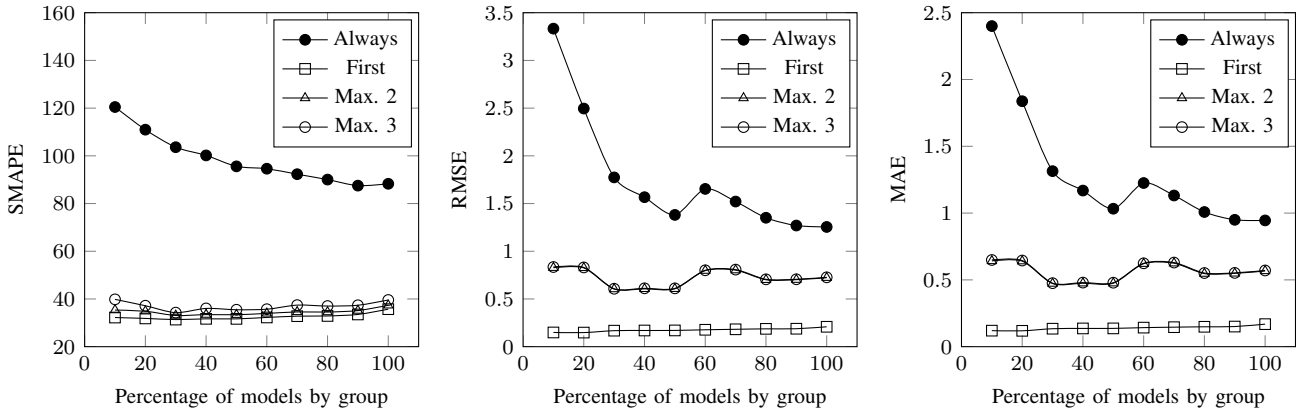

Fig. 8. Performance of the proposed method for each rule, using average as combination strategy. Results of each rule are marked as: "Always" for #1, "First" for #2, "Max. 2" for #3 and "Max. 3" for #4.

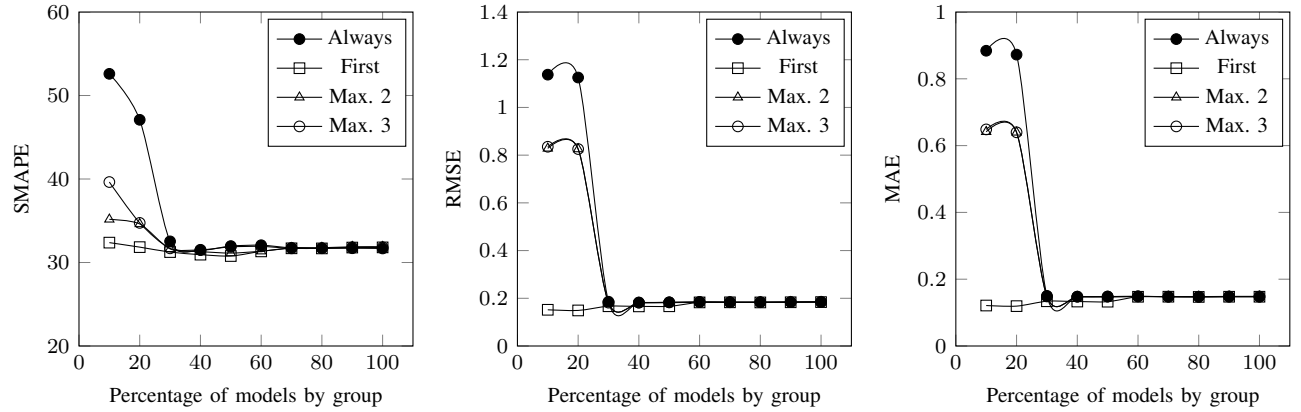

Fig. 9. Performance of the proposed method for each rule, using median as combination strategy. Results of each rule are marked as: "Always" for #1, "First" for #2, "Max. 2" for #3 and "Max. 3" for #4.

REFERENCES

[1] C. Lemke and B. Gabrys, "Meta-learning for time series forecasting and forecast combination," *Neurocomputing*, vol. 73, no. 10-12, pp. 2006 – 2016, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231210001074

[2] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 635 – 660, 2011.

[3] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451 – 476, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207000000571

[4] X. Wang, K. Smith-Miles, and R. Hyndman, "Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series," *Neurocomputing*, vol. 72, no. 10-12, pp. 2581 – 2594, 2009.

[5] C. Castiello and A. M. Fanelli, "Computational intelligence for meta-learning: A promising avenue of research," in *Meta-Learning in Computational Intelligence*, ser. Studies in Computational Intelligence, N. Jankowski, W. Duch, and K. Grabczewski, Eds. Springer Berlin Heidelberg, 2011, vol. 358, pp. 157–177.

[6] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[7] H. Kantz, "A robust method to estimate the maximal lyapunov exponent of a time series," *Physics Letters A*, vol. 185, no. 1, pp. 77 – 87, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0375960194909911

[8] J. G. De-Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443 – 473, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207006000021

[9] D. Helbing, *Social Self-Organization Agent-Based Simulations and Experiments to Study Emergent Social Behavior*, J. S. Kelso, Ed. Springer-Verlag, 2012.

[10] H. Haken, *Information and Self-Organization A Macroscopic Approach to Complex Systems*, 3rd ed. Springer Berlin, 2006, no. ISBN-10 3-540-33021-6.

[11] S. A. Kauffman, *The Origins of Order Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[12] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of anfis models," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3494 – 3506, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411013492

[13] J. Soto, P. Melin, and O. Castillo, "Time series prediction using ensembles of neuro-fuzzy models with interval type-2 and type-1 fuzzy integrators," in *Proceedings of International Joint Conference on Neural Networks*. IEEE, 2013, pp. 189 – 194.

[14] M. Pulido, P. Melin, and O. Castillo, "Optimization of ensemble neural networks with type-2 fuzzy response integration for predicting the mackey-glass time series," in *Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on*, Aug 2013, pp. 16–21.

[15] S. Haykin, *Neural Networks A Comprehensive Foundation*, 2nd ed. Pearson Prentice Hall, 1999.

[16] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis Forecasting and Control*, 3rd ed., J. Grant, Ed. Prentice-Hall Interna-

tional, 1994.

[17] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems part ii: stochastic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 329–344, 1985. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/0020718508961130

[18] S. F. Crone. (2010, Feb) Competition instructions. Web. BI3S-lab. Hamburg, Germany. [Online]. Available: http://www.neural-forecasting-competition.com/instructions.htm

[19] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[20] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, 2nd ed. Springer, 2006.

[21] H. Poincare, "Memoire sur les courbes definies par une equation differentielle." *Resal J.*, vol. 3, pp. VII. 375–422, 1881.

[22] H. D. Abarbanel and J. P. Gollub, "Analysis of observed chaotic data," *Physics Today*, vol. 49, p. 86, 1996.

[23] C. Dhanya and D. N. Kumar, "Nonlinear ensemble prediction of chaotic daily rainfall," *Advances in Water Resources*, vol. 33, no. 3, pp. 327 – 347, 2010.

[24] P. Gómez-Gil. (2010, 07) Advanced neural networks. INAOE. [Online]. Available: http://ccc.inaoep.mx/~pgomez

[25] M. H. Beale, M. T. Hagan, and B. Demuth, Howard, *Neural Network Toolbox User's Guide R2012b*. MathWorks, 2012.

[26] A. Sorjamaa and A. Lendasse, "Time series prediction as a problem of missing values: Application to estsp2007 and nn3 competition benchmarks," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, Aug 2007, pp. 2948–2953.

[27] A. Sorjamaa, F. Corona, Y. Miche, P. Merlin, B. Maillet, E. Severin, and A. Lendasse, "Sparse linear combination of soms for data imputation: Application to financial database," in *Advances in Self-Organizing Maps*, ser. Lecture Notes in Computer Science, J. C. Principe and R. Miikkulainen, Eds. Springer Berlin Heidelberg, 2009, vol. 5629, pp. 290–297. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02397-2_33

[28] H. A. Martens and P. Dardenne, "Validation and verification of regression in small data sets," *Chemometrics and Intelligent Laboratory Systems*, vol. 44, no. 1-2, pp. 99 – 121, 1998.

[29] J. S. Armstrong, *Long-range forecasting from crystall ball to computer*, 2nd ed. John Wiley & Sons, 1985, no. ISBN 0-471-82360-0.

[30] S. F. Crone, K. Nikolopoulos, and M. Hibon, "Automatic modelling and forecasting with artificial neural networks - a forecasting competition evaluation," International Institute of Forecasters, Final Report for the IIF/SAS Grant 2005/6, 04 2008.

[31] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273 – 324, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000437029700043X

[32] R. R. Picard and R. D. Cook, "Cross-validation of regression models," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 575–583, 1984. [Online]. Available: http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1984.10478083

[33] Q.-S. Xu, Y.-Z. Liang, and Y.-P. Du, "Monte carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration," *Journal of Chemometrics*, vol. 18, no. 2, pp. 112–120, 2004. [Online]. Available: http://dx.doi.org/10.1002/cem.858

[34] R. Fonseca-Delgado and P. Gómez-Gil, "An assessment of ten-fold and monte carlo cross validations for time series forecasting," *To be published in the Proccedings of the 2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2013.

[35] H. Siegelmann, B. Horne, and C. Giles, "Computational capabilities of recurrent narx neural networks," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27, no. 2, pp. 208–215, 1997.

[36] E. Diaconescu, "The use of narx neural networks to predict chaotic time series," *WSEAS Trans. Comp. Res.*, vol. 3, no. 3, pp. 182–191, Mar. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1466884.1466892

[37] P. Cunningham and J. Carney, "Diversity versus quality in classification ensembles based on feature selection," in *Machine Learning: ECML 2000*, ser. Lecture Notes in Computer Science, R. Lopez de Mantaras and E. Plaza, Eds. Springer Berlin Heidelberg, 2000, vol. 1810, pp. 109–116. [Online]. Available: http://dx.doi.org/10.1007/3-540-45164-1_12

[38] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003. [Online]. Available: http://dx.doi.org/10.1023/A%3A1022859003006

[39] R. Fonseca-Delgado and P. Gómez-Gil, "Temporal validated meta-learning for long-term forecasting of chaotic time series using monte-carlo cross-validation," in *To be published in the Proccedings of the International Seminar of Computational Intelligence ISCI 2013, Tijuana, Mexico*. IEEE - Computational Intelligence Society, Mexico-Chapter, 2013.

[40] G. K. Kanji, *100 Statistical Tests*, 3rd ed. Sage Publications, 2006, no. ISBN-10 1 4129 2375 1.