Dynamic fuzzy systems for modelling non-stationary time series

Ivette Luna
Departamento de Teoria Econômica, DTE
Instituto de Economia, IE
Universidade Estadual de Campinas, UNICAMP
Email: ivette@eco.unicamp.br

***Resumo*—Time series modelling is a subject of interest to several fields of knowledge. The challenge of dealing with this type of data is to develop computational instrumentals able to handle with different features related to non-stationarity as well as uncertainty present in most of real processes. In that context, dynamic fuzzy systems have shown capable of dealing with these aspects. Hence, this paper presents a case study that aims to provide empirical evidence that validate the ability of dynamic fuzzy systems for modelling and forecasting non-stationary time series. The dynamic fuzzy model is based on Takagi-Sugeno fuzzy systems, with a learning algorithm based on the recursive version of the Expectation Maximization optimization method. The study considers the modelling of a bond price time series. The model is evaluated with and without the dynamic learning in order to verify the effect of the learning process over the model performance. Additionally, the fuzzy model is also compared to an offline neural network. The results show the potential of fuzzy systems with dynamic learning for modelling non-stationary time series and changes over time.**

I. Introduction

Several problems arise in the process of obtaining models for time series, beginning with the availability of data, the selection of the input variables, the set of hypotheses to consider as a basis for the development of the model structure and the proper choice of the optimization algorithm in order to achieve an appropriate model in terms of structure, performance and robustness. A variety of these processes arise in various sectors such as the energy sector, financial and economics among others, where processes are most often admittedly not stationary.

Most of the real problems from which time series are collected be highly non-linear, and the application of linear estimation methods may lead to significant misspecification of both the structure and dynamics of the system. A second major problem arises from the fact that many of the theoretical concepts underlying empirical models are actually quite vague and there is considerable uncertainty about the precise meaning and range of key input variables.

In recent years, dynamic models based on computational intelligence tools have emerged as powerful new tools for the development of time series models, where the ability to adaptation to changes in the dynamics of the system is critical. Among all these tools in the area of computational intelligence, dynamic fuzzy systems show capable of handling the uncertainties inherent to those real problems. Furthermore, the fuzzy systems allow the extraction of knowledge through the analysis based on their structure rules, which is favorable in analyzing complex problems in fields such as automatic control, data classification, decision analysis, expert systems, and time series forecasting [1].

The main advantage of these models is the simultaneous adjustment of parameters and the structure, while providing structures under the principle of parsimony and the balance between complexity and performance, as well. A second advantage is the nature of the learning algorithm. Because of the dynamic algorithm, the model is adjusted at each iteration of the optimization process considering only recent information, which is advantageous in terms of computing power when compared to models with batch learning or *offline*. The works detailed in [2], [3] and [4] are just some examples proposals on models with dynamic learning.

In that context, this paper presents a case study that aims to provide empirical evidence that validate the ability of dynamic fuzzy systems for modelling and forecasting non-stationary time series. The dynamic fuzzy model used for this purpose is the one proposed in [5], which is based on Takagi-Sugeno fuzzy systems [6].

The model structure is defined in two phases. In the first phase, an initial rule based system composed by a set of fuzzy rules is generated using a Subtractive Clustering algorithm (SC), originally proposed in [7], as well as the offline Expectation Maximization algorithm [8] for the initial adjustment of all the parameters.

In a second phase, the model is dynamically modified by a recursive Expectation Maximization algorithm. This adaptation depends on the complexity of the problem and the input space partition at each step. That is, input space is divided into a number of subspaces, and a local model is assigned for each one of these regions. Because input partition is modified at each iteration, learning implies in parameters and structure adjustment.

The fuzzy model is evaluated with and without the dynamic learning in order to verify the effect of the learning process over the model performance. Results are also compared to the ones obtained by an offline multilayer neural network. The case study is developed considering the modelling of the Brazilian sovereign bonds time series (Global 40), a non-linear time series from 2000 to 2007. Results show empirical evidence in favor of dynamic fuzzy models against offline models when dealing with non-stationary time series. After this introduction, the paper proceeds as follows. Section II presents the fuzzy inference system and the learning method proposed. Section III the data and simulation results. Finally, some conclusions are presented in Section IV.

II. Fuzzy inference system

This section introduces the general structure of the fuzzy inference system and the learning algorithm for model structure and parameters update.

A. Model structure

Let $\mathbf{x}^k = [x_1^k,\ x_2^k,\ \ldots,\ x_p^k] \in \mathbb{R}^p$ denotes the input vector at instant k, $k \in \mathbb{Z}_0^+$; $\hat{y}^k \in \mathbb{R}$ is the output model, for the correspondent input $\mathbf{x}^k$. The input space represented by $\mathbf{x}^k \in \mathbb{R}^p$, is partitioned into M sub-regions, and each of these is represented by a fuzzy rule; $k = 0, 1, 2, \ldots$ is the time index (Figure 1). The antecedents of each fuzzy **If-Then** rule (R_i) are represented by their respective centers $\mathbf{c}_i \in \mathbb{R}^p$ and covariance matrices $\mathbf{V}_i|_{p\times p}$. The consequents are represented by local linear models, with output y_i, $i = 1, \ldots, M$ defined by:

$$y_i^k = \phi^k \times \theta_i{}^T \tag{1}$$

where $\phi^k = [1\ x_1^k\ x_2^k\ \ldots x_p^k]$; $\theta_\mathbf{i} = [\theta_{i0}\ \theta_{i1}\ \ldots\ \theta_{ip}]$ is the coefficient vector of the local linear model for the $i-th$ rule.

Each input pattern has a membership degree associated with each region of the input space partition. This is calculated through membership functions $g_i(\mathbf{x}^k)$ that vary according to centers and covariance matrices related to the fuzzy partition, and are computed by:

$$g_i(\mathbf{x}^k) = g_i^k = \frac{\alpha_i \cdot P[\ i \mid \mathbf{x}^k\ ]}{\sum_{q=1}^{M} \alpha_q \cdot P[\ q \mid \mathbf{x}^k\ ]} \tag{2}$$

where α_i are positive coefficients satisfying $\sum_{i=1}^{M} \alpha_i = 1$ and $P[\ i \mid \mathbf{x}^k]$ is defined according to

$$P[\ i \mid \mathbf{x}^k\ ] = \frac{1}{(2\pi)^{p/2}\det(\mathbf{V}_i)^{1/2}} \times \exp\left\{-\frac{1}{2}(\mathbf{x}^k - \mathbf{c}_i)\mathbf{V}_i^{-1}(\mathbf{x}^k - \mathbf{c}_i)^T\right\} \tag{3}$$

where $\det(\cdot)$ is the determinant function. The model output $y(k) = \hat{y}^k$, which represents the predicted value for future time instant k is calculated by means of a non-linear weighted averaging of local outputs y_i^k and its respective membership degrees g_i^k, i.e.

$$\hat{y}(\mathbf{x}^k) = \hat{y}^k = \sum_{i=1}^{M} g_i^k\ y_i^k \tag{4}$$

B. Learning algorithm

First, an initial structure composed by fuzzy rules is defined, and its parameters are adjusted via the traditional Expectation Maximization algorithm, originally proposed in [8] for mixture of experts models.

Model structure is initialized using the unsupervised clustering algorithm called the Subtractive Clustering Algorithm (SC), proposed in [7]. This algorithm provide a set of M clusters from an specific training data set presented to the algorithm. Patterns processed by the SC algorithm are composed by the input-output patterns used in a second stage for model optimization.

These groups are associated to a set of fuzzy rules codified in the FIS structure. Therefore, after the number of fuzzy rules is defined, we proceed to initialize the model parameters, for $i = 1, \ldots, M$, according to the following criteria:

- $\mathbf{c}_i^0 = \psi_i^0|_{1\ldots p}$, where $\psi_i^0|_{1\ldots p}$ is composed by the first p components of the $i-$th center found by the SC algorithm;
- $\sigma_i^0 = 1.0$;
- $\theta_i^0 = [\psi_i^0|_{p+1}\ 0\ \ldots\ 0]_{1\times p+1}$, where $\psi_i^0|_{p+1}$ is the $p+1-$th component of the $i-$th center found by the SC algorithm;
- $\mathbf{V}_i^0 = 10^{-4}\mathbf{I}$, where $\mathbf{I}$ is a $p \times p$ identity matrix;
- $\alpha_i^0 = 1/M$.

Initialization is finalized with the model optimization of all the parameters using the traditional offline EM algorithm [8]. After this stage we already have an adjusted model ready for estimation purposes. However, as mentioned before, financial time series usually present multiple overlying seasonality, trends, structural breaks and unknown and non-observed causal forces that impact the phenomenon. In that sense, the necessity of a kind of model with the capability of self-adaptation to all these changes is suitable.

Adaptation presents advantages that go from automatic structure and parameter selection to adaptation to changes in the reasoning environment. Adaptation at each iteration considers a window size (T) over time. That is, the last T patterns will influence model parameters and structure. Therefore, after the FIS has been initialized, it may activate its capability of self-adaptation by using a recursive version of the EM algorithm, as well as adding and pruning operators.

Based on the offline EM algorithm, a recursive version can be written, obtaining the following estimates:

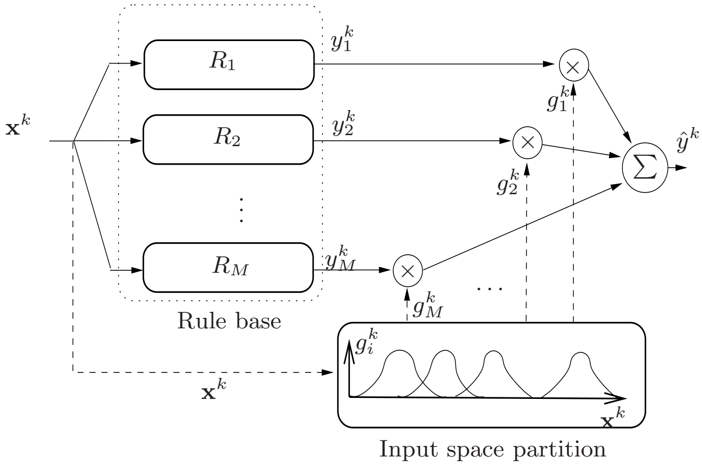

Figura 1. A general FIS structure.

$$\alpha_i^{k+1} = \alpha_i^k + \frac{1}{T}[h_i^k - \alpha_i^k] \tag{5}$$

$$\mathbf{c}_i^{k+1} = \mathbf{c}_i^k + \frac{1}{\gamma_i^{k+1}}[\mathbf{x}^k - \mathbf{c}_i^k] \tag{6}$$

$$\mathbf{V}_i^{k+1} = \mathbf{V}_i^k + \frac{1}{\gamma_i^{k+1}}[(\mathbf{x}^k - \mathbf{c}_i^k)(\mathbf{x}^k - \mathbf{c}_i^k)' - \mathbf{V}_i^k] \tag{7}$$

$$(\sigma_i^2)^{k+1} = (\sigma_i^2)^k + \frac{1}{\gamma_i^{k+1}}[(y^k - y_i^k)^2 - (\sigma_i^2)^k] \tag{8}$$

where:

$$\frac{1}{\gamma_i^{k+1}} = \frac{h_i^{k+1}}{\sum_{t=1}^{k+1} h_i^t} \tag{9}$$

An approximation of $\sum_{t=1}^{k+1} h_i^t$ can be built considering the window size T and the recursive equation form inspired in the adaptive learning of a fuzzy system detailed in [9]. Let $S_i^{k+1} = \sum_{t=1}^{k+1} h_i^t$, and $S_i(\mathbf{x}^{k+1}) = h_i^{k+1}$. Then S_i^{k+1} can be estimated as:

$$S_i^{k+1} \approx S_i(\mathbf{x}^{k+1}) + \frac{T-1}{T} S_i^k \tag{10}$$

Parameters θ_i are estimated applying a weighted recursive least square algorithm (wRLS), which considers a forgetting factor over time f_{forget} [10]. Equations of the RLS algorithm adapted to our problem are defined as:

$$\theta_i^{k+1} = \theta_i^k + \mathbf{C}_i^{k+1}\phi^{\mathbf{k}} \times h_i^k(y^k - y_i^k) \tag{11}$$

where:

$$\mathbf{C}_i^{k+1} = \frac{\mathbf{C}_i^k}{f_{forget}^k + h_i^k(\phi^k)^T \mathbf{C}_i^k \phi^k} \tag{12}$$

is the covariance matrix associated with each θ_i during the online adaptation. The forgetting factor $f_{forget} \in (0, 1]$. To guarantee stability f_{forget} was slowly increased so that after a long time $f_{forget}^k \to 1.0$.

Initial conditions for $\theta_i^0,\ i = 1, \ldots, M$ were given by the values obtained through model initialization, while $\mathbf{C}_i^0 = \mu\mathbb{I}$, where $\mu = 10^4$ and $\mathbb{I}$ is an identity matrix with dimensions $p+1 \times p+1$.

After the initialization phase, online adaptation was undertaken via structure modification based on adding and pruning operators and parameters update using Equations (5)-(12).

- ***Adding:*** The criterion to judge whether to generate a new fuzzy rule was based on the if-part criterion, which verifies if some existing fuzzy rule clusters the input vector. Assuming a normal input data distribution, with a confidence level of $\gamma\%$, we can construct a confidence interval $[\mathbf{c}_i - z_\gamma\sqrt{diag(\mathbf{V}_i)}, \mathbf{c}_i + z_\gamma\sqrt{diag(\mathbf{V}_i)}]$, where $diag(\mathbf{V}_i)$ is the main diagonal of the covariance matrix $\mathbf{V}_i$. In this paper, we get a confidence level of $\gamma = 72,86\%$ which requires a z_γ value of 1.1, obtained from the normal distribution table. It is clear that $\gamma = 72,86\%$ is the middle chunk, leaving 13,57% probability excluded in each tail. That is:

$$\max\left(P[\ i \mid \mathbf{x}^k\ ]\right)_{i=1,\ldots,M} > 0.1357 \tag{13}$$

  If this condition is not satisfied, it means that there is no rule that can cluster this input vector, that is, the input pattern is not covered by any fuzzy partition. Hence, it is necessary to add a new rule to the structure, expanding the input space region detected during the initialization phase or before the actual time instant k. If it happens, a new rule is generated with the next initialization:
  - $\mathbf{c}_{M+1}^{k+1} = \mathbf{x}^k$
  - $\sigma_{M+1}^{k+1} = 1.0;$
  - $\theta_{M+1}^{k+1} = [y^k\ 0\ \ldots\ 0]_{1\times p+1}$

- $\mathbf{V}_{M+1}^{k+1} = 10^{-4}\mathbf{I}$, where $\mathbf{I}$ is a $p \times p$ identity matrix;
- $\alpha_{M+1}^{k+1} = 10^{-5}$.

Even though this value is too small to interfere in the dynamic of the actual structure, all the α_i, $i = 1, \ldots, M+1$ are re-normalized, so that the sum of all these coefficients will always be equal to the unity.

- ***Pruning:*** According to the offline Em algorithm, $\alpha_i = \frac{1}{N}\sum_{k=1}^{N} h_i^k$. Thus, α_i can be considered a measure of the importance that each fuzzy rule has for the corresponding topology when compared to the other rules. It occurs because α_i is proportional to the sum of all posterior estimates of membership functions g_i^k over all the data set. Hence, a threshold for α_i is defined, so that every rule with $\alpha_i < \alpha_{min}$ at each iteration is pruned and eliminated from the actual model structure. However, after a new rule is created, its corresponding α_i will have a small value. If the pruning operator were applied immediately, the new rule would thus be eliminated and there would be no time to verify its relevance for the model structure.
This problem is resolved by the creation of a new index, called index of permanence τ. Every time a new rule is created, its respective $\tau_i$will also be created. As this rule is activated over time, this index is increased, that is $\tau_i^{k+1} = \tau_i^k + 1$. Thus, a rule will be a candidate pruning only if its α_i is very small and $\tau_i^k > \epsilon T$, where $\epsilon > 0$ and T is the same window size used during the sequential learning. This condition ensures that no new rule will be pruned immediately after its creation, allowing it to adjust for a minimum period of time and avoiding useless and abrupt oscillations in the model structure.

III. Case study

As mentioned in Section I, the aim of this case study is to provide empirical evidence that show the gains resulting from the use of a dynamic model for a non-stationary time series. To delimit the case study, it considered a particular case of a non-stationary time series, which presents a trend component, implying at least a change in the first moment of the series (mean) over time. For this purpose, we considered the bond price time series namely the Brazilian Global 40, illustrated in Fig. 2. Data goes from November 8^th, 2000 to January 16^th, 2007, with a total of 1668 observations on a daily basis.

The FIS model was run in two modes, the online and the offline mode. The online mode performed the model structure and parameters update permanently using the recursive algorithm detailed in Section II. Only the first 165 input-output patterns were used during the initialization stage, corresponding to approximately 10% of the total of available data. Parameter T, which represents a window size over time, was set up in 21. On the other hand, $r_a = 0.05$; $r_{ba} = 1.0$ $\alpha_{min} = 0.001$ and $f_{forget} = 0.95$.

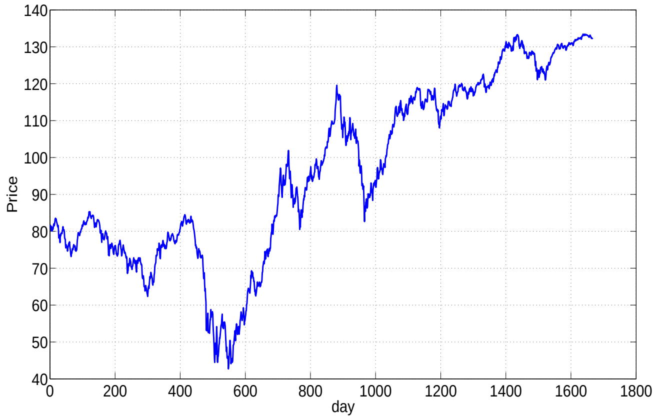

Figura 2. Global 40 bond price.

The offline mode of FIS consisted of defining the model structure by the SC algorithm, while the parameters adjustment was performed by the offline version of the EM algorithm. For doing so, the data set was divided in two subsets: the training dataset and the testing dataset. The first one was used for the adjustment of the offline FIS (batch mode), whereas the second one was used for validation purposes. The testing data set was composed by the last thousand of input-output patterns available.

Likewise, a multilayer neural network (MLP) was adjusted with a gradient descent algorithm as optimization technique. All the three models were evaluated for a one step ahead forecasting task, using as performance index the root mean square error (RMSE).

Two experiments were carried out; the processing of the series:

(a) in its first difference (a stationary series) and

(b) in level (a non-stationary series).

Therefore, as a first part of the study, all the three models processed the first difference of the series. The differentiation aims to remove the trend component and consequently, transforms the series into an stationary one, which is verified applying the ADF teste over the series in its first difference. The second part considers the processing of the series with the trend component.

Input selection was performed using a nonparametric approach known as the partial mutual information (PMI) criterion [11]. The PMI criterion is a measure of the partial or additional information that a new input can add to the existing prediction model [12]. PMI values achieved for the first twelve lags of the stationary series as possible explanatory variables are shown in Fig. 3.

Considering the most significative values of the PMI achieved by each lag, the final set of input variables was composed by lags 1, 3, 6 and 8. The insertion of lag 5 as input variable did not bring any gain in the models performance. The variation of the number of rules in the online FIS during the testing data set is illustrated in Fig. 4, with a number of rules in the model structure varying

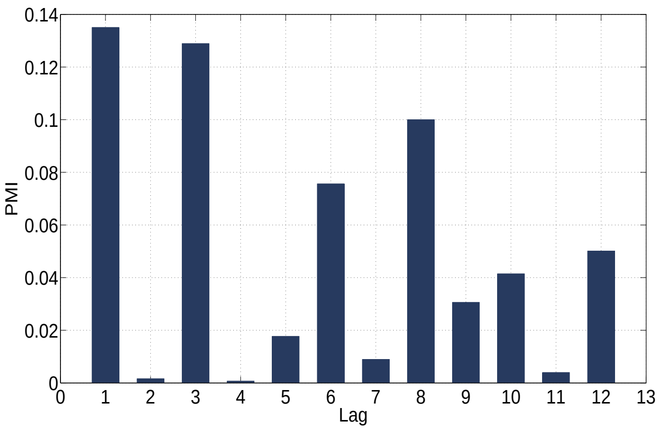

Figura 3. Input selection - PMI.

from 2 to 8. For comparison purpose, the same set of input variables was used by all the models.

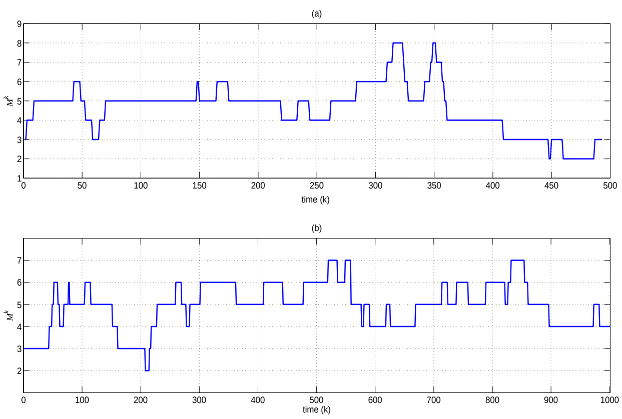

Figura 4. Evolution of the number of rules - online FIS.

In terms of RMSE, when the models have to deal with the stationary time series, there is no significative difference in the RMSE of the three models, as observed in the first line of Table I. However, the offline ones were not capable of dealing with the changes over the mean of the series when the series had to be processed in its original version (in level). RMSE achieved for each model over the non-stationary time series is presented in the second line of Table I.

Forecasting results are also depicted in Fig. 5, where we can see the difficulty that offline FIS and the MLP model have to follow the unexpected behaviour of the price over time; this because those models have no ability of extrapolation. On the other hand, the dynamic nature of the online model allow the model to adapt permanently being thus able to handle these variations resulting from the non-stationary nature of the process.

Thus, it is important to point that these results do not intend to favor any model, but to emphasize the advantage that dynamic models offer when the hypothesis of a stationarity process is not guaranteed, as it happens in several real-world problems.

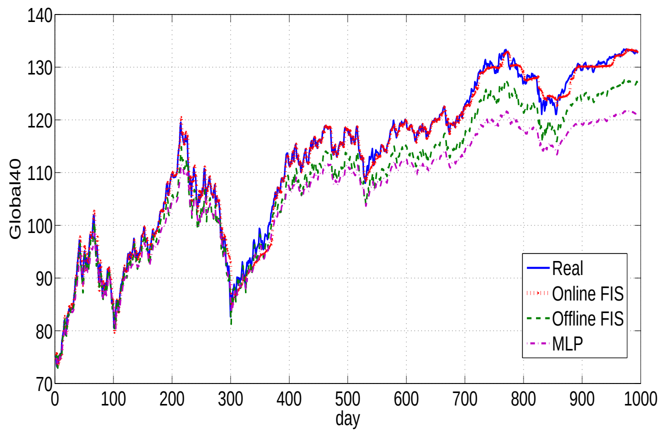

Figura 5. Results over the non-stationary time series.

IV. Conclusions

This paper presents a case study that considers the comparison of a model with dynamic learning with other two models with offline or batch learning in order to determine the circumstances where a dynamic model could present some advantages in terms of a one step ahead RMSE index. The dynamic modelo is based on a Takagi-Sugeno fuzzy system, whit an online learning based on the recursive version of the EM algorithm. As reference to determine possible gains using this model, it was compared to its offline version and a conventional MLP model, in a one step ahead bond price time series forecasting. Thus to delimit the case study, it was considered a time series with the presence of non-stationary trend component. The results show that in the case of stationary processes, the superiority of any model is not guaranteed, since all the three models are able to deal with this kind of problem. Moreover, in the case of non-stationary processes, more specifically with trend component, the dynamic model considered significantly outperformed the offline models. Therefore, in those cases, the dynamic model shows to be more adequate for dealing with changes over time, and therefore, for the analysis of dynamical systems. Future research will consider the analysis of online models for other non-stationary processes.

Referências

[1] P. Angelov, D. P. Filev, and N. Kasabov, *Evolving Intelligent Systems: Methodology and Applications*. Wiley-IEEE Press, March 2010.

[2] N. K. Kasabov and Q. Song, "DENFIS: Dynamic Evolving Neural-Fuzzy Inference Systems and Its Application for Time-Series Prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, Abril 2002.

[3] G. Prasad, G. Leng, T. McGinnity, and D. Coyle, *Evolving Intelligent Systems*. John Wiley & Sons, Inc., 2010, ch. Online Identification of Self-Organizing Fuzzy Neural Networks for Modeling Time-Varying Complex Systems, pp. 201–228.

RMSE	Online FIS	Offline FIS	MLP
Stationary	0.98	0.97	0.96
non-stationary	1.43	4.42	7.05

Tabela I. Performance index for a one step ahead forecasting task.

[4] A. Lemos, W. Caminhas, and F. Gomide, "Multivariable gaussian evolving fuzzy modeling system," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 91 –104, feb. 2011.

[5] I. Luna and R. Ballini, "Adaptive fuzzy system to forecast financial time series volatility," *Journal of Intelligent and Fuzzy Systems*, vol. 23, no. 1, pp. 27–38, 01 2012.

[6] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, January/February 1985.

[7] S. Chiu, "A cluster estimation method with extension to fuzzy model identification," in *Proceedings of The IEEE International Conference on Fuzzy Systems*, vol. 2, June 1994, pp. 1240–1245.

[8] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive Mixture of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[9] V. Ramamurti and J. Ghosh, "Structurally adaptive modular networks for non-stationary environments," 1997. [Online]. Available: citeseer.ist.psu.edu/192756.html

[10] S. Haykin, *Kalman Filtering and Neural Networks*. New York, USA: John Wiley & Sons, Inc. , 2001.

[11] A. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 – A strategy for system predictor identification," *Journal of Hydrology*, no. 239, pp. 232–239, 2000.

[12] I. Luna, S. Soares, and R. Ballini, "Partial Mutual Information Criterion For Modelling Time Series Via Neural Networks," *Proceedings of the 11th Information Processing and Management of Uncertainty International Conference*, July 2006.