Evolving Systems (2010) 1:43–58
DOI 10.1007/s12530-010-9005-y

ORIGINAL PAPER

Evolving fuzzy optimally pruned extreme learning machine for regression problems

Federico Montesino Pouzols · Amaury Lendasse

Received: 27 March 2010 / Accepted: 16 June 2010 / Published online: 7 July 2010

Abstract This paper proposes an approach to the identification of evolving fuzzy Takagi–Sugeno systems based on the optimally pruned extreme learning machine (OP-ELM) methodology. First, we describe ELM, a simple yet accurate learning algorithm for training single-hidden layer feed-forward artificial neural networks with random hidden neurons. We then describe the OP-ELM methodology for building ELM models in a robust and simplified manner suitable for evolving approaches. Based on the previously proposed ELM method, and the OP-ELM methodology, we propose an identification method for self-developing or evolving neuro-fuzzy systems applicable to regression problems. This method, evolving fuzzy optimally pruned extreme learning machine (eF-OP-ELM), follows a random projection based approach to extracting evolving fuzzy rulebases. In this approach systems are not only evolving but their structure is defined on the basis of randomly generated fuzzy basis functions. A comparative analysis of eF-OP-ELM is performed over a diverse collection of benchmark datasets against well known evolving neuro-fuzzy methods, namely eTS and DENFIS. Results show that the method proposed yields compact rulebases, is robust and competitive in terms of accuracy.

Keywords Evolving fuzzy systems · Extreme learning machine · Optimally pruned extreme learning machine · Evolving Takagi–Sugeno · Regression · Time series

1 Introduction

Evolving, online or adaptive intelligent systems are meant to be applied on sequential data or streams of data. Evolving systems distinguish themselves from traditional, offline learning methods and previous online or adaptive methods in that their structure (in addition to their parameters) gradually evolves in order to account for new data. In general, any aspect of an evolving system, including the learning mechanism, is subject to self-adaptation (Angelov et al. 2010).

The interest in self-developing artificial neural network methods can be tracked back to some early works in the field (Platt 1991). During the last decade there has been an increase of interest in this field and in particular within the area of evolving fuzzy systems for modeling, classification and control (Angelov et al. 2008, 2010). The need for such systems arises in a variety of contexts and successful applications can be found in areas such as process industry, autonomous systems, intelligent agents, signal processing and bioinformatics, among others (Kasabov 2007; Angelov et al. 2010). Some recent advances in evolving neuro-fuzzy systems include DENFIS (Kasabov and Song 2002), and the more general Evolving Connectionist Systems framework (Kasabov 2007), FLEXFIS (Lughofer 2008), and evolving Takagi–Sugeno (eTS) (Angelov and Filev 2004b) as well as its variants (Angelov and Filev 2005; Angelov et al. 2010).

For instance, evolving TS fuzzy systems (Angelov and Filev 2004b) combine supervised and unsupervised

F. M. Pouzols (✉) · A. Lendasse
Department of Information and Computer Science,
Aalto University School of Science and Technology,
P.O. Box 15400, 00076, Aalto, Espoo, Finland
e-mail: federico.pouzols@tkk.fi

A. Lendasse
e-mail: amaury.lendasse@tkk.fi

learning techniques to identify and evolve the TS model structure as well as its parameters as new data become available. This way, new rules can be added, existing rules can be reorganized, and in general any aspect of an evolving fuzzy inference model is subject to self-development.

Evolving fuzzy systems represent a relative recent step beyond the paradigms of self-tuning neuro-fuzzy systems (Moreno-Velo et al. 2007) and online neuro-fuzzy systems (Angelov and Filev 2004b). In most situations, the evolving approach implies the need for simple, one-pass learning methods as opposed to traditional, iterative algorithms. Evolving fuzzy systems are particularly useful for online prediction and predictive control. Among other advantages, evolving fuzzy systems provide an inherent capability for novelty detection and an enhanced robustness against nonstationarities.

To date, identification methods proposed for evolving fuzzy systems rely on traditional approaches, such as structure-oriented or, more often, cluster-oriented identification (Angelov et al. 2010; Kasabov 2007) or hybrid approaches. The aim of this paper is to define an alternative approach to the identification of evolving fuzzy systems based on the concept of random projections.

Recent results in machine learning and computational intelligence have led to an increase of interest in methods based on random projections (Miche et al. 2010a; Achlioptas 2003; Fradkin and Madigan 2003). These systems have proved to be comparable to traditional approaches for identification purposes (Feng et al. 2009; Miche et al. 2010b; Liang et al. 2006; Miche et al. 2010a). This paradigm seems specially suitable for evolving intelligent systems for data streams, and can provide an alternative approach to identifying evolving fuzzy systems.

This paper proposes a method for the identification of evolving Takagi–Sugeno fuzzy inference systems (FIS) based on the optimally pruned extreme learning machine (OP-ELM) methodology. We leverage in three previous developments: the ELM learning method, introduced by Huang et al. (2006b), the OP-ELM methodology, introduced by Miche et al. (2010b), and the online sequential ELM algorithm, introduced by Liang et al. (2006) and extended for fuzzy systems by Rong et al. (2009).

The (ELM) (Huang et al. 2006a, b) is a simple yet effective learning algorithm for training single-hidden-layer feed-forward artificial neural networks (SLFNs) with random hidden nodes. ELM challenges conventional learning methods and theories. ELM has been shown to be accurate and fast both theoretically and experimentally. Indeed, ELM is a remarkably fast method but can achieve a performance in terms of generalization comparable to other accurate yet costly learning techniques.

OP-ELM has been introduced by Miche et al. (2010b) in order to improve the robustness of ELM models. This is achieved through a three stages methodology which includes steps for fast ranking of hidden neurons as well as model selection.

Rong et al. (2009) have previously shown that SLFNs can be regarded as equivalent to FIS. They use this equivalence in order to derive an online sequential method for fuzzy systems (OS-Fuzzy-ELM) based on the online sequential ELM (Liang et al. 2006; Rong et al. 2009). Here we introduce an evolving approach to the identification of evolving Takagi–Sugeno fuzzy inference systems based on the original offline OP-ELM methodology. We exploit the equivalence between SLFNs and FIS, and bring the good performance and robustness of the OP-ELM methodology together with an online approach in order to define the evolving fuzzy OP-ELM (eF-OP-ELM). This method follows a random projection based approach to extracting evolving fuzzy rulebases. In this approach systems are not only evolving but their structure is defined in terms of randomly generated fuzzy basis functions.

When compared with previous ELM methods, the contribution of the method proposed in this paper, eF-OP-ELM is twofold. First, as opposed to the original proposal of OP-ELM which is an offline method, eF-OP-ELM addresses online learning. Second, as opposed to OS-Fuzzy-ELM, eF-OP-ELM is a fully evolving fuzzy method where both the structure and all the parameters of the model evolve in an online manner. This way, we bring into the field of evolving intelligent systems the alternative identification approach of ELM as well as the robustness of the OP-ELM methodology.

The paper is organized as follows. Section 2 describes the ELM, the OP-ELM methodology and the OS-ELM sequential learning algorithm. In Sect. 3 we introduce the evolving fuzzy OP-ELM (eF-OP-ELM) modeling approach for building evolving fuzzy inference systems. Then, in Sect. 4 a comparative analysis is performed against other well-known alternatives. Section 5 discusses results. Finally we give some concluding remarks. Appendix describes an incremental initialization procedure for eF-OP-ELM.

2 Extreme learning machine

The extreme learning machine (ELM) (Huang et al. 2006a, b) is a simple yet effective learning algorithm for training SLFNs with random hidden nodes. In ELM, the hidden neuron parameters are randomly assigned whereas the output weights are analytically determined. ELM is a unified framework of generalized SLFNs that has the universal approximation capability for a wide range of hidden node types (see Huang et al. 2006a for a proof and discussion of the topic). The training process for ELM can be several orders of magnitude faster than traditional learning algorithms for feed-forward neural networks, while attaining comparable or even better approximation and generalization capabilities.

Let us consider a dataset consisting of M observations $(\mathbf{x}_j, \mathbf{y}_j) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, with d_1 the dimension of the input space and d_2 the dimension of the output space. An SLFN with N neurons in the hidden layer is defined by the following expression:

$$\sum_{i=1}^{N} \beta_i f(\mathbf{x}_j, \mathbf{c}_i, a_i), \quad 1 \leq j \leq M,$$

where $f(\cdot)$ is the activation function and $\beta_i \in \mathbb{R}$ are the output weights. Let us illustrate two widely studied architectures of SLFNs: SFLNs with hidden additive nodes and radial basis function (RBF) networks which use RBF nodes in the hidden layer. Additive nodes have the form $f(\mathbf{c}_i \cdot \mathbf{x}_j + a_i)$, where $\mathbf{c}_i \in \mathbb{R}^{d_1}$ are the input weights and a_i the biases. RBFs have the form $f(||\frac{\mathbf{x}_j - \mathbf{c}_i}{a_i}||)$, where $\mathbf{c}_i \in \mathbb{R}^{d_1}$ are the centers and a_i the spreads or radii of the RBF nodes.

If the SFLN perfectly fits the data, then the difference between the estimated outputs $\hat{\mathbf{y}}_j$ and the actual output values is zero and thus the following holds:

$$\sum_{i=1}^{N} \boldsymbol{\beta}_i f(\mathbf{x}_j, \mathbf{c}_i, a_i) = \mathbf{y}_j, \quad 1 \leq j \leq M,$$

which can be written as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}, \tag{1}$$

with

$$\mathbf{H} = \begin{bmatrix} f(\mathbf{x}_1, \mathbf{c}_1, a_1) & & f(\mathbf{x}_1, \mathbf{c}_N, a_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{x}_M, \mathbf{c}_1, a_1) & & f(\mathbf{x}_M, \mathbf{c}_N, a_N) \end{bmatrix},$$

$\boldsymbol{\beta} = (\beta_1^T, \ldots, \beta_N^T)$ and $\mathbf{Y} = (\mathbf{y}_1^T, \ldots, \mathbf{y}_M^T)^T$. In the ELM method, the hidden layer output matrix, $\mathbf{H}$, is generated in a random way, independently of the training dataset. Then, the output weights of the SLFN, $\boldsymbol{\beta}$ can be determined analytically. The estimated output weights are computed as:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{Y}, \tag{2}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose pseudoinverse (or natural inverse) of $\mathbf{H}$. For the applicable implementation methods, usually based on singular value decomposition, refer to (Liang et al. 2006; Huang et al. 2006b). As pointed out by Huang (2008) the ELM theory claims that parameter tuning is not required. This approach challenges conventional learning methods and theories.

It has been theoretically shown that for function approximation all the parameters of the hidden nodes can be randomly generated without any prior knowledge. In (Huang et al. 2006b) it is proved that the hidden layer output matrix can be computed and achieves an approximation error as small as desired for $N \leq M$, under the assumption that the activation function is infinitely differentiable. The universal approximation capability of ELM has been proved in (Huang et al. 2006a).

This way, ELM is a extremely fast method (Huang et al. 2006b), several orders of magnitude faster than traditional feedforward neural networks methods while competitive in terms of accuracy. As a final remark, for a given type of activation function, ELM only requires the number of neurons in the hidden layer to be specified, as opposed to other learning methods which commonly have several hyperparameters to be tuned.

2.1 Optimally Pruned ELM

The optimally pruned extreme learning machine (OP-ELM) (Miche et al. 2008, 2010b; Sorjamaa et al. 2008) is a methodology based on the ELM. OP-ELM models are built in three stages and use Gaussian, sigmoid and linear kernels in general. First, an ELM is constructed, then, an exact ranking of the neurons in the hidden layer is performed, and finally the decision on how many neurons are pruned is made based on an exact leave-one-out error estimation method (Miche et al. 2010b; Myers 2000), as described in the next subsections. These stages are performed by means of fast methods and lead to extremely fast yet accurate models.

ELM OP-ELM has been shown to provide a compromise between the speed of ELM and the accuracy and robustness of other much more computationally intensive methods. OP-ELM models achieve roughly the same level of accuracy as that of other well known computational intelligence methods (Miche et al. 2008, 2010b), such as Support Vector Machines (Schölkopf and Smola 2002) and Least Squares Support Vector Machines (Suykens et al. 2002), Gaussian Processes and Multilayer Perceptrons (Haykin 1998), while being significantly faster.

As explained above, for a given type of activation function, only one parameter has to be tuned in order to build accurate ELM models: the number of hidden neurons. In principle the only feasible approaches to a sensible tuning of the number of hidden neurons are based on the definition of validation subsets. This is the approach used generally in the literature (Huang et al. 2006b; Liang et al. 2006).

However, validation approaches such as cross-validation and bootstrapping methods raise several issues. In particular, computational cost increases significantly, which is specially troublesome for online, adaptive and possibly real-time systems. In addition, validation methods assume the different subsets used for training, validation and test are drawn from the same population. For systems that evolve or exhibit nonstationarity, whether statistical or dynamical, this assumption may lead to wrong models.

The OP-ELM method introduced a sound approach to the selection of the subset of best nodes in such a way that less relevant neurons are pruned. This brings in a fundamental advantage for evolving methods besides enhancing the robustness of ELM models against irrelevant and redundant variables (Miche et al. 2010b). In what follows we outline the three stages of the OP-ELM methodology.

2.1.1 *Construction of an initial SLFN*

This step is performed using the standard ELM algorithm for a large enough number of neurons N.[1] While the original ELM proposal uses sigmoid kernels, and ELM models are usually defined on the basis of a single type of activation function or kernel, in the OP-ELM methodology three types of kernels are used in combination for better robustness and generality. The following types are used: Sigmoid, Gaussian and linear. Refer to (Miche et al. 2008, 2010b) for the details about the procedure followed to randomly initialize the parameters of these types of functions.

2.1.2 *Ranking of hidden neurons*

As second step in the OP-ELM methodology, the multiresponse sparse regression (MRSR) algorithm (Similä and Tikka 2005) is applied in order to rank the hidden neurons according to their accuracy. MRSR is in essence a generalization of the well-known least angle regression (LARS) algorithm (Efron et al. 2004), and is thus able to find an exact ranking for linear problems. Since in a ELM model the output is linear with respect to the randomly initialized hidden nodes, the MRSR ranking within the OP-ELM methodology is exact.

The MRSR algorithm is defined as follows. For a $M \times N$ regressor matrix $\mathbf{X}$, each column is added one by one to the model in successive steps of the algorithm. This way, for step k, the model is defined as $\hat{\mathbf{Y}}^k = \mathbf{X}\mathbf{W}^k$, where $\hat{\mathbf{Y}}^k$ is the model output. The weight matrix, $\mathbf{W}^k$ has k nonzero rows at the kth step of the MRSR algorithm. For the full details refer to (Similä and Tikka 2005). It should be noted that, similarly to LARS, MRSR is a variable ranking rather than selection algorithm.

This way, the variables for the MRSR algorithm within the OP-ELM methodology, $\mathbf{h}_i$ (the outputs of the hidden nodes or columns of the hidden layer output matrix, $\mathbf{H}$), are ranked exactly by their performance.

2.1.3 *Model selection*

Once a ranking of the kernel has been obtained the best number of neurons for the model has to be chosen. Methods based on offline validation, such as leave-one-out are often used for this kind of task. They can be however extremely expensive in computational terms and quickly become unaffordable for large datasets or online, evolving or time constrained systems.

However, the LOO can be directly calculated for linear models by using the PRESS (PREdiction Sum of Squares) statistics (Myers 2000; Birattari et al. 1999), which provides a closed-form expression for the LOO error of linear models. For OP-ELM models it is as follows:

$$\varepsilon_i^{PRESS} = \frac{y_i - \mathbf{h}_i \mathbf{b}_i}{1 - \mathbf{h}_i \mathbf{P} \mathbf{h}_i^T},$$

where i denotes the ith hidden node, $\mathbf{h}_i$ are the columns of the ranked hidden layer output matrix ($\mathbf{H}$), $\mathbf{b}_i$ are the output weights of the SLFN, and $\mathbf{P}$ is defined as $\mathbf{P} = (\mathbf{H}^T\mathbf{H})^{-1}$.

The optimal number of neurons can be found by estimating the LOO error for different numbers of nodes (already ranked by accuracy) and selecting the number of neurons L such that minimizes the error:

$$L = \underset{j \in \{1,\ldots,N\}}{\operatorname{argmin}} \sum_{i=1}^{j} \varepsilon_i^{PRESS} \tag{3}$$

It has been shown that the ranking (previous) stage of the OP-ELM method has two positive effects: convergence is faster and the number of neurons required to achieve the lowest LOO error is lower (Miche et al. 2010b).

2.2 Online sequential ELM

The original ELM method is designed for offline modeling. However thanks to the simplicity of the computations in the ELM method, it is possible to define efficient online extensions for ELM. Liang et al. (2006) have proposed the OS-ELM algorithm which we outline in this section and will be used as a reference method for experiments in the next sections.

The algorithm first computes a standard ELM model for an initialization training set, with output matrix $\mathbf{Y}_0$, hidden nodes matrix $\mathbf{H}_0$, and solution $\boldsymbol{\beta}^{(0)} = (\mathbf{H}_0^T\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{Y}_0$, using the Moore–Penrose generalized inversion according to (2). Let us define $\mathbf{K}_0 = \mathbf{H}_0^T\mathbf{H}_0$. Then, for each new observation or chunk of observations the model is updated online efficiently by means of the following recursive expressions:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T(\mathbf{Y}_{k+1} - \mathbf{H}_{k+1}\boldsymbol{\beta}^{(k)}), \tag{4}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k\mathbf{H}_{k+1}^T(\mathbf{I} + \mathbf{H}_{k+1}\mathbf{P}_k\mathbf{H}_{k+1}^T)^{-1}\mathbf{H}_{k+1}\mathbf{P}_k, \tag{5}$$

[1] By default the initial number of neurons used in the OP-ELM Toolbox (Lendasse et al. 2010; Miche et al. 2008) is 100.

where $\mathbf{P}_{k+1} = \mathbf{K}_{k+1}^{-1}$. These recursive update rules are obtained by using the Sherman–Morrison–Woodbury formula for computing the inverse of a rank-k correction of matrices (Higham 2002). Refer to (Liang et al. 2006) for the full details.

As a special case, when the online update of the model is done by single data samples, $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$, as opposed to data chunks, Eqs. 4 and 5 take the following simplified forms, respectively:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \mathbf{P}_{k+1}\mathbf{h}_{k+1}\left(\mathbf{y}_{k+1}^T - \mathbf{h}_{k+1}^T\boldsymbol{\beta}^{(k)}\right),$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k\mathbf{h}_{k+1}^T\mathbf{P}_k}{1 + \mathbf{h}_{k+1}^T\mathbf{P}_k\mathbf{h}_{k+1}},$$

where $\mathbf{h}_{k+1} = [f(\mathbf{x}_{k+1}, \mathbf{c}_1, a_1)\ldots f(\mathbf{x}_{k+1}, \mathbf{c}_N, a_N)]$.

3 Evolving fuzzy OP-ELM

The rules, R_i, of a Takagi–Sugeno (TS) fuzzy inference model, applied to a certain input $\mathbf{x}_j$, can be generally expressed as (Angelov and Filev 2004b):

$$R^i : \text{IF}(x_{j1} \text{ is } A_{i1}) \text{ AND}\ldots\text{AND } (x_{jd_1} \text{is } A_{id_1}),$$
$$\text{THEN } (y_{j1} \text{ is } \beta_{i1})\ldots(y_{jd_2} \text{ is } \beta_{id_2}),$$

where d_1 is the dimension of the input space, d_2 is the dimension of the output space, $i = 1, ..., L$ for a rulebase consisting of L rules, and A_{ik}, ($k = 1, ..., d_1$; $i = 1, ..., L$) are the fuzzy sets for the kth input variable, x_{jk}, in the ith rule. β_{ik} ($k = 1, ..., d_2$; $i = 1, ..., L$) are crisp values, linear combinations of the input variables in the form $\beta_{ik} = q_{ik,0} + q_{ik,1}x_{j1} + \cdots + q_{ik,d_1}x_{jd_1}$ for a first-order TS model.

For each fuzzy set, A_{ik}, the degree of membership of a given input x_{jk} is specified by its corresponding membership function $\mu_{A_{ik}}(x_{jk})$. A nonconstant piecewise continuous membership function $f(\mathbf{c}, a)$ can be considered as in (Rong et al. 2009). This kind of function includes most common membership functions such as Gaussian and triangular as well as virtually all practical possibilities. The membership function can thus be defined by any bounded nonconstant piecewise continuous membership function as follows:

$$\mu_{A_{ik}}(x_{jk}; c_{ik}, a_i) = f(x_{jk}; c_{ik}, a_i),$$

where a_i and c_{ik} are the parameters of the membership function $f(\cdot)$ for the ith rule and the kth component of the input vectors $\mathbf{x}_j$, x_{jk} with $k = 1, ..., d_1$.

In a fuzzy inference system of this type, the output of the model is computed as the weighted sum of the output of each rule, where the weights are the activation degrees of the rules. Thus, the system output $\hat{\mathbf{y}}_j$ for an input $\mathbf{x}_j$ is given as follows:

$$\hat{\mathbf{y}}_j = \frac{\sum_{i=1}^{L} \boldsymbol{\beta}_i R_i(\mathbf{x}_j; \mathbf{c}_i, a_i)}{\sum_{i=1}^{L} R_i(\mathbf{x}_j; \mathbf{c}_i, a_i)} = \sum_{i=1}^{L} \boldsymbol{\beta}_i F(\mathbf{x}_j; , \mathbf{c}_i, a_i), \qquad (6)$$

where $R_i(\cdot)$ denotes the activation degree of the ith rule for input $\mathbf{x}_j, \boldsymbol{\beta}_i = (\beta_{i1}, \ldots, \beta_{im})$, and $F(\cdot)$ can be seen as a normalized rule:

$$F(\mathbf{x}_j; \mathbf{c}_i, a_i) = \frac{R_i(\mathbf{x}_j; \mathbf{c}_i, a_i)}{\sum_{i=1}^{L} R_i(\mathbf{x}_j; \mathbf{c}_i, a_i)}, \qquad (7)$$

with $R_i(\mathbf{x}_j, \mathbf{c}_i, a_i) = \mu_{A_{i1}}(x_{j1}; c_{i1}, a_i)$ AND ... AND $\mu_{A_{id_1}}$ $(x_{jd_1}; c_{id_1}, a_i)$.

According to the equivalence between generalized SFLN and FIS, (Rong et al. 2009) have established the interpretation of OS-ELM as an online fuzzy model applicable to both regression and classification problems. As noted in (Rong et al. 2009) F as expressed in (7) is what can be called a fuzzy basis function (FBF) (Zeng and Singh, 1995).

In (6) it is evident that a FIS is equivalent to a generalized SLFN, where the $F(\cdot)$ represent the output functions of the hidden layer and the $\boldsymbol{\beta}_i$ represent the output weight vector. This way, the output functions of the hidden nodes of the SLFN are equivalent to the FBFs of the FIS, which in turn are based on the membership functions.

This equivalence is used in (Rong et al. 2009) to develop an online method for the identification of fuzzy inference systems of the Takagi–Sugeno type (OS-Fuzzy-ELM). Note though that this method is online [as OS-ELM (Liang et al. 2006)] yet not fully evolving, i.e., the system self-tunes in an online manner but the system structure (rulebase) is set in an initialization phase.

As a particular case of fuzzy or neuro-fuzzy system, an evolving TS fuzzy system can be represented as a neural network (Angelov and Filev 2004a). We show in what follows how to extend for an online learning approach the following two elements in the OP-ELM methodology: the LARS-based ranking process, and the PRESS statistics-based estimation of the LOO error.

Following the equivalence between the structure of SFLN and FIS, the latter can be expressed in terms of the former:

$$\mathcal{F}(\mathbf{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i F(\mathbf{x}_j; \mathbf{c}_i, a_i) = \mathbf{y}_j, \qquad (8)$$

for a certain number of fuzzy rules L. In the TS type models the consequent of each fuzzy rule is a linear equation of the input variables. If the coefficients, $q_{ik,j}$, $k = 1, \ldots, d_2$, $j = 1, \ldots, d_1$, of the linear equations are arranged in a matrix of parameters of the model for the ith rule as follows:

$$\mathbf{q_i} = \begin{bmatrix} q_{i1,0} & \cdots & q_{id_2,0} \\ \vdots & \ddots & \vdots \\ q_{i1,d_1} & \cdots & q_{id_2,d_1} \end{bmatrix},$$

for $i = 1, ..., L,$ then $\boldsymbol{\beta}_i = \mathbf{x}_{je}^T \mathbf{q}_i$, where $\mathbf{x}_{je} = [1, \mathbf{x}_j^T]^T$ is obtained by appending a 1 to the input vector in order to generate a linear equation. This way, expanding $\boldsymbol{\beta}_i$ in (8), the output of the model can be expressed as follows:

$$\mathcal{F}(\mathbf{x}_j) = \sum_{i=1}^{L} \mathbf{x}_{je}^T \mathbf{q}_i F(\mathbf{x}_j; \mathbf{c}_i, a_i) = \mathbf{y}_j, \quad j = 1, \ldots, M.$$

The above expression in compact form is:

$$\mathbf{HQ} = \mathbf{Y},$$

which is a generalization of (1), with

$$\begin{aligned} \mathbf{H} &= \mathbf{H}(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{c}_1, \ldots, \mathbf{c}_L, a_1 \ldots, a_L) \\ &= [x_{je}^T F(\mathbf{x}_j; \mathbf{c}_1, a_1), \ldots, x_{je}^T F(\mathbf{x}_j; \mathbf{c}_L, a_L)] \end{aligned}$$

redefined as the hidden layer output matrix (or fuzzy basis function output matrix) weighted by the extended input vectors, and with Q being the output parameter matrix of the TS model:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q_1} \\ \vdots \\ \mathbf{q_L} \end{bmatrix}, \text{ with } \mathbf{q}_i = \begin{bmatrix} q_{i1,0} & \cdots & q_{id_2,0} \\ \vdots & \ddots & \vdots \\ q_{i1,d_1} & \cdots & q_{id_2,d_1} \end{bmatrix}.$$

Since a TS FIS is equivalent to an SLFN, the ELM learning method can be applied to a fuzzy system. Furthermore, given that $\mathbf{H}$ is initialized randomly and $\mathbf{Y}$ is known, $\mathbf{Q}$ can be computed online using the same approach as in ELM.

In the evolving fuzzy OP-ELM (eF-OP-ELM) proposed here, the structure of the rulebase can change as new observations become available. Hence, the output of the TS model can be expressed as follows:

$$\mathcal{F}(j, \mathbf{x}_j) = \sum_{i=1}^{L(j)} \mathbf{x}_{ie}^T \mathbf{q}_i(j) f(\mathbf{x}_j, \mathbf{c}_i(j), a_i(j)) = \mathbf{y}_j, \tag{9}$$

where both the parameters of the fuzzy basis functions (or rule antecedents) and the consequent parameters evolve. In compact form an eF-OP-ELM model can be expressed as: $\mathbf{H}(j)\mathbf{Q}(j) = \mathbf{Y}(j), \quad j \geq 1$

for $\mathbf{x}_j$ being the last available observation of inputs. Here, $\mathbf{H}(j)$ is an evolving matrix, i.e., $L(j)$ (the number of rules or nodes in the hidden layer of the equivalent ELM) can evolve as well as the input membership functions [through parameters $\mathbf{c}_i(j)$ and $a_i(j)$] and the consequent parameters, $\mathbf{q}_i(j)$. In this method the rulebase is fully evolving and thus the hidden layer output matrix takes the following form:

$$\mathbf{H}(j) = \begin{bmatrix} F(\mathbf{x}_k, \mathbf{c}_1(j), a_1(j)) & \ldots & F(\mathbf{x}_k, \mathbf{c}_{L(j)}(j), a_L(j)) \\ \vdots & \ddots & \vdots \\ F(\mathbf{x}_j, \mathbf{c}_1(j), a_1(j)) & \ldots & F(\mathbf{x}_j, \mathbf{c}_{L(j)}(j), a_L(j)) \end{bmatrix},$$

where $1 \leq k \leq j$ defines a sliding window on the incoming data stream, $L(j)$ is the evolving number of rules, $\mathbf{c}_i(j) \in C$ and $a_i(j) \in A$ are the parameters of the input membership functions, and C and A are sets of parameters values generated randomly. Here, we consider 0-order TS models for simplicity, and thus the $\mathbf{x}_{ie}$ in (9) verify $\mathbf{x}_{ie} = 1$.

Note that the hidden layer output matrix can be seen as a matrix of outputs of fuzzy basis functions. Putting all pieces together, we can describe how the components of an evolving TS fuzzy inference system are defined in the eF-OP-ELM method:

- The antecedents ("*IF*" part) belong to a set of antecedents that is created randomly by generating the sets of parameters C and A at the initialization stage.
- The number of fuzzy rules and concrete subset of rules are selected as in the 2nd and 3rd stages of the OP-ELM methodology.
- The corresponding consequents ("*THEN*" part) are generated analytically using the ELM method, according to (2).

The eF-OP-ELM algorithm consists of the following steps:

Algorithm 1 Evolving Fuzzy OP-ELM (eF-OP-ELM)

```
1: Create an ordered set of antecedents by generating the sets of parameters $C$ and $A$ (of cardinality $N$).
2: Generate an $M$-by-$N$ matrix, $\mathbf{H}_0(j), j = M$, by applying the $N$ fuzzy basis functions as in (7), for a given initialization sequence consisting of $M$ observations $(\mathbf{x}_j, \mathbf{y}_j) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$.
3: Rank the fuzzy basis function outputs in $\mathbf{H}_0(j)$ using the MRSR algorithm, and select the best number of rules (as in (3)). The result, $\mathbf{H}(j), j = M$, allows for the computation of $\mathbf{Q}(j), j = M$, (as in (2)), and the initial evolving TS model $\mathcal{F}(\cdot)$, with the form of (9).
4: For each a new input-output observation, $(\mathbf{x}_j, \mathbf{y}_j)$ with $j \geq M$, do:
  4.1: Update $\mathbf{H}_0(j)$ with the new observation, rank the fuzzy basis function outputs in $\mathbf{H}_0(j)$ and select the best number of rules in order to generate $\mathbf{H}(j)$.
  4.2: Find the analytical solution for $\mathbf{Q}(j)$ (as in (2)) and compute the model output, $\hat{y}_j + 1$.
```

Note that an initialization stage is considered here for simplicity, as the initialization sequence is small enough so as to be negligible in many applications. An incremental learning algorithm for the initial observations can be found in Appendix. This incremental algorithm can be seamlessly integrated into the basic algorithm above, where it would replace the first three steps.

This way, the fuzzy rule antecedents are ranked by the MRSR algorithm (described in Sect. 2.1.2), whereas the optimum number of rules is selected according to (3), following the same scheme as the OP-ELM methodology.

In the implementation that will be used in the next section, C and A are sets of random centers and radii, respectively, for Gaussian functions, and are generated following an scheme similar to that of the offline OP-ELM method (Miche et al. 2008, 2010b). More specifically, the sets C and A are initially generated randomly with a uniform distribution. The values for the centers $\mathbf{c}_i \in C$ are randomly selected from the observed data points, similarly as in (Poggio and Girosi, 1989). The radii, $a_i \in A$ are randomly drawn in the range between percentiles 20% and 80% of the distance distribution of the input space, as proposed in (Schölkopf and Smola, 2002) and assessed for OP-ELM models in (Miche et al. 2010b).

To accommodate for new data, the following update procedure is applied on the sets of antecedent parameters. For each new observation, $\mathbf{x}_j$, a new center and radius are generated. The center is defined by the new observation, whereas the radius is generated randomly within the updated input distance distribution, according to the scheme described above. The corresponding new fuzzy basis function replaces the worst ranked basis function in $H_0(j)$.

The rulebases of eF-OP-ELM models are evolved in step 4.1 by adding new rows and possibly removing old rows, generating $\mathbf{H}_0(j)$, and then ranking and selecting the best number of rules in order to compute $\mathbf{H}(j)$. In the initial matrix, $\mathbf{H}_0(j)$, which can be seen as a pool of potentially useful fuzzy basis functions, a maximum set of rules is defined, then the best subset of rules is dynamically selected. The number of columns of $\mathbf{H}_0(j)$ is given by the cardinality of the sets of parameters C and A, N. In practice, N does not need to be high. In the next section $N = 50$ is used for the reported experiments.

Thus, the size of the initialization sequence can be set to a small value, $M \geq N$. In this paper, $M = N = 50$. Note that the cardinality of the parameters set, N, imposes a ceiling on the maximum number of fuzzy rules that can be identified in practice. In addition, ranking and selection of rules can be effectively performed with a reduced subset of recent values. In this paper, the maximum length of the sequence of inputs used for the ranking stage, i.e., the maximum number of rows of the $\mathbf{H}_0(j)$ and $\mathbf{H}(j)$ matrices is 500, and thus $j - k + 1 \leq 500$. The product T-norm is used for conjunction operations as well as product inference of rules. The fuzzy mean method is used for defuzzification.

4 Experimental results

For our experiments, two well known methods in the field of evolving fuzzy systems are taken as reference: DENFIS and eTS. DENFIS (Dynamic evolving neural-fuzzy inference system) (Kasabov and Song 2002)). is one implementation of the more general ECOS (Evolving Connectionist Systems) framework (Kasabov 2007). eTS (Angelov and Filev 2004b) was applied using global parameter estimation with a recursive least squares filter (RLS), and the default parameters of the implementation used, detailed later on in this section. For both DENFIS and eTS, first-order TS systems were built. In addition, OS-Fuzzy-ELM is considered as a reference. Note though that while it is an online method it is not fully evolving, as detailed in previous sections.

We compare these methods against eF-OP-ELM, the method proposed in this paper which is implemented as described in previous sections, with a maximum number of rules of 50. Even though the standard OP-ELM uses three different kinds of kernels, here we only use kernels of Gaussian type for simplicity's sake.

For OS-Fuzzy-ELM the same validation approach originally proposed in Rong et al. (2009) is used. The initial training data is randomly split into two nonoverlapping subsets for training (75%) and validation (25%). The optimal number of rules is selected such that the validation error is minimized. Different numbers of rules are evaluated, with the number increasing by 1 in the range [1, 100]. Within this range and for each case, the average cross-validation error for 25 trials is computed. The block size is set to 1. Finally, the OS-Fuzzy-ELM with the lowest average cross-validation error is selected. We should note however that this approach may require a high computational effort in the context of evolving systems. In general, the length of the initialization training subsequence is set to (50 + number of rules).

The main characteristics of the datasets used are shown in Table 1. The datasets were chosen in order to find a compromise between the following objectives: (a) easing comparison with the related literature, and (b) selecting datasets for a broad range of characteristics (variables, size, dynamical behavior, etc.). In particular, some datasets represent clearly nonstationary processes, while some others lie in the domain of regression problems where nonstationarity is not relevant in all cases.

Table 1 Datasets: number of inputs, training observations and test observations

Dataset	# Inputs	Training length	Test length
Abalone	8	2,784	1,393
Auto-MPG	7	258	134
Bank	8	3,000	1,500
Boston Housing	13	337	317
Breast Cancer	32	129	65
California Housing	8	13,760	6,880
Delta Ailerons	5	4,752	2,377
Delta Elevators	6	6,344	3,173
Servo	4	111	56
Stocks	9	633	317
Darwin SLP	5	904	467
ENSO	3	465	400
GSTA	4	1,028	520
Internet2	4	708	730
Mackey-Glass	4	500	–
NAO	7	1,146	572
Santa Fe Laser	3	988	9,093
Sunspots	7	2,085	1,000
Tree Rings	8	1,013	511

The first ten datasets are machine learning benchmarks for regression problems while the last nine datasets correspond to time series

The first ten datasets are well known regression problems in the field of machine learning. These datasets are benchmarks from the UCI Machine Learning Repository (Asuncion and Newman 2010) or the StatLib repository (StatLib 2010), and can be also found online from http://www.liaad.up.pt/~ltorgo/Regression/DataSets.htm. These are included in order to analyze general regression problems and ease comparison with the literature on related ELM based methods (Miche et al. 2010b; Rong et al. 2009; Liang et al. 2006).

The last nine datasets correspond to diverse time series applications. The Darwin SLP time series consists of monthly values of the Darwin Sea sea level pressure (SLP) for the years 1882–1998. The dataset can be found at the Time Series Data Library (Hyndman 2010). The SLP for the next month, $y(t+1)$ has to be predicted using five known values from the past, $y(t-11)$, $y(t-6)$, $y(t-3)$, $y(t-2)$, and $y(t-1)$. The ENSO series is the data set from the ESTSP 2007 time series prediction competition (ESTSP07 2010). This dataset consists of 875 samples of temperatures of the El Niño-Southern Oscillation phenomenon. $y(t+1)$ has to be predicted using $y(t)$, $y(t-2)$, and $y(t-7)$ as inputs.

The global surface temperature anomaly (GSTA) time series consists of monthly global (land and ocean combined) temperature anomaly indexes, as provided by the National Climatic Data Center of the National Oceanic and Atmospheric Administration, available online at http://www.ncdc.noaa.gov/oa/climate/research/anomalies. This data set is generated from anomalies on a 5° × 5° grid across Earth's land and ocean surfaces. Temperature anomalies are defined as deviations from the average on the 1901–2000 period. The time series consists of 1560 values, spanning from January 1880 through December 2009. In this case the problem is defined as that of predicting $y(t+1)$ (anomaly for next month) as a function of $y(t)$, $y(t-1)$, $y(t-2)$ and $y(t-6)$. As an example, the GSTA series is depicted in Fig. 1.

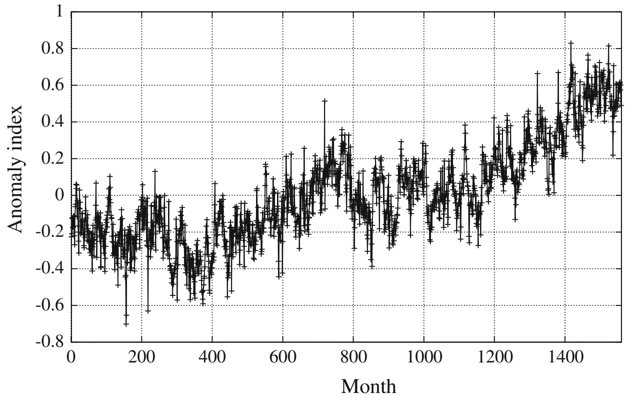

Fig. 1 GSTA time series: 1,560 monthly temperature anomaly values

The Internet2 time series represents the total amount of aggregated incoming traffic in the routers of the Abilene network, the Internet2 backbone. The series consists of 1,458 daily averages from the 4 January 2003 through the 31 December 2006. The data are available from the Abilene Observatory (Internet2Observatory 2008). $y(t+7)$ (next week) has to be predicted using $y(t)$, $y(t-2)$, $y(t-4)$, and $y(t-11)$.

Mackey–Glass is a well-known example of chaotic system (Mackey and Glass 1977) that can describe a complex physiological process. Here we included a synthetic instance for comparison purposes (Angelov and Filev 2004b; Kasabov and Song 2002). The dataset is generated using the 4th order Runge-Kutta method with time step 0.1 s for the following differential equation:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t).$$

The initial conditions and delay parameter are $x(0) = 1.2, x(t) = 0$ for $t<0, \tau = 17$, and the value of the series 85 steps ahead ($x(t+85)$) has to be modeled based on 4 inputs: $x(t)$, $x(t-6)$, $x(t-12)$ and $x(t-18)$. Refer to (Kasabov and Song 2002) for the full details required to generate the dataset. In order to replicate the subseries of 500 values used in (Angelov and Filev 2004b), the values selected for prediction (shown in Fig. 2) lie in the range $x(443.2)$, ..., $x(493.1)$.

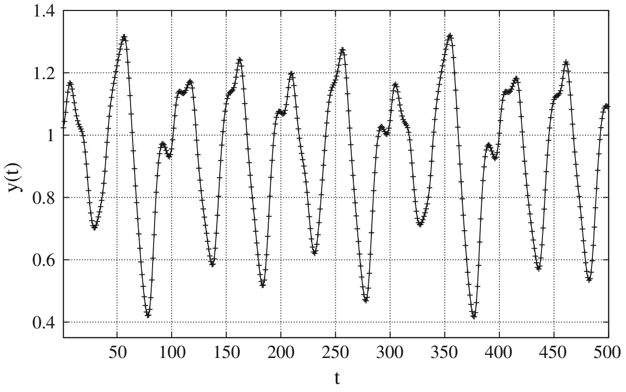

Fig. 2 Mackey–Glass time series: 500 samples to be predicted

The NAO time series corresponds to the North Atlantic Oscillation index (Hurrell and Deser 2009) as provided by the Climate Analysis Section, NCAR, Boulder, available on line at http://www.cgd.ucar.edu/cas/jhurrell/indices.html. The series describes the temporal evolution of the NAO phenomenon, which plays a key role in the dynamics of the Northern Hemisphere climate. Monthly indices are defined from January 1865 through August 2009. The next month's index, $y(t + 1)$ has to be predicted based on seven inputs representative of previous seasons within a year and a half: $y(t)$, $y(t - 1)$, $y(t - 3)$, $y(t - 5)$, $y(t - 7)$, $y(t - 12)$, and $y(t - 14)$.

The Santa Fe Laser dataset of the Santa Fe time series competition (SantaFeLaser 2010; Weigend and Gershenfeld 1994). represents the intensity of a far-infrared-laser in a chaotic state, measured in a physics laboratory experiment. The series is a cross-cut through periodic to chaotic pulsations of the laser, and can be closely modeled analytically (Weigend and Gershenfeld 1994). This series is a remarkable example of noise-free complicated behavior in a clean, stationary, low-dimensional physical system for which the underlying dynamics is well understood. In this case, the next value, $y(t + 1)$ has to be modeled based on three inputs: $y(t)$, $y(t - 1)$, $y(t - 2)$ and $y(t - 12)$. This subset of inputs is optimal for a maximum regressor size of 12 (Montesino Pouzols et al. 2010).

We also analyzed the series of monthly averaged sunspot numbers covering from January 1749 through December 2007, as provided by the National Geographical Data Center from the US National Oceanic and Atmospheric Administration.[2] Given the yearly periodicity of the series, a maximum regressor size of 12 was defined. $y(t + 12)$ (next year) has to be predicted using $y(t)$, $y(t - 1)$, $y(t - 2)$, $y(t - 3)$, $y(t - 4)$, $y(t - 8)$ and $y(t - 10)$.

Finally, the series of tree rings contains yearly measures of tree rings widths in dimensionless units. This series was measured in Argentina for the 441–1974 period and corresponds to the *arge030* dataset of the Time Series Data Library (Hyndman 2010). In this case, $y(t + 1)$ (width for the next year) has to be predicted using the 10 previous values $y(t - 9)$, ..., $y(t)$, except $y(t - 4)$ and $y(t - 6)$. Further descriptions of these datasets and other details to reproduce results for the time series datasets can be found in the literature (Montesino Pouzols et al. 2008a, b, 2010).

In Table 1 training and tests subsets are distinguished. Note that the training set is in fact a sequence and is defined as the sequence of values beginning at the first observation, while the test set is defined as the sequence of last observations. This way, training and out-of-sample or test errors for offline methods can be analyzed while accounting for the evolution in time of the datasets and its effects on models.

Training and test errors for offline modeling for OP-ELM and DENFIS (in offline mode) are listed in Tables 2 and 3, for the regression and time series benchmarks, respectively. The tables also show the number of hidden nodes or fuzzy rules identified as well as the time required.

Table 2 Offline modeling errors for training and test subsets

Dataset	Method	Training NDEI	Test NDEI	Rules	Time (s)
Abalone	DENFIS	3.21e+00	3.89e+00	20	9.20e+00
	OP-ELM	6.33e−01	6.91e−01	58	9.83e+00
Auto-MPG	DENFIS	1.44e+00	1.97e+00	31	1.24e+00
	OP-ELM	2.90e−01	6.30e−01	32	9.00e−01
Bank	DENFIS	1.04e+00	1.09e+00	730	1.48e+02
	OP-ELM	2.09e−01	2.18e−01	103	1.34e+01
Boston Housing	DENFIS	1.97e+00	2.47e+00	59	2.26e+00
	OP-ELM	2.13e−01	6.62e+00	88	2.45e+00
Breast Cancer	DENFIS	3.32e−01	3.93e+00	40	2.27e+00
	OP-ELM	7.61e−01	2.17e+00	16	1.94e+00
California Housing	DENFIS	5.29e−01	6.47e−01	61	1.21e+02
	OP-ELM	5.04e−01	6.43e−01	103	4.18e+01
Delta Ailerons	DENFIS	5.32e−01	5.37e−01	87	4.56e+01
	OP-ELM	5.36e−01	5.42e−01	90	2.45e+01
Delta Elevators	DENFIS	5.94e−01	5.94e−01	202	1.36e+02
	OP-ELM	5.96e−01	5.92e−01	101	2.84e+01
Servo	DENFIS	4.36e−01	5.22e−01	47	1.03e+00
	OP-ELM	3.83e−01	4.60e−01	19	5.80e−01
Stocks	DENFIS	8.12e−02	2.09e+00	19	1.78e+00
	OP-ELM	7.73e−02	2.51e+00	109	2.62e+00

Regression benchmarks

[2] The series is available online from http://www.ngdc.noaa.gov/stp/SOLAR/. The International Sunspot Number is produced by the Solar Influence Data Analysis Center (SIDC) at the Royal Observatory of Belgium (Van der Linden and the SIDC Team 2008).

Table 3 Offline modeling errors for training and test subsets

Dataset	Method	Training NDEI	Test NDEI	Rules	Time (s)
Darwin SLP	DENFIS	3.88e−01	4.34e−01	34	2.50e+00
	OP-ELM	3.90e−01	4.39e−01	40	3.45e+00
ENSO	DENFIS	1.41e−01	1.73e−01	13	9.10e−01
	OP-ELM	1.41e−01	1.76e−01	18	1.20e+00
GSTA	DENFIS	5.20e−01	5.49e−01	24	2.40e+00
	OP-ELM	5.18e−01	5.27e−01	14	4.14e+00
Internet2	DENFIS	5.85e−01	9.00e−01	27	1.64e+00
	OP-ELM	5.74e−01	8.62e−01	49	2.34e+00
NAO	DENFIS	8.98e−01	1.11e+00	161	1.09e+01
	OP-ELM	9.66e−01	1.01e−01	27	3.55e+00
Santa Fe Laser	DENFIS	2.34e−01	2.34e−01	25	2.17e+00
	OP-ELM	1.14e−01	1.30e−01	63	2.70e+00
Sunspots	DENFIS	6.21e−01	6.55e−01	32	7.56e+00
	OP-ELM	6.10e−01	6.09e−01	47	8.43e+00
Tree Rings	DENFIS	7.03e−01	8.31e−01	102	9.74e+00
	OP-ELM	7.75e−01	7.72e−01	7	3.29e+00

Time series benchmarks

These results are intended to give an approximate estimation of the errors that can be achieved using some related offline methods and is not meant to be exhaustive. It should be noted that we provide training and test errors so that the results shown in this paper can be compared with the literature on online methods, such as Liang et al. 2006 and Rong et al. 2006. We focus our analysis however on the evolving modeling of the datasets, where the datasets are modeled sequentially as a whole and no distinction is made between training, validation and test subsets.

Errors are given as nondimensional error index (NDEI): The root mean square error (RMSE) divided by the standard deviation of the target sequence. That is,

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(y_j - \hat{y}_j)^2}, \quad \mathrm{NDEI} = \frac{\mathrm{RMSE}}{\mathrm{std}(y_j)},$$

where M is the total number of samples, $\hat{y}_j$ is the target output, $\hat{y}_j$ the model output, and $\mathrm{std}(y_j)$ is the sample standard deviation of the target output. The NDEI is used in order to ease comparison with previous results in the literature of evolving systems, such as (Angelov and Filev 2004b; Kasabov and Song 2002). Note however that other references dealing with the ELM related methods applied in this paper use different error measures, such as the RMSE for the dataset normalized in the range [0, 1], as in (Rong et al. 2006; Liang et al. 2006; Rong et al. 2009), or the absolute MSE (Miche et al. 2010b).

Considering now online and evolving methods, Tables 4 and 5 show the NDEI, the standard deviation of the nondimensional errors (NDE), the final number of rules and the time required for online training. The time column shows the processor time consumed for the learning process on the same environment. The lowest NDEI values achieved by the evolving methods for each dataset are highlighted in bold face. For this study, we employed a standard PC with 8 GB of RAM, and an Intel® Core$^{\mathrm{TM}}$ 2 Quad CPU Q9550 supporting a maximum frequency of 2.83 GHz, running Matlab R2009b on the GNU/Linux operating system. Tests were run with no significative competing load.

Finally, we specify some implementation details for the sake of reproducibility. For DENFIS we used the implementation available from the Knowledge Engineering and Discovery Research Institute (KEDRI) (http://www.aut.ac.nz/research/research-institutes/kedri/books). For OS-ELM the implementation by Huang et al. available from http://www3.ntu.edu.sg/home/egbhuang/ was employed. Tests with eTS were performed using the eFSLab toolbox (Dourado et al. 2009), available online from http://eden.dei.uc.pt/~dourado. The OP-ELM Toolbox (Miche et al. 2008) was used with modifications to implement the eF-OP-ELM. In general, default parameters were used.

5 Discussion

For the regression benchmarks, both OS-Fuzzy-ELM and eF-OP-ELM are overall comparable or better in terms of accuracy than DENFIS and eTS, with some exceptions, specially for the Stocks dataset. This dataset, for which OS-Fuzzy-ELM performs poorly, is considerably nonstationary. If the properties of the datasets are considered, the advantages of evolving methods over OS-Fuzzy-ELM for nonstationary series become clear. For the Stocks dataset and for most of the time series datasets (with exceptions for GSTA and Tree Rings), the evolving options yield in general better results than OS-Fuzzy-ELM. This comparison has to be interpreted with care, and would merely confirm the capability of evolving methods to better handle dynamical changes.

Considering the three evolving methods included in the tables, eF-OP-ELM achieves a NDEI in some cases better and in general comparable to that of DENFIS and eTS. More specifically, eF-OP-ELM is the best or second best method in terms of NDEI for six out of the nine time series benchmarks and nine out of the ten regression benchmarks.

In terms of computational time, all the methods provide satisfactory results, with DENFIS being the fastest in most cases (as an exception it is significantly slower for the Bank dataset). Conversely, eF-OP-ELM is the slowest method in

Table 4 Comparison in terms of accuracy, complexity and computational time

Dataset	Method	NDEI	Std NDE	Rules	Time (s)
Abalone	DENFIS	7.14e−01	5.16e−01	9	6.65e+00
	eTS	7.47e−01	4.85e−01	20	1.32e+01
	OS-Fuzzy-ELM	7.64e−01	5.25e−01	4	9.35e+01
	eF-OP-ELM	**6.92e−01**	4.71e−01	23	1.34e+03
Auto-MPG	DENFIS	1.40e+00	1.09e+00	41	9.80e−01
	eTS	1.22e+00	8.40e−01	23	5.10e−01
	OS-Fuzzy-ELM	1.13e+00	6.73e−01	37	7.41e+01
	eF-OP-ELM	**6.52e−01**	4.60e−01	40	6.59e+01
Bank	DENFIS	**2.79e−01**	1.81e−01	933	1.40e+02
	eTS	3.48e−01	2.46e−01	15	1.17e+01
	OS-Fuzzy-ELM	2.44e−01	1.69e−01	52	8.15e+01
	eF-OP-ELM	3.30e−01	2.71e−01	16	1.54e+03
Boston Housing	DENFIS	2.89e+00	1.23e+00	52	2.21e+00
	eTS	9.02e−01	8.56e−01	24	6.60e−01
	OS-Fuzzy-ELM	9.60e−01	6.70e−01	8	7.72e+01
	eF-OP-ELM	**8.44e−01**	5.67e−01	40	9.40e+01
Breast Cancer	DENFIS	1.36e+00	8.93e−01	54	2.74e+00
	eTS	1.25e+00	8.31e−01	14	2.80e−01
	OS-Fuzzy-ELM	9.46e−01	5.62e−01	70	2.08e+01
	eF-OP-ELM	**9.34e−01**	5.34e−01	16	1.81e+01
California Housing	DENFIS	**4.68e−01**	3.67e−01	57	1.37e+01
	eTS	1.45e+00	9.17e−01	46	3.14e+01
	OS-Fuzzy-ELM	1.88e+00	9.72e−01	3	6.52e+01
	eF-OP-ELM	6.67e−01	4.84e−01	12	1.88e+03
Delta Ailerons	DENFIS	6.30e−01	4.45e−01	85	2.75e+01
	eTS	**6.23e−01**	4.37e−01	1	3.82e+00
	OS-Fuzzy-ELM	5.66e−01	3.93e−01	7	7.07e+01
	eF-OP-ELM	8.15e−01	6.29e−01	17	3.32e+03
Delta Elevators	DENFIS	**6.95e−01**	4.53e−01	214	7.70e+01
	eTS	3.04e+00	2.89e+00	29	9.72e+01
	OS-Fuzzy-ELM	9.14e−01	5.36e−01	5	7.30e+01
	eF-OP-ELM	9.12e−01	6.33e−01	14	1.05e+03
Servo	DENFIS	6.50e−01	4.26e−01	64	5.60e−01
	eTS	4.01e−01	3.13e−01	10	1.50e−01
	OS-Fuzzy-ELM	3.78e−01	2.68e−01	29	1.10e+01
	eF-OP-ELM	**3.21e−01**	2.31e−01	45	1.89e+01
Stocks	DENFIS	**1.31e−01**	8.75e−02	19	2.20e+00
	eTS	5.71e−01	4.26e−01	97	6.01e+00
	OS-Fuzzy-ELM	1.92e+00	1.70e+00	90	4.81e+01
	eF-OP-ELM	2.40e−01	1.73e−01	50	2.19e+02

Regression benchmarks. The stock series is highly nonstationary. The results for OS-Fuzzy-ELM were obtained using an initialization training sequence of (500 + number of nodes) observations. For shorter sequences, the structure identified in the initialization stage is unable to yield sensible results

most cases, with some exceptions. Nonetheless, it can be observed that both DENFIS and eTS take a much higher time for some particular cases, while the eF-OP-ELM method is not affected by this problem.

In general, eF-OP-ELM exhibits adequate responsiveness and stability, though achieving in general a more compact rulebase than DENFIS and eTS. As an example, we show the evolution of the number of rules for the GSTA

Table 5 Comparison in terms of accuracy, complexity and computational time

Dataset	Method	NDEI	std NDE	Rules	Time (s)
Darwin SLP	DENFIS	4.84e−01	3.03e−01	38	3.34e+00
	eTS	**4.01e−01**	2.48e−01	17	1.91e+00
	OS-Fuzzy-ELM	7.87e−01	4.46e−01	7	5.69e+01
	eF-OP-ELM	5.29e−01	3.23e−01	15	3.30e+02
ENSO	DENFIS	**1.71e−01**	1.05e−01	16	1.34e+00
	eTS	2.00e−01	1.27e−01	44	1.94e+00
	OS-Fuzzy-ELM	2.55e−01	1.74e−01	16	4.96e+01
	eF-OP-ELM	2.10e−01	1.32e−01	19	1.76e+02
GSTA	DENFIS	4.05e−01	2.69e−01	17	2.32e+00
	eTS	**3.56e−01**	2.33e−01	36	2.44e+00
	OS-Fuzzy-ELM	3.60e−01	2.35e−01	5	6.39e+01
	eF-OP-ELM	4.09e−01	2.67e−01	13	3.49e+02
Internet2	DENFIS	**5.70e−01**	4.64e−01	29	2.41e+00
	eTS	6.54e−01	5.48e−01	47	3.92e+00
	OS-Fuzzy-ELM	9.51e−01	5.65e−01	25	3.97e+01
	eF-OP-ELM	6.18e−01	4.54e−01	21	3.16e+02
Mackey–Glass	DENFIS	3.75e−01	2.51e−01	21	9.60e−01
	eTS	3.35e−01	2.19e−01	31	8.60e−01
	OS-Fuzzy-ELM	4.33e−01	2.34e−01	27	6.30e+01
	eF-OP-ELM	**2.38e−01**	1.61e−01	50	9.97e+01
NAO	DENFIS	1.17e+00	7.09e−01	187	1.16e+01
	eTS	1.04e+00	6.05e−01	13	1.80e+00
	OS-Fuzzy-ELM	1.01e−01	5.789-01	9	9.03e+01
	eF-OP-ELM	**9.98e−01**	5.67e−01	7	1.05e+02
Santa Fe Laser	DENFIS	**2.31e−01**	2.00e−01	39	2.60e+01
	eTS	4.61e−01	3.67e−01	52	9.10e+01
	OS-Fuzzy-ELM	9.32e−01	5.70e−01	50	6.44e+01
	eF-OP-ELM	4.19e−01	3.97e−01	26	6.36e+03
Sunspots	DENFIS	**6.18e−01**	4.28e−01	37	7.25e+00
	eTS	8.10e−01	5.61e−01	33	1.06e+01
	OS-Fuzzy-ELM	8.98e−01	5.67e−01	7	6.99e+01
	eF-OP-ELM	6.30e−01	4.20e−01	30	8.88e+02
Tree Rings	DENFIS	9.59e−01	6.24e−01	119	7.53e+00
	eTS	**7.14e−01**	4.57e−01	37	4.72e+00
	OS-Fuzzy-ELM	7.94e−01	5.11e−01	8	7.78e+01
	eF-OP-ELM	8.41e−01	5.36e−01	20	3.48e+02

Time series benchmarks

and Tree Rings time series in Figs. 3 and 4, respectively. The evolution of the absolute error for the same two cases is depicted in Figs. 5 and 6, respectively. Finally, we note that eF-OP-ELM identifies rules ranked by accuracy, which should ease the interpretation process.

It should be noted that some of the sequences analyzed come from highly nonstationary systems, specially in the dynamical sense. The application cases considered include some remarkable cases of processes where nonstationarity, drift points, sudden regime shifts, and transitions between chaotic states are well documented. This is the case for instance for the GSTA (Tsonis et al. 2007; Wu et al. 2007), Santa Fe Laser (SantaFeLaser 2010; Weigend and Gershenfeld 1994) and Internet2 series (Internet2Observatory, 2008). For example, the clear increase and sudden peak in the number of rules that can be observed in Fig. 3 after observation 600 for the GSTA series corresponds to the climate shift that happened around 1910, after which global

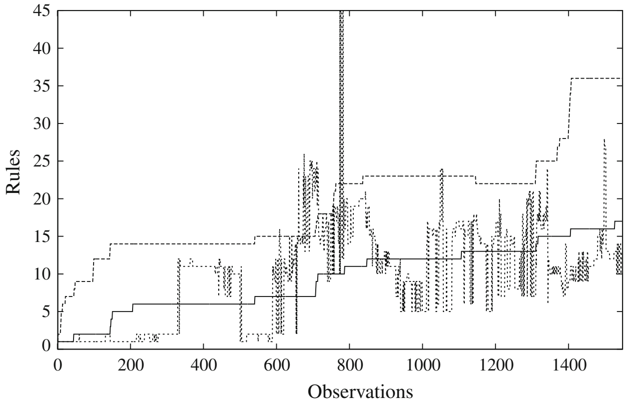

Fig. 3 Evolution of the number of rules. GSTA dataset. eTS (*dashed*), ef-OP-ELM (*short-dashed*), DENFIS (*cont.*)

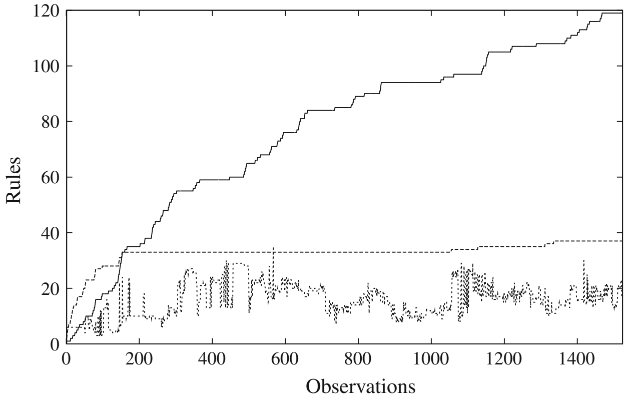

Fig. 4 Evolution of the number of rules. Tree Rings dataset. DENFIS (*cont.*), eTS (*dashed*), ef-OP-ELM (*short-dashed*)

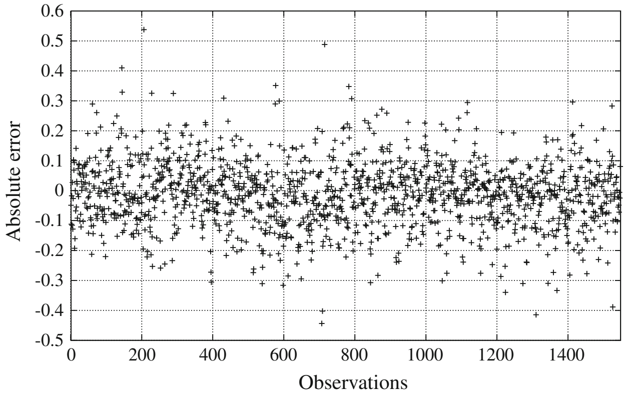

Fig. 5 Evolution of the absolute error. GSTA dataset. eF-OP-ELM model

climate transitioned to a completely new state. A similar but slightly effect can be observed for the climate shifts occured around 1940 and 1970. Regarding the machine learning benchmarks, the experiments shown consider several cases

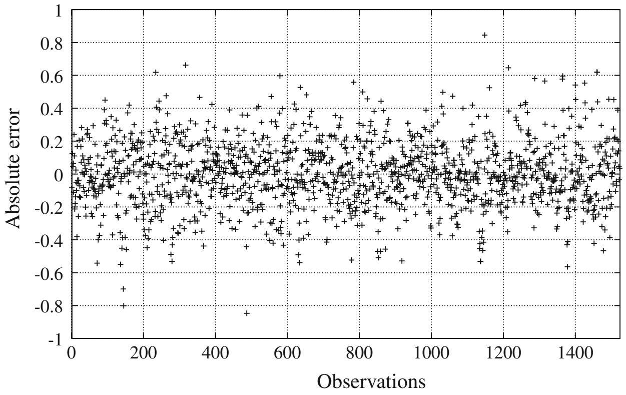

Fig. 6 Evolution of the absolute error. Tree Rings dataset. eF-OP-ELM model

where temporal and spatial nonstationary behavior has been identified (Asuncion and Newman 2010).

The constant evolution of the number of rules just reflects the fact that the dynamical complexity of the streams is evolving, with abrupt changes at some points. An ef-OP-ELM model can be seen as a projection of the target dataset on a space or basis of expansion defined by the fuzzy basis functions (the ELM machinery). The number of FBFs required to model the system can be seen as an indicator of its (evolving) complexity in terms of that basis of expansion.

Let us now discuss interpretability issues in the context of the method proposed. Automatic, data-driven identification methods for fuzzy systems can suffer from interpretability issues in general (Mikut et al. 2005; Zhou and Gan 2008). Pruning and post-processing steps are often required (Ramos and Dourado 2006) in order to attempt an interpretation of results by experts. Recently, research directions toward interpretable fuzzy rule based systems have led to results related to the granular computing paradigm in data maning and machine learning (Zadeh 1997; Pedrycz 2005; Yager 2008; Leite et al. 2009). These research efforts lay out an approach and a number of principles, where concepts such as proximity, similarity, indistinguishability, specificity or functionality are central. While the approach followed in this paper may appear in contradiction with the cited works, it should be noted that this paper proposes an alternative identification approach.

The fact that the antecedent parameters are generated following a process where a random projection is involved does not necessarily mean that the rules show a random arrangement in the sense of meaningless. As explained in Sect. 3, the parameters are generated within certain ranges that are related to the distribution of the incoming data. In fact, this approach can help discover underlying structures in the incoming streams. Furthermore, experimental results support the fact that the number of rules identified by eF-OP-ELM can be in actuality comparable or significantly lower as compared with more traditional approaches represented by DENFIS and eTS. In addition, the fuzzy basis functions (and consequently the fuzzy rules) are exactly ranked by accuracy, which can be of help in the interpretation process.

Regarding direct extensions of the eF-OP-ELM discussed here, we can mention the use of faster model selection approaches recently proposed, as well as the possibility of a chunk-by-chunk learning mode. Further work is needed as well to analyze the impact of different types of membership functions and fuzzy operators on performance and interpretability.

6 Conclusions

An approach to the identification of evolving fuzzy inference systems based on the OP-ELM methodology has been proposed. The approach extends OP-ELM in order to get a fast, online evolving learning algorithm. The new method has been shown to be competitive in terms of accuracy and compactness. It constitutes a case of application of a random projection method to identifying evolving fuzzy systems.

The proposed method, eF-OP-ELM, randomly generates a set of simple antecedents of fuzzy rules, with random structure for the antecedents and random values for the parameters of input membership functions of Gaussian type. Then, a generalized LARS method is used to rank the fuzzy basis functions (or normalized fuzzy rules) according to their accuracy. The best number of fuzzy rules is selected by performing a fast computation of the leave-one-out validation error based on the PRESS statistics. Finally, the consequents parameters are determined analytically.

The method is general and can be applied in areas such as time series prediction, process control, robotics and autonomous systems. Its accuracy compares favorably against other well known methods in the field of evolving fuzzy systems, namely DENFIS and eTS, as well as the online method OS-Fuzzy-ELM.

From the perspective of evolving systems and fuzzy logic, the method proposed here provides an alternative approach to identifying evolving fuzzy systems. In essence, it has been shown that a random projection scheme such as ELM is competitive in terms of accuracy and rulebase compactness as compared with traditional approaches. In particular, the model selection approach of OP-ELM provides a way to extract compact fuzzy rulebases based on the ELM scheme. In view of these results, the random projection approach is not only an appealing alternative but it also seems worth being investigated further in the context of evolving systems.

Acknowledgments The respective authors of the software tools used in this work (see references and links in previous sections) are acknowledged for making their software publicly available. FMP is supported by a Marie Curie Intra-European Fellowship for Career Development (grant agreement PIEF-GA-2009-237450) within the European Community's Seventh Framework Programme (FP7/2007–2013).

Appendix: Incremental initialization of eF-OP-ELM

This appendix discusses an incremental procedure for the initialization of ef-OP-ELM models. It was not included in previous sections for clarity's sake and may not be required in many practical setups, as the number of samples required for batch initialization (as explained in Sect. 3) is small.

The incremental algorithm presented here can start with the first data sample and continues until the minimum initialization number of samples has been observed. More specifically, the initialization continues until the number of observations reaches M, the maximum number of antecedent parameters generated for $\mathbf{H}_0(j)$. The procedure consists of the following steps, which can replace steps 1, 2 and 3 in Algorithm 1:

Algorithm 2 Incremental Initialization of Evolving Fuzzy OP-ELM (eF-OP-ELM)

```
1: Create an empty ordered set of antecedent parameters (centers and radii), $C$ and $A$, and an empty matrix of fuzzy basis functions, $\mathbf{H}_0(0)$.
2: For each new input-output observation, $(\mathbf{x}_j, \mathbf{y}_j)$, with $j = 1, \ldots, M$, do:
  2.1: Generate a new center and radius and add them to the sets of antecedent parameters $C$ and $A$.
  2.2: Add an additional column (with the newly created antecedent) and an additional row (for the newly observed sample) to $\mathbf{H}_0(j)$.
  2.3: Rank the fuzzy rule antecedents in $\mathbf{H}_0(j)$ and select the best number of rules in order to generate $\mathbf{H}(j)$.
  2.4: Find the analytical solution for $\mathbf{Q}(j)$ (as in (2)) and compute the model output, $\hat{y}_{j+1}$.
```

This way, during step 2, a total of *j* fuzzy basis functions are available when *j* input-output samples have been observed. That is, $H_0(j)$ has *j* rows (observations) and *j* columns (fuzzy basis functions), which are then subject to ranking ans selection in the next steps. In step 2.1, the procedure used to generate the centers and radii from a uniform random distribution follows the same scheme as described in Sect. 3.

References

Achlioptas D (2003) Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J Comput Syst Sci 66(4):671–687

Angelov P, Filev D (2004) Flexible models with evolving structure. Int J Intell Syst 19(4):327–340

Angelov P, Filev D, Kasabov N (2008) Evolving fuzzy systems—preface to the special section. IEEE Trans Fuzzy Syst 16(6):1390–1392

Angelov P, Filev D, Kasabov N (eds) (2010) Evolving intelligent systems. Methodoloy and applications. Wiley/IEEE Press

Angelov PP, Filev DP (2004) An approach to online identification of Takagi–Sugeno fuzzy models. IEEE Trans Syst Man Cybern B 34(1):484–498

Angelov PP, Filev DP (2005) Simpl_eTS: a simplified method for learning evolving Takagi–Sugeno fuzzy models. In: Proceedings of the IEEE International Conference on fuzzy systems, Reno, NV, USA, pp 1068–1073

Asuncion A, Newman DJ (2010) UCI machine learning repository. University of California, Irvine, Center for Machine Learning and Intelligent Systems. http://archive.ics.uci.edu/ml/

Birattari M, Bontempi G, Bersini H (1999) Lazy learning meets the recursive least squares algorithm. In: Advances in Neural Information Processing Systems (NIPS), vol 11. MIT Press, Cambridge, pp 375–381

Dourado A, Aires L, Victor J (2009) eFSLab: Developing evolving fuzzy systems from data in a friendly environment. In: Proc 10th Eur Control Conf, Prague, Czech Republic, pp 922–927

Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499

ESTSP07 (2010) ESTSP: European Symposium on Time Series Prediction. http://www.estsp.org

Feng G, Huang GB, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Netw 20(8):1362–1357

Fradkin D, Madigan D (2003) Experiments with random projections for machine learning. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, pp 517–522

Haykin SS (1998) Neural networks: a comprehensive foundation, 2nd edn. Prentice-Hall, Englewood Cliffs. ISBN: 978-0132733502

Higham NJ (2002) Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia

Huang GB (2008) Reply to "comment on the extreme learning machine". IEEE Trans Neural Netw 19(8):1495–1496

Huang GB, Chen L, Siew CK (2006) Universal approximaion using incremental constructuve feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489-5-01

Hurrell JW, Deser C (2009) North atlantic climate variability: the role of the north atlantic oscillation. J Mar Syst 78(1):28–41

Hyndman RJ (2010) Time series data library. http://www.robjhyndman.com/TSDL

Internet2Observatory (2008) The Internet2 Observatory. http://www.internet2.edu/observatory/

Kasabov N (2007) Evolving connectionist systems: the knowledge engineering approach, 2nd edn. Springer, Berlin

Kasabov NK, Song Q (2002) DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans Fuzzy Syst 10(2):144–154

Leite D, Costa P, Gomide F (2009) Interval-based evolving modeling. In: IEEE Workshop on evolving and self-developing intelligent systems, Nasville, TN, USA, pp 1–8

Lendasse A, Sorjamaa A, Miche Y (2010) The OP-ELM toolbox. Time Series Prediction and Chemoinformatics Group. Department of Information and Computer Science. Aalto University School of Science and Technology. http://www.cis.hut.fi/projects/tsp/index.php?page=opelm

Liang NY, Huang GB, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17(6):1411–1423

Lughofer ED (2008) Flexfis: A robust incremental learning approach for evolving Takagi–Sugeno fuzzy models. IEEE Trans Fuzzy Syst 16(6):1393–1410

Mackey MC, Glass L (1977) Oscillations and chaos in physiological control systems. Science 197(4300):287–289

Miche Y, Sorjamaa A, Lendasse A (2008) OP-ELM: Theory, experiments and a toolbox. In: Proceedings of the international conference on artificial neural networks. Lecture notes in computer science, vol 5163, Prague, Czech Republic, pp 145–154

Miche Y, Schrauwen B, Lendasse A (2010) Machine learning techniques based on random projections. In: 18th European symposium on artificial neural networks, computational intelligence and machine learning, pp 295–302

Miche Y, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A (2010) OP-ELM: optimally pruned extreme learning machine. IEEE Trans Neural Netw 21(1):158–162

Mikut R, Jakel J, Groll L (2005) Interpretability issues in data-based learning of fuzzy systems. Fuzzy Sets Syst 150(2):179–197

Montesino Pouzols F, Lendasse A, Barriga A (2008) Fuzzy Inference Based Autoregressors for Time Series Prediction Using Nonparametric Residual Variance Estimation. In: Proceedings of the IEEE International conference on fuzzy systems, Hong Kong, China, pp 613–618

Montesino Pouzols F, Lendasse A, Barriga A (2008) xftsp: a tool for time series prediction by means of fuzzy inference systems. In: Proceedings of the IEEE international conference on intelligent systems, Varna, Bulgaria, pp 2-2-2-7

Montesino Pouzols F, Lendasse A, Barriga A (2010) Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation. Fuzzy Sets Syst 161(4):471–497

Moreno-Velo FJ, Baturone I, Barriga A, Sánchez-Solano S (2007) Automatic tuning of complex fuzzy systems with Xfuzzy. Fuzzy Sets Syst 158(18):2026–2038

Myers RH (2000) Classical and modern regression with applications, 2nd edn. Duxbury Press, North Scituate

Pedrycz W(2005) Knowledge-based clustering: from data to information granules. Wiley, New York

Platt J (1991) A resource-allocating network for function interpolation. Neural Comput 3(2):213–225

Poggio T, Girosi F (1989) A Theory of Networks for Approximation and Learning, vol 1140. MIT Press, Cambridge

Ramos JV, Dourado A (2006) Pruning for interpretability of large spanned eTS. In: Proceedings of the 2006 international symposium on evolving fuzzy systems, EFS'06, IEEE Press, Ambelside, Lake District, UK, pp 55–60

Rong HJ, Sundararajan N, Huang GB, Saratchandran P (2006) Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. Fuzzy Sets Syst 157(9):1260–1275

Rong HJ, Huang GB, Sundararajan N, Saratchandran P (2009) Online sequential fuzzy extreme learning machine for function approximation and classification problems. IEEE Trans Syst Man Cybern B 39(4):1067–1072

SantaFeLaser (2010) The Santa Fe Time Series Competition Data. Data Set A: Laser generated data. http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

Schölkopf B, Smola AJ (2002) Learning with kernels. Support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge. ISBN: 0262194759

Similä T, Tikka J (2005) Multiresponse sparse regression with application to multidimensional scaling. In: Proceedings of the international conference on artificial neural networks, Warsaw, Poland, vol 3967, pp 97–102

Sorjamaa A, Miche Y, Weiss R, Lendasse A (2008) Long-Term Prediction of Time Series using NNE-based Projection and OP-ELM. In: Proceedings of the international joint conference on neural networks, Hong Kong, China, pp 2675–2681

StatLib (2010) Department of Statistics, Carnegie Mellon University. http://lib.stat.cmu.edu/datasets/

Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore

Tsonis AA, Swanson K, Kravtsov S (2007) A new dynamical mechanism for major climate shifts. Geophys Res Lett 34:L13,705, 5 pp

Van der Linden RAM, the SIDC Team (2008) Online catalogue of the Sunspot Index. RWC Belgium, World Data Center for the Sunspot Index, Royal Observatory of Belgium, years 1748–2007, http://sidc.oma.be/html/sunspot.html

Weigend A, Gershenfeld N (1994) Times series prediction: forecasting the future and understanding the past. Addison-Wesley, Reading

Wu Z, Huang NE, Long SR, Peng CK (2007) On the trend, detrending, and variability of nonlinear and nonstationary time series. Proc Nat Acad Sci 104(38):14889–14894

Yager R (2008) Measures of specificity over continuous spaces under similarity relations. Fuzzy Sets Syst 159(17):2193–2210

Zadeh LA (1997) Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst 90(2):111–127

Zeng XJ, Singh MG (1995) Approximation theory of fuzzy systems—MIMO case. IEEE Trans Fuzzy Syst 3(2):219–235

Zhou SM, Gan JQ (2008) Low-level interpretability and high-level interpretability: a unified view of datadriven interpretable fuzzy system modelling. Fuzzy Sets Syst 159(23):3091–3131