# MSAFIS: An evolving Fuzzy Inference System

José de Jesús Rubio<sup>1</sup>, Abdelhamid Bouchachia<sup>2</sup>

1 Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional

Av. de las Granjas no.682, Col. Santa Catarina, México D.F., 02250, México (phone:(+52)55-57296000-64497)

(e-mail: jrubioa@ipn.mx; rubio.josedejesus@gmail.com)

2 Dept of Computing and Informatics, Faculty of Science & Technology, Bournemouth University, UK

(e-mail: abouchachia@bournemouth.ac.uk)

#### Abstract

In this paper, the problem of learning in big data is considered. To solve this problem, a new algorithm is proposed as the combination of two important evolving and stable intelligent algorithms: the sequential adaptive fuzzy inference system (SAFIS), uniform stable backpropagation algorithm (SBP). The modified sequential adaptive fuzzy inference system (MSAFIS) is the SAFIS with the difference that the SBP is used instead of the Kalman filter for the updating of parameters. The SBP has the advantage that is obtains a better learning than the Kalman filter in big data. The effectiveness of the studied methods is verified by two experiments.

**Keywords:** Evolving intelligent systems, stable intelligent systems, learning, big data, brain signals, eye signals.

### 1 Introduction

The recent years have witnessed the emergence of an important topic related to process learning which is learning from big data (LBD). LBD is concerned with the development and application of learning algorithms for very large, possibly complex, datasets that cannot be accommodated in the main memory. To cope with this requirement, different techniques and technologies have been proposed:

- 1. Parallel and distributed computing (e.g., Hadoop): data is split into portions and sent to parallel machines to be processed and learned from.
- 2. Online learning, known also as sequential learning, one-pass learning, real-time learning, evolving systems, etc.): the learning algorithms learn sequentially, either batchbased or point-based, potentially using one single machine.

Although these techniques are not new from a pure scientific point of view, the deluge of data available everywhere has given a refreshing and renewable interest to them. In this paper we will focus on online learning.

OL faces the challenge of accurately estimating models using incoming data whose statistical characteristics are not known a priori. In non-stationary environments, the challenge becomes even more important, since the model's behavior may need to change drastically over time [11]. OL aims at ensuring continuous adaptation of the model being fitted to the data. When learning, ideally only the model should be stored in memory. For instance in rule-based systems (RBS), only rules should be memorized. The model is then adjusted in future learning steps. In the case of RBS, as new data arrive, new rules may be created and existing ones may be modified or removed allowing the overall model to evolve over time [8, 36].

In general evolving intelligent systems requires the development of online learning algorithms which are computationally efficient and whose structure is very flexible in order to adapt to ever-changing environments. Online processing of data with a particular focus on the design issues of online evolving systems is considered in [6]. In [8], online self-learning fuzzy classifier, called GT2FC standing for "Growing Type-2 Fuzzy Classifier" is presented. The proposed approach shows how type-2 fuzzy rules can be learned online in an evolving way from data streams. GT2FC was applied in the context of smart homes. In [7], the authors explore the application of interactive and online learning of user profiles in the context of information filtering using evolutionary algorithms. In [16], an evolving algorithm for learning computer user behavior is introduced. In the same vein, the authors in [4] present an online inference approach for developing context-aware systems in ubiquitous applications.

Evolving fuzzy systems have been very popular, for instance in [20] a granular neural network framework is proposed to develop an evolving fuzzy system from fuzzy data streams. The same authors [21] adopt a granular fuzzy control approach as a step toward the development of a more general framework for online modeling and control of unknown non-stationary processes with no human intervention. The work described in [23] suggests an enhanced version of the evolving participatory learning approach which includes both a utility measure to shrink rule bases and a variable cluster radius mechanism to improve the cluster structure. The use of evolving classifiers for activity recognition from sensor readings in ambient assisted living environments is described by [27]. In [30] and [31], two novel evolving neurofuzzy algorithms are developed. A new concept is addressed in [41] for handling drifts in data streams during the run of online evolving modeling processes in a regression context.

The paper [40] provides a comprehensive literature survey on the evolving spiking neural network, while certain methods of spiking neural networks are reviewed in [18] and claimed to be suitable for the creation of a unifying computational framework for learning and understanding of various spatio- and spectro-temporal brain data. In [48], the authors propose a task recommendation framework based on an unified probabilistic matrix factorization, aiming to recommend tasks to workers in dynamic scenarios. The paper [34] presents some methods for training pattern (prototype) selection, class-specific feature selection and classification for automated learning. In [39], the authors introduce the usage of multivariate orthogonal space transformations and vectorized time-series models in combination with data-driven system identification models to achieve an enhanced performance of residualbased fault detection A novel approach for time series forecasting based on ordered weighted averaging operators as linear filter and forecasting models is addressed in [5]. In [10] and [43], acquisition systems to obtain real data of wind turbines and wind farms are used for the fault detection. An interactive method and the underlying theory for solving multiple objective mathematical programming problems is introduced in [24]. The paper [22] demonstrates the concept of developing a cellular computational networks using dimensionality reduction in a dynamic recurrent network for scalability and better performance for the dynamics learning of large networked systems.

Stable intelligent systems are characterized by the stability criterion, i.e., for bounded inputs in the algorithms, there are also bounded outputs and bounded parameters. In [1], the author uses an induced  $L\infty$  approach to create a new filter with a finite impulse response structure for state-space models with external disturbances. The model predictive stabilization problem for Takagi-Sugeno fuzzy multilayer neural networks with general terminal weighting matrix are investigated in [2]. In [3], an error passivation approach is used to derive a new passive and exponential filter for switched Hopfield neural networks with time-delay and noise disturbance. Two stable intelligent controllers for nonlinear systems with dead-zone are addressed in [28] and [29]. In [35] and [42], two stable optimal controls are introduced. A framework to assess and improve the stability of biomarkers is derived in [44] by a novel method.

However, most of these algorithm operate offline and are not designed to handle big data. The present paper presents the combination of two algorithms: the sequential adaptive fuzzy inference system (SAFIS) [33] which is an evolving algorithm and the stable backpropagation algorithm (SBP) [36] which is a stable algorithm. Such combination, called the Modified Sequential Adaptive Fuzzy Inference System (MSAFIS), aims to devise an efficient evolving algorithm that can cope with data streams as a case of dig data. MSAFIS exploits the SBP algorithm to update of parameters, while in SAFIS relies on the Kalman filter. SBP has the advantage that it outperforms Kalman filter [36].

The paper is organized as follows. In section 2, the SAFIS, SBP, and MSAFIS algorithms are detailed. In section 3, the brain encephalography (EEG) and the eye electro-oculogram (EOG) signals are described. Using an EEG and EOG dataset, SAFIS, SBP and MSAFIS are evaluated and compared in section 4. Section 5 concludes the paper and suggests future research directions.

## 2 Presentation of the algorithms

In this section the three algorithms SAFIS, SBP, and MSAFIS are described.

#### 2.1 SAFIS algorithm

The sequential adaptive fuzzy inference system (SAFIS) is developed based on the functional equivalence between a radial basis function network and a fuzzy inference system (FIS) resulting in a neuro-fuzzy system. In SAFIS, the concept of "Influence" of a fuzzy rule is introduced and using this the fuzzy rules are added or removed based on the input data received so far. If the input data do not warrant adding of fuzzy rules, then only the parameters of the "closest" (in a Euclidean sense) rule are updated using an extended Kalman filter (EKF) scheme.

The SAFIS algorithm is summarized as below [33]: For each observation (x(k), y(k)) where  $x(k) \in \Re^N$ ,  $y(k) \in \Re$  and k = 1, 2, ..., do 1) Compute the overall system output:

$$\widehat{y}_{k} = \frac{\sum_{j=1}^{M} v_{j}(k) R_{j}(x_{i}(k))}{\sum_{j=1}^{M} R_{j}(x_{i}(k))}$$
(1)

where

$$R_{j}(x_{i}(k)) = \exp\left(-\frac{1}{\sigma_{j}^{2}(k)} \|x_{i}(k) - c_{ij}(k)\|^{2}\right)$$

and M is the number of fuzzy rules,  $R_j(x_i(k))$  is the firing strength of the  $j^{th}$  rule,  $v_j(k)$  is the weight of the normalized rule. Note that each rule is represented as a radial basis function described by its center  $c_{ij}(k)$  and its spread  $\sigma_j(k)$ .

2) Calculate the parameters required in the growth criterion:

$$\varepsilon(k) = \max\left\{\varepsilon_{\max}\gamma^k, \varepsilon_{\min}\right\}, \ 0 < \gamma < 1 \tag{2}$$

where  $\varepsilon_{\text{max}}$  and  $\varepsilon_{\text{min}}$  are the threshold largest and smallest distances admitted between the inputs and corresponding nearest center of rules. The parameter  $\gamma$  (0 <  $\gamma$  < 1) indicates the decay constant. The error of the  $k^{th}$  input is given as follows:

$$e(k) = y(k) - \hat{y}(k) \tag{3}$$

where y(k) and  $\hat{y}(k)$  are the output and the estimated output, respectively.

3) Apply the criterion for adding rules, if the following two conditions are satisfied: If

$$\|x_i(k) - c_{ij}(k)\| > \varepsilon(k) \tag{4}$$

and

$$E_{\inf}(M+1) = |e(k)| \frac{(1.8K ||x_i(k) - c_{ij}(k)||)^N}{\sum_{j=1}^{M+1} (1.8\sigma_j(k))^N} > e_g$$
(5)

where  $e_g$  is the growing threshold. A new rule M + 1 is added if  $e_g$  is exceeded.

The new rule M + 1 is given as follows:

$$v_{M+1} = e(k) c_{M+1} = x_i(k)$$
(6)  
$$\sigma_{M+1} = K ||x_i(k) - c_{ij}(k)||$$

If no rule is added, the nearest rule jm is obtained as follows:

$$\min_{j} R_j(x_k) \Longrightarrow jm = j \tag{7}$$

and adjust the system parameters  $v_j(k)$ ,  $c_{ij}(k)$ ,  $\sigma_j(k)$  for the nearest rule only by using the extended Kalman filter (EKF) method:

$$\theta(k) = \theta(k-1) + P_{k-1}B_{k-1} \left[ a + B_{k-1}^T P_{k-1}B_{k-1} \right]^{-1} e(k)$$

$$P_k = P_{k-1} - P_{k-1}B_{k-1} \left[ p + B_{k-1}^T P_{k-1}B_{k-1} \right]^{-1} B_{k-1}^T P_{k-1} + qI$$
(8)

where  $\theta(k) = [\theta_1(k) \cdots \theta_{N+2}]^T = [c_{1jm}(k), \dots, c_{Njm}(k), v_{jm}(k), \sigma_{jm}(k)], q \text{ and } p \text{ are}$ parameters selected by the designer,  $0 < q < 1, 0 < p < 1, B_{k-1} = [B_{1,k-1}, \cdots, B_{N+2,k-1}^T]^T$ ,  $B_{i,k-1} = \frac{2[v_{jm}(k) - \hat{y}(k-1)]R_{jm}(x_i(k))[x_i(k) - c_{ijm}(k)]}{\left[\sum_{j=1}^{M} R_j(x_i(k))\right]\sigma_{jm}^2(k)}$  for  $i = \overline{1, N}, B_{N+1,k-1} = \frac{2[v_{jm}(k) - \hat{y}(k-1)]R_{jm}(x_i(k))||x_i(k) - c_{ijm}(k)||^2}{\left[\sum_{j=1}^{M} R_j(x_i(k))\right]\sigma_{jm}^2(k)}, I$  is the identity matrix.

4) If the following criterion is satisfied:

$$E_{\rm inf}(jm) = |v_{jm}(k)| \frac{(1.8\sigma_{jm}(k))^N}{\sum_{j=1}^M (1.8\sigma_j(k))^N} < e_p \tag{9}$$

then, remove the jm rule and reduce the dimensionality of EKF. Note that  $e_p$  is the pruning threshold.

**Remark 1** The significance of a neuron proposed in growing and pruning radial basis function (GAP-RBF) neural network is defined based on the average contribution of an individual neuron to the output of the RBF network. Under this definition, one may need to estimate the input distribution range  $S(X) = \frac{|v_{jm}(k)|}{M}$ . However, the influence of a rule intro- $\sum_{j=1}^{M} (1.8\sigma_j(k))^N$ 

duced in this paper is different from the significance of a neuron proposed in GAP-RBF. In fact, the influence of a neuron is defined as the relevant significance of the neuron compared to summation of significance of all the existing RBF neurons. As seen from equation (7), with the introduction of influence one need not estimate the input distribution range and the implementation has been simplified. **Remark 2** In parameter modification, SAFIS utilizes a winner rule strategy similar to the work done by [14]. The key idea of the winner rule strategy is that only the parameters related to the selected winner rule are updated by the EKF algorithm in every step. The 'winner rule' is defined as the rule that is closest (in the Euclidean distance sense) to the current input data. As a result, SAFIS is computationally efficient.

**Remark 3** In SAFIS, some parameters need to be decided in advance according to the problems considered. They include the distance thresholds ( $\varepsilon_{\max}$ ,  $\varepsilon_{\min}$ ,  $\gamma$ ), the overlap factor K for determining the width of the newly added rule, the growing threshold ( $e_g$ ) for a new rule and the pruning threshold ( $e_p$ ) for removing an insignificant rule. A general selection procedure for the predefined parameters is given as follows: max is set to around the upper bound of input variables;  $\varepsilon_{\min}$  is set to around 10% of  $\varepsilon_{\max}$ ;  $\gamma$  is set to around 0.99.  $e_p$  is set to around 10% of  $e_g$ .  $\varepsilon_{\max}$  is observed in the range [1.0, 10.0]. The overlap factor K is utilized to initialize the width of the newly added rule and chosen according to different problems, it is observed in the range [1.0, 2.0]. The growing threshold  $e_g$  is chosen according to the system performance, it is observed in the range [0.001, 0.05]. The smaller the  $e_g$ , the better the system performance, but the resulting system structure is more complex.

#### 2.2 SBP algorithm

The stable backpropagation (SBP) algorithm is developed with a new time-varying rate to guarantee its uniformly stability for online identification and its identification error converges to a small zone bounded by the uncertainty. The weights' error is bounded by the initial weights' error, i.e., hence the overfitting is avoided. The SBP algorithm is as follows [36]:

1) Compute the output of the nonlinear system y(k) with equation (10). Note, that the nonlinear system may have the structure represented by equation (10) and the parameter N is selected according to this nonlinear system.

$$y(k) = f[x_k] \tag{10}$$

where  $x_k = [x_1(k) \dots, x_i(k), \dots, x_N(k)]^T = [y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m)]^T \in \Re^{N \times 1}$  (N = n + m) is the input vector,  $u(k-1) \in \Re$  is the input of the plant,  $y(k) \in \Re$  is the output of the plant, and f is an unknown nonlinear function,  $f \in C^{\infty}$ .

2) Select the following parameters; v(1) and w(1) as random numbers between 0 and 1; M as an integer number, and  $\alpha_0$  as a positive value smaller or equal to 1; obtain the output  $\hat{y}(1)$  using equation (11).

$$\widehat{y}(k) = v(k)\Phi(k) = \sum_{j=1}^{M} v_j(k)\phi_j(k)$$

$$\Phi(k) = \left[\phi_1(k), \dots, \phi_j(k), \dots, \phi_M(k)\right]^T$$

$$\phi_j(k) = \tanh\left(\sum_{i=1}^{N} w_{ij}(k)x_i(k)\right)$$
(11)

3) For each iteration k, obtain the output  $\hat{y}(k)$  with equation (11), obtain the identification error e(k) with equation (12):

$$e(k) = \hat{y}(k) - y(k) \tag{12}$$

and update the parameters  $v_j(k)$  and  $w_{ij}(k)$  using equation (13):

$$v_j(k) = v_j(k-1) - \alpha(k-1)\phi_j(k-1)e(k-1)$$
  

$$w_{ij}(k) = w_{ij}(k-1) - \alpha(k-1)\sigma_{ij}(k-1)e(k-1)$$
(13)

where the new time varying rate  $\alpha(k)$  is:

$$\alpha(k-1) = \frac{\alpha_0}{2\left(\frac{1}{2} + \sum_{j=1}^M \phi_j^2(k-1) + \sum_{j=1}^M \sum_{i=1}^N \sigma_{ij}^2(k-1)\right)}$$
  
where  $i = 1, \dots, N, j = 1, \dots, M, \sigma_{ij}(k-1) = v_j(k) \operatorname{sech}^2(\sum_{i=1}^N w_{ij}(k-1)x_i(k-1))x_i(k-1) \in \Re.$ 

**Remark 4** There are two conditions for applying this algorithm for nonlinear systems: the first one is that the nonlinear system may have the form described by (10), and the second one is that the uncertainty  $\mu(k) = y(k) - \sum_{j=1}^{M} v_j^* \phi_j^*$  may be bounded,  $\phi_j^* = \tanh(\sum_{i=1}^{N} w_{ij}^* x_i(k)), v^*$  and  $w_{ij}^*$  are unknown weights such that the uncertainty  $\mu(k)$  is minimized.

**Remark 5** The value of the parameter used for the stability of the algorithm  $\overline{\mu}$  is unimportant, because this parameter is not used in the algorithm. The bound of  $\mu(k)$  is needed to guarantee the stability of the algorithm, but it is not used in the SBP algorithm (11), (12), (13).

**Remark 6** The proposed SBP has one hidden layer. It was reported in the literature that a feedforward neural network with one hidden layer is enough to approximate any nonlinear system. **Remark 7** Note that the behavior of the algorithm could be improved or deteriorated by changing the values of M or  $\alpha_0$ .

#### 2.3 MSAFIS

The modified sequential adaptive fuzzy inference system (MSAFIS) is the SAFIS algorithm with the modification of the equations (3) and (8) by the equations (12) and (13), and using the parameters of the SAFIS algorithm  $c_{ij}(k)$ ,  $\sigma_j(k)$ ,  $v_j(k)$  instead of the parameters of the SBP algorithm  $w_{ij}(k)$ ,  $v_j(k)$ . The MSAFIS algorithm is summarized as follows.

For each observation (x(k), y(k)) where  $x(k) \in \Re^N$ ,  $y(k) \in \Re$  and k = 1, 2, ..., do 1) Compute the overall system output:

$$\widehat{y}_{k} = \frac{\sum_{j=1}^{M} v_{j}(k) R_{j}(x_{i}(k))}{\sum_{j=1}^{M} R_{j}(x_{i}(k))}$$
(14)

where

$$R_{j}(x_{i}(k)) = \exp\left(-\frac{1}{\sigma_{j}^{2}(k)} \|x_{i}(k) - c_{ij}(k)\|^{2}\right)$$

and M is the number of fuzzy rules,  $R_j(x_i(k))$  is the firing strength of the  $j^{th}$  rule,  $v_j(k)$  is the weight of the normalized rule. Note that each rule is represented as a radial basis function described by its center  $c_{ij}(k)$  and its spread  $\sigma_j(k)$ .

2) Calculate the parameters required in the growth criterion:

$$\varepsilon(k) = \max\left\{\varepsilon_{\max}\gamma^k, \varepsilon_{\min}\right\}, \ 0 < \gamma < 1 \tag{15}$$

where  $\varepsilon_{\text{max}}$  and  $\varepsilon_{\text{min}}$  are the threshold largest and smallest distances admitted between the inputs and corresponding nearest center of rules. The parameter  $\gamma$  (0 <  $\gamma$  < 1) indicates the decay constant. The error of the  $k^{th}$  input is given as follows:

$$e(k) = \hat{y}(k) - y(k) \tag{16}$$

3) Apply the criterion for adding rules, if the following two conditions are satisfied: If

$$\|x_i(k) - c_{ij}(k)\| > \varepsilon(k) \tag{17}$$

and

$$E_{\inf}(M+1) = |e(k)| \frac{\left(1.8K \|x_i(k) - c_{ij}(k)\|\right)^N}{\sum_{j=1}^{M+1} (1.8\sigma_j(k))^N} > e_g$$
(18)

where  $e_g$  is the growing threshold. A new rule M + 1 is added if  $e_g$  is exceeded.

The new rule M + 1 is given as follows:

$$v_{M+1} = e(k) c_{M+1} = x_i(k)$$
(19)  
$$\sigma_{M+1} = K ||x_i(k) - c_{ij}(k)||$$

If no rule is added, the nearest rule jm is obtained as follows:

$$\min_{j} R_j(x_k) \Longrightarrow jm = j \tag{20}$$

and adjust the system parameters  $v_j(k)$ ,  $c_{ij}(k)$ ,  $\sigma_j(k)$  for the nearest rule only by using the stable brackpropagation algorithm:

$$\theta(k) = \theta(k-1) - \alpha(k-1)B_{k-1}e(k-1)$$
(21)

where 
$$\theta(k) = [\theta_1(k), \cdots, \theta_{N+2}]^T = [c_{1jm}(k), \dots, c_{Njm}(k), v_{jm}(k), \sigma_{jm}(k)], B_{k-1} = [B_{1,k-1}, \cdots, B_{N+2,k-1}^T]^T,$$
  
 $B_{i,k-1} = \frac{2[v_{jm}(k) - \hat{y}(k-1)]R_{jm}(x_i(k))[x_i(k) - c_{ijm}(k)]}{\left[\sum_{j=1}^M R_j(x_i(k))\right]\sigma_{jm}^2(k)}$  for  $i = \overline{1, N}, B_{N+1,k-1} = \frac{2[v_{jm}(k) - \hat{y}(k-1)]R_{jm}(x_i(k))||x_i(k) - c_{ijm}(k)||^2}{\left[\sum_{j=1}^M R_j(x_i(k))\right]\sigma_{jm}^3(k)}$   
 $B_{N+2,k-1} = \frac{R_{jm}(x_i(k))}{\left[\sum_{j=1}^M R_j(x_i(k))\right]},$  the new time varying rate  $\alpha(k-1)$  is:  
 $\alpha(k-1) = \frac{\alpha_0}{2\left(\frac{1}{2} + \sum_{j=1}^{N+2} B_{k-1}^2\right)}$ 

where  $\alpha_0$  is a parameter selected by the designer,  $0 < \alpha_0 < 1$ , . 4) If the following criterion is satisfied:

If

$$E_{\rm inf}(jm) = |v_{jm}(k)| \frac{(1.8\sigma_{jm}(k))^N}{\sum_{j=1}^M (1.8\sigma_j(k))^N} < e_p$$
(22)

then, remove the jm rule and reduce the dimensionality of SBP. Note that  $e_p$  is the pruning threshold.

**Remark 8** In MSAFIS, some parameters need to be decided in advance according to the problems considered. They include the distance thresholds ( $\varepsilon_{\max}$ ,  $\varepsilon_{\min}$ ,  $\gamma$ ), the overlap factor K for determining the width of the newly added rule, the growing threshold ( $e_g$ ) for a new rule and the pruning threshold ( $e_p$ ) for removing an insignificant rule. A general selection procedure for the predefined parameters is given as follows: max is set to around the upper bound of input variables;  $\varepsilon_{\min}$  is set to around 10% of  $\varepsilon_{\max}$ ;  $\gamma$  is set to around 0.99.  $e_p$  is set to around 10% of  $e_g$ .  $\varepsilon_{\max}$  is observed in the range [1.0, 10.0]. The overlap factor K is utilized to initialize the width of the newly added rule and chosen according to different problems, it is observed in the range [1.0, 2.0]. The growing threshold  $e_g$  is chosen according to the system performance, it is observed in the range [0.001, 0.05]. The smaller the  $e_g$ , the better the system performance, but the resulting system structure is more complex.

## 3 The brain and eye signals

This section describes the characteristics of the brain and eye signals.

#### 3.1 The EEG signals

The difference of the potential in one membrane is obtained by the exchange between the ions (Na+,Cl-,K+) being in the same. The neurons have a potential difference's between the inside and outside which is called rest potential, this potential represents constant changes because of the impulses given by the neighbor neurons [25], [45]. This potential difference's can be measured in the brain cortex using electrodes which convert the ion flow in electric flow. The characteristic of the encephalography signal (EEG) is of  $5 - 300\mu$ V in amplitude and of 0 - 150Hz in frequency [32], [38].

The EEG signals are waves similar to periodic but the waves can change from one time to other, and they have some characteristics which allow the learning [12], [19], as are the amplitude, the frequency, the morphology, the band, the rhythm, and the duration [25], [38].

The following paragraphs show the characteristics which are considered for an adult in vigilance [25], [38].

Alpha signal. Is the normal rhythm of the bottom, is the most stable and typical in the human. It is found in the frequencies of 8 - 12Hz $\pm$  1Hz. The amplitude is between 20 and  $60\mu$ V. It can be seen generally in posterior regions with more amplitude in the occipital lobes. See Figure 1. It is more evident when the patient is awake with closed eyes and in physical and mental rest, it is stopped when the eyes are opened or with the mental activity

[25], [9].

Beta signal. It is found in the frequencies > 13Hz, in general between 14 – 35Hz. The amplitude is usually low, from  $5 - 10\mu V$  and is symmetric [25], [9]. See Fig. 1.

Theta signal. It has a frequency of 4 - 8Hz; is of half of low voltage, and is found in the temporal regions [25], [9]. See Fig. 1.

Delta signal. It is found in the second and the third stages of the dream. It has a frequency of 0.5 - 3.5Hz and the amplitude is generally higher than  $75\mu$ V [25], [9]. See Fig. 1.



Figure 1: EEG signals

#### 3.2 The EOG signals

The electro-oculograms (EOG) are the signals obtained as a result of the eye movements of a patient and these EOG are detected using three electrodes, one electrode on the temple, one above and other underneath of the eye. Usually, the detected signals are by direct current (DC) coupling to specify the direction of the gaze. In the experiments of this paper, three electrodes are placed on the dominant side of the patient eye according to the optimum positions suggested by [13], [37], and [47].

Figure 2 shows the relationship between real eye movements (input) and the EOG signals (output) of the system. Denoted the upper and lower Thresholds of the vertical channel Ch.V as V1 and V2, respectively, and denote the upper and lower Thresholds of the horizontal channel Ch.H as H1 and H2, respectively. When the EOG potential exceeds one of these Thresholds, the output assumes ON, and when the EOG potential does not exceed one of these Thresholds, the output assumes OFF. The process of transforming the EOG signals from the intention of the patient is as follows [37], [47]:

1.- Output Up is when it is obtained an Up behavior, first, Threshold V1 of the vertical channel becomes ON while Threshold V2 is OFF, second, Threshold V2 of the vertical channel becomes ON while Threshold V1 becomes OFF. H1 and H2 of the horizontal channel remain OFF all the time.

2.- Output Down is when it is obtained a Down behavior, first, Threshold V2 of the vertical channel becomes ON while Threshold V1 is OFF, second, Threshold V1 of the vertical channel becomes ON while Threshold V2 becomes OFF. H1 and H2 of the horizontal channel remain OFF all the time.

Input	Logical ( Ch V	Output	
⊙≠⊙	Threshold V1	Threshold H1	Up
<b>⊙</b> ≠⊙			Down

Figure 2: EOG signals

### 4 Results

In this section, the three above detailed algorithms are applied for the learning of brain and eye signals with big data. The aforementioned signals could be applied for patient who cannot move their bodies; consequently, they could use their brains or their eyes to say what they want or need. The SAFIS of [33], SBP of [36], and MSAFIS are compared for the learning sequentially

- Brain signals: experiment 1,
- Eye signals: experiment 2.

The training of the learning phase, the parameters of the algorithms are incrementally learned as data is presented, while in the testing phase such parameters do not change and hence the algorithms can be compared in terms of performance.

The root mean square error (RMSE) of [35], [36], [38] is used to measure the performance and is expressed as:

$$RMSE = \left(\frac{1}{N}\sum_{k=1}^{N}e^{2}(k)\right)^{\frac{1}{2}}$$
(23)

where e(k) is the learning error expressed by equations (3), (12), and (16).

#### 4.1 Experiment 1

Here a real dataset of brain signals consisting of 20000 pairs (u(k), y(k)) of 20 s are used to train the training, 2000 pairs (u(k), y(k)) for 2000 s are used to test the learning. The alpha signal is obtained in this study because it has more probabilities to be found. The acquisition system is applied with a 28 years old healthy man when its eyes are closed. The inputs of all the intelligent systems are y(k), y(k+1), y(k+2), y(k+3), and the output of the intelligent systems is y(k+4).

Considering the remark 3, the parameters for the SAFIS algorithm [33] are N = 4,  $\gamma = 0.99$ , K = 2,  $\varepsilon_{\text{max}} = 1$ ,  $\varepsilon_{\text{min}} = 0.1$ ,  $e_g = 0.01$ ,  $e_p = 0.001$ , q = 0.1, p = 0.1. Considering the remark 7, the parameters of the SBP algorithm of [36] are N = 4, M = 5,  $\alpha_0 = 0.5$ . Considering the remark 8, the parameters of the MSAFIS are N = 4,  $\gamma = 0.99$ , K = 2,  $\varepsilon_{\text{max}} = 2$ ,  $\varepsilon_{\text{min}} = 0.2$ ,  $e_g = 0.05$ ,  $e_p = 0.005$ ,  $\alpha_0 = 1$ .

Figure 3 shows the comparison results for the training of learning in the three algorithms. Figure 4 introduces the illustration of the rule (neuron) evolution for the three algorithms during training. Figure 5 presents the comparison results for the testing of learning in the three algorithms. Table 1 shows the RMSE comparison results for the algorithms using (23).



Figure 3: Training for experiment 1

Table 1: Results for experiment 1

Methods	Neurons(Rules)	Training RMSE	Testing RMSE
$\operatorname{SBP}$	5	0.0043	0.0217
SAFIS	29	0.0145	0.0177
MSAFIS	3	0.0331	0.0045

From Figures 3, 4, 5, and Table 1, it can be seen that the SBP presents the smallest training RMSE, the MSAFIS presents the smallest testing RMSE, and the MSAFIS obtains the smallest number of neurons.

### 4.2 Experiment 2

Here a dataset of eye signals of the down behavior is considered where 3572 pairs (u(k), y(k)) of 3.572 s are used to train the learning, 1192 pairs (u(k), y(k)) for 1.192 s are used to test



Figure 4: Rule (neuron) evolution for experiment 1



Figure 5: Testing for experiment 1

the learning. The acquisition system is applied with a 25 years old healthy man when its eyes are moving two electrodes are used to find the signals as described in the aforementioned section. The inputs of all the intelligent systems are y(k), y(k + 1), y(k + 2), y(k + 3), and the output of the intelligent systems is y(k + 4).

Considering the remark 3, the parameters for the SAFIS [33] are N = 4,  $\gamma = 0.986$ , K = 2,  $\varepsilon_{\text{max}} = 2$ ,  $\varepsilon_{\text{min}} = 0.2$ ,  $e_g = 0.01$ ,  $e_p = 0.001$ , q = 0.1, p = 0.1. Considering the remark 7, the parameters of the SBP [36] are N = 4, M = 9,  $\alpha_0 = 0.5$ . Considering the remark 8, the parameters of the MSAFIS are N = 4,  $\gamma = 0.986$ , K = 2,  $\varepsilon_{\text{max}} = 2$ ,  $\varepsilon_{\text{min}} = 0.2$ ,  $e_g = 0.01$ ,  $e_p = 0.001$ ,  $\alpha_0 = 1$ .

Figure 6 shows the comparison results for the training of learning in the three algorithms. Figure 7 introduces the illustration of the rule (neuron) evolution for the three algorithms during training. Figure 8 presents the comparison results for the testing of learning in the three algorithms. Table 2 shows the RMSE comparison results for the algorithms using (23).



Figure 6: Training for experiment 2



Figure 7: Rule (neuron) evolution for experiment 2



Figure 8: Testing for experiment 2

Table 4: Results for experiment 2						
Methods	Neurons(Rules)	Training RMSE	Testing RMSE			
$\operatorname{SBP}$	9	0.0252	0.0290			
SAFIS	10	0.0263	0.0404			
MSAFIS	9	0.0706	0.0172			

From Figures 6, 7, 8, and Table 2, it can be seen that the SBP presents the smallest training RMSE, the MSAFIS presents the smallest testing RMSE, and the MSAFIS and SBP obtain the smallest number of neurons.

**Remark 9** The SAFIS algorithm is applied in two synthetic examples and in the Makey-Glass time series prediction problem [33]. The SBP algorithm is applied in a synthetic example and in the prediction of the loads distribution in a warehouse [36]. This study is novel because it shows that the three algorithms can be used for the learning of other different kind of systems which are the real brain and eye signals with big data.

## 5 Conclusion

This study proposed a combination of two algorithms SAFIS and SBP resulting in MSAFIS. Considering the different experiments, this new algorithm provides better compactness and higher accuracy compared to the original ones. It is worthwhile to mention, because as MSAFIS as well as SAFIS and SBP are based on online learning, they can handle big datasets of any size. They can also be applied to control, prediction, classification, and diagnosis. Here they were successfully used to learn from a challenging dataset of brain and eye signals. As a future work, the stability of the MSAFIS will be analyzed.

### 6 Acknowledgements

The authors are grateful to the editors and the reviewers for their valuable comments. The first author thanks the Secretaría de Investigación y Posgrado, Comisión de Operación y Fomento de Actividades Académicas del IPN, and Consejo Nacional de Ciencia y Tecnología for their help in this research.

## References

- Ahn C. K. (2014) A new solution to the induced l∞ finite impulse response filtering problem based on two matrix inequalities, International Journal of Control, 87, (2), 404-409.
- [2] Ahn C. K., Lim M. T. (2013) Model predictive stabilizer for T-S fuzzy recurrent multilayer neural network models with general terminal weighting matrix, Neural Computing and Applications, 23 (Suppl 1), S271-S277.
- [3] Ahn C. K. (2012) An error passivation approach to filtering for switched neural networks with noise disturbance, Neural Computing and Applications, 21 (5), 853-861.
- [4] Andreu J., Angelov P. (2013) Towards generic human activity recognition for ubiquitous applications, J. Ambient. Intell. Human Comput., 4, 155-156.
- [5] Ballini R., Yager R. R. (2014) OWA filters and forecasting models applied to electric power load time series, Evolving Systems, 5, 159-173.
- [6] Bouchachia A. (2014) Online dataprocessing, Neurocomputing, 126, 116-117.
- [7] Bouchachia A., Lena A., Vanaret C. (2014) Online and interactive self-adaptive learning of user profile using incremental evolutionary algorithms, Evolving Systems, 5, 143-157.
- [8] Bouchachia A., Vanaret C. (2014) GT2FC: An Online Growing Interval Type-2 Self-Learning Fuzzy Classifier, IEEE Transactions on Fuzzy Systems, 22, (4), 999-1018.
- [9] De Castro S., Pérez J. (2006) Manual de patología general, 6ta Edición, Masson Elsevier, Barcelona, ISBN: 978 84 458 1540 3.
- [10] Duviella E., Serir L., Sayed-Mouchaweh M. (2013) An evolving classification approach for fault diagnosis and prognosis of a wind farm, Conference on Control and Fault-Tolerant Systems (SysTol), 377-382.
- [11] Gama J., Zliobaite I., Bifet A., Pechenizkiy M., Bouchachia A. (2014), A survey on concept drift adaptation, ACM Comput. Surv., 46 (4), 44.
- [12] Gibbs F., Gibbs E. (1950) Atlas of electroencephalography, Changes whit age, asleep., Addison Wesley, Massachusetts, 1, 82-89.

- [13] Hori J., Sakano K., Saitoh Y. (2006) Development of a communication support device controlled by eye movements and voluntary eye blink, IEICE Transactions Information and Systems, E89D (6), 1790-1797.
- [14] Huang G.-B., Saratchandran P., Sundararajan N. (2004) An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, IEEE Trans. Systems Man Cybernet.-Part B: Cybernet., 34 (6), 2284-2292.
- [15] Iglesias J. A., Skrjanc I. (2014) Applications, results and future direction (EAIS 12), Evolving Systems, 5, 1-2.
- [16] Iglesias J. A., Ledezma A., Sanchis A. (2014) Evolving classification of UNIX users behaviors, Evolving Systems. DOI 10.1007/s12530-014-9104-2
- [17] Kalhor A., Araabia B. N., Lucas C. (2013) Evolving Takagi–Sugeno fuzzy model based on switching to neighboring models, Applied Soft Computing, 13, pp. 939-946.
- [18] Kasabov N. K. (2014) NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, Neural Networks, 52, 62-76,
- [19] Klass D., Daly D. (1975) Current Practice of clinical electroencephalography, Chap 5, Raven Press, New York, 69-109.
- [20] Leite D., Costa P., Gomide F. (2013) Evolving granular neural networks from fuzzy data streams, Neural Networks, 38, 1-16
- [21] Leite D., Palhares R., Campos V., Gomide F. (2014) Evolving Granular Fuzzy Model-Based Control of Nonlinear Dynamic Systems, IEEE Transactions on Fuzzy Systems. DOI 10.1109/TFUZZ.2014.2333774
- [22] Luitel B., Venayagamoorthy G. K. (2014) Cellular computational networks-A scalable architecture for learning the dynamics of large networked systems, Neural Networks, 50, 120-123.
- [23] Maciel L., Gomide F., Ballini R. (2014) Enhanced evolving participatory learning fuzzy modeling: an application for asset returns volatility forecasting, Evolving Systems, 5, 75-88.
- [24] Mackin P. D., Roy A., Wallenius J. (2011) An interactive weight space reduction procedure for nonlinear multiple objective mathematical programming, Math. Program., Ser. A , 127, 425-444.

- [25] Martínez C., Rojas B. (1998) Técnicas de Electroencefalografía, 2da edición, Secretaria de Educación Pública, Comunicaciones Científicas Mexicanas S.A. de C.V., México, ISBN: 968-7858-12-5.
- [26] Molina D., Venayagamoorthy G. K., Liang J., Harley R. G. (2013) Intelligent Local Area Signals Based Damping of Power System Oscillations Using Virtual Generators and Approximate Dynamic Programming, IEEE Transactions on Smart Grid, 4 (1), 498-508.
- [27] Ordoñez F. J., Iglesias J. A., de Toledo P., Ledezma A. (2013) Online activity recognition using evolving classifiers, Expert Systems with Applications, 40 (4), 1248-1255.
- [28] Pérez-Cruz J. H., Rubio J. J., Pacheco J., Soriano E. (2014) State estimation in MIMO nonlinear systems subject to unknown deadzones using recurrent neural networks, Neural Computing and Applications, 25, (3-4), 693-701.
- [29] Pérez-Cruz J. H., Rubio J. J., Encinas R., Balcazar R. (2014) Singularity-free neural control for the exponential trajectory tracking in multiple-input uncertain systems with unknown deadzone nonlinearities, The Scientific World Journal, 2014, 1-10.
- [30] Pratama M., Anavatti S. G., Angelov P. P., Lughofer E. (2014) PANFIS: A novel incremental learning machine, IEEE Transactions on Neural Networks and Learning Systems, 25, (1), pp. 55-68.
- [31] Pratama M., Anavatti S. G., Lughofer E. (2014) GENEFIS: Towards An Effective Localist Network, IEEE Transactions on Fuzzy Systems. DOI: 10.1109/TFUZZ.2013.2264938
- [32] Ramírez C., Hernández M., Procesamiento en tiempo real de variables Fisiológicas, Universidad Nacional de Experimental de Táchira, Decanato de Investigación, Grupo de Biomédica.
- [33] Rong H. J., Sundararajan N., Huang G. B., Saratchandran P. (2006) Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction, Fuzzy Sets and Systems, 157 (9), 1260-1275.
- [34] Roy A., Mackin P. D., Mukhopadhyay S. (2013) Methods for pattern selection, classspecific feature selection and classification for automated learning, Neural Networks 41, 113-129.

- [35] Rubio J. J. (2014) Stable and optimal controls of a proton exchange membrane fuel cell, International Journal of Control, 87 (11), 2338-2347.
- [36] Rubio J. J., Angelov P., Pacheco J. (2011) An uniformly Stable Backpropagation Algorithm to Train a Feedforward Neural Network, IEEE Transactions on Neural Networks, 22 (3), 356-366.
- [37] Rubio J. J., Ortiz F., Mariaca C. R., Tovar J. C. (2013) A method for online pattern recognition for abnormal eye movements, Neural Computing and Applications, 22 (3-4), 597-605.
- [38] Rubio J. J., Vázquez D. M., Mújica-Vargas D. (2013) Acquisition system and approximation of brain signals, IET Science, Measurement and Technology, 7 (4), 232-239.
- [39] Serdio F., Lughofer E., Pichler K., Buchegger T., Pichler M., Efendic H. (2014) Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations, Information Fusion 20, 272-291.
- [40] Schliebs S., Kasabov N. (2013), Evolving spiking neural network-a survey, Evolving Systems, 4, 87-98.
- [41] Shaker A., Lughofer E. (2014) Self-adaptive and local strategies for a smooth treatment of drifts in data streams, Evolving Systems, DOI 10.1007/s12530-014-9108-y
- [42] Torres C., Rubio J. J., Aguilar-Ibáñez C., Pérez-Cruz J. H. (2014) Stable optimal control applied to a cylindrical robotic arm, Neural Computing and Applications, 24 (3-4), 937-944.
- [43] Toubakh H., Sayed-mouchaweh M., Duviella E. (2013) Advanced Pattern Recognition Approach for Fault Diagnosis of Wind Turbines, 12th International Conference on Machine Learning and Applications, 368-373.
- [44] Trevisan J., Park J., Angelov P. P., Ahmadzai A. A., Gajjar K., Scott A. D., Carmichael P. L., Martin F. L. (2014) Measuring similarity and improving stability in biomarker identification methods applied to Fourier-transform infrared (FTIR) spectroscopy, J. Biophotonics, 7 (3–4), 254-265.
- [45] Webster J. B. (2010) Medical Instrumentation, Aplication and Design, Jonh Wiley & Sons, Inc., Fourth Edition., United States Of America.

- [46] Xu B., Huang K., King I., Liu C.-L., Sun J., Satoshi N. (2014) Graphical lasso quadratic discriminant function and its application to character recognition, Neurocomputing, 129, 33-40.
- [47] Yamagishi K., Hori J., Miyakama M. (2006) Development of EOG-Based Communication System Controlled by Eight-Directional Eye Movements, Proceedings of the 28th IEEE EMBS Annual International Conference, 2574-2577.
- [48] Yuen M.-C., King I., Leung K.-S. (2014) TaskRec: A Task Recommendation Framework in Crowdsourcing Systems, Neural Process Lett. DOI 10.1007/s11063-014-9343-z