Online Learning of Data Streams: Evolving Fuzzy Predictor with Multivariable Gaussian Participatory Learning and Recursive Weighted Total Least Squares

Fernanda P. S. Rodrigues[1], Alisson Marques da Silva[1]

[1]*Graduate Program in Mathematical and Computational Modeling*
CEFET-MG - Federal Center of Education of Minas Gerais
fernandaprodrigues@yahoo.com, alisson@cefetmg.br

Abstract. This paper introduces an evolving fuzzy system called eFTLS (evolving Fuzzy with Multivariable Gaussian Participatory Learning and Recursive Weighted Total Least Squares) constructed based on a non-supervised recursive clustering algorithm with participatory learning and multivariate Gaussian membership functions. The eFTLS uses a clustering algorithm to extract the first-order Takagi-Sugeno functional rules. The clustering algorithm can add a new cluster, delete, merge, or update existing clusters. The clusters are created using a compatibility measure and an alert mechanism. The compatibility measure is computed by Euclidian or Mahalanobis distance according to the number of samples in the cluster. An age and population based-method excludes inactive clusters. Redundant clusters are merged whenever there is a noticeable overlap between two clusters. An algorithm of recursive weighted total least squares updates the consequent parameters. The performance of the eFTLS is evaluated and compared with alternative state-of-the-art models in forecasting tasks. Computational experiments and comparisons suggest that the eFTLS perform better or are similar to alternative models.

Keywords: Evolving Systems, Adaptive Models, Fuzzy Systems, Clustering Algorithms, Forecasting.

1 Introduction

The evolving systems are intelligent computational systems capable of tackling problems in dynamic and non-stationary environments across various domains. These systems can adapt their structure and parameters concurrently as new samples are continuously fed into the data stream [1]. An evolving system adjusts its structure and parameters based on demand, tailored to process-specific characteristics and operational conditions [2]. Essentially, evolving systems are characterized by their flexible structure and capacity for incremental and continuous learning [3]. Throughout this learning process, data samples from a stream are processed just once before being discarded, thereby mitigating memory consumption [4]. Alternatively, one could posit that the learning mechanism relies solely on the current data sample.

In recent years, innovative approaches have emerged, leading to significant strides in evolving fuzzy systems, yielding practical and compelling solutions [1]. Nevertheless, there continues to be a mounting demand for developing evolving fuzzy systems, with primary prospects tied to the addition, deletion, division, and union of clusters, neurons, granules, leaves, or clouds. This adaptability ensures greater flexibility should the data dynamics change [5]. Furthermore, an ongoing challenge lies in the quest for algorithms that exhibit high precision, superior adaptability, autonomy, computational efficiency, and interpretability [1].

This study proposes a novel approach to evolving fuzzy systems for regression tasks, such as forecasting and system identification. The proposed approach is founded on a modified version of the participatory learning algorithm with multivariate Gaussian membership functions, as introduced by [6]. Unlike previous models, cluster estimation employs Euclidean and Mahalanobis distance in the proposed approach. This implies that a sample is attributed to either microcluster (Euclidean distance) or clusters (Mahalanobis distance). Using both distances aims to circumvent the singularity issue when calculating the inverse of the cluster's scatter matrix, which arises when the cluster contains a small number of samples [5]. An exclusion method for clusters has been introduced based on the concepts of age [7] and cluster population [8]. The elimination of rules is tied to the model's capacity to discard obsolete knowledge when it becomes irrelevant. A new method for merging clusters has been implemented, relying on the notable overlap between specific pairs of clusters. This approach seeks to enhance cluster merging performance. Lastly, the consequent parameters are updated by a recursive weighted total least squares algorithm.

2 Proposed Model

The eFTLS (evolving Fuzzy with Multivariable Gaussian Participatory Learning and Recursive Weighted Total Least Squares) employs an incremental learning algorithm to dynamically adapt and optimize the rules' parameters while generating an output. All calculations are performed recursively, eliminating the necessity to retain past data. The structure evolves by incorporating, combining, or excluding rules through the spatial input data organization. The consequent parameters of the rules are continuously updated using the recursive weighted total least squares algorithm.

The structure of the eFTLS comprises first-order Takagi-Sugeno functional rules, with their antecedents derived from the clusters and represented by multivariable Gaussian membership functions. The fuzzy rules are structured as follows:

$$R_i: \quad \text{If } x^t \text{ is } B_i \text{ then } y_i^t = h_{i0}^t + \sum_{j=1}^{m} h_{ij}^t x_j^t$$

where R_i is the i-th rule, i is the index of fuzzy rules and clusters, x^t is the current data sample described as $[x_1^t \cdots x_j^t \cdots x_m^t]^T$, t is the current time step, j is the index of input variables, m is the number of input variables, B_i is a multivariate Gaussian membership function with parameters derived from the center of the corresponding cluster, y_i^t is the consequent of the i-th rule (rule output), h_{i0}^t and h_{ij}^t are the parameters of the consequent of the i-th rule, and c is the number of rules and clusters. A multivariate Gaussian membership function is described by:

$$f(x) = e^{-\frac{1}{2}(x^t-\mu_i^t)(\sum_i^t)^{-1}(x^t-\mu_i^t)^T}, \tag{1}$$

in which μ is a vector containing the centers of the clusters (modal value), defined as $[\mu_1^t \ldots \mu_i^t \cdots \mu_{c^t}^t]^T$, c^t is the number of clusters and microclusters, and $\sum$ is a positive definite symmetric matrix of size $m \times m$. In the proposed approach, similar to previous versions, the number of clusters equals the number of fuzzy rules. In other words, each cluster represents a rule.

2.1 Model Initialization and Model Output

The first sample is utilized to create the initial cluster, with its center defined by the sample values. Subsequently, a rule is generated, with the antecedent's modal value serving as the cluster's center. The consequent parameters and the dispersion matrix are initialized to predefined values. The algorithm proceeds from the second sample by selecting the distance measure. The output is obtained through a weighted average of the contributions from each rule, as follows:

$$\hat{y}^t = \sum_{i=1}^{c^t} \tau_i^t y_i^t, \tag{2}$$

where y_i^t represents the consequent of the i-th rule, and τ_i^t are the normalized membership functions calculated at step t as:

$$\tau_i^t = \frac{e^{D_{(x^t,\mu_i^t)}}}{\sum_{i=1}^{c^t} e^{D_{(x^t,\mu_i^t)}}}, \tag{3}$$

in which x^t is the current data sample, μ_i^t is the center of the i-th cluster, $\sum_i^{c^t}$ is a dispersion matrix computed using the Mahalanobis distance or an identity matrix utilizing the Euclidean distance. The term $D_{(x^t,\mu_i^t)}$ can be described as:

$$D_{(x^t,\mu_i^t)} = (x^t - \mu_i^t)(\sum_i^t)^{-1}(x^t - \mu_i^t)^T. \tag{4}$$

2.2 Creating and Updating Clusters and Rules

The eFTLS structure is updated as new data are input. This process relies on a compatibility measure and an alert mechanism. The compatibility measure, denoted as γ_i^t, lies within the range [0, 1], representing the extent of compatibility between the new sample and each cluster. In simpler terms, this measure computes the compatibility between the current sample x^t and the model's existing knowledge, represented by the centers of existing clusters [6]:

$$\gamma_i^t = f(x^t, \mu_i^t) = e^{-\frac{1}{2}D(x^t, \mu_i^t)}, \tag{5}$$

in which $D(x^t, \mu_i^t)$ is the distance metric (Euclidean or Mahalanobis) between the sample and the center of cluster i at time t. The cluster structure is adjusted based on the number of samples in each cluster, with a maximum limit set by the user as N_{max}.

If the number of samples in cluster i, denoted as n_i, is less than N_{max}, Euclidean distances are calculated using an identity matrix $\sum_i^t$ of size $m \times m$, where m is the dimensionality of the data. Otherwise, Mahalanobis distance is computed, utilizing the covariance matrix $\sum_i^t$ specific to cluster i at time t [5]. The compatibility measure threshold Γ_{max} for Euclidean distance calculations is determined by $\Gamma_{max} = e^{-\min_i(\frac{F}{2\omega})^2}$, where $F = \max(x_i) - \min(x_i)$ and ω represents the window size. For Mahalanobis distance, $\Gamma_{max} = e^{-\frac{1}{2}\chi^2_{m,\alpha}}$ in which $\chi^2_{m,\alpha}$ follows a Chi-squared distribution with m degrees of freedom and α as a one-sided confidence interval.

The alert mechanism indicates when the cluster structure inadequately represents the current system knowledge, requiring revision [6]. The arousal index, $a_i^t \in$ [0,1], is obtained for each cluster as new samples are input. To compute the arousal index, it is necessary to determine the significance level α and the count of threshold violations. The value of α can be automatically calculated, utilizing the window size ω, as demonstrated in [6]:

$$\alpha = \begin{cases} 0.01, & \text{if } \omega \geq 100 \\ 0.05, & \text{if } 20 \leq \omega < 100 \\ 0.1, & \text{if } 10 \leq \omega < 20. \end{cases} \tag{6}$$

The value z_i^t represents the number of threshold violations and is computed as:

$$z_i^t = \begin{cases} \sum_{k=0}^{\omega-1} r_i^{t-k}, & t > \omega \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where r_i^t is defined by:

$$r^t i = \begin{cases} 0, & D(x^t, \mu_i^t) < \chi^2_{m,\alpha} \text{ (for Mahalanobis distance)} \\ 0, & D_{(x^t, \mu_i^t)} < \min_i \left(\frac{\max(x_i) - \min(x_i)}{2\omega} \right)^2 \text{ (for Euclidean distance)} \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

The cumulative probability, denoted as V^t, is employed to estimate the arousal index, represented as $a_i^t \in [0, 1]$, where $a_i^t = p(V^t < z)$, and $p(V^t = z)$ follows a binomial distribution. The probability function of $p(V^t = z)$ is defined as per [6]:

$$p(V^t = z) = \begin{cases} \frac{w!}{z!(w-z)!} \alpha^z (1-\alpha)^{\omega - z}, & z = 0, ..., \omega \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

A new cluster is established if the compatibility measures γ_i^t for the current sample is lower than its respective threshold Γ_{max} for all clusters. The alert index a_i^t of the cluster with the highest compatibility (i) surpasses its respective threshold Γ_a. Specifically, if $\gamma_i^t < \Gamma_{max}$ for all $i = 1, ..., c^t$ and $a_i^t > \Gamma_a$ for $i = \max_i(\gamma_i^t)$, a new cluster is created.

The current number of clusters, denoted as c^t, is updated after creation. The number of samples in the newly formed cluster is initialized to $n_{c^t}^t = 1$, and the center of the new cluster is set as $\mu_{c^t}^t = x^t$. The dispersion matrix for this cluster is initialized as $\sum_{c^t}^t = \sum_{init}$. Subsequently, the consequent parameters $h_{c^t}^t$ are determined, as illustrated by eq. (10), and a new rule is established.

$$h_{c^t}^t = \frac{\sum_{i=1}^{c^t} h_i^t \gamma_i^t}{\sum_{i=1}^{c^t} \gamma_i^t}. \tag{10}$$

Alternatively, if $\gamma_i^t > \Gamma_{max}$ for all $i = 1, \ldots, c^t$ or $a_i^t < \Gamma_a$ for $i = \max_i(\gamma_i^t)$, then the sample is assigned to the most compatible cluster. The center of this cluster is updated using the following formula:

$$\mu_i^{t+1} = \mu_i^t + S_i^t(x^t - \mu_i^t), \tag{11}$$

Here, $S_{i^*}^t$ is computed as:

$$S_i^t = \lambda(\gamma_i^t)^{1-a_{i^*}^t}, \tag{12}$$

where λ represents the learning rate. The value of λ is usually defined between 10^{-1} and 10^{-5} [6].

2.3 Merging of the Clusters and Rules

In the proposed approach, clusters are merged whenever a notable overlap between two clusters is detected [9]. More formally, if two clusters i and i are sufficiently close, meaning that their centers, μ_i and μ_i, satisfy the condition:

$$||\mu_{i^*}^t - \mu_i^t|| \leq \rho \ \ (i = 1...c^t \ e \ i \neq i^*), \tag{13}$$

in which ρ is the threshold for cluster merging. The center of the new cluster $\mu_{i^* \cup i}^t$ is determined using a weighted average method and calculated as:

$$\mu_{i^* \cup i}^t = \mu_i^t - \frac{n_i^t}{n_i^t + n_i^t}(\mu_{i^*}^t - \mu_i^t), \tag{14}$$

where $n^t i$ represents the number of samples in the i-th cluster, and the same applies to $n^t i$. The new cluster $i \cup i$ is positioned between clusters i^* and i, its exact location depending on the sample count of the merged clusters. The dispersion matrix for the resulting cluster is updated and defined as:

$$\sum_{i^* \cup i}^t = \frac{\sum_{i^*}^t + \sum_i^t}{2}. \tag{15}$$

The consequent parameters for the newly generated rule are obtained by $h_{i^* \cup i}^t = \frac{h_i^t \gamma_i^t + h_i^t \gamma^t i}{\gamma_{i^*}^t + \gamma_i^t}$. The sample count of the new cluster is the sum of the sample counts of the two merged clusters, i.e., $n_{i^* \cup i}^t = n_i^t + n_{i^*}^t$. Finally, the cluster count and indices are updated.

2.4 Exclusion of the Clusters and Rules

The proposed approach for cluster elimination is founded on the concepts of age and population [7, 8]. In this work, age determines the time interval a cluster remains inactive, meaning its membership degree is zero. The age of a cluster is calculated by $age_i^t = t - A_i$, where i is the index of the cluster, A_i is the time instance when the i-th cluster was last activated, and t represents the current time step. For each sample x^t, the index of the oldest inactive cluster $\bar{i}$ is found. The oldest inactive cluster is eliminated if $age_{\bar{i}}^t > \omega$, where $age_{\bar{i}}^t$ denotes the period of inactivity for the cluster indexed by $\bar{i}$, and ω is the window size.

Another method employed for cluster elimination is based on the population, which denotes the number of samples n_i^t assigned to a cluster [8]. The population of a cluster is monitored, and if it falls below 1% of the total samples at time t, the cluster is eliminated. Mathematically, a cluster indexed by $\bar{i}$ is removed if $n_{\bar{i}}^t/t < 0.01$.

For each new sample x^t, the index of the oldest inactive cluster $\bar{i}$ is identified to eliminate a cluster. The cluster indexed by $\bar{i}$ is eliminated if the following condition holds $age_{\bar{i}}^t > \omega$ and $n_{\bar{i}}^t/t < 0.01$.

Following removing a cluster, the cluster index and count are updated. Combining these two mechanisms ensures that newly formed clusters are not prematurely removed.

2.5 Update of Consequent Parameters

The eFTLS uses the Recursive Weighted Total Least Squares (RWTLS) algorithm to update the consequent parameters [10]. The commonly used parameter update algorithms optimize the error concerning the output. However, real-world data often contains input noise. Therefore, RWTS is proposed to mitigate these anomalies, aiming to obtain unbiased parameter estimates in the presence of noisy inputs and outputs [11]. The formula for parameter updating is given by:

$$h_i^t = \frac{g_i^t u_i^t - (u_{i1}^t x_1^t + ... + u_{im}^t x_m^t)}{u_{i0}^t}, \tag{16}$$

in which m is the dimension of the input space, u_i^t is the smallest eigenvector of the dispersion matrix, g_i^t is a point through which the regression model passes, and $g_i^t u_i^t$ is obtained as $g_i^t u_i^t = u_{i0}^t y^t + u_{i1}^t x_1^t + ... + u_{im}^t x_m^t$.

However, before that, the regressor vector $\hat{r}_i^t$ and the weighted mean vector $\nu_{r_i}^t$ of the output and inputs of the current sample are updated. Specifically,

$$\hat{r}_i^t = \overline{r}_i^t - \nu_{r_i}^t, \text{ in which } \nu_{r_i}^t = \frac{(\overline{r}_i^t)^T \tau_i^t}{\mathbf{1}^T \tau_i^t}, \tag{17}$$

where τ_i^t represents the normalized membership functions, and $\overline{r}_i^t$ is obtained as:

$$\overline{r_i}^t = [y^t | x^t]. \tag{18}$$

Next, the inverse of the weighted Hessian matrix P_i is updated by:

$$P_i^t = I - L^{t-1} (\hat{r}_i^t)^T P_i^{t-1}, \tag{19}$$

in which L_i^{t-1} is obtained by:

$$L_i^{t-1} = \frac{P_i^{t-1} \hat{r}_i^t}{\frac{1}{\tau_i^t} + (\hat{r}_i^t)^T P_i^{t-1} \hat{r}_i^t}. \tag{20}$$

The unit normal vector to the affine hyperplane of rule i is then computed as:

$$u_i^t = \sqrt{Q_i} \left(\overline{r}_i^t - \hat{r}_i^t, \right), \tag{21}$$

$$u_i^t = \frac{u_i^t}{||u_i^t||_2} \tag{22}$$

in which Q_i is the weighting diagonal matrix of cluster i. The consequent parameters are initialized as $h_i^0 = [y^0 \ 0 \ ... \ 0]$ and I is an identity matrix of size $m + 1$ x $m + 1$.

3 Computational Experiments

In this section, the eFTLS is evaluated in forecasting problems. The results obtained from eFTLS are compared with three alternative evolving systems: eFCE [3], eMG [6], and eOGS [12], with code provided in Matlab by their respective authors. The proposed model itself was developed using Matlab.

The datasets are split into two subsets with 50% of the samples each. The first subset is used to find the best values of parameters, whereas the second is used for performance evaluation. The best parameters were obtained through an exhaustive search. The parameters that achieved the lowest forecasting error in the first subset are used to evaluate the models' performance in the second subset. Table 1 shows the range of parameters and the best values. All data has been normalized to the [0,1] interval for all experiments.

The performance of the models is assessed for all samples in the dataset using the Root Mean Square Error (RMSE) and the Non-Dimensional Error Index (NDEI).

3.1 Death Valley

This section evaluates models predicting average temperature in Death Valley[1]. The meteorological dataset from Death Valley comprises 1306 observations, recording monthly average temperatures from 1901 to 2009, measured in degrees Celsius. The objective is to forecast the monthly average temperature one step ahead. [13] and [14] suggest using the first twelve lagged values of the series as inputs. The model for this dataset is defined as follows: $\hat{y}^t = f(y^{t-1}, ..., y^{t-11}, y^{t-12})$.

[1] https://www.nps.gov/deva/planyourvisit/weather.htm

Table 1. User-defined parameters used in the experiments.

Models	Parameter	Step	Sec. 3.1	Sec. Sec.3.2
eFCE	$\lambda = 0.05$	(fixo)	0.05	0.05
	$w = 5, ..., 55$	5.00	5	50
	$\sum_{init} = (10^{-1}, ..., 10^{-4}).I_m$	1.00	$10^{-1}.I_{12}$	$10^{-2}.I_9$
	$N_{max} = 0, ..., 78$	1.00	5	45
	$\rho = 0.10, ..., 0.40$	0.15	0.25	0.25
eFTLS	$\lambda = 0.05$	(fixo)	0.05	0.05
	$w = 5, ..., 55$	5.00	10	5
	$\sum_{init} = (10^{-1}, ..., 10^{-4}).I_m$	1.00	$10^{-1}.I_{12}$	$10^{-1}.I_9$
	$N_{max} = 0, ..., 78$	1.00	7	11
	$\rho = 0.10, ..., 0.40$	0.15	0.10	0.25
eMG	$\lambda = 0.05$	(fixo)	0.05	0.05
	$w = 5, ..., 55$	5.00	5	10
	$\sum_{init} = (10^{-1}, ..., 10^{-4}).I_m$	1.00	$10^{-1}.I_{12}$	$10^{-1}.I_9$
	$\alpha = 0.01$	(fixo)	0.01	0.01
eOGS	$\psi = 2$	(fixo)	2	2
	$\alpha = 0.1, ..., 0.9$	0.10	0.9	0.1
	$v = 50, ..., 2950$	100	250	150
	$\omega = 0.01$	(fixed)	0.01	0.01

Table 2 illustrates the results obtained in temperature prediction for Death Valley. The best performance in terms of RMSE and NDEI was achieved by eFCE, followed by eMG, eFTLS, and eOGS. The results of the eFCE, eMG, and eFTLS models are comparable.

Table 2. Performance in predicting Death Valley.

Models	**RMSE**	**NDEI**
eFCE	0.0508	0.2164
eMG	0.0596	0.2307
eFTLS	0.0597	0.2389
eOGS	0.0785	0.3056

3.2 Bicycle Rental

This section addresses the bicycle rental prediction problem. In bike-sharing systems, users can rent and return bicycles at various locations within the city. The Capital Bike Sharing dataset contains 731 samples, representing information collected over two years. These bike-sharing lending system data were collected in Washington, D.C.. The objective is to forecast the number of rented bicycles using 9 input variables: station (x_1^t), month (x_2^t), holiday (x_3^t), day of the week (x_4^t), weather condition (x_5^t), temperature (x_6^t), apparent temperature (x_7^t), humidity (x_8^t), and wind speed (x_9^t) [14]. The model for this dataset is described as: $\hat{y}^t = f(x_1^t, x_2^t, ..., x_8^t, x_9^t)$.

The RMSE and NDEI results obtained in the bicycle rental prediction are displayed in Table 3. The eFTLS achieved the best performance, followed by eMG, eFCE, and eOGS.

4 Conclusion

This manuscript introduced a novel approach for constructing evolving fuzzy models based on a participatory clustering algorithm and multivariate Gaussian membership functions. The structure of the proposed method evolves through the inclusion, exclusion, merging, and updating of clusters and rules. Creating a new cluster/rule employs a similarity measure based on Euclidean distance for microclusters and Mahalanobis distance for clusters. Using these two distances resolves the issue of calculating the inverse of the scatter matrix for clusters with few

Table 3. Performance in predicting bicycle rental.

Models	RMSE	NDEI
eFTLS	0.1068	0.4794
eMG	0.1096	0.4916
eFCE	0.1284	0.5761
eOGS	0.1376	0.5936

samples. The exclusion of clusters and rules draws inspiration from the concepts of age and population, enabling the removal of inactive or poorly represented clusters. The merging mechanism is based on the notable overlap of two clusters. The recursive weighted total least squares algorithm updates the consequent parameters.

The performance of the proposed model was evaluated and compared to state-of-the-art evolving models using forecasting tasks. Computational results indicate that the proposed models exhibit superior or comparable performance compared to alternative models.

Future work shall address new algorithms to adjust the consequent parameters and generalization of the model for problems with multiple outputs.

References

[1] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Information Sciences*, vol. 490, pp. 344–368, 2019.

[2] A. Silva, W. Caminhas, A. Lemos, and F. Gomide. A fast learning algorithm for evolving neo-fuzzy neuron. *Applied Soft Computing*, vol. 14, pp. 194–209, 2014.

[3] F. P. S. Rodrigues, A. M. Silva, and A. P. Lemos. Evolving fuzzy system with multivariable gaussian participatory learning and recursive maximum correntropy - efce. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, volume 1, pp. 1–6, Luxembourg, Luxembourg. IEEE, 2021.

[4] E. Lughofer. On-line assurance of interpretability criteria in evolving fuzzy systems - Achievements, new concepts and open issues. *Information Sciences*, vol. 251, pp. 22–46, 2013.

[5] I. Škrjanc. Cluster-volume based merging concept for incrementally evolving fuzzy Gaussian clustering - eGAUSS+. *IEEE Transactions on Fuzzy Systems*, vol. 28, n. 9, pp. 2222–2231, 2020.

[6] A. Lemos, W. Caminhas, and F. Gomide. Multivariable Gaussian Evolving Fuzzy Modeling System. *IEEE Transactions on Fuzzy Systems*, vol. 19, n. 1, pp. 91–104, 2011.

[7] E. Lughofer and P. Angelov. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing*, vol. 11, n. 2, pp. 2057–2068, 2011.

[8] P. Angelov and D. Filev. Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In *Proceedings of the 14th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE '05*, pp. 1068–1073, Reno, NV, USA. IEEE, 2005.

[9] C. Aguiar and D. Leite. Unsurpervised Fuzzy eIX:Evolving Internal-eXternal Fuzzy Clustering. In *Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1–8, Bari, Italy. IEEE, 2020.

[10] C. Davila. An efficient recursive total least squares algorithm for fir adaptive filtering. *IEEE Transactions on Signal Processing*, vol. 42, n. 2, pp. 268–280, 1994.

[11] E. Lughofer. Variants of Recursive Consequent Parameters Learning in Evolving Neuro-Fuzzy Systems. In *Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1–10, Bari, Italy. IEEE, 2020.

[12] D. Leite and I. Škrjanc. Ensemble of Evolving Optimal Granular Experts, OWA Aggregation, and Time Series Prediction. *Information Sciences*, vol. 504, pp. 95–112, 2019.

[13] D. Leite, P. Costa, and F. Gomide. Interval Approach for Evolving Granular System Modeling. In M. Sayed-Mouchaweh and E. Lughofer, eds, *Learning in Non-Stationary Environments*, pp. 271–300. Springer, New York, NY, 2012.

[14] C. Garcia, D. Leite, and I. Škrjanc. Incremental missing-data imputation for evolving fuzzy granular prediction. *IEEE Transactions on Fuzzy Systems*, vol. 28, n. 10, pp. 2348–2362, 2020.