

Stability of Evolving Fuzzy Systems based on Data Clouds

Hai-Jun Rong, *Member, IEEE*, Plamen Angelov, *Fellow, IEEE*,
Xiaowei Gu, *Student Member, IEEE*, and Jian-Ming Bai

Abstract—Evolving fuzzy systems (EFSs) are now well developed and widely used thanks to their ability to self-adapt both their structures and parameters online. Since the concept was firstly introduced two decades ago, many different types of EFSs have been successfully implemented. However, there are only very few works considering the stability of the EFSs, and these studies were limited to certain types of membership functions with specifically pre-defined parameters, which largely increases the complexity of the learning process. At the same time, stability analysis is of paramount importance for control applications and provides the theoretical guarantees for the convergence of the learning algorithms. In this paper, we introduce the stability proof of a class of EFSs based on data clouds, which are grounded at the AnYa type fuzzy systems and the recently introduced empirical data analysis (EDA) methodological framework. By employing data clouds, the class of EFSs of AnYa type considered in this work avoids the traditional way of defining membership functions for each input variable in an explicit manner and its learning process is entirely data-driven. The stability of the considered EFS of AnYa type is proven through the Lyapunov theory, and the proof of stability shows that the average identification error converges to a small neighborhood of zero. Although, the stability proof presented in this paper is specially elaborated for the considered EFS, it is also applicable to general EFSs. The proposed method is illustrated with Box-Jenkins gas furnace problem, one nonlinear system identification problem, Mackey-Glass time series prediction problem, eight real-world benchmark regression problems as well as a high frequency trading prediction problem. Compared with other EFSs, the numerical examples show that the considered EFS in this paper provides guaranteed stability as well as a better approximation accuracy.

Index Terms—Evolving Fuzzy Systems, Data Clouds, AnYa type fuzzy systems, Stability

I. INTRODUCTION

Consisting of a number of IF-THEN rules, fuzzy inference systems (FISs) can deal with ill-defined and uncertain problems without precise quantitative analysis [1]. They have been applied to a wide range of areas [1]–[3]. Conventional FISs are typically built by domain experts and lack the learning capability [2], [3]. Thus, it is likely for these FISs to suffer from poor performance if inappropriate fuzzy rules are introduced. Some optimization techniques, such as back-propagation [1] or genetic algorithm [4], [5] were further

applied to ensure the learning capability of the fuzzy systems. These works involve structure identification and parameter adjustment of the fuzzy systems. The structure identification of FISs aims to determine the number of fuzzy IF-THEN rules while the parameter adjustment realizes the modification of the antecedent parameters of the “IF” part and consequent parameters of the “THEN” part.

These methods, however, need to assume that all the data is available at the beginning of the training process and their training involves cycling/iterations over a number of epochs. Therefore, they are suitable for offline applications since it is hard to guarantee that these pre-trained systems have a satisfactory performance for online applications with nonstationary and changeable behavior over time. To tackle this problem, in 21st century, intensive research works have been concentrated on developing evolving fuzzy systems (EFSs) [6]–[22] aiming at adapting system structure and parameters simultaneously online to capture the dynamical changes of data patterns. These include evolving rule-based models [6], dynamic evolving neuro fuzzy inference system (DENFIS) [7], a family of evolving Takagi-Sugeno (eTS) models [8]–[11], a meta-cognitive neuro-fuzzy inference system (McFIS) [12], sequential adaptive fuzzy inference system (SAFIS) [13], extended SAFIS (ESAFIS) [14], evolving possibilistic fuzzy modeling approach (ePFM) [15], multivariable Gaussian evolving fuzzy modeling system (eMG) [16], evolving fuzzy model (eFuMo) [17], incremental learning machine (PANFIS) [18], effective localist network (GENEFIS) [19], flexible fuzzy inference systems (FLEXFIS) [20], sparse fuzzy inference systems (SparseFIS) [21] and generalized smart evolving fuzzy systems (Gen-Smart-EFS) [22], etc. All these algorithms require specified types of membership function to calculate the firing strength of each rule. Also, these algorithms propose different rule evolving schemes, such as density/potential criterion [6]–[11], distance criterion [16], [17], [20]–[22], error criterion [12]–[14], statistical contribution of the rule [13], [14], [18], [19]. To achieve better performance, some works further update the parameters of the membership functions using some optimization techniques, such as extended Kalman filter [12], [13]. However, these optimization techniques largely increase the computation complexity.

Although, the algorithms mentioned above have their successful applications, they lack a thorough stability analysis. In real applications, the stability is very important. Some identification algorithms, such as gradient descent and least square methods are stable in ideal conditions [23]. However, in the cases of unmodeled dynamics, they may become unstable.

Hai-Jun Rong and Jian-Ming Bai are with State Key Laboratory for Strength and Vibration of Mechanical Structures, Shaanxi Key Laboratory of Environment and Control for Flight Vehicle, School of Aerospace, Xi’an Jiaotong University, Shaanxi, China 710049 (e-mail:hjrong@mail.xjtu.edu.cn). Plamen Angelov and Xiaowei Gu are with the School of Computing and Communications, Lancaster University, Lancaster, UK, LA1 4WA (e-mail:p.angelov@lancaster.ac.uk). Plamen Angelov also holds an Honorary Professor title with Technical University, Sofia, Bulgaria.

Such unmodeled dynamics are caused by non-linearities, non-Gaussian distributions that are much more likely to present in real problems. Some robust modification algorithms [23] are proposed to assure the stability of FISs in dealing with uncertainties. However, in these work, the fuzzy structure includes explicit membership functions which need to be fixed in advance and cannot self-evolve during training. As one of the very few examples of exiting works on stability of EFSs, an online self-organizing fuzzy modified least square (SOFMLS) network [24] was recently proposed. In this algorithm, structure and parameters learning are conducted at the same time. The proposed network considers the stability analysis, but it uses unidimensional membership functions for each rule and a modified least-square algorithm to train the antecedent and consequent parameters of the system. As another example, in [25], a growing-and-pruning fuzzy neural network (GPFNN) with concurrent structure learning and parameter training is proposed, which discusses the convergence of the learning process. However, it relies on a supervised gradient decent method to update the center and width vectors of the Gaussian membership functions as well as the corresponding consequent parameters. In these works, the determination of membership function and the adjustment of antecedent parameters leads to a more complicated computation process. In both works, which are the only works on stability of EFSs we identified, explicit membership functions are considered.

Recently, a simplified type of fuzzy rule-based system was introduced, called AnYa [26], which offers a new way of defining the “IF” part of the rules without defining the membership functions per variable in an explicit manner. The antecedent parts of fuzzy rules are formed upon the *data clouds*, which are the sets of data samples attracted around focal points. The *data clouds* in the AnYa system can be formed “from scratch”. The self-evolving mechanism of the *data clouds* is based on determination of the focal points. In the original paper, introducing AnYa [26], focal points were identified using eClustering [27] algorithm. The importance of this approach is that AnYa type of fuzzy systems can be considered as the third alternative type of fuzzy systems to Mamdani and Takagi-Sugeno type models (both of which share same type of antecedent/IF part) with a different, explicit membership function free antecedent/IF part [27].

In this paper, we introduce a stability proof for the class of EFSs, which are based on *data clouds*. The systems considered in this paper differ from the one in [26] as they form *data clouds* based on the empirical data analytics (EDA) [28] computational framework and empirical fuzzy sets (ε FSSs) [29]. EDA is a nonparametric, assumptions-free, entirely data-driven methodological framework [28] which is entirely based on the empirical observations and the ensemble properties of the data. It is close to statistical learning in its nature but is free from the assumptions required by traditional probability theory and statistical learning methods [30]. The stability of the EFS based on *data clouds* considered in this paper is proven through the Lyapunov theory and the proof of stability shows that the average identification error converges to a small neighborhood of zero and the parameter error is bounded by the initial parameter error. Although, the stability

proof presented in this paper specially refers to the EFSs based on *data clouds*, it is applicable to some EFSs that are comprised of the structure learning of the ‘IF’ part and the fuzzily weighted recursive least square parameter update of the ‘THEN’ part as well.

The rest of the paper is organized as follows. The EFS based on *data clouds* is introduced in Section II. The learning process of the considered EFS including the formation of the *data clouds* according to EDA and the parameter learning is described in Section III. Section IV gives the stability and convergence analysis of the EFS based on data clouds. Section V presents the performance evaluation of the proposed system. This paper is concluded by Section VI.

II. EFS BASED ON DATA CLOUDS

Let’s consider a nonlinear system in a discrete time framework expressed by:

$$y_k = f(\mathbf{x}_k) \quad (1)$$

where y_k denotes the system output, $\mathbf{x}_k = [x_{k1}, \dots, x_{kn}]^T$ is the input vector of the system, $f(\cdot)$ is an unknown nonlinear function. Subscript $k \in \mathbb{N}$ is the discrete time index.

In practical cases, the exact analytical expression of the function $f(\cdot)$ is hard to identify due to nonlinear and dynamic characteristics of the system. Thus, the system (1) considered in the study is unknown. To achieve the estimation of the system output y_k , a fuzzy system is used to approximate the function $f(\cdot)$ as follows

$$\hat{y}_k = \hat{f}(\mathbf{x}_k) \quad (2)$$

where \hat{y}_k is the output of the fuzzy system.

In this study, we consider the EFS based on *data clouds* to approximate $f(\cdot)$ with $\hat{f}(\cdot)$. The concept of this class of EFSs we consider is grounded at the recently introduced AnYa type fuzzy system [26].

An AnYa type fuzzy rule has the following form:

Rule i : IF ($\mathbf{x}_k \sim \gamma_k^i$) THEN (\hat{y}_k is a_k^i)

where γ_k^i is the focal point of the i th *data cloud* in the input space. a_k^i ($i = 1, 2, \dots, N$) represents the crisp consequence of the i th rule that can be a constant or a linear combination of input variables. In this study, a linear consequence $a_k^i = q_{k0}^i + q_{k1}^i x_{k1} + \dots + q_{kn}^i x_{kn}$ is used. N represents the number of *data clouds*.

The structure of the EFS we consider is depicted in Fig. 1(a) and the structure of a traditional TS fuzzy system is given in Fig. 1(b) for comparison. One can see the difference between the two figures. The considered EFS based on *data clouds* consists of four layers. Layer 1 extracts the local density of each *data cloud*. The value of the local density is normalized in layer 2. Layer 3 is for applying the weighted average defuzzification. The approximated output is represented in layer 4.

It is well-known that complex systems can be decomposed into a number of loosely connected local simpler systems aggregated in a fuzzy way. In the EFS that we consider, each local sub-system is represented by a *data cloud* that describes a certain sub-set of the entire data set. Thus, the proposed

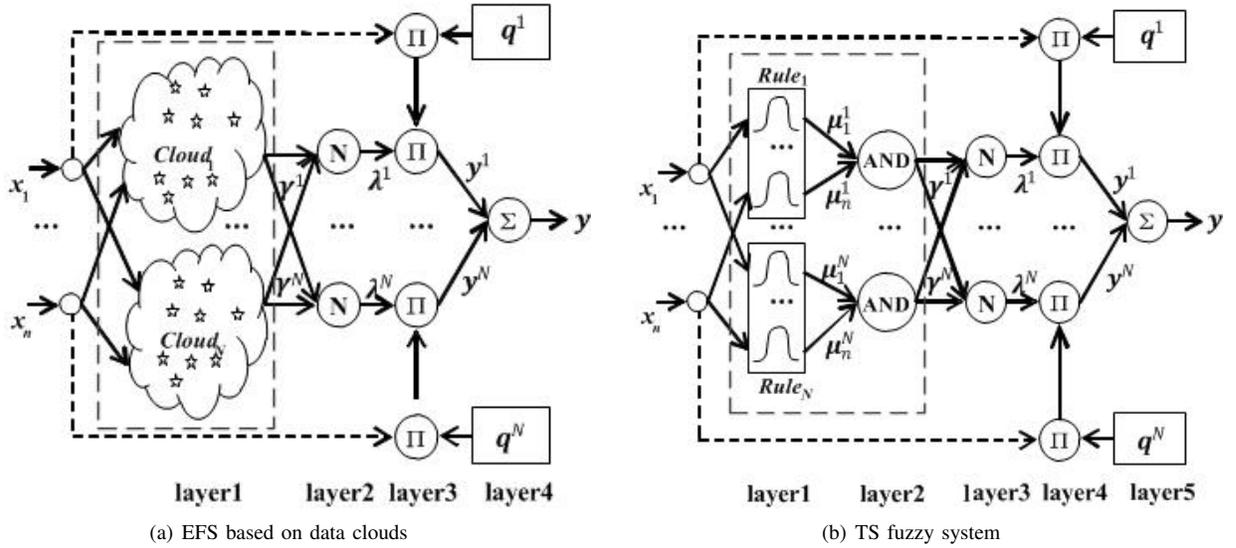


Fig. 1. Structure of EFS based on data clouds (AnYa type) and traditional TS fuzzy system

approach replaces the scalar membership functions with a non-parametric function that is represented by the local densities. Considering Euclidean type of distance, in this paper, the local density of the i th data cloud is defined as follows [28]:

$$\gamma_k^i = \frac{1}{1 + \frac{\|\mathbf{x}_k - \Gamma_k^i\|^2}{\Xi_k^i - \|\Gamma_k^i\|^2}} \quad (3)$$

Other types of distances can also be considered in principle, but without limiting the concept, we consider the Euclidean type of distance due to its simplicity. Here Γ_k^i and Ξ_k^i are the mean and the scalar product of the data samples within the i th data cloud. They can be updated recursively by [27]:

$$\begin{aligned} \Gamma_k^i &= \frac{M^i - 1}{M^i} \Gamma_{k-1}^i + \frac{1}{M^i} \mathbf{x}_k, & \Gamma_1^i &= \mathbf{x}_1 \\ \Xi_k^i &= \frac{M^i - 1}{M^i} \Xi_{k-1}^i + \frac{1}{M^i} \|\mathbf{x}_k\|^2, & \Xi_1^i &= \|\mathbf{x}_1\|^2 \end{aligned} \quad (4)$$

M^i denotes the number of data samples within the i th data cloud.

Assuming there are N data clouds, the normalization value of the local density for each data cloud is expressed as:

$$\lambda_k^i = \frac{\gamma_k^i}{\sum_{i=1}^N \gamma_k^i}, i = 1, 2, \dots, N \quad (5)$$

With the weighted average defuzzification, the output of the EFS that we consider is equal to

$$\hat{y}_k = \sum_{i=1}^N \lambda_k^i a_k^i; a_k^i = \mathbf{x}_e^T \mathbf{q}_k^i \quad (6)$$

where $\mathbf{x}_e = [1, \mathbf{x}_k^T]^T \in \mathbb{R}^{(n+1) \times 1}$ is the extended input vector by appending the input vector \mathbf{x}_k with 1; \mathbf{q}_k^i is the vector representing the consequent parameters of the k th data cloud and given by

$$\mathbf{q}_k^i = [q_{k0}^i, \dots, q_{kn}^i]^T \in \mathbb{R}^{(n+1) \times 1} \quad (7)$$

The output \hat{y}_k in Eq.(6) is further rewritten in a compact matrix form as,

$$\hat{y}_k = \mathbf{H}_k^T \mathbf{Q}_k \quad (8)$$

where \mathbf{Q}_k is the consequent parameter vector of all N data clouds and \mathbf{H}_k is the inputs weighted by the normalized local density vector of all data clouds. They are reformulated as,

$$\begin{aligned} \mathbf{Q}_k &= [\mathbf{q}_k^1, \dots, \mathbf{q}_k^N]^T_{N(n+1) \times 1} \\ \mathbf{H}_k &= [\mathbf{x}_e^T \lambda_k^1, \dots, \mathbf{x}_e^T \lambda_k^N]^T_{N(n+1) \times 1} \end{aligned} \quad (9)$$

Remark 1: In this EFS, the antecedent part of the fuzzy system is represented by the data clouds, and the membership degree to a particular data cloud is measured by the normalized local density of the current data sample as shown in Eq. (5). This is different from the traditional fuzzy systems where the membership degree to a certain rule is determined by the normalized firing strength computed from specifically predefined membership functions based on sum-product composition. Indeed, the EFS considered in this paper does not require membership functions (also see the difference in the dashed line boxes in Figs. 1(a) and 1(b)).

III. LEARNING OF EFS BASED ON DATA CLOUDS

A. Formation of Data Clouds

In this study, the number of data clouds, N are determined automatically during the learning process. This enables the EFS based on data clouds to have a self-evolvable structure and to be independent from the prior knowledge about the system model. The global density defined within the EDA framework [28] is applied here as the mechanism to evolve the structure of the EFS in an online manner. Similar to the centers of fuzzy sets, each data cloud has one focal point ξ with the highest local density. For the first data sample, it is selected as the focal point of the first data cloud, that is $\xi_1 = \mathbf{x}_1$. Then, with more data samples arriving, data clouds are formed based on the global densities that are calculated at the focal points of the existing data clouds. For the i th data

cloud, its global density at the k th time instance according to EDA [28] is expressed as :

$$\gamma_k^{i(G)} = \frac{1}{1 + \frac{\|\Gamma_k^i - \Gamma_k^G\|^2}{\Xi_k^G - \|\Gamma_k^G\|^2}} \quad (10)$$

where Γ_k^G and Ξ_k^G are the global mean and the global scalar product of the observed data samples. The global density $\gamma_k^{i(G)}$ is similar to the local density, but Γ_k^G and Ξ_k^G consider all the samples including \mathbf{x}_k at the current time instant, k and all the previously observed samples $\mathbf{x}_j, j = 1, 2, \dots, k-1$.

When the new data sample appears, the global densities of all the existing *data clouds* are influenced and are updated accordingly. The global density produced by the new observation \mathbf{x}_k according to EDA [28] is given as,

$$\Upsilon_k = \frac{1}{1 + \frac{\|\mathbf{x}_k - \Gamma_k^G\|^2}{\Xi_k^G - \|\Gamma_k^G\|^2}} \quad (11)$$

The global density of the new data sample is compared to the updated global densities of the existing focal points to judge whether a new *data cloud* needs to be formed. If the global density of the new observation is higher than the global densities of the existing focal points, the new observation is more descriptive and has a stronger summarization power than all the existing focal points. In this case, the new data sample may initialize a new *data cloud*. On the other hand, if the new data \mathbf{x}_k is sufficiently far from the existing *data clouds*, it has a smaller global density. However, the new data sample is possible to represent a new operating regime and the new data pattern. In this case, the new data sample may be accepted as a new *data cloud*, even though its global density is lower than the global densities of the existing focal points. The first case is described as follows:

$$(\Upsilon_k - \gamma_k^{i(G)} > 0) \text{ OR } (\Upsilon_k - \gamma_k^{i(G)} < 0); \forall i \in 1, \dots, N \quad (12)$$

Besides the global density, the new data \mathbf{x}_k is required to be sufficiently far from the focal points of the existing *data clouds*. The second case is described as:

$$\zeta_{ki} > \rho_k^i; \quad \forall i \in 1, \dots, N \quad (13)$$

where ζ_{ki} denotes the distance between the current input data \mathbf{x}_k and the focal point of the i th *data clouds* ξ_i , that is $\|\mathbf{x}_k - \xi_i\|$. ρ_k^i represents the radius describing the spread of the *data cloud*. It is noteworthy that the radius represents only an approximated spread of the *data cloud* in the input data space because these *data clouds* lack a specific shape or boundary. In real implementation, the radius is hard to pre-determined and, thus, is recursively calculated based on the local ensemble information. It is updated as [11]:

$$\rho_k^i = \frac{1}{2}(\rho_{(k-1)}^i + \varrho_k^i); \quad \rho_0^i = 1 \quad (14)$$

where ϱ_k^i represents the local scatter of the i th *data cloud* over the input data space at the k th time instance and is expressed as,

$$\varrho_k^i = \sqrt{\Xi_k^i - \|\Gamma_k^i\|^2} \quad (15)$$

The global density and distance information are used to evolve dynamically the *data clouds*. When the condition (12) and condition (13) are both satisfied, a new *data cloud* is formed and the current data sample is assigned as its focal point, $\xi_{N+1} = \mathbf{x}_k$. If the condition (12) and condition (13) are not satisfied, the focal point of the nearest “*data cloud*” is updated by the new data, that is $\xi_f = \mathbf{x}_k; f = \text{argmin}_{i=1}^N \|\mathbf{x}_k - \xi_i\|$. When the focal points are obtained, all other data samples are then assigned to the nearest focal point so that *data clouds* are generated according to the following condition:

$$\text{cloud label} = \text{argmin}_{i=1,2,\dots,N} \|\mathbf{x}_k - \xi_i\| \quad (16)$$

After a new *data cloud* is formed or an existing *data cloud* is updated, \mathbf{x}_k is further utilized to update the consequent parameters \mathbf{Q}_k of the fuzzy system (this is described in the next subsection).

Remark 2: The system output and the evolution of the *data clouds* considered in this paper are based on local and global densities. But different from [26], the local and global densities used (Eq. (3) and Eq. (10)) are derived from the EDA as in [28] and in ε FSS [29].

Remark 3: The EFS considered in this paper employs the global density of the new data sample and its distances to all existing focal points together as criteria to trigger the formation of new *data clouds*. The intrinsic self-evolving learning mechanism ensures that these *data clouds* are more general to represent all the empirically observed data samples. While condition (13) guarantees that the *data clouds* initialised by outliers play a negligible role due to their much lower activation levels and it further avoids the overlap or conflict between *data clouds*.

B. Parameter Learning

In this subsection, the approximation mechanism of the considered EFS of the unknown nonlinear function is studied. Eq. (1) is further expressed as,

$$\hat{y}_k = \mathbf{H}_k^T \mathbf{Q}_k \quad (17)$$

According to the universal approximation property of fuzzy systems, there exists optimal parameters \mathbf{Q}^* that approximate the nonlinear dynamic function $f(\cdot)$ as

$$y_k = \mathbf{H}_k^T \mathbf{Q}^* + \epsilon_k \quad (18)$$

where ϵ_k is the inherent approximation error. By definition, EFSs are capable to dynamically evolve their structure. With the number of *data clouds* increased, the inherent approximation error can be reduced arbitrarily. Therefore, it is reasonable to assume that ϵ_k caused by approximating $f(\mathbf{x})$ is bounded with the constant ε , which is given by

$$|\epsilon_k| \leq \varepsilon \quad (19)$$

On the basis of Eqs. (17) and (18), the system error becomes:

$$e_k = y_k - \hat{y}_k = \mathbf{H}_k^T \tilde{\mathbf{Q}}_k + \epsilon_k \quad (20)$$

where $\tilde{\mathbf{Q}}_k = (\mathbf{Q}^* - \mathbf{Q}_k)$ is the parameter error. Since, there are no antecedent parameters to be adjusted, the parameter

optimization of this EFS is simply equivalent to finding a least-square solution of the parameters \mathbf{Q}_k of the consequent linear model, which is defined as,

$$J_k = \min \sum_{l=1}^k (y_l - \mathbf{H}_l^T \mathbf{Q}_l)^2 \quad (21)$$

The fuzzily weighted recursive least square algorithm introduced in [9] is applied to update the parameters (\mathbf{Q}_k). The adaption laws for updating the parameters \mathbf{Q}_k is defined as,

$$\begin{aligned} \mathbf{Q}_{k+1} &= \mathbf{Q}_k + \alpha_k \Sigma_k \mathbf{H}_k e_k \\ \Sigma_{k+1} &= \Sigma_k - \alpha_k \Sigma_k \mathbf{H}_k \mathbf{H}_k^T \Sigma_k \end{aligned} \quad (22)$$

where α_k is the time-varying learning rate defined as,

$$\alpha_k = \frac{1}{1 + \mathbf{H}_k^T \Sigma_k \mathbf{H}_k} \quad (23)$$

When a *data cloud* is added, the consequent parameters \mathbf{Q}_k becomes:

$$\mathbf{Q}_k = [\mathbf{Q}_{k-1}, \mathbf{q}_{N+1}]^T \quad (24)$$

with the parameters of the new *data cloud* determined by:

$$q_{(N+1)i} = 0, i = 0, 1, 2, \dots, n \quad (25)$$

Also, in the case that a new *data cloud* is added, the dimensionality of the covariance matrix Σ_k increases to:

$$\Sigma_k = \begin{bmatrix} \Sigma_{k-1} & 0 \\ 0 & p_0 I_{n+1 \times n+1} \end{bmatrix} \quad (26)$$

where p_0 is an initial value of the uncertainty assigned to the newly allocated rule. In the paper, p_0 is set to 500 for all the cases. When the focal point of a *data cloud* is replaced by a new data sample, the parameters \mathbf{Q}_k and covariance matrix Σ_k are inherited from the previous time step.

The pseudo-code of the EFS based on data clouds considered in the paper is summarized in **Algorithm 1**.

Algorithm 1 Parameter Learning of EFS based on data clouds

Sample: $\mathbf{x}_k \in \mathbb{R}^n, y_k \in \mathbb{R}$

Initialize: setting $N = 0, \xi_1 = \mathbf{x}_1$

1. Calculate the system output and error according to Eqs. (8) and (20).

2. Compute the global density of the existing data clouds and the new data point based on Eqs.(10)-(11).

3. Apply the rule addition criteria

if $\{(\Upsilon_k - \gamma_k^{i(G)} > 0) \text{ or } (\Upsilon_k - \gamma_k^{i(G)} < 0)\}; \forall i \in 1, \dots, N$ and $\zeta_{ki} > \rho_k^i$ **then**

form a new *data cloud* with

$$N = N + 1, \xi_{N+1} = \mathbf{x}_k, M^{N+1} = 1$$

$$q_{(N+1)i} = 0, i = 0, 1, 2, \dots, n$$

else

The new data sample replaces the focal point of the nearest cloud by applying $\xi_f = \mathbf{x}_k; f = \operatorname{argmin}_{i=1}^N \|\mathbf{x}_k - \xi_i\|$.

end if

4. Adjust the consequent parameters using Eqs. (22) and (23).

IV. STABILITY AND CONVERGENCE ANALYSIS

The main contribution and novelty/originality of this paper is the theoretical proof of the stability of this class of EFSs considered here. Before describing the theorems, a useful lemma is firstly described below.

Lemma 1 [31]: Define $V(s) : \mathbb{R}^n \rightarrow \mathbb{R} \geq 0$ as a Lyapunov function for a nonlinear system. If there exists K_∞ functions $\delta_1(\cdot), \delta_2(\cdot), \delta_3(\cdot)$ and K function $\delta_4(\cdot)$, and for any $s \in \mathbb{R}^n$, each $\nu_k \in \mathbb{R}^n, \sigma \in \mathbb{R}$ satisfies

$$\begin{aligned} \delta_1(s) &\leq V(s) \leq \delta_2(s) \\ V_{k+1} - V_k &= \Delta V_k \leq -\delta_3(\|\nu_k\|) + \delta_4(\sigma) \end{aligned} \quad (27)$$

then the nonlinear system is uniformly stable.

Theorem 1: Consider the EFS described by Eq. (8) with the self-evolving *data cloud*-based structure as described in section III.A. The updating equations for the parameters (\mathbf{Q}_k) are described by Eq.(22). Then, the uniform stability of the fuzzy system described by Eq. (8) is ensured. The error between the system output and the reference output converges to a small neighborhood of zero in which the average identification error satisfies $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e_k^2 \leq (\varepsilon/\tau)^2$. Here, τ is the lower bound of α_k in Eq. (23) satisfying $\tau = \min(\alpha_k)$, and ε is the upper bound of the uncertainty ϵ_k satisfying $|\epsilon_k| < \varepsilon$. Besides, the parameter error $\|\tilde{\mathbf{Q}}_k\|$ is bounded and satisfies the inequality condition $\|\tilde{\mathbf{Q}}_k\| \leq \|\tilde{\mathbf{Q}}_1\|$.

Proof: Consider the following Lyapunov function

$$V_k = \tilde{\mathbf{Q}}_k^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_k \quad (28)$$

According to the matrix inversion lemma ([32], p. 258),

$$(\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{B}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{B}\mathbf{B}^T\mathbf{A}^{-1}}{\mathbf{C}^{-1} + \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}} \quad (29)$$

and by selecting $\mathbf{A}^{-1} = \Sigma_k, \mathbf{B} = \mathbf{H}_k$ and $\mathbf{C}^{-1} = 1$, one can get

$$\Sigma_{k+1}^{-1} = \Sigma_k^{-1} + \mathbf{H}_k \mathbf{H}_k^T \quad (30)$$

Combining Eqs.(28) and (29), we get

$$\begin{aligned} V_{k+1} &= \tilde{\mathbf{Q}}_{k+1}^T \Sigma_{k+1}^{-1} \tilde{\mathbf{Q}}_{k+1} \\ &= \tilde{\mathbf{Q}}_{k+1}^T (\Sigma_k^{-1} + \mathbf{H}_k \mathbf{H}_k^T) \tilde{\mathbf{Q}}_{k+1} \\ &= \tilde{\mathbf{Q}}_{k+1}^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_{k+1} + (\tilde{\mathbf{Q}}_{k+1}^T \mathbf{H}_k)^2 \\ &= (\tilde{\mathbf{Q}}_k - \alpha_k \Sigma_k \mathbf{H}_k e_k)^T \Sigma_k^{-1} (\tilde{\mathbf{Q}}_k - \alpha_k \Sigma_k \mathbf{H}_k e_k) \\ &\quad + (\tilde{\mathbf{Q}}_{k+1}^T \mathbf{H}_k)^2 \\ &= V_k - 2\alpha_k \tilde{\mathbf{Q}}_k^T \mathbf{H}_k e_k + \mathbf{H}_k^T \Sigma_k \mathbf{H}_k \alpha_k^2 e_k^2 + (\tilde{\mathbf{Q}}_{k+1}^T \mathbf{H}_k)^2 \end{aligned} \quad (31)$$

From Eqs.(20), (22) and (23), the following relationship is obtained as,

$$\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k = \alpha_k (\mathbf{H}_k^T \tilde{\mathbf{Q}}_k + \epsilon_k) = \alpha_k e_k \quad (32)$$

Substituting Eq. (32) into Eq. (31) yields,

$$\begin{aligned} V_{k+1} &= V_k - 2\tilde{\mathbf{Q}}_k^T \mathbf{H}_k (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k) \\ &\quad + \mathbf{H}_k^T \Sigma_k \mathbf{H}_k (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k)^2 + (\tilde{\mathbf{Q}}_{k+1}^T \mathbf{H}_k)^2 \end{aligned} \quad (33)$$

Considering Eq. (32), Eq. (33) becomes:

$$\begin{aligned} V_{k+1} &= V_k - (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1})^2 - \mathbf{H}_k^T \Sigma_k \mathbf{H}_k (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k)^2 \\ &\quad - 2\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} \epsilon_k \end{aligned} \quad (34)$$

Because $\mathbf{H}_k^T \Sigma_k \mathbf{H}_k (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k)^2 > 0$, Eq. (34) can be further transformed to:

$$\begin{aligned} V_{k+1} &\leq V_k - (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1})^2 - 2\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} \epsilon_k \\ &\leq V_k + \epsilon_k^2 - (\mathbf{H}_k^T \tilde{\mathbf{Q}}_{k+1} + \epsilon_k)^2 \\ &\leq V_k + \epsilon_k^2 - \alpha_k^2 \epsilon_k^2 \end{aligned} \quad (35)$$

Since $\epsilon_k^2 \leq \varepsilon^2$, Eq. (35) is further expressed as,

$$\Delta V_k \leq -\alpha_k^2 \epsilon_k^2 + \varepsilon^2 \quad (36)$$

For the K_∞ functions $\delta_1(\cdot)$ and $\delta_2(\cdot)$ [23]

$$\begin{aligned} \delta_1(\cdot) &= N(n+1)\min(\|\tilde{\mathbf{Q}}_k\|^2) \\ \delta_2(\cdot) &= N(n+1)\max(\|\tilde{\mathbf{Q}}_k\|^2) \end{aligned} \quad (37)$$

the following inequality holds

$$N(n+1)\min(\|\tilde{\mathbf{Q}}_k\|^2) \leq V_k \leq N(n+1)\max(\|\tilde{\mathbf{Q}}_k\|^2) \quad (38)$$

In the inequality (36), $\alpha_k^2 \epsilon_k^2$ is a K_∞ function and ϵ_k^2 is a K function. It is shown that both inequalities (36) and (38) satisfy the conditions of (27). According to the Lemma 1, the uniform stability of the considered EFS is ensured. By summing up both sides of inequality (36) from 1 up to T , we get:

$$\sum_{k=1}^T (\alpha_k^2 \epsilon_k^2 - \varepsilon^2) \leq V_1 - V_T \quad (39)$$

Since $V_T > 0$ is bounded, Eq. (39) is re-written as,

$$\frac{1}{T} \sum_{k=1}^T \alpha_k^2 \epsilon_k^2 \leq \varepsilon^2 + \frac{1}{T} V_1 \quad (40)$$

As $T \rightarrow \infty$, we obtain

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T \alpha_k^2 \epsilon_k^2 \leq \varepsilon^2 \quad (41)$$

Taking $\tau = \min(\alpha_k)$, one can get

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T \epsilon_k^2 \leq (\varepsilon/\tau)^2 \quad (42)$$

Since $V_k \leq V_1$, this implies that

$$\tilde{\mathbf{Q}}_k^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_k \leq \tilde{\mathbf{Q}}_1^T \Sigma_1^{-1} \tilde{\mathbf{Q}}_1 \quad (43)$$

Combining with Eq. (30), we get:

$$\lambda_{\min}(\Sigma_k^{-1}) \geq \lambda_{\min}(\Sigma_{k-1}^{-1}) \geq \lambda_{\min}(\Sigma_1^{-1}) \quad (44)$$

The inequality (44) can be further transformed into

$$\begin{aligned} \lambda_{\min}(\Sigma_1^{-1}) \|\tilde{\mathbf{Q}}_k\|^2 &\leq \lambda_{\min}(\Sigma_k^{-1}) \|\tilde{\mathbf{Q}}_k\|^2 \leq \tilde{\mathbf{Q}}_k^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_k \\ &\leq \tilde{\mathbf{Q}}_1^T \Sigma_1^{-1} \tilde{\mathbf{Q}}_1 \leq \lambda_{\min}(\Sigma_1^{-1}) \|\tilde{\mathbf{Q}}_1\|^2 \end{aligned} \quad (45)$$

Apparently, the following inequality holds:

$$\|\tilde{\mathbf{Q}}_k\| \leq \|\tilde{\mathbf{Q}}_1\| \quad (46)$$

Therefore, one can conclude that $\|\tilde{\mathbf{Q}}_k\|$ is bounded.

Remark 4: Theorem 1 states that the average error of the system converges to a small neighborhood of zero. The approximation error is caused by the parameter errors. Ideally, when the parameters of the system converge to their optimal

values, the approximation error becomes zero. In practice, zero approximation error is hard to be achieved. Instead, the approximation error obtained is only expected to be smaller than a small nonzero value. Although, the parameters cannot converge to their optimal values, the average identification error will converge to a very small value around zero. This can be further verified through the simulation studies described in the next section.

In the EFS that we consider, the number of *data clouds* N is determined by the learning process. To further demonstrate that the stability and convergence properties will not be affected by adding of the new *data cloud*, we provide and prove the following theorem.

Theorem 2: Considering the EFS described by Eq. (8), when a new *data cloud* is added, \mathbf{Q}_k and Σ_k are updated by Eqs. (24) and (26). The stability of the system described by Eq. (8) is still ensured.

Proof: Consider the following Lyapunov function

$$V_k = \tilde{\mathbf{Q}}_k^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_k \quad (47)$$

A new *data cloud* ($N+1$) is added. Based on Eqs. (24) and (26), Eq. (47) becomes,

$$\begin{aligned} V_{k+1} &= \tilde{\mathbf{Q}}_{k+1}^T \Sigma_{k+1}^{-1} \tilde{\mathbf{Q}}_{k+1} \\ &= \begin{bmatrix} \tilde{\mathbf{Q}}_k^T & \mathbf{q}_{N+1}^T \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & p_0 I_{n+1 \times n+1} \end{bmatrix}^{-1} \\ &\quad \begin{bmatrix} \tilde{\mathbf{Q}}_k \\ \mathbf{q}_{N+1} \end{bmatrix} \\ &= \tilde{\mathbf{Q}}_k^T \Sigma_k^{-1} \tilde{\mathbf{Q}}_k + \frac{1}{p_0} \mathbf{q}_{N+1}^T \mathbf{q}_{N+1} \\ &= V_k \end{aligned} \quad (48)$$

From Eq. (48), it can be found that the newly added *data cloud* has no influence on the Lyapunov function V_k . Based on Theorem 1, the stability of the EFS described by Eq. (8) is ensured.

Remark 5: The stability proof presented in this study is specifically applicable to the EFS considered and described in this paper. However, it is also applicable to more general EFSs satisfying Eq. (8) that are comprised of the structure learning and the fuzzily weighted recursive least square parameter update method, such as eTS [9], Simpl_eTS [10], ESAFIS [14], AnYa [26] and so on.

V. NUMERICAL EXAMPLES

In this section, a number of numerical examples are carried out for evaluating the EFS based on *data clouds* considered in the work and the convergence of the error is visualized. The performance evaluation is based on the Box-Jenkins gas furnace problem, one system identification problem, one Mackey-Glass time series prediction problem, eight real-world benchmark regression problems and one high frequency trading prediction problem. The results from other published self-evolving fuzzy system algorithms, such as SAFIS [13], E-SAFIS [14], eTS [9], Simpl_eTS [10], SONFNN [33], McFIS [12], DENFIS [7], SOFMLS [24], FlexFIS [20], CEFNS [34], SparseFIS [21], Gen-Smart-EFS [22] and AnYa (using eClustering) [26] are used for comparison. All the algorithms go

through the training data sample by sample sequentially in an online scenario, and performance comparison is conducted in terms of the testing accuracy as well as the system complexity (the number of *data clouds* for the proposed system and the original AnYa, the number of rules for other algorithms).

A. Box-Jenkins gas furnace problem

Box-Jenkins gas furnace problem is a well-known benchmark problem [12], [24]. The detailed description of this problem is given in supplementary material.

TABLE I
RESULTS OF BOX-JENKINS GAS FURNACE PROBLEM

Algorithms	Testing RMSE	No. of Rules
AnYa (eClustering) [26]	0.0166	1
CEFNS [34]	0.0207	2
eTS [9]	0.049	5
DENFIS [7]	0.1774	18
SAFIS [13]	0.071	5
Simpl_eTS [10]	0.0485	3
SOFMLS [24]	0.047	5
SONFNN [33]	0.48	4
<i>this paper</i>	0.0107	10

Table I compares the performance of the EFS based on *data clouds* in this paper and the other algorithms. The root mean square error (RMSE) for the testing data is applied as the learning accuracy index. From the table, one can see that the considered EFS in this paper achieves the best testing accuracy compared with other algorithms. Moreover, it has a guaranteed, proven convergence, see Fig. 2(c) (and later supplementary Figs. 1(c) and 2(c)). In addition, it does not require an explicit membership function to be defined per feature/variable. For all alternative algorithms, certain type of membership functions has to be determined in advance. In SAFIS, eTS and CEFNS, Gaussian membership functions are required, while Cauchy membership function is required for the Simpl_eTS. Then the parameters of membership functions are updated according to the evolving learning procedure. The focal points of the EFS described in the original AnYa paper are automatically determined through the eClustering method. Due to the different mechanisms for data clouds formation ([26] vs [28]), the EFS in this paper achieves better testing accuracy. Fig. 2(a) shows the identified focal points and the *data clouds* around them. From the figure, one can see that the EFS considered in this paper generates a somewhat large number of fuzzy rules during the system identification process than other algorithms except DENFIS but achieves much better testing accuracy than all other algorithms.

Fig. 2(c) shows the evolution of the average error during the learning process. From this figure it can be observed that the average error is bounded and converges to a small neighborhood of zero, as stated in Theorem 1. Fig. 2(d) gives the evolution of the parameter α_k and illustrates that the parameter has a lower bound satisfying $\alpha_k \geq 0.5$. The learning error at each time index is depicted in Fig. 2(b), from which it can be seen that the error is bounded and converges to a small neighborhood of zero.

B. Nonlinear System Identification

We further consider the nonlinear system identification problem [13], [24] as a benchmark. The problem description is given in supplementary material in details.

TABLE II
PERFORMANCE COMPARISON FOR NONLINEAR IDENTIFICATION PROBLEM

Algorithms	Testing RMSE	No. of Rules
AnYa (eClustering) [26]	0.0546	2
CEFNS [34]	0.0183	12
DENFIS [7]	0.2599	3
ESAFIS [14]	0.0338	15
eTS [9]	0.0212	49
FlexFIS [20]	0.0171	8
SAFIS [13]	0.0221	17
Simpl_eTS [10]	0.0225	22
SOFMLS [24]	0.0201	5
<i>this paper</i>	0.0020	9

Table II gives the testing accuracy and the number of rules obtained by different algorithms. It indicates that the EFS considered in this paper achieves much better testing accuracy with less fuzzy rules than SAFIS, ESAFIS, eTS, Simpl_eTS and CEFNS, and with slightly more fuzzy rules than SOFMLS, FlexFIS and DENFIS. This table further states that the EFS considered here produces much better testing accuracy than the original AnYa. Supplementary Fig. 1(a) depicts the reference outputs and the prediction outputs of the considered EFS. The evolution of the error and the average error at each time index are presented in supplementary Fig. 1(b) and Fig. 1(c), which clearly show that they are bounded and converge to a small neighborhood of zero. The evolution of the parameter α_k is shown in supplementary Fig. 1(d) and it is clear that the error is bounded, satisfying $\alpha_k \geq 0.5$.

C. Mackey-Glass Time Series Prediction

The fuzzy system is further evaluated by predicting the chaotic Mackey-Glass time series [13]. The detailed problem description is presented in supplementary material.

TABLE III
PERFORMANCE COMPARISON FOR MACKEY-GLASS TIME SERIES PREDICTION

Algorithms	Testing NDEI	No. of Rules
AnYa (eClustering) [26]	0.501	1
CEFNS [34]	0.303	8
DENFIS [7]	0.276	58
eTS [9]	0.356	99
FlexFIS [20]	0.157	89
ESAFIS [14]	0.312	10
SAFIS [13]	0.380	21
Simpl_eTS [10]	0.376	21
<i>this paper</i>	0.124	17

The prediction accuracy and the number of rules obtained by different algorithms are presented in Table III. The Non-Dimensional Error Index (NDEI), which is defined as the RMSE divided by the standard deviation of the true output values, is utilized as the performance index here. From Table

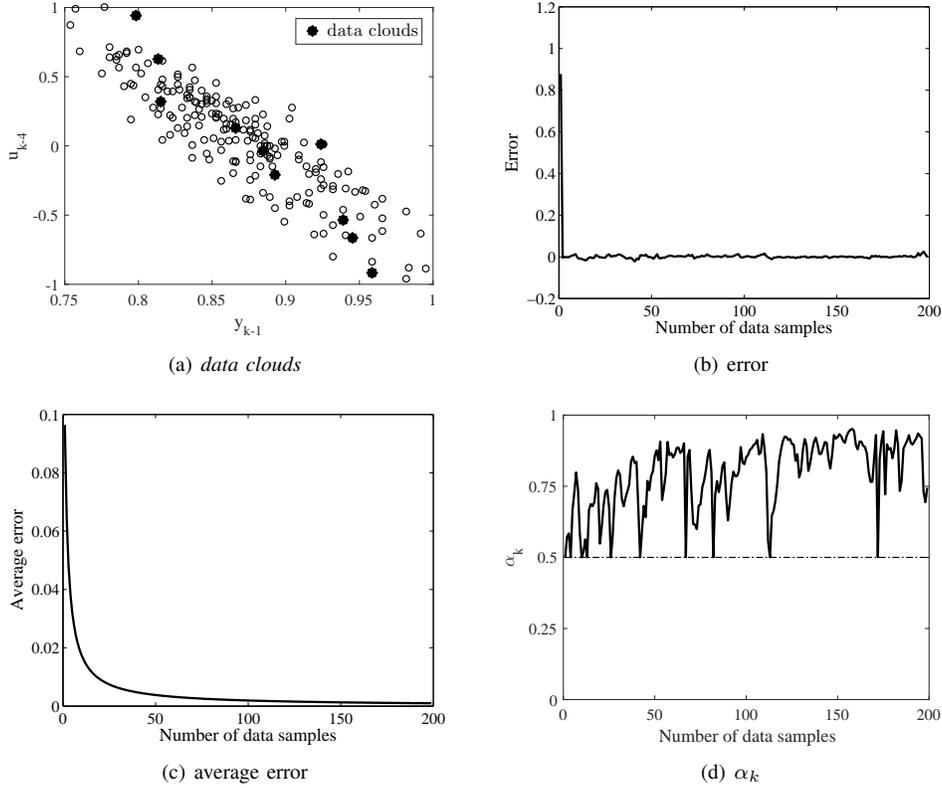


Fig. 2. Evolution of focal points, averaged identification error, training error at each time index and α_k for Box-Jenkins furnace problem.

III, one can find that the EFS considered in this paper achieves a better prediction accuracy with slightly more fuzzy rules than ESAFIS, CEFNS and AnYa with eClustering, and with less rules than DENFIS, eTS, FlexFIS, SAFIS and Simpl_eTS. Supplementary Fig. 2(a) and Fig. 2(b) show the evolution of the average error and error at each time index. From these figure, one can find that they are bounded. The parameter α_k is shown in supplementary Fig. 2(c) and it is clear that it is also bounded, satisfying $\alpha_k \geq 0.5$. Supplementary Fig. 2(d) depicts the reference outputs and the prediction outputs for the EFS that we consider, which indicates that they match very well.

D. Regression Problems

In this subsection, five real-world regression problems are further considered to evaluate the performance of the proposed EFS based on *data clouds*. Details of the problems including the attributes of input and output, the number of training and testing data are listed in supplementary Table I. In these problems, the input and output attributes are normalized in the range $[0, 1]$. All of the training data as listed in the table is used for training and the testing data is used for verifying the generalization performance of the algorithms. Table IV presents the performance comparison between different algorithms. From the table, it can be found that the EFS considered here produces better testing accuracy than all other algorithms for all the problems. For the Trazines problem, DENFIS failed because of the high-dimensional input attributes and, thus, its results for this problem are not provided here. Besides, it is

obvious that in this problem the EFS considered in this paper achieves much better testing accuracy and also requires less fuzzy rules than other algorithms except CEFNS and AnYa with eClustering. This further demonstrates the advantage of the considered EFS on the high dimensional problems, in which the antecedent part of fuzzy rules is represented in a vector form and is, thus, suitable for high-dimensional problems. The curse of dimensionality problem is not going to happen for this type of EFSs thanks to the AnYa type fuzzy rules it employs.

Moreover, the same three data sets as in [22] are further considered to assess the performance of the considered EFS from statistical point of view. The characteristics of these data sets used in this evaluation are summarized in supplementary Table II. As in [22], the evaluation is on the basis of the accuracy of the models obtained by applying tenfold cross-validation (CV). The results from different algorithms on the three data sets are summarized in table V. As in [22], the table includes the minimal cross validation error in terms of mean absolute error (MAE) between reference and estimated values, together with the standard deviation (STD) over the CV-folds (after the \pm symbol), the maximal error between reference and estimated over all training data samples and the average model complexity corresponding to the minimal CV error and the average number of rules and parameters over the CV-folds. From the table, it can be seen that the EFS based on *data clouds* has the least CV error compared with all the other algorithms on the three datasets and also obtains the lower maximal error than all other algorithms on the Auto-

TABLE V
STATISTICAL RESULT COMPARISON BETWEEN DIFFERENT ALGORITHMS ON THREE UCI DATA SETS.

Algorithms	Auto-MPG			Concrete			Boston Housing		
	MAE±STD	Max	Rules/Params	MAE±STD	Max	Rules/Params	MAE±STD	Max	Rules/Params
ANFIS [35]	2.41±0.84	4.07	16/104	11.25±9.98	39.13	8/50	3.59±1.39	6.56	4/24
AnYa (eClustering) [26]	2.50±3.17	4.57	11/88	4.84±6.45	18.16	8/64	4.38±3.77	10.26	7/98
FMCLUST [36]	2.35±0.91	3.99	20/200	7.75±2.06	12.11	3/30	2.84±1.08	5.38	6/60
FLEXFIS [20]	2.17±0.73	3.59	11/143	7.73±1.97	11.96	8/80	2.98±1.27	6.09	6/60
genfis2 [37]	2.23±0.85	3.88	6/114	8.37±1.70	11.52	3/30	3.14±1.31	6.53	4/40
Gen-Smart-EFS [22]	2.09±0.62	3.60	5.8/*	6.35±0.52	7.14	3.8/*	2.94±1.18	5.84	4.9/*
SparseFIS [21]	2.01±0.55	3.29	9/144	7.73±1.77	21.01	7/70	3.02±1.19	6.04	6/60
SparseFIS uncon. [21]	2.14±0.78	4.08	22/352	8.05±1.85	12.52	11/110	3.60±1.56	6.86	19/180
<i>this paper</i>	1.83±0.58	2.58	13/104	4.80±1.62	16.13	11/99	2.07±1.99	3.70	14/196

TABLE IV
PERFORMANCE COMPARISON FOR REGRESSION BENCHMARK PROBLEMS

Datasets	Algorithms	Testing RMSE	No. of Rules
Auto-MPG	AnYa (eClustering) [26]	0.0958	2
	CEFNS [34]	0.0750	2
	DENFIS [7]	0.1458	7
	ESAFIS [14]	0.0731	33
	eTS [9]	0.0864	6
	McFIS [12]	0.0806	4
	SAFIS [13]	0.0993	2
	Simpl_eTS [10]	0.0765	5
<i>this paper</i>	0.0725	8	
Autos	AnYa (eClustering) [26]	0.0604	3
	CEFNS [34]	0.0666	2
	DENFIS [7]	0.4516	8
	ESAFIS [14]	0.0604	3
	eTS [9]	0.0535	3
	McFIS [12]	0.0687	3
	SAFIS [13]	0.1184	5
	Simpl_eTS [10]	0.0689	10
<i>this paper</i>	0.0399	8	
California Housing	AnYa (eClustering) [26]	0.0818	3
	CEFNS [34]	0.0878	2
	DENFIS [7]	0.0715	14
	ESAFIS [14]	0.0892	6
	eTS [9]	0.0772	3
	McFIS [12]	0.0822	15
	SAFIS [13]	0.0988	12
	Simpl_eTS [10]	0.0773	3
<i>this paper</i>	0.0711	11	
Delta Ailerons	AnYa (eClustering) [26]	0.0515	1
	CEFNS [34]	0.0502	3
	DENFIS [7]	0.0497	11
	ESAFIS [14]	0.0506	13
	eTS [9]	0.0513	4
	McFIS [12]	0.0509	15
	SAFIS [13]	0.0549	14
	Simpl_eTS [10]	0.0512	4
<i>this paper</i>	0.0491	14	
Triazines	AnYa (eClustering) [26]	0.0224	6
	CEFNS [34]	0.0452	6
	ESAFIS [14]	0.0331	19
	eTS [9]	0.0179	9
	McFIS	0.0556	12
	SAFIS [13]	0.0581	9
	Simpl_eTS [10]	0.0197	9
	<i>this paper</i>	0.0086	7

MPG and Boston housing datasets.

E. High Frequency Trading (HFT) Prediction Problem

In this subsection, a HFT prediction problem is considered to verify the performance of the EFS based on *data clouds* in handling non-stationary data. The dataset is obtained from the QuantQuote Second Resolution Market Database [38] and contains tick-by-tick data on all NASDAQ, NYSE, and AMEX securities from 1998 to the present moment in time. The frequency of the tick data varies from one second to few minutes. This dataset is comprised of 19144 data samples. The input x_k includes the following five attributes: 1) Time; 2) Open price; 3) High price; 4) Low price, and 5) Close price. The task considered in the work is to predict the future 8 and 24 step values of high price, namely $y_k = x_{k+8,2}$ and $y_k = x_{k+24,2}$, respectively. The data samples are standardized online before prediction.

For comparison, two algorithms widely used in the fields of finance and economy, that is, least square linear regression (LSLR) algorithm and sliding window least square linear regression (SWLSLR) algorithm are considered here. Moreover, other evolving fuzzy algorithms like AnYa with eClustering, eTS, DENFIS and SAFIS are implemented for the purpose of comparison. The width of the sliding window for SWLSLR algorithm is set to 200. The NDEI and the number of rules are taken into account to evaluate the performance. Table VI shows the prediction results for 8 and 24 steps ahead. It is clear from the table that the EFS considered in this paper exhibits better prediction performance than other algorithms in both cases. Supplementary Fig. 3(a) and Fig. 3(b) show the reference and prediction outputs for the EFS based on *data clouds*. One can see from the two figures that the considered HFT problem here contains abnormal data samples and random fluctuations in the data stream. Large fluctuations and abnormal data frequently appear in the beginning and the end of data streams. As observed in the two figures, the EFS that we consider is capable to successfully follow the non-stationary data pattern and exhibits accurate prediction for both 8 and 24 step prediction cases with the error converging to a small value.

VI. CONCLUSIONS

In this paper, we provide systematical stability proof for a class of EFSs based on *data clouds*. The identified *data*

TABLE VI
RESULT COMPARISON BETWEEN DIFFERENT ALGORITHMS FOR HFT
PROBLEM.

Algorithms	y_{k+8}		y_{k+24}	
	Testing	No. of	Testing	No. of
	NDEI	Rules	NDEI	Rules
AnYa (eClustering) [26]	0.164	3	0.231	3
DENFIS [7]	1.598	12	1.582	12
eTS [9]	0.183	6	0.271	7
OLSLR [39]	0.169	/	0.242	/
SAFIS [13]	0.554	20	0.779	14
SWLSLR [40]	0.157	/	0.222	/
this paper	0.152	20	0.188	26

clouds as described in section III-A can objectively represent the local modes of the data distribution and are used as the antecedent (IF) parts of fuzzy rules. Instead of using eClustering, the method of forming *data clouds* described in section III-A is grounded at the empirical data analytics technique and empirical fuzzy sets, which significantly reduces the involvements of human experts and, at the same time, largely enhances the objectiveness of the fuzzy system.

The main contribution of this paper, the stability proof for the EFSs is of great importance in real applications. Nearly all the existing EFSs lack a throughout stability analysis and the very few considered the stability of the EFSs with specific types of membership functions and parameter update technique. Different from the existing works, the stability of the type of EFSs considered is proven through the Lyapunov theory and also the stability proof can be applicable to more general EFSs which satisfy Eq.(8) and are comprised of the structure learning and the fuzzily weighted recursive least square parameter update method, such as eTS [9], Simpl_eTS [10], ESAFIS [14], AnYa using eClustering [26] and so on. The simulation results from the Box-Jenkins furnace problem, one system identification problem, Mackey-Glass time series prediction, 8 real-world benchmark regression problems and one high frequency trading prediction problem show that the considered EFS is not only with a theoretically guaranteed stability, but also obtains better learning accuracy. The simulation results also verify that the average error is bounded and converges to a small neighborhood of zero.

ACKNOWLEDGMENT

This work was supported by the National Science Council of ShaanXi Province (2014JM8337), the National Natural Science Foundation of China (61403300) and the Fundamental Research Funds for the Central Universities. This work was performed while the first author was a visiting scholar at Lancaster University.

REFERENCES

- [1] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice Hall, 1997.
- [2] E. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," in *Proceedings of IEE*, vol. 121, pp. 1585–1588, 1974.
- [3] M. Sugeno and T. Yasukawa, "A fuzzy-logic based approach to qualitative modeling," *IEEE Transactions on fuzzy systems*, vol. 1, no. 1, pp. 7–31, 1993.

- [4] L. Wang and J. Yen, "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithm and Kalman Filter," *Fuzzy Sets and Systems*, vol. 101, pp. 353–362, 1999.
- [5] C. W. Lee and Y. C. Shin, "Construction of fuzzy systems using least-squares method and genetic algorithm," *Fuzzy Sets and Systems*, vol. 137, pp. 297–323, 2003.
- [6] P. Angelov and R. Buswell, "Evolving rule-based models: A tool for intelligent adaptation," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pp. 1062–1067, 2001.
- [7] N. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.
- [8] P. Angelov and D. Filev, "On-line design of takagi-sugeno models," in *Fuzzy Sets and Systems - IFSA 2003, International Fuzzy Systems Association World Congress, Istanbul, Turkey, June 30 - July 2, 2003, Proceedings*, pp. 576–584, 2003.
- [9] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 484–498, 2004.
- [10] P. Angelov and D. Filev, "Simpl_eTS: a simplified method for learning evolving takagi-sugeno fuzzy models," in *The 14th IEEE International Conference on Fuzzy Systems*, pp. 1068–1073, 2005.
- [11] P. Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time," in *2006 International Symposium on Evolving Fuzzy Systems*, pp. 29–35, 2006.
- [12] K. Subramanian and S. Suresh, "A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system," *Applied Soft Computing*, vol. 12, pp. 3603–3614, 2012.
- [13] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction," *Fuzzy Sets and Systems*, vol. 157, no. 9, pp. 1260–1275, 2006.
- [14] H.-J. Rong, N. Sundararajan, G.-B. Huang, and G.-S. Zhao, "Extended sequential adaptive fuzzy inference system for classification problems," *Evolving Systems*, vol. 2, no. 2, pp. 71–82, 2011.
- [15] L. Maciel, R. Ballini, and F. Gomide, "Evolving possibilistic fuzzy modeling for realized volatility forecasting with jumps," *IEEE Transactions on Fuzzy Systems*, 2016.
- [16] A. Lemos, W. Caminhas, and F. Gomide, "Multivariable gaussian evolving fuzzy modeling system," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 91–104, 2011.
- [17] D. Dovzan, V. Logar, and I. Skrjanc, "Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste water treatment process," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, 2014.
- [18] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "PANFIS: A novel incremental learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 55–68, 2014.
- [19] M. Pratama, S. G. Anavatti, and E. Lughofer, "GENEFIS: Toward an effective localist network," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 547–562, 2014.
- [20] E. D. Lughofer, "FLEXFIS: A robust incremental learning approach for evolving takagisugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1393–1410, 2008.
- [21] E. Lughofer and S. Kindermann, "SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 396–411, 2010.
- [22] E. Lughofer, C. Cernuda, S. Kindermann, and M. Pratama, "Generalized smart evolving fuzzy systems," *Evolving Systems*, vol. 6, pp. 269–292, 2015.
- [23] W. Yu and X. Li, "Fuzzy identification using fuzzy neural networks with stable learning algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 3, pp. 411–420, 2004.
- [24] J. de Jesus Rubio, "SOFMLS: Online self-organizing fuzzy modified least-squares network," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1296–1309, 2009.
- [25] H. Han and J. Qiao, "A self-organizing fuzzy neural network based on a growing-and-pruning algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 6, pp. 1129–1143, 2010.
- [26] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *International Journal of Intelligent Systems*, vol. 41, no. 2, pp. 163–185, 2012.
- [27] P. Angelov, *Autonomous Learning Systems: From Data Streams to Knowledge in Real-time*. A John Wiley & Sons, Ltd, 2012.
- [28] P. Angelov, X. Gu, and D. Kangin, "Empirical data analytics," *International Journal of Intelligent Systems*, DOI: 10.1002/int.21899, 2017.
- [29] P. Angelov and X. Gu, "Empirical fuzzy sets," to appear in *International Journal of Intelligent Systems*, 2017, DOI: 10.1002/int.21935.

- [30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of statistical learning: Data mining, inference and prediction, Second Edition*. Springer, 2008.
- [31] J. de Jesus Rubio, P. Angelov, and J. Pacheco, "Uniformly stable backpropagation algorithm to train a feedforward neural network," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 356–366, 2011.
- [32] N. Higham, *Accuracy and Stability of Numerical Algorithms (2nd ed.)*. Philadelphia, PA, USA: SIAM, 2002.
- [33] G. Leng, T. M. McGinnity, and G. Prasad, "An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network," *Fuzzy sets and systems*, vol. 150, no. 2, pp. 211–243, 2005.
- [34] R.-J. Bao, H.-J. Rong, P. Angelov, B. Chen, and P. K. Wong, "Correntropy-based evolving fuzzy neural system," *IEEE Transactions on Fuzzy Systems*, DOI: 10.1109/TFUZZ.2017.2719619, 2017.
- [35] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [36] R. Babuska, *Fuzzy Modeling for Control*. Boston, MA: Kluwer, 1998.
- [37] S. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent and Fuzzy Systems*, vol. 2, no. 3, pp. 267–278, 1994.
- [38] *QuantQuote Second Resolution Market Database*. <https://quantquote.com/historical-stock-data>.
- [39] C. H. Nadungodage, Y. Xia, F. Li, J. J. Lee, and J. Ge, *StreamFitter: A Real Time Linear Regression Analysis System for Continuous Data Streams*. Springer Berlin Heidelberg, 2011.
- [40] K. Tschumitschew and F. Klawonn, "Effects of drift and noise on the optimal sliding window size for data stream regression models," *Communication in Statistics - Theory and Methods*, 2016.