This is the Pre-Published Version.
The following publication J. Zhang, Z. Deng, K. Choi and S. Wang, "Data-Driven Elastic Fuzzy Logic System Modeling: Constructing a Concise System With Human-Like Inference Mechanism," in IEEE Transactions on Fuzzy Systems, vol. 26, no. 4, pp. 2160-2173, Aug. 2018 is available at http://doi.org/10.1109/TFUZZ.2017.2767025.

Data-Driven Elastic Fuzzy Logic System Modeling: Constructing a Concise System with Human-like Inference Mechanism

Jiangbin Zhang, Zhaohong Deng, *IEEE Senior Member*, Kup-Sze Choi, *IEEE Member,* Shitong Wang

***Abstract*—The construction of fuzzy logic systems (FLSs) using data-driven techniques has become the most popular modeling approach. However, this approach still faces critical challenges, including the difficulty in obtaining concise models for high-dimensional data and generating accurate fuzzy rules to simulate human inference mechanism. To tackle these issues, a new FLS modeling framework called data-driven elastic FLS (DD-EFLS) is proposed in this paper. The DD-EFLS has two key characteristics. First, the fuzzy rules in the rule base can use different feature subspaces that are extracted from the original high-dimensional space to yield simple and accurate rules in feature spaces of lower dimensionality. Second, fuzzy inferences from various views are implemented by embedding different rules in the corresponding subspaces to imitate human inference mechanism. Based on the DD-EFLS framework, an elastic Takagi-Sugeno-Kang (TSK) FLS modeling method (ETSK-FLS) is proposed to train the elastic TSK FLS using the concise rules and a more human-like inference mechanism for modeling tasks based on high-dimensional datasets. The characteristics and advantages of the proposed framework and the ETSK-FLS method are validated experimentally using both synthetic and real-world datasets.**

***Index Terms*—Elastic fuzzy logic systems, high-dimensional data, concise and interpretable model, TSK fuzzy logic system.**

I. Introduction

Fuzzy logic systems (FLSs) are intelligent models based on fuzzy sets and fuzzy logic [1], which have a wide range of applications, e.g. pattern recognition, intelligent control, data mining and image processing [2-5, 28-41]. Different from conventional intelligent models, such as neural networks, FLSs can be interpreted easily with rules described in linguistic terms. Meanwhile, FLSs also demonstrate strong learnability when different data-driven (DD) learning techniques are introduced to optimize the parameters of the models [49].

This work was supported in part by the Outstanding Youth Fund of Jiangsu Province (BK20140001), the National Natural Science Foundation of China (61272210), the Hong Kong Research Grants Council (PolyU 152040/16E) and the Hong Kong Polytechnic University (G-UA68, G-UA3W). (Corresponding author: Zhaohong Deng)

J.B. Zhang, Z.H. Deng, and S.T. Wang are with the School of Digital Media, Jiangnan University and Jiangsu Key Laboratory of Digital Design and Software Technology, Wuxi 214122, China (e-mail: JiangbinZhang@jiangnan.edu.cn; dengzhaohong@jiangnan.edu.cn; wxwangst@aliyun.com)

K.S. Choi is with The Centre for Smart Health, the Hong Kong Polytechnic University (e-mail: kschoi@ieee.org)

The introduction of DD learning techniques has made FLSs a very powerful modeling approach for different tasks, e.g. time series predictions, medical diagnosis, pattern classification and so on. Indeed, data-driven FLS (DD-FLS) modeling methods have become the most popular approach for constructing FLSs. Many classical DD-FLS construction methods have been proposed in the past two decades, including Singular Value Decomposition-QR decomposition (SVD-QR) method [6], dynamic optimal training method [7], support vector learning-based method [8, 46-48], self-organizing evolving-based methods [9], hybrid learning algorithms based on different mechanisms [10], bio-inspired methods and methods based on evolutionary learning [11, 42-45].

Although the existing DD-FLSs methods are effective for constructing different types of FLSs, they still face common challenges. Degeneration of interpretability and/or generalizability due to the high dimensionality of the training data is one of the issues. When an FLS is constructed using classical DD-FLS training methods based on high-dimensional data, the fuzzy rules acquired become very complex and exceedingly long, which severely deteriorates the interpretability of the fuzzy rules due to the tedious linguistic descriptions. The decision models thus developed turn out to be impractical for real-life applications, e.g. medical diagnosis. In order to enhance the conciseness and interpretability of FLSs constructed with high-dimensional data, DD-FLS training methods based on feature reduction have been proposed [13-15, 50-54]. Principle component analysis (PCA) is used to capture the prominent components from the training data [13]. However, the features extracted by PCA lose the physical meaning of the original features and it becomes difficult to make sense from the linguistic interpretation of the PCA-based FLS. A scalable two-stage multi-objective genetic algorithm is proposed for constructing more precise FLSs [14]. In the first stage, an evolutionary data-driven based learning technique and an inductive rule-based learning technique are used at the same time. A post-processing process is then followed to perform rule selection and scatter-based tuning of the membership functions to further refine the trained model. On the other hand, a hybrid heuristic approach is integrated with integer-programming formulation to develop a new linguistic fuzzy rule-based classification system for high-dimensional classification problems [15]. In this method, many rules are generated heuristically for each class, which are then selected to form a pool of rules via integer-programming formulation. Other feature selection based methods have also been adopted to develop more compact fuzzy systems [50-54]. For example, feature selection and granularity learning have been used to generate genetic fuzzy rule-based classification systems [51].

Besides interpretability, the generalization performance of the trained FLS may also be deteriorated due to the high-dimensional data. If the size of the high-dimensional datasets is small, the trained models are prone to over-fitting. This situation can be further aggravated if noisy features are embedded in the data space. To overcome the effect of high-dimensional data on generalizability, the structural risk minimization technique that is used in support vector machine has been introduced for parameter learning in DD-FLS construction methods [12, 21].

Another issue with most existing DD-FLS construction methods is that the FLSs are all developed to generate all the fuzzy rules in the same feature space [6-10, 42-45], which does not always in line with the operations of human inference mechanism. To better simulate the mechanism, elastic feature space is more appropriate for characterizing different views.

The above analyses indicate that constructing FLSs with concise and interpretable fuzzy rules remains a challenge. In addition, in most existing DD-FLS construction methods, the assumption that all rules are generated from the same feature space is not necessarily valid and inconsistent with human inference mechanism. To this end, a data-driven elastic FLS (DD-EFLS) framework is proposed in this study. Based on this framework, an elastic TSK-FLS (ETSK-FLS) algorithm is developed for the construction of TSK FLS based on high-dimensional data. In the ETSK-FLS, soft subspace clustering (SSC) technique is first adopted to determine the optimal partition of the input space and to obtain important feature subsets for different clusters. With the SSC results obtained based on the input data of the training dataset, the antecedent parameters of the TSK FLS are estimated and fixed in different subspaces. The parameter learning of the consequents is then transformed to the parameter estimation of a linear model in the mapping hidden feature space. Finally, L2-norm penalty and structural risk minimization-based techniques are used to optimize the consequent parameters.

The contributions of this study are three folded. First, the elastic framework DD-EFLS is proposed for FLS construction. When compared with the classical DD-FLS framework, it is advantageous in that the FLSs constructed under the proposed framework can extract important features for different rules from high-dimensional data. Besides, the antecedents and consequents in different fuzzy rules can be described in the corresponding feature subspaces, which is analogous to the inference mechanism of humans, e.g. different experts usually make an inference from different views for the same problem.

Second, the ETSK-FLS algorithm, proposed within the DD-EFLS framework for TSK FLS construction, adopts fewer features to construct the fuzzy rules. Hence, the rules obtained are more concise and can be interpreted more easily by simple linguistic description. Furthermore, since noisy features can be removed effectively by the ETSK-FLS algorithm, the robustness of TSK FLS constructed by ETSK-FLS remains high even if the original feature space of the high-dimensional data contains noisy features. Within the DD-EFLS framework, human-like inference mechanism also exhibits in the TSK FLS constructed by ETSK-FLS since each fuzzy rule of the trained system implements the inference in its individual view.

Third, the proposed DD-EFLS framework and the ETSK-FLS algorithm are validated comprehensively by experiments conducted on both synthetic and real-world high-dimensional datasets.

The rest of this paper is organized as follows. The model structure of the classical DD-FLS framework is briefly described in Section II. The DD-EFLS framework proposed for data-driven FLS construction with high-dimensional data is presented in Section III. In Section IV, the ETSK-FLS algorithm is proposed for the construction of TSK FLS based on the DD-EFLS framework. The experimental studies conducted to validate the DD-EFLS framework and the proposed ETSK-FLS algorism are reported in Section V. Finally, conclusions and future work are given in Section VI. For clarity and easy reference, the abbreviations used in this paper are listed in Table I.

Table I The abbreviations used in this paper

Abbreviations	Descriptions
FLS	Fuzzy logic system
ML FLS	Mamdani-Larsen fuzzy logic system
TSK FLS	Takagi-Sugeno-Kang fuzzy logic system
DD-FLS	Data-driven fuzzy logic system
DD-EFLS	Data-driven elastic fuzzy logic system
ETSK-FLS	Elastic Takagi-Sugeno-Kang fuzzy logic system
SSC	Soft subspace clustering
EWKM	Entropy weighting k-means

II. Classical DD-FLS Construction

The two major FLSs used in classical DD-FLS construction methods are the Mamdani-Larsen fuzzy logic system (ML FLS) [16, 17] and the TSK FLS [18, 19]. While having the same antecedents (If-parts), the two models differ in the consequents (THEN-parts) of the fuzzy rules. As shown in Table II, A_i^k is a fuzzy subset subscribed by the input variable x_i for the *k*th rule in the antecedents; $\wedge$ is a fuzzy conjunction operator and K is the number of fuzzy rules in the rule base. Each rule is premised on the input vector $\mathbf{x}=(x_1,x_2,\cdots,x_d)^T$ and maps the fuzzy sets in the input space $\mathbf{A}^k\subset R^d$ to a fuzzy set in the output space, denoted by $B^k(b_k,v_k)$ in ML FLS or by a varying fuzzy singleton $f^k(\mathbf{x})$ in TSK FLS. For the consequents of ML FLS, $B^k(b_k,v_k)$ is a fuzzy set with centroid b_k and fuzziness index v_k, while for TSK FLS, the consequent $f^k(\mathbf{x})$ denotes a varying singleton which is a function of the input vector $\mathbf{x}$. The fuzzy membership function in the input space $\mathbf{A}^k\subset R^d$ is denoted by $\mu^k(\mathbf{x})$, which is the firing strength of the *k*th rule. $\mu^k(\mathbf{x})$ is obtained by taking a fuzzy conjunction of the membership functions of all the fuzzy subsets in the antecedents, i.e.,

$$\mu^k(\mathbf{x}) = \mu_1^k(x_1) \wedge \mu_2^k(x_2) \wedge \cdots \wedge \mu_d^k(x_d) \quad (1)$$

Here, a commonly used conjunction operator is multiplication. When this operator is used, $\mu^k(\mathbf{x})$ can be expressed as follows:

$$\mu^k(\mathbf{x}) = \prod_{i=1}^{d} \mu_i^k(x_i) \quad (2)$$

When multiplication and addition are employed respectively as the implication operator and the combination operator, and the center of gravity as the defuzzification operator, the output of the ML FLS and TS FLS can be presented respectively as follows.

$$y_{ML}^0 = \sum_{k=1}^{K} \frac{\mu^k(\mathbf{x}).v_k}{\sum_{k'=1}^{K} \mu^{k'}(\mathbf{x}).v_{k'}}.b_k \quad (3)$$

$$y_{TSK}^0 = \sum_{k=1}^{K} \frac{\mu^k(\mathbf{x})}{\sum_{k'=1}^{K} \mu^{k'}(\mathbf{x})}.f^k(\mathbf{x}) \quad (4)$$

Compared with conventional intelligent models, classical FLSs are distinguished by their interpretability and learnability. Since real-world data are more readily available nowadays, DD-FLS construction methods have become a popular approach. However, its advantages are largely realized for low-dimensional data only. When the data dimensionality is high, the classical DD-FLS construction methods become problematic, as discussed in Section I.

Table II Fuzzy rules used in classical DD-FLS models

Type	Fuzzy Rules	
	Antecedents (IF-parts)	Consequents (THEN-parts)
ML FLS	IF x_1 is $A_1^k \wedge x_2$ is $A_2^k \wedge \ldots \wedge x_d$ is A_d^k, $k = 1, \ldots, K$	THEN y^k is $B^k(b_k, v_k)$
TSK FLS		THEN y^k is $f^k(\mathbf{x})$
Note: All the rules have a common feature space, i.e., inference is implemented from a common view for all fuzzy rules.		

Table III Fuzzy rules developed within the DD-EFLS framework

Type	Fuzzy Rules	
	Antecedents (IF-parts)	Consequents (THEN-parts)
ML FLS	IF x_1^k *is* $A_1^k \wedge x_2^k$ *is* $A_2^k \wedge \ldots \wedge x_{m_k}^k$ *is* $A_{m_k}^k$	THEN y^k is $B^k(b_k, v_k)$
TSK FLS	$\mathbf{x}^k = (x_1, x_2, \ldots, x_{m_k})^T$, $k = 1, \ldots, K$	THEN y^k is $f^k(\mathbf{x}^k)$
Note: Each rule has its individual feature space. Inference is implemented with fuzzy rules from different views.		

III. DD-EFLS Framework for High-dimensional Data

In this section, the new DD-EFLS framework is proposed for the construction of data-driven FLSs with high-dimensional data. Table III shows the structure of the fuzzy rules that are developed for the classical FLS models in the DD-EFLS framework.

The differences between DD-FLS and DD-EFLS can be seen by comparing Table II with Table III. For the antecedents (IF-part), each rule of a classical FLS used with the DD-FLS framework is constructed by using all the features of the input vector, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$. Thus, all the rules have the same input space. However, in the proposed DD-EFLS framework, different feature subsets are adopted for different rules. For example, the kth rule in the DD-EFLS framework is associated with the input vector $\mathbf{x}^k = (x_1^k, x_1^k, \ldots, x_{m_k}^k)^T$, which is a vector containing m_k features extracted from the full features. While the THEN-part of the ML FLS in the DD-FLS framework and the proposed DD-EFLS framework are the same, the THEN-part of the TSK FLS in the DD-EFLS is distinct from that in the classical DD-FLS framework. Different feature sets are adopted as variables in the varying singleton functions of the consequents for TSK FLS in the two frameworks.

Remarks: For the classical ML FLS and TSK FLS, where the parameters of the model are adjusted based on expert knowledge, the use of the fuzzy rules in Table III is a natural choice and can be readily realized. However, it is non-trivial to use these fuzzy rules in the DD-EFLS construction methods due to the difficulty in determining the subspaces for each fuzzy rule based on the available training data. When different subspaces are considered for different rules, the data-driven learning algorithms of the model can become very complicated. Hence, the novelty here lies in the use of the fuzzy rules in Table III for the DD-EFLS construction methods.

IV. Soft Subspace Clustering and L2 Norm Penalty-based Elastic TSK FLS Construction

A. The ETSK-FLS

While the DD-EFLS is a promising framework for constructing high-dimensional data-driven FLSs, the realization of these types of FLSs is not straightforward. In this section, the ETSK-FLS algorithm is proposed for the construction of the TSK FLS within the DD-EFLS framework. The process is described in the diagram shown in Fig. 1.

The process contains two main parts. In Part 1, a rule generation method based on SSC technique [20, 22] is proposed. The main purpose of this part is to acquire important features for each rule. For the different rules, various feature subsets are adopted to construct the fuzzy sets in the antecedents. The feature subsets used for the varying singleton functions of different rules in the consequents are determined accordingly. In Part 2, an ε-insensitive learning strategy based on an L2-norm penalty [21] is used to optimize the consequent parameters by introducing a learning technique for minimizing the structural risks. These two parts will be described in detail in the two subsections below.

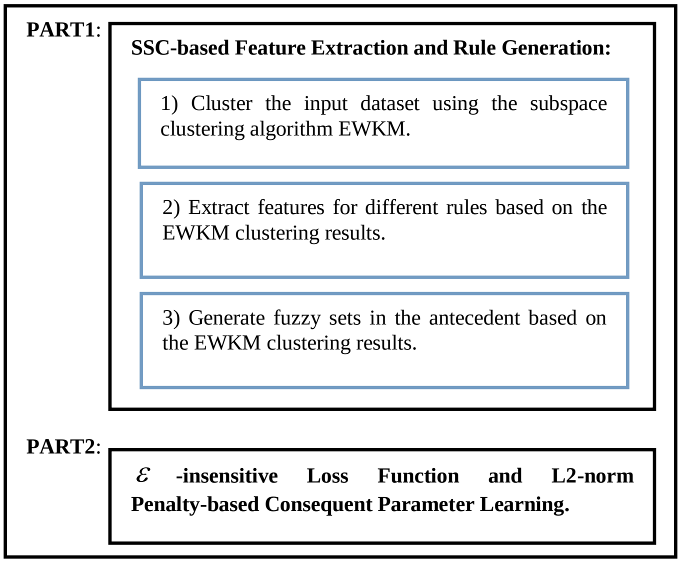

Fig. 1 The process of ETSK-FLS construction.

B. SSC-based Feature Extraction and Antecedents Generation

SSC techniques have attracted increasing attention due to their promising performance in high-dimensional data clustering [20, 22], where the data can be grouped into different clusters and their related subspace, i.e., the important feature subsets associated with different clusters. In FLS modeling, clustering is widely used as effective space partitioning techniques. However, classical clustering techniques do not give the importance of the features for each fuzzy rule, and therefore not effective for FLS modeling in the DD-EFLS framework. On the contrary, SSC not only partitions the data into different clusters but also informs the importance of the features. These characteristics make SSC a very appropriate method for generating the fuzzy rules of the FLS in the proposed DD-EFLS framework.

1) Subspace Clustering Algorithm EWKM

Many SSC algorithms have been proposed, including various fuzzy weighting subspace clustering algorithms and entropy weighting subspace clustering algorithms [20, 28]. In this study, a benchmarking SSC algorithm – the entropy weighting K-means clustering algorithm (EWKM) [22] – is adopted to generate the initial fuzzy rules of the TSK FLS in the DD-EFLS framework. In the EWKM, the following objective is used to optimize the partitioning of the data,

$$J_{EWKM}(\mathbf{U},\mathbf{V},\mathbf{W})=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2+\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\ln w_{ik}$$

$$s.t.\ \ u_{ij}\in\{0,1\},\sum_{i=1}^{C}u_{ij}=1,0<\sum_{j=1}^{N}u_{ij}<N,0\le w_{ij}\le 1,\sum_{k=1}^{D}w_{ik}=1. \quad (5)$$

where C, N and D are the number of clusters, data and features respectively. $\mathbf{U}=[u_{ij}]_{C\times N}$ denotes the hard partition matrix, with its element u_{ij} denoting whether the *j*th data belongs to the *i*th cluster. Note that the column matrix $\mathbf{U}$ only has one element equal to 1, indicating that each instance only belongs to a certain cluster. $\mathbf{V}=[v_{ik}]_{C\times D}$ is the cluster center matrix, with element v_{ik} denoting the *k*th feature of the *i*th cluster center. $\mathbf{W}=[w_{ik}]_{C\times D}$ is the weight matrix, with w_{ik} denoting the importance of the *k*th feature in the *i*th cluster.

Based on the available data and the EWKM clustering algorithm, the optimal clustering center matrix $\mathbf{V}^*$, the partition matrix $\mathbf{U}^*$ and the feature weight matrix $\mathbf{W}^*$ can be obtained. Knowledge of the clustering results can then be used to generate the initial fuzzy rules for the TSK FLS in the DD-EFLS framework.

2) Extracting Features for Different Rules based on EWKM

The optimal feature weight matrix $\mathbf{W}^*$ can be used to determine the important features for each rule. The row vector $\mathbf{w}_k$ of $\mathbf{W}^*$ denotes the importance of the features in the associated cluster. Once the information of the kth cluster is employed to generate the kth fuzzy rule, $\mathbf{w}_k$ can be used to identify the important features in this rule, where features with large weight value should be selected. If the full feature space is expressed as $\mathbf{x}=(x_1,x_2,\cdots,x_d)^T\in R^d$, the important features that have been selected for the kth fuzzy rule can be expressed as follows:

$$(x_1^k,\ \ x_2^k,\ \ \cdots\ \ ,x_{m_k}^k)\qquad k=1,2\cdots,K, \quad (6)$$

where m^k is the number of features extracted from the full feature space for the kth fuzzy rule. In practical applications, we can use different feature selection criteria to determine the final feature subsets for different rules based on $\mathbf{W}^*$. Two feasible approaches are given below:

(a) Fix the number of important features. Given a number m, where $1\le m\le D$, the m features with the largest weight values in $\mathbf{w}_k$ are selected for the kth fuzzy rule. In this strategy, equal number of features are selected from all the fuzzy rules, i.e., $m=m_1=m_2=\cdots=m_K$. Note that the feature subsets of different rules may differ greatly.

(b) Fix the threshold of the weight. For example, we can set a threshold $\eta=\gamma/D$, where $\gamma\ (0\le\gamma\le D)$ is an adjustable parameter. The features with the weights in $\mathbf{w}_k$ that are larger than η are taken as the important features for the kth fuzzy rule. In this strategy, the numbers of features selected for different rules may differ.

For the consequents of TSK FLS in the DD-EFLS framework, the corresponding input variable can be determined according to the selected features of each rule. For example, if $\mathbf{x}^k=(x_1^k,\ldots,x_{m_k}^k)^T$ denotes the feature vector selected for the kth fuzzy rule, the corresponding input vector of the fuzzy singleton function $f^k(\mathbf{x}^k)$ in the consequents can be determined. In this case, if a linear function is adopted, the corresponding fuzzy singleton function $f^k(\mathbf{x}^k)$ can be expressed as

$$f^k(\mathbf{x}^k)=p_0^k+p_1^k x_1^k+,\cdots,+p_{m_k}^k x_{m_k}^k,\quad k=1,2,\ldots,K. \quad (7)$$

3) Generating Fuzzy Sets in the Antecedent based on EWKM

The optimal clustering center matrix $\mathbf{V}^*$ and partition matrix $\mathbf{U}^*$ can be used to determine the antecedent parameters of the fuzzy membership function in a way similar to that in other clustering-based FLS construction methods [12, 27]. The details are given below. Suppose the commonly used Gaussian membership function is employed as the fuzzy membership function, the fuzzy membership function of the ith fuzzy set A_i^k in the kth fuzzy rule can be expressed as

$$\mu_{A_i^k}(x_i^k)=\exp[\frac{-(x_i^k-c_i^k)^2}{2\delta_i^k}] \quad (8)$$

where the parameters c_i^k and δ_i^k can be estimated by applying the clustering results of the EWKM clustering algorithm to the input data of the training dataset, i.e.,

$$c_i^k=\sum_{j=1}^N u_{kj}x_{ji}^k \Big/ \sum_{j=1}^N u_{kj}\ ,\quad i=1,\ldots,m_k, \quad (9)$$

$$\delta_i^k=h\sum_{j=1}^N u_{kj}(x_{ji}^k-c_i^k)^2 \Big/ \sum_{j=1}^N u_{kj}\ . \quad (10)$$

Here, x_{ji}^k denotes the ith feature in the vector $\mathbf{x}_j^k=(x_{j,1}^k,\ldots,x_{j,m_k}^k)^T$; $\mathbf{x}_j^k$ contains m_k important features associated with the kth fuzzy rule, and these features are extracted from the original input instance $\mathbf{x}_j$ in the original training dataset; u_{kj} denotes whether the jth input instance $\mathbf{x}_j$ belongs to the kth cluster; h in (10) is a scale parameter that can be adjusted manually or determined using the cross-validation strategy to find the optimal value [12].

C. ε-insensitive Loss Function and L2-norm Penalty-based Consequent Parameter Learning

Once the antecedent parameters of the fuzzy rules in the DD-FLS framework are determined, the next step is to optimize the consequent parameters of the trained TSK FLS. This can be achieved using the ε-insensitive loss function and L2-norm penalty-based learning strategy.

If the fuzzy sets in the antecedents are fixed, the firing strength of each rule can be calculated. When multiplication is used as the conjunction operator, the firing strength of the kth rule can be obtained as

$$\mu^k(\mathbf{x}^k)=\prod_{i=1}^{m_k}\mu_{A_i^k}(x_i^k)\ . \quad (11)$$

Furthermore, if additive combination operator is used, the output of the TSK FLS in the DD-EFLS framework can be expressed as

$$y=\sum_{k=1}^K\frac{\mu^k(\mathbf{x}^k)}{\sum_{k'=1}^K\mu^{k'}(\mathbf{x}^{k'})}f^k(\mathbf{x}^k)=\sum_{k=1}^K\tilde{\mu}^k(\mathbf{x}^k)f^k(\mathbf{x}^k).,\quad (12)$$

where

$$\tilde{\mu}^k(\mathbf{x}^k)=\mu^k(\mathbf{x}^k)\Big/\sum_{k'=1}^K\mu^{k'}(\mathbf{x}^{k'})\ . \quad (13)$$

The output of the TSK FLS in (12) can be further transformed into the linear form as follows,

$$y_{ETSK\ FS}=\mathbf{p}_g^T\mathbf{x}_g\ , \quad (14)$$

where $\mathbf{p}_g$ is the combined vector of the consequent parameters and $\mathbf{x}_g$ is the mapping vector derived with the fuzzy inference rules from the original input instance $\mathbf{x}$. These two vectors $\mathbf{x}_g$ and $\mathbf{p}_g$ can be constructed as follows,

$$\mathbf{x}_e^k=(1,\mathbf{x}^k)^T=(1,x_1^k,x_2^k,\ldots,x_{m_k}^k)^T,\quad k=1,2,\ldots,K; \quad (15)$$

$$\tilde{\mathbf{x}}^k=\tilde{\mu}^k(\mathbf{x}^k)\mathbf{x}_e^k\quad k=1,2,\ldots,K\ ; \quad (16)$$

$$\mathbf{x}_g=((\tilde{\mathbf{x}}^1)^T,(\tilde{\mathbf{x}}^2)^T,\cdots,(\tilde{\mathbf{x}}^K)^T)^T; \quad (17)$$

$$\mathbf{p}^k=(p_0^k,p_1^k,\cdots,p_{m_k}^k)^T,\quad k=1,2,\ldots,K; \quad (18)$$

$$\mathbf{p}_g=((\mathbf{p}^1)^T,(\mathbf{p}^2)^T,\cdots,(\mathbf{p}^K)^T)^T\ . \quad (19)$$

Many objective criteria are available to optimize the consequent parameters of the ETSK FLS, e.g. the least square method [23] and methods based on the ε-insensitive loss function [21, 24]. In this work, the consequent parameter $\mathbf{p}_g$ is optimized based on the ε-insensitive loss function and the L2-norm penalty-based method. This learning strategy has been shown to be particularly effective [12, 21].

Given a scalar g_i and a vector $\mathbf{g}=(g_1,g_2,\cdots,g_d)^T$, the definition of the ε-insensitive loss function can be expressed as

$$|g_i|_\varepsilon=\begin{cases}g_i-\varepsilon, & g_i>\varepsilon\\ 0, & g_i\le\varepsilon\end{cases}, \quad (20)$$

$$|\mathbf{g}|_\varepsilon=\sum_{i=1}^d|g_i|_\varepsilon\ . \quad (21)$$

where ε is a positive constant.

Given a training dataset $D_{train}=\{\mathbf{x}_i,y_i\}$, $\mathbf{x}_i\in R^d$, $y_i\in R$, $i=1,2,\cdots,N$, the corresponding optimization criterion of the TSK FLS in (12), based on the ε-insensitive loss function, is defined as

$$\min_{\mathbf{p}_g}\ E=\sum_{i=1}^N|y_{ETSK\text{-}FLS,i}-y_i|_\varepsilon=\sum_{i=1}^N|\mathbf{p}_g^T\mathbf{x}_{gi}-y_i|_\varepsilon\ . \quad (22)$$

In general, the inequalities $\mathbf{p}_g^T\mathbf{x}_{gi}-y_i<\varepsilon$ and $y_i-\mathbf{p}_g^T\mathbf{x}_{gi}<\varepsilon$

are not satisfied for all training data. By introducing the slack variables $\xi_i^+ \geq 0$, $\xi_i^- \geq 0$ and the L2-norm penalty terms, the optimization can also be written as

$$\min_{\mathbf{p}_g,\xi_i^+,\xi_i^-} \quad E=\sum_{i=1}^{N}((\xi_i^+)^2+(\xi_i^-)^2),$$

$$s.t \begin{cases} y_i-\mathbf{p}_g^T\mathbf{x}_{gi}<\varepsilon+\xi_i^+ \\ \mathbf{p}_g^T\mathbf{x}_{gi}-y_i<\varepsilon+\xi_i^- \end{cases} \forall i. \tag{23}$$

By introducing the regularization term and the penalty term of the insensitive parameter ε, as in some existing L2-norm penalty-based methods like L2-SVR [25], the optimization objective can be further expressed as

$$\min_{\mathbf{p}_g,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\varepsilon} \quad L(\mathbf{p}_g,\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\varepsilon)=\frac{1}{\tau}.\frac{1}{N}.\sum_{j=1}^{N}((\xi_i^+)^2+(\xi_i^-)^2)+\frac{2}{\tau}\varepsilon+\frac{1}{2}\mathbf{p}_g^T\mathbf{p}_g$$

$$s.t \begin{cases} y_i-\mathbf{p}_g^T\mathbf{x}_{gi}<\varepsilon+\xi_i^+ \\ \mathbf{p}_g^T\mathbf{x}_{gi}-y_i<\varepsilon+\xi_i^- \end{cases} \forall i. \tag{24}$$

In (24), τ is a positive constant used to balance the complexity of the regression model and the error tolerance. However, solving this optimization problem is non-trivial. A commonly used strategy is to transform it into the dual problem as follows (The derivation is given in Appendix 1 of the *Supplementary Materials* section).

$$\max_{\boldsymbol{\lambda}^+,\boldsymbol{\lambda}^-} \; -\sum_{i=1}^{N}\sum_{j=1}^{N}(\lambda_i^+-\lambda_i^-)(\lambda_j^+-\lambda_j^-)\cdot\mathbf{x}_{gi}^T\mathbf{x}_{gj}-\frac{N\tau}{2}\sum_{i=1}^{N}(\lambda_i^+)^2$$

$$-\frac{N\tau}{2}\sum_{i=1}^{N}(\lambda_i^-)^2+\sum_{i=1}^{N}\lambda_i^+\cdot y_i\cdot\tau-\sum_{i=1}^{N}\lambda_i^-\cdot y_i\cdot\tau$$

$$\text{s.t.} \; \sum_{i=1}^{N}(\lambda_i^++\lambda_i^-)=1,\lambda_i^+\geq 0,\lambda_i^-\geq 0 \qquad \forall i. \tag{25}$$

In (25), the solution variables are the Lagrangian multipliers. According to the Karush-Kuhn-Tucker (KKT) conditions, the relationship between the optimal solutions of the primal problem in (24) and the dual problem in (25) can be expressed as follows.

$$\mathbf{p}_g=\frac{2}{\tau}\sum_{i=1}^{N}(\lambda_i^+-\lambda_i^-)\mathbf{x}_{gi}, \tag{26}$$

$$\xi_i^+=N\lambda_i^+, \tag{27}$$

$$\xi_i^-=N\lambda_i^-, \tag{28}$$

$$\varepsilon=\sum_{i=1}^{N}(\lambda_i^+-\lambda_i^-)\mathbf{x}_{gi}-\frac{N}{2}\sum_{i=1}^{N}((\lambda_i^+)^2+(\lambda_i^-)^2)-\frac{1}{\tau}\sum_{i=1}^{N}\sum_{j=1}^{N}(\lambda_i^+-\lambda_i^-)(\lambda_j^+-\lambda_j^-)\mathbf{x}_{gi}^T\mathbf{x}_{gj}. \tag{29}$$

Once the solution of (25) is obtained, the optimal consequent parameters of the TSK FLS in the DD-EFLS framework can be determined using (26).

D. The ETSK-FLS Algorithm

With the results obtained from sections IV-B and IV-C, the construction of the TSK FLS in the DD-EFLS framework based on SSC and ε-insensitive loss-based ETSK-FLS algorithm is described as follows.

Algorithm of ETSK-FLS

Stage 1: The construction of the antecedents of the TSK FLS based on SSC

Step1: Initialize the number of fuzzy rules K and the width in the Gaussian membership function h. Set the weight threshold γ or the number of selected features $m=m_1=m_2=\cdots=m_K$. Set the training dataset $D_{train}=\{\mathbf{x}_i,y_i\}$, $\mathbf{x}_i\in R^d$, $y_i\in R$, $i=1,\cdots,N$.

Step2: Implement EWKM clustering on the input dataset $\{\mathbf{x}_i\}$. Divide $\{\mathbf{x}_i\}$ into K clusters and obtain the partition matrix $\mathbf{U}$. Set the cluster center matrix $\mathbf{V}$ and the feature weight matrix $\mathbf{W}$.

Step3: Match each cluster to a fuzzy rule. Determine the important features for each rule using $\mathbf{W}$ and γ (or m).

Step4: Estimate the parameters of the fuzzy membership functions with (9) and (10).

Stage 2: The learning of consequents of the TSK FLS based on ε-insensitive loss and the structural risk minimization strategy

Step5: Set the regularization parameter τ in (24) and (25).

Step6: Solve the optimal consequent parameters using (25) and (26).

Stage 3: The integrated stage of the TSK FLS

Step7: The final TSK FLS in the DD-EFLS framework is constructed with the antecedent and consequent parameters obtained in Stage 1 and Stage 2.

E. Theoretical Analysis of the ETSK-FLS Algorithm

1) Model Complexity

The complexity of the TSK FLS constructed by the ETSK-TLS algorithm is determined by the total number of parameters involved in the final decision model. If m_k features are selected in the k th rule and the Gaussian membership function is adopted, the numbers of parameters in the antecedents and consequents are $2m_k$ and m_k+1 respectively. Therefore, the complexity of the final decision model is $\sum_{k=1}^{K}(3m_k+1)$, where K is the number of fuzzy rules. For a TSK FLS that is trained by the classical DD-FLS algorithms, the model complexity is $(3d+1)K$, where d is the number of features. For high-dimensional data, $m_k\leq d$, and therefore $\sum_{k=1}^{K}(3m_k+1)\leq(3d+1)K$, i.e., the model complexity of the TSK FLS trained by the proposed ETSK-FLS algorithm is less than that trained by the classical DD-FLS constructions algorithms.

2) Computational Time Complexity

The ETSK-FLS algorithm contains two main parts: the acquisition of the antecedent parameters of the fuzzy rules by using the EWKM subspace clustering algorithm, and the learning of the consequent parameters based on the classical ε-insensitive criterion and L2-norm penalty terms. For the first part, the computational complexity of the EWKM algorithm is $O(TNCd)$ where T, N, C and d are the number of iterations, data, clusters and features respectively. For the second part, the training of the consequent parameters is a QP optimization problem. Since the time complexity of a typical QP solution is $O(N^2)$, the total time complexity of the proposed ETSK-FLS algorithm is $O(TNCd+N^2)$. However, if sophisticated QP algorithms such as the working set based algorithm are adopted, the time complexity can be reduced to between $O(N)$ and $O(N^2)$, depending on the algorithm used.

The time complexity of the proposed ETSK-FLS algorithm is competitive with that of existing TSK FLS construction methods. Take the algorithm that employs PCA and L2-TSK-FLS, i.e., PCA+L2-TSK-FLS, as an example. It uses PCA to extract the key features and then L2-TSK-FLS algorithm to construct the TSK FLS based on the new data with the extracted features. The computational time complexity of the PCA+L2-TSK-FLS algorithm contains three components: (i) the time complexity of PCA, which is typically $O(N^3)$; (ii) the time complexity of fuzzy c-means (FCM) clustering, which is $O(TNC)$ where T, N and C are the number of iterations, data and clusters respectively; and (iii) the time complexity of consequents training of the fuzzy rules based on the ε-insensitive loss function and L2-norm penalty, which is the same as that of the proposed ETSK-FLS algorithm. Therefore, the time complexity of the PCA+L2-TSK-FLS algorithm is $O(N^3+TNC+N^2)$. The experiments conducted to evaluate the computational time complexity of the proposed ETSK-FLS in comparison with the PCA+L2-TSK-FLS will be discussed in Section V-E.

V Experiments

In this section, the performance of the proposed ETSK-FLS algorithm is evaluated and compared comprehensively with the related methods. The methods adopted for comparison and the experimental settings are first described, followed by the analysis and discussion of their performance on four high-dimensional datasets, involving one synthetic dataset and three real-world datasets.

A. Methods and Settings

1) Methods Adopted for Performance Comparison

In the experiments, the L2-norm penalty-based L2-TSK-FLS [12], the L1-penalty-based L1-TSK-FLS (IQP) [21], and the L1-TSK-FLS (LSSLI) [21] were compared with the proposed ETSK-FLS. Meanwhile, PCA was combined respectively with these three methods to reduce the dimension of the high-dimensional data. Thus, the proposed ETSK-FLS was compared with a total of six methods. A brief description of these methods is given in Table S1 in the *Supplementary Materials* section.

2) Parameter Setting

For the algorithms adopted in the experiments, FCM clustering was used to partition the input space, except for the proposed ETSK-FLS which used the subspace clustering algorithm EWKM instead for the input space partitioning and feature extraction. The fuzzy index of FCM was set to 2. The five-fold cross-validation strategy was adopted to obtain the optimal hyper parameters within a given search grid for each algorithm. The ranges of the search grids of various hyper parameters in the algorithms are given in Table S2 in the *Supplementary Materials* section.

3) Performance Indices

The classification accuracy J_{acc}, precision J_{pre} and recall J_{rec} are used to evaluate the generalization performance of the classification tasks. Besides, the per-class accuracy

$$J_{perclas}=\frac{\text{Number of test samples classified correctly in a certain class}}{\text{Number of test samples in a certain class}} \quad (30)$$

is also used in some experiments.

In particular, another index J_{comp} is adopted to evaluate the model complexity of the final decision model, which is defined as

$$J_{comp} = \text{total number of parameters involved in the final decision model}. \quad (31)$$

4) Other Settings

In the experiments, all the features of the original input data were normalized to the range [-1, 1].

B. Synthetic Dataset

A synthetic dataset was prepared to evaluate the performance of the proposed ETSK-FLS algorithm effectively. It had three characteristics: (1) a large number of features existed in the dataset; (2) the important features varied in different data subsets; and (3) noisy features embedded in the full feature space. The synthetic dataset thus generated contained 300 instances with 150 features. The instances belonged to three different classes and each class contained 100 instances. Notably, different classes were assigned with important features in the corresponding subspaces. The important features for different classes are listed in Table IV. Random Gaussian noise was introduced into all the other features. The noisy features usually reduce the performance of the classical DD-FLS construction algorithms.

Table IV The important features of each class in the synthetic dataset

Class labels	Index number of important features
Class 1	2,10,17,30,41,50,88,100,114,120
Class 2	10,11,30,31,50,51,80,81,100,101
Class 3	6,10,50,63,80,94,100,109,120,123

In the classification task, the class labels were used as the output in the model training procedure. Once the TSK FLSs of the different methods were constructed, for a test instance, the label nearest to the FLS output was taken as the predicted label. The details of the procedure are explained as follows. In the training procedure, the labels of the examples (e.g. ‘1’, ‘2’…, ‘5’) were directly used as the outputs for the training of the TSK FLS regression model. Once the TSK FLS was trained, given an input vector of a test example, the output of the TSK FLS would be a real value, say, ‘2.11’. Then, among all the labels, the label ‘2’ is nearest to the output and would be taken as the label of the test example. The optimal performance achieved by the proposed method with the five-fold cross-validation strategy and that by the six methods under comparison are reported in Table V. From these results, the following observations can be obtained.

(1) It can be seen from Table V that, in terms of classification accuracy and model complexity, the proposed ETSK-FLS method showed distinctive advantages over the other six methods. In particular, since the final decision model obtained by the ETSK-FLS algorithm had lower model complexity and produced more concise rules, the fuzzy rules obtained could be interpreted more easily using linguistic expressions. Also, fast decision making was expected with this concise model.

(2) For L2-TSK-F LS, ε -TSK-FLS(IQP), and ε -TSK-FLS(LSSLI), since all features were regardless used to construct the fuzzy rules, the fuzzy inference was adversely affected by the noisy features and the classification performance was reduced. The rules also became too tedious to be readily explained using linguistic terms. Even though the PCA feature extraction technique was used, many features were still required by the three conventional methods to attain the optimal classification accuracy. For example, the PCA+L2-TSK-FLS method used 86 new features to construct a TSK FLS that could yield optimal classification result. It is noted that although fewer features could be extracted using PCA, the new features were indeed a combination of the original features, which may not have a clear physical meaning and makes it difficult to give a reasonable linguistic explanation of the fuzzy rules.

(3) For the proposed ETSK-FLS, various features of the different fuzzy rules were extracted. The rule base of the TSK FLS obtained by ETSK-FLS thus resembled an expert base, where each rule corresponded to an expert and the associated feature subset corresponded to the view of the expert.

Fig. 2 shows the weight distribution of each cluster obtained by the EWKM clustering algorithm when the optimal classification accuracy was attained. It can be seen from this figure that the features of the three clusters were different, i.e., the corresponding fuzzy rules were generated in different subspaces and implemented by fuzzy inference from different views. Based on the weight distributions and the optimal threshold parameter obtained, i.e. $\gamma = 1$ as reported in Table V, the extracted features of the three fuzzy rules, namely R1, R2 and R3, are listed in Table VI. By comparing Table IV and Table VI, it is clear that the most important features embedded in different data subsets were effectively detected.

Referring to the antecedent and consequent parameters of the TSK FLS that were finally constructed with the ETSK-FLS and ε -TSK-FLS (IQP) methods, as shown respectively in Table S3 and Table S4 in the *Supplementary Materials* section, we can see that the TSK FLS trained by ETSK-FLS with high-dimensional dataset was concise, yielding a system that could be described easily and clearly with linguistic terms. The final decision model constructed by the ε -TSK-FLS (IQP) was much more complicated which severely deteriorated the interpretability of the model.

The classification performance of the ETSK-FLS algorithm with different number of rules is given in Table VII. From the results, we can see that the number of clusters, i.e., the number of rules, was a key parameter for the proposed ETSK-FLS algorithm. In particular, a large number of clusters would result in fuzzy model with exceedingly many fuzzy rules. In this case, the model may be overfitted when trained by the training data and the generalizability of the model would be degenerated for the test data. Hence, an appropriate value is necessary for this parameter, which can be obtained with some existing strategies. In this study, this parameter was considered as a hyper parameter and the commonly used cross-validation strategy was adopted to determine the appropriate value within a predefined search grid, as shown in Table S2 in the Supplementary Materials section.

Table V Performance of the seven TSK FLS construction methods on synthetic dataset

Method	**Optimal Parameters**	**Number of features adopted**	**Classification Indices (%) (Mean±SD)***						**Model Complexity**
			J_{acc}	J_{pre}	J_{res}	$J_{perclas}$			J_{comp}
						Class 1	Class 2	Class 3	
L2-TSK-FLS	$k=3$, $h=0.01$, $\tau=0.01$	150 for each rule	72.00 ±6.17	75.00 ±12.24	78.73 ±4.03	75.00 ±12.25	67.00 ±5.70	74.00 ±6.52	3×(2×150+151)= 1353
ε -TSK-FLS (IQP)	$k=3$, $h=100$, $\tau=1$	150 for each rule	70.67 ±6.41	72.00 ±12.55	76.31 ±3.39	72.00 ±12.55	61.00 ±6.52	79.00 ±6.52	3×(2×150+151)= 1353
ε -TSK-FLS (LSSLI)	$k=5$, $h=0.01$, $\tau=10$	150 for each rule	73.67 ±5.45	72.00 ±8.37	83.69 ±4.87	72.00 ±8.37	76.00 ±4.18	73.00 ±10.95	5×(2×150+151)= 2255
PCA+ L2-TSK-FLS	$k=5$, $h=0.01$, $\tau=0.01$, $\eta=0.85$	86 for each rule	70.00 ±6.35	72.00 ±12.04	79.85 ±6.40	72.00 ±12.04	70.00 ±6.12	68.00 ±5.70	150×86+ 5×(2×86+87)=14195
PCA+ ε -TSK-FLS (IQP)	$k=3$, $h=0.01$, $\tau=10$, $\eta=0.85$	86 for each rule	45.00 ±3.91	53.00 ±15.65	50.08 ±6.70	53.00 ±15.65	24.00 ±10.84	58.00 ±11.51	150×86+ 3×(2×86+87)=13677
PCA+ ε -TSK-FLS (LSSLI)	$k=5$, $h=0.01$, $\tau=10$, $\eta=0.85$	86 for each rule	69.67 ±8.03	66.00 ±15.17	86.00 ±9.02	66.00 ±15.17	78.00 ±7.58	65.00 ±7.07	150×86+ 5×(2×86+87)=14195
ETSK-FLS	$k=3$, $\gamma=1$, $h=0.01$, $\tau=0.01$, $r=10$	(11+10+12)[#]	99.67 ±0.75	100.00 ±0	97.23 ±4.09	100.00 ±0	100.00 ±0	99.00 ±2.24	(2×11+12)+(2×10+11) +(2×12+13)=102

***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**
#The three rules R1, R2 and R2 contained on average 13, 9 and 12 features respectively.

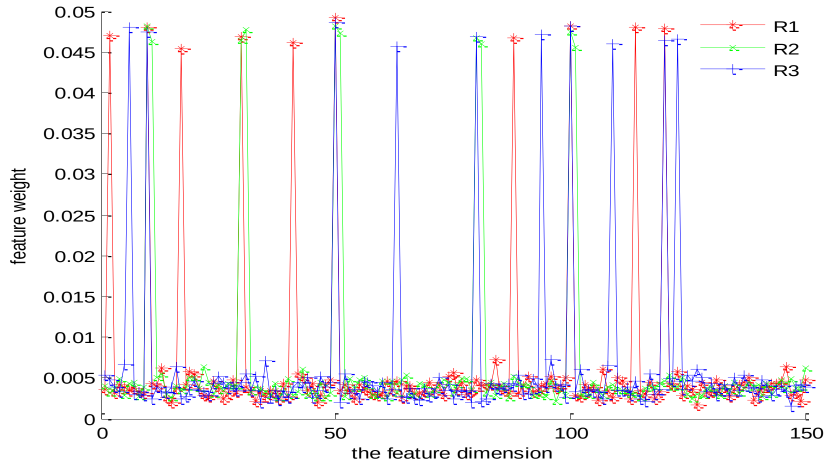

Fig 2 The distribution of weights in the weight matrix W^*, obtained by EWKM clustering on the input data of the synthetic dataset in a certain run, when the optimal classification accuracy was attained by the ETSK-FLS.

Table VI Features extracted for each rule by the ETSK-FLS algorithm in a certain run when the best classification accuracy was attained.

Rule	R_1	R_2	R_3
Index number of extracted features	2,10,17,30,41,50, 84,88,100,114,120	10,11,30,31, 50,51,80,81, 100,101	6,10,35,50,63,80,94,96,100,109,120,123

Table VII Classification performance of the ETSK-FLS on synthetic data with different number of rules

Number of rules	2	3	4	5	6	7
Classification Accuracy (%)	99.20 ±2.24	99.67 ±0.75	96.32 ±2.79	95.00 ±2.04	93.67 ±1.39	92.67 ±3.02
Number of rules	8	9	10	11	25	32
Classification Accuracy (%)	91.67 ±2.64	91.00 ±3.03	90.08 ±1.18	88.24 ±4.31	85.67 ±9.02	81.63 ±9.65

C. Real-world Datasets

This section presents the performance of the proposed ETSK-FLS algorithm on three high-dimensional real-world datasets obtained from the UCI database [26]. The details of the datasets are listed in Table VIII. The optimal performance attained by the seven methods with the five-fold cross-validation strategy is given in Tables IX-XI.

Table VIII The real-world classification datasets adopted

Dataset	**Size of data**	**Number of features**	**Number of classes**
LSVT-Voice-Rehabilitation	126	310	3
Sonar	208	60	2
Musk	476	167	2
Smartphone	5744	561	6

Table IX Performance on the dataset LSVT-Voice-Rehabilitation

Method	**Optimal Parameters**	**Number of features adopted**	**Classification Indices (%) (Mean±SD)***			**Model Complexity**
			J_{acc}	J_{pre}	J_{res}	J_{comp}
L2-TSK-FLS	$k=3$, $h=100$, $\tau=0.1$	310 for each rule	83.83 ±9.01	85.00 ±16.29	74.25 ±14.38	3×(2x310+311)=2793
ε-TSK-FLS (IQP)	$k=7$, $h=0.001$, $\tau=0.0001$	310 for each rule	76.33 ±3.98	82.50 ±20.91	60.90 ±4.57	7×(2x310+311)=6517
ε-TSK-FLS (LSSLI)	$k=3$, $h=0.01$, $\tau=100$	310 for each rule	86.83 ±14.09	85.25 ±20.92	74.57 ±13.87	3×(2x310+311)=2793
PCA+ L2-TSK-FLS	$k=3$, $h=10$, $\tau=0.01$, $\eta=1$	125 for each rule	77.67 ±12.64	85.00 ±16.30	64.41 ±15.14	310×125+3×(2×125+126)=39878
PCA+ ε-TSK-FLS (IQP)	$k=3$, $h=100$, $\tau=1$, $\eta=1$	125 for each rule	74.00 ±9.12	75.00 ±15.30	55.73 ±11.40	310×125+3× (2×125+126)=39878
PCA+ ε-TSK-FLS (LSSLI)	$k=7$, $h=0.1$, $\tau=100$, $\eta=1$	125 for each rule	87.83 ±12.26	83.53 ±14.19	92.70 ±8.59	310×125+7×(2×125+126)=41382
ETSK-FLS	$k=3$, $\gamma=0.99$, $h=0.01$, $\tau=0.1$, $r=100$	(140+199+161)#	84.83 ±3.60	84.37 ±11.96	72.22 ±13.39	(2×140+141)+(2×199+200)+(2×161+162) =1502

***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**
#The three rules contained on average 140, 199 and 161 features respectively.

The following conclusions can be drawn from the experimental results. First, the proposed ETSK-FLS was advantageous over the classical methods in model complexity. The fuzzy rules of the TSK FLS constructed by the ETSK-FLS employed fewer features. Different features were also extracted for different rules. Thus, the linguistic interpretation of the resulted TSK FLS was much more concise. The mechanism adopted for making inference was similar to the human inference mechanism, where each rule implemented the inference from an individual view, resembling the way that the knowledge of individual experts is sought in solving a problem. On the other hand, the results also indicated clearly that the classification performance of the proposed ETSK-FLS on real-world datasets was highly competitive with the existing state-of-the-art TSK construction methods.

Referring to the classification performance on the dataset Sonar, when the optimal accuracy was attained, the weight distribution of each cluster obtained by the EWKM clustering algorithm is shown in Fig. 3. We can see that the two clusters had different features, indicating that the fuzzy rules associated with these clusters were generated in different subspaces and the fuzzy inference was made from different views. Table XII shows the features extracted for the two fuzzy rules obtained based on the weight distributions in Fig. 3 and the optimal threshold parameter $\gamma = 1.2$.

The final TSK FLS constructed by ETSK-FS with the dataset Sonar, described with its antecedent and consequent parameters, is shown in Table XIII. The trained model was very concise and the linguistic description of the TSK FLS thus obtained was clear.

Table X Classification performance on the dataset Sonar

Method	Optimal Parameters	Number of features adopted	Classification Indices (%) (Mean±SD)*			Model Complexity
			J_{acc}	J_{pre}	J_{res}	J_{comp}
L2-TSK-FLS	$k=11$, $h=0.1$, $\tau=0.01$	60 for each rule	82.22 ±6.98	88.30 ±8.22	80.30 ±5.53	11×(2×60+61)=1991
ε-TSK-FLS (IQP)	$k=32$, $h=0.1$, $\tau=0.01$	60 for each rule	86.12 ±6.53	85.69 ±10.00	87.97 ±4.82	32×(2×60+61)=5792
ε-TSK-FLS (LSSLI)	$k=32$, $h=0.1$, $\tau=0.1$	60 for each rule	89.01 ±5.33	88.37 ±9.14	91.20 ±6.59	32×(2×60+61)=5792
PCA+ L2-TSK-FLS	$k=3$, $h=10$, $\tau=0.01$, $\eta=0.85$	15 for each rule	78.74 ±5.62	82.01 ±9.53	79.81 ±6.82	60×15+3×(2×15+16) =938
PCA+ ε-TSK-FLS (IQP)	$k=3$, $h=0.01$, $\tau=1$, $\eta=0.85$	15 for each rule	79.36 ±2.90	75.65 ±14.62	80.64 ±4.10	60×15+3×(2×15+16)=938
PCA+ ε-TSK-FLS (LSSLI)	$k=3$, $h=0.01$, $\tau=1$, $\eta=0.85$	15 for each rule	81.30 ±3.89	78.42 ±5.67	81.57 ±4.53	60×15+3×(2×15+16) =938
ETSK-FLS	$k=2$, $\gamma=1.2$, $h=0.01$, $\tau=0.01$, $r=10$	(19+19)#	84.39 ±5.61	82.92 ±7.2	81.13 ±4.8	(2×19+20)+(2×19+20)=116

***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**
+The two rules contained on average 19 and 19 features respectively.

Table XI Classification performance on the dataset Musk

Method	Optimal Parameters	Number of features adopted	Classification Accuracy (%) (Mean±SD)*			Model Complexity
			J_{acc}	J_{pre}	J_{res}	J_{comp}
L2-TSK-FLS	$k=3$, $h=0.01$, $\tau=0.1$	166 for each rule	70.18 ±6.27	74.09 ±12.80	75.34 ±9.72	3×(2×166+167)=1497
ε-TSK-FLS (IQP)	$k=3$, $h=10$, $\tau=0.1$	166 for each rule	72.60 ±9.79	70.44 ±18.85	79.95 ±9.31	3×(2×166+167)=1497
ε-TSK-FLS (LSSLI)	$k=3$, $h=10$, $\tau=1$	166 for each rule	74.69 ±16.16	68.53 ±27.55	84.13 ±11.34	3×(2×150+151)=1353
PCA+ L2-TSK-FLS	$k=3$, $h=10$, $\tau=0.1$, $\eta=0.85$	13 for each rule	55.48 ±1.24	85.11 ±9.72	64.87 ±4.05	166×13+3×(2×13+14) =2276
PCA+ ε-TSK-FLS (IQP)	$k=3$, $h=1$, $\tau=0.1$, $\eta=0.85$	13 for each rule	71.49 ±8.73	75.89 ±14.31	73.19 ±12.17	166×13+ 3×(2×13+14)=2276
PCA+ ε-TSK-FLS (LSSLI)	$k=9$, $h=0.01$, $\tau=10$, $\eta=0.85$	13 for each rule	71.49 ±8.73	74.36 ±13.17	72.14 ±11.70	166×13+ 3×(2×13+14)=2276
ETSK-FLS	$k=3$, $\gamma=0.8$, $h=100$, $\tau=0.1$, $r=10$	(126+89+84)#	81.49 ±2.56	85.28 ±15.0	83.31 ±1.76	(2×126+127)+(2×89+90)+ (2×84+85)=900

***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**
#The three rules contained on average 126, 89 and 84 features respectively.

Table XII Classification performance on the dataset Smartphone

Method	Optimal Parameters	Number of features adopted	Classification Accuracy (%) (Mean±SD)*			Model Complexity
			J_{acc}	J_{pre}	J_{res}	J_{comp}
L2-TSK-FLS	$k=3$, $h=0.01$, $\tau=100$	561 for each rule	18.15 ±0.19	49.89 ±5.11	13.09 ±0.85	3×(561×2+562)=5052
ε-TSK-FLS (IQP)	$k=3$, $h=0.1$, $\tau=10$	561 for each rule	63.92 ±2.29	61.89 ±4.80	81.22 ±2.44	3×(561×2+562)=5052
ε-TSK-FLS (LSSLI)	$k=3$, $h=0.01$, $\tau=10$	561 for each rule	42.96 ±0.73	41.23 ±13.19	56.54 ±8.25	3×(561×2+562)=5052
PCA+ L2-TSK-FLS	$k=3$, $h=100$, $\tau=10$, $\eta=0.9$	41 for each rule	14.35 ±1.24	27.65 ±4.53	7.81 ±1.07	561×41+3×(41×2+42)= 23373
PCA+ ε-TSK-FLS (IQP)	$k=3$, $h=0.001$, $\tau=100$, $\eta=0.95$	76 for each rule	59.55 ±2.48	49.65 ±3.82	84.04 ±4.64	561×76+3×(76×2+77)= 43323
PCA+ ε-TSK-FLS (LSSLI)	$k=3$, $h=0.1$, $\tau=10$, $\eta=0.95$	76 for each rule	54.51 ±1.49	47.27 ±2.80	85.86 ±5.06	561×76+3×(76×2+77)= 43323
ETSK-FLS	$k=3$, $\gamma=1.05$, $h=0.01$, $\tau=0.01$, $r=100$	(263+250+363)[#]	57.15 ±2.60	45.73 ±6.20	89.32 ±5.07	(263×2+264)+(250 ×2+251)+(363×2+364)= 2631

***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**
#The three rules contained on average 263, 250 and 363 features respectively.

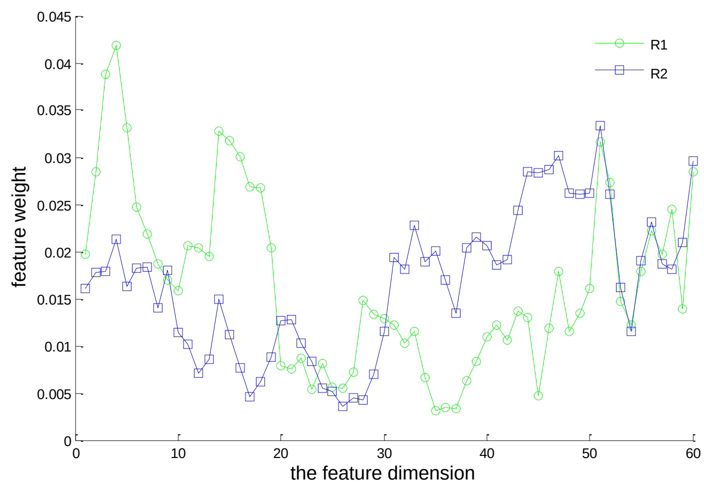

Fig 3 The distribution of weights in the weight matrix $\mathbf{W}^*$, obtained by EWKM clustering on the input data of the dataset Sonar in a certain run, when the optimal classification accuracy was attained by ETSK-FLS.

Table XII Features extracted for the two rules by the ETSK-FLS in a certain run when the best classification accuracy was attained for the dataset Sonar.

Rule	Features extracted for each rule
R1	3, 4, 6, 40, 41, 44, 51, 59, 60
R2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 46, 47, 50, 51, 52, 55, 56, 57, 58, 59, 60

D.Real-world Datasets with Noisy Features Embedded

Here, noisy features were introduced to the two real-world high-dimensional datasets LSVT-Voice-Rehabilitation and Sonar to evaluate the robustness of the proposed ETSK-FLS. The experimental procedure was as follows. First, 50% of the features were randomly selected, to which Gaussian white noise with a power of 50 dBW were introduced. The same experimental procedure described in Section V-C was then implemented on the noisy data. Details of the results are reported in Tables S5 and S6 in the *Supplementary Materials* section. The results show that the proposed ETSK-FLS was advantageous over the other methods in both classification accuracy and model complexity.

Furthermore, the classification performance on the two datasets LSVT-Voice-Rehabilitation and Sonar, with and without noisy features embedded, are compared in Table XIV. It is obvious that the proposed ETSK-FLS was very robust against the noisy features, while the performance of the other TSK-FLS construction methods was sensitive to and easily affected by the noisy features. This can be explained by the fact that in the ETSK-FLS, instead of using all features regardless, each rule only makes use of a certain feature set for fuzzy inference, which greatly reduces the impact of the noisy features. Besides, different feature subsets were also used in different rules to improve the adaptability.

E. Running Time

The running time of the experiments discussed in Section V-B and V-C was shown in Table S7 in the *Supplementary Materials* section. The results indicate that the computational efficiency of the proposed ETSK-FLS was comparable to that of the classical TSK FLS construction algorithms.

F. Performance of Regression

Although the experimental studies above focus on classification, the proposed method can be applied for regression as well. In this subsection, four real-world regression datasets were adopted for performance evaluation. The performance index in (32), commonly used for regression, was employed for the analysis [12],

$$J_{reg} = \sqrt{\frac{\sum_{i=1}^{N}(y_i' - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}}\ , \tag{32}$$

where N is the number of test data; y_i is the output for the ith test input; y_i' is the fuzzy model output for the ith test input and $\bar{y} = \sum_{i=1}^{N} y_i \big/ N$. The smaller the value of J, the better the generalization performance.

The results of the regression tasks are given in Tables S8-S12 in the *Supplementary Materials* section, which are similar to the results obtained from the experimental analyses of classification, i.e., while the regression performance of the proposed method was comparable or better than that of the existing classical fuzzy system modeling methods, the fuzzy system generated was more concise and the elastic fuzzy rules could be interpreted more easily, attributed to the human-like inference mechanism.

Table XIII The rule base generated when ETSK-FLS achieved the optimal generalization performance on the dataset Sonar.

TSK FLS rule R^k :

IF x_1^k *is* $(c_1^k, \delta_1^k) \wedge x_2^k$ *is* $A_2^k(c_2^k, \delta_2^k) \wedge \ldots \wedge x_{m_k}^k$ *is* $A_{m_k}^k(c_{m_k}^k, \delta_{m_k}^k)$, $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_{m_k}^k)^T$

THEN $y^k = f^k(\mathbf{x}^k) = p_0^k + p_1^k x_1^k + \cdots + p_{m_k}^k x_{m_k}^k$

Rule	Antecedent parameters (The parameters of the Gaussian membership function)	Consequent parameters (The parameters of the linear function)
R1	$\mathbf{c}_1$ =[-0.4264 -0.5099 -0.1832 -0.2655 -0.3056 -0.3979 -0.492343127490040 -0.2238 -0.5311] $\boldsymbol{\delta}_1$ =[0.1504 0.1314 0.1507 0.0947 0.1191 0.0931 0.1463 0.1490 0.1396]	$\mathbf{p}_1$ =[0.6104, 0.1676, -0.5445, -0.1015, 0.3178, 0.0104, -0.6078, -0.4403, -0.2194, 0.0995]
R2	$\mathbf{c}_2$ =[-0.6941 -0.7615 -0.7865 -0.8303 -0.7077 -0.5521 -0.4199 -0.5141 -0.5587 -0.5079 -0.6257 -0.6043 -0.5531 -0.7248 -0.6802 -0.6777 -0.6524 -0.6352 -0.7303 -0.6525 -0.7727] $\boldsymbol{\delta}_2$ =[0.0390 0.0247 0.0231 0.0152 0.0381 0.0695 0.0711 0.0613 0.0575 0.0725 0.0704 0.0652 0.0779 0.0286 0.0439 0.0576 0.0502 0.0654 0.0366 0.0627 0.0260]	$\mathbf{p}_2$ =[0.2610, 0.0819, -0.1149, 0.0243, 0.0165, -0.1213, -0.2593, 0.1322, -0.0592, -0.5609, -1.1407, -0.0955, -0.4334, 0.5765, 0.1179, 0.0976, -0.1634, -0.0333, 0.3158, -0.2730, 0.1561, -0.1174]

Table XIV Performance on two real-world datasets with noisy features embedded

Dataset	Method and Classification Accuracy (Mean±SD)*						
	L2-TSK -FLS	ε-TSK -FLS(IQP)	ε-TSK -FLS(LSSLI)	PCA+ L2-TSK-FLS	PCA+ ε-TSK-FLS (IQP)	PCA+ ε-TSK-FLS (LSSLI)	ETSK-FLS
LSVT+	83.83 ±9.01	76.33 ±3.98	86.83 ±14.09	77.67 ±12.64	74.00 ±9.12	87.83 ±12.26	84.83 ±3.60
LSVT (noisy)#	73.00 ±6.91	70.67 ±6.60	72.33 ±6.38	64.67 ±6.79	63.17 ±9.25	67.83 ±6.96	82.50 ±3.48
Sonar	82.22 ±6.98	86.12 ±6.53	89.01 ±5.33	78.74 ±5.62	79.36 ±2.90	81.30 ±3.89	84.39 ±5.61
Sonar (noisy)#	53.38 ±0.62	70.22 ±9.67	71.16 ±8.17	59.1 ±3.67	59.00 ±2.90	59.52 ±5.80	81.05 ±4.42

+LSVT denotes the dataset LSVT-Voice-Rehabilitation.
#Noisy features were embedded in the dataset.
***Mean and SD of the classification accuracies obtained using five-fold cross-validation strategy.**

VI CONCLUSIONS

In this paper, the DD-EFLS framework is proposed to overcome the degeneration of generalizability and interpretability of the classical DD-FLS construction methods due to high-dimensional data. With this framework, the ETSK-FLS algorithm is proposed for the construction of TSK FLS, where SSC technique was introduced to extract important features for each fuzzy rule. Compared with classical methods, the proposed ETSK-FLS has two distinctive advantages. First, the fuzzy rules are generated using only a few features extracted from the input data, and are robust against noisy features. Second, human-like inference mechanism is realized with each fuzzy rule only implementing the inference using its individual feature subspace.

Although the performance of the proposed ETSK-FLS is promising, further issues remain to be addressed. For example, the TSK FLS model is only considered in this study for developing the FLS construction algorithm in the DD-EFLS framework. Algorithms for other types of FLS models, particularly the ML FLS, will be a topic for future research. In addition, type-2 FLS construction algorithms under the proposed DD-EFLS framework are also worth further investigation [27-29].

Another interesting work is to develop elastic data-driven fuzzy systems by integrating subspace extraction technique with clustering methods that can automatically find the appropriate number of clusters. If the appropriate number of fuzzy rules can be determined more efficiently for fuzzy modeling, the learning time can be reduced to a certain extent.

In this study, the SSC and ε-insensitive loss function are adopted to realize the ETSK-FLS in the DD-EFLS framework. Research on better strategies that can be applied to develop elastic fuzzy systems in the DD-EFLS framework will be a significant future work. Furthermore, the scalability of the elastic fuzzy systems in the DD-EFLS framework is also important due to the increasing size of large-scale data.

REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.

[2] A. K. Jain, R. P. W. Duin, J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

[3] L. X. Wang, A course in fuzzy systems and control, 1997. Prentice-Hall, Englewood Cliffs, New Jersey, ISBN 0 13.540882: 2.

[4] E. A. El-Sebakhy, "Data mining in forecasting PVT correlations of crude oil systems based on Type1 fuzzy logic inference systems," *Computers & Geosciences*, vol. 35, no. 9, pp. 1817-1826, 2009.

[5] F. L. Chung, Z. H. Deng, S. T. Wang, "An adaptive fuzzy-inference-rule-based flexible model for automatic elastic image registration," *IEEE Trans. Fuzzy Systems*, vol. 17, no. 5, pp. 995-1010, 2009.

[6] C. C. Chen, C. C. Wong, "Significant fuzzy rules extraction by an SVD-QR-based approach," *Cybernetics and Systems*, vol. 36, no. 6, pp. 597-622, 2005.

[7] M. O. Efe, O. Kaynak, "A novel optimization procedure for training of fuzzy inference systems by combining variable structure systems technique and Levenberg–Marquardt algorithm," *Fuzzy Sets Syst.*, vol. 122, no. 1, pp. 153-165, 2001.

[8] R. Ji, Y. P. Yang, W. D. Zhang, "TS-fuzzy modeling based on ε-insensitive smooth support vector regression," *Journal of Intelligent and Fuzzy Systems*, vol. 24, no. 4, pp. 805-817, 2014.

[9] M. T. Su, C. H. Chen, C. Jian, "A Rule-Based Symbiotic Modified Differential Evolution for Self-Organizing Neuro-Fuzzy Systems," *Applied Soft Computing*, vol. 11, no. 8, pp. 4847-4858, 2011.

[10] M. 1. M. Gerardo, J. C. Martinez, et al., "Orthogonal-least-squares and backpropagation hybrid learning algorithm for interval A2-C1 singleton type-2 Takagi-Sugeno-Kang fuzzy logic systems," *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 2, pp. 125-135, 2014.

[11] G. G. Sebastián, P. D. P Rocío, et al., "Uncertainty-tolerant scheduling strategies for grid computing: knowledge-based techniques with bio-inspired learning," *Image Processing & Communications*, vol. 18, no. 1, pp. 37-44, 2013.

[12] Z. H. Deng, K. S. Choi, F. L. Chung, S. T. Wang, "Scalable TSK Fuzzy Modeling for Very Large Datasets Using Minimal-Enclosing-Ball Approximation," *IEEE Trans. Fuzzy Systems*, vol. 19, no. 2, pp. 210-226, 2011.

[13] Z. Liu, C. L. Philip Chen, Y. Zhang, et al., "Type-2 hierarchical fuzzy system for high-dimensional data-based modeling with uncertainties," Soft Computing, vol. 16, no. 11, pp. 1945-1957, 2012.

[14] M. J. Gacto, M. Galende, et al., "METSK-HDe: A multiobjective evolutionary algorithm to learn accurate TSK-fuzzy systems in high-dimensional and large-scale regression problems," *Information Sciences*, vol. 276, pp. 63-79, 2014.

[15] E. K. Aydogan, I. Karaoglan, P. M. Pardalos, "hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems," *Applied Soft Computing*, vol. 12, no. 2, pp. 800-806, 2012.

[16] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Computers*, vol. 100, no. 12, pp. 1182-1191, 1977.

[17] P. M. Larsen, "Industrial applications of fuzzy logic control," *International Journal of Man-Machine Studies*, vol. 12, no. 1, pp. 3-10, 1980.

[18] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Systems, Man, Cybernetics.*, vol. 15, no. 1, pp. 116-132, 1985.

[19] M. Sugeno, G. T. Kang, "Structure identification of fuzzy model," *Fuzzy sets and Systems*, vol. 28, no. 1, pp. 15-33, 1988.

[20] Z. H. Deng, K. S. Choi, F. L. Chung, et al, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," *Pattern Recognition*, vol. 43, no. 3, pp. 767-781, 2010.

[21] J. M. Leski, "TSK-fuzzy modeling based on ε-insensitive learning," *IEEE Trans. Fuzzy Systems*, vol. 13, no. 2, pp. 181-193, 2005.

[22] L. P. Jing, M. K. Ng, J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026-1041, 2007.

[23] J. Yen, L. Wang, C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. Fuzzy Systems.*, vol. 6, no. 4, pp. 530-537, 1998.

[24] M. F. Azeem, M. Hanmandlu, N. Ahmad, "Generalization of adaptive neuro-fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 11, no. 6, pp. 1332-1346, 2000.

[25] I. W. H. Tsang, J. T. Y. Kwok, J. A. Zurada, "Generalized core vector machines," *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1126-1140, 2006.

[26] K. Bache, M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013

[27] Z. H. Deng, K. S. Choi, L. B. Cao, et al., "T2FELA: Type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 664-676, 2014.

[28] M. J. Mendel, *Uncertain rule-based fuzzy logic system: introduction and new directions*, Prentice-Hall, Upper-Saddle River, NJ, 2001.

[29] M. Biglarbegian, W. W. Melek, J. M. Mendel, "On the stability of interval type-2 TSK fuzzy logic control systems," *IEEE Trans. Systems, Man, Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 798-818, 2010.

[30] J. M. Mendel, R. I. John, F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. Fuzzy Systems*, vol. 14, no. 6, pp. 808-821, 2006.

[31] Q. Liang, J. M. Mendel, "An introduction to type-2 TSK fuzzy logic systems," *Proceedings of FUZZ-IEEE'99*, vol. 3, pp. 1534-1539, 1999.

[32] M. Almaraashi, R. John, S. Coupland, A. Hopgood, "Time series forecasting using a TSK fuzzy system tuned with simulated annealing," *Proceedings of FUZZ-IEEE'2010*, pp. 1-6, 2010.

[33] G. M. Mendez, O. Castillo, "Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm," *Proceedings of FUZZ-IEEE'2005*, pp. 230-235, 2005.

[34] J. R. Castro, O. Castillo, P. Melin and A. Rodríguez-Díaz, "A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks," *Information Sciences*, vol. 179, no. 13, pp. 2175-2193, 2009.

[35] O. Castillo and P. Melin, "Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review," *Information Sciences*, vol. 205, pp. 1-19, 2012.

[36] P. Melin, O. Castillo, "A review on the applications of type-2 fuzzy logic in classification and pattern recognition," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5413-5423, 2013.

[37] G. M. Mendez, "Interval type-1 non-singleton type-2 TSK fuzzy logic systems using the hybrid training method RLS-BP," *In Analysis and Design of Intelligent Systems Using Soft Computing Techniques*, pp. 36-44, Springer Berlin Heidelberg, 2007.

[38] G. M. Mendez, L. A. Leduc, "First-order interval type-1 non-singleton type-2 TSK fuzzy logic systems," *In MICAI 2006: Advances in Artificial Intelligence*, pp. 81-89, Springer Berlin Heidelberg, 2006.

[39] Z. H. Deng, K. S. Choi, Y. Z Jiang, S. T. Wang, "Generalized Hidden-Mapping Ridge Regression, Knowledge-Leveraged Inductive Transfer Learning for Neural Networks, Fuzzy Systems and Kernel Methods," *IEEE Trans. Cybernetics*, vol. 44, no. 12, pp. 2585-2599, 2014.

[40] Z. H. Deng, K. S. Choi, L. B. Cao, S. T. Wang, "T2FELA: Type-2 Fuzzy Extreme Learning Algorithm for Fast Training of Interval Type-2 TSK Fuzzy Logic System," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 664-676, 2014.

[41] Z. H. Deng, Y. Z Jiang, F. L. Chung, H. Ishibuchi, S. T. Wang, "Knowledge-Leverage-Based Fuzzy System and Its Modeling," *IEEE Trans. Fuzzy Systems*, vol. 21, no. 4, pp. 597-609, 2013.

[42] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol. 3, no. 3, pp. 260-270, 1995.

[43] H. Ishibuchi, T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy sets and Systems*, vol. 141, no. 1, pp. 59-88, 2004.

[44] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27-46, 2008.

[45] O. Cordón, F. Herrera, P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Tran. Fuzzy Systems*, vol. 9, no. 4, pp. 667-674, 2001.

[46] C. F. Juang, R. B. Huang, W. Y. Cheng, "An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems," *IEEE Trans. Fuzzy Systems*, vol. 18, no. 4, pp. 686-699, 2010.

[47] C. F. Juang, S. H. Chiu, S. W. Chang, "A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems," *IEEE Trans. Fuzzy Systems*, vol. 15, no. 5, pp. 998-1008, 2007.

[48] C. F. Juang, S. H. Chiu, S. J. Shiu, "Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation," IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Human, vol. 37, no. 6, pp. 1077-1087, 2007.

[49] X. S. Hu, S. E. Li, Y. Yang, "Advanced machine learning approach for lithiumion battery state estimation in electric vehicles," *IEEE Trans. Transportation Electrification*, vol. 2, no. 2, pp. 140-149, 2016.

[50] P. Villar, A. Fernandez, R. A. Carrasco, F. Herrera, "Feature selection and granularity learning in genetic fuzzy rule-based classification systems for highly imbalanced data-sets," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 03, pp. 369-397, 2010.

[51] E. Lughofer, S. Kindermann, "Rule weight optimization and feature selection in fuzzy systems with sparsity constraints," *Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*, Lisbon, Portugal, 2009:950-956.

[52] N. Xiong, P. Funk, "Construction of fuzzy knowledge bases incorporating feature selection," *Soft Computing*, vol. 10, no. 9, pp. 796-804, 2006.

[53] N. Xiong, "A hybrid approach to input selection for complex processes," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 32, no.4, pp. 532-536, 2002.

[54] R. Alcala, M. J. Gacto, F. Herrera, J. Alcalafdez, "A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 15, no. 5, pp. 539-557, 2011.

[55] S. R. Bulò, M. Pelillo, I. M. Bomze, "Graph-based quadratic optimization: A fast evolutionary approach," *Computer Vision & Image Understanding*, vol. 115, no. 7, pp. 984-995, 2011.

[56] A. Chakeri, L. O. Hall, "Large data clustering using quadratic programming: a comprehensive quantitative analysis," *Proceedings of IEEE International Conference on Data Mining Workshop*, pp.806-813, 2015.

[57] L. O. Hall, A. Chakeri, "Relational data partitioning using evolutionary game theor," *Proceedings of 2014 IEEE Symposium on Computational Intelligence and Data Mining*, vol. 357, pp.113-120, 2014.

[58] A. Chakeri, L. O. Hall, "Dominant sets as a framework for cluster ensembles: an evolutionary game theory approach," *Proceedings of the 22nd International Conference on Pattern Recognition*, pp.3457-3462, 2014.