IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA-ENABLED INTELLIGENCE FOR DIGITAL HEALTH

Received August 14, 2019, accepted August 22, 2019, date of publication August 26, 2019, date of current version September 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937657

Common and Special Knowledge-Driven TSK Fuzzy System and Its Modeling and Application for Epileptic EEG Signals Recognition

YUANPENG ZHANG[1,2], (Member, IEEE), JIANCHENG DONG[3], JUNQING ZHU[4], AND CHUNYING WU[4]

[1]Department of Medical Informatics, Medical School, Nantong University, Nantong 226001, China
[2]Department of Health Technology and Informatics, The Hong Kong Polytechnic University, Hong Kong
[3]Medical Big Data Research Center, First Affiliated Hospital of Zhengzhou University, Zhengzhou 450001, China
[4]Case Center for Imaging Research, Department of Radiology, Case Western Reserve University, Cleveland, OH 44106, USA

Corresponding authors: Jiancheng Dong (dongjc@ntu.edu.cn), Yuanpeng Zhang (maxbirdzhang@ntu.edu.cn), and Junqing Zhu (yxj410@case.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 81701793, and in part by the Hong Kong Scholars Program under Grant XJ2019056.

ABSTRACT Takagi-Sugeno-Kang (TSK) fuzzy systems are well known for their good balances between approximation accuracy and interpretability. Among a wide variety of existing TSK fuzzy systems, most of them are driven by special knowledge since the learned parameters of each fuzzy rule are totally different. However, common knowledge is equally important and useful in practice and hence a TSK fuzzy system embedded with common knowledge should be more intuitive and interpretable when tackling with real-world problems. In this paper, we propose a common and special knowledge-driven TSK fuzzy system (CSK-TSK-FS), in which the parameters corresponding to each feature in then-parts of fuzzy rules always keep invariant and these parameters are viewed as common knowledge. As for its modeling, except the gradient descent techniques and other existing training algorithms, we can obtain a trained CSK-TSK-FS from a trained GMM or a trained FLNN because the proposed fuzzy system CSK-TSK-FS is mathematically equivalent to a special GMM and a FLNN. CSK-TSK-FS has three characteristics: (1) with the classical centroid defuzzification strategy, the involved common knowledge can be separated from fuzzy rules such that the interpretability of CSK-TSK-FS can be enhanced; (2) it can be trained quickly by the proposed LLM-based training algorithm; (3) the equivalence relationships among CSK-TSK-FS, GMM and FLNN allow them to share some commonality in training such that the proposed LLM-based training algorithm provides a novel fast training tool for training GMM and FLNN. Experimental results on UCI, KEEL and epileptic EEG datasets demonstrate the promising classification of CSK-TSK-FS.

INDEX TERMS Common knowledge, FLNN, GMM, LLM, special knowledge, TSK fuzzy systems.

I. INTRODUCTION

Epilepsy is a finite episode of brain dysfunction caused by abnormal discharge of cerebral neurons. With regards to the clinical diagnosis of epilepsy, electroencephalogram (EEG) signals are often employed to decide its presence and type [3]. Many machine learning approaches including SVM, fuzzy systems, KNN, decision trees [1]–[3], [39] have been developed and successfully used for epileptic EEG signals recognition. Among these machine learning approaches, the Takagi–Sugeno–Kang (TSK) fuzzy system is a fuzzy rule-based inference system [1]–[3], which have been most used for EEG signals recognition and other applications [46]–[48] because of its strong approximation capability and good interpretability. Generally speaking, a TSK fuzzy system, e.g., zero-order-TSK [4] or one-order-TSK [4] can be taken

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Cheng.

as a knowledge-driven model in which the knowledge is scattered in each fuzzy rule. Undoubtedly, the knowledge is the cornerstone of strong approximation capability and good interpretability of TSK fuzzy systems. More specifically, we can consider the knowledge as the parameters learned in each fuzzy rule and hence the types of knowledge are decided by the way the parameters presented in fuzzy rules. If a TSK fuzzy system is considered as an expert system, then each fuzzy rule can be taken as an expert with different/special knowledge. Based on the special knowledge, for a problem that is as the input, the expert system can output a decision result effectively in most cases. However, although the special knowledge is effective in driving an expert system, the common knowledge between experts sometimes is also useful for the deduction of an expert system. In clinical diagnosis, the common knowledge between medical experts can help them make a more accurate clinical diagnosis decision. For instance, the common knowledge ''diabetic eye disease is most likely caused by retinopathy'' can help medical experts earn reputations in the clinical diagnosis of eye disease. Therefore, from the perspective of the application, constructing a TSK fuzzy system with special knowledge associating with common knowledge is very significant.

As we stated before, knowledge is represented by parameters learned in each fuzzy rule. That is to say, the difference between a TSK fuzzy system only with special knowledge and a TSK fuzzy system with both special and common knowledge is their different combination modes of input features. In [5], the authors carry out the existing simple regression models on about 60 real-world datasets, the conclusion hints us that the mode of knowledge presentation (the combination mode of input features) in a simple regression model can be flexible and varied.

Therefore, in this paper, inspired by the conclusion in [5] and considering the requirements of the application in real-world, we re-design the mode of knowledge presented in the classical one-order TSK fuzzy system and propose a novel TSK fuzzy system, termed as CSK-TSK-FS that is driven by common and special knowledge. Our CSK-TSK-FS is different from the classical one-order TSK fuzzy system. As for the classical one-order TSK fuzzy system, learned parameters in each fuzzy rule are special, and hence, we also consider one-order TSK fuzzy systems as special knowledge-driven fuzzy systems. But for the proposed fuzzy system CSK-TSK-FS, except for special knowledge, common knowledge is also embedded as realized by the parameters involved in one-order parts of then-parts of fuzzy rules. In other words, parameters involved in one-order parts always keep invariant for all fuzzy rules.

With the embedded common knowledge, CSK-TSK-FS becomes more interpretable as a result having its fuzzy rules shortened implicitly. More importantly, its modeling is no longer limited by traditional algorithms as the proposed fuzzy system CSK-TSK-FS is mathematically equivalent to a special Gaussian mixture model (GMM) [6] and a functional link neural network (FLNN) [7] such that the algorithms of modeling GMM and FLNN can also be transferred to CSK-TSK-FS.

The contributions of this paper can be summarized as the following three aspects:

1) A novel TSK fuzzy system embedded with common knowledge and special knowledge is proposed. Compared with the classical one-order TSK fuzzy system, the proposed one is more interpretable because the common knowledge in the then-parts of fuzzy rules can implicitly shorten the length of fuzzy rules, at least to a certain extent. Besides, the performance of the proposed TSK fuzzy system can be guaranteed by the conclusion deduced in [5] that the combination mode of input features in a simple regression model is flexible.
2) We reveal a relationship between the proposed fuzzy system CSK-TSK-FS and GMM with a certain constraint. Thus, from a trained GMM, we can obtain the proposed fuzzy system. In other words, we find a new training algorithm for the proposed fuzzy system. In addition, we also find that the Gaussian FLNN is mathematically equivalent to the proposed fuzzy system, so through the proposed fuzzy system, we establish a relationship between GMM and FLNN, and accordingly extend their training algorithms, respectively.
3) An LLM-based fast learning algorithm for CSK-TSK-FS is proposed. In other words, we also develop a new learning algorithm for GMM and FLNN because of their equivalence relations.

The following sections are organized as: section II gives some preliminaries; Section III gives the detail information about CSK-TSK-FS and reveals the relationship between it and GMM and FLNN; Section IV reports the experimental results and section V concludes our works.

II. PRELIMINARY

Since the proposed fuzzy system CSK-TSK-FS can be viewed as a special GMM [6] and a Gaussian FLNN [7], we prepare some preliminaries about the TSK fuzzy system, GMM and FLNN in this section.

A. TSK FUZZY SYSTEM

TSK fuzzy systems are fuzzy rule-driven inference systems in which the most commonly used fuzzy rules, e.g., the kth, can formally consist of their respective if-parts and then-parts, i.e.,

$$\begin{aligned} &\text{If} \quad \omega_1 \text{ is } \Omega_1^k \wedge \omega_2 \text{ is } \Omega_2^k \wedge \ldots \wedge \omega_d \text{ is } \Omega_d^k, \\ &\text{then} \quad \phi^k(\boldsymbol{\omega}) = \rho_0^k + \rho_1^k \omega_1 + \ldots + \rho_d^k \omega_d, \\ &\quad k = 1, 2, \ldots, K, \end{aligned} \tag{1}$$

where Ω_i^k is a fuzzy subset subscribed by feature ω_i involved in the feature space denoted as $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_d]^T$, K is the total number of fuzzy rules, and notation $\wedge$ represents a fuzzy conjunction operator. Each fuzzy rule is premised on the feature space and maps fuzzy sets

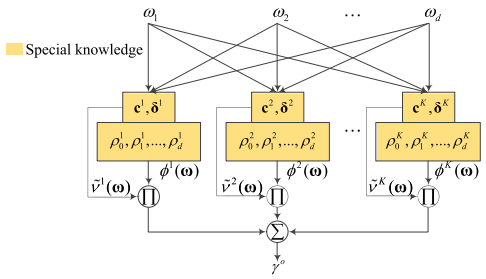


FIGURE 1. Framework of a special knowledge-driven TSK fuzzy system.

$\mathbf{\Omega}^k = [\Omega_1^k, \Omega_2^k, \ldots, \Omega_d^k]$ from the feature space $\mathbf{\Omega}^k \subset R^d$ to a linear function or a constant here represented by $\phi^k(\boldsymbol{\omega})$. Usually, the defuzzification process is achieved by a straightforward weighted summation. Therefore, the output γ^o for a potentially new sample $\boldsymbol{\omega}$ can be formulated as

$$\begin{aligned}\gamma^o &= \sum_{k=1}^{K}\left(\nu^k(\boldsymbol{\omega})/\sum_{m=1}^{K}\nu^m(\boldsymbol{\omega})\right)\phi^k(\boldsymbol{\omega}) \\ &= \sum_{k=1}^{K}\tilde{\nu}^k(\boldsymbol{\omega})f^k(\boldsymbol{\omega}).\end{aligned} \tag{2}$$

In (2), $\nu^k(\boldsymbol{\omega})$ and $\tilde{\nu}^k(\boldsymbol{\omega})$ denote the compatibility and the normalized compatibility of $\boldsymbol{\omega}$ associating with the fuzzy set $\mathbf{\Omega}^k$ of the kth fuzzy rule, respectively, which can be computed as

$$\nu^k(\boldsymbol{\omega}) = \prod_{i=1}^{d}\nu_{\Omega_i^k}(\omega_i) \quad \text{and } \tilde{\nu}^k(\boldsymbol{\omega}) = \nu^k(\boldsymbol{\omega})/\sum_{m=1}^{K}\nu^m(\boldsymbol{\omega}). \tag{3}$$

The Gaussian membership function is often considered as the fuzzy membership function used in (3) which can be formulated as

$$\nu_{\Omega_i^k}(\omega_i) = \exp\left(-1/2((\omega_i - c_i^k)/(\delta_i^k))^2\right), \tag{4}$$

where $\mathbf{c}^k = [c_1^k, c_2^k, \ldots, c_d^k]$ and $\boldsymbol{\delta}^k = [\delta_1^k, \delta_2^k, \ldots, \delta_d^k]$ in each rule represent the kernel center vector and the kernel width vector needed to be learned in the if-parts. Fig.1 shows the framework of a knowledge-driven TSK fuzzy system in which the parts in the shaded rectangle can be considered as knowledge. Generally speaking, parameters involved in each fuzzy rule are different from those of others, hence we call parameters involved in each fuzzy rule special knowledge and accordingly the classical TSK fuzzy system shown in Fig.1 is a special knowledge-driven TSK fuzzy system. However, as the application scenarios we stated in the first section, a special knowledge-driven TSK fuzzy system sometimes cannot solve a real-world problem interpretably.

Usually, the learning process of the classic TSK fuzzy system can be divided into two parts, the if-parts learning and the then-parts learning. Also, they are often achieved in a separate manner. As for the if-parts learning, clustering algorithms [10]–[13] are usually adopted. For example, by introducing FCM [11], c_i^k in $\mathbf{c}^k$ and δ_i^k in $\boldsymbol{\delta}^k$ can be computed by

$$c_i^k = \sum_{j=1}^{N}u_{jk}\omega_{ji}/\sum_{j=1}^{N}u_{jk}, \tag{5}$$

$$\begin{aligned}\delta_i^k &= h\sum_{j=1}^{N}u_{jk}(\omega_{ji}-c_i^k)^2/\sum_{j=1}^{N}u_{jk}, \quad i=1,\ldots,d, \\ &\quad k=1,\ldots,K,\end{aligned} \tag{6}$$

where u_{jk} denotes the fuzzy membership degree of ω_j belonging to the kth cluster, and h is a scale parameter which can be set manually. As for the then-parts learning, the commonly used optimization strategy is the quadratic programming (QP) with different criteria, e.g., the least square criterion [14], ε-insensitive criterion [15] and L1-Norm penalty [15], ε-insensitive criterion and L2-Norm penalty [16] and so on. In additional to QP, the gradient descent-based approaches sometimes are used.

Whether it is QP, gradient descent, or FCM, they are less efficient in the face of large-scale datasets. Therefore, a high-efficiency optimization algorithm is desired in TSK fuzzy system modeling.

B. GAUSSIAN MIXTURE MODEL

GMM (Gaussian mixture model) is one type of mixture distributions where its each component is a normal component. For an arbitrary random variable $\boldsymbol{\omega}$ in d dimensional feature space, the Gaussian mixture probability density function (PDF) [6] can be formulated by

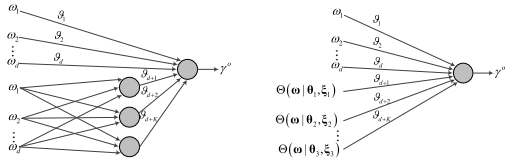
$$\Theta(\boldsymbol{\omega};\boldsymbol{\kappa},\boldsymbol{\theta},\boldsymbol{\Xi}) = \sum_{c=1}^{C}\kappa_c\Upsilon^d(\boldsymbol{\omega};\boldsymbol{\theta}_c,\boldsymbol{\Xi}_c). \tag{7}$$

In (7), C is the total number of components, and $\boldsymbol{\kappa} = [\kappa_1, \kappa_2, \ldots, \kappa_C]$ is a weight vector in which each element represents the weight of each component, where $0 < \kappa_c < 1$, $\sum_{c=1}^{C}\kappa_c = 1$. $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_C]$ is a $d \times C$ matrix in which each element is the mean vector, $\boldsymbol{\Xi} = [\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2, \ldots, \boldsymbol{\Xi}_C]$ is a $d \times Cd$ matrix in which each element $\boldsymbol{\Xi}_c$ denotes a covariance matrix. $\Upsilon^d(\boldsymbol{\omega};\boldsymbol{\theta}_c,\boldsymbol{\Xi}_c)$ is the multivariate (d-dimensional) normal density of the component c, which can be expressed as

$$\begin{aligned}\Upsilon^d(\boldsymbol{\omega};\boldsymbol{\theta}_c,\boldsymbol{\Xi}_c) &= (2\pi)^{-\frac{d}{2}}|\boldsymbol{\Xi}_c|^{-\frac{1}{2}} \\ &\quad \times \exp\left\{-\frac{1}{2}(\boldsymbol{\omega}-\boldsymbol{\theta}_c)^T\boldsymbol{\Xi}_c^{-1}(\boldsymbol{\omega}-\boldsymbol{\theta}_c)\right\},\end{aligned} \tag{8}$$

where $|\boldsymbol{\Xi}_c|$ and $\boldsymbol{\Xi}_c^{-1}$ are the determinant and inverse of $\boldsymbol{\Xi}_c$, respectively.

In [17], the authors demonstrate that radial basis function (RBF) networks are universal approximators. In fact, an RBF network is merely a linear superposition of RBFs, of which Gaussian functions are a particular type. In [18], the authors further prove the ability of RBF networks with



(a) Gaussian FLNN (b) Flat structure

FIGURE 2. Structure of the Gaussian FLNN.

superposition of Gaussian functions under the conditions that the functions to be approximated are imposed with some constraints like non-continuity. Combining the above theoretical analysis results together, it is obvious that a GMM can achieve approximation of any probability distribution with arbitrary accuracy if its parameters are set appropriately.

C. FUNCTIONAL LINK NEURAL NETWORK

FLNN (functional link neural network) is a special single layer neural network in which the hidden layer is replaced by higher-order representations of its input features. FLNN overcomes some disadvantages which are usually contained in multilayer networks such as initial weight dependence, weight inference, saturation and overfitting. Moreover, although FLNN is a single layer neural network, it is still able to handle non-linear separable classification tasks. Basically, the architecture of FLNN is a flat network without any hidden layer, which accordingly makes the parameters learning algorithm less complicated. Many simple learning algorithms, e.g., BP [19], artificial bee colony [20], adaptive learning [21], and pseudoinverse [22] have been proposed for FLNN and its variants learning.

Suppose that the Gaussian function is taken as a high order representation of FLNN, Fig.2(a) illustrates the Gaussian FLNN and the corresponding flat structure is shown in Fig.2(b).

The output of the Gaussian FLNN can be formulated as

$$\gamma^o = \sum_{k=1}^{K} \vartheta_{d+k} \Theta\left(\boldsymbol{\omega}|\boldsymbol{\theta}_k, \boldsymbol{\xi}_k\right) + \vartheta_1\omega_1 + \vartheta_2\omega_2 + \ldots + \vartheta_d\omega_d. \tag{9}$$

III. CSK-TSK-FS: THE PROPOSED COMMON AND SPECIAL KNOWLEDGE-DRIVEN TSK FUZZY SYSTEM

In this section, we will incorporate common knowledge into the classical one-order TSK fuzzy system and accordingly propose the new common and special knowledge-driven TSK fuzzy system CSK-TSK-FS. Simultaneously, we will mathematically analyze its equivalences between GMM and FLNN. Lastly, we present a fast training algorithm for CSK-TSK-FS.

A. ARCHITECTURE OF CSK-TSK-FS

The structure of the proposed fuzzy system CSK-TSK-FS is illustrated in Fig.3, where an input sample in the d dimensional feature space is expressed as $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_d]^T$,

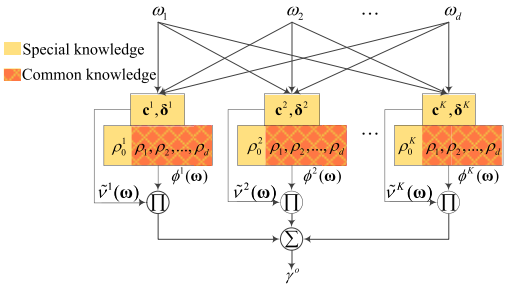


FIGURE 3. Structure of CSK-TSK-FS.

and $\mathbf{c}^k = [c_1^k, c_2^k, \ldots, c_d^k]$ and $\boldsymbol{\delta}^k = [\delta_1^k, \delta_2^k, \ldots, \delta_d^k], k = 1, 2, \ldots, K$ are the kernel center vector and the kernel width vector needed to be learned in the if-parts of each fuzzy rule. Comparing Fig.3 with Fig.1, the distinctive characteristic of CSK-TSK-FS is that there exists a common part (i.e., the parameters $\rho_1, \rho_2, \ldots, \rho_d$) in all fuzzy rules.

Based on the structure illustrated in Fig.3, the kth fuzzy rule of CSK-TSK-FS can be formulated as

$$\begin{aligned} &\text{If} \quad \omega_1 \text{ is } \Omega_1^k \wedge \omega_2 \text{ is } \Omega_2^k \wedge \ldots \wedge \omega_d \text{ is } \Omega_d^k, \\ &\text{then} \quad \phi^k(\boldsymbol{\omega}) = \rho_0^k + \rho_1\omega_1 + \ldots + \rho_d\omega_d, \\ &\quad k = 1, 2, \ldots, K. \end{aligned} \tag{10}$$

In (10), we can see that for each fuzzy rule, the parameters $\rho_1, \rho_2, \ldots, \rho_d$ always keep invariant. We call this common part *common knowledge* in the proposed fuzzy system CSK-TSK-FS. Therefore, compare with the classical TSK fuzzy system shown in Fig.1, which is only driven by special knowledge, CSK-TSK-FS is driven by both special and common knowledge and accordingly becomes more suitable and applicable for simulating the application scenarios. Also, we find that, if $\rho_1, \rho_2, \ldots, \rho_d$ in the then-parts are set to zero, CSK-TSK-FS would degenerate into a classical zero-order TSK fuzzy system. Therefore, we can consider our proposed fuzzy system CSK-TSK-FS as a generalized zero-order TSK fuzzy system. In other words, a zero-order TSK fuzzy system can also be considered as a special case of our proposed fuzzy system CSK-TSK-FS.

As we all know that the interpretability of a TSK fuzzy system can be quantitatively measured by the number of parameters the system needs to learn [23], [24]. During the training process of CSK-TSK-FS, $2Kd$ parameters in the if-parts and $K + d$ in the then-parts need to be learned. Hence, the interpretability of CSK-TSK-FS can be quantitatively measured by $2Kd + K + d$. As for the classical zero-order TSK fuzzy system, during the learning process of the if-parts, it also needs to learn $2Kd$ parameters. But during the learning process of the then-parts, it needs to learn $Kd + K$ parameters. In our application scenarios, K and d are two integer numbers and often bigger than 1, therefore, by comparing CSK-TSK-FS with the classical zero-order TSK fuzzy system, the interpretability of CSK-TSK-FS is improved.

When the if-parts of CSK-TSK-FS are determined, and let

$$\begin{aligned} \boldsymbol{\omega}_g &= \boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_d]^T, && \text{(11a)} \\ \boldsymbol{\rho}_g &= [\rho_1, \rho_2, \ldots, \rho_d]^T, && \text{(11b)} \\ \boldsymbol{v}_g &= [\tilde{v}^1(\boldsymbol{\omega}), \tilde{v}^2(\boldsymbol{\omega}), \ldots, \tilde{v}^K(\boldsymbol{\omega})]^T && \text{(11c)} \end{aligned}$$

and

$$\tilde{\boldsymbol{\rho}}_g = [\rho_0^1, \rho_0^2, \ldots, \rho_0^K]^T, \quad \text{(11d)}$$

where $\tilde{v}^k(\boldsymbol{\omega})$ has been defined in (3), $k = 1, 2, \ldots, K$. By introducing the classical centroid defuzzification strategy [4], the output of CSK-TSK-FS can be defined as

$$\begin{aligned} \gamma^o &= \sum_{k=1}^{K} \left(v^k(\boldsymbol{\omega}) / \sum_{m=1}^{K} v^m(\boldsymbol{\omega}) \right) \phi^k(\boldsymbol{\omega}) \\ &= \sum_{k=1}^{K} \left(v^k(\boldsymbol{\omega}) / \sum_{m=1}^{K} v^m(\boldsymbol{\omega}) \right) (\rho_0^k + \rho_1\omega_1 + \ldots + \rho_d\omega_d) \\ &= \sum_{k=1}^{K} \left(v^k(\boldsymbol{\omega}) / \sum_{m=1}^{K} v^m(\boldsymbol{\omega}) \right) \rho_0^k + (\rho_1\omega_1 + \ldots + \rho_d\omega_d) \\ &= \sum_{k=1}^{K} \left(\tilde{v}^k(\boldsymbol{\omega}) \right) \rho_0^k + (\rho_1\omega_1 + \ldots + \rho_d\omega_d) \\ &= \tilde{\boldsymbol{\rho}}_g^T \boldsymbol{v}_g + \boldsymbol{\rho}_g^T \boldsymbol{\omega}_g. \end{aligned} \quad \text{(12)}$$

The output defined in (12) indeed reveals a notable merit that the common knowledge can be *independent of* each fuzzy rule. That is to say, with the classical centroid defuzzification strategy, each fuzzy rule can be implicitly shortened as

$$\begin{aligned} &\text{If} \quad \omega_1 \text{ is } \Omega_1^k \wedge \omega_2 \text{ is } \Omega_2^k \wedge \ldots \wedge \omega_d \text{ is } \Omega_d^k, \\ &\text{then} \quad \phi^k(\boldsymbol{\omega}) = \rho_0^k, \quad k = 1, 2, \ldots, K. \end{aligned}$$

Therefore, the interpretability of CSK-TSK-FS can be further enhanced from the perspective of the rule length [49], [50]. By comparing the expressions in (12) and (9), we can easily find that the proposed fuzzy system CSK-TSK-FS is equivalent to FLNN as a matter of fact. On the contrary, FLNN also can be considered as a special fuzzy system such that it no longer works in a black way. To the best of our knowledge, this is the first attempt that we reveal the relationship between fuzzy systems and FLNN. The common knowledge denoted as $\boldsymbol{\rho}_g$ contributes the linear approximator $\boldsymbol{\rho}_g^T \boldsymbol{\omega}_g$ to the second term of the output of CSK-TSK-FS. Moreover, the relationship between CSK-TSK-FS and GMM will also be theoretically analyzed in the next subsection.

B. FROM GMM TO CSK-TSK-FS

Theoretically, GMM can approximate any probability distribution to arbitrary accuracy [17], [18] such that it can be taken as a high-efficiency approximator. Suppose $\chi = \{\boldsymbol{\omega}_i, \gamma_i | \boldsymbol{\omega}_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}]^T \in R^d, \gamma_i \in R, i = 1, 2, \ldots, N\}$ is a training dataset for a Gaussian mixture model who contains C components. With the training dataset χ, the means $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_C)$ and covariance matrices $(\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2, \ldots, \boldsymbol{\Xi}_C)$ of all components can be obtained by a training algorithm, e.g., EM [30], where $\boldsymbol{\theta}_c = [\boldsymbol{\theta}_{c\boldsymbol{\omega}}, \theta_{c\gamma}]$. If we use τ_{cab} and τ^{cab} to denote the elements in $\boldsymbol{\Xi}_c$ and its inverse $\boldsymbol{\Xi}_c^{-1}$, respectively, then $\boldsymbol{\Xi}_c$ and $\boldsymbol{\Xi}_c^{-1}$ can be expressed as

$$\begin{aligned} \boldsymbol{\Xi}_c &= \begin{bmatrix} \boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}} & \boldsymbol{\tau}_{c\boldsymbol{\omega}\gamma} \\ \boldsymbol{\tau}_{c\gamma\boldsymbol{\omega}} & \boldsymbol{\tau}_{c\gamma\gamma} \end{bmatrix} \\ &= \begin{bmatrix} \{\tau_{cij}\}_{d\times d} & \{\tau_{cj(d+1)}\}_{d\times 1} \\ \{\tau_{c(d+1)j}\}_{1\times d} & \tau_{c(d+1)(d+1)} \end{bmatrix}, \end{aligned} \quad \text{(13)}$$

where $i, j = 1, 2, \ldots, d$. $\boldsymbol{\Xi}_c$ is a symmetric matrix, hence $\boldsymbol{\tau}_{c\boldsymbol{\omega}\gamma}$ is equal to $\boldsymbol{\tau}_{c\gamma\boldsymbol{\omega}}^T$. Thus, $\boldsymbol{\Xi}_c^{-1}$ can be expressed as the following form,

$$\boldsymbol{\Xi}_c^{-1} = \begin{bmatrix} \boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}} & \boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma} \\ \boldsymbol{\tau}^{c\gamma\boldsymbol{\omega}} & \tau^{c\gamma\gamma} \end{bmatrix}. \quad \text{(14)}$$

It is obvious that $\boldsymbol{\tau}_{c\boldsymbol{\omega}\gamma}$ in $\boldsymbol{\Xi}_c$ reveals the correlation degree between $\boldsymbol{\omega}$ and γ in component c, and $\tau_{c\gamma\gamma}$ in $\boldsymbol{\Xi}_c$ reveals the correlation degree between γs in component c. In many cases, we are generally uninformative for each component in advance, hence, a mild assumption that $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}/\tau^{c\gamma\gamma}$ keeps invariant for each component may be considered. That is to say, $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}/\tau^{c\gamma\gamma} = \boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_d]^T$ is a constant vector for each component. Therefore, the joint PDF of $(\boldsymbol{\omega}, \gamma)$ can be approximated by a special GMM approximator,

$$\begin{aligned} \Theta(\boldsymbol{\omega}, \gamma) &= \sum_{c=1}^{C} \kappa_c \Upsilon^{d+1} \left(\begin{bmatrix} \boldsymbol{\omega} \\ \gamma \end{bmatrix}; \begin{bmatrix} \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \theta_{c\gamma} \end{bmatrix}, \boldsymbol{\Xi}_c \right), \\ s.t.\ \boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}/\tau^{c\gamma\gamma} &= \boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_d]^T, \end{aligned} \quad \text{(15)}$$

where $\boldsymbol{\theta}_{c\boldsymbol{\omega}} = [\theta_{c1}, \theta_{c2}, \ldots, \theta_{cd}]^T$.

By the approximator $\Theta(\boldsymbol{\omega}, \gamma)$ trained by χ, the output γ^o for an unseen sample $\boldsymbol{\omega}$ can be formulated as,

$$\begin{aligned} \gamma^o &= E[\gamma|\boldsymbol{\omega}] = \int_{-\infty}^{+\infty} \gamma \Theta(\gamma|\boldsymbol{\omega}) d\gamma \\ &= \frac{\int_{-\infty}^{+\infty} \gamma \Theta(\boldsymbol{\omega}, \gamma) d\gamma}{\Theta(\boldsymbol{\omega})}. \end{aligned} \quad \text{(16)}$$

Since $\Theta(\boldsymbol{\omega}, \gamma) = \Theta_1(\boldsymbol{\omega}, \gamma) + \Theta_2(\boldsymbol{\omega}, \gamma) + \ldots + \Theta_C(\boldsymbol{\omega}, \gamma)$, we can obtain $\Theta(\boldsymbol{\omega}) = \int_{-\infty}^{+\infty} \Theta(\boldsymbol{\omega}, \gamma) d\gamma = \sum_{c=1}^{C} \int_{-\infty}^{+\infty} \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma$. Hence, (16) can be re-organized as

$$\gamma^o = E[\gamma|\boldsymbol{\omega}] = \frac{\sum_{c=1}^{C} \int_{-\infty}^{+\infty} \gamma \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma}{\sum_{c=1}^{C} \int_{-\infty}^{+\infty} \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma}. \quad \text{(17)}$$

In (17), $\int_{-\infty}^{+\infty} \gamma \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma$ can be expanded as (see the Appendix)

$$\begin{aligned}\int_{-\infty}^{+\infty} & \gamma \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma \\ = & \frac{\kappa_c}{(2\pi)^{\frac{d}{2}} \sqrt{|\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}|}} \\ & \times \exp\left\{-\frac{1}{2}\left([\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{x}}]^T (\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})^{-1} [\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{x}}]\right)\right\} \\ & \times \left(\theta_{c\gamma} - [\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{x}}]^T \boldsymbol{\Psi}\right). \end{aligned} \tag{18}$$

Component c of the GMM approximator can be formulated as

$$\begin{aligned}\Theta_c(\boldsymbol{\omega}, \gamma) &= \kappa_c \Upsilon^{d+1}\left(\begin{bmatrix} \boldsymbol{\omega} \\ \gamma \end{bmatrix}; \begin{bmatrix} \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \theta_{c\gamma} \end{bmatrix}, \boldsymbol{\Xi}_c\right) \\ &= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}} \sqrt{|\boldsymbol{\Xi}_c|}} \exp\left\{-\frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \gamma - \theta_{c\gamma} \end{bmatrix}^T \right. \\ &\quad \left. \times \boldsymbol{\Xi}_c^{-1} \begin{bmatrix} \boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \gamma - \theta_{c\gamma} \end{bmatrix}\right\}. \end{aligned} \tag{19}$$

Similarly, like the derivation procedures in the Appendix, the marginal PDF of $\boldsymbol{\omega}$ for component c of $\Theta(\boldsymbol{\omega}, \gamma)$ can be formulated as

$$\begin{aligned}\Theta_c(\boldsymbol{\omega}) &= \int_{-\infty}^{+\infty} \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma \\ &= \int_{-\infty}^{+\infty} \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}} \sqrt{|\boldsymbol{\Xi}_c|}} \\ &\quad \times \exp\left\{-\frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \gamma - \theta_{c\gamma} \end{bmatrix}^T \boldsymbol{\Xi}_c^{-1} \begin{bmatrix} \boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \gamma - \theta_{c\gamma} \end{bmatrix}\right\} d\gamma \\ &= \frac{\kappa_c}{(2\pi)^{\frac{d}{2}} \sqrt{|\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}|}} \\ &\quad \times \exp\left\{-\frac{1}{2}\left([\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T (\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})^{-1} [\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right)\right\} \\ &= \kappa_c \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c\boldsymbol{\omega}}, \boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}). \end{aligned} \tag{20}$$

Therefore, the marginal PDF of $\boldsymbol{\omega}$ can be deduced as

$$\begin{aligned}\Theta(\boldsymbol{\omega}) &= \int_{-\infty}^{+\infty} \Theta(\boldsymbol{\omega}, \gamma) d\gamma \\ &= \sum_{c=1}^{C} \int_{-\infty}^{+\infty} \kappa_c \Upsilon^{d+1}\left(\begin{bmatrix} \boldsymbol{\omega} \\ \gamma \end{bmatrix}; \begin{bmatrix} \boldsymbol{\theta}_{c\boldsymbol{\omega}} \\ \theta_{c\gamma} \end{bmatrix}, \boldsymbol{\Xi}_c\right) d\gamma \\ &= \sum_{c=1}^{C} \kappa_c \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c\boldsymbol{\omega}}, \boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}). \end{aligned} \tag{21}$$

By substituting (18) and (21) into (17), the expected output γ^o for the unseen sample $\boldsymbol{\omega}$ can be re-organized as

$$\begin{aligned}\gamma^o &= \sum_{c=1}^{C} \frac{\kappa_c \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c\boldsymbol{\omega}}, \boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})}{\sum_{c'=1}^{C} \kappa_{c'} \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c'\boldsymbol{\omega}}, \boldsymbol{\tau}_{c'\boldsymbol{\omega}\boldsymbol{\omega}})} (\theta_{c\gamma} - [\boldsymbol{\omega} - \boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T \boldsymbol{\Psi}) \\ &= \sum_{c=1}^{C} \frac{\kappa_c \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c\boldsymbol{\omega}}, \boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})}{\sum_{c'=1}^{C} \kappa_{c'} \Upsilon^d(\boldsymbol{\omega}; \boldsymbol{\theta}_{c'\boldsymbol{\omega}}, \boldsymbol{\tau}_{c'\boldsymbol{\omega}\boldsymbol{\omega}})} (\theta_{c\gamma} + \boldsymbol{\theta}_{c\boldsymbol{\omega}}^T \boldsymbol{\Psi}) \\ &\quad + (-\Psi_1\omega_1 - \cdots - \Psi_d\omega_d). \end{aligned} \tag{22}$$

In (22), with the assumption that the each feature in $\boldsymbol{\omega}$ is mutually independent, we can express the output γ^o as the following form,

$$\gamma^o = \sum_{c=1}^{C} \tilde{\Upsilon}_c(\boldsymbol{\omega})(\theta_{c\gamma} + \boldsymbol{\theta}_{c\boldsymbol{\omega}}^T \boldsymbol{\Psi}) + (-\Psi_1\omega_1 - \cdots - \Psi_d\omega_d), \tag{23}$$

where $\tilde{\Upsilon}_c(\boldsymbol{\omega}) = \kappa_c \prod_{j=1}^{d} \Upsilon^j\left(\omega_j; \theta_{cj}, \tau_{cjj}\right) / \sum_{c'=1}^{C} \tau_{c'} \prod_{j=1}^{d} \Upsilon^j\left(\omega_j; \theta_{c'j}, \tau_{c'jj}\right)$.

After comparing the output γ^o in (23) with that in (9) or (12), we can easily find that the GMM approximator with the assumption $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}/\boldsymbol{\tau}^{c\gamma\gamma} = \boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_d]^T$ is equivalent to CSK-TSK-FS (also FLNN) where each component in GMM can be taken as a fuzzy rule in CSK-TSK-FS in which $\Upsilon^j\left(\omega_j; \theta_{cj}, \tau_{cjj}\right)$ is considered as the fuzzy membership function. Here, please note that κ_c in GMM should be uniform, i.e., κ_c should be set to $1/K$. The common knowledge denoted as $[-\Psi_1, -\Psi_2, \ldots, -\Psi_d]$ contributes the linear approximator $(-\Psi_1\omega_1 - \cdots - \Psi_d\omega_d)$ in (23).

Based on the above analysis, the relationship between CSK-TSK-FS and the special GMM approximator indicates the following three results:

1) The training of CSK-TSK-FS can be achieved by using a density estimation algorithm, e.g., EM [30] to train a special Gaussian mixture model.
2) With the relationship between CSK-TSK-FS and GMM, CSK-TSK-FS can be interpreted from the perspective of probability statistics. Therefore, some statistical tools for GMM can also be applied to CSK-TSK-FS. For example, we know that the number of fuzzy rules in CSK-TSK-FS is equal to the number of components in GMM, hence, many useful tools for searching the optimal number of components can be used for searching the optimal number of fuzzy rules.
3) The promising approximation ability of CSK-TSK-FS can be insured since GMM is a global approximator.

C. TRAINING ALGORITHM OF CSK-TSK-FS

By giving a training set $\chi = \{\boldsymbol{\omega}_i, \gamma_i | \boldsymbol{\omega}_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}]^T \in R^d, i = 1, 2, \ldots, N\}$, the training of CSK-TSK-FS can be achieved by many criteria [14]–[16]. For example, with the determined if-parts, we can employ the gradient descent algorithm [26] to minimize the error criterion function $(1/2)\sum_{i=1}^{N}\left(\gamma_i^o - \gamma_i\right)^2$.

In addition to gradient descent, the optimization of CSK-TSK-FS can also be considered as a QP problem that

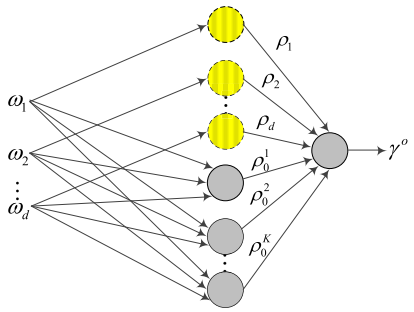


FIGURE 4. Network of CSK-TSK-FS.

can be solved by QP-based learning [14]–[16]. Although gradient descent-based algorithms and QP-based algorithms are easy to implement, both of them consume many CPU seconds for large-scale datasets. Moreover, clustering techniques used in the if-parts learning also consume many CPU seconds for large-scale datasets. Therefore, a fast training algorithm for CSK-TSK-FS is desired. Since the equivalence relationship between CSK-TSK-FS and GMM and FLNN, some of the training algorithms for GMM and FLNN, e.g., EM [30] for GMM, BP [19] artificial bee colony [20], adaptive learning [21], and pseudoinverse for FLNN can also be used for CSK-TSK-FS. However, in this study, we propose a new fast training algorithm for CSK-TSK-FS, which can be also used for GMM and FLNN.

In [35], the authors demonstrate that the modeling of a TSK fuzzy system can be replaced by modeling a fuzzy neural network. Since CSK-TSK-FS is indeed a TSK fuzzy system, it also can be considered as a fuzzy neural network, see in Fig.4.

Essentially, CSK-TSK-FS is a special TSK fuzzy system. Our previous work [35] reveals that a TSK fuzzy system can be equivalent to a fuzzy neural network, so CSK-TSK-FS can also be considered as a special neural network shown in Fig.4, where only one feature is involved in each yellow node of the hidden layer. In [8], [9], authors demonstrate that the optimization of a single-layer feedforward neural network is equivalence to solving a ridge regression problem that can be fast solved by the least learning machine (LLM). Therefore, obviously, the neural network in Fig.4 can also be fast solved by LLM only by augmenting original input features into the hidden layer.

Therefore, the solution of CSK-TSK-FS in Fig.4 can be formulated as

$$\begin{aligned} &\min \left(\frac{1}{2}\boldsymbol{\beta}^2 + C\sum_{i=1}^{N}\zeta_i^2\right), \\ &s.t.\ (\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_1), \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_2), \ldots, \\ &\quad \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_K))\boldsymbol{\beta}^T \\ &\quad = \gamma_i + \zeta_i, \quad i = 1, 2, \ldots, N, \end{aligned} \tag{24}$$

where $\boldsymbol{\beta} = [\rho_1, \rho_2, \ldots, \rho_d, \rho_0^1, \rho_0^2, \ldots, \rho_0^K]$ represents the weight vector for the output of CSK-TSK-FS, $\varphi(\bullet)$ is the activation function used for feature mapping, γ_i is training label of sample $\boldsymbol{\omega}_i$ and C is a given constant.

As for the ridge regression problem in (24), its analytical solution $\tilde{\boldsymbol{\beta}}$ for the weight vector $\boldsymbol{\beta}$ can be derived as,

$$\tilde{\boldsymbol{\beta}} = \mho^T\left(\mho\mho^T + \frac{1}{2C}\mathbf{I}\right)^{-1}\boldsymbol{\gamma}, \tag{25}$$

where $\mathbf{I}$ is an N by N identity matrix, $\mho = [\mho_1, \mho_2, \ldots, \mho_N]^T$, $\mho_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_1), \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_2), \ldots, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_K)]$ and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_N]^T$ is the output of the training set.

Different from the BP-like learning algorithms [32]–[34], only parameters in the output layer need to be trained in LLM. Therefore, LLM can achieve fast learning of such a SLFN illustrated in Fig.4.

Detailed steps of CSK-TSK-FS training are listed in Algorithm 1.

Algorithm 1 Fast Training of CSK-TSK-FS

Input: Training dataset $\chi = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_N]^T$, training label dataset $\boldsymbol{\gamma}_{actual} = [\gamma_{actual}^1, \gamma_{actual}^2, \ldots, \gamma_{actual}^N]^T \in R^N$, where $\boldsymbol{\omega}_i \in R^d$, $\gamma_{actual}^i \in R$, $i = 1, 2, \ldots\ldots, N$, and number of fuzzy rules K.

Output: Weight vector $\boldsymbol{\beta} = [\rho_1, \rho_2, \ldots, \rho_d, \rho_0^1, \rho_0^2, \ldots, \rho_0^K]$.

Procedure:

Step 1: Randomly assign the parameter vectors $\boldsymbol{\theta}_k = [c_k, \sigma_k]$ in the Gaussian functions $\varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_1), \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_2), \ldots, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_K)$, where c_k and σ_k represent the centers and kernel widths of Gaussian functions, respectively, $k = 1, 2, \ldots, K$.

Step 2: Construct $= [\mho_1, \mho_2, \ldots, \mho_N]^T$, where $_1$ can be expressed as $\mho_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_1), \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_2), \ldots, \varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_K)]$, and $\varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_k)$ can be computed by $\varphi(\boldsymbol{\omega}_i, \boldsymbol{\sigma}_k) = \prod_{j=1}^{d}\exp\left(-1/2((\omega_{ij} - c_j^k)/(\delta_j^k))^2\right)$.

Step 3: Compute the output weight by using LLM, i.e.,

$$\tilde{\boldsymbol{\beta}} = \left(\frac{1}{2C}I + \mho^T\mho\right)^{-1}\mho\boldsymbol{\gamma}, \tag{26}$$

where I is an $(K + d)$ by $(K + d)$ identity matrix.

When the weight vector for the output layer is determined, CSK-TSK-FS can make a prediction for an unseen sample. Next, we give some remarks about the fast learning algorithm listed in Algorithm 1.

Remark 1: In this fast training algorithm, the parameters in the if-parts are randomly assigned rather than obtained by clustering techniques. The effectiveness of the randomness strategy has been demonstrated in ELM [35]. Comparing with clustering techniques, undoubtedly, the randomness strategy can significantly reduce the CPU seconds consuming. Moreover, with the high-efficiency analytical solution to the

then-parts learning in (25), the CPU seconds consuming is also reduced compared with BP-like algorithms in which all parameters in the network need to be iteratively adjusted in a backward gradient descent way.

Remark 2: As all we know that the inverse computation of a matrix becomes very time-consuming when the number of elements is very huge. So, the solution in (25) still becomes out of service for large-scale datasets. However, with the second property in [8] about LLM, the analytical solution of LLM in (25) can be re-organized in another form in (26). By the new analytical solution, the time complexity is independent on the number of samples, it now only dependents on the number of fuzzy rules K and the number of features d. For large-scale datasets, K and d are very smaller than N. Hence, the time complexity is significantly reduced.

Remark 3: In (26), C is a user-dependent parameter. According to our previous work [35], it is set to a comparatively large value, e.g., 200 in our following experiments.

IV. EXPERIMENTAL RESULTS

In this section, CSK-TSK-FS is mainly evaluated from two aspects: its classification ability on UCI and KEEL datasets and its application for epileptic EEG signals recognition. In addition, in order to highlight the performance of CSK-TSK-FS, several benchmarking approaches including SVM (with the linear kernel and the Gaussian kernel, respectively) [39], FS-FCSVM [24], zero-order-TSK-FS [4], GFS-AdaBoost-C [38], FH-GBML-C [36], [37] and L2-TSK-FS [4] are introduced for comparison studies. The following experiments are organized as: subsection IV.A gives the experimental setups, subsection IV.B shows the experimental results on UCI and KEEL datasets, and subsection IV.C gives an application for epileptic EEG signals recognition.

A. EXPERIMENTAL SETUP

With regards to the all introduced benchmarking approaches, SVM, FS-FCSVM, zero-order-TSK-FS and L2-TSK-FS are coded in the MATLAB platform, while FH-GBML-C and GFS-AdaBoost-C are provided by the KEEL toolbox [40]. L2-TSK-FS, zero-order-TSK-FS, and the proposed approach CSK-TSK-FS are originally designed for regression problems. In our experiments, according to [41], all of them can be trained for classification tasks by considering the class labels of the training set as their regression values. For an unseen object, they predict its label as the one which is nearest to their outputs.

We use the default parameters provided by the KEEL toolbox to fix FH-GBML-C and GFS-AdaBoost-C. As for the remnant approaches, 20% validation objects are used to find the optimal parameters by 10-fold CV strategy. Table 1 gives the trial intervals for CV in the corresponding approaches. After we get the optimal parameters of each approach, 70% objects are selected for training and 10% objects are selected for testing. The results are reported in terms of the average testing accuracy (including the corresponding standard deviation) and the maximum testing accuracy for 30 trials.

TABLE 1. The trial intervals for CV in the corresponding approaches.

L2-TSK-FC zero-order-TSK-FC	Number of fuzzy rules $K \in \{2,3,...,15\}$ [41], fuzzy index $m \in \{1.1,1.5,2,2.5,3\}$, scale parameter $h \in \{0.2^2,0.4^2,0.8^2,1^0,1.6^2,2^2\}$.
FS-FCSVM	Learning threshold parameter $\sigma \in \{0.2,0.3,...,0.8\}$, regularization parameter $C \in \{2^{-3},2^{-1},2^0,...,2^5,2^7\}$, number of rules $K \in \{2,3,...,15\}$.
SVM	Regularization parameter $C \in \{10^{-5},...,10^0,...,10^5\}$, Gaussian kernel width parameter $\sigma \in \{10^{-5},...,10^0,...,10^5\}$.
CSK-TSK-FS	Number of fuzzy rules $K \in \{2,3,...,15\}$.

TABLE 2. Detailed information of selected UCI and KEEL datasets.

Datasets	#Features	#Objects	#Classes
Diabetes	8	768	2
Australia	14	690	2
Breast	10	699	2
Adult	14	48841	2
Musk	166	6598	2
Magic04	11	19020	2
Sonar	60	208	2
Vote	16	435	2
Skin-Segmentation	4	245057	2
Liver	6	345	2
Seismic-bumps	18	2584	2
Landsat	36	2000	6
Balance	4	625	3
Page_blocks	10	5473	5
Monk2	6	601	2
Kddcup99	41	494021	23

The experiments are conducted on a personal computer with 4 cores of I5-7200U with 64G Bytes of memory.

B. ON UCI AND KEEL DATASETS

Since UCI [45] and KEEL [40] are two commonly used repositories for verifying machine learning approaches, we select sixteen real-life datasets including binary-class and multi-class to verify the classification performance of CSK-TSK-FS. Table 2 shows the detailed information of the selected datasets, in which some medium scale (i.e., *Adult*, *Magic04*) and large scale (i.e., *Skin-Segmentation*,*Kddcup99*) datasets are used to observe the CPU seconds consuming of CSK-TSK-FS.

Table 3 reports the classification performance of all approaches in terms of different criteria, i.e., ''Max'' represents the maximum accuracy of 30 trials, ''Rules'' represents the optimal number of fuzzy rules obtained by CV, ''Mean'' represents the average accuracy of 30 trials, and ''Std'' represents the standard deviation. Although GFS-AdaBoost-C is also a fuzzy rule-based classifier, the number of fuzzy rules is not provided by the KEEL toolbox. Therefore, we

TABLE 3. The classification performance on UCI and KEEL datasets.

Datasets	FH-GBML-C		GFS-AdaBoost-C		zero-order-TSK-FC		L2-TSK-FC		FS-FCSVM		SVM (Linear)		SVM (Gaussian)		CSK-TSK-FS	
	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std	Max Mean	Rules Std
Diabetes	0.7396 0.7109	81 0.0197	0.7325 0.7139	--- 0.0255	0.7348 0.6721	**2** 0.0362	0.6870 0.6678	**2** 0.0125	0.6913 0.6704	**2** 0.0245	0.7783 0.7339	--- 0.0279	0.7696 0.7478	--- 0.0177	**0.8217** **0.7861**	**2** 0.0294
Australia	0.8720 0.8561	48 0.0126	0.8197 0.8037	--- 0.0155	0.8309 0.8135	**2** 0.0169	0.8502 0.7704	**2** 0.1608	0.8599 0.7353	**2** 0.1888	0.8213 0.7845	--- 0.0321	0.8261 0.7893	--- 0.0212	**0.8889** **0.8705**	5 0.0147
Breast	0.9587 0.9510	83 0.0056	0.9542 0.9427	--- 0.0572	0.9761 0.9543	3 0.0198	0.9667 0.9514	3 0.0189	0.9762 0.9610	**2** 0.0170	0.9571 0.9381	--- 0.0168	0.9523 0.9381	--- 0.0135	**0.9810** **0.9638**	3 0.0137
Adult	0.8206 0.8158	65 0.0042	0.8326 0.8180	--- 0.0120	0.7753 0.7576	14 0.0030	0.8123 0.8001	**13** 0.0020	0.7654 0.7623	15 0.0024	**0.8395** 0.8378	--- 0.00	0.8535 **0.8523**	--- 0.0010	0.8333 0.8300	14 0.0028
Musk	0.2396 0.1646	67 0.0562	0.1545 0.1541	--- 3.3e-4	0.8550 0.8463	13 0.0063	0.8509 0.8437	**12** 0.0057	0.8484 0.8398	14 0.0053	**0.9964** **0.9928**	--- 0.0032	0.8479 0.8466	--- 0.0013	0.8499 0.8476	14 0.0021
Magic04	0.8233 0.8151	86 0.0069	0.8155 0.8038	--- 0.0072	0.6538 0.6456	14 0.0012	0.7489 0.7432	11 0.0051	0.6539 0.6469	**2** 0.0070	0.6498 0.6489	--- 0.0011	0.6538 0.6484	--- 0.0037	**0.8590** **0.8460**	15 0.0098
Sonar	0.4808 0.4087	66 0.0797	0.4807 0.4663	--- 0.0083	0.6129 0.5258	3 0.0631	0.8047 0.761	10 0.0302	0.5968 0.5355	**2** 0.0402	0.5968 0.5258	--- 0.0588	**0.8226** **0.7968**	--- 0.0270	0.7904 0.7873	**2** 0.0033
Vote	0.9633 0.9080	43 0.0364	0.6238 0.5331	--- 0.0879	0.8846 0.8477	**2** 0.0252	0.9000 0.8815	3 0.0208	0.8692 0.8615	**2** 0.0054	0.9538 0.9354	--- 0.0185	0.9462 0.9323	--- 0.0126	**0.9615** **0.9462**	**2** 0.0126
Skin-Segmentation	0.9521 0.9267	85 0.0221	0.9388 0.9079	--- 0.0502	0.8906 0.8566	12 0.0478	0.9045 0.8990	13 0.0078	0.9633 0.9231	15 0.0232	0.9698 **0.9549**	--- 0.0277	0.9771 0.9478	--- 0.0210	**0.9796** 0.9399	**9** 0.0287
Liver	0.6913 0.6508	73 0.0423	0.6909 0.6501	--- 0.0367	0.4807 0.4231	8 0.0340	0.5480 0.4962	3 0.0376	0.5769 0.5615	15 0.0109	0.7211 0.6962	--- 0.0251	0.6941 0.6456	--- 0.0312	**0.7019** **0.6596**	3 0.0388
Seismic-bumps	0.9349 0.9326	26 0.0017	0.9303 0.9256	--- 0.0743	**0.9410** 0.9381	7 0.0020	0.9251 0.9321	9 0.0014	0.9168 0.9023	9 0.0011	0.9292 0.2387	--- 0.3928	0.9349 **0.9383**	--- 0.0019	0.9342 0.9308	**3** 0.0054
Landsat	0.5556 0.5509	54 0.0010	0.7595 0.7363	--- 0.0618	0.4317 0.2910	5 0.0788	0.3617 0.3240	**2** 0.0242	0.8050 0.7727	13 0.0352	**0.8800** **0.8693**	--- 0.0137	0.8733 0.8623	--- 0.0093	0.8333 0.8237	15 0.0105
Balance	0.8653 0.8511	96 0.0184	0.7115 0.6784	--- 0.0198	0.8354 0.8025	5 0.0521	0.5882 0.5519	**4** 0.0237	0.8021 0.7294	**4** 0.0872	**1.0000** **0.9947**	--- 0.0065	0.9178 0.8863	--- 0.0304	0.9198 0.8941	5 0.0187
Page_blocks	0.9393 0.9382	56 0.0011	0.9044 0.8995	--- 0.0039	0.9044 0.8995	14 0.0039	0.9074 0.8985	15 0.0089	0.9038 0.8971	15 0.0075	**0.9744** **0.9697**	--- 0.0057	0.9542 0.9478	--- 0.0051	0.9330 0.9153	**12** 0.0201
Monk2	0.6623 0.6485	73 0.0125	0.6667 0.6438	--- 0.2061	0.7278 0.6722	15 0.0322	0.6222 0.5833	15 0.0269	0.7111 0.6778	**4** 0.0380	0.6751 0.6283	--- 0.0464	0.6711 0.6420	--- 0.0334	**0.7400** **0.6989**	12 0.0486
Kddcup99	0.4123 0.3691	92 0.0358	0.5189 0.4892	--- 0.0174	0.3998 0.3771	**10** 0.0222	0.4279 0.4006	14 0.0301	0.4345 0.4028	12 0.0210	0.5167 0.5098	--- 0.0087	**0.5532** **0.5328**	--- 0.0198	0.5388 0.5045	12 0.0300

use "—" to represent the number of fuzzy rules in Table 3. Next, we will contrastively analysis the results from the perspectives of classification performance and interpretability.

1) CSK-TSK-FS wins the best average accuracy and the maximum accuracy in 7 and 8 out of the 16 UCI and KEEL datasets. As for some datasets, CSK-TSK-FS performs a little worse than other benchmarking approaches, e.g., Gaussian kernel based SVM on *Adult*, *Sonar*, *Seismic-bumps*and*Kddcup99*, linear kernel based SVM on *Musk*, *Skin-Segmentation*, *Balance*and*Page_blocks*. However, we should keep in mind that CSK-TSK-FS is interpretable while SVMs work in a black-box way. Moreover, by comparing CSK-TSK-FS with zero-order-TSK-FC, L2-TSK-FC and FS-FCSVM, we find that CSK-TSK-FS often wins the best classification performance which indicates that the improved generalization capability of CSK-TSK-FS is insured in contrast to the similar approaches. In fact, the promising performance of CSK-TSK-FS on most datasets indicates that it indeed inherits the good approximation ability of GMM.
2) We know that the number of fuzzy rules is relative to the interpretability of a fuzzy system. From Table 3, we can see that, in some cases, the number of fuzzy rules CSK-TSK-FS used is more than that zero-order-TSK-FC, L2-TSK-FC or FS-FCSVM used. For example, on the dataset *Balance*, 5 fuzzy rules are identified by CSK-TSK-FS to get its best performance. As for L2-TSK-FC and FS-FCSVM, both of them need 4 fuzzy rules to get their best performance, respectively. Therefore, it seems that comparing with L2-TSK-FC and FS-FCSVM, the interpretability of CSK-TSK-FS is reduced. However, the interpretability is also high relative to the number of parameters involved in fuzzy rules. As for *Balance*, L2-TSK-FC and FS-FCSVM need to train $2 \times 4 \times 4 + (4 + 4 \times 4) = 52$

TABLE 4. The CPU seconds each approach consumes on UCI and KEEL datasets.

Datasets	zero-order-TSK-FC		L2-TSK-FC		FS-FCSVM		SVM (Linear)		SVM (Gaussian)		CSK-TSK-FS	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Diabetes	1.4460	**<1e-4**	1.3556	**<1e-4**	0.0348	0.0008	1.2822	0.0066	1.6664	0.0022	**<1e-4**	**<1e-4**
Australia	1.1186	0.0008	0.8696	0.0008	0.0308	0.0008	0.2728	0.0030	0.3836	0.0020	**0.0014**	**0.0006**
Breast	0.9670	0.0008	0.9524	0.0002	0.0204	0.0008	0.0250	0.0000	0.0196	0.0006	**0.0010**	**<1e-4**
Adult	409.1280	0.0789	552.7532	0.0388	305.852	0.0388	2987.52	3.2578	3433.4688	5.9692	**0.0858**	**0.0348**
Musk	22.0606	0.0536	52.57	0.1054	6.4544	0.0474	2.2270	0.1752	2.5440	0.1842	**0.1294**	**0.0554**
Magic04	12.5378	0.0041	14.9817	0.0056	9.6826	0.0012	6.0254	1.6116	6.5208	1.5810	**0.0064**	**0.0012**
Sonar	0.0942	0.0008	0.07562	0.0025	0.0118	0.0000	0.0032	0.0030	0.0062	0.0032	**0.0014**	**0.0006**
Vote	0.3694	0.0006	0.397	0.0006	0.0100	0.0006	0.0032	**<1e-4**	0.0030	**<1e-4**	**0.0008**	**<1e-4**
Skin-Segmentation	5.165e3	30.2482	4.235e3	21.785	3.336e3	19.785	2.236e2	9.7892	1.956e2	10.235	**8.6598**	**0.898**
Liver	0.2992	0.0006	0.2702	0.0004	0.0770	0.0006	0.2164	0.0000	0.272	**<1e-4**	**0.0010**	**<1e-4**
Seismic-bumps	1.5428	0.0008	1.4200	0.0006	0.4520	0.0016	0.0032	0.0032	0.0068	0.0002	**0.0026**	**0.0006**
Landsat	14.7064	0.0010	14.5092	0.0010	0.8626	0.0018	0.3312	0.0212	0.2848	0.0180	**0.0048**	**0.0004**
Balance	1.0354	0.0004	0.9408	0.0008	0.1486	0.0010	0.0156	0.003	0.0154	**<1e-4**	**0.0018**	**<1e-4**
Page_blocks	10.7078	0.0022	10.6958	0.0064	0.9034	0.0020	0.7436	0.0216	0.7728	0.0186	**0.0046**	**0.0016**
Monk2	0.7468	0.0008	0.8122	0.0010	0.0264	0.0006	1.2784	0.0000	0.894	**<1e-4**	**0.0014**	**<1e-4**
Kddcup99	5.364e5	7.889e3	4.269e5	4.781e3	9.654e5	6.358e3	1.256e5	8.265e2	1.568e5	9.589e2	**984.256**	**45.628**

parameters, while CSK-TSK-FS needs $2 \times 5 \times 4 + (5 + 4) = 49$ because of the common knowledge being involved. This phenomenon indicates that common knowledge involved in CSK-TSK-FS can improve the interpretability because it shortens the length of fuzzy rules implicitly.

In Table 4, we report the CPU seconds each approach consumes during the testing and training procedures. From careful observation from Table 4, we find that CSK-TSK-FS perform more efficient than other benchmarking approaches, especially for medium-scale or large-scale datasets (e.g., *Magic04*, *Adult*, *Skin-Segmentation*, and *Kddcup99*).

From the experimental results on UCI and KEEL datasets, we can draw the following conclusions.

1) With the common knowledge, CSK-TSK-FS becomes FLNN and hence a single layer fuzzy natural network. Thus, it can be considered as a ridge regression problem that can be fast solved by LLM.
2) With the common knowledge, the length of fuzzy rules can be implicitly reduced such that the interpretability is improved.

C. APPLICATION FOR EPILEPTIC EEG RECOGNITION

Here, we use a medical dataset, i.e., the epileptic EEG data to demonstrate the application ability of CSK-TSK-FS. The data is provided by the University of Bonn, Germany (http://www.meb.uni-bonn.de /epileptologie /science/ physik/ eegdata.html). The dataset consists of five groups, i.e., group A to group E, with each one containing 100 single channel EEG segments of 23.6 duration. The sampling rate is 173.6Hz. Segments in group A and group B are obtained from healthy volunteer subjects and segments in groups C, D and E are acquired from volunteer subjects with epilepsy. Table 5 gives the detailed description about the epileptic EEG data, and Fig.5 illustrates some representative original epileptic EEG signals in five groups.

TABLE 5. Detailed information about the epileptic EEG data.

Subjects	Group	#Size	Description
Healthy	A	100	Signals captured from volunteers with eyes open
	B	100	Signals captured from volunteers with eyes closed
Epileptic	C	100	Signals captured from volunteers during seizure silence intervals.
	D	100	Signals captured from volunteers during seizure silence intervals.
	E	100	Signals captured from volunteers during seizure activity.

The training results of CSK-TSK-FS on the epileptic EEG data is listed in Table 6.

According to the training results listed in Table 6, all fuzzy rules involved in CSK-TSK-FS can be easily written. For example, the first two fuzzy rules can be formulated as

Fuzzy Rule 1: **If** ω_1 is $\Omega_1^1(c_1^1 = 0.6146, \delta_1^1 = 1.89\text{e-}05) \wedge \omega_2$ is $\Omega_2^1(c_2^1 = 0.6571, \delta_2^1 = 1.42\text{e-}05) \wedge \omega_3$ is $\Omega_3^1(c_3^1 = 0.7192, \delta_3^1 = 1.11\text{e-}05) \wedge \omega_4$ is $\Omega_4^1(c_4^1 = 0.5783, \delta_4^1 = 2.45\text{e-}05) \wedge \omega_5$ is $\Omega_5^1(c_5^1 = 0.5031, \delta_5^1 = 2.78\text{e-}05) \wedge \omega_6$ is $\Omega_6^1(c_6^1 = -0.4781, \delta_6^1 = 1.50\text{e-}05)$ **then** $\phi^1(\boldsymbol{\omega}) = 0.4561 - 1.0581\omega_1 - 0.5632\omega_2 - 0.0865\omega_3 + 1.4891\omega_4 - 1.6326\omega_5 + 0.2962\omega_6$.

Fuzzy Rule 2: **If** ω_1 is $\Omega_1^1(c_1^1 = -0.0714, \delta_1^1 = 2.44\text{e-}05) \wedge \omega_2$ is $\Omega_2^1(c_2^1 = 0.7855, \delta_2^1 = 2.01\text{e-}05) \wedge \omega_3$ is $\Omega_3^1(c_3^1 = 0.6302, \delta_3^1 = 1.77\text{e-}05) \wedge \omega_4$ is $\Omega_4^1(c_4^1 = 0.3356, \delta_4^1 = 1.87\text{e-}05) \wedge \omega_5$ is $\Omega_5^1(c_5^1 = 0.4046, \delta_5^1 = 1.50\text{e-}05) \wedge \omega_6$

TABLE 6. Training results of CSK-TSK-FS on the epileptic EEG data.

Fuzzy rules If ω_1 is $\Omega_1^k \wedge \omega_2$ is $\Omega_2^k \wedge \ldots \wedge \omega_d$ is Ω_d^k, then $\phi^k(\boldsymbol{\omega}) = \rho_0^k + \rho_1\omega_1 + \ldots + \rho_d\omega_d$			
#Rules	Antecedents Center: $\mathbf{c}_k = [c_{k1}, c_{k2}, \ldots, c_{kd}]$ Kernel width: $\boldsymbol{\delta}_k = [\delta_{k1}, \delta_{k2}, \ldots, \delta_{kd}]$	Consequents	
1	$\mathbf{c}_1 = [0.6146, 0.6571, 0.7192, 0.5783, 0.5031, -0.4781]$ $\boldsymbol{\delta}_1 = [1.89\text{e-}05, 1.42\text{e-}5, 1.11\text{e-}5, 2.45\text{e-}5, 2.78\text{e-}5, 1.50\text{e-}5]$	$\rho_0^1 = 0.4561$	$\rho_1 = -1.0581$ $\rho_2 = -0.5632$ $\rho_3 = -0.0865$ $\rho_4 = 1.4891$ $\rho_5 = -1.6326$ $\rho_6 = 0.2926$
2	$\mathbf{c}_2 = [-0.0714, 0.7855, 0.6302, 0.3356, 0.4046, -0.3013]$ $\boldsymbol{\delta}_2 = [2.44\text{e-}05, 2.01\text{e-}5, 1.77\text{e-}5, 1.87\text{e-}5, 1.50\text{e-}5, 7.02\text{e-}6]$	$\rho_0^2 = 6.9810$	
3	$\mathbf{c}_3 = [0.5011, -0.1321, 0.5703, 0.5557, 0.9682, -0.8921]$ $\boldsymbol{\delta}_3 = [3.76\text{e-}05, 3.64\text{e-}5, 3.80\text{e-}5, 3.46\text{e-}5, 3.78\text{e-}5, 3.99\text{e-}5]$	$\rho_0^3 = -1.0540$	
4	$\mathbf{c}_4 = [0.4561, 0.1000, 0.2447, 0.0986, 0.0975, -0.8875]$ $\boldsymbol{\delta}_4 = [2.45\text{e-}05, 1.98\text{e-}5, 1.47\text{e-}5, 1.56\text{e-}5, 1.89\text{e-}5, 0.97\text{e-}5]$	$\rho_0^4 = -1.5794$	
5	$\mathbf{c}_5 = [-0.6873, -0.6032, -0.6121, -0.7777, -0.8451, -0.4074]$ $\boldsymbol{\delta}_5 = [1.52\text{e-}05, 1.30\text{e-}5, 1.39\text{e-}5, 1.42\text{e-}5, 1.10\text{e-}5, 1.82\text{e-}5]$	$\rho_0^5 = 0.1871$	
6	$\mathbf{c}_6 = [-0.2400, -0.6011, -0.4028, -0.4513, -0.4931, -0.9742]$ $\boldsymbol{\delta}_6 = [2.68\text{e-}05, 1.47\text{e-}5, 2.59\text{e-}5, 2.31\text{e-}5, 2.48\text{e-}5, 8.01\text{e-}6]$	$\rho_0^6 = 0.3016$	

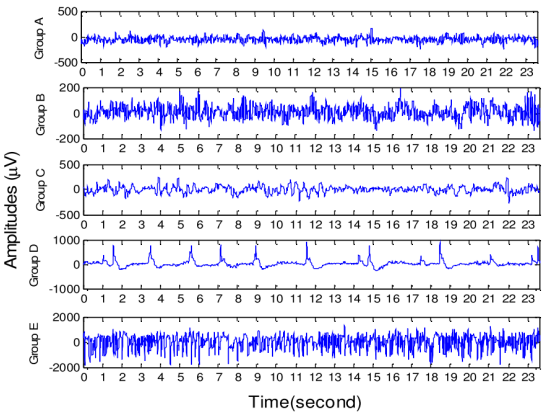


FIGURE 5. Original signals in five groups.

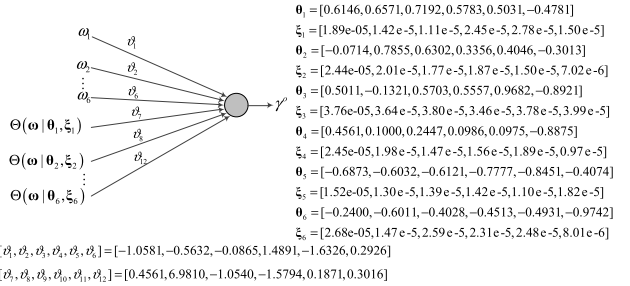


FIGURE 6. FLNN obtained from the trained CSK-TSK-FS.

is $\Omega_6^1(c_6^1 = -0.3013, \delta_6^1 = 7.02\text{e-}06)$ **then** $\phi^1(\boldsymbol{\omega}) = 6.9810 - 1.0581\omega_1 - 0.5632\omega_2 - 0.0865\omega_3 + 1.4891\omega_4 - 1.6326\omega_5 + 0.2962\omega_6$.

In above fuzzy rules, ω_1, ω_2 et al. are the features extracted from the original EEG signals, each feature can be interpreted as the frequency band and its value denotes the energy of the EEG signal in the corresponding band. Obviously, ρ_0^k represents the special knowledge involved in CSK-TSK-FS and $[\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6] = [-1.0581, -0.5632, -0.0865, 1.4891, -1.6326, 0.2926]$ is the common knowledge which also provides a linear approximator $-1.0581\omega_1 - 0.5632\omega_2 - 0.0865\omega_3 + 1.4891\omega_4 - 1.6326\omega_5 + 2.2926\omega_6$ for CSK-TSK-FS.

Moreover, since the common knowledge is not dependent on each fuzzy rule, the consequent of each fuzzy rule can be implicitly shortened as, e.g., $\phi^1(\boldsymbol{\omega}) = 6.9810$ in the first fuzzy rule. Therefore, the interpretability of CSK-TSK-FS is accordingly enhanced comparing with only special knowledge-driven TSK fuzzy systems.

With the trained CSK-TSK-FS, it is very easy for us to present a corresponding FLNN based on the equivalence between them, see in Fig.6. Conversely, with a trained FLNN, we can also immediately write all fuzzy rules of CSK-TSK-FS. In addition, since the equivalence, FLNN is no longer a black box, it can be interpreted from the perspective of fuzzy rules.

Similarly, from the trained CSK-TSK-FS, we also can deduce the corresponding GMM, see the mean vector and covariance matrix of each component in Table 7.

In Table 7, θ_{cy} in $\boldsymbol{\theta}_c$ can be calculated by $\rho_0^k - \boldsymbol{\theta}_{c\boldsymbol{\omega}}^T \boldsymbol{\Psi}$ where $c = k$ and $\kappa_c = 1/K$. Also, we find that τ_{cij} in $\boldsymbol{\Xi}_c$ can be obtained from the trained CSK-TSK-FS. With the assumption that $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma} / \boldsymbol{\tau}^{c\gamma\gamma} = \boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_d]^T$, the values of $\boldsymbol{\tau}_{c\boldsymbol{\omega}\gamma}$ and $\boldsymbol{\tau}_{c\gamma\gamma}$ have many choices. That is to say, multiple

TABLE 7. Each component of GMM based on CSK-TSK-FS.

c	$\boldsymbol{\theta}_c = \begin{bmatrix} \theta_{c1} \\ \vdots \\ \theta_{cd} \\ \theta_{cy} \end{bmatrix}$	$\boldsymbol{\Xi}_c = \begin{bmatrix} \boldsymbol{\tau}_{c\omega\omega} & \boldsymbol{\tau}_{c\omega\gamma} \\ \boldsymbol{\tau}_{c\gamma\omega} & \boldsymbol{\tau}_{c\gamma\gamma} \end{bmatrix} = \begin{bmatrix} \{\tau_{cij}\}_{d\times d} & \{\tau_{cj(d+1)}\}_{d\times 1} \\ \{\tau_{c(d+1)j}\}_{1\times d} & \tau_{c(d+1)(d+1)} \end{bmatrix}$
1	0.6146 0.6571 0.7192 0.5783 0.5031 -0.4781 1.6388	1.89e-5 τ_{112} τ_{113} τ_{114} τ_{115} τ_{116} $\tau_{11\gamma}$ 1.42e-5 τ_{123} τ_{124} τ_{125} τ_{126} $\tau_{12\gamma}$ 1.11e-5 τ_{134} τ_{135} τ_{136} $\tau_{13\gamma}$ 2.45e-5 τ_{145} τ_{146} $\tau_{14\gamma}$ 2.78e-5 τ_{156} $\tau_{15\gamma}$ 1.50e-5 $\tau_{16\gamma}$ τ_{177}
2	-0.0714 0.7855 0.6302 0.3356 0.4046 -0.3013 7.6513	2.44e-5 τ_{212} τ_{213} τ_{214} τ_{215} τ_{216} $\tau_{21\gamma}$ 2.01e-5 τ_{223} τ_{224} τ_{225} τ_{226} $\tau_{22\gamma}$ 1.77e-5 τ_{234} τ_{235} τ_{236} $\tau_{23\gamma}$ 1.87e-5 τ_{245} τ_{246} $\tau_{24\gamma}$ 1.50e-5 τ_{256} $\tau_{25\gamma}$ 7.02e-5 $\tau_{26\gamma}$ τ_{277}
3	0.5011 -0.1321 0.5703 0.5557 0.9682 -0.8921 -0.0978	3.76e-5 τ_{312} τ_{313} τ_{314} τ_{315} τ_{316} $\tau_{31\gamma}$ 3.64e-5 τ_{323} τ_{324} τ_{325} τ_{326} $\tau_{32\gamma}$ 3.80e-5 τ_{334} τ_{335} τ_{336} $\tau_{33\gamma}$ 3.46e-5 τ_{345} τ_{346} $\tau_{34\gamma}$ 3.78e-5 τ_{356} $\tau_{35\gamma}$ 3.99e-5 $\tau_{36\gamma}$ τ_{377}
4	0.4561 0.1000 0.2447 0.0986 0.0975 -0.8875 -0.7473	2.45e-5 τ_{412} τ_{413} τ_{414} τ_{415} τ_{416} $\tau_{41\gamma}$ 1.98e-5 τ_{423} τ_{424} τ_{425} τ_{426} $\tau_{42\gamma}$ 1.47e-5 τ_{434} τ_{435} τ_{436} $\tau_{43\gamma}$ 1.56e-5 τ_{445} τ_{446} $\tau_{44\gamma}$ 1.89e-5 τ_{456} $\tau_{45\gamma}$ 0.97e-5 $\tau_{46\gamma}$ τ_{477}
5	-0.6873 -0.6032 -0.6121 -0.7777 -0.8451 -0.4074 -1.0352	1.52e-5 τ_{512} τ_{513} τ_{514} τ_{515} τ_{516} $\tau_{51\gamma}$ 1.30e-5 τ_{523} τ_{524} τ_{525} τ_{526} $\tau_{52\gamma}$ 1.39e-5 τ_{534} τ_{535} τ_{536} $\tau_{53\gamma}$ 1.42e-5 τ_{545} τ_{546} $\tau_{54\gamma}$ 1.10e-5 τ_{556} $\tau_{55\gamma}$ 1.82e-5 $\tau_{56\gamma}$ τ_{577}
6	-0.2400 -0.6011 -0.4028 -0.4513 -0.4931 -0.9742 -0.1737	2.68e-5 τ_{612} τ_{613} τ_{614} τ_{615} τ_{616} $\tau_{61\gamma}$ 1.47e-5 τ_{623} τ_{624} τ_{625} τ_{626} $\tau_{62\gamma}$ 2.59e-5 τ_{634} τ_{635} τ_{636} $\tau_{63\gamma}$ 2.31e-5 τ_{645} τ_{646} $\tau_{64\gamma}$ 2.48e-5 τ_{656} $\tau_{65\gamma}$ 8.01e-5 $\tau_{66\gamma}$ τ_{677}

choices for the equivalent GMM from CSK-TSK-FS can be provided, which is very interesting and will be beneficial for various practical requirements.

Table 8 gives the training accuracy of all approaches in terms of "Max", "Mean", "Std" and "Rules". CSK-TSK-FS wins the best performance in terms of "Mean" and "Rules".

TABLE 8. Each component of GMM based on CSK-TSK-FS.

Approaches	Max Mean	Rules Std
FH-GBML-C	0.7901 0.7321	42 0.0431
GFS-AdaBoost-C	0.8211 0.7981	--- 0.0235
zero-order-TSK-FC	0.8290 0.8193	8 0.0090
L2-TSK-FC	0.8129 0.7982	42 0.0122
FS-FCSVM	0.7990 0.7772	11 0.0308
SVM (Linear)	0.8211 0.8012	--- 0.0193
SVM (Gaussian)	0.8489 0.8043	--- 0.0511
CSK-TSK-FS	0.8432 **0.8223**	**6** 0.0221

V. CONCLUSION

In this paper, a novel fuzzy system CSK-TSK-FS driven by special and common knowledge is proposed in which the common knowledge is defined as the common parts among all fuzzy rules while the special knowledge corresponds to difference parts. More specifically, parameters assigned to the one-order part in then-parts always keep invariant. When the classical centroid defuzzification method is adopted, the involved common knowledge can be separated from fuzzy rules such that the interpretability is enhanced and the model complexity is reduced. In addition, for the modeling of CSK-TSK-FS, except for traditional gradient descent-based and QP-based approaches, we demonstrate that CSK-TSK-FS is mathematically equivalent to a special Gaussian mixture model and a functional linked natural network such that it can also be determined from a trained GMM or a trained FLNN. In other words, traditional training algorithms like EM and BP of GMM and FLNN can also be applied to CSK-TSK-FS. Furthermore, since CSK-TSK-FS is a special natural network, we develop a fast LLM-based algorithm for its modeling in this study. That is to say, we also find a new fast training algorithm for GMM and FLNN. In our experiments, UCI and KEEL datasets are first taken to demonstrate the classification ability of CSK-TSK-FS, an application dataset, i.e., the epileptic EEG data is introduced for abnormal signals recognition.

In the future work, we are interested in developing a deep TSK fuzzy system in which common knowledge is embedded among different layers.

APPENDIX

In (17), $\int_{-\infty}^{+\infty} \gamma \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma$ in the numerator can be computed by,

$$\int_{-\infty}^{+\infty} \gamma \Theta_c(\boldsymbol{\omega}, \gamma) d\gamma = \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}} \sqrt{|\boldsymbol{\Xi}_c|}} \int_{-\infty}^{+\infty} \gamma$$

$$\times \exp\left\{-\frac{1}{2}\begin{bmatrix}\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}\\ \gamma-\theta_{c\gamma}\end{bmatrix}^T \boldsymbol{\Xi}_c^{-1}\begin{bmatrix}\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}\\ \gamma-\theta_{c\gamma}\end{bmatrix}\right\} d\gamma$$
$$= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\int_{-\infty}^{+\infty} \times \gamma \exp\left\{-\frac{1}{2}\begin{pmatrix}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]+\\ (\gamma-\theta_{c\gamma})\boldsymbol{\tau}^{c\gamma\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]+\\ [\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}(\gamma-\theta_{c\gamma})+\\ (\gamma-\theta_{c\gamma})\boldsymbol{\tau}^{c\gamma\gamma}(\gamma-\theta_{c\gamma})\end{pmatrix}\right\} d\gamma. \tag{A.1}$$

Based on the theorem in [28] that the inverse of a symmetric matrix is also symmetric, thus, in (A.1), $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma} = (\boldsymbol{\tau}^{c\gamma\boldsymbol{\omega}})^T$ and accordingly $(\gamma-\theta_{c\gamma})\boldsymbol{\tau}^{c\gamma\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}] = [\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}(\gamma-\theta_{c\gamma})$. Therefore, we can simplify (A.1) as

$$\begin{aligned}&\int_{-\infty}^{+\infty}\gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\int_{-\infty}^{+\infty}\gamma\\ &\quad\times\exp\left\{-\frac{1}{2}\begin{pmatrix}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]+\\ 2(\gamma-\theta_{c\gamma})[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}+\\ (\gamma-\theta_{c\gamma})^2\boldsymbol{\tau}^{c\gamma\gamma}\end{pmatrix}\right\}d\gamma\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\exp\left\{-\frac{1}{2}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right\}\\ &\quad\times\int_{-\infty}^{+\infty}\gamma\exp\left\{-\frac{1}{2}\left(2(\gamma-\theta_{c\gamma})[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}\right.\right.\\ &\quad\left.\left.+(\gamma-\theta_{c\gamma})^2\tau^{c\gamma\gamma}\right)\right\}d\gamma.\end{aligned} \tag{A.2}$$

With the assumption $\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}/\boldsymbol{\tau}^{c\gamma\gamma} = \boldsymbol{\Psi} = [\Psi_1, \Psi_2, \ldots, \Psi_d]^T$, the square for the integral on the right-hand side can be completed and $\int_{-\infty}^{+\infty}\gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma$ is accordingly updated as

$$\begin{aligned}&\int_{-\infty}^{+\infty}\gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\exp\left\{-\frac{1}{2}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right\}\\ &\quad\times\exp\left\{\frac{1}{2}\left(\frac{[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}}{\sqrt{\tau^{c\gamma\gamma}}}\right)^2\right\}\\ &\quad\times\int_{-\infty}^{+\infty}\gamma\exp\left\{-\frac{1}{2}\left((\gamma-\theta_{c\gamma})\tau^{c\gamma\gamma}\right.\right.\\ &\quad\left.\left.+\frac{[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}}{\sqrt{\tau^{c\gamma\gamma}}}\right)^2\right\}d\gamma\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\exp\left\{-\frac{1}{2}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right\}\\ &\quad\times\exp\left\{\frac{1}{2}\left(\frac{[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}}{\sqrt{\tau^{c\gamma\gamma}}}\right)^2\right\}\\ &\quad\times\int_{-\infty}^{+\infty}\gamma\exp\left\{-\frac{1}{2\left(\frac{1}{\tau^{c\gamma\gamma}}\right)}\right.\\ &\quad\left.\times\left(\gamma-\left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right)\right)^2\right\}d\gamma.\end{aligned} \tag{A.3}$$

For simplicity, wes define a Gaussian distribution as the following form

$$\Upsilon^1\left(\gamma;\theta_c,\tau_c^2\right) = \Upsilon^1\left(\gamma;\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi},\frac{1}{\tau^{c\gamma\gamma}}\right) \tag{A.4}$$

and then substitute it into (A.3). Then, we can obtain

$$\begin{aligned}&\int_{-\infty}^{+\infty}\gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma\\ &= \frac{\kappa_c\sqrt{2\pi\left(\frac{1}{\tau^{c\gamma\gamma}}\right)}}{(2\pi)^{\frac{d+1}{2}}\sqrt{|\boldsymbol{\Xi}_c|}}\\ &\quad\times\exp\left\{-\frac{1}{2}\begin{pmatrix}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\\ -\left(\frac{[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}}{\sqrt{\tau^{c\gamma\gamma}}}\right)^2\end{pmatrix}\right\}\\ &\quad\times\int_{-\infty}^{+\infty}\gamma\Upsilon^1\left(\gamma;\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi},\frac{1}{\delta^{cyy}}\right)d\gamma\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d}{2}}\sqrt{\tau^{c\gamma\gamma}\,|\boldsymbol{\Xi}_c|}}\\ &\quad\times\exp\left\{-\frac{1}{2}\begin{pmatrix}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\\ -\left(\frac{[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}}{\sqrt{\tau^{c\gamma\gamma}}}\right)^2\end{pmatrix}\right\}\\ &\quad\times\left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right)\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d}{2}}\sqrt{\tau^{c\gamma\gamma}\,|\boldsymbol{\Xi}_c|}}\\ &\quad\times\exp\left\{-\frac{1}{2}\begin{pmatrix}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\\ -\frac{([\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}])}{\tau^{c\gamma\gamma}}\end{pmatrix}\right\}\\ &\quad\times\left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right)\\ &= \frac{\kappa_c}{(2\pi)^{\frac{d}{2}}\sqrt{\tau^{c\gamma\gamma}\,|\boldsymbol{\Xi}_c|}}\\ &\quad\times\exp\left\{-\frac{1}{2}\left([\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T[\boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}-\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}(\tau^{c\gamma\gamma})^{-1}\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}]\right.\right.\\ &\quad\left.\left.\times[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right)\right\}\times\left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right).\end{aligned} \tag{A.5}$$

Based on the theorem in [29] that

$$\mathbf{A}^{11} = \left(\mathbf{A}_{11}-\mathbf{A}_{12}\mathbf{A}_{11}^{-1}\mathbf{A}_{21}\right)^{-1} \tag{A.6}$$

we can have

$$\mathbf{A}_{11} = \left(\mathbf{A}^{11}-\mathbf{A}^{12}(\mathbf{A}^{22})^{-1}\mathbf{A}^{21}\right)^{-1}$$

or

$$(\mathbf{A}_{11})^{-1} = \mathbf{A}^{11}-\mathbf{A}^{12}(\mathbf{A}^{22})^{-1}\mathbf{A}^{21}. \tag{A.7}$$

Therefore, we can obtain

$$(\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})^{-1} = \boldsymbol{\tau}^{c\boldsymbol{\omega}\boldsymbol{\omega}}-\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}(\tau^{c\gamma\gamma})^{-1}\boldsymbol{\tau}^{c\boldsymbol{\omega}\gamma}, \tag{A.8}$$

and substitute it to (A.5), accordingly we have

$$\begin{aligned}\int_{-\infty}^{+\infty} & \gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma \\ = & \frac{\kappa_c}{(2\pi)^{\frac{d}{2}}\sqrt{\tau^{c\gamma\gamma}\,|\boldsymbol{\Xi}_c|}} \\ & \times \exp\left\{-\frac{1}{2}\left([\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T(\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})^{-1}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right)\right\} \\ & \times \left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right). \end{aligned} \tag{A.9}$$

Since $\tau^{c\gamma\gamma}\,|\boldsymbol{\Xi}_c|$ is the $(d+1)(d+1)th$ cofactor of the covariance matrix $\boldsymbol{\Xi}_c$, and it is also the determinant of $\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}$, we can simplify (A.9) to the following form

$$\begin{aligned}\int_{-\infty}^{+\infty} & \gamma\Theta_c(\boldsymbol{\omega},\gamma)d\gamma \\ = & \frac{\kappa_c}{(2\pi)^{\frac{d}{2}}\sqrt{|\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}}|}} \\ & \times \exp\left\{-\frac{1}{2}\left([\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T(\boldsymbol{\tau}_{c\boldsymbol{\omega}\boldsymbol{\omega}})^{-1}[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]\right)\right\} \\ & \times \left(\theta_{c\gamma}-[\boldsymbol{\omega}-\boldsymbol{\theta}_{c\boldsymbol{\omega}}]^T\boldsymbol{\Psi}\right). \end{aligned} \tag{A.10}$$

REFERENCES

[1] Y. Jiang, F. Chung, H. Ishibuchi, Z. Deng, and S. Wang, "Multitask TSK fuzzy system modeling by mining intertask common hidden structure," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 548–561, Mar. 2015.

[2] Y. Jiang, Z. Deng, F.-L. Chung, G. Wang, P. Qian, K.-S. Choi, and S. Wang, "Recognition of epileptic EEG signals using a novel multiview TSK fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 3–20, Feb. 2017.

[3] Y. Jiang, D. Wu, Z. Deng, P. Qian, J. Wang, G. Wang, F.-L. Chung, K.-S. Choi, and S. Wang, "Seizure classification from EEG signals using transfer learning, semi-supervised learning and TSK fuzzy system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 12, pp. 2270–2284, Dec. 2017.

[4] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-leverage-based fuzzy system and its modeling," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 597–609, Aug. 2013.

[5] M. J. Lichtenberg and Ö. Şimşek, "Simple regression model," in *Proc. NIPS*, vol. 58, 2017, pp. 13–25.

[6] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*. London, U.K.: Chapman & Hall, 1981.

[7] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, May 1992.

[8] S. Wang, F.-L. Chung, J. Wu, and J. Wang, "Least learning machine and its experimental studies on regression capability," *Appl. Soft Comput.*, vol. 21, pp. 677–684, Aug. 2014.

[9] S. Wang, Y. Jiang, F.-L. Chung, and P. Qian, "Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification," *Appl. Soft Comput.*, vol. 37, pp. 125–141, Dec. 2015.

[10] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, K. S. Choi, and S. Wang, "Transfer prototype-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1210–1232, Oct. 2016.

[11] K. Kilic, O. Uncu, and I. B. Türksen, "Comparison of different strategies of utilizing fuzzy clustering in structure identification," *Inf. Sci.*, vol. 177, pp. 5153–5162, Dec. 2007.

[12] K. Xia, H. Yin, P. Qian, Y. Jiang, and S. Wang, "Liver semantic segmentation algorithm based on improved deep adversarial networks in combination of weighted loss function on abdominal CT images," *IEEE Access*, vol. 7, pp. 96349–96358, 2019.

[13] J. Wang, H. Liu, X. Qian, Y. Jiang, Z. Deng, and S. Wang, "Cascaded hidden space feature mapping, fuzzy clustering, and nonlinear switching regression on large datasets," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 640–655, Apr. 2018.

[14] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[15] J. M. Leski, "TSK-fuzzy modeling based on $\sum$-insensitive learning," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 2, pp. 181–193, 2005.

[16] Z. Deng, K.-S. Choi, F.-L. Chung, and S. Wang, "Scalable TSK fuzzy modeling for very large datasets using minimal-enclosing-ball approximation," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 210–226, Apr. 2011.

[17] J. Park and I. W. Sandberg, "Universal approximation using radial-basis function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.

[18] P. J. S. G. Ferreira, "Neural networks and approximation by superposition of Gaussians," in *Proc. ICASSP*, vol. 97, Apr. 1997, pp. 3197–3200.

[19] S. Dehuri and S.-B. Cho, "A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 187–205, Mar. 2010.

[20] Y. M. M. Hassim and R. Ghazali, "Training a functional link neural network using an artificial bee colony for solving a classification problems," *Zoological Res.*, vol. 33, no. 3, pp. 298–303, 2012.

[21] I.-A. Abu-Mahfouz, "A comparative study of three artificial neural networks for the detection and classification of gear faults," *Int. J. Gen. Syst.*, vol. 34, no. 3, pp. 261–277, Jan. 2005.

[22] S. Haring and J. Kok, "Finding functional links for neural networks by evolutionary computation," in *Proc. 5th Belgian-Dutch Conf. Mach. Learn.*, Brussels, Belgium, 1995, pp. 427–437.

[23] Y. Jiang, Z. Deng, F.-L. Chung, and S. T. Wang, "Realizing two-view TSK fuzzy classification system by using collaborative learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 1, pp. 145–160, Jan. 2017.

[24] Z. Deng, L. Cao, Y. Jiang, and S. Wang, "Minimax probability TSK fuzzy system classifier: A more transparent and highly interpretable classification model," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 813–826, Aug. 2015.

[25] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Sep. 2008, pp. 550–565.

[26] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2834–2848, Jun. 2018.

[27] M.-T. Gan, M. Hanmandlu, and A. H. Tan, "From a Gaussian mixture model to additive fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 3, pp. 303–316, Jun. 2005.

[28] E. Kreyzig, *Advanced Engineering Mathematics*, 7th ed. Singapore: Wiley, 1993.

[29] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. New York, NY, USA: Academic, 1979.

[30] J. Zhang, D. Cabric, F. Wang, and Z. Zhong, "Cooperative modulation classification for multipath fading channels via expectation-maximization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6698–6711, Oct. 2017.

[31] E. Sahin, "A new higher-order binary-input neural unit: Learning and generalizing effectively via using minimal number of monomials," M.S. thesis, Dept. Comput. Eng., Middle East Tech. Univ. Ankara, Ankara, Turkey, 1994.

[32] L. Zhang, H.-L. Li, Z.-J. Qiao, and Z.-W. Xu, "A fast BP algorithm with wavenumber spectrum fusion for high-resolution spotlight SAR imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 9, pp. 1460–1464, Jan. 2014.

[33] S.-S. Yang, C.-L. Ho, and C.-M. Lee, "HBP: Improvement in BP algorithm for an adaptive MLP decision feedback equalizer," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 53, no. 3, pp. 240–244, Mar. 2006.

[34] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.

[35] T. Zhou, F.-L. Chung, and S. Wang, "Deep TSK fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1207–1221, Oct. 2017.

[36] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Hybridization of fuzzy GBML approaches for pattern classification problems," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 35, no. 2, pp. 359–365, Apr. 2005.

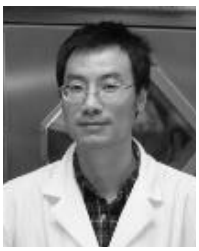
[37] H. Ishibuchi, S. Mihara, and Y. Nojima, "Parallel distributed hybrid fuzzy GBML models with rule set migration and training data rotation," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 2, pp. 355–368, Apr. 2013.

[38] M. J. D. Jesus, F. Hoffmann, L. J. Navascues, and L. Sanchez, "Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 3, pp. 296–308, Jun. 2004.

[39] C.-C. Chang and C.-J. Lin. (2005). *LIBSVM: A Library for Support Vector Machines*. [Online]. Available: http://www.csie.ntu.edu.tw/?cjlin/libsvm
[40] J. Alcal-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, l. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, no. 2, pp. 255–287, Jun. 2011.
[41] X. Gu, F.-L. Chung, and S. Wang, "Bayesian Takagi–Sugeno–Kang fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1655–1671, Dec. 2017.
[42] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[43] J. L. Hodges and E. L. Lehmann, "Rank methods for combination of independent experiments in analysis of variance," *Ann. Math. Statist.*, vol. 33, no. 2, pp. 482–497, 1962.
[44] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandin. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.
[45] K. Bache and M. Lichman. (2015). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[46] A. Lemos, W. Caminhas, and F. Gomide, "Adaptive fault detection and diagnosis using an evolving fuzzy classifier," *Inf. Sci.*, vol. 220, pp. 64–85, Jan. 2013.
[47] J. A. Iglesias, P. Angelov, A. Ledezma, and A. Sanchis, "Human activity recognition based on evolving fuzzy systems," *Int. J. Neural Syst.*, vol. 20, no. 5, pp. 355–364, 2010.
[48] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Hoboken, NJ, USA: Wiley, 2007.
[49] J. Casillas, O. Cordon, F. Herrera and L. Magdalena, *Interpretability Issues in Fuzzy Modeling*. Berlin, Germany: Springer Verlag,2003.
[50] M.J. Gacto and R. Alcala and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inf. Sci.*, vol. 181, no. 20, pp. 4340–4360, Oct. 2011.

YUANPENG ZHANG (M'17) received the Ph.D. degree in information engineering from the School of Computer Application Technology, Jiangnan University, in 2018. He is currently an Associate Professor with the Department of Medical Informatics, Nantong University. He is also a Postdoctoral Fellow with the Department of Health Information Technology, The Hong Kong Polytechnic University. He has published about 20 articles in international/national journals, including the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and the *ACM Transactions on Multimedia Computing Communications and Applications*. His main research interests include pattern recognition and data mining.

JIANCHENG DONG received the M.S. degree from the Information Management Department, Peking University. He is currently a Chair Professor with the First Affiliated Hospital of Zhengzhou University. He has published more than 100 articles. His main research interests include medical big data processing, machine learning, and clinical medicine.



JUNQING ZHU received the bachelor's degree in physics from the Shanghai University of Science and Technology, Shanghai, China, in 1989. In 2006, he joined Case Western Reserve University, where he is currently a Senior Staff of radiology. In the past three decades, he has been focused on *in vivo* molecular imaging. Currently, he put a special emphasis on molecular imaging in neurodegenerative diseases, such as multiple sclerosis (MS), Alzheimer's disease (AD), Parkinson's disease, Epilepsy disease, and DNA damage and repair in cancer. By continuously working with molecular imaging probe group, he pioneered imaging of myelin based on different imaging modalities, such as PET, MRI, and near-infrared fluorescence imaging.

CHUNYING WU received the Ph.D. degree in imaging medicine and nuclear medicine from the School of Medicine, Fudan University, Shanghai, China, in 2003. In 2004, she was a Postdoctoral Fellow with the University of Illinois at Chicago. In 2006, she joined Case Western Reserve University, where she is currently an Instructor of radiology with the Division of Molecular Imaging Center, Case Center for Imaging Research, School of Medicine. She has extensive experience in radiopharmaceutical development for PET and SPECT imaging. Over the past ten years, her research has focused on the development of small-molecular probes for PET imaging in Alzheimer's disease, multiple sclerosis, and DNA damage and repair in cancer. Her research was selected for Molecular Imaging/CMIIT Basic Science and Neuroscience Summary Session highlight talk in the Society of Nuclear Medicine (SNM) Annual Meeting, in 2013 and 2016, respectively. Some of her work has also been selected twice as the Second Place in the poster competition at the annual SNM conferences.