IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 November 2022, accepted 15 January 2023, date of publication 23 January 2023, date of current version 26 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3238798

METHODS

Noise-Robust Fuzzy Classifier Designed With the Aid of Type-2 Fuzzy Clustering and Enhanced Learning

ZIWU JIANG[1], ZHENG WANG[1,2], (Member, IEEE), AND EUN-HU KIM[1], (Member, IEEE)

[1]Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi 276005, China
[2]ICT School, The University of Suwon, Hwaseong, Gyeonggi 445-743, South Korea

Corresponding authors: Zheng Wang (wangzheng@suwon.ac.kr) and Eun-Hu Kim (eunhu84@gmail.com)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2022R1I1A1A01071671, in part by the Shandong Excellent Young Scientists Fund Program (Overseas) in China, and in part by Taishan Young Scholar Experts Project in China.

ABSTRACT This paper introduces the design methodology of a noise-robust fuzzy classifier based on type-2 fuzzy clustering and enhanced learning methods. The design procedure for the noise-robust fuzzy classifier (NrFC) can be divided into two parts. First, interval type-2 fuzzy c-means clustering is applied to the hidden layer to minimize the effect of noise or outliers when training the model. Second, an enhanced learning method is employed to train the connection weights between the hidden and output layers. The proposed NrFC uses a cross-entropy error function as its cost function. The Softmax function represents a categorical distribution located at the output layer nodes. In addition, the connection weights of the output layer are adjusted through nonlinear least squares-based learning, and L_2 norm-regularization is considered to avoid the degradation of the generalization ability caused by overfitting. The learning mechanism is realized by adding the L_2 penalty term to the cross-entropy error function. It is used to cope with overfitting and multicollinearity problems, which generally appear in conventional fuzzy neural networks. The design methodology of the NrFC is discussed and analyzed using several publicly available benchmark datasets. The performance of the proposed networks is quantified through comprehensive experiments and comparative analysis.

INDEX TERMS Interval type-2 fuzzy C-means, L_2-norm regularization, multicollinearity, nonlinear least square, noise-robust fuzzy classifier.

I. INTRODUCTION

Fuzzy neural networks (FNNs) have emerged as one of the most prominent research areas in the synergy between fuzzy logic and neural networks. Significant advances have been made over the past two decades [1], [2]. There are many successful methods for the synthesis of FNNs. The essential advantages of neural networks include their adaptive nature and substantial learning abilities. To establish a strong synergy between these two areas, the FNN combines fuzzy "if-then" rules with neural networks developed using the standard back-propagation (BP) learning algorithm [3], [4], [5]. As a result, FNNs have recently been applied to various research tasks as the core technique for prediction, control, or classification [6], [7], [8], [9]. Recently, type-1/2 fuzzy set theory has been applied to various control system fields such as fuzzy networked singularly perturbed systems, fuzzy passive filters, and T-S fuzzy Markov jump chaotic systems [40], [41], [42]. To use FNNs for real-world problems, Oh and Pedrycz proposed a variety of fuzzy rule-based neural networks combined with clustering, optimization, and dimensionality reduction [10], [11], [12], [13], [14], [15], [16].

Type-2 fuzzy sets generalize the standard type-1 fuzzy sets to handle more uncertainties. They have been widely used in several applications that cannot be solved entirely using

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

only type-1 fuzzy sets [17], [18]. However, type-2 fuzzy sets require more computational complexity than type-1 fuzzy sets because type-2 fuzzy sets contain secondary (type-2) membership grades and each primary (type-1) membership grade. In addition, the type-2 TSK fuzzy logic system (FLS) only uses BP-based learning to update the consequent parameters (coefficients). Nonetheless, the advantages of type-2 fuzzy sets, which deal more effectively with the uncertainty associated with given problems, may countervail these drawbacks [19], [20], [21].

Through the sum of squared error (SSE) function, which is a cost function, the connection weights of NNs are typically trained through BP or LSE-based learning. This learning mechanism was applied simultaneously to both the regression and classification problems. However, in classification problems, the SSE function is not suitable for obtaining the best classification accuracy because the SSE function only considers the error between model outputs and target outputs for all classes [22], [23], [24], [25], [26], [27]. For this reason, the proposed NrFC is combined with the learning method of multinomial logistic regression, which is a representative probabilistic model, to cope with multiclass problems. In addition, the L_2-norm regularization of ridge regression is applied. Ridge regression is a modified technique used to alleviate multicollinearity among predictor (input) variables by adding a small bias factor to these variables. In addition, ridge regression can serve as a shrinkage estimator if there is no multicollinearity, whereas the shrinkage estimator can offer opportunities to improve the generalization abilities [28].

A fuzzy rule can be divided into a premise (antecedent) part and conclusion (consequent) part. In conventional type-2 fuzzy systems, the premise and conclusion parts consist of type-2 fuzzy sets, and then type reduction is performed to estimate the coefficients in the consequent part. The parameters, such as the center and width of the membership functions in the premise part and the coefficients in the conclusion part, are trained by back-propagation-based learning. This learning mechanism provides convenience for learning the model. However, the computational complexity increases significantly, and it is more likely to fall into local minima owing to many parameters and constraints. For these reasons, we propose NrFC to reduce computational complexity and preserve the function of type-2 fuzzy sets to deal with uncertainty due to noise or outliers. In the proposed networks, the premise part is expressed by type-2 fuzzy sets, and the conclusion part consists of type-1 fuzzy sets. The originality of this study can be discussed more specifically as follows: First, the output of the premise part is obtained from IT2FCM clustering. Processing, such as training the centers and computing the lower and upper membership degrees, is performed by considering type-2 fuzzy sets. Finally, the final degree of belonging is obtained after type reduction. Second, because type reduction is completed in the premise part, the conclusion part consists of linear functions that are commonly used in fuzzy systems. In the case of type-2 fuzzy systems, the conclusion part consists of interval linear functions described by the upper and lower values. This requires more parameters than the type-1 fuzzy sets. However, by applying type-2 fuzzy sets only to the premier part, the conclusion part consists of non-interval linear functions. Therefore, we can use a non-iterative learning method, such as the LSE method, instead of BP. Third, unlike conventional FNNs and fuzzy rule-based systems, the cost function of the proposed networks uses a cross-entropy error (CEE) function. This function is more suitable than the mean squares error (MSE) function for classification problems. However, this function is non-closed; therefore, we cannot apply LSE-based learning. We apply the re-weighted least squares error method based on Newton's method to solve the problem. In addition, the L_2-norm regularization method is considered to alleviate the degradation of the generalization ability caused by multicollinearity. As a result, the proposed classifier is more concise than the existing type-2 fuzzy systems. The computational complexity owing to the number of parameters and the learning process is significantly reduced. Despite the reduction of model complexity, the proposed classifier shows robustness to noise compared with the other classifiers previously reported in the literature.

The proposed robust fuzzy classifier is constructed using interval type-2 FCM clustering [29], [30] and nonlinear least-squares estimation (NLSE) with L_2-norm regularization [31]. Compared to FNNs and interval type-2 TSK fuzzy models, the key points of the proposed classifier are summarized as follows:

1) Use of type-2 fuzzy clustering in the hidden layer: IT2FCM was used in the hidden layer to minimize the effect of uncertainty, and the hidden layer of fuzzy rule-based neural networks was composed of clustering algorithms such as K-means or FCM clustering. The clustering algorithm replaces the definitions and parameter settings of the membership function. However, clustering techniques are vulnerable when distinguishing between noisy patterns and outliers. Therefore, applying IT2FCM to the hidden layer maintains the advantage of the clustering algorithm and secures robustness against uncertain information, such as noise or outliers.

2) Training of parameters through an enhanced learning mechanism: FNNs perform model training using SSE function. In classification problems, the SSE function does not guarantee the training of the model to improve classification accuracy because it reduces the mean squared error between the model output and the target output. To overcome this drawback, the proposed NrFC applies the CEE function commonly used in logistic regression instead of the SSE function. Because the CEE function is known as a non-closed-form expression, the LSE-based learning typically used in conventional FNNs or FIS cannot be applied to estimate the connection weights. Therefore, we use Newton's method-based NLSE method instead of LSE.

3) Improvement of generalization abilities through L_2-norm regularization: Overfitting occurs because of

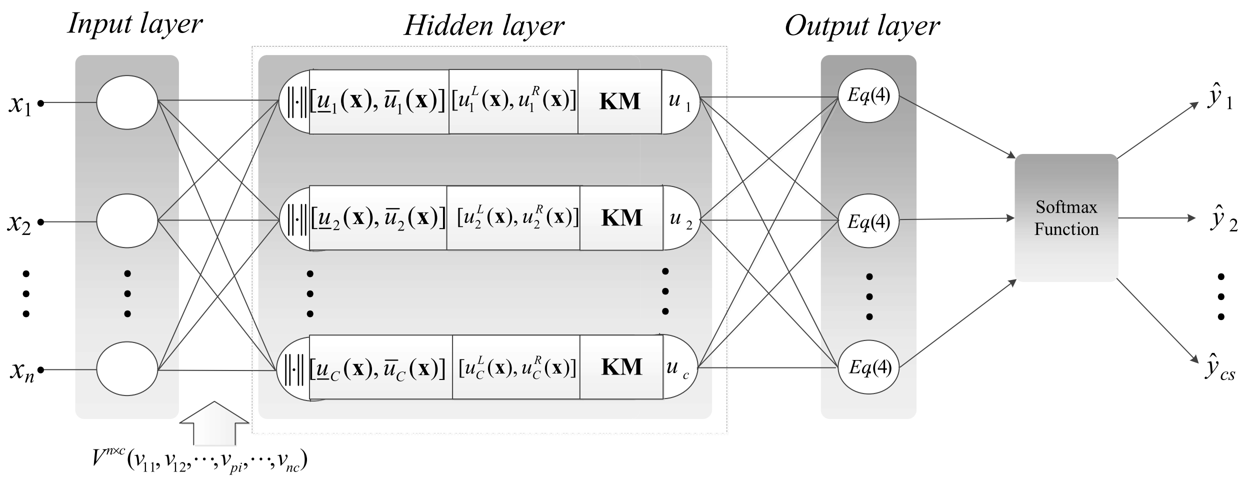


FIGURE 1. Overall architecture and core algorithmic details of the proposed NrFC.

various factors during model training, which reduces the generalization ability of the classifier. Multicollinearity is a representative factor that causes overfitting. When learning the connection weights, multicollinearity or similar forms cause problems with considerable deviations between the coefficients. Consequently, this leads to a decrease in the generalization ability of the model. Although there is no exact multicollinearity among the input variables, a suitable selection of the regularization parameter in the L_2-norm regularization decreases the deviation between the connections (coefficients). This method is known as the shrinkage estimator. As an effect of this, L_2-norm regularization provides a method for model selection through an analysis of the bias-variance tradeoff to avoid overfitting by excessive training and prevent the degradation of generalization ability. L_2-norm regularization is easily applicable by adding the L_2 penalty term to the cost function used in the existing model [28], [31].

Compared with conventional FNNs, the difference of the proposed NrFC is highlighted in three parts. First, IT2FCM clustering is used in a hidden layer to minimize the effect of uncertainty and handle noise or outliers more efficiently. Second, we use the CEE function as a cost function for NrFC to train connection weights that are more suitable for improving classification accuracy. Third, L_2-norm regularization is applied to the cost function to mitigate the overfitting problem caused by multicollinearity during weight learning. As a result, we realize the noise-robust fuzzy classifier through three design strategies.

The remaining of this paper is organized as follows. Section II elaborates on the design methodology and architecture of the noise-robust fuzzy classifier. Section III describes the learning method of the NrFC with the aid of the L_2-norm regularization-based NLSE. In Section IV, a comprehensive set of experiments is presented. Finally, concluding remarks are presented in Section V.

II. ARCHITECTURE OF NOISE-ROBUST FUZZY CLASSIFIER

The architecture of the proposed NrFC is illustrated in figure 1. The basic structure is the same as that of conventional FNNs or RBFNNs. However, the hidden layer of the proposed networks is replaced with interval type-2 FCM, which means that clustering is used to handle the uncertainties in the input space efficiently. A collection of fuzzy rules represents the architecture of the proposed NrFC in the following form:

$$R_i^j : \textbf{If } x \text{ is } u_{\text{i}} \text{ with } v_i \text{ then } g_{ij}(x). \tag{1}$$

Here, i represents the i^{th} fuzzy rule, and j denotes the class index. u_i and $\boldsymbol{v}_i$ represent the degree of belonging and the prototype (center) of the cluster, respectively. Although IT2FCM is employed in the hidden layer, the output of the hidden layer is expressed by the values of type-1 fuzzy sets owing to the type reduction through the KM algorithm. The connection weight $g_{ij}(\boldsymbol{x})$ of the 'then' clause is expressed as a linear function including a constant term in the following form:

$$g_{ij}(\boldsymbol{x}) = a_{i0}^j + \sum_{k=1}^{n} a_{ik}^j x_k, \tag{2}$$

where k means k^{th} input variable.

The coefficients of the linear function were estimated using the NLSE method. Because the CEE function is not provided in closed form, the conventional LSE-based learning method is not applicable here. Hence, we use an iterative learning method called Newton's method-based NLSE as an alternative to LSE [31]. The design of the output layer consists of two steps. The first column of nodes in the output layer represents the output (score) of merging the local models corresponding to each class, and the output value is the real number.

$$z_{jp} = \sum_{i=1}^{c} u_{ip}(x) g_{ij}(x) \tag{3}$$

where p is the data index, and the decision corresponding to an output with the maximum value among the outputs is the final output of the proposed NrFC.

The second step calculates the probability according to each class through a softmax function.

$$p(t_{jp}|x_p) = y_{jp} = \frac{e^{z_{jp}}}{\sum_{l=1}^{cs} e^{z_{jp}}} \tag{4}$$

Here, $t_{jp} = \{t|t \in \{0, 1\}^{cs}, \|t\|_1 = 1\}$represents the corresponding vector of the desired output, and y_{jp} denotes the conditional probability of the node corresponding to each class. The class with the highest probability is chosen as the final output of the classifier.

A. INTERVAL TYPE-2 FUZZY C-MEANS CLUSTERING

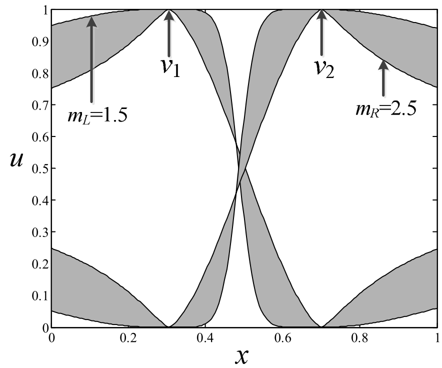
The proposed NrFC model involves two learning mechanisms. The first is IT2FCM clustering, which is used to form a hidden layer. We used IT2FCM clustering because it is more convenient for determining the parameters of the activation (membership) function by IT2FCM than for selecting parameters by the user. In particular, selecting the activation center corresponding to each node is very important from the perspective of the performance of the classifier or model. It takes a lot of time to select a proper parameter by trial and error without any navigator because there are no specific criteria for choosing the center point of the activation function. From this perspective, fuzzy clustering is an effective technique for saving computing loads and obtaining reasonable values (centers) based on a minimized objective function.

FCM algorithm is a soft clustering technique in which a dataset is grouped into c clusters [32]. The largest difference compared with the other clustering algorithms indicates how close the data are to the cluster as a degree of belonging. FCM clustering is applied to the hidden layer of the FNNs to split the input space into c fuzzy sets (groups). By applying type-2 fuzzy sets to general FCM clustering, we can obtain the lower- and upper-interval membership degrees using two different fuzzifiers. The objective function of IT2FCM is the same as that of standard FCM clustering, and is expressed as follows: [29], [30].

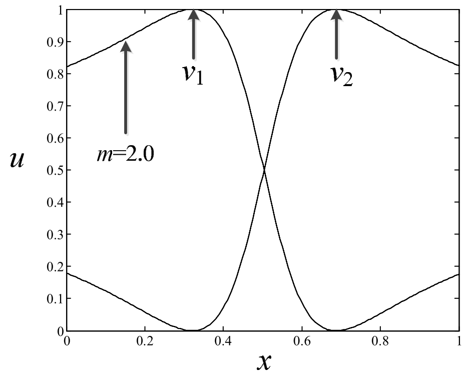
$$J(U, v) = \sum_{i=1}^{c} \sum_{p=1}^{N} u_{ip}^{m} \left\| x_p - v_i \right\|^2 \tag{5}$$

Here, |||| denotes the Euclidean distance, n is the number of patterns (data), and m is a fuzzifier that changes to m_1 and m_2 ($m_1 < m_2$) to form the interval membership degrees. Different fuzzifiers determine the width of the space between the two membership functions. This is commonly referred to as the footprint of uncertainty (FOU), as shown in the shaded area in figure 2.

The minimization of $J(\boldsymbol{U}, \boldsymbol{v})$ is realized iteratively by adjusting both the prototypes and the entries of the partition matrix. The well-known formulas used in an iterative manner are as follows:



(a) IT2 FCM



(b) FCM

FIGURE 2. Comparison of membership function by fuzzifiers.

$$\underline{u}_{ip}(x) = \min\left(\frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ip}}{d_{jp}}\right)^{2/(m_L-1)}}, \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ip}}{d_{jp}}\right)^{2/(m_R-1)}}\right) \tag{6}$$

$$\bar{u}_{ip}(x) = \max\left(\frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ip}}{d_{jp}}\right)^{2/(m_L-1)}}, \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ip}}{d_{jp}}\right)^{2/(m_R-1)}}\right) \tag{7}$$

Unlike conventional FCM, the prototypes of IT2FCM cannot be computed directly because partition values are not Type-1 sets but interval type-2 fuzzy sets. To do this, a centroid type-reducer is employed to obtain accurate prototypes, and then a centroid defuzzifier is used to obtain crisp centers from the type-reduced type-1 fuzzy sets. After type reduction using the (KM) algorithm, the final prototypes of the clusters are described as follows:

$$v_i = \frac{v_i^L + v_i^R}{2}. \tag{8}$$

The membership values $u_{ip}^R(\boldsymbol{x})$ and $\mathrm{u}_{ip}^L(\boldsymbol{x})$ differ according to each input variable for a pattern (data). Thus, the hard partitioning method proposed by Hwang and Rhee computes precise values for the left and right degrees of membership for pattern x_p.

$$u_{ip}(\boldsymbol{x}) = \frac{u_{ip}^L(x) + u_{ip}^R(\boldsymbol{x})}{2} \tag{9}$$

The left and right membership values are computed as

$$u_{ip}^{L}(x)=\sum_{k=1}^{n}\widehat{u_{ip}^{L}}(x)\Big/n$$

where,

$$\widehat{u_{ip}^{L}}(\boldsymbol{x})=\begin{cases}\bar{u}_{ip}(\boldsymbol{x}), & if\ x_{pk}\ uses\ \bar{u}_{ip}(x) for\ v_i^L\\ \underline{u}_{ip}(\boldsymbol{x}), & Otherwise.\end{cases} \tag{10}$$

and

$$u_{ip}^{R}(x)=\sum_{k=1}^{n}\widehat{u_{ip}^{R}}(x)\Big/n$$

where,

$$\widehat{u_{ip}^{R}}(x)=\begin{cases}\bar{u}_{ip}(\boldsymbol{x}), & if\ x_{pk}\ uses\ \bar{u}_{ip}(x) for\ v_i^R\\ \underline{u}_{ip}(\boldsymbol{x}), & Otherwise\end{cases} \tag{11}$$

In the proposed classifier, the output of the hidden layer is $u_{ip}(\boldsymbol{x})$ in (9). The output is not an interval because the type reduction is completed in IT2FCM.

B. KARNICK AND MENDEL FOR TYPE DEDUCTION

To obtain type-1 prototypes from the IT2FCM clustering, the two endpoints must be calculated from the interval values of the partition matrix. This is carried out through Karnick and Mendel (KM) iterative procedure [33]. The process of type reduction can be summarized as follows:

[Step 1] Arrange input patterns in ascending order.

$$x_{1k}\le\cdots\le x_{pk}\le\cdots\le x_{Nk} \tag{12}$$

[Step 2] Initialize $u_{ip}^{R}(t)$ and then compute $\boldsymbol{v}_i^{R}(t)$.

$$u_{ip}^{R}(t)=\frac{\underline{u}_{ip}+\bar{u}_{ip}}{2} \tag{13}$$

$$v_{ik}^{R}(t)=\frac{\sum_{p=1}^{n}u_{ip}^{(m_1\ or\ m_2)}x_{pk}}{\sum_{p=1}^{n}u_{ip}^{(m_1\ or\ m_2)}} \tag{14}$$

[Step 3] Find switching point $s\in[1,N-1]$ such that

$$x_{sk}\le v_{ik}^{R}<x_{(s+1)k} \tag{15}$$

[Step 4] Update $u_{ip}^{R}(t+1)$ and then compute $v_{ik}^{R}(t+1)$.

$$u_{ip}^{R}(t{+}1)=\begin{cases}\underline{u}_{ip} & k\le s\\ \bar{u}_{ip} & k>s\end{cases},\ u_{ip}^{R}=(\underline{u}_{i1},\ldots,\underline{u}_{is},\bar{u}_{i(s+1)},\ldots,\bar{u}_{iN}) \tag{16}$$

$$v_{ik}^{R}(t+1)=\frac{\sum_{p=1}^{n}\mathrm{u}_{\mathrm{ip}}^{(\mathrm{m}_1\ or\ m_2)}x_{\mathrm{pk}}}{\sum_{p=1}^{n}\mathrm{u}_{\mathrm{ip}}^{(\mathrm{m}_1\ or\ m_2)}} \tag{17}$$

[Step 5] If $v_{ik}^{R}(t+1)$ is equal to $v_{ik}^{R}(t)$, the procedure is terminated. Otherwise, proceed to step 3.

In the case of calculating v_{ik}^{L}, the procedure is the same as above, but (19) should be used instead of (17) in step 4.

$$u_{ip}^{L}(t{+}1)=\begin{cases}\bar{u}_{ip} & k\le s\\ \underline{u}_{ip} & k>s\end{cases},\ u_{ip}^{L}=(\bar{u}_{i1},\ldots,\bar{u}_{is},\underline{u}_{i(s+1)},\ldots,\underline{u}_{iN}) \tag{18}$$

III. ENHANCED LEARNING TECHNIQUES THROUGH NEWTON'S METHOD-BASED NONLINEAR LEAST SQUARE ERROR METHOD

This study employs NLSE learning based on Newton's method to adjust the connection weights. NLSE is a learning method that estimates an approximate value through an iterative learning process using the Gauss-Newton method. However, this method can only be in closed-form expressions (solutions) such as the SSE. Because the proposed network uses a generalized cross-entropy error (GCEE) function as a cost function, Newton's method-based NLSE is considered instead of the previous method. Newton's method is typically a learning mechanism, finding an approximate value that satisfies the relationship $f(x)\ =0$ by running an iterative learning procedure by finding extrema using the first and second derivatives of function $f(x)$ [31]. The cost function for parameter training of the proposed network is defined as the GCEE function described in the following form:

$$GCEE=-\frac{1}{N}\sum_{p=1}^{N}\sum_{j=1}^{cs}t_{\mathrm{jp}}\ln y_{jp} \tag{19}$$

Here, t_{jp} represents the desired output and y_{jp} denotes the output (probability) of the proposed network.

In addition, L_2-norm regularization is employed to avoid degradation of the generalization ability caused by possible overfitting. To achieve this, the cost function of the proposed networks adds a penalty term, as in the ridge regression model. This method helps to reduce variations among coefficients and prevents degradation of the generalization ability [28], [31]. Multicollinearity yields a high variance model that becomes increasingly unrealistic as correlation increases. The high variance model is very sensitive to a small change in each input variable because it consists of weights with large deviations from each other. Thus, the result of producing high-variance models is directly related to the potential performance instability. L_2-norm regularization addresses the numerical instability of atrix inversion and subsequently produces low-variance models. This method adds a positive constant to the diagonals of $\boldsymbol{X}^T\boldsymbol{X}$ to make the matrix nonsingular. The analytic solution to the problem becomes:

$$\boldsymbol{A}=\left(\boldsymbol{X}^T\boldsymbol{X}+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}. \tag{20}$$

Here λ is a regularization parameter that assumes a positive value. In some cases, a ridge trace is used to search for the optimal regularization parameter. In the proposed classifier, the GCEE function applied with L_2-norm regularization is

TABLE 1. List of the parameters of the proposed NrFC.

Layers	Parameters	Values
Input	Data preprocessing	Standardization
Hidden	Number of clusters (c)	2 to 5
	Fuzzifier (m)	FCM [1.1,2.0,3.0] IT2FCM [1.1~2.0,1.1~3.0, 2.0~3.0]
	Maximum iteration of FCM	100
Output	Maximum iteration of LSE	N/A
	Maximum iteration of NLSE	2000
	Regularization parameter (λ)	0.0001, 0.001, 0.01, 0.1

depicted in the following form:

$$GCEE = -\frac{1}{N}\sum_{p=1}^{N}\sum_{j=1}^{cs} t_{\text{jp}} \ln y_{jp} + \sum_{j=1}^{cs}\sum_{l=1}^{c\times n} a_{jl}^2, \tag{21}$$

and through Newton's method, the above cost function specializes as follows:

$$A_j\left(t+1\right) = A_j(t) - \frac{\nabla GCCE_{L_2}(A_j(t))}{\nabla^2 GCCE_{L_2}(A_j(t))}. \tag{22}$$

The matrix equation for the estimating weights (coefficients) $\boldsymbol{A}$ is represented through the first- and second-order partial derivatives of the cost function as follows:

$$A_j(t+1) = A_j(t) - (X^T Q_j X + \lambda I)^{-1}(X^T(Y_j - T_j) + \lambda A_j(t)) \tag{23}$$

Consequently, this equation can be rearranged into a formula grouped by $\left(\boldsymbol{X}^T\boldsymbol{Q}_j\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}$, and which is similar to the expression encountered in ridge regression:

$$A_j(t+1) = (X^T Q_j X + \lambda I)^{-1} X^T Q_j N(t) \tag{24}$$

where $N(t) = XA_j(t) - Q_j^{-1}(Y_j - T_j)$, and t is the iteration number. At this point, $\boldsymbol{T}_j$changes according to the class index as follows:

$$t_p = \begin{cases} 1 & if\ y_p = j \\ 0 & Otherwise \end{cases} \tag{25}$$

where $y_p$denotes the actual classes represented as integers, and the target classes are transformed to 0 or 1 for NLSE-based learning. To estimate the connections of the proposed networks, the initial values of the weights were selected randomly, as in the BP-based learning scheme. The learning process is repeated until a predefined number of iterations has been exceeded, or the difference between $(t+1)$ and (t) is below a predefined threshold.

TABLE 2. Information of benchmark machine learning datasets.

No.	Data types	No. of attributes	No. of classes	No. of instances
1	Iris	4	3	150
2	BTSC	4	2	748
3	Balance	4	3	625
4	Haberman	3	2	306

TABLE 3. Results of comparative analysis of T1LSE, T2LSE, and NrFC for Iris dataset.

Classifiers	c	m	λ	Training data	Testing data
T1LSE	2	1.1	0	98.00 ±0.75	98.00 ±1.53
T1LSE with L_2	3	1.1	0.1	98.17 ±0.70	98.00 ±1.53
T2LSE	2	1.1~3.0	0	98.33 ±0.95	**98.67 ±2.98**
T2LSE with L_2	2	1.1~3.0	0.01	98.83 ±0.85	**98.67 ±2.98**
NrFC	2	1.1~3.0	0	99.33 ±0.37	96.67 ±2.36
NrFC with L_2	2	1.1~2.0	0.01	97.83 ±0.95	97.33 ±3.65

IV. EXPERIMENTAL STUDIES

We considered two types of fuzzy classifiers as comparative models. The first classifier, T1LSE, consists of standard FCM clustering in the hidden layer and LSE-based learning to train connection weights. The second classifier, T2LSE, comprises IT2FCM clustering and LSE-based learning. The first classifier was used to show the differences between type-1 and type-2 fuzzy sets. The second one was to compare the differences between the LSE and NLSE learning methods.

TABLE 1 lists the parameter settings of the classifiers used in the experiments. The experimental conditions were the same for all the classifiers. The classification accuracy was expressed as the mean and its standard deviation through 5-fold cross-validation, which is a popular performance evaluation method.

We used several machine learning datasets to compare the classification accuracy of the Type-1 fuzzy set-based FNNs and the proposed NrFC. These datasets were obtained from the University of California Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html).

TABLE 2 presents a summary of the datasets. White Gaussian noise was added to the dataset to evaluate the noise robustness of each classifier. The noise ratio is set at 5 dB, 10 dB, and 15 dB, where 5 dB indicates that 95% of the data are additionally affected by white Gaussian noise [25].

A. IRIS DATA SET

This is the most well-known database in pattern recognition literature. Fisher's paper is classic in the field and is frequently referenced. The dataset contained three classes of 50 instances, where each class referred to a type of iris plant. One class is linearly separable from the

TABLE 4. Results of comparative analysis of T1LSE, T2LSE, and NrFC according to the effect of white Gaussian noise for Iris dataset.

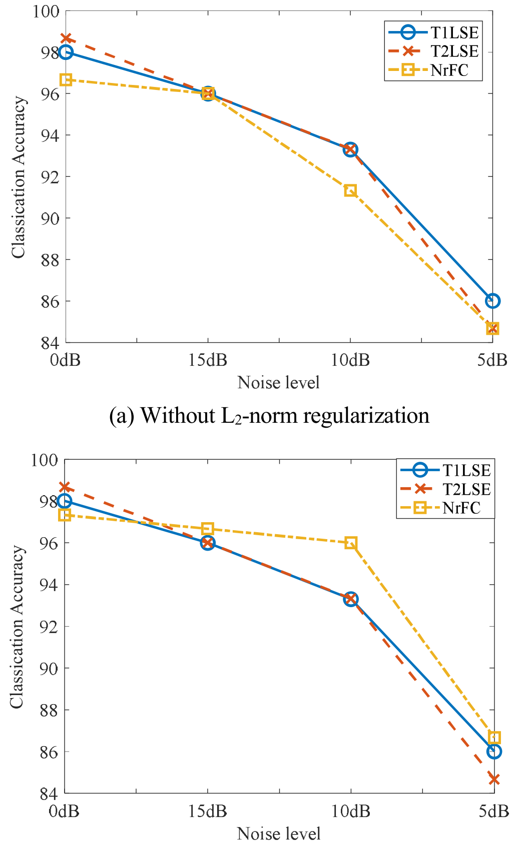
Classifiers	c	*m*	λ	Training data	Testing data
15dB					
T1LSE	2	1.1	0	97.17 ±0.95	96.00 ±4.35
T1LSE with L_2	2	1.1	0.01	97.17 ±0.95	96.00 ±4.35
T2LSE	2	1.1~2.0	0	96.67 ±1.32	96.00 ±4.35
T2LSE with L_2	2	1.1~2.0	0.1	96.83 ±1.24	96.00 ±4.35
NrFC	2	1.1~3.0	0	98.33 ±0.59	96.00 ±4.35
NrFC with L_2	5	2.0~3.0	0.001	98.67 ±0.95	**96.67 ±2.36**
10dB					
T1LSE	2	1.1	0	92.17 ±1.83	93.33 ±4.71
T1LSE with L_2	2	1.1	0.1	92.17 ±1.83	93.33 ±4.71
T2LSE	2	1.1~2.0	0	92.00 ±1.92	93.33 ±4.71
T2LSE with L_2	2	1.1~2.0	0.1	92.00 ±1.92	93.33 ±4.71
NrFC	2	1.1~2.0	0	93.33 ±1.73	91.33 ±5.06
NrFC with L_2	5	1.1~2.0	0.001	95.67 ±2.07	**96.00 ±4.35**
5dB					
T1LSE	5	3.0	0	87.83 ±0.95	86.00 ±4.35
T1LSE with L_2	5	3.0	0.001	87.83 ±0.95	86.00 ±4.35
T2LSE	5	2.0~3.0	0	88.50 ±1.81	84.67 ±5.06
T2LSE with L_2	5	2.0~3.0	0.001	88.50 ±1.81	84.67 ±5.06
NrFC	3	1.1~3.0	0	89.67 ±1.12	84.67 ±5.06
NrFC with L_2	3	2.0~3.0	0.001	90.17 ±1.49	**86.67 ±4.08**

other two classes, and the latter is not linearly separable. (https://archive.ics.uci.edu/ml/datasets/iris).

TABLE 3 shows the classification accuracy (CA) of T1LSE, T2LSE, and the proposed NrFC. The CA is reported as the mean and its standard deviation, and the bold faces indicate the best classification accuracy based on the testing dataset of each classifier. Among the three classifiers, T2LSE achieved the highest classification accuracy.

TABLE 4 lists the classification accuracies of the classifiers according to the various noise levels. After adding noise to the dataset, the classification rate of the proposed NrFC is preferable to that of other classifiers, such as T1LSE and T2LSE. Moreover, the performance was improved by applying the L_2-norm regularization.

Figure 3 shows the pattern of classification accuracy according to the change in noise intensity. When using the L_2-norm regularization, the classification accuracy of the testing dataset was better than that without regularization. Moreover, the proposed NrFC exhibited noise robustness compared with



(a) Without L_2-norm regularization

(b) With L_2-norm regularization

FIGURE 3. Comparison analysis of the classification accuracy of classifiers according to the change of noise intensity for the Iris dataset.

TABLE 5. Results of comparative analysis of T1LSE, T2LSE, and NrFC for BTSC dataset.

Classifiers	c	*m*	λ	Training data	Testing data
T1LSE	2	1.1	0	78.17 ±0.46	79.01 ±3.49
T1LSE with L_2	2	1.1	0.1	78.17 ±0.46	79.01 ±3.49
T2LSE	2	1.1~2.0	0	78.50 ±0.56	79.01 ±3.67
T2LSE with L_2	2	1.1~2.0	0.01	78.50 ±0.56	79.01 ±3.67
NrFC	2	1.1~3.0	0	78.34 ±0.55	**79.27 ±4.04**
NrFC with L_2	2	1.1~3.0	0.1	78.34 ±0.55	**79.27 ±4.04**

TABLE 6. Results of comparative analysis of T1LSE, T2LSE, and NrFC according to the effect of white Gaussian noise for BTSC dataset.

Classifiers	c	*m*	λ	Training data	Testing data
15DB					
T1LSE	2	1.1	0	78.41 ±0.37	78.87 ±4.21
T1LSE with L_2	2	1.1	0.1	78.49 ±0.35	78.87 ±4.21
T2LSE	2	1.1~2.0	0	79.81 ±0.89	77.67 ±2.90
T2LSE with L_2	2	1.1~2.0	0.1	79.71 ±1.02	77.67 ±2.20
NrFC	2	1.1~3.0	0	78.41 ±0.35	**79.01 ±3.49**
NrFC with L_2	2	1.1~3.0	0.01	78.44 ±0.36	**79.01 ±3.49**
10DB					
T1LSE	2	1.1	0	78.30 ±0.35	78.34 ±2.02
T1LSE with L_2	5	3.0	0.1	79.71 ±0.68	78.47 ±1.65
T2LSE	2	2.0~3.0	0	78.74 ±0.68	77.81 ±2.02
T2LSE with L_2	2	2.0~3.0	0.1	78.77 ±0.57	78.07 ±2.18
NrFC	2	1.1~3.0	0	78.27 ±1.92	79.21 ±3.54
NrFC with L_2	5	1.1~2.0	0.01	78.27 ±1.92	**79.21 ±3.71**
5DB					
T1LSE	3	3.0	0	78.07 ±0.38	78.34 ±2.40
T1LSE with L_2	3	2.0	0.1	78.14 ±0.50	78.47 ±1.93
T2LSE	3	1.1~3.0	0	78.54 ±0.41	78.61 ±1.67
T2LSE with L_2	3	1.1~3.0	0.001	78.57 ±0.41	78.61 ±1.67
NrFC	3	1.1~3.0	0	78.34 ±0.33	**78.61 ±1.03**
NrFC with L_2	3	1.1~3.0	0.1	78.34 ±0.33	**78.61 ±1.03**

the two classifiers. In particular, there was almost no degradation in classification accuracy from 0 dB to 10 dB.

B. BTSC DATA SET

This data was obtained from the donor database of the Blood Transfusion Service Center in Hsin-Chu City, Taiwan. (https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center) [34]. TABLE 5 shows the performance comparison of the proposed classifier and the other two classifiers. The performance of NrFC with L_2-norm regularization is better than that of the different classifiers.

TABLE 7. Results of comparative analysis of T1LSE, T2LSE, and NrFC for the Balance dataset.

Classifiers	c	*m*	λ	Training data	Testing data
T1LSE	2	2.0	0	90.04 ±0.92	88.35 ±5.03
T1LSE with L_2	3	2.0	0.1	89.80 ±1.21	88.64 ±3.67
T2LSE	5	1.1~3.0	0	90.28 ±0.82	89.46 ±2.68
TLSE with L_2	4	2.0~3.0	0.0001	90.96 ±0.79	90.09 ±1.99
NrFC	4	1.1~2.0	0	98.75 ±0.79	**94.70 ±1.78**
NrFC with L_2	4	1.1~2.0	0.0001	98.35 ±0.55	**94.70 ±1.28**

TABLE 8. Results of comparative analysis of T1LSE, T2LSE, and NrFC according to the effect of white Gaussian noise for Balance dataset.

Classifiers	c	*m*	λ	Training data	Testing data
15DB					
T1LSE	4	2.0	0	87.92 ±1.24	87.68 ±3.45
T1LSE with L_2	4	2.0	0.1	88.24 ±1.04	87.68 ±3.45
T2LSE	5	2.0~3.0	0	88.60 ±0.99	88.03 ±5.35
TLSE with L_2	4	2.0~3.0	0.1	88.36 ±1.31	88.18 ±3.46
NrFC	3	1.1~3.0	0	90.83 ±0.87	**88.81 ±2.86**
NrFC with L_2	3	1.1~3.0	0.01	90.64 ±0.94	**89.13 ±2.50**
10DB					
T1LSE	4	2.0	0	85.44 ±1.24	84.95 ±3.96
T1LSE with L_2	4	2.0	0.1	85.40 ±1.13	84.95 ±3.96
T2LSE	5	1.1~3.0	0	85.04 ±0.70	**85.93 ±2.42**
TLSE with L_2	5	1.1~3.0	0.1	84.91 ±0.64	**85.93 ±2.42**
NrFC	5	1.1~3.0	0	88.44 ±0.77	85.92 ±3.14
NrFC with L_2	5	1.1~3.0	0.1	87.74 ±0.72	85.92 ±3.14
5DB					
T1LSE	2	1.1	0	79.56 ±1.43	78.86 ±5.46
T1LSE with L_2	2	1.1	0.1	79.56 ±1.43	78.86 ±5.46
T2LSE	4	1.1~2.0	0	79.20 ±1.06	79.78 ±4.85
TLSE with L_2	4	1.1~2.0	0.1	79.20 ±1.06	79.78 ±4.85
NrFC	2	2.0~3.0	0	80.24 ±1.32	**80.31 ±4.39**
NrFC with L_2	4	1.1~3.0	0.1	81.16 ±1.34	**80.62 ±3.83**

TABLE 6 lists the classification accuracies of the classifiers for various noise levels. In the case of 5 dB noise, the

TABLE 9. Comparison results of classification accuracy between proposed classifier and other classifiers in WEKA tool.

Classifiers / Data	Waka				Proposed NrFC	
	MLP	SVM	KNN	C4.5	Without L_2-norm regularization	With L_2-norm regularization
Iris	95.3	96.7	96.7	96.0	96.7	**97.3**
BTSC	77.1	76.1	72.1	77.1	**79.3**	**79.3**
Balance	88.6	87.8	82.6	80.0	**94.7**	**94.7**
Haberman	74.5	73.5	69.3	72.2	74.9	**75.2**

TABLE 10. Comparison results of classification accuracy between proposed classifier and other classifiers in the literature.

Classifiers / Data	GBML	FkNN	GFS	ANFISC	ANFISL	DTSK	Proposed classifier	
							Without L_2-norm regularization	With L_2-norm regularization
Iris	95.3	95.3	96.0	**97.3**	89.3	96.0	96.7	**97.3**
BTSC	74.9	68.2	77.3	77.8	76.2	75.4	**79.3**	**79.3**
Balance	85.1	76.7	67.8	73.5	84.2	72.3	**94.7**	**94.7**
Haberman	71.9	68.9	72.5	73.5	75.2	**76.5**	74.9	75.2

classification accuracy of the type-2 fuzzy set-based classifiers, such as the proposed NrFC and T2LSE, is better than that of T1LSE.

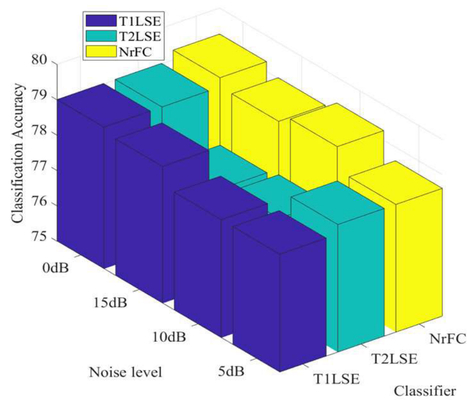
Figure 4 shows the pattern of classification accuracy according to noise change in intensity. The proposed NrFC maintains classification accuracy despite noise effects. However, the other classifiers showed unstable performance changes owing to noise.

C. BALANCE DATA SET

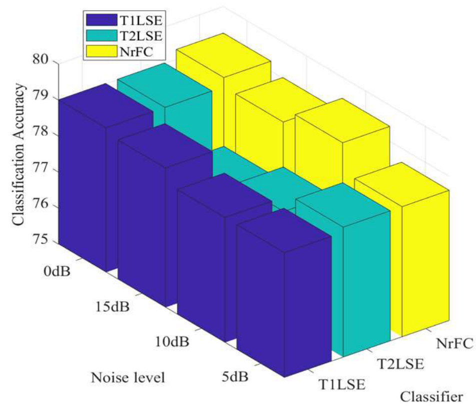
This dataset was generated to model the psychological results. (https://archive.ics.uci.edu/ml/datasets/balance+scale). TABLE 7 shows a performance comparison of the proposed networks and the other two classifiers. Compared to T1LSE and T2LSE, the classification accuracy of the proposed NrFC was improved remarkably.

TABLE 8 lists the classification accuracies of the classifiers for various noise levels. The performance of the classifiers applying IT2FCM was superior to that of the T1LSE classifier. Among the type-2 fuzzy set-based classifiers, NLSE-based learning improves the performance of the classifier.

TABLE 9 summarizes the comparison of the classification accuracies obtained by the proposed classifier and other classifiers. Multinomial logistic regression (MLR), support vector machine (SVM), K-Near Neighbor (KNN), and C4.5 were used with the help of the WEKA toolkit initially developed at the University of Waikato in New Zealand. WEKA (http://www.cs.waikato.ac.nz/ml/weka/) has become popular among academic and industrial researchers and is widely used for educational purposes [31]. Consequently, we can conclude that the classification rate of the proposed NrFC is superior to those of the four comparative classifiers.



(a) Without L_2-norm regularization



(b) With L_2-norm regularization

FIGURE 4. Comparison of the classification accuracy of classifiers according to the change of noise intensity for BTSC dataset.

TABLE 10 shows the comparative results between the proposed classifier and other classifiers previously reported in the literature. GBML [35] is a hybrid algorithm of two fuzzy genetics-based machine learning approaches, FkNN [36] considers the weight (distance) for each nearest neighbor by employing the geometrical relation among the nearest neighbors. GFS [37] is a fuzzy rule-based classifier learned by a novel evolutionary AdaBoost algorithm. ANFIS [38] is an adaptive network-based fuzzy inference system. The model is divided into two types, ANFISC and ANFISL, according to the order of polynomials in the conclusion part. DTKS [39] is a deep TSK fuzzy classifier based on shared linguistic fuzzy rules. The proposed classifier is equal to or superior to other classifiers in the Iris, BTSC, and Balance datasets. In particular, the classification accuracy of the classifier was approximately 10% higher on the Balance dataset.

V. CONCLUSION

In this study, a noise-robust fuzzy classifier was proposed to cope with uncertainty through type-2 fuzzy sets. In the hidden layer, IT2FCM clustering was employed to minimize the effects of noise or outliers. The connection weight is learned using Newton's NLSE-based learning method. In addition, L_2-norm regularization alleviates multicollinearity when training the connection weights. Through the proposed design methodologies, the proposed NrFC significantly ensures noise robustness and improves generalization abilities through L_2-norm regularization by providing a method for model selection through an analysis of the bias-variance tradeoff to alleviate overfitting problems. Consequently, the proposed network outperformed conventional FNNs through the synergy of type-2 fuzzy sets and enhanced learning mechanisms through experiments using several publicly available datasets.

One of the problems with the fuzzy classifier is that there is no dimensionality reduction function. Researchers have applied feature extraction approaches to fuzzy models using machine learning algorithms. However, these methods are performed by considering only the data without the model structure. Therefore, the feature data obtained from the preprocessing could not be suitable for the model. Future studies might focus on developing a dimensionality reduction method for the proposed networks to be applied to high-dimensional big data problems. To do this, the preprocessing part is included in the model, and then the feature data is extracted through model learning. As a result, we would be able to obtain the appropriate feature data for the model structure.

REFERENCES

[1] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey," *Fuzzy Sets Syst.*, vol. 66, no. 1, pp. 1–13, Aug. 1994.

[2] M. M. Gupta and D. H. Rao, "On the principles of fuzzy neural networks," *Fuzzy Sets Syst.*, vol. 61, no. 1, pp. 1–18, Jan. 1994.

[3] 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.

[4] M. Sasaki and M. Gen, "Fuzzy multiple objective optimal system design by hybrid genetic algorithm," *Appl. Soft Comput.*, vol. 2, no. 3, pp. 189–196, Jan. 2003.

[5] Y. E. Hawas, M. Sherif, and M. D. Alam, "Optimized multistage fuzzy-based model for incident detection and management on urban streets," *Fuzzy Sets Syst.*, vol. 381, pp. 78–104, Feb. 2020.

[6] H.-P. Zhang, Y. Ouyang, and B. De Baets, "Constructions of uni-nullnorms and null-uninorms on a bounded lattice," *Fuzzy Sets Syst.*, vol. 403, pp. 78–87, Jan. 2021.

[7] J. Kerr-Wilson and W. Pedrycz, "Generating a hierarchical fuzzy rule-based model," *Fuzzy Sets Syst.*, vol. 381, pp. 124–139, Feb. 2020.

[8] S.-B. Roh, S.-K. Oh, W. Pedrycz, Z. Wang, Z. Fu, and K. Seo, "Design of iterative fuzzy radial basis function neural networks based on iterative weighted fuzzy C-means clustering and weighted LSE estimation," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4273–4285, Oct. 2022.

[9] Z. Wang, C. Yang, S.-K. Oh, Z. Fu, and W. Pedrycz, "Robust multi-linear fuzzy SVR designed with the aid of fuzzy C-means clustering based on insensitive data information," *IEEE Access*, vol. 8, pp. 184997–185011, 2020.

[10] Z. Wang, S.-K. Oh, W. Pedrycz, E.-H. Kim, and Z. Fu, "Design of stabilized fuzzy relation-based neural networks driven to ensemble neurons/layers and multi-optimization," *Neurocomputing*, vol. 486, pp. 27–46, May 2022.

[11] S.-K. Oh and W. Pedrycz, "Fuzzy polynomial neuron-based self-organizing neural networks," *Int. J. Gen. Syst.*, vol. 32, no. 3, pp. 237–250, Jan. 2003.

[12] J.-N. Choi, S.-K. Oh, and W. Pedrycz, "Identification of fuzzy models using a successive tuning method with a variant identification ratio," *Fuzzy Sets Syst.*, vol. 159, no. 21, pp. 2873–2889, Nov. 2008.

[13] W. Pedrycz and K. C. Kwak, "The development of incremental models," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 3, pp. 507–518, Jun. 2007.

[14] W. Pedrycz and K.-C. Kwak, "Linguistic models as a framework of user-centric system modeling," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 4, pp. 727–745, Jul. 2006.

[15] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Design of double fuzzy clustering-driven context neural network," *Neural Netw.*, vol. 104, pp. 1–14, Aug. 2018.

[16] B. Park, J. Choi, W. Kim, and S. Oh, "Analytic design of information granulation-based fuzzy radial basis function neural networks with the aid of multiobjective particle swarm optimization," *Int. J. Intell. Comput. Cybern.*, vol. 5, no. 1, pp. 4–35, Mar. 2012.

[17] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets Syst.*, vol. 122, no. 2, pp. 327–348, Sep. 2001.

[18] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inf. Sci.*, vol. 132, nos. 1–4, pp. 195–220, Feb. 2001.

[19] L. Livi, H. Tahayori, A. Rizzi, A. Sadeghian, and W. Pedrycz, "Classification of type-2 fuzzy sets represented as sequences of vertical slices," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1022–1034, Oct. 2016.

[20] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Design of reinforced interval type-2 fuzzy C-means-based fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3054–3068, Oct. 2018.

[21] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Reinforced hybrid interval fuzzy neural networks architecture: Design and analysis," *Neurocomputing*, vol. 303, pp. 20–36, Aug. 2018.

[22] Z. Wang, C. Yang, S.-K. Oh, and Z. Fu, "Multi-radial basis function SVM classifier: Design and analysis," *J. Electr. Eng. Technol.*, vol. 13, no. 6, pp. 2511–2520, 2018.

[23] B.-J. Park, W. Pedrycz, and S.-K. Oh, "Fuzzy polynomial neural networks: Hybrid architectures of fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 607–621, Oct. 2002.

[24] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimized approximation algorithm in neural networks without overfitting," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 983–995, Jun. 2008.

[25] S.-B. Roh, S.-K. Oh, and W. Pedrycz, "Design of fuzzy radial basis function-based polynomial neural networks," *Fuzzy Sets Syst.*, vol. 185, no. 1, pp. 15–37, Dec. 2011.

[26] C. Yang, Z. Wang, S.-K. Oh, W. Pedrycz, and B. Yang, "Ensemble fuzzy radial basis function neural networks architecture driven with the aid of multi-optimization through clustering techniques and polynomial-based learning," *Fuzzy Sets Syst.*, vol. 438, pp. 62–83, Jun. 2022.

[27] S.-K. Oh, W.-D. Kim, and W. Pedrycz, "Design of radial basis function neural network classifier realized with the aid of data preprocessing techniques: Design and analysis," *Int. J. Gen. Syst.*, vol. 45, no. 4, pp. 434–454, May 2016.

[28] Q. Fan, J. M. Zurada, and W. Wu, "Convergence of online gradient method for feedforward neural networks with smoothing $L_{1/2}$ regularization penalty," *Neurocomputing*, vol. 131, pp. 208–216, May 2014.
[29] C. Hwang and F. C. H. Rhee, "Uncertain fuzzy clustering: Interval type-2 fuzzy approach to C-means," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 107–120, Feb. 2007.
[30] F. Rhee, "Uncertain fuzzy clustering: Insights and recommendations," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 44–56, Feb. 2007.
[31] E.-H. Kim, S.-K. Oh, W. Pedrycz, and Z. Fu, "Reinforced fuzzy clustering-based ensemble neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 3, pp. 569–582, Mar. 2020.
[32] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, NY, USA: Plenum, 1981.
[33] F. Liu and J. M. Mendel, "Aggregation using the fuzzy weighted average as computed by the Karnik–Mendel algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 1–12, Feb. 2008.
[34] I.-C. Yeh, K.-J. Yang, and T.-M. Ting, "Knowledge discovery on RFM model using Bernoulli sequence," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5866–5871, Apr. 2009.
[35] 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.
[36] N. Guler Bayazit and U. Bayazit, "Fuzzy k-NN classification with weights modified by most informative neighbors of nearest neighbors," *J. Intell. Fuzzy Syst.*, vol. 36, no. 6, pp. 6717–6729, Jun. 2019.
[37] M. J. D. Jesus, F. Hoffmann, L. J. Navascués, 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.
[38] L. Maciel, R. Ballini, and F. Gomide, "Adaptive fuzzy modeling of interval-valued stream data and application in cryptocurrencies prediction," *Neural Comput. Appl.*, Jul. 2021, doi: 10.1007/s00521-021-06263-5.
[39] Y. Zhang, H. Ishibuchi, and S. Wang, "Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1535–1549, Jun. 2018.
[40] J. Wang, C. Yang, J. Xia, Z.-G. Wu, and H. Shen, "Observer-based sliding mode control for networked fuzzy singularly perturbed systems under weighted try-once-discard protocol," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 1889–1899, Jun. 2022.
[41] J. Wang, J. Xia, H. Shen, M. Xing, and J. H. Park, "$\mathcal{H}_\infty$ synchronization for fuzzy Markov jump chaotic systems with piecewise-constant transition probabilities subject to PDT switching rule," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 10, pp. 3082–3092, Oct. 2021.
[42] X. Liu, J. Xia, J. Wang, and H. Shen, "Interval type-2 fuzzy passive filtering for nonlinear singularly perturbed PDT-switched systems and its application," *J. Syst. Sci. Complex.*, vol. 34, no. 6, pp. 2195–2218, Jan. 2021.

ZIWU JIANG received the B.Sc. degree in applied mathematics from Qufu Normal University, China, in 2004, and the Ph.D. degree in computational mathematics from the Dalian University of Technology, China, in 2010. He is currently a Professor with the Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi, China. His research interests include numerical approximation, meshless methods, and neural networks.

ZHENG WANG (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Suwon University, Hwaseong, South Korea, in 2012, 2016, and 2021, respectively. He is currently a Postdoctoral Fellow with the Department of Computer Science, Suwon University. His research interests include fuzzy sets and fuzzy systems, evolution computing, and neural networks.

EUN-HU KIM (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Suwon University, Hwaseong, South Korea, in 2009, 2011, and 2016, respectively. From 2017 to 2018, he was a Postdoctoral Fellow at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. From 2018 to 2021, he was a Postdoctoral Fellow at the School of Electrical and Electronic Engineering, Suwon University. He is currently a Professor with the Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi, China. His research interests include fuzzy sets, neural networks, evolutionary algorithms, big data processing, and statistical learning.