IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 13, 2021, accepted December 24, 2021, date of publication January 5, 2022, date of current version January 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140397

Hierarchically Reorganized Multi-Layer Fuzzy Neural Networks Architecture Driven With the Aid of Node Selection Strategies and Structural Network Optimization

ZHENG WANG[1], SUNG-KWUN OH[1,2], (Member, IEEE), EUN-HU KIM[1,2], ZUNWEI FU[2,3], (Member, IEEE), AND WITOLD PEDRYCZ[4,5,6], (Life Fellow, IEEE)

[1]School of Electrical and Electronic Engineering, The University of Suwon, Hwaseong-si, Gyeonggi-do 445-743, South Korea
[2]Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi 276005, China
[3]ICT School, The University of Suwon, Hwaseong-si, Gyeonggi-do 445-743, South Korea
[4]Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada
[5]Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[6]Systems Research Institute, Polish Academy of Sciences, 01447 Warsaw, Poland

Corresponding authors: Sung-Kwun Oh (ohsk@suwon.ac.kr), Eun-Hu Kim (dong2home@gmail.com), and Zunwei Fu (fuzunwei@lyu.edu.cn)

This work was supported in part by Korea Electric Power Corporation under Grant R19XO01-18, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant NRF-2021R1A2C1095739.

ABSTRACT In this study, a design methodology based on fuzzy sets inference and polynomial neural network(PNN) for hierarchically reorganized self-organizing network architecture is introduced to cope with over-fitting as well as multi-collinearity problems which generally appear in a conventional fuzzy neural network. The design method of the proposed self-organizing network structure provides an efficient solution to construct the hierarchically reorganized multi-layer fuzzy neural networks (HRmFNN) architecture through a synergy of multi-techniques such as L_2-norm regularization, probability theory, and multi-optimization. The overall network structure is realized with the aid of parallel network structure with newly added inputs as well as effective neuron selection method through the exponential-based roulette selection technique for each layer in HRmFNN, and the least square error estimation (LSE)-based learning method with L2-norm regularization is used for constructing the stabilized network architecture, and their ensuring design methodologies result in alleviating the overfitting phenomenon and also enhancing the generalization ability. For the performance enhancement of HRmFNN directly affected by some parameters such as the number of input variables, collocation of the specific subset of input variables, the number of membership functions per each variable, and the order of polynomial in the consequent parts of the fuzzy rules, multi-particle swarm optimization (MPSO) is exploited for the effectively structural as well as parametric optimization of the proposed network. That is, the multi-optimization helps achieve a compromise between the better generation performance and the alleviated over-fitting leading to the stabilization of the proposed multi-layered self-organizing network structure with the aid of synergistic multi-techniques such as a) L_2-norm regularization-based LSE learning, b) probability theory for effective neuron selection, and c) novel parallel network structure including newly added inputs and neuron selection method. The performance of the proposed network structure is quantified by comprehensive experiments and comparative analysis. It is also demonstrated through the application to cement compressive strength.

INDEX TERMS Hierarchically reorganized multi-layer fuzzy neural networks (HRmFNN), overfitting, multi-collinearity, exponential-based roulette selection technique, LSE-based learning with L_2-norm regularization, parallel network structure, multi-particle swarm optimization.

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Donato Marino.

I. INTRODUCTION

In recent years, a lot of research has been devoted to machine learning and multi-layered network The research on the stabilization of multi-layer neural networks is on the rise as well [1]. Novel regression algorithms and neural networks have been paid increasing attention to stabilized model prediction [2]–[4]. Many researchers have provided persuasive theories combined with real-world applications, especially in time series by regression algorithm [3]–[6].

As one of the classical neural networks, the Group Method of Data Handling (GMDH) [2] was applied to a great variety of areas including multilayer network and knowledge discovery, prediction and data mining, optimization, and pattern recognition. The GMDH algorithm can automatically find the relationship among the data, select the optimal structure of the network, and improve the accuracy of the existing algorithm [3]–[5]. Polynomial Neural Network (PNN) [6] extended with the aid of the GMDH algorithm come with a flexible architecture whose potential could be utilized by regression models. In PNN, every node exhibits a high level of flexibility and realizes a polynomial type of mapping (viz. linear, quadratic, and cubic) between input and output variables. Fuzzy Polynomial Neural Networks (FPNN) [7] based on PNN and Fuzzy Relation-based Polynomial Neuron (FrPN) with a different type of polynomial were expected to be more flexible generation capability. Especially every node exhibits a high level of flexibility and realizes a polynomial type of mapping (linear, quadratic, and cubic) between input and output variables in FPNN. In the case of the probabilistic or regression model, an over-fitting problem as well as a multi-collinearity problem occurs depending on the structure of the model and the characteristics of the data [10]. In order to alleviate such over-fitting and multi-collinearity problems, we present a novel learning method in fuzzy-neuro models as the solution of the over-fitting and multi-collinearity problem from several points of view:

In the study, we propose a novel hierarchically reorganized network architecture as well as a learning technique for designing multi-layer fuzzy neural networks. The key issues and advantages of constructing HRmFNN are highlighted in the following:

First, in the generation process of each layer of the overall network architecture, the inputs of each layer are reorganized with the use of parallel layer structure newly added through original inputs and front layer outputs. Compared with the fuzzy relation, the fuzzy space divided by the fuzzy set can reduce the number of fuzzy rules, which is equivalent to greatly reducing the parameters required to form the model, thus reducing the complexity of the model and alleviating the emergence with the increase of the number of layers overfitting problem.

Second, the HRmFNN structure that takes a parallel connection network is introduced. As the number of layers of the self-organizing network increases, the similarity of nodes in each layer is very high, and overfitting caused by multicollinearity will occur. Therefore, in the existing network structure, each layer adds the original input, increasing the diversity of the input nodes of each layer, allowing the self-organizing network to fully train while alleviating the overfitting phenomenon caused by multicollinearity.

Third, a new criterion for node selection that increases the diversity of nodes and reduces the complexity of the model is introduced in order to alleviate over-fitting being caused by multi-collinearity problems. in order to reduce the possibility that several potential nodes are discarded on the ground that they do not show good ability in the current layer, we use the exponential-based roulette selection which is a statistical selection method based on the probability which is defined based on the exponential-based performance index.

Forth, we estimate the coefficients by considering the regularization factor in order to reduce the complexity of the proposed model and to alleviate the over-fitting caused by the large deviation between coefficients. To solve this problem, the least square error (LSE) method with L_2-norm regularization is used as a learning method for constructing HRmFNN. So by doing this, we expect that the variance between coefficients in each node will be decreased and thereby the generalization ability will be enhanced.

Fifth, the performance of HRmFNN is directly affected by some hyperparameters such as the number of input variables, collocation of the specific subset of input variables, the number of rules, and the orders of the polynomials in the consequent parts of the rules. The structure and parameters of HRmFNN are optimized by MPSO to achieve better performance. Three objective functions including performance are used to evaluate the accuracy, complexity, and interpretability of HRmFNN. The objective functions are the performance of the model, the entropy of partition, and the sum of squared coefficients in HRmFNN to be estimated. Three objective functions used in the optimization process would form a sound tradeoff between better generation performance and alleviating the over-fitting problem. The proposed multi-optimization method is exploited to carry out the structural and parametric optimization of the hierarchically reorganized network architecture for the enhancement of generation capability through the minimization of complexity and the maximization of accuracy.

Finally, when it comes to the structural design, techniques such as parallel network structure and MPSO as well as algorithmic design methodologies such as exponential-based roulette selection technique and LSE-based learning with L_2-norm regularization, and also their ensuing synthesis technologies of the proposed HRmFNN could lead to the realization of the stabilized multi-layered network architecture through the alleviation of over-fitting and multi-collinearity.

The HRmFNN architecture is not fixed in advance as in case of a self-organizing network but becomes organized through the growth process of the generation of the layers and nodes(neurons) of the network. Along with the use of synergistic design methodologies such as exponential-based roulette selection technique, LSE-based learning, and two types of parallel network structures as shown in section 2, the structural and parametric multi-optimization of the network through them is more effectively carried out for constructing the stabilized as well as an enhanced multi-layered network of HRmFNN. First, in the design process of HRmFNN, the structure of fuzzy set-based polynomial neuron (FsPN) as a node (neuron) of each layer of the network is optimized. Four kinds of structural parameters of FsPN contain a) the number of input variables, b) a collocation of the specific subset of input variables, c) the number of membership functions, and d) the type of polynomial. Next, after obtaining the optimized FsPN, the optimization design of both nodes selected in the current layer and their ensuing layer (next layer) leads to the optimized HRmFNN structure [41]–[44].

In the sequel, the main contributions of our work can be summarized in a concise way as follows: First, HRmFNN structure consisting of the original network based on a fuzzy set as well as a newly added parallel network is proposed in order to get better performance in the stabilized multi-layer network structure. Second, in order to alleviate over-fitting being caused by multi-collinearity problems, the exponential-based roulette selection technique for node selection and LSE-based learning with L_2-norm regularization is proposed. Third, the synergistic effect of a) structural design techniques such as parallel network structure and MPSO as well as b) algorithmic design methodologies such as exponential-based roulette selection technique and LSE-based learning with L_2-norm regularization leads to the stabilized multi-layered network architecture through the alleviation of over-fitting and multi-collinearity.

The structure of this study is organized as follows. Section 2 elaborates on the architectural framework of hierarchically reorganized multi-layer fuzzy neural networks. Section 3 provides a multi-optimization of the overall framework of HRmFNN with the aid of MPSO. Section 4 reports a comprehensive set of experiments. Finally, concluding remarks are shown in Section 5.

II. ARCHITECTURAL FRAMEWORK OF HIERARCHICALLY REORGANIZED MULTI-LAYER FUZZY NEURAL NETWORKS

This section elaborates on the design of HRmFNN based on polynomial reasoning. The main differences between the previous works (FPNN) and the proposed HRmFNN are shown in Table 1. A detailed explanation of the differences and the improvements by the proposed techniques are summarized as follows.

1) *Division of fuzzy space:* The fuzzy space divided by the fuzzy set can reduce the number of fuzzy rules, thus reducing the complexity of the model and alleviating the emergence with the increase of the number of layers Overfitting problem.

2) *Structure: Parallel network structure including newly added inputs. —— See Section 2.B and Figure 2.*

When compared with the previous studies, the proposed network structure leads to the stabilization of the unstable network structure, which is caused by overfitting and multicollinearity of the high similarity between each input node in the multi-layer network structure.

3) *Performance criteria for neurons to be selected in each layer: Probability theory for effective neuron selection through the exponential-based roulette selection technique. —— See Section 2.C.*

The nodes in FPNN are selected based on the performance index (PI) of the training dataset. Thus, the model is over-fitted toward the training dataset, and this design methodology results in the degradation of generalization ability.

In this study, we consider the probability mechanisms for effective neuron selection through the exponential-based roulette selection. The roulette nodes selection based on the probability theory enables the network structure to alleviate the overfitting phenomenon and to get better EPI in the nodes of each layer as a compromise technique (exponential-based roulette selection) in the proposed network.

4) *Learning method: L_2-norm regularization-based LSE learning. -See Section 2.D.*

In the existing FPNN, the coefficients of the consequent part in a fuzzy rule are trained by LSE-based learning. As the size of the layer increases, the pattern of the input variables in the node becomes very similar, and this leads to the overfitting caused by the multicollinearity problem in LSE learning. The singular matrix by the multicollinearity interrupts the training of coefficients, and this phenomenon more frequently occurs in the deep layer.

To overcome such a problem, in this study, we apply LSE-based learning with the aid of L_2-norm regularization. The learning process is the same with FPNN, but the problem by multicollinearity is alleviated by the L_2-norm regularization and as a result, it helps the proposed model to be deeper architecture than the FPNN.

5) *Optimization: Optimization of the node's structure by multi-particle swarm optimization. -See Section 3, and Figures 5, 6, and 7.*

The structure of each node is determined by particle swarm optimization. Because the node consists of a fuzzy model, we require the proper selection of structural factors such as the number of input variables, selected input variables, the number of membership functions, and polynomial types. The previous models (FPNN) use a single objective function-based genetic algorithm, and the objective function is the performance index (PI) determined by the training dataset. As a result, the optimization technique using the PI-based objective function may lead to the risk of overfitting.

In this study, we consider multi-particle swarm optimization (MPSO) that uses multi-objective functions than one objective function to prevent the overfitting of the model. In the proposed model, three objective functions, such as weighted Performance Index (MPI), Sum of Squared Coefficients (SSC), and Entropy (H), are used to consider the performance, the deviation of coefficients, and the structural complexity, respectively.

TABLE 1. Comparison of design methodologies used in FPNN and HRmFNN.

Design point	Design Approach	FPNN	Proposed HRmFNN and key points	
Different points	1)Type of neurons	Fuzzy relation rule-based neurons	Fuzzy set rule-based neurons	Reduction of number of rules parameters, and complexity
	2)Structure (Section 2.B and Fig. 2)	Generic network structure	Newly inputs added parallel layer-based network structures	Increase of node diversity o alleviate overfitting and improve performance
	3)Performance criteria for neurons to be selected in each layer (Section 2.C)	PI	Roulette neurons selection technique	Alleviation of over-fitting and multi-collinearity
	4) Learning method (Section 2.D)	LSE-based learning	LSE-based learning with L2-norm regularization	Alleviation of over-fitting and multi-collinearity
	5)Optimization (Section 3, and Fig. 5, 6, and 7)	GA	Multi-PSO (MPSO)	Generation of nearly optimized network structure for reduction of both bias and variance as well as alleviation of over-fitting and multi-collinearity

A. HIERARCHICALLY REORGANIZED MULTI-LAYER FUZZY NEURAL NETWORKS BASED ON FUZZY SET-BASED POLYNOMIAL NEURONS

Hierarchically reorganized multi-layer fuzzy neural networks (HRmFNN) is based on ‘if-then’ rule-based fuzzy network with the extended structure of the premise and the consequence parts of the fuzzy rules. The layer consists of fuzzy set-based polynomial neurons (FsPN) generated by newly added parallel layer structure. These neurons are fully reflective of regularity (inherent pattern) involved in numeric data, which are granulated with the aid of fuzzy rule and fuzzy set inference. Polynomial neuron dwells on the concepts of a collection of fuzzy membership function and nonlinear polynomial processing. The number of input variables and their membership functions realized by the input variables implies the partitions of the input space and could be constructed by considering some relationships between inputs and output [17]. The FsPN encapsulates a family of nonlinear “if-then” rules. When arranged together, FsPN forms a neural network architecture. This neuron, which is regarded as a generic type of processing unit, dwells on the concepts of fuzzy sets and neural networks. As visualized in figure 1, the FsPN consists of two basic functional modules. The first one, the fuzzy part based on input and output experimental data has become the standard method for dealing with uncertain nonlinear dynamic systems [1], [7]. The fuzzy reasoning usually discussed in the literature attempts to decompose the input space into fuzzy subspaces and then approximates the system in each subspace by linear regression models [12], [22]. The rules of each node of HRmFNN are constructed by means of fuzzy partitioning of spaces that is implemented based on the fuzzy granularity of input space. The approach taken to obtain the shape of any particular membership function is usually oriented to a given application. Usually, we consider trigonometric and Gaussian membership functions here. The second part refers to the function-based polynomial processing that involves some input variables. The activation levels of the individual rules contribute to the output of the FsPN, which is computed as a weighted average of the individual condition parts $P_j(x)$. The fuzzy rules of FsPN are shown as follows

$$R^j: \text{ If } x_k \text{ is } u_j, \quad \text{then } P_j(x), \tag{1}$$

where R^j is the j-th fuzzy rule, x_k is the k-th input variable, u_j is the membership value of the j-th fuzzy rule and the $P_j(x)$ stands for a polynomial of the consequence part in FsPN. The polynomial types used in the proposed model are shown in Table 2. As shown in figure 1, the notation used in the figure requires some clarification. The “circles” denote units of the FPN, “N” refers to a normalization procedure that is applied to the membership grades, “$\sum$” is the product and the summation operations of all incoming signals, respectively. The output z of FsPN is determined as follows:

$$z = \frac{\sum_{j=1}^{K} u_j P_j(\mathbf{x})}{\sum_{j=1}^{K} u_j} = \sum_{j=1}^{K} \hat{u}_j P_j(\mathbf{x}), \quad \hat{u}_j = \frac{u_j}{\sum_{j=1}^{K} u_j}, \tag{2}$$

where K stands for the number of fuzzy rules. $\hat{u}_j$ is the normalized fuzzy membership value of the j-th fuzzy rule.

As shown in figure 1. Compared with the fuzzy relation, the fuzzy space divided by the fuzzy set can reduce the number of fuzzy rules, which is equivalent to greatly reducing the parameters required to form the model, thus reducing the complexity of the model and alleviating the emergence with the increase of the number of layers overfitting problem.

B. ARCHITECTURAL DESIGN OF HIERARCHICALLY REORGANIZED MULTI-LAYER FUZZY NEURAL NETWORKS

In contrast to the typical architectures encountered in FPNN, the main challenges in this study are targeted at stabilizing the multilayer network structure designed with the aid of learning method and structural changes for improving the performance of HRmFNN.

As shown in (1) of Figure 2, the conventional FsPN-based multi-layered structure as well as design methodology in generic FPNN as the previous works leads to unstable network structure caused by over-fitting and multi-collinearity

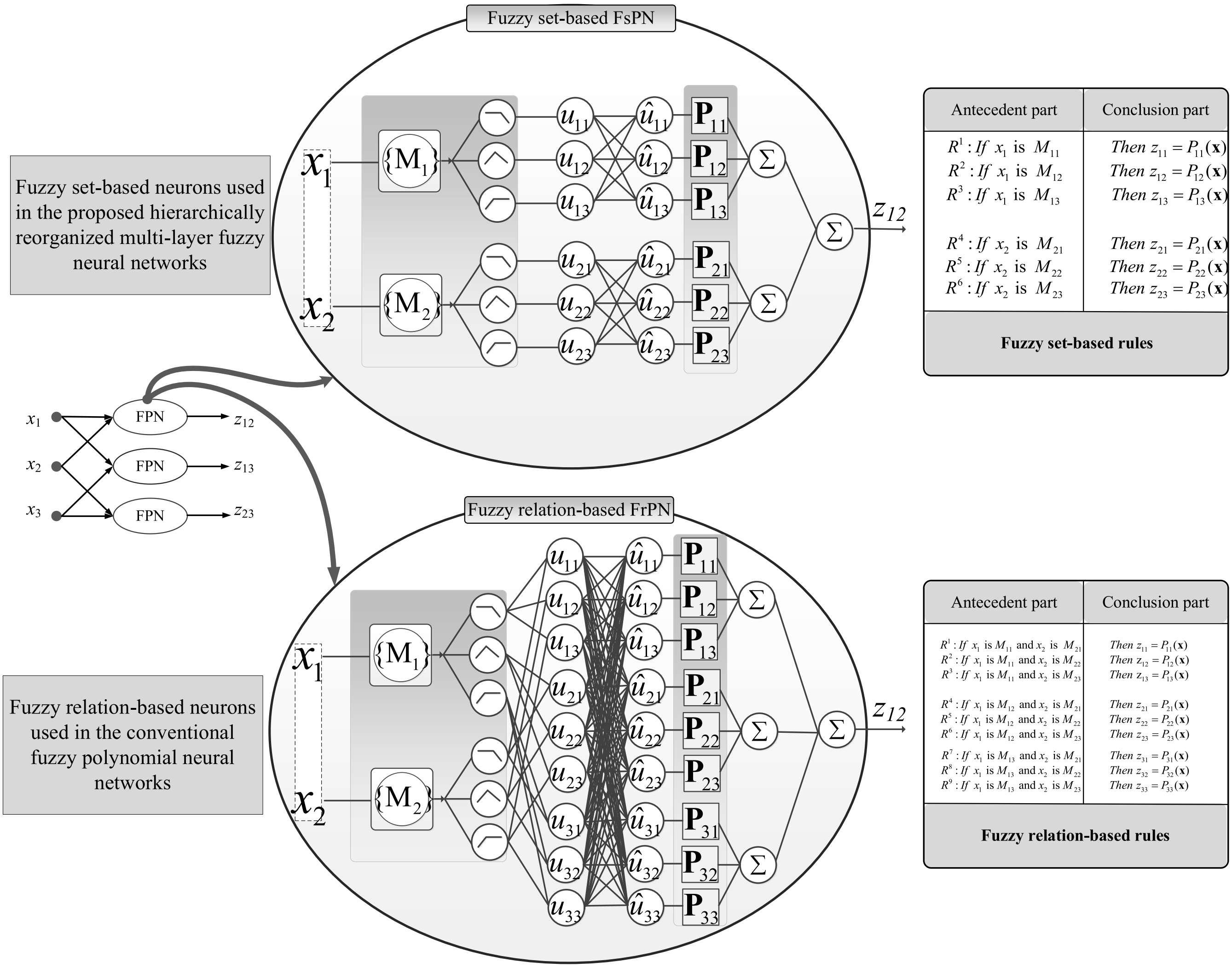
TABLE 2. Polynomial type forms in fuzzy set-based polynomial neutrons.

Type	Polynomial
Constant	$P_j = a_{j0}$
Linear	$P_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k$
Quadratic	$P_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k + a_{j(k+1)}x_1{}^2 + \cdots$ $+a_{j(2k)}x_k{}^2 + a_{j(2k+1)}x_1x_2 + \cdots + a_{j((k+2)(k+1)/2)}x_{k-1}x_k$
Constant	$P_j = a_{j0}$
Linear	$P_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k$
Modified Quadratic	$P_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jk}x_k$ $+a_{j(k+1)}x_1x_2 + a_{j(k(k+1)/2)}x_{k-1}x_k f_j = a_{j0}$

being occasionally occurred in the multi-layered network structure [21]. Also, bias and variance of the model affect the over-fitting and under-fitting, the bias denotes the degree of deviation between the expected output of the learning algorithm and the target output, which means the fitting ability of the algorithm (model). The high bias indicates that the prediction result (model output) is very different from the real result (target output) [10]. Variance represents the difference between training and test results in the same model (viz. model constructed by N-fold cross-validation method). Highly complex models become unstable based on changes in training data and testing data. Low bias and low variance will produce a low error, but low bias and low variance are often not compatible. If we want to reduce the bias of the model, it will increase the variance of the model to some extent, and vice versa [22].

As shown in (2) and (3) in Figure 2, in the proposed HRmFNN structure, two types of parallel connection network are additionally considered as the previous layer-based parallel network structure and the 1st layer (with original input variables)-based parallel network structure for the stabilization (viz. the reduction of both bias and variance) of unstable network structure caused by over-fitting and multi-collinearity

As shown in (4) of figure 2, the lower part of figure 2 shows the bias-variance tradeoff according to the change of two factors for model complexity. figure 2(a) shows an under-fitting


Fuzzy set-based FsPN
Fuzzy set-based neurons used in the proposed hierarchically reorganized multi-layer fuzzy neural networks
x_1
x_2
{M$_1$}
{M$_2$}
u_{11} u_{12} u_{13} u_{21} u_{22} u_{23}
$\hat{u}_{11}$ $\hat{u}_{12}$ $\hat{u}_{13}$ $\hat{u}_{21}$ $\hat{u}_{22}$ $\hat{u}_{23}$
P$_{11}$ P$_{12}$ P$_{13}$ P$_{21}$ P$_{22}$ P$_{23}$
Σ
z_{12}
x_1 x_2 x_3
FPN
FPN
FPN
z_{12} z_{13} z_{23}
Fuzzy relation-based FrPN
Fuzzy relation-based neurons used in the conventional fuzzy polynomial neural networks
u_{11} u_{12} u_{13} u_{21} u_{22} u_{23} u_{31} u_{32} u_{33}
$\hat{u}_{11}$ $\hat{u}_{12}$ $\hat{u}_{13}$ $\hat{u}_{21}$ $\hat{u}_{22}$ $\hat{u}_{23}$ $\hat{u}_{31}$ $\hat{u}_{32}$ $\hat{u}_{33}$
P$_{11}$ P$_{12}$ P$_{13}$ P$_{21}$ P$_{22}$ P$_{23}$ P$_{31}$ P$_{32}$ P$_{33}$

Antecedent part	Conclusion part
$R^1 : If\ x_1$ is M_{11}	$Then\ z_{11} = P_{11}(\mathbf{x})$
$R^2 : If\ x_1$ is M_{12}	$Then\ z_{12} = P_{12}(\mathbf{x})$
$R^3 : If\ x_1$ is M_{13}	$Then\ z_{13} = P_{13}(\mathbf{x})$
$R^4 : If\ x_2$ is M_{21}	$Then\ z_{21} = P_{21}(\mathbf{x})$
$R^5 : If\ x_2$ is M_{22}	$Then\ z_{22} = P_{22}(\mathbf{x})$
$R^6 : If\ x_2$ is M_{23}	$Then\ z_{23} = P_{23}(\mathbf{x})$
Fuzzy set-based rules	

Antecedent part	Conclusion part
$R^1 : If\ x_1$ is M_{11} and x_2 is M_{21}	$Then\ z_{11} = P_{11}(\mathbf{x})$
$R^2 : If\ x_1$ is M_{11} and x_2 is M_{22}	$Then\ z_{12} = P_{12}(\mathbf{x})$
$R^3 : If\ x_1$ is M_{11} and x_2 is M_{23}	$Then\ z_{13} = P_{13}(\mathbf{x})$
$R^4 : If\ x_1$ is M_{12} and x_2 is M_{21}	$Then\ z_{21} = P_{21}(\mathbf{x})$
$R^5 : If\ x_1$ is M_{12} and x_2 is M_{22}	$Then\ z_{22} = P_{22}(\mathbf{x})$
$R^6 : If\ x_1$ is M_{12} and x_2 is M_{23}	$Then\ z_{23} = P_{23}(\mathbf{x})$
$R^7 : If\ x_1$ is M_{13} and x_2 is M_{21}	$Then\ z_{31} = P_{31}(\mathbf{x})$
$R^8 : If\ x_1$ is M_{13} and x_2 is M_{22}	$Then\ z_{32} = P_{32}(\mathbf{x})$
$R^9 : If\ x_1$ is M_{13} and x_2 is M_{23}	$Then\ z_{33} = P_{33}(\mathbf{x})$
Fuzzy relation-based rules	

FIGURE 1. Compare the difference between FsPN and FrPN in the proposed HRmFNN.

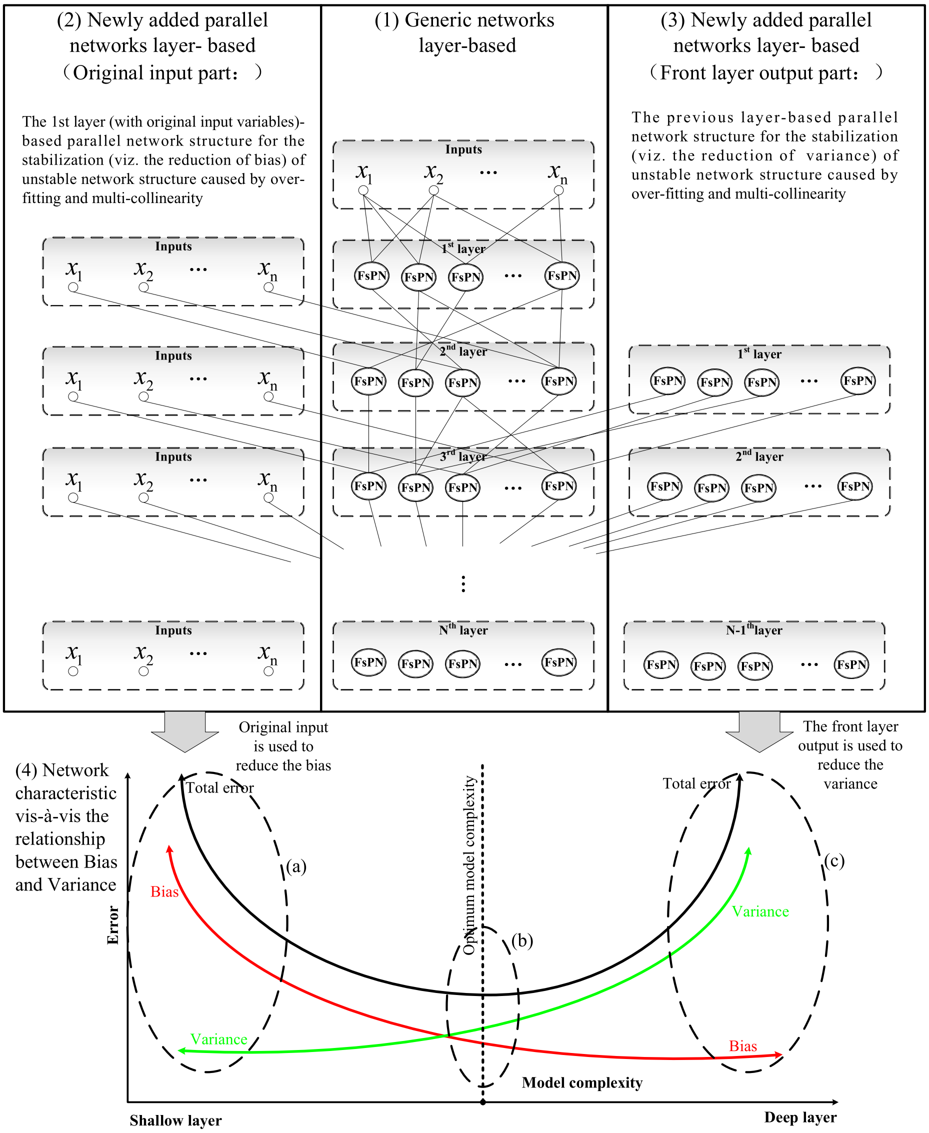

(2) Newly added parallel networks layer- based （Original input part：）
(1) Generic networks layer-based
(3) Newly added parallel networks layer- based （Front layer output part：）
The 1st layer (with original input variables)-based parallel network structure for the stabilization (viz. the reduction of bias) of unstable network structure caused by over-fitting and multi-collinearity
The previous layer-based parallel network structure for the stabilization (viz. the reduction of variance) of unstable network structure caused by over-fitting and multi-collinearity
Inputs
x_1 x_2 ··· x_n
1st layer
2nd layer
3rd layer
Nth layer
N-1th layer
FsPN
Original input is used to reduce the bias
The front layer output is used to reduce the variance
(4) Network characteristic vis-à-vis the relationship between Bias and Variance
Error
Total error
Bias
Variance
(a)
(b)
(c)
Optimum model complexity
Model complexity
Shallow layer
Deep layer

FIGURE 2. Structure of HRmFNN; "bias": the difference between target out and model output, "variance": the difference between the outputs of individual models (models constructed through N fold cross-validation method).

model with high deviation and low variance, figure 2(b) shows the low deviation and low variance considered as nearly optimized network structure of the proposed model, and figure 2(c) shows an over-fitting model with low deviation and high variance, Ideally, the smaller the bias and variance are, the better the performance of the model becomes, but it is closely impossible because most models exposure the problem called the bias-variance dilemma.

A summary of the characteristics of the conventional model (FPNN) and the proposed HRmFNN evaluated vis-à-vis the relationship between bias and variance is presented as follows:

a) In the case of Bias: High and Variance: Low

In the early stage of training (up to the first three layers as shallow layers), the model's ability to fit is not strong enough, thus the bias is relatively large, and the fitting ability is not strong. That is the case of under-fitting by low variance and high bias as shown in figure 2 (a).

b) In the case of Bias: Low and Variance: Low

figure 2 (b) shows the bias-variance tradeoff of the optimized model. That is, low bias and low variance will get a low error, however low bias and low variance are not compatible in most cases. If we want to reduce the bias of the model by increasing the complexity of the model, it will increase the variance of the model to some extent, and vice versa. Therefore, we need to determine the proper structural design parameters such as the number of layers and rules, as well as methodological design parameters related to both the proposed performance criterion and learning technique in order to get performance close to low bias and low variance in HRmFNN.

Especially, In HRmFNN, we take into consideration a parallel architecture with the newly added layers. when constructing the proposed multi-layered network architecture, two types of parallel network structures are considered as shown in (2) of figure 2.

c) In the case of Bias: Low and Variance: High

Under the deepening of the training level (caused by an increase of layers as well as fuzzy rules), the fitting ability of the model is gradually improved, and even the noises and outliers in the training data can be trained through the network model. After completing sufficient training (up to deep layers), the model's ability to fit is very strong, and the slight disturbance such as noise and outlier of the training data will lead to significant changes like the increase of variance of the model. As a result, such changes result in over-fitting of the test results as shown in figure 2 (c), but the proposed design methodologies lead to alleviating the overfitting as well as decreasing the variance of the proposed model.

C. EXPONENTIAL-BASED ROULETTE SELECTION TECHNIQUE

The better the modeling performance of a node is, the higher the possibility that the associated node can be chosen becomes. This means that a node, which is inferior to other better nodes in the current state and may have the potential to enhance the performance of the whole networks later, cannot be selected to compose the current layer. In other words, the selection after sorting in terms of the modeling performance may discard the useful nodes which can contribute to the improvement of the performance.

In order to reduce the possibility that several potential nodes are discarded on the ground that they do not show good ability in the current layer, we use the roulette wheel selection which is a statistical selection method based on the probability which is defined based on the performance index. In this paper, we use RMSE as the performance index.

$$PI = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}. \tag{3}$$

where $\hat{y}$ is the output of the model, and m is the number of data. The probability of each node being selected among all nodes is inversely proportional to its performance index (RMSE). In other words, the better the model performance of each node, the smaller the error and the higher the probability of this note being selected. Here we use exponential function as the fitness function of the probability of each node being selected. The fitness exponential function used is shown in figure 3.

The fitness function of the k-th node is as follows

$$f_k = e^{-PI_k}. \tag{4}$$

The probability P_k for each individual is defined in the for

$$P_k = \frac{f_k}{\sum\limits_{n} f_n}, \tag{5}$$

where f_k means the fitness value of the k-th node.

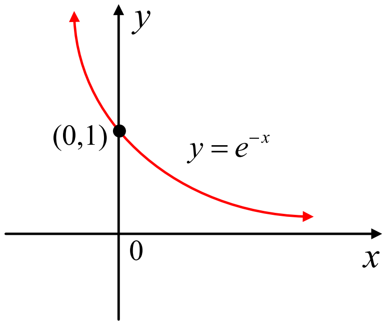


FIGURE 3. **Exponential-based fitness function.**

A series of k random numbers are generated and compared against the cumulative probability $Q_\mathrm{i} = \sum_{\mathrm{j=1}}^{\mathrm{i}} \mathrm{P_i}$ of the nodes. The appropriate individual k is selected and applied as a new input variable, which can be used as input variables of a node in the next layer if the following condition is satisfied.

$$Q_{i-1} < U(0,1) \leq Q_i, \tag{6}$$

where U(0,1) denotes a random number derived from the uniform distribution over [0,1].

Algorithm 1 describes each step of the exponential-based roulette selection (ERS) technique in detail, and figure 4 shows the entire ERS process.

Algorithm 1 Exponential-Based Roulette Selection Technique

```
Input: Original nodes (FsPN) in the current layer
Output: Selected nodes (FsPN)
1: for layer = $1 \ldots L$ do
2:     Set max number of selected neurons ($G^*$), number of neuron
       combinations ($G^c$), adjust factor ($\theta$) and selection pressure ($\eta$)
3:     Calculate the performance index of each node through Eq. (3)
4:     Sort all nodes of current layer based on their performance
5:     Adjust $G^*$ dynamically according to Eq. (4) and $G^*$
6: if $G^c \geq G^*$ then
7:     Calculate the selection probability of each node through Eq. (5) and
   Eq. (6)
8: Compute the cumulative probability and construct the probability pool
   by Eq. (6)
9:     Randomly select its corresponding node from the probability pool
10:    Repeat steps 2-9 until the number of selected nodes is equal to $G^*$
11: else
12:    Select all nodes of current layer
13: end if
14: end for
```

D. LEARNING DESIGN OF ENHANCED ENSEMBLE FUZZY SET-BASED POLYNOMIAL NEURAL NETWORK

The goal of most models is targeted at enhancing the generalization ability from training data in order to make the good predictive ability for unseen data. Overfitting happens when too much learning of model is carried out from training dataset including noise and outlier data, so the constructed model couldn't generalize well, and the poor predictive performance is caused by testing dataset. Especially it occurs as a very common problem when the dataset is too small when compared with the number of model parameters that

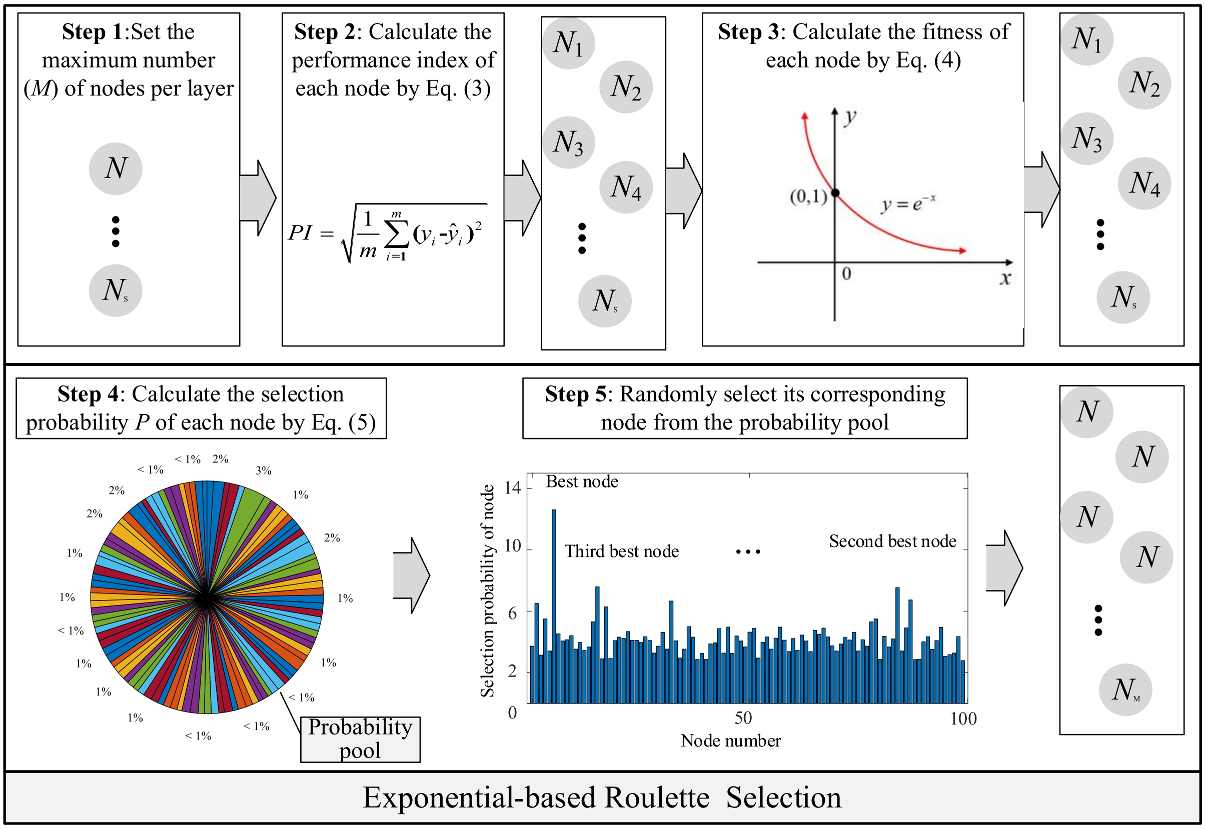

Step 1:Set the maximum number (M) of nodes per layer
N
N_s
Step 2: Calculate the performance index of each node by Eq. (3)
$PI = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$
N1
N2
N3
N4
N_s
Step 3: Calculate the fitness of each node by Eq. (4)
y
(0,1)
y = e^-x
0
x
N1
N2
N3
N4
N_s
Step 4: Calculate the selection probability P of each node by Eq. (5)
< 1%
< 1%
2%
3%
1%
2%
2%
2%
1%
1%
1%
< 1%
1%
1%
1%
1%
< 1%
< 1%
< 1%
Probability pool
Step 5: Randomly select its corresponding node from the probability pool
Selection probability of node
14
10
6
2
0
Best node
Third best node
Second best node
50
100
Node number
N
N
N
N
N_M
Exponential-based Roulette Selection

FIGURE 4. Exponential-based roulette selection technique of HRmFNN.

need to be learned. This problem is particularly acute in the multi-layered FPNN with lots of parameters.

Regularization is an important method in alleviating overfitting [20]. Furthermore, some techniques of regularization can be used to reduce model capacity while maintaining accuracy, for example, to drive some values of the parameters to zero.

In the regularization technique, the weight penalty is a standard way and widely used for the training of the model. The penalties try to keep the weights (coefficients) small or non-zero except the big gradients counteracting, which makes models more interpretable. An alternative name for weight penalties is called "weight decay" in the regularization technique.

The penalty term is specified as the square of coefficients of a regression model, which is called L_2-norm regularization in machine learning, shrinkage in statistics, and weight decay in neural networks. λ is called the regularization coefficient and controls how much we value fitting the data well, as a simple hypothesis.

In fact, many fuzzy neural networks have been encountered in the overfitting problem. As the most common and simplest kind of parameter norm penalty, L_2-norm regularization is one of the effective methods to alleviate this problem. During the design process, the L_2 penalty term is added to the objective function. This method is used to reduce the variation between coefficients being led to multi-collinearity as well as prevent the degradation of generalization ability.

$$E = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \lambda \sum_{p=1}^{n}(a_p)^2. \tag{7}$$

Here, λ stands for a regularization parameter and a_p are the coefficients of the polynomial.

Assume that $(x_1, x_2, \ldots, x_k)$ and y represent the input and output of training data, then the least square error estimation (LSE)-based learning method with L_2-norm regularization algorithm can be summarized as shown in Algorithm 2.

Algorithm 2 Least Square Error Estimation (LSE)-Based Learning Method With L_2-Norm Regularization

```
Input: $(x_1, x_2, \ldots, x_k)$, Y
Output: Coefficient matrix A
1. Set regularization factor $\lambda$
2. Set $X = (1, x_1, x_2, \ldots, x_k)$, Y
3. Take the partial difference of error function
     $E = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \lambda \sum_{p=1}^{n}(a_p)^2$ and set it to be zero
4. Estimate the coefficients A by using the formula
     $A = (X^T X + \lambda I)^{-1} X^T Y$
```

For convenience, we consider a linear format of polynomial in the following way:

$$f(x) = a_0 + a_1 x_1 + \cdots + a_k x_k. \tag{8}$$

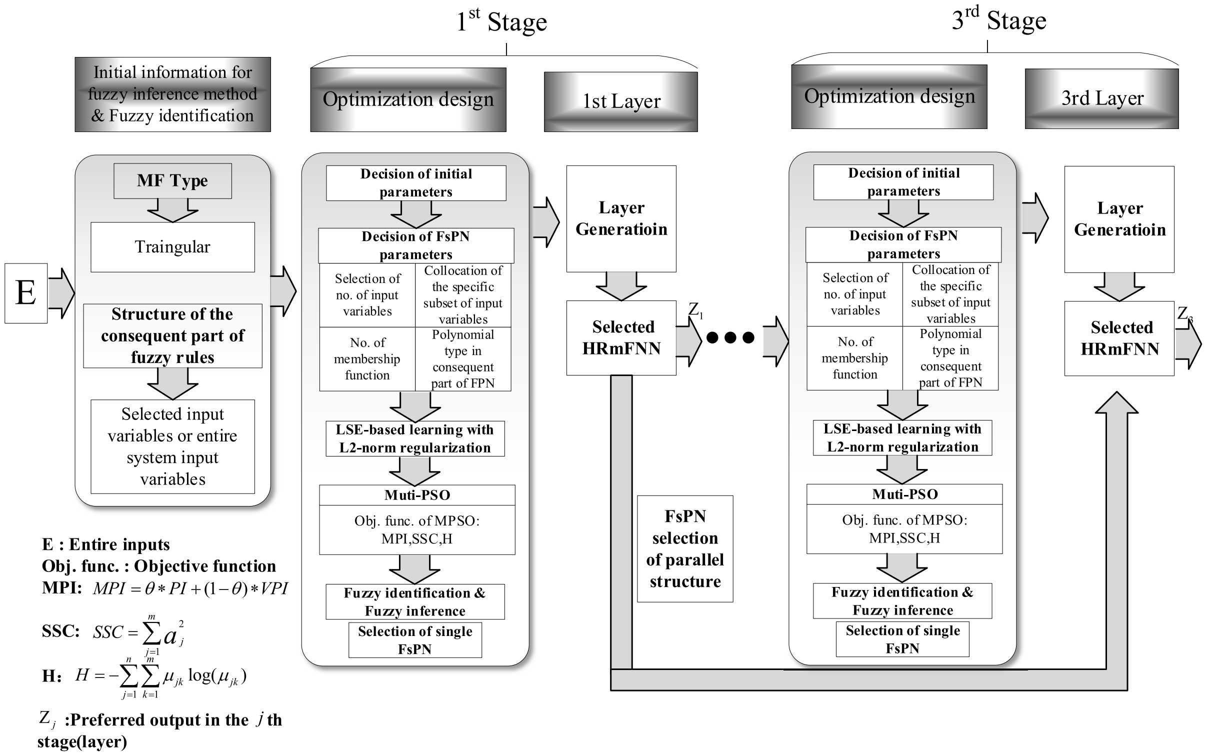

1st Stage
3rd Stage
Initial information for fuzzy inference method & Fuzzy identification
Optimization design
1st Layer
Optimization design
3rd Layer
E
MF Type
Traingular
Structure of the consequent part of fuzzy rules
Selected input variables or entire system input variables
Decision of initial parameters
Decision of FsPN parameters
Selection of no. of input variables
Collocation of the specific subset of input variables
No. of membership function
Polynomial type in consequent part of FPN
LSE-based learning with L2-norm regularization
Muti-PSO
Obj. func. of MPSO: MPI,SSC,H
Fuzzy identification & Fuzzy inference
Selection of single FsPN
Layer Generatioin
Selected HRmFNN
Z_1
FsPN selection of parallel structure
Layer Generatioin
Selected HRmFNN
Z_3
E : Entire inputs
Obj. func. : Objective function
MPI: $MPI = \theta * PI + (1-\theta) * VPI$
SSC: $SSC = \sum_{j=1}^{m} a_j^2$
H: $H = -\sum_{j=1}^{n}\sum_{k=1}^{m} \mu_{jk}\log(\mu_{jk})$
Z_j :Preferred output in the j th stage(layer)

FIGURE 5. Overall scheme of the optimization process of HRmFN.

[Step1] Determine the combination of input variables for the first layer.

Calculate all combinations of input variables to build up the polynomial neuron in the next layer.

[Step2] Calculate the value of the membership function by a triangular membership function.

$$A(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \tag{9}$$

where a, b, and c denote the centers of membership functions.

[Step3] Normalize the firing strength of each fuzzy rule.

$$u_j = \prod_{j=1}^{K} A_i^j(x_i), \tag{10}$$

$$\hat{u}_j = \frac{u_j}{\sum_{j=1}^{K} u_j}, \tag{11}$$

where u_j is the fuzzy membership value of the j-th rule and $\hat{u}_j$ is the fuzzy membership value of the j-th rule after normalization.

[Step4] For convenience, the expression can be described as follows:

$$Y = A_i X_i.$$

where $\boldsymbol{A}_i$ is the vector of coefficients of i-th consequent polynomial and $\boldsymbol{X}_i$ is a matrix that includes input data with fitness. In case the consequent polynomial is linear, $\boldsymbol{X}_i$ and $\boldsymbol{A}_i$ read as shown at the bottom of the page.

According to the LSE method, the coefficient can be calculated by the following expression:

$$A_i = (X_i^T X_i)^{-1} X_i^T Y. \tag{12}$$

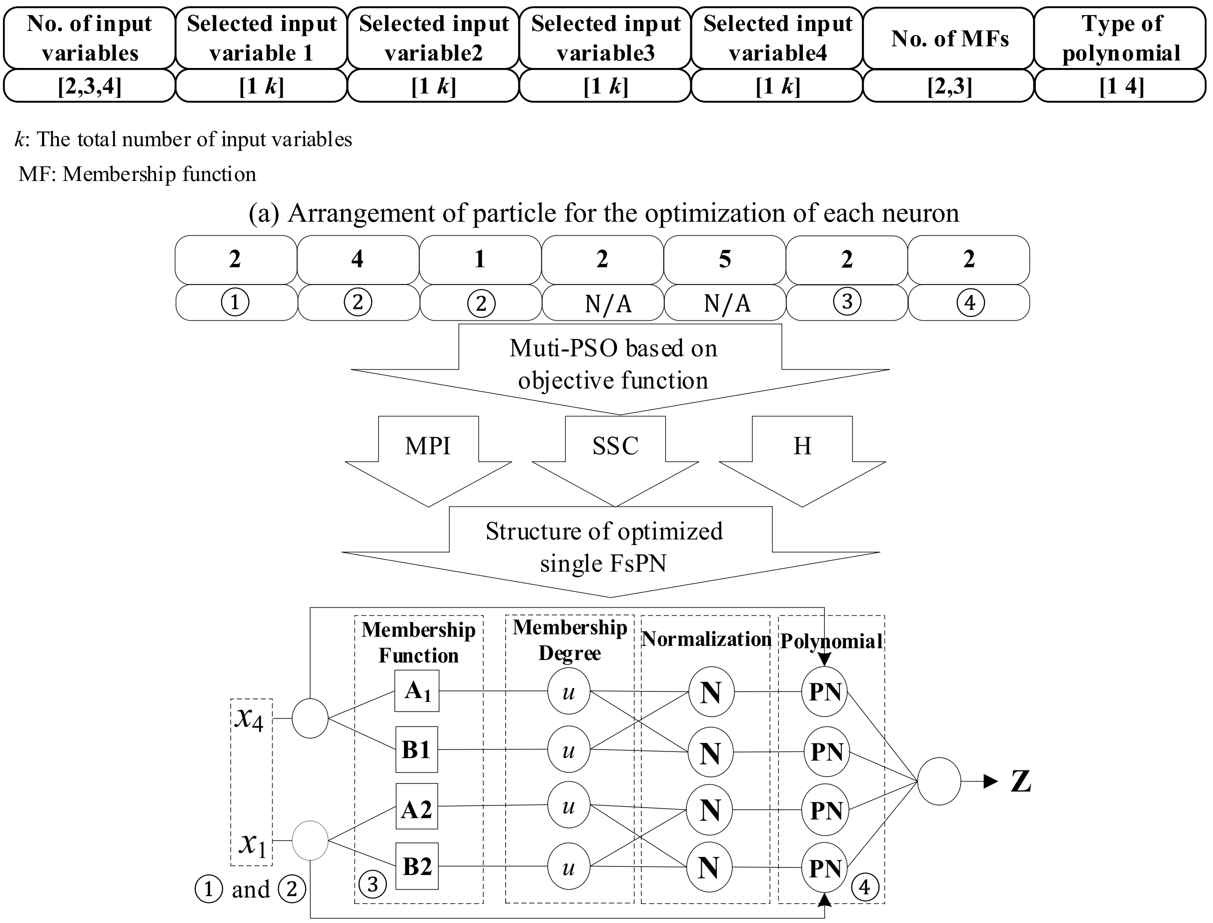
The HRmFNN sometimes causes an overfitting problem with the increasing number of layers. To alleviate the overfitting problem, we use L2–norm regularization method that adds the penalty term to the cost function in the following way:

$$A_i = (X_i^T X_i + \lambda I)^{-1} X_i^T Y. \tag{13}$$

where I is the unit matrix, and λ denotes a regularization parameter. In this study, λ is fixed at 0.01. The coefficients of the polynomial are estimated by the least square

$$X_i = \begin{bmatrix} \hat{u}_{11} & \cdots & \hat{u}_{n1} & x_{11}\hat{u}_{11} & \cdots & x_{11}\hat{u}_{n1} & \cdots & x_{k1}w_{11} & \cdots & x_{k1}\hat{u}_{n1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \hat{u}_{1m} & \cdots & \hat{u}_{nm} & x_{1m}\hat{u}_{1m} & \cdots & x_{1m}\hat{u}_{nm} & \cdots & x_{km}w_{1m} & \cdots & x_{km}\hat{u}_{nm} \end{bmatrix}^T$$

$$A_i = \begin{bmatrix} a_{10} & \cdots & a_{n0} & a_{11} & \cdots & a_{n1} & \cdots & a_{1k} & \cdots & a_{nk} \end{bmatrix}^T$$



No. of input variables	Selected input variable 1	Selected input variable2	Selected input variable3	Selected input variable4	No. of MFs	Type of polynomial
[2,3,4]	[1 *k*]	[1 *k*]	[1 *k*]	[1 *k*]	[2,3]	[1 4]

k: The total number of input variables

MF: Membership function

(a) Arrangement of particle for the optimization of each neuron

(b) Example of the structure of FsPN designed by particle arrangement: ① means the number of input variables used to FsPN; ② means the selected input variables (x_4 and x_1); ③ means the number of MFs per input variable; and ④ means polynomial type (2 denotes type 2, that is, linear function)

FIGURE 6. Overall framework for single/multi-optimization of FsPN.

error estimation (LSE)-based learning method with L_2-norm regularization.

L_2-norm regularization commonly added the L_2-norm penalty to the objective function, which is known as ridge regression or Tikhonov regularization. In most learning algorithms, regularization plays an important role in order to improve the performance of the regression model. In the case of ridge regression, the objective function can be regularized by adding a penalty item $(\lambda||a||^2)$ as the regularization of parameters to limit the parameter of the solutions. The simple penalty term takes the form of a sum of the square of all of the coefficients, leading to the objective function in the form presented in [21].

The coefficient λ governs the relative importance of the regularization term $(\lambda||a||^2)$ compared with the term, the sum of squared error. In order to alleviate the over-fitting problem in the conventional FPNN, L_2-norm regularization is considered in the LSE learning method, which is used for the minimum coefficient estimation by adding a penalty term to the cost function. It's a kind of representative method of reducing the influence of noise by flattening the solution space and also lessening the coefficient size. In the regularization approach, a parameter $(\lambda||a||^2)$ is included and the reasonable value is implemented in the associated cost function. The parameter of (10), A_i could not be obtained if it is a singular matrix or almost a singular matrix, $X_i^T X_i$. Therefore, the L_2-norm regularization technique adds a penalty item $(\lambda \sum_{p=1}^{n} (a_p)^2$ in (5)) for the additional diagonal elements (λI) of the matrix to make the singular matrix invertible as shown in $(X_i^T X_i + \lambda I)$ of (11), Then the overfitting problem would be alleviated.

Multi-collinearity yields a high-variance model according to the increase of correlation values. L_2-norm regularization by dealing with the numerical instability of the matrix inversion leads to lower variance and a better prediction model. In the sequel, the combination of both this method and exponential-based roulette selection technique enables the proposed multi-layered network structure to do the decrease overfitting as well as multi-collinearity problem.

III. MULTI-OPTIMIZATION OF OVERALL FRAMEWORK OF ENHANCED ENSEMBLE FUZZY SET-BASED POLYNOMIAL NEURAL NETWORK

The HRmFNN architecture is not fixed in advance as a self-organizing network but becomes organized through the growth process of the generation of the layers and nodes (neurons) of the network. Along with the use of synergistic design methodologies such as exponential-based roulette selection technique, LSE-based learning, and two types of parallel network structures as shown in Section 2, the structural and parametric multi-optimization of the network through them is more effectively carried out for constructing the stabilized as well as an enhanced multi-layered network of HRmFNN [45]. In the training process, each layer of the proposed model uses an optimization algorithm, so it takes lot of computing time for training the model. Besides, it is proportional to the number of generations of PSO and the number of layers of the model. For this reason, we have decided to apply particle swarm optimization to the proposed model because PSO is conceptually simple, easy to implement, and computationally efficient. Unlike many other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance global and local exploration abilities.

First, in the design process of HRmFNN, the structure of fuzzy set-based polynomial neuron (FsPN) as a node (neuron) of each layer of the network is optimized. Four kinds of structural parameters of FsPN contain a) the number of input variables, b) a collocation of the specific subset of input variables, c) the number of fuzzy membership functions, and d) the type of polynomial.

Next, after obtaining the optimized FsPN, the optimization design of both nodes selected in the current layer and their ensuing layer (next layer) leads to the optimized HRmFNN structure. The overall optimization process of HRmFNN is schematically displayed in figure 5.

The design procedure for the optimization of HRmFNN comprises the following steps.

A. STEP 1) CONSTRUCT TRAINING DATA, VALIDATION DATA, AND TESTING DATA

Determine input variables, the original data set is divided into three parts: training data, verification data, and test data. The training and validation data are used to design the HRmFNN, and the test data is used to evaluate the performance of the model.

B. STEP 2) SPECIFY INITIAL DESIGN PARAMETERS

There are many detailed decisions to be made about basic design parameters: 1) Use prior domain knowledge that affects the proposed network topology. 2) Select a certain stopping criterion. 3) Enter the maximal number of input variables for each node in the corresponding layer. 4) Determine the total number (N) of nodes entering the next layer from the current layer in the network. 5) Select the network depth of the HRmFNN to reduce possible conflict between over-fitting and generalization capabilities of the network; 6) Determine the depth and width of the network that needs to be selected due to some tradeoff between the accuracy and complexity of the overall model.

C. STEP 3) DESIGN FsPN AS NODE(NEURON) OF EACH LAYER

In the design of FsPN, the membership function is used to construct the premise part while the least square error estimation (LSE)-based learning with L_2-norm regularization is utilized to estimate the parameters of the polynomial used in the consequence part. figure 5 visualizes an example of particle interpretation being used for the optimization of FsPN, that is, the number of input variables, collocation of the specific subset of input variables, the number of membership functions per one variable, and order of polynomial and the values. The values of such parameters of FsPN are determined by PSO. The related parameters to be optimized are shown in figure 5. The values of particles are arbitrarily selected to explain specifically the optimization process of the FsPN. The random values shown in four parts of the particle are rounded up to the first decimal digit. The first particle is assigned for the selection of the number of input variables, whose range is set the real number between 2 and 4; the second particle is assigned for the non-repeated and selected input variables, whose range is the real number from 1 to the maximal number of input variables; the third particle is used to decide the polynomial type, which is selected from 2 to 3; there are four types of a polynomial can be selected, the polynomial type is randomly selected from one to four.

The objective function is divided into two parts; MPI is used as the objective function including training data and validation data and comes as a convex combination of these two components:

$$MPI = \theta \times PI + (1 - \theta) \times VPI \tag{14}$$

where MPI means weighted performance index obtained by considering both training and validation dataset. θ is a weighting factor that helps form a sound balance between training data and validation data. Regarding the choice of θ, we generally consider the following two situations:

1) The value of θ is set to 1. The HRmFNN is optimized only based on the training data, regardless of the validation data.
2) The value of θ is set between 0 and 1, where $\theta \in (0, 1)$. In this case, the training data and the validation data will be considered. The choice of θ establishes some trade-offs between the approximation and prediction capabilities of the HRmFNN.

Assume that an input-output dataset is denoted as $(x_k, y_k) = (x_{1k}, x_{2k}, \ldots, x_{nk}, y_k), k = 1, 2, \ldots, m$, where m is the number of data patterns, then the performance

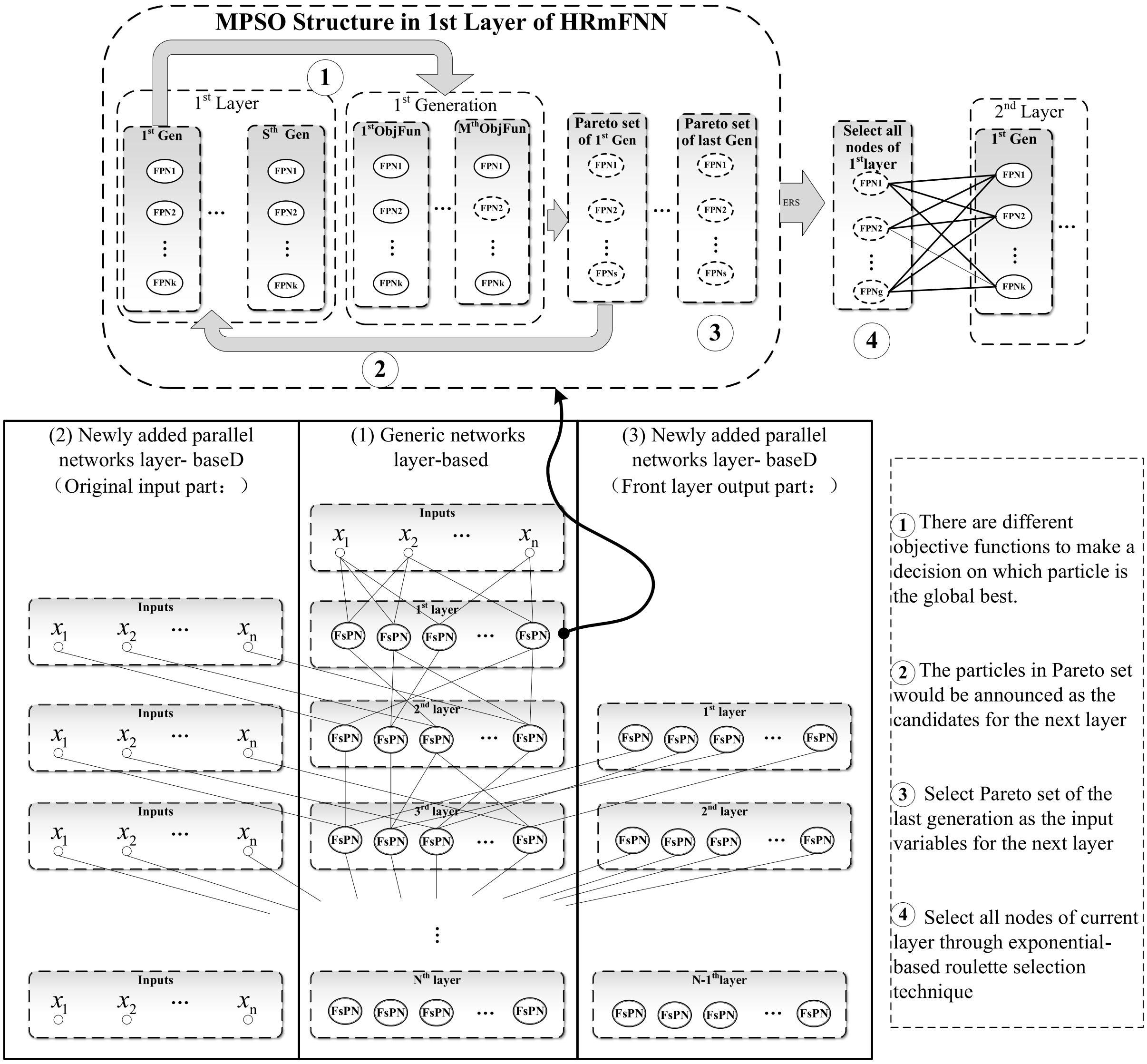

MPSO Structure in 1st Layer of HRmFNN
1
1st Layer
1st Gen
Sth Gen
FPN1
FPN2
FPNk
1st Generation
1stObjFun
MthObjFun
Pareto set of 1st Gen
FPNs
Pareto set of last Gen
2
3
ERS
Select all nodes of 1stlayer
FPNg
4
2nd Layer
1st Gen
(2) Newly added parallel networks layer- baseD （Original input part：）
(1) Generic networks layer-based
(3) Newly added parallel networks layer- baseD （Front layer output part：）
Inputs
x_1
x_2
x_n
1st layer
2nd layer
3rd layer
Nth layer
N-1thlayer
FsPN
1 There are different objective functions to make a decision on which particle is the global best.
2 The particles in Pareto set would be announced as the candidates for the next layer
3 Select Pareto set of the last generation as the input variables for the next layer
4 Select all nodes of current layer through exponential-based roulette selection technique

FIGURE 7. Overall optimization procedure of HRmFNN designed with the use of MPSO.

index (PI) is used as the performance index of a model as follows:

$$PI(VPI, EPI) = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(y_k - \hat{y}_k)^2}. \tag{15}$$

where $\hat{y}$ is the output of the model, and m is the overall number of data, *VPI* and *EPI* stand for performance index for validation and testing datasets.

Generally, most optimization problems encountered in real-world problems involve more than a single objective. In the multi-optimization part, three objective functions including MPI are used to evaluate the accuracy, complexity, and interpretability of HRmFNN. The objective functions are the MPI as PI, the entropy of partition, and the sum of squared coefficients in HRmFNN to be estimated [33].

1) The PI is the accuracy criterion of the HRmFNN like (13)

2) As a measure for evaluating the coefficient limitation of the fuzzy model we consider the sum of squared coefficients of HRmFNN, which is expressed as follows:

$$SSC = \sum_{j=1}^{m} a_j^2 \tag{16}$$

3) As a measure for evaluating the structural complexity of a model, we consider the entropy of the partition [17]. The entropy of partition reflects a degree of overlap between the regions of fuzzy relations for every fuzzy rule. Considering the training dataset, the entropy of the fuzzy partition reads as

$$H = -\sum_{j=1}^{n}\sum_{k=1}^{m}\mu_{jk}\log(\mu_{jk}) \tag{17}$$

where μ_{jk} is the fuzzy membership value of the *j*-th rule, *k* is the number of data.

To optimize the structure selection of FsPN, we realize the following steps

1) Select the number of input variables and collocation of the specific subset of input variables based on multi-PSO.

2) Determine the number of membership functions, and the order of polynomial based on the selected input variables with the aid of multi-PSO.

3) Calculate the output of the FsPN by LSE-based learning with L_2-norm regularization.

D. STEP 4) SELECT NODES(FsPN) AND CONSTRUCT THEIR CORRESPONDING LAYER

Select nodes by PSO and construct HRmFNN by training data and validation data, as shown in figure 6. Overall optimization procedure of HRmFNN designed with the use of PSO is carried out by MPSO-optimized HRmFNN (HRmFNN).

Through the exponential-based roulette selection, technique node selection is conducted by the multi-objective optimization from each layer combined with both the generic network and added parallel network. As shown in Figure 7, we find the Pareto optimal sets A by minimizing the objective functions related to {MPI, SSC, H}. The Pareto optimal set A is used in order to select the input neurons entering the next layer.

E. STEP 5) MODEL EVALUATION

Use testing data to calculate the performance index (EPI) of each FsPN, M nodes (neurons) with the best performance (EPI) are selected from all FsPN of each layer, and one of M nodes becomes the final result.

IV. EXPERIMENTAL STUDIES

In this section, a series of experiments were performed to evaluate the performance of the proposed neural network. We used the benchmark machine learning datasets for statistical analysis as well as performance comparison. In addition, the experiment is carried out through five-fold cross-validation, and the entire dataset is divided into three parts: training data (50%), validation data (30%), and test data (20%). In the experiment, we evaluate the performance index (PI) of HRmFNN with the aid of L_2-norm regularization for each neuron. The objective function of MPSO is given as three types of MPI. SSC, and H. In MPSO, the parameters such as c_1, c_2, w_{min}, w_{max}, v_{max} and the mutation ratio are selected as 2.0, 2.0, 0.4%, 0.6%, 20% and 0.5, respectively. these values are selected by trial and error in the experiment. The parameters used in the experiment are shown in Table 3.

TABLE 3. List of the parameters of HRmFNN and MPSO, and boundaries of search space (FsPN).

Parameter		Value
Number of input variables		[2 4]
Maximal number of selected nodes		30
Number of membership functions		[2 3]
Membership function type		Triangular
Polynomial type		1~4 (Type1 ~ Type 4)
Evaluation Method		RMSE
Regularization parameter (λ)		0.001 0.01 0.1
Data split	Without optimization	4:1
	With optimization	5:3:2
Optimization parameters		
Swarm Size		200
Number of Generations		100
Vmax		20%
Acceleration Coefficient (c_1, c_2)		2.0
Inertia Weight(w_{min}, w_{max})		[0.4 0.6]
Size of Pareto Set		100
Mutation ratio		0.5
Object function		MPI : SSC : Entropy
A boundary of search space of decision variables		
Number of input variables		2~4
Collocation of the specific subset of input variables		1~ Number of input variables
Number of membership functions		2~3
Order of polynomial of each node		1~4 (Type1 ~ Type 4)

A. BOSTON HOUSING(BH) DATASET

This dataset concerns real estate in the Boston area (ftp://ftp.ics.uci.edu/pub/machine-learning databases/ housing/ housing.data). The median price of the house (MEDV) is considered as an output variable. This dataset includes 506 input-output pairs. There are 13 input variables. Tables 4 show the performance index of FPNN and the proposed HRmFNN. The performance value is reported as the mean and its standard deviation, and the bold faces indicate the best performance index based on a testing dataset of each model. The conventional model based on FrPN has the best performance in the second layer because the sharp increase of parameters shows over-fitting from the third layer, and the proposed HRmFNN based on FsPN has poor performance in the first layer, but more stability running to the fourth layer, and get better performance than the conventional model FPNN.

Table 5 shows the performance comparison by the parallel network structure. The parallel network structure has higher performance as well as exhibits higher level of stabilization.

Table 6 shows the performance comparison by 2 types such as PI-based node selection, and exponential-based roulette selection technique in the HRmFNN structure. The HRmFNN using the exponential-based roulette selection technique has higher performance as well as high level of stability.

Figure 8 illustrates the comparison of exponential-based roulette selection technique and PI-based node selection

TABLE 4. Comparative analysis of FPNN and HRmFNN.

	FPNN		Proposed HRmFNN	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1stLayer	3.45±0.14	4.24±0.44	4.36±0.24	4.69±0.61
2nd Layer	2.90±0.22	**3.90±0.60**	3.52±0.33	4.11±0.76
3rd Layer	2.67±0.20	20.80±35.33	3.42±0.36	4.04±0.87
4th Layer	2.47±0.18	106.80±201.15	2.85±0.26	**3.82±0.31**
5th Layer	2.35±0.16	N/A	2.60±0.12	4.29±0.61
6th Layer	2.23±0.14	N/A	2.48±0.10	4.42±0.47
7th Layer	2.16±0.15	N/A	2.39±0.07	4.58±0.88
8th Layer	2.10±0.13	N/A	2.21±0.05	18.26±21.76
9th Layer	2.04±0.11	N/A	2.14±0.07	47.27±68.84
10th Layer	1.99±0.16	N/A	2.13±0.05	N/A

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

TABLE 5. Performance values in each layer between the existing and proposed network structure for BH dataset.

	Existing structure		Proposed structure	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	4.36±0.24	4.69±0.61	4.36±0.24	4.69±0.61
2nd Layer	3.52±0.33	4.11±0.76	3.52±0.26	3.83±0.45
3rd Layer	3.42±0.36	4.04±0.87	2.63±0.20	3.88±0.30
4th Layer	2.85±0.26	**3.82±0.31**	2.53±0.21	4.06±0.34
5th Layer	2.60±0.12	4.29±0.61	2.11±0.17	**3.73±0.35**
6th Layer	2.48±0.10	4.42±0.47	2.02±0.15	3.79±0.23
7th Layer	2.39±0.07	4.58±0.88	1.98±0.15	3.81±0.27
8th Layer	2.21±0.05	18.26±21.76	1.92±0.13	3.97±0.32
9th Layer	2.14±0.07	47.27±68.84	1.85±0.14	4.29±0.28
10th Layer	2.13±0.05	N/A	1.81±0.13	4.41±0.28

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

TABLE 6. Performance comparison of node selection methods (viz. PNS, ERS) for BH dataset.

	Node selection by using PNS		Node selection by using proposed ERS	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	4.36±0.24	4.69±0.61	4.36±0.24	4.69±0.61
2nd Layer	3.52±0.33	4.11±0.76	3.67±0.10	4.24±0.51
3rd Layer	3.42±0.36	4.04±0.87	3.27±0.07	3.88±0.53
4th Layer	2.85±0.26	**3.82±0.31**	2.78±0.06	3.90±0.39
5th Layer	2.60±0.12	4.29±0.61	2.28±0.06	3.80±0.32
6th Layer	2.48±0.10	4.42±0.47	2.16±0.07	3.81±0.37
7th Layer	2.39±0.07	4.58±0.88	2.14±0.06	**3.67±0.57**
8th Layer	2.21±0.05	18.26±21.76	2.08±0.07	4.12±0.54
9th Layer	2.14±0.07	47.27±68.84	2.02±0.08	3.84±0.53
10th Layer	2.13±0.05	N/A	2.01±0.07	3.95±0.51

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation), PNS: PI based node selection, ERS: Exponential-based roulette selection

TABLE 7. Values of Performance Index by the change of Lambda for BH dataset.

Layers	*Lambda=0*		*Lambda=0.001*		*Lambda=0.01*		*Lambda=0.1*	
	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	4.36 ±0.24	4.69 ±0.61	4.49 ±0.26	4.12 ±0.87	4.43 ±0.39	4.16 ±0.43	4.52 ±0.43	4.06 ±0.44
2nd Layer	3.52 ±0.33	4.11 ±0.76	3.54 ±0.17	3.84 ±0.74	4.18 ±0.32	4.08 ±0.51	4.12 ±0.31	4.10 ±0.63
3rd Layer	3.42 ±0.36	4.04 ±0.87	3.19 ±0.21	3.81 ±0.71	3.88 ±0.26	4.07 ±0.53	3.84 ±0.33	3.88 ±0.58
4th Layer	2.85 ±0.26	**3.82 ±0.31**	3.14 ±0.15	3.67 ±0.57	3.30 ±0.26	3.95 ±0.39	3.62 ±0.25	4.01 ±0.52
5th Layer	2.60 ±0.12	4.29 ±0.61	2.86 ±0.20	3.56 ±0.53	3.27 ±0.26	3.77 ±0.32	3.58 ±0.16	3.94 ±0.61
6th Layer	2.48 ±0.10	4.42 ±0.47	2.79 ±0.18	3.47 ±0.57	3.20 ±0.29	3.63 ±0.37	3.29 ±0.14	3.90 ±0.50
7th Layer	2.39 ±0.07	4.58 ±0.88	2.51 ±0.16	**3.47 ±0.51**	3.07 ±0.18	3.59 ±0.36	2.97 ±0.18	3.89 ±0.37
8th Layer	2.21 ±0.05	18.26 ±21.76	2.40 ±0.12	3.55 ±0.57	2.96 ±0.17	3.47 ±0.39	2.85 ±0.22	3.90 ±0.53
9th Layer	2.14 ±0.07	47.27 ±68.84	2.34 ±0.10	3.63 ±0.49	2.57 ±0.19	3.46 ±0.39	2.79 ±0.18	3.90 ±0.48
10th Layer	2.13 ±0.05	N/A	2.25 ±0.09	3.58 ±0.48	2.46 ±0.16	**3.45 ±0.36**	2.61 ±0.18	**3.80 ±0.55**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

methods in the proposed HRmFNN. figure 8(a) and 8(c) respectively represent the optimal topology generated by PI-based node selection and the topology generated by exponential-based roulette selection under the same conditions in HRmFNN. The number in the node (neuron) represents the index of the node that builds the current layer. It is obvious from figure 8(b) that 26 identical nodes are selected by the two node selection methods, and the remaining 15 nodes are different (because the maximum number of nodes in each layer is 30). Although the current structure constructed by ERS is not yet optimal (7-layer topology), the performance of the network composed of its selected nodes is better than the network constructed by PNS in the 7th layer.

As shown in Table 7 and experimental results, $\lambda = 0$ leads to the high possibility of overfitting for testing dataset resulting from higher values of polynomial coefficients of the fuzzy rules, while through the estimation of polynomial coefficients by the change of lambda(λ), the possibility of model overfitting gets more lessened and its ensuing results lead to the stabilization(viz. the alleviation of overfitting for testing dataset) as well as the higher performance of the model according to the change of lambda.

Table 8 shows the performance index of optimized HRmFNN, the performance is reported in terms of its mean and the standard deviation. Boldface entries denote the best performance of each model. When increasing the number of layers, the performance obtained on the testing dataset is improved without overfitting. The performance of the proposed HRmFNN using MPSO is much better than that of FPNN.

Figure 9 illustrates the details of the optimized topology of the 10th layer of HRmFNN with MPSO, it can be seen that the node with the best performance is generated by parallel structure with newly added layers (the 29th and 31st nodes of the 8th layer are selected).

As shown in figure 10, the output performances of each layer through layer generation are depicted from the viewpoint of an unstabilized network caused by overfitting between the conventional models and the proposed models. Especially, figure 10 shows HRmFNN based on multi-optimization continues to maintain the stabilized multi-layered network structure during the growth(generation) process of layers. In the case of using multi-optimization, the optimization procedure is carried out based on 3-dimensional

TABLE 8. Summary of results of the optimized HRmFNN(BH).

	Without MPSO		With MPSO		
Layers	**PI (STD)**	**EPI (STD)**	**PI (STD)**	**VPI (STD)**	**EPI (STD)**
1st Layer	4.36±0.24	4.69±0.61	3.29±0.10	3.80±0.38	3.82±0.35
2nd Layer	3.52±0.33	4.11±0.76	2.97±0.09	3.53±0.37	3.78±0.46
3rd Layer	3.42±0.36	4.04±0.87	2.85±0.08	3.47±0.45	3.48±0.46
4th Layer	2.85±0.26	**3.82±0.31**	2.79±0.06	3.35±0.36	3.39±0.44
5th Layer	2.60±0.12	4.29±0.61	2.61±0.07	3.22±0.24	3.38±0.40
6th Layer	2.48±0.10	4.42±0.47	2.32±0.11	3.10±0.27	3.28±0.48
7th Layer	2.39±0.07	4.58±0.88	2.29±0.10	3.23±0.24	3.29±0.43
8th Layer	2.21±0.05	18.26±21.7	2.15±0.12	2.94±0.34	3.23±0.42
9th Layer	2.14±0.07	47.27±68.8	2.12±0.011	2.91±0.31	3.10±0.46
10th Layer	2.13±0.05	N/A	2.03±0.17	2.89±0.32	**3.04±0.39**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

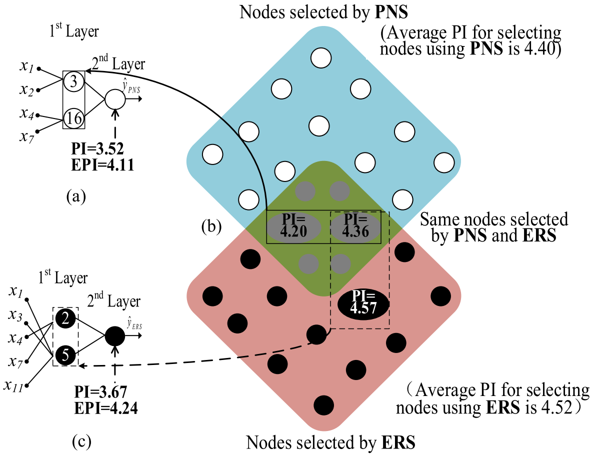


FIGURE 8. Comparison of PNS and ERS selection used in HRmFNN (BH) (a/c) Topology of HRmFNN with PNS/ERS (b) Input nodes of the 2nd layer of HRmFNN selected through PNS/ERS selection.

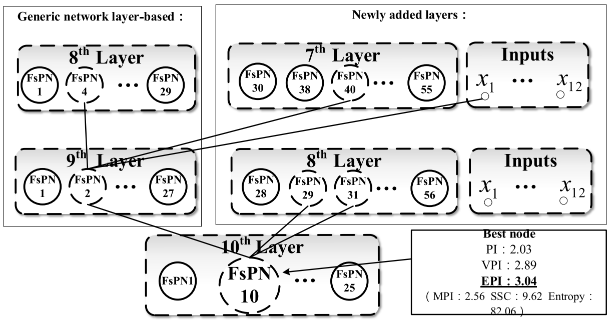


FIGURE 9. Optimized HRmFNN architecture for BH data.

values (MPI, SSC, and H) for enhancing the generalization ability of network layers.

As shown in figure 10, the strictly different point between using optimizations is shown from the viewpoint of the stabilization as well as generalization ability of deeper network structure through the process of layer growth. In the construction of deeper network architecture according to the increase of the number of network layers, the without optimization causes the divergence of output performance by over-fitting problem, while the use of multi-optimization leads to the preferred value of output performance as the stable network structure. As shown in Fig 11 and 12, the characteristics of 3-dimensional values (MPI, SSC, and H) are depicted as the following descriptors: a) EPI: the smaller, the better, b) SSC: the smaller, the better in general, c) Entropy (H): the smaller, the better in general.

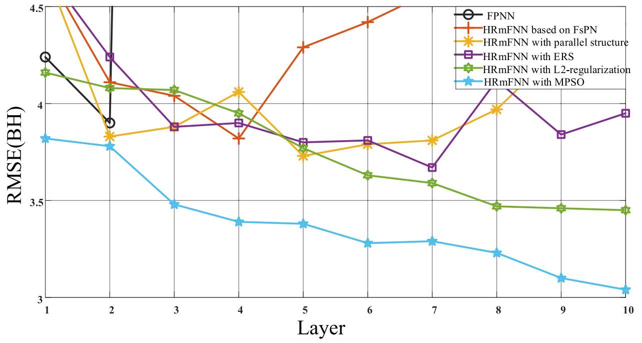


FIGURE 10. Experimental results of comparison of FPNN, HRmFNN based on FsPN, HRmFNN with parallel structure, HRmFNN with ERS selection technique, HRmFNN with L2-norm regularization (lambada = 0.01), and HRmFNN using MPSO for testing dataset(BH).

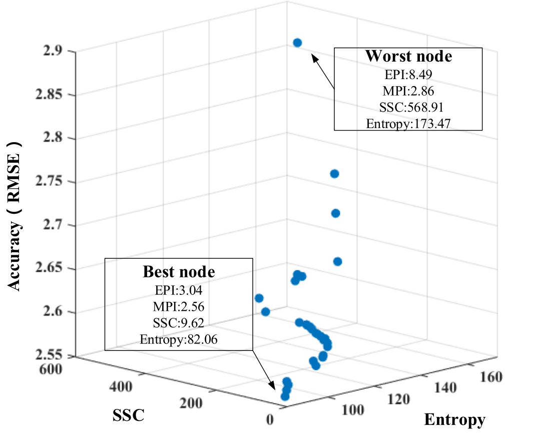
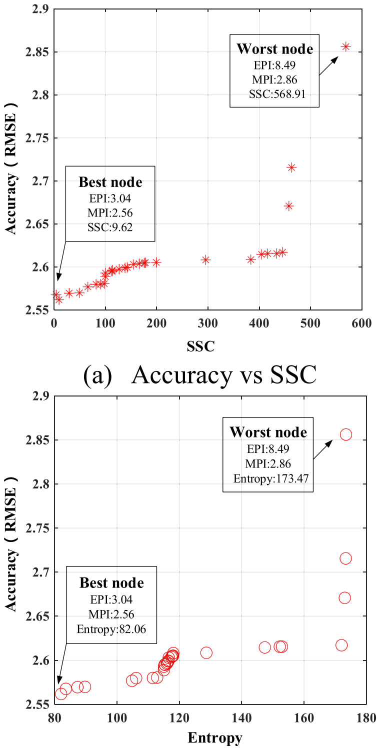


FIGURE 11. 3-D pareto set of HRmFNN based on multi-optimization for BH data.

Figures 11 and 12 depict the Pareto fronts generated using MPSO, we can notice that there is an interesting accuracy-interpretability tradeoff. As the entropy (H) of fuzzy partition and the sum of squared coefficient (SSC) are getting lower, the performance (PI as RMSE) of the model is relatively getting better.

Three different dimensional values o set as shown in figure 11. The symbol "•" (dot) denotes results of objective functions obtained from individual nodes of the 10th (final) layer of the proposed HRmFNN. Here, the number of nodes in the 10th (final) layer of the network is equal to 30. Figure 12 shows the results of *accuracy* versus *SSC* and *accuracy* versus *entropy* in 2-dimensional space, respectively.

Table 9 offers a comparative summary of the proposed model when being contrasted with other models. The approximation and generalization abilities of the proposed model are largely improved in comparison with the abilities of other models. Especially among FPNN, and HRmFNN from the viewpoint of deep-layer structure as a multi-layered network, HRmFNN is depicted as a much more stabilized network structure as shown in figure10.



(a) Accuracy vs SSC

(b) Accuracy vs Entropy

FIGURE 12. 2D pareto set of HRmFNN based on multi-optimization for BH data.

TABLE 9. Results of comparative analysis (BH).

Model		No. of rules	PI RMSE	EPI RMSE
RBFNN [31]		25	6.36 ±0.24	6.94 ±0.31
Linguistic model[31]	Without optimization	25	5.21 ±0.12	6.14 ±0.28
	One-loop optimization	25	4.80 ± 0.52	5.22 ± 0.58
	Multi-step optimization	25	4.12 ± 0.35	5.32 ± 0.96
RBFNN with FCM[31]	With Euclidian distance	19	7.64 ±0.23	7.61 ± 0.38
	With weighted Euclidian distance	22	6.77 ±0.38	6.58 ±0.44
DFCCNN [35]	FCNN with LSE	3	3.325 ± 0.05	3.71 ± 0.50
	FCNN with WLSE	7	3.17 ±0.05	3.46 ±0.63
	DFCCNN	12	2.69 ±0.06	3.20 ±0.66
Conventional model	FPNN (1st layer)	8	3.45 ± 0.14	3.90 ±0.60
Proposed model	HRmFNN (10th layer)	4	2.03 ±0.17	**3.04 ±0.39**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

B. AUTOMOBILE MILES PER GALLON (MPG) DATASET

We consider the automobile MPG datasets (http://archive.ics.uci.edu/ml/datasets/autompg). The output is the automobile's fuel consumption expressed in miles per gallon. This dataset includes 392 input-output pairs (after the removal of incomplete data points). There are seven input variables such as cylinder, displacement, horsepower, weight, acceleration, model year, and origin. Table 10 shows the performance index of the FPNN and the proposed HRmFNN. The performance value is reported as the mean and its standard deviation, and the bold faces indicate the best performance index based on a testing dataset of each model. We compare the output of the proposed HRmFNN with this of the FPNN, and the performance is depicted for the validation data set and testing dataset during the generation of a layer. EPI is better in HRmFNN from the viewpoint of the generation as well as overfitting of multi-layered network structure.

TABLE 10. Comparative analysis of FPNN and HRmFNN.

	FPNN		HRmFNN(Proposed)	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	2.23±0.09	3.14±0.49	2.43±0.25	2.94±0.40
2nd Layer	2.11±0.15	3.05±0.35	2.14±0.29	2.89±0.22
3rd Layer	1.91±0.14	**3.03±0.33**	1.99±0.25	2.84±0.50
4th Layer	1.86±0.11	3.26±0.39	2.05±0.12	2.80±0.38
5th Layer	1.85±0.11	3.33±0.39	2.01±0.13	**2.78±0.46**
6th Layer	1.83±0.10	3.62±0.39	1.95±0.13	2.98±0.45
7th Layer	1.68±0.09	4.49±3.10	1.85±0.13	2.95±0.40
8th Layer	1.63±0.10	7.13±8.58	1.84±0.13	2.95±0.20
9th Layer	1.63±0.10	16.49±24.10	1.73±0.16	2.99±0.20
10th Layer	1.63±0.09	108.13±85.58	1.68±0.17	2.94±0.29

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

TABLE 11. Performance values in each layer between the existing and proposed network structure for MPG dataset.

	Existing structure		Proposed structure	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	2.43±0.25	2.94±0.40	2.43±0.25	2.94±0.40
2nd Layer	2.14±0.29	2.89±0.22	2.06±0.12	2.90±0.32
3rd Layer	1.99±0.25	2.84±0.50	1.63±0.06	2.85±0.48
4th Layer	2.05±0.12	2.80±0.38	1.61±0.07	2.84±0.48
5th Layer	2.01±0.13	**2.78±0.46**	1.58±0.08	2.82±0.48
6th Layer	1.95±0.13	2.98±0.45	1.78±0.10	2.87±0.41
7th Layer	1.85±0.13	2.95±0.40	1.77±0.10	2.79±0.40
8th Layer	1.84±0.13	2.95±0.20	1.72±0.09	2.78±0.25
9th Layer	1.73±0.16	2.99±0.20	1.51±0.06	2.78±0.53
10th Layer	1.68±0.17	2.94±0.29	1.50±0.06	**2.75±0.40**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

Table 11 shows the performance comparison by the parallel network structure. The parallel network structure has higher performance and stabilizes better.

Table 12 shows the performance comparison by 2 types such as PI-based node selection, and exponential-based roulette selection technique in the HRmFNN structure. The HRmFNN using the exponential-based roulette selection technique has higher performance as well as more stabilization.

figure 13 illustrates the comparison of exponential-based roulette selection technique and PI-based node selection

TABLE 12. Performance comparison of node selection methods (viz. PNS, ERS) for MPG dataset.

	Node selection by using PNS		Node selection by using proposed ERS	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	2.43±0.25	2.94±0.40	2.43±0.25	2.94±0.40
2nd Layer	2.14±0.29	2.89±0.22	2.38±0.20	2.92±0.57
3rd Layer	1.99±0.25	2.84±0.50	2.22±0.20	2.75±0.42
4th Layer	2.05±0.12	2.80±0.38	1.74±0.21	2.83±0.48
5th Layer	2.01±0.13	**2.78±0.46**	1.73±0.22	2.75±0.48
6th Layer	1.95±0.13	2.98±0.45	1.71±0.21	**2.70±0.48**
7th Layer	1.85±0.13	2.95±0.40	1.65±0.20	2.85±0.40
8th Layer	1.84±0.13	2.95±0.20	1.64±0.22	2.87±0.41
9th Layer	1.73±0.16	2.99±0.20	1.64±0.23	2.87±0.46
10th Layer	1.68±0.17	2.94±0.29	1.64±0.22	2.87±0.53

PI(STD): Training_RMSE(standard deviation), EPI(STD): Training_RMSE(standard deviation), PNS: PI based node selection, ERS: Exponential-based roulette selection

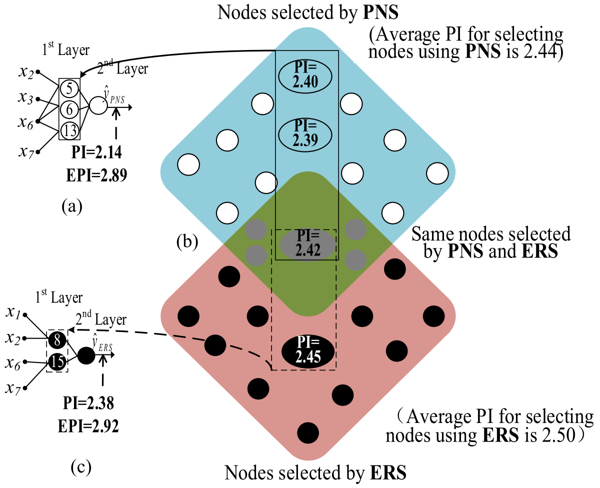

Nodes selected by **PNS**
(Average PI for selecting nodes using **PNS** is 2.44)
1st Layer
2nd Layer
x_2
x_3
x_6
x_7
5
6
13
$\hat{y}_{PNS}$
PI=2.14
EPI=2.89
(a)
PI=2.40
PI=2.39
PI=2.42
PI=2.45
(b)
Same nodes selected by **PNS** and **ERS**
1st Layer
2nd Layer
x_1
x_2
x_6
x_7
8
15
$\hat{y}_{ERS}$
PI=2.38
EPI=2.92
(c)
(Average PI for selecting nodes using **ERS** is 2.50)
Nodes selected by **ERS**

FIGURE 13. Comparison of PNS and ERS selection used in HRmFNN on MPG dataset: (a/c) Topology of HRmFNN with PNS/ERS (b) Input nodes of the 2nd layer of HRmFNN selected through PNS/ERS selection.

methods in the proposed HRmFNN. figure 13(a) and 13(c) respectively represent the optimal topology generated by PI-based node selection and the topology generated by exponential-based roulette selection under the same conditions in HRmFNN. The number in the node (neuron) represents the index of the node that builds the current layer. It is obvious from figure 13(b) that 30 identical nodes are selected by the two different kinds of node selection methods, and the remaining 18 nodes are different (because the maximum number of nodes in each layer is 30). Although the current structure constructed by ERS is not yet optimal (6-layer topology), the performance of the network composed of its selected nodes is better than the network constructed by PNS in the 6th layer.

As shown in Table 13, λ = 0 leads to the high possibility of overfitting for testing dataset from bigger values of polynomial coefficients of the fuzzy rules, while through the estimation of polynomial coefficients by the change of lambda(λ), the possibility of model overfitting gets more lessened and its ensuing results lead to the stabilization(viz. the alleviation of overfitting for testing dataset) as well as the higher performance of the model according to the change of lambda.

TABLE 13. Values of Performance Index by the change of Lambda for MPG dataset.

	Lambda=0		*Lambda=0.001*		*Lambda=0.01*		*Lambda=0.1*	
Layers	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)	PI (STD)	EPI (STD)
1st Layer	2.43 ±0.25	2.94 ±0.40	2.50 ±0.27	2.93 ±0.74	2.55 ±0.13	2.87 ±0.32	2.41 ±0.11	2.92 ±0.37
2nd Layer	2.14 ±0.29	2.89 ±0.22	2.33 ±0.22	2.84 ±0.73	2.44 ±0.13	2.74 ±0.24	2.11 ±0.10	2.93 ±0.24
3rd Layer	1.99 ±0.25	2.84 ±0.50	2.19 ±0.26	2.97 ±0.54	2.39 ±0.12	2.73 ±0.27	2.10 ±0.08	2.91 ±0.22
4th Layer	2.05 ±0.12	2.80 ±0.38	2.17 ±0.26	2.81 ±0.57	2.19 ±0.12	2.73 ±0.25	2.11 ±0.05	2.89 ±0.16
5th Layer	2.01 ±0.13	**2.78** **±0.46**	2.17 ±0.26	2.79 ±0.48	2.15 ±0.11	2.73 ±0.25	2.28 ±0.04	2.87 ±0.17
6th Layer	1.95 ±0.13	2.98 ±0.45	1.94 ±0.17	2.79 ±0.15	2.08 ±0.11	2.73 ±0.25	2.14 ±0.04	2.87 ±0.17
7th Layer	1.85 ±0.13	2.95 ±0.40	1.83 ±0.11	2.79 ±0.46	2.06 ±0.10	2.72 ±0.25	2.12 ±0.04	2.87 ±0.17
8th Layer	1.84 ±0.13	2.95 ±0.20	1.84 ±0.11	2.77 ±0.37	2.05 ±0.10	2.68 ±0.42	2.12 ±0.04	2.85 ±0.16
9th Layer	1.73 ±0.16	2.99 ±0.20	1.71 ±0.09	2.76 ±0.41	2.04 ±0.10	2.67 ±0.41	2.12 ±0.03	2.83 ±0.18
10th Layer	1.68 ±0.17	2.94 ±0.29	1.70 ±0.09	**2.76** **±0.36**	2.04 ±0.09	**2.66** **±0.41**	2.12 ±0.03	**2.83** **±0.17**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

TABLE 14. Summary of results of the optimized HRmFNN (MPG).

	Without MPSO		With MPSO		
Layers	PI (STD)	EPI (STD)	PI (STD)	VPI (STD)	EPI (STD)
1st Layer	2.43±0.25	2.94±0.40	2.16±0.10	2.76±0.23	2.80±0.27
2nd Layer	2.14±0.29	2.89±0.22	2.06±0.09	2.66±0.41	2.75±0.22
3rd Layer	1.99±0.25	2.84±0.50	1.74±0.08	2.64±0.30	2.74±0.25
4th Layer	2.05±0.12	2.80±0.38	1.73±0.07	2.65±0.31	2.65±0.26
5th Layer	2.01±0.13	**2.78±0.46**	1.71±0.05	2.62±0.36	2.62±0.26
6th Layer	1.95±0.13	2.98±0.45	1.30±0.05	2.58±0.30	2.62±0.26
7th Layer	1.85±0.13	2.95±0.40	1.29±0.05	2.55±0.38	2.61±0.21
8th Layer	1.84±0.13	2.95±0.20	1.24±0.04	2.55±0.37	2.61±0.20
9th Layer	1.73±0.16	2.99±0.20	1.23±0.04	2.55±0.38	2.61±0.20
10th Layer	1.68±0.17	2.94±0.29	1.21±0.05	2.54±0.42	**2.61±0.20**

PI(STD): Training_RMSE (standard deviation), EPI(STD): Training_RMSE (standard deviation)

Table 14 shows the performance index of optimized HRmFNN, the performance is reported in terms of its mean and the standard deviation. Boldface entries denote the best performance of each model. When increasing the number of layers, the performance obtained on the testing dataset is improved without overfitting. The performance of the proposed HRmFNN using MPSO is much better than that of FPNN.

Figure 14 shows a series of performance values for the testing dataset according to the increase in the number of layers. In the case of the conventional FPNN, HRmFNN based on FsPN, HRmFNN with ERS node selection technique they cause overfitting in the third layer and higher. Although there does exist any overfitting tendency in the HRmFNN with parallel structure, the performance seems to be unstable during the growth of the layer. On the other hand, The HRmFNN based on MPSO keeps stable performance without overfitting until the 10th layer.

Figures 15 and 16 depict the Pareto fronts generated using MPSO, we can notice that there is an interesting accuracy-interpretability tradeoff. As the entropy (H) of fuzzy partition

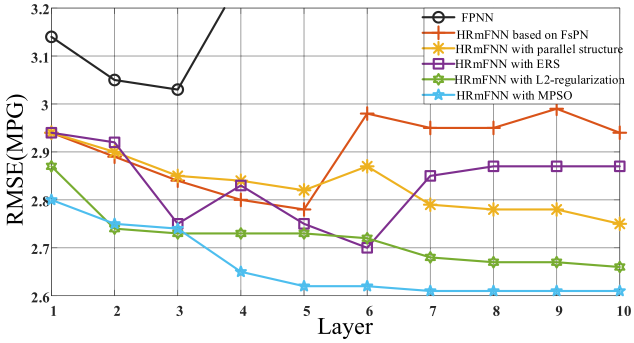


FIGURE 14. Experimental results of comparison of FPNN, HRmFNN based on FsPN, HRmFNN with parallel structure, HRmFNN with ERS selection technique, HRmFNN with L$_2$-norm regularization ($\lambda = 0.01$), and HRmFNN using MPSO for testing dataset(MPG).

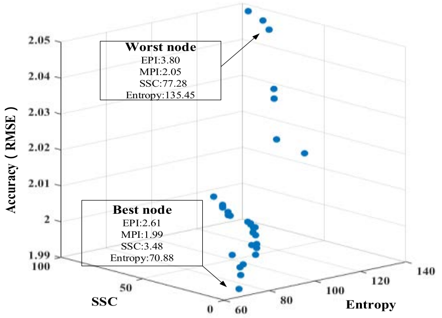
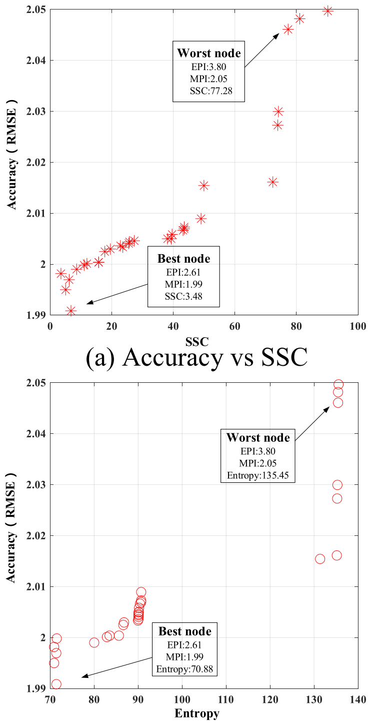


FIGURE 15. 3-D pareto set of HRmFNN based on multi-optimization for MPG data.



(a) Accuracy vs SSC

(b) Accuracy vs Entropy

FIGURE 16. 2D pareto set of HRmFNN based on multi-optimization for MPG data.

and the sum of squared coefficient (SSC) are getting lower, the performance (PI as RMSE) of the model is relatively getting better.

TABLE 15. Results of comparative analysis (MPG).

Model		No. of rules	PI RMSE	EPI RMSE
RBFNN [31]		36	3.24±0.24	3.62±0.31
Linguistic model[31]	Without optimization	36	3.78±1.52	4.22±1.22
	One-loop optimization	36	2.90±0.52	3.17±1.01
	Multi-step optimization	36	2.86±0.83	3.14±1.01
RBFNN with FCM[31]	With Euclidian distance	18	4.13±0.14	4.09±0.22
	With weighted Euclidian distance	27	3.71±0.13	3.52±0.31
Reinforced rule-based fuzzy models [34]		2	2.32±0.09	3.01±0.11
		3	2.60±0.09	3.03±0.13
Fuzzy model [36]	With random basis function	-	3.05±0.09	3.18±0.46
DFCCNN [35]	FCNN with LSE	2	2.78±0.05	2.85±0.49
	FCNN with WLSE	4	2.74±0.07	2.86±0.61
	DFCCNN	2	2.66±0.05	2.74±0.48
Convention al model	FrPNN (3th layer)	8	1.91±0.14	3.03±0.33
Proposed model	HRmFNN (10th layer)	6	1.21±0.05	2.61±0.20

TABLE 16. Summary of 14 machine learning datasets.

No.	Data type	Abbreviation	No. of attributes	No. of instances
1	Boston housing	BH	13	506
2	Concrete compressive strength	CCS	8	1030
3	Energy efficiency	EE	9	768
4	Forestfires	FF	10	517
5	Friedman_1	F1	10	250
6	MIS	MIS	11	390
7	Nox	Nox	5	260
8	Treasury	TR	15	1049
9	Compactiv	CV	21	8192
10	Friedman_4	F4	100	250
11	Puma	Pu	32	8192
12	Winequality-red	WR	11	1599
13	Mortgage	MO	15	1049
14	Stock	St	9	950

Table 15 offers a comparative summary of the proposed model when being contrasted with other models reported in previous literature.

C. OTHER DATASETS

To evaluate the performance of the proposed models and the effect of the composite kernel function, different algorithms on 14 well-known benchmark datasets are compared. These datasets are obtained from the University of California at Irvine (UCI) Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html). Table 16 presents a summary of the datasets.

Table 17 shows the performance index of the conventional FPNN and the proposed HRmFNN. Testing results are better in HRmFNN from the complexity as well as the performance of the model. To further analyze whether the proposed model

TABLE 17. Comparative analysis of FPNN and HRmFNN.

Data	Conventional FPNN	Proposed HRmFNN
BH	3.90	3.04
CCS	7.4	6.80
EE	2.2	1.14
FF	52.50	50.40
F1	0.91	0.64
MIS	7.30	7.03
Nox	2.33	1.08
TR	0.46	0.31
CV	3.80	2.98
F4	0.96	0.68
Pu	0.02	0.02
WR	0.69	0.65
MO	0.11	0.05
St	1.02	0.98

TABLE 18. Results of comparative analysis between the proposed HRmFNN and other models.

Data	Weka					Proposed
	A	B	C	D	E	HRmFNN
BH	4.88(3)	4.99(5)	4.89(4)	4.85(2)	5.30(6)	**3.04 (1)**
CCS	10.48 (4)	8.16 (3)	10.57 (5)	10.81 (6)	7.08 (2)	**6.80 (1)**
EE	1.96 (4)	2.67 (6)	2.02 (5)	1.95 (3)	1.72 (2)	**1.14 (1)**
FF	52.58 (3)	55.13 (4)	51.87 (2)	66.87 (5)	100.86 (6)	**50.40 (1)**
F1	0.84 (2)	0.88 (3.5)	0.88 (3.5)	1.04 (5)	1.06 (6)	**0.64 (1)**
MIS	**6.50 (1)**	8.84 (5)	7.91 (3)	8.25 (4)	10.24 (6)	7.03 (2)
Nox	4.37 (4)	6.48 (5)	12.69 (6)	2.77 (3)	2.00 (2)	**1.08 (1)**
TR	**0.24 (1)**	0.77 (6)	0.52 (5)	0.29 (2)	0.39 (4)	0.31 (3)
CV	9.70 (4)	13.52 (6)	11.43 (5)	3.31 (2)	4.14 (3)	**2.98 (1)**
F4	1.13 (3)	1.14 (4)	1.11 (2)	1.62 (6)	1.47 (5)	**0.68 (1)**
Pu	0.03 (3.5)	0.03 (3.5)	0.03 (3.5)	0.04 (6)	0.03 (3.5)	**0.02 (1)**
WR	0.65 (1.5)	0.67 (3)	0.70 (4)	0.73 (5)	0.78 (6)	**0.65 (1.5)**
MO	0.12 (2.5)	0.19 (4)	0.45 (6)	0.12 (2.5)	0.26 (5)	**0.05 (1)**
St	2.34 (4)	4.60 (6)	3.98 (5)	1.73 (3)	1.27 (2)	**0.98 (1)**
Avg. rank	2.89	4.57	4.21	3.89	4.18	**1.25**

A: Linear Regression, B: Additive Regression, C: Gaussian Processes, D: MLP, E: Random Tree

is statistically significantly better than the other comparative models, we use the Bonferroni-Dunn test in Table 18, which fits situations where all models are only compared to the control model and not between themselves [40]. If the corresponding average rank differs by at least the critical difference (CD), the performance of any two models is significantly different. At p = 0.10 (significance level), the CD value is 1.50. Table 18 covers the difference of average rank between the five comparative models (Weka) and the proposed models, as well as the comparison results with CD. Since the difference between the average rank of all comparative models and the proposed HRmFNN with MPSO is greater than CD (2.89-1.25 = 1.64> 1.50, 4.57-1.25 = 3.35 > 1.50, 4.21-1.25 = 2.96 > 1.50, 3.89-1.25 = 2.64 > 1.50, 4.18-1.25 = 2.93 > 1.50), we conclude that the prediction accuracy of HRmFNN is statistically superior to the five comparative models.

D. PRACTICAL APPLICATION FOR ESTIMATE CEMENT COMPRESSIVE STRENGTH (CCS)

In this application, we used the proposed model to estimate the application of cement compressive strength (CCS). Cement is one of the most widely used building materials in the world. Its physical characteristics greatly affect the safety of the building. Among these physical properties, CCS is the most important physical and mechanical property reflecting the quality of cement. The extracted 3D microstructure image data of cement is shown in Figure 17. First, cylindrical cement samples were scanned by μ computed tomography (CT) to obtain their 3D microstructure images [38]. Secondly, by capturing the cubic volume of interest (VOI) to generate an image data set, the influence of the air film outside the sample area can be eliminated. Third, extract 3D microstructure image features. The detailed extraction process is shown in [39] uses gray-level histogram (GLH) and gray-level co-occurrence matrix (GLCM) to describe the characteristics of 3D images. For further statistical analysis, the feature values of GLH and GLCM are calculated as 3D microstructure image features [39]. The CCS data set has 56 input dimensions, and the output is the compressive strength of cement.

TABLE 19. Results of comparative analysis.

	HRmFNN (RMSE)		
Layers	PI(STD)	VPI(STD)	EPI(STD)
1st Layer	3.65±0.04	3.70±0.05	3.68±0.13
2nd Layer	3.34±0.04	3.39±0.08	3.37±0.14
3rd Layer	3.19±0.04	3.28±0.04	3.24±0.12
4th Layer	3.11±0.04	3.13±0.06	3.17±0.19
5th Layer	3.10±0.05	3.12±0.05	3.19±0.11
6th Layer	2.89±0.02	2.97±0.06	2.96±0.11
7th Layer	2.81±0.03	2.93±0.05	2.89±0.11
8th Layer	2.74±0.03	2.83±0.05	2.84±0.10
9th Layer	2.73±0.04	2.85±0.05	2.81±0.10
10th Layer	2.71±0.04	2.79±0.05	**2.75±0.10**

TABLE 20. Results of comparative analysis.

Data	Weka					Proposed HRmFNN
	A	B	C	D	E	
CCS	3.64	4.03	3.38	3.22	3.08	**2.75**

A: Linear Regression, B: Additive Regression, C: Gaussian Processes, D: Multi-Layer Perceptron, E: Random Tree

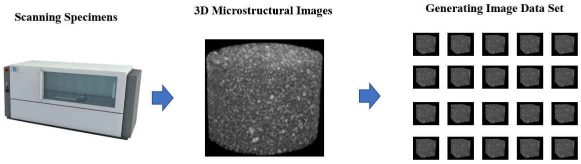


FIGURE 17. 3D microstructure image data obtained by μCT scanning cylindrical cement samples.

Table 19 shows the performance index of HRmFNN with MPSO, and the comparison of performance is summarized in Table 20. After many experiments, the proposed HRmFNN has better performance when comparing other models.

V. CONCLUSION

In this study, the hierarchically reorganized multi-layer fuzzy neural networks (HRmFNN) architecture is developed as a multi-layered self-organizing network constructed through the hierarchical as well as newly added layer generation process of layers of the HRmFNN. Besides the proposed multi-layered self-organizing network structure is realized with the aid of several synergistic multi-techniques such as multi-PSO, additional parallel structure design, LSE-based learning with L_2 norm regularization, and exponential-based roulette selection technique. In the deep multi-layered network structure, multi-PSO based design combined with some synergistic techniques mentioned previously could effectively produce the preferred as well as stabilized network structure through the growth process of layers by considering data characteristic, data dimensionality, and size between input variables, and others.

When the dataset is complex and more sensitive to over-fitting, multi-PSO design based on the previously synergistic techniques could enhance the diversity as well as the selection of the layers- and nodes- structure in the multi-layered Network to a certain extent, and their ensuing results lead to the deeply stabilized multi-layered network architecture through alleviation of the over-fitting phenomenon caused by multi-collinearity. In the series of experimental studies, the proposed HRmFNN is much more effectively stabilized than the conventional models from the viewpoint of deeply layered structure as well as performance.

Possible future studies might focus on exploring structural design methodologies to make the proposed network more stabilized, more performance-improved, and highly multi-layered to cope with classification as well as regression problems.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, Nov. 2015.

[2] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-1, no. 4, pp. 364–378, Oct. 1971.

[3] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Inf. Sci.*, vol. 177, no. 22, pp. 5033–5049, Nov. 2007.

[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. Didarul 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] S.-K. Oh, W. Pedrycz, and B.-J. Park, "Polynomial neural network architecture: Analysis and design," *Comput. Elect. Eng.*, vol. 29, pp. 703–725, Aug. 2003.

[7] S.-K. Oh and W. Pedrycz, "Fuzzy polynomial neuron-based self-organizing neural network," *Int. J. Gen. Syst.*, vol. 32 pp. 237–250, Jun. 2003.

[8] 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.

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

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural Network from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

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

[12] M. G. Wichmann, M. Gäde, and T. S. Spengler, "A fuzzy robustness measure for the scheduling of commissioned product development projects," *Fuzzy Sets Syst.*, vol. 377, pp. 125–149, Dec. 2019.

[13] 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.

[14] S.-K. Oh and W. Pedrycz, "Self-organizing polynomial neural networks based on polynomial and fuzzy polynomial neurons: Analysis and design," *Fuzzy Sets Syst.*, vol. 142, no. 2, pp. 163–198, Mar. 2004.

[15] 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.

[16] 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.

[17] 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.

[18] H. Tamura and T. Kondo, "Heuristics free group method of data handling algorithm of generating optimal partial polynomials with application to air pollution prediction," *Int. J. Syst. Sci.*, vol. 11, no. 9, pp. 1095–1111, Jan. 1980.

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

[20] C. Cortes, M. Mohri, and A. Rostamizadeh, "L_2 regularization for learning kernels," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 109–116.

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

[22] 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.

[23] G. G. Yen and W. Fung Leong, "Dynamic multiple swarms in multiobjective particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst. Humans*, vol. 39, no. 4, pp. 890–911, Jul. 2009.

[24] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Comput.*, vol. 43, no. 11, pp. 1329–1333, 1994.

[25] 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.*, Jul. 2021.

[26] S.-K. Oh, D.-W. Kim, and W. Pedrycz, "Hybrid fuzzy polynomial neural network," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 257–280, Jun. 2002.

[27] H.-S. Park, W. Pedrycz, and S.-K. Oh, "Evolutionary design of hybrid self-organizing fuzzy polynomial neural networks with the aid of information granulation," *Expert Syst. Appl.*, vol. 33, no. 4, pp. 830–846, Nov. 2007.

[28] K.-A. Toh and H.-L. Eng, "Between classification-error approximation and weighted least-squares learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 658–669, Apr. 2008.

[29] S. Yilmaz and Y. Oysal, "Fuzzy wavelet neural network models for prediction and identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1599–1609, Oct. 2010.

[30] Z. Fan, J. Gou, C. Wang, and W. Luo, "Fuzzy model identification based on fuzzy-rule clustering and its application for airfoil noise prediction," *J. Intell. Fuzzy Syst.*, vol. 33, no. 3, pp. 1603–1611, Aug. 2017.

[31] 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.

[32] S.-K. Oh, W.-D. Kim, W. Pedrycz, and S.-C. Joo, "Design of K-means clustering-based polynomial radial basis function neural networks (pRBF NNs) realized with the aid of particle swarm optimization and differential evolution," *Neurocomputing*, vol. 78, no. 1, pp. 121–132, Feb. 2012.

[33] M.-Y. Cheng and M.-T. Cao, "Accurately predicting building energy performance using evolutionary multivariate adaptive regression splines," *Appl. Soft Comput.*, vol. 22, pp. 178–188, Sep. 2014.

[34] A. Ahmad, S. S. Khan, and A. Kumar, "Learning regression problems by using classifiers," *J. Intell. Fuzzy Syst.*, vol. 35, pp. 1–11, Jan. 2018.

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

[36] M.-W. Lee and K.-C. Kwak, "An incremental radial basis function network based on information granules and its application," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–6, Sep. 2016.

[37] W. Huang, S.-K. Oh, and W. Pedrycz, "Fuzzy wavelet polynomial neural networks: Analysis and design," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1329–1341, Oct. 2017.

[38] L. Wang, B. Yang, A. Abraham, L. Qi, X. Zhao, and Z. Chen, "Construction of dynamic three-dimensional microstructure for the hydration of cement using 3D image registration," *Pattern Anal. Appl.*, vol. 17, no. 3, pp. 655–665, Aug. 2014.

[39] L. Zhang, S.-K. Oh, W. Pedrycz, B. Yang, and Y. Han, "Building fuzzy relationships between compressive strength and 3D microstructural image features for cement hydration using Gaussian mixture model-based polynomial radial basis function neural networks," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107766.

[40] 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.

[41] H. Yi and K.-H.-N. Bui, "An automated hyperparameter search-based deep learning model for highway traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5486–5495, Sep. 2021.

[42] N. Calik, M. A. Belen, P. Mahouti, and S. Koziel, "Accurate modeling of frequency selective surfaces using fully-connected regression model with automated architecture determination and parameter selection based on Bayesian optimization," *IEEE Access*, vol. 9, pp. 38396–38410, 2021.

[43] S. Koziel, N. Calik, P. Mahouti, and M. A. Belen, "Accurate modeling of antenna structures by means of domain confinement and pyramidal deep neural networks," *IEEE Trans. Antennas Propag.*, early access, Sep. 15, 2021, doi: 10.1109/TAP.2021.3111299.

[44] O. Owoyele, P. Pal, and A. Vidal Torreira, "An automated machine learning-genetic algorithm framework with active learning for design optimization," *J. Energy Resour. Technol.*, vol. 143, no. 8, Aug. 2021.

[45] S. N. Ghorpade, M. Zennaro, B. S. Chaudhari, R. A. Saeed, H. Alhumyani, and S. Abdel-Khalek, "Enhanced differential crossover and quantum particle swarm optimization for IoT applications," *IEEE Access*, vol. 9, pp. 93831–93846, 2021.

ZHENG WANG received the B.Sc. and M.Sc. degrees in communication engineering from The University of Suwon, Suwon, South Korea, in 2012 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. His research interests include fuzzy set and fuzzy systems, evolution computing, and neural networks.

SUNG-KWUN OH (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 1981, 1983, and 1993, respectively. From 1983 to 1989, he was a Senior Researcher with the Research and Development Laboratory, Lucky-Goldstar Industrial Systems Company Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the School of Electrical and Electronic Engineering, The University of Suwon, Suwon, South Korea. He is also with the Research Center of Big Data and Artificial Intelligence, Linyi University, China. His research interests include fuzzy systems, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He also serves as an Associate Editor for the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic and Intelligent Systems of the KFIS*, and *Information Sciences*.

EUN-HU KIM received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Suwon University, Hwaseong-si, South Korea, in 2009, 2011, and 2016, respectively. From 2017 to 2018, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is currently a Postdoctoral Fellow in electrical engineering with The University of Suwon. He is also with the Research Center of Big Data and Artificial Intelligence, Linyi University, China. His research interests include fuzzy set, neural networks, evolutionary algorithm, computational intelligence, and statistical learning.

ZUNWEI FU (Member, IEEE) received the M.Sc. degree in applied mathematics from the China University of Mining and Technology, Beijing, China, in 2005, and the Ph.D. degree in mathematical science from Beijing Normal University, Beijing, in 2008. Since 2011, he has been a Professor with Linyi University (LU). His current research interests include fractional integral operators and the functional space theory in artificial intelligence. He is a Reviewer of *Mathematical Reviews* (American Mathematical Society). He is also a leading expert in Key Disciplines at the Provincial Level and National Outstanding Teacher. He has served as the Associate Editor for the *Journal of LU* and the Dean of the Research Center of Big Data and Artificial Intelligence, LU.

WITOLD PEDRYCZ (Life Fellow, IEEE) received the M.Sc., Ph.D., and D.Sci. degrees from the Silesian University of Technology, Gliwice, Poland. He is currently a Professor and a Canada Research Chair (CRC) in computational intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland. He has published numerous articles in these areas. He is also the author of 15 research monographs. His research interests include computational intelligence, fuzzy modeling, knowledge discovery, data mining, and fuzzy control, including fuzzy controllers, pattern recognition, knowledge-based neural networks, granular and relational computing, and software engineering. He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He is also the Editor-in-Chief of *Information Sciences* (Elsevier), *WIREs Data Mining and Knowledge Discovery* (Wiley), and the *International Journal of Granular Computing* (Springer). He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS—PART A and the President of IFSA. He also serves on the Advisory Board for the IEEE TRANSACTIONS ON FUZZY SYSTEMS.