Received 10 August 2024, accepted 2 September 2024, date of publication 4 September 2024, date of current version 13 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3454296

RESEARCH ARTICLE

A Novel Fuzzy Hypersphere Neural Network Classifier Using Class Specific Clustering for Robust Pattern Classification

DEEPAK MANE[1], ARUN KULKARNI[2], BISWAJEET PRADHAN[3], (Senior Member, IEEE), SHILPA GITE[4,5], RANJEET VASANT BIDWE[4], (Senior Member, IEEE), CHANG-WOOK LEE[6], (Senior Member, IEEE), AND ABDULLAH ALAMRI[7]

[1]Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India
[2]Thadomal Shahani Engineering College, Mumbai, Maharashtra 400050, India
[3]Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), School of Civil and Environmental Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[4]Symbiosis Institute of Technology, Symbiosis International (Deemed University) (SIU), Pune Campus, Pune, Maharashtra 412115, India
[5]Symbiosis Centre of Applied AI (SCAAI), Symbiosis International (Deemed University), Pune 412115, India
[6]Department of Science Education, Kangwon National University, Chuncheon 24341, South Korea
[7]Department of Geology and Geophysics, College of Science, King Saud University, Riyadh 11421, Saudi Arabia

Corresponding authors: Biswajeet Pradhan (Biswajeet.Pradhan@uts.edu.au) and Chang-Wook Lee (cwlee@kangwon.ac.kr)

This work was supported in part by the Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and Information Technology (IT), University of Technology Sydney; in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant NRF–2023R1A2C1007742; and in part by the Researchers Supporting Project, King Saud University, Riyadh, Saudi Arabia, under Project RSP2024 R14.

ABSTRACT This paper introduces two novel class-specific fuzzy clustering algorithms: Mean-based Supervised Clustering (MSC) and Density-based Mean Supervised Clustering (DMSC). These algorithms are designed to construct the hidden layer of the Fuzzy Hypersphere Neural Network (FHNN) classifier, which is structured on the framework of the Radial Basis Function Neural Network (RBFNN). The FHNN classifier utilizes fuzzy sets as labeled pattern clusters in its hidden layer, with classes represented in the output layer formed by the aggregation of these fuzzy sets. An important characteristic of this classifier is its independence from tuning parameters. It meticulously determines centroids and radii for labeled clusters, consistently achieving 100% accuracy across any training set. The FHNN classifier effectively handles outliers and is robust to variations in data presentation, ensuring clear data visualization for users. During the creation of labeled clusters in the hidden layer, binary weight values are adjusted concurrently between the hidden and output layers. This study proposes the formation of fuzzy clusters with varying dimensions tailored to the dataset. The classifier architecture, rooted in the radial basis function neural network, achieves 100% training accuracy due to precise fuzzy cluster formation. Experimental comparisons with RBFNN and similar classifiers using sixteen benchmark datasets demonstrate the superiority of the proposed classifier in pattern recognition tasks.

INDEX TERMS Fuzzy neural network, fuzzy set hypersphere, pattern classification, radial basis function neural networks, sep fuzzy membership function.

I. INTRODUCTION

Pattern recognition is a highly favored domain among researchers, with Neural Network classifiers gaining popularity over minimum distance classifiers [1]. In the last few decades, Artificial Neural Networks (ANN) have been widely used in pattern recognition. Error back-propagation and RBFNN are basically used for pattern classification. In RBFNN radial basis function is used to create clusters in the hidden layer [2], [3]. Two heuristic clustering

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

algorithms for RBFNN were proposed by Rouhani in the year 2016 [4] while improvements to one of these algorithms were suggested by [5]. Apart from understanding the role of RBFNN, fuzzy neural networks are widely used in pattern recognition. In 1992, there was considerable focus on merging fuzzy logic with neural networks for pattern recognition, leading to the proposal of the fuzzy min–max neural network (FMNN) for classification by [6]. This approach involved creating hyperboxes corresponding to different classes using learning algorithms incorporating expansion and contraction, with hyperboxes described by minimum and maximum points. This concept was further extended by [7] who considered labeled and unlabeled data to create the hyperboxes. In 2004, a new concept of compensatory neuron was introduced [8]. Various new architectures and learning algorithms have been proposed to enhance the performance of FMNN [9], [10], [11], [12].

Various clustering algorithms have been proposed to determine the appropriate centroid and width of the hidden neurons in the hidden layer, alongside Gaussian functions, to enhance performance in terms of recognition rate [4], [13], [14], [15]. Both supervised and unsupervised clustering methods are utilized to ascertain the centers and radii parameters of hidden nodes for Radial Basis Function Neural Networks (RBFNN) [16]. Numerous clustering algorithms incorporate techniques such as K-means clustering [17], subtractive clustering [18], fuzzy clustering [15], [19], [20], [21], ART [22], scatter based clustering [23], input-output clustering [24], artificial fish swarm optimization (AFSO) [25], [26], [27], among others, to illustrate how to create neurons in the hidden layer of RBFNN. Similarly other methods are described in [28], [29], [30], and [31].

The Fuzzy Min-Max Neural Network (FMNN) faces significant disadvantages such as the challenge of altering the hyperbox size, its expansion, contraction, and the overlap test [33]. In contrast, the proposed approach offers several advantages over fuzzy classifiers and Radial Basis Function Neural Networks (RBFNNs) [32]. Firstly, the proposed algorithms, Mean-based Supervised Clustering (MSC) and Density-based Mean Supervised Clustering (DMSC), are utilized to create clusters, known as fuzzy set hyperspheres (FSHs), in the hidden layer without overlap of classes, ensuring 100 % efficiency for the training set. During cluster formation using MSC and DMSC the binary weights are assigned between hidden layer to output layer which is not the case with RBFNN [1]. The training between the hidden layer to output layer which is done by using the LMS algorithm in RBFNN is avoided, and the output is determined by using the fuzzy union operation. Another aspect is that FNN guarantees the training and does not get stuck in local minima which make arise in RBFNN. FMNN describes about the hyperbox creation limited by size stated during initialization. The process of resizing the hyperbox depends on the overlap test. The training process is tedious and takes longer time to form the appropriate number of hyper boxes [6]. RBFNN is a classifier which uses clustering algorithms to create the hidden layer of multilayer network. Different traditional clustering methods where used with constraints like Size of cluster, number of clusters, and cover up of clustered patterns. The training between the hidden layer to output layer which is done by using the LMS algorithm in RBFNN is avoided, and the output is determined by using the fuzzy union operation. Thus FNN reduces the computation time, guarantees 100% accuracy for any training set, and provides superior and comparable recognition accuracy for the datasets with the precise number of FSHs in the hidden layer.

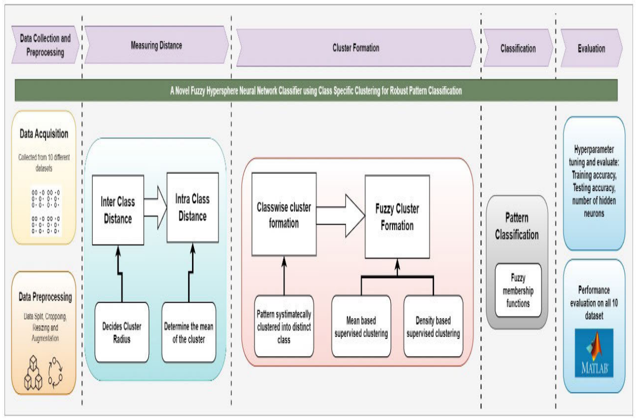


FIGURE 1. Work flow of proposed classifier.

The paper is arranged as follows. The FHNN architecture and two learning algorithms are represented in sections II and III, respectively. Under experimental results, four case studies have been explained in section IV and the conclusion is described in section V.

II. PROPOSED FHNN CLASSIFIER ARCHITECTURE

Here, we proposed the fuzzy cluster formation having different dimensions in accordance with the dataset. The architecture of the classifiers is based on a radial basis function neural network having 100 % training due to precise fuzzy cluster formation. There are no restrictions on fuzzy set hyperspheres (FSHs) size and eliminates the need for overlap tests of clusters, also its robustness extends to any training dataset, leveraging fuzzy membership functions to obscure patterns such that they remain unaffected by neighbouring class patterns which is not the case in traditional clustering methods [34].

Fig. 1 represented the work flow of proposed classifiers. The initial step in the classification process involved determining the inter-class distance between patterns in the dataset, which entailed calculating the distance of one class from the others. This step was essential for deciding the cluster radius. Subsequently, the intra-class distance to the cluster of the same class was computed. Once both inter-class and intra-class distances were determined, clusters were created for each specific class. During this process, the class patterns were systematically clustered until all class patterns were separated into distinct clusters. These clusters were

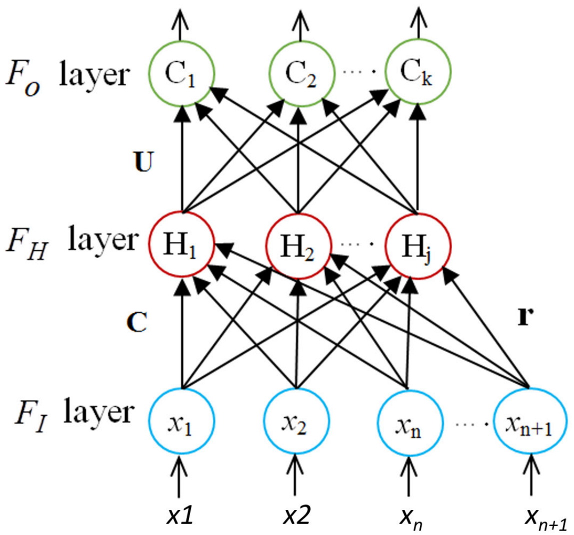


FIGURE 2. Architecture of FHNN.

governed by a fuzzy membership function that determined the output. Ultimately, these clusters formed the hidden layer of the FHSNN for pattern classification. Performance was evaluated using various standard datasets.

This classifier, employing a fuzzy approach, ensures the training of any dataset while providing a clear view of different classes and their outliers. Unlike traditional methods, this classifier does not require parameter adjustments, as it learns autonomously, akin to generative AI [20]. Moreover, this model is well-suited for datasets of any dimensionality and demonstrates improved accuracy for the testing dataset, achieving 100% accuracy for the training dataset.

Fig. 2 represents the proposed FHNN classifier architecture. The input layer is designated as the F_I layer. It consists of $N + 1$ nodes. However, as shown, the N nodes in the F_I layer accept N dimensional pattern. The input to the $(N+1)^{th}$ node is fixed to 1. The nodes in this layer do not perform any processing. Therefore, the transfer function of these nodes is output equal to the input. The next layer is a hidden layer. It is designated as the F_H layer and consists of J nodes. Each node in this layer represents a fuzzy set hypersphere (FSH). There is no need to initialize the parameters. The fuzzy membership function for FSH has been stated in equation 1 and equation 2 and described so as to camouflage the clustered patterns. Two learning algorithms which are described in section III are proposed to construct the FHNN classifier.

In the F_H layer, FSHs are created during learning. The fuzzy membership function, which is defined as:

$$m_j(\mathbf{X}_h, \mathbf{C}_j, r_j) = f(l, r_j) \tag{1}$$

characterizes these FSHs.

where input pattern is represented by $\mathbf{X}_h = (x_{h_1}, x_{h_2}, \ldots, x_{h_N})$, with radius and cenroid of H_j is r_j and $\mathbf{C}_j = (c_{j1}, c_{j2}, \ldots, c_{jN})$.

The function $f()$ is described as:

$$f(len, rad_j) = \begin{cases} 1 & len \leq rad_j \\ rad_j/len & \text{if not} \end{cases} \tag{2}$$

where len is an euclidean distance between $\mathbf{X}_h$ and $\mathbf{C}_j$.

The centroid of FSHs is stored in the matrix $\mathbf{C}$ which are weights between F_I and F_H along with the radii stored in $\mathbf{r}$. The connections between $(N+1)^{th}$ node and FSHs represent respective radii whereas the remaining connections represent centroids. Using MSC or DMSC algorithm the FSH is created in F_H layer with appropriate centroid and radii.

The output layer is designated as the F_O layer. It consists of K nodes which are created during learning representing the class node. The matrix $\mathbf{U}$ contains the binary weights between F_H and F_O.

These weights are assigned as

$$\mathbf{U}(k, j) = \begin{cases} 1 & \text{if } H_j \text{ belongs to class } C_k \\ 0 & \text{if not,} \end{cases} \tag{3}$$

where $k = 1, 2, \ldots.., K$ and $j = 1, 2, \ldots.., J$. The processing of nodes in the F_O layer is defined as

$$C_k = \max_{j=1}^{J} \mathbf{U}(k, j)\, m_j, \quad k = 1, 2, \ldots.., K. \tag{4}$$

The $\mathbf{C}$, $\mathbf{r}$ and $\mathbf{U}$ change during learning as FSH and output class nodes are created

III. LEARNING ALGORITHMS

Two mean-based supervised clustering algorithms have been proposed to construct the FHNN classifier. The detailed description of these two algorithms is given below.

A. MEAN-BASED SUPERVISED CLUSTERING (MSC) ALGORITHM

A training set $\mathbf{Z}$ containing P training pairs is described as $\{\mathbf{X}_j, d_j\}$ for j^{th} input pattern where $d_j \in \{1, 2, \ldots., K\}$ is a desired output for $\mathbf{X}_j$.

Let the k^{th} subset of set $\mathbf{Z}$ contain α_k patterns of class C_k where $k = 1, 2, \ldots, K$. For K classes following steps (1-11) are executed with $t_k = P - \alpha_k$, where t_k is the total number of patterns that do not belong to class C_k.

1) Calculate the inter-class matrix A^k

$$\mathbf{A}^k = \left[\|\mathbf{X}_i - \mathbf{X}_j\|\right]_{\alpha_k \times t_k},$$
$$j = 1, 2, \ldots, t_k \text{ and} i = 1, 2, \ldots, \alpha_k. \tag{5}$$

where $\mathbf{X}_j \notin C_k$, $\mathbf{X}_i \in C_k$.

2) Calculate the intra-class distance matrix B^k as

$$\mathbf{B}^k = \left[\|\mathbf{X}_i - \mathbf{X}_j\|\right]_{\alpha_k \times \alpha_k}, i, j = 1, 2, \ldots, \alpha_k \tag{6}$$

where $\mathbf{X}_i$ and $\mathbf{X}_j \in C_k$.

3) Determine the mean for each pattern using the intra-class distance matrix as

$$\mathtt{M}_i^k = \left(\frac{1}{\alpha_k}\right) * \sum_{j=1}^{\alpha_k} B^k(i,j) \tag{7}$$

in which $i = 1, 2, \ldots, \alpha_k$.

4) Determine the pattern $\mathbf{X}_c^k$ with minimum mean in M^k. Let l_c^k is the location of $\mathbf{X}_c^k$ in the k^{th} subset.

5) Determine minimum inter-class distance d_c^k for this pattern $\mathbf{X}_c^k$ using $\mathbf{A}^k$ as

$$d_c^k = \min_{j=1,2,\ldots,t_k} \mathbf{A}^k(l_c^k, j) \tag{8}$$

6) Create a labeled cluster i.e. FSH with the pattern $\mathbf{X}_c^k$ as a centroid and radius equal to d_c^k in hidden layer of FHNN. Determine the total number of patterns clustered by this FSH using algorithm 1.

7) If $n > 1$ then the revised radius r_c^k is computed as

$$r_c^k = \max_{i=1}^{n} d_i, \tag{9}$$

where Euclidean distance d_i is between i^{th} clustered pattern and centroid $\mathbf{X}_c^k$.
Else If $n = 1$, then to accommodate this outlier, the radius of created FSH is assigned half of d_c^k for good generalization and to minimize the possibility of overlap with clusters belonging to other classes.

8) Create a output node if not created earlier in F_O layer for class C_k and adjust weights linking F_H and F_O layer using (3).

9) Recompute α_k as $\alpha_k \leftarrow \alpha_k - n$, if $\alpha_k \neq 0$. Delete corresponding rows of grouped patterns from $\mathbf{A}^k$ and $\mathbf{B}^k$ and go to step 3. If $k \neq K$ then go to step 1.

10) If $k = K$ then stop.

Algorithm 1 To Determine Patterns Clustered by FSHs

Input: B^k, l_c^k, d_c^k
Output: n
$n \leftarrow 0$
for $j = 1$ **to** α_k **do**
  if $f(\mathbf{B}^k(l_c^k,\ j), d_c^k) <= 1$ **then**
    $n \leftarrow n + 1$
  end if
end for

B. DENSITY-BASED MEAN SUPERVISED CLUSTERING (DMSC) ALGORITHM

Considering the similar data notations from MSC the execution of the following steps is done for K classes.

1) Calculate $\mathbf{A}^k$, $\mathbf{B}^k$ and $\mathbf{M}^k$ as described by steps 1-3 in MSC.

2) Sort mean vector $\mathbf{M}^k$ in ascending order to determine $s = ceiling\ (\sqrt{\alpha_k})$ patterns which have minimum means. Let $\mathbf{X}_1^k, \mathbf{X}_2^k, \ldots\ldots\ldots, \mathbf{X}_s^k$ are these patterns with corresponding locations $l_1^k,\ l_2^k, \ldots\ldots\ldots, l_s^k$.

3) Determine minimum inter-class distance for each pattern using $\mathbf{A}^k$ as

$$\mathbf{d}_i^k = \min_{j=1,2,\ldots\ldots,t_k} (\mathbf{A}^k(l_i^k, j)), \ \text{where } i = 1, 2, \ldots, s. \tag{10}$$

4) Count the number of patterns clustered by s possible clusters i.e. FSHs with $\mathbf{X}_1^k, \mathbf{X}_2^k, \ldots\ldots\ldots, \mathbf{X}_s^k$ as centroids and corresponding radii $d_1^k,\ d_2^k,\ \ldots\ldots\ldots d_s^k$ using algorithm 2.

5) If c^{th} cluster out of s clusters has clustered more patterns then the FSH is created in the hidden layer of FHNN with the centroid $\mathbf{X}_c^k$ and radius d_c^k.

6) Assign $n = \mathbf{n}_c$

7) Steps 7-10, same as MSC.

Algorithm 2 To Determine Patterns Clustered by FSHs

Input: B^k, locations - l_1^k, l_2^k, $\ldots\ldots\ldots l_s^k$ and radii - $d_1^k, d_2^k, \ldots.., d_s^k$
Output: Number of patterns clustered $\mathbf{n} = (n_1, n_2, \ldots., n_s)$

for $i = 1 \rightarrow\ s$ **do**
  $n \leftarrow 0$
  for $j = 1 \rightarrow \alpha_k$ **do**
    if $f(\mathbf{B}^k(l_i^k,\ j), d_i^k) <= 1$ **then**
      $n \leftarrow n + 1$
    end if
  end for
  $\mathbf{n}_i \leftarrow n$
end for

Therefore, the FHNN classifier is constructed with the number of FSHs, which equals the number of clusters created for each class by the above algorithms, forming a hidden layer. Appropriate binary weights are assigned to the links between the hidden layer and the output class neuron. Thus in traditional clustering, the clusters size and number of clusters where restricted by initializing the parameters. Where as in MSC and DMSC, there is no restrictions on fuzzy set hyperspheres (FSHs) that is on cluster size and number of clusters. Even there is no need of overlap tests of clusters as the formed clusters camouflage the clustered patterns. The cluster formation is done based on the appropriate density of patterns which is not the case in the traditional clustering. The MSC and DMSC algorithms are data dependent and to avoid typical traditional clustering no tuning parameter are used because of which there is no restriction of size in cluster formation [1]. Both the MSC and DMSC algorithms do not impose any restriction on number of clusters so no tunning of hyperparameters is required in the hidden layer. Since each FSHs is govern by fuzzy membership functions which obscures cluster patterns such that they remain unaffected by neighboring class patterns so training result for all dataset is 100 %.

TABLE 1. UCI dataset.

Dataset	Total Samples	Features	No. of Classes
Hepatitis	80	19	2
Glass	214	9	7
Heart	297	13	2
Zoo	101	16	7
Segmentation	2310	19	7
Dermatol	150	4	3
Breast	599	9	2
Pima	768	8	2
Ecoli	336	7	8
Liver	345	6	2
Ionosphere	351	34	2
Monks-3	432	6	2
Thyroid	215	5	3
Wine	178	13	3
Iris	150	4	3
sonar	208	60	2

IV. EXPERIMENTAL RESULTS

The performance of the FHNN regarding training accuracy, testing accuracy, and the number of hidden neurons was evaluated [35] using Matlab 2016a and the same system configuration discussed in [36]. The following subsections provide detailed explanations of the experimental results.

A. CASE STUDY I

To gain a proper understanding of the MSC and DMSC learning algorithms, a three-class, two-dimensional example is provided. Twenty-four patterns are included in the training set, with Class 1 consisting of the following patterns:: (1, 5), (1, 7.5), (1.5, 7),(2, 4.5), (0.75, 6), (1.5, 8.5), (1, 9), (0.75, 8). Class 2 has the following patterns: (1, 1), (0.75, 3), (1.5, 2.5), (2, 3.5), (3.25, 4), (4, 1),(2, 1.5), (3, 1.5) and finally class 3 has the following pattern (3.7, 6), (3.75, 8), (3.5, 8.5),(4, 5), (4, 7),(3.75, 9), (3, 9),(3, 8).

1) 2-DIMENSIONAL EXAMPLE USING MSC ALGORITHM

When the FHNN is trained using the MSC algorithm, it constructs seven FSHs in the middle layer. The detailed process of constructing the FHNN classifier using the MSC algorithm is shown in Fig.3(a-f). A scatter plot for the dataset mentioned above is shown in Fig.3(a).

Initially, the training data from class one is used to create the FSHs. After calculating the inter-class and intra-class distance matrices (steps 1 and 2), steps 3 and 4 compute the centroid pattern (1, 7.5) as it has the minimum mean distance with respect to the remaining patterns of class 1. According to step 5, among all the patterns of the other classes (i.e., classes 2 and 3), the pattern (3.5, 8) of class 3 is the nearest to the centroid (1, 7.5), with a distance of 2.6. Step 6 creates the first FSH with this distance as the initial radius and the pattern (1, 7.5) as the centroid. The created FSH as per algorithm 1 clusters the patterns, (.75, 6), (1, 7.5), (1.5, 7), (.75, 8), (1.5, 8.5), (1, 9), of class 1. Since $n > 1$, step 7 assigns the final radius to the created FSH by computing the maximum distance between clustered patterns and centroid which is 1.52. The centroid and final radius of this FSH are stored in **C** and **r**, respectively. As the first FSH of class one is created, according to step nine, the corresponding class node is constructed in the output layer and the matrix **U** is updated accordingly. As per step 10, since $\alpha_k \neq 0$, learning continues further from step 3 and the remaining patterns i.e.(1, 5), (2, 4.5) of class 1 are included in the second FSH with the centroid (1, 5) and radius 1.11. Fig.3(b) shows the class 1 FSHs constructed during learning. Since all patterns of class one are clustered, learning of patterns for class 2 is initiated according to step 10.

Using the same steps, three FSHs were created for class 2 with centroids (1.5, 2.5), (3.25, 4), and (4, 1) with radii 1.0828, 2, and 0.625 respectively as shown in Fig.3(c).

Similarly, for class 3 two FSHs with centroids (3.75, 8) and (4, 5), and with radii 2.0156 and 0.625 respectively, are created as shown in Fig.3(d). The learning also creates the class nodes for these classes and **C**, **r** and **U** are updated accordingly. Since the learning for all 3 classes is over the process stops as per step 11.

So overall created seven FSHs i.e. 02 for class one, 03 for class two, and 02 for class three are shown in Fig.3(e). The architecture of the FHNN classifier with these seven FSHs and 3 class nodes, is shown in Fig.3(f). The Fig.3(f) also depicts the final status of **C**, **r** and **U**.

2) 2-DIMENSIONAL EXAMPLE USING DMSC ALGORITHM

The implementation of the DMSC algorithm using above-mentioned data, constructed five FSHs i.e. 02 FSHs for class one, 01 FSH for class two, and 02 FSHs for class three. The radii and centroids of class one FSHs were the same as the MSC learning algorithm. The centroid of class 2 FSH is (2, 1.5) with a radius of 2.79 and the centroid of class three FSHs are (3.75, 8), (4, 5) with radii 2.01 and 0.625, respectively.

Fig.4(a-f) shows the whole process of construction of the FHNN classifier using the DMSC algorithm. With the same scatter plot as shown in Fig.3(a), Fig.4(a) shows the created first FSH with centroid (1, 7.5) and radius equal to 1.52 for class 1 which clusters the six patterns. Fig.4(b) shows both FSHs of class 1. The second FSH with centroid (1, 5) and radius equal to 1.11 clusters the remaining patterns.

For class 2, there are eight patterns to be clustered. According to the DMSC algorithm, three patterns (1.5, 2.5), (3, 1.5), and (2, 1.5) with the minimum means in ascending order are selected as the initial centroids in step 2. After performing steps 3, 4, and 5, the pattern (2, 1.5) is selected as the centroid instead of the first two patterns, even though their mean values are lower. Fig. 4(c) shows the FSH created for class 2 with centroid (2, 1.5) and a radius of 2.79.

Fig. 4(d) shows the formation of two clusters for class 3 with centroids at (3.75, 8.00) and (4.00, 5.00) and respective radii of 2.01 and 0.62. The overall FSHs created for all three classes are shown in Fig. 4(e).

A class node is created whenever the first FSH for that class is created and accordingly upadation of **r**, **C**, and **U** is done.

The architecture of FHNN classifier with centroids in **C**, radius in **r** for FSHs and the weight matrix **U** for 2-D problem using DMSC algorithm is shown in Fig.4(f).

TABLE 2. Comparison of average % test accuracy with 5-fold validation with existing algorithms.

Dataset	**MSC**	**DMSC**	RBF-R	RBF-N	RBF-WTA	DKP	MLP	KNN	PNN	RBF
Hepatitis	**92.9**	**94.1**	81.9	81.1	82.1	72.0	61.0	56	62.0	65.0
Zoo	93.3	93.3	95.2	94.3	96.2	89.6	99.4	93.5	92.0	83.8
wine	65.0	70.6	95.5	94,4	93.5	93.3	82.7	67.3	68	94
Glass	**76.3**	**79.0**	66.1	66.3	69.1	70.4	52.8	62.4	70.2	38.7
Ecoli	**82.4**	**81.4**	78.5	79.3	81.0	92.9	88.9	92.7	94.4	69.5
Liver	**66.9**	66.1	62.2	62.8	61.0	65.5	64.2	66.6	65.3	53.8.
Ionosphere	**96.3**	92.3	95.5	95.2	94.3	90.3	87.8	85.8	85.8	81.5
Monks-3	85.9	88.1	99.0	95.8	68.6	89.6	94.9	97.1	96.8	97.5
Dermatol	74.9	80.6	92.8	92.1	92.7	90.3	95.9	94.7	94.9	70.2
Breast	96.0	96.6	96.3	96.4	97.0	96.1	94.5	95.7	95.9	94.1
Pima	71.0	**78.0**	75.3	72.1	73.8	74.7	67.5	73.2	70.5	71

TABLE 3. Comparison of % recognition error.

Dataset	MSC	DMSC	Complex-valued	Real-valued	RBF-R	RBF-N	RBF-WTA	RBF	SVM	LIBSVM	SMOSVM
Thyroid	8.4	6.05	2.95	3.78	4.4	5.3	3.7	9.8	/	4.18	/
Heart	21.5	18.2	17.08	19.68	18.1	19.5	19.4	21.1	16.4	16.67	18.1
Ionosphere	3.7	7.7	7.14	6.13	4,5	4.8	5.7	12.9	10.5	6.25	6.3
Breast	4.0	3.4	2.9	1.1	3.7	3.6	3.0	6.7	3.3	3.3	3.1
Segmentation	10.7	11	3.53	4.54	6.4	5.7	5.4	35.1	OM	6.2	/

TABLE 4. Average number of hidden neurons.

Dataset	MSC	DMSC	Complex-valued	Real-valued	RBF-R	RBF-N	RBF-WTA	RBF	SVM	LIBSVM	SMOSVM
Thyroid	26.8	21.2	19.65	20.78	15.1	18.7	14.6	43	/	58.8	/
Heart	91.6	79.4	44.12	46.15	24	27	46	54	216	124/	88.8
Ionosphere	74.2	97	117.12	116.45	65	48	66.6	70	280.8	105.2	58
Breast	76.2	49.4	28.12	29.48	40	35	40	140	527.9	90.2	46.1
Segmentation	274	252	39.52	41.58	200	241	200	462	/	632.8	/

The comparison shows that the DMSC algorithm constructs fewer FSHs than MSC.

B. CASE STUDY II

To evaluate the performance of FHNN, it was compared with other algorithms using sixteen UCI datasets [37], as shown in Table 1. During experimentation, the selected data was divided into 5-fold sequences to facilitate researchers in comparing their results with the FHNN classifier. Table 2 presents the average percentages of 5-fold validation test accuracies of the FHNN classifier and other specified classifiers [4]. Proposed MSC and DMSC experiment results are compared with existing state of the art method like RBF-R, RBF-N, RBF-WTA in Table 2. For dataset Hepatitis, Glass, Ecoli, Liver, ionosphere and Pima are better than existing state of the art methods, while results for other datasets are comparable.

C. CASE STUDY III

The primary concern with the RBFNN lies in the number of hidden neurons formed within its hidden layer [38]. The FHNN classifier's performance, in terms of both the average number of FSHs and relative recognition error, is compared with various classifiers. Tables 3 and 4 present the comparison of FHNN with other classifiers as detailed in [5]. Despite potentially having a greater total number of FSHs for certain datasets compared to other algorithms, the computation time for FHNN is generally lower. Table 4 indicates that FHNN is a swift pattern classifier in terms of retrieval.

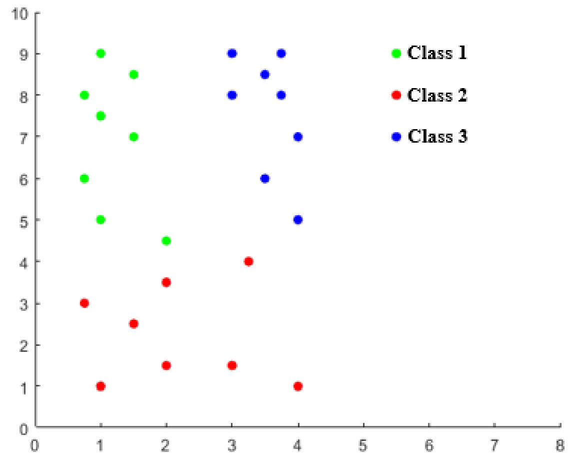
TABLE 5. Comparison table for accuracy in training and testing.

	AP-ELM		ELM		**MSC**		**DMSC**	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis	.92	.87	.98	.83	1	.93	1	.94
Heart	.83	.84	.86	.79	1	.79	1	.82
Sonar	.99	.83	.78	.70	1	.56	1	.51
Wine	1.0	.98	.98	.95	1	.65	1	.71
Zoo	.92	.94	1	.74	1	.94	1	.93
Iris	.97	.98	1	.94	1	.94	1	.93

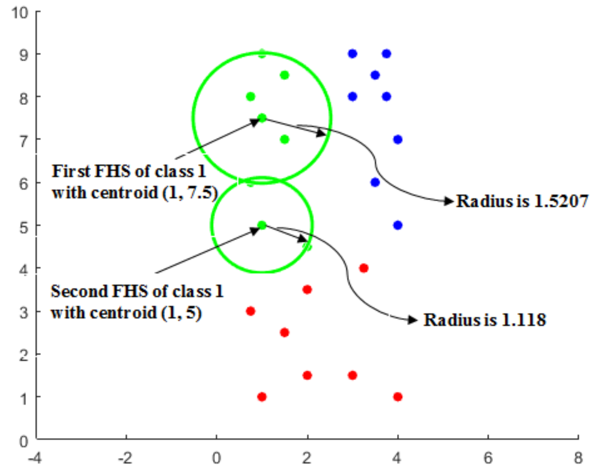
D. CASE STUDY IV

The FSHs in FHNN are characterized by membership functions that accommodate clustered patterns, providing 100% accuracy for any training dataset. In this case study, the performance of the FHNN classifier is compared with the fuzzy neural network proposed by [39] in terms of training and testing accuracy. Table 5 presents the training and testing results. As shown in the comparison table, the training results of the FHNN classifier are 100% for all datasets, whereas the testing results are superior for three datasets and comparable for others. However, averaging the training and testing results reveals that FHNN performs better across all datasets, except for the wine dataset.

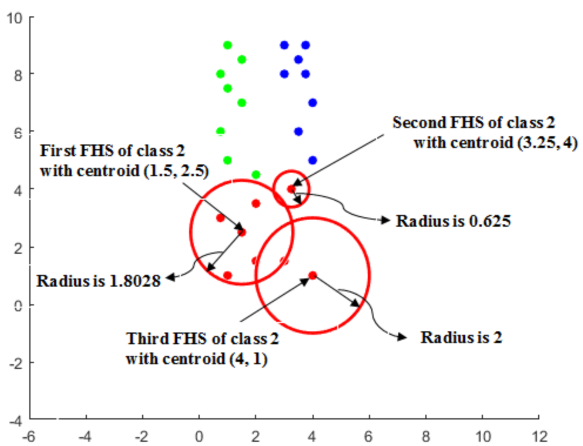
In summary, the simulation results clearly indicate that the FHNN classifier outperforms other algorithms in terms of 5-fold validation test accuracy, number of neurons with recognition rate, and training with testing accuracy, both in superiority and comparability. Additionally, it is observed that the FHNN classifier learns faster and requires less retrieval time compared to other algorithms. The FHNN classifier can be trained on any training dataset without falling



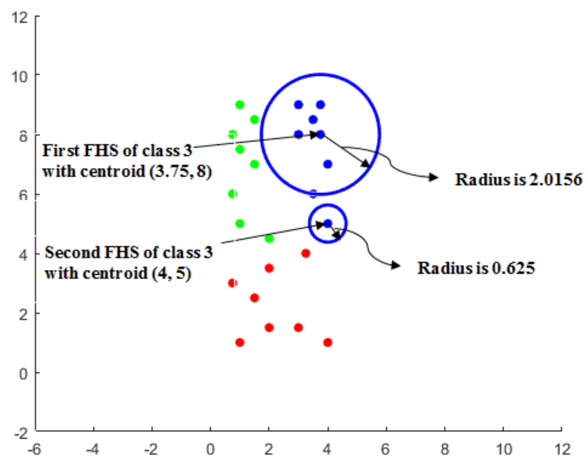
(a) Scatter plot of patterns in 2D space



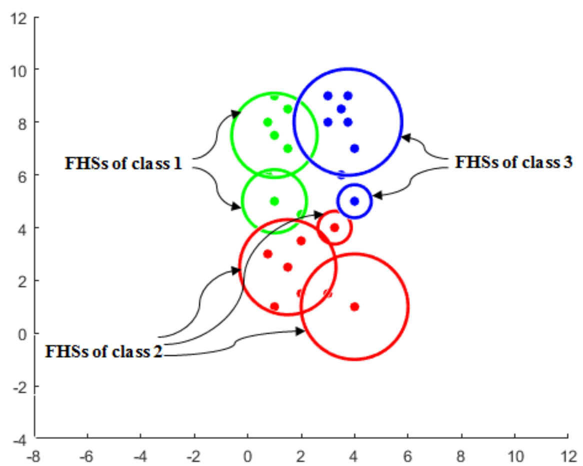
(b) Class 1 FSHs



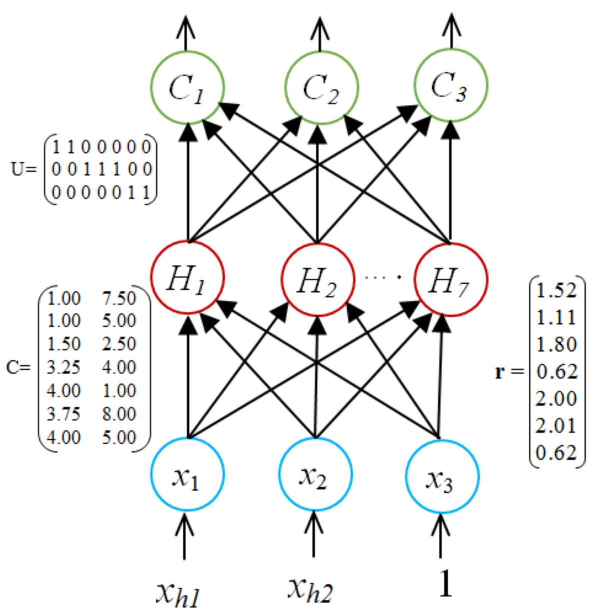
(c) Class 2 FSHs



(d) Class 3 FSHs



(e) FSHs for all classes



(f) FHNN classifier using MSC algorithm

FIGURE 3. **Process in creation of FHNN classifier using MSC algorithm.**

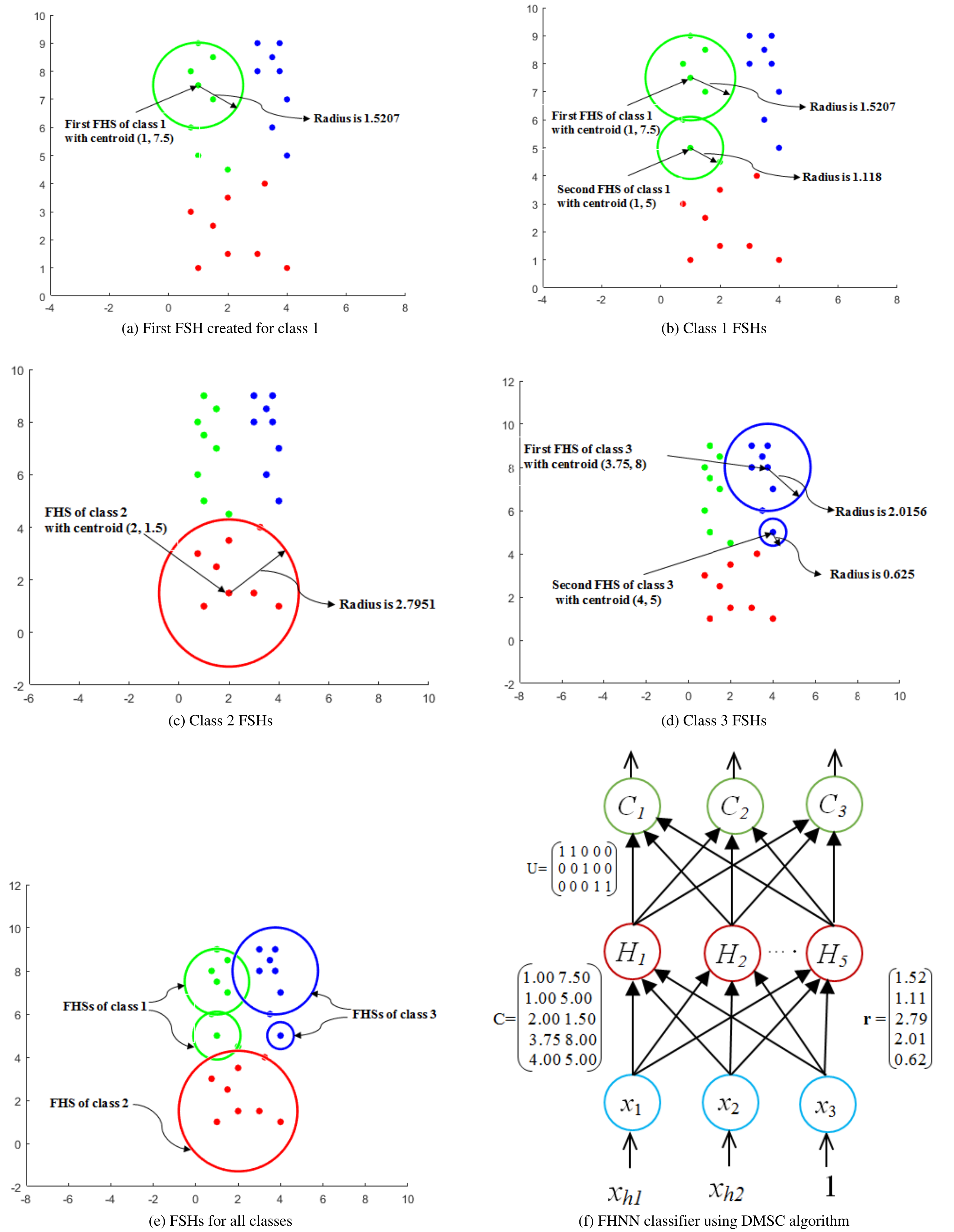


FIGURE 4. **Process in creation of FHNN classifier using DMSC algoritham.**

into local minima. Therefore, the FHNN classifier is highly efficient for pattern recognition, as previously discussed in [40].

V. DISCUSSION

The cluster formation in the hidden layer of the FHNN classifier is refined by the MSC and DMSC algorithms. These algorithms utilize density-based clustering [41] and measure inter-class and intra-class distances [42]. Average percentage test accuracy with five-fold validation for various datasets was reported in recent research papers. Table 2 shows that the test accuracy is higher for Hepatitis, Glass, E. coli, Liver, Ionosphere, and Pima compared to other classifiers. For the remaining datasets, results are comparable. Table 3 indicates that the recognition error for these five datasets is also comparable.

Table 4 demonstrates that both proposed algorithms, MSC and DMSC, exhibit robustness in terms of training and testing time complexity. Table 5 reveals that both algorithms achieve 100% training and testing accuracy for Hepatitis, Zoo, and Iris datasets, outperforming other algorithms. The primary goal of developing these algorithms was to achieve 100% training accuracy with minimal cluster overlap to enhance testing accuracy.

Figure 3 illustrates that for a 2D dataset, seven clusters are formed in the hidden layer due to centroid selection based on cluster density. In contrast, Figure 4 shows that for the same 2D dataset, five clusters are formed in the hidden layer due to centroid selection based on both cluster density and inter-class distance. Comparing Figure 3 and Figure 4, for class 2, MSC creates three clusters while DMSC creates only one cluster. The proposed classifier based on the MSC and DMSC algorithms presents several compelling advantages over traditional classifiers, positioning it as a robust choice in the field of pattern recognition. One of its primary strengths lies in its parameter independence, requiring no tuning parameters yet consistently achieving 100% training accuracy across diverse datasets. This feature not only simplifies implementation but also enhances the algorithm's adaptability to various training scenarios. By employing strategies that circumvent the local minima problem, the algorithms effectively determine an optimal number of clusters suited to each dataset's unique characteristics, thereby ensuring efficient and thorough training.

Furthermore, the algorithms' insensitivity to the order of data presentation ensures stability in clustering outcomes, fostering reliable and reproducible results crucial for data visualization and interpretation. This attribute is particularly advantageous in exploratory data analysis and understanding complex dataset structures. In terms of computational efficiency, the MSC and DMSC algorithms demonstrate competitive performance, achieving minimal training and testing times, especially beneficial for smaller datasets where rapid processing is pivotal.

A distinctive feature of the proposed approach is the incorporation of fuzzy neurons in the hidden layer, characterized by fuzzy membership functions. This design choice enhances the classifier's ability to accurately model and categorize complex patterns within data, contributing to its high training accuracy.

In summary, the MSC and DMSC algorithms represent a significant advancement in pattern recognition due to their robust performance across various metrics including accuracy, computational efficiency, and flexibility in handling different dataset characteristics. These attributes make them well-suited for applications requiring precise and reliable classification, contributing to advancements in fields ranging from biomedical research to industrial automation. Future research could involve optimizing these methods for the number of clusters. Large datasets with a higher prevalence of outliers may result in a larger number of clusters, thus increasing the processing overhead during training, hence a strategy for removing outliers must be formed. The next goal should be to increase dataset accuracy by avoiding overfitting, and experimentation should encompass use of parameters for cluster size and cluster number. Finally, an approach would be to change the fuzzy membership function.

VI. CONCLUSION

The proposed FHNN classifier, utilizing the MSC and DMSC algorithms, constructs FSHs in the hidden layer based on intra-class and inter-class metrics, incorporating fuzzy membership functions within the framework of RBFNN. Unlike traditional RBFNNs that use Gaussian neurons, the FHNN achieves 100% training efficiency across all datasets by employing FSHs. The weights between the hidden and output layers are binary and adapt in real-time during node construction, reducing both training and retrieval time complexities. The MSC and DMSC algorithms effectively select centroids, minimizing the influence of outliers and optimizing cluster width. DMSC, specifically, generates fewer FSHs compared to MSC.

The FHNN classifier imposes no restrictions on FSH size and eliminates the need for overlap tests typical in FMNN classifiers. Its robustness extends to any training dataset, leveraging fuzzy membership functions to obscure patterns such that they remain unaffected by neighboring class patterns. Future research directions could explore limiting FSH size and altering the order of classwise data to further enhance performance. Simulation results using standard datasets demonstrate that the FHNN classifier excels in pattern recognition tasks, affirming its suitability and effectiveness in practical applications.

REFERENCES

[1] M. J. Er, W. Chen, and S. Wu, "High-speed face recognition based on discrete cosine transform and RBF neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 679–691, May 2005.
[2] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed., Upper Saddle River, NJ, USA: Prentice-Hall, 1994.
[3] D. Lowe, "Multi-variable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, no. 3, pp. 321–355, 1988.

[4] M. Rouhani and D. S. Javan, "Two fast and accurate heuristic RBF learning rules for data classification," *Neural Netw.*, vol. 75, pp. 150–161, Mar. 2016.
[5] Y. Liu, H. Huang, T. Huang, and X. Qian, "An improved maximum spread algorithm with application to complex-valued RBF neural networks," *Neurocomputing*, vol. 216, pp. 261–267, Dec. 2016.
[6] P. K. Simpson, "Fuzzy min-max neural networks—Part 1: Classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 776–786, Sep. 1992.
[7] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 769–783, May 2000.
[8] A. V. Nandedkar and P. K. Biswas, "A fuzzy min-max neural network classifier with compensatory neuron architecture," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2004, pp. 553–556.
[9] H. Zhang, J. Liu, D. Ma, and Z. Wang, "Data-core-based fuzzy min–max neural network for pattern classification," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2339–2352, Dec. 2011.
[10] R. Davtalab, M. H. Dezfoulian, and M. Mansoorizadeh, "Multi-level fuzzy min-max neural network classifier," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 470–482, Mar. 2014.
[11] M. F. Mohammed and C. P. Lim, "An enhanced fuzzy minmax neural network for pattern classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 417–429, Apr. 2015.
[12] M. F. Mohammed and C. P. Lim, "Improving the fuzzy min-max neural network with a K-nearest hyperbox expansion rule for pattern classification," *Appl. Soft Comput.*, vol. 52, pp. 135–145, Mar. 2017.
[13] J. Raitoharju, S. Kiranyaz, and M. Gabbouj, "Training radial basis function neural networks for classification via class-specific clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2458–2471, Dec. 2016.
[14] G. E. Tsekouras, A. Manousakis, C. Vasilakos, and K. Kalabokidis, "Improving the effect of fuzzy clustering on RBF network's performance in terms of particle swarm optimization," *Adv. Eng. Softw.*, vol. 82, pp. 25–37, Apr. 2015.
[15] G. E. Tsekouras and J. Tsimikas, "On training RBF neural networks using input–output fuzzy clustering and particle swarm optimization," *Fuzzy Sets Syst.*, vol. 221, pp. 65–89, Jun. 2013.
[16] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, no. 2, pp. 281–294, Jun. 1989.
[17] S. Chen, "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electron. Lett.*, vol. 31, no. 2, pp. 117–118, Jan. 1995.
[18] H. Sarimveis, A. Alexandridis, and G. Bafas, "A fast training algorithm for RBF networks based on subtractive clustering," *Neurocomputing*, vol. 51, pp. 501–505, Apr. 2003.
[19] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, and C.-P. Lim, "An incremental meta-cognitive-based scaffolding fuzzy neural network," *Neurocomputing*, vol. 171, pp. 89–105, Jan. 2016.
[20] A. D. Niros and G. E. Tsekouras, "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach," *Fuzzy Sets Syst.*, vol. 193, pp. 62–84, Apr. 2012.
[21] W. Huang, S.-K. Oh, and W. Pedrycz, "Design of hybrid radial basis function neural networks (HRBFNNs) realized with the aid of hybridization of fuzzy clustering method (FCM) and polynomial neural networks (PNNs)," *Neural Netw.*, vol. 60, pp. 166–181, Dec. 2014.
[22] S.-J. Lee and C.-L. Hou, "An ART-based construction of RBF networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1308–1321, Nov. 2002.
[23] I. Sohn and N. Ansari, "Configuring RBF neural networks," *Electron. Lett.*, vol. 34, no. 7, p. 684, 1998.
[24] Z. Uykan, C. Guzelis, M. E. Celebi, and H. N. Koivo, "Analysis of input-output clustering for determining centers of RBFN," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 851–858, Jul. 2000.
[25] W. Shen, X. Guo, C. Wu, and D. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm," *Knowl.-Based Syst.*, vol. 24, no. 3, pp. 378–385, Apr. 2011.
[26] V. Fathi and G. A. Montazer, "An improvement in RBF learning algorithm based on PSO for real time applications," *Neurocomputing*, vol. 111, pp. 169–176, Jul. 2013.
[27] S.-K. Oh, W.-D. Kim, W. Pedrycz, and B.-J. Park, "Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization," *Fuzzy Sets Syst.*, vol. 163, no. 1, pp. 54–77, Jan. 2011.
[28] K. Ganapathy, V. Vaidehi, and J. B. Chandrasekar, "Optimum steepest descent higher level learning radial basis function network," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 8064–8077, Nov. 2015.
[29] D. P. F. Cruz, R. D. Maia, L. A. da Silva, and L. N. de Castro, "BeeRBF: A bee-inspired data clustering approach to design RBF neural network classifiers," *Neurocomputing*, vol. 172, pp. 427–437, Jan. 2016.
[30] B. M. Ozyildirim and M. Avci, "One pass learning for generalized classifier neural network," *Neural Netw.*, vol. 73, pp. 70–76, Jan. 2016.
[31] A. B. Kulkarni, S. V. Bonde, and U. V. Kulkarni, "Class-specific fuzzy hypersphere neural network," *Proc. Comput. Sci.*, vol. 143, pp. 285–294, Jan. 2018.
[32] D. D'Agostino, I. Ilievski, and C. A. Shoemaker, "Learning active subspaces and discovering important features with Gaussian radial basis functions neural networks," *Neural Netw.*, vol. 176, Aug. 2024, Art. no. 106335.
[33] E. Alhroob, M. F. Mohammed, O. N. Al Sayaydeh, F. Hujainah, N. A. Ghani, and C. P. Lim, "A flexible enhanced fuzzy min-max neural network for pattern classification," *Expert Syst. Appl.*, vol. 251, Oct. 2024, Art. no. 124030.
[34] Y. Zhang and Z. Wang, "A novel radial basis function neural network classifier with fuzzy C-means clustering," *IEEE Access*, vol. 7, pp. 56355–56365, 2019.
[35] R. V. Bidwe, S. Mishra, S. Patil, K. Shaw, D. R. Vora, K. Kotecha, and B. Zope, "Deep learning approaches for video compression: A bibliometric analysis," *Big Data Cognit. Comput.*, vol. 6, no. 2, p. 44, Apr. 2022.
[36] R. V. Bidwe, S. Mishra, S. K. Bajaj, and K. Kotecha, "Attention-focused eye gaze analysis to predict autistic traits using transfer learning," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, pp. 1–33, May 2024.
[37] W. Li, S. Zhai, W. Xu, W. Pedrycz, Y. Qian, W. Ding, and T. Zhan, "Feature selection approach based on improved fuzzy C-means with principle of refined justifiable granularity," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 7, pp. 2112–2126, Jul. 2023.
[38] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 1991.
[39] M. F. Mohammed and C. P. Lim, "A new hyperbox selection rule and a pruning strategy for the enhanced fuzzy min–max neural network," *Neural Netw.*, vol. 86, pp. 69–79, Feb. 2017.
[40] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep. 1992.
[41] O. Nikiforova, A. Romanovs, V. Zabiniako, and J. Kornienko, "Detecting and identifying insider threats based on advanced clustering methods," *IEEE Access*, vol. 12, pp. 30242–30253, 2024.
[42] M. Tareq, E. A. Sundararajan, A. Harwood, and A. A. Bakar, "A systematic review of density grid-based clustering for data streams," *IEEE Access*, vol. 10, pp. 579–596, 2022.

DEEPAK MANE received the Bachelor of Engineering degree in information technology from Dr. Babasaheb Ambedkar Maratwada University, Aurangabad, in 2004, the Master of Engineering degree in computer engineering from Bharti Vidyapeeth University, Pune, in 2011, and the Ph.D. degree in computer science and engineering from the SGGS Institute of Engineering and Technology, Nanded, in 2019. He has more than 19 years of teaching experience. He is currently an Associate Professor with the Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India. He has completed the funded research project "Amalgam Based to Design Translator and Synthesizer for English to the Sanskrit Language" from BCUD, SPPU, Pune. He has published more than 100 research articles in reputed Scopus-indexed international journals, such as Springer, Elsevier, Inderscience, IGI Global, and MDPI. His main research interests include machine learning, deep learning, and supervised clustering. He is a reviewer of many reputed international conferences and Scopus and SCI journals.

ARUN KULKARNI received the Bachelor of Engineering degree in electronics from Marathwada University, in 1990, the M.E. degree in electronics, with a specialization in computer technology, in 2001, and the Ph.D. degree in computer science and engineering from Swami Ramanand Teerth Marathwada University. He has teaching experience of 33 years. He is a Professor with the Department of Information Technology, Thadomal Shahani Engineering College, Bandra, Mumbai, Maharashtra. During the M.E. studies, he has published the paper in IEE (U.K.) journal on fuzzy hyperline segment neural network, under the guidance of Dr. U. V. Kulkarni. He has published papers on speech processing, vector quantization, stock predication, fuzzy neural networks, and color image processing. His area of interests include digital signal processing, neural networks and fuzzy systems, image processing, speech processing, embedded systems, and deep learning. He is a reviewer of many international conferences and journals.

BISWAJEET PRADHAN (Senior Member, IEEE) received the Habilitation degree in remote sensing from Dresden University of Technology, Germany, in 2011. He is currently the Director of the Centre for Advanced Modeling and Geospatial Information Systems (CAMGIS), Faculty of Engineering and IT. He is also the Distinguished Professor with the University of Technology Sydney. He is also an internationally established Scientist in the fields of geospatial information systems (GIS), remote sensing and image processing, complex modeling/geo-computing, machine learning and soft-computing applications, natural hazards, and environmental modeling. He has widely travelled abroad, visiting more than 55 countries to present his research findings. Since 2015, he has been serving as the Ambassador Scientist for the Alexander Humboldt Foundation, Germany. He has published over 850 articles of which more than 750 have been published in Science Citation Index (SCI/SCIE) technical journals. He has authored eight books and 13 book chapters. He was a recipient of the Alexander von Humboldt Fellowship from Germany. He received 55 awards in recognition of his excellence in teaching, service, and research, since 2006. From 2016 to 2018, he was listed as the World's Most Highly Cited Researcher by Clarivate Analytics Report as one of the world's most influential minds. In 2018, 2019, and 2020, he was awarded as the World Class Professor by the Ministry of Research, Technology and Higher Education, Indonesia. He is an associate editor and an editorial board member of more than eight ISI journals.

SHILPA GITE is a passionate Educationist and a Researcher with the Symbiosis Institute of Technology. She is currently working in medical imaging, explainable AI, and quantum AI. She has published impactful manuscripts in reputed international conferences and Scopus/Web of Science-indexed journals and books. She is a guide to several national/international undergraduates, postgraduate, and Ph.D. students from computer engineering and other disciplines. Her research interests include deep learning, computer vision, multi-sensor data fusion, and assistive driving. She was a recipient of the Best Paper Awards at IEMERA Conference, U.K., in October 2020, and at the RAiSE Conference, Dubai, in October 2023. In addition to academics and research, she is also a Reviewer of reputed journals, such as IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, *Neurocomputing*, and *PeerJ Computer Science*. She has also served as a guest editor for two SCIE-indexed journals and also a Scopus-indexed book on computer vision.

RANJEET VASANT BIDWE (Senior Member, IEEE) received the bachelor's degree in computer science engineering from the M. S. Bidve College of Engineering, Latur, and the master's degree in computer engineering from Pune Institute of Computer Technology, Pune. He is currently pursuing the Ph.D. degree with the Symbiosis International Deemed University. He is also an Assistant Professor with the Symbiosis Institute of Technology, Pune. He has written more than 30 peer-reviewed articles published by organizations including IEEE, Elsevier, Springer, MDPI, and Scopus. He has delivered eight presentations at international conferences and two papers at national conferences. He has 12 years of expertise in teaching and research. His areas of research interests include deep learning, computer vision, and video analytics. He is a member of ACM, a Life Member of ISTE, and a Professional Member of IAENG. He has acted as a member of the TPC and a reviewer of several prestigious IEEE and Springer journals and conferences.

CHANG-WOOK LEE (Senior Member, IEEE) received the B.S. degree from Kangwon National University, Chuncheon, South Korea, and the M.S. and Ph.D. degrees from Yonsei University, Seoul, South Korea, in 1999 and 2009, respectively. He had filled the postdoctoral position in InSAR for the ARTS contract with the U.S. Geological Survey EROS Data Center and all work was to be performed as determined by the USGS Project Manager working in coordination with SSSC Management. The work was performed at the USGS Cascades Volcano Observatory in Vancouver, Washington by the National Aeronautics and Space Administration (NASA) Project support, from 2009 to 2011. He is a Full Professor with the Department of Science of Education, Kangwon National University. His research interests include SAR, InSAR, and time-series processing technique development on natural disaster monitoring, and resource characterization. He has authored more than 100 articles in these research fields.

ABDULLAH ALAMRI received the B.S. degree in geology from King Saud University (KSU), Riyadh, Saudi Arabia, in 1981, the M.Sc. degree in applied geophysics from the University of South Florida, Tampa, FL, USA, in 1985, and the Ph.D. degree in earthquake seismology from the University of Minnesota, Minneapolis, MN, USA, in 1990. He is currently a Professor of earthquake seismology and the Director of the Seismic Studies Center, KSU. His current projects involved the applications of electromagnetic (EM) and magnetotelluric (MT) in deep groundwater exploration of the empty quarter and geothermal prospecting of volcanic Harrats in the Arabian shield. He has published more than 150 research papers, achieved more than 45 research projects, and authored several books and technical reports. His research interests include crustal structures and seismic micro zoning of the Arabian Peninsula.