International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I7P210

Volume 71 Issue 7, 94-104, July 2023
© 2023 Seventh Sense Research Group®

Original Article

An Enhanced Fuzzy Hypersphere Neural Network for Pattern Classification

Deepak Mane[1], Sunil D. Kale[2], Anushka Hedaoo[3], Prathamesh Kulkarni[4], Sandip Shinde[5], Prashant Dhotre[6]

[1,3,4,5]*Vishwakarma Institute of Technology, Pune, Maharashtra, India.*
[2]*Vishwakarma Institute of Information Technology, Pune, Maharashtra, India*
[6]*MIT Art, Design and Technology University, Pune, Maharashtra, India*

[1]*Corresponding Author : dtmane@gmail.com*

Received: 24 March 2023 Revised: 16 May 2023 Accepted: 21 June 2023 Published: 21 July 2023

Abstract - *One of the fundamental problems in machine learning and data mining is pattern classification. One frequently employed method for addressing uncertainty in pattern categorization is fuzzy set theory. A supervised clustering technique called the Fuzzy Hypersphere Neural Network (FHSNN) uses fuzzy set theory to categorize patterns. However, FHSNN has certain limitations in handling complex datasets with overlapping classes. This paper proposed an Enhanced Fuzzy Hypersphere Algorithm (EFHSNN) that improves the pattern classification accuracy of existing algorithms. The proposed algorithm uses a modified membership function that adapts to the dataset's characteristics. EFHSNN introduces a new distance metric that captures the similarity between patterns more accurately. It uses a modified version of Murkowski distance instead of Euclidean distance to calculate the distance. Using four popular datasets, we tested the performance of EFHSNN and compared the results to FHSNN and other cutting-edge classifiers. The experimental results show that EFHSNN outperforms the accuracy of FHSNN and other widely used pattern classifiers. The proposed algorithm achieves an average accuracy improvement of 5% over FHSNN on the four datasets. The technique applies to several tasks, including audio recognition, image recognition, and natural language processing*.

Keywords - *Fuzzy set, Fuzzy Hypersphere Neural Network, Modified Fuzzy Hypersphere Algorithm, Supervised clustering, Pattern Classification.*

1. Introduction

Recent pattern classification is a central task providing valuable insights into complex datasets. Recent years have seen significant progress in the field of pattern classification. Methods used for traditional pattern classification problems involve categorizing data into distinct groups, where the task can be approached with either supervised or unsupervised learning. Although many researchers have contributed pattern-matching algorithms using standard datasets, these approaches have not yielded significant improvements in accuracy. In 2001, U.V. Kulkarni proposed an FHSNN, which blends fuzzy collections and hypersphere to build a model cluster. It has been shown that FHSNN outperforms fuzzy min-max and fuzzy neural networks regarding detection ratio. Despite the significant progress observed in pattern classification, some challenges still need to be solved in the proposed algorithms, including issues with previously proposed membership functions.

Uncertainty in pattern categorization is a common challenge in this field, and one effective approach to address it is by leveraging fuzzy set theory. The Fuzzy Hypersphere Neural Network (FHSNN) is a widely used supervised clustering technique that employs fuzzy set theory for pattern categorization. However, FHSNN exhibits limitations when dealing with intricate datasets that contain overlapping classes. An innovative design for pattern categorization has been introduced, incorporating the FHSNN concept and an updated membership function to tackle the previously stated issues and enhance the algorithm's effectiveness. This paper introduces the Enhanced Fuzzy Hypersphere Algorithm (EFHSNN). The proposed algorithm incorporates several modifications to improve the performance of pattern classification. Firstly, a modified membership function is introduced, allowing for greater adaptability to the dataset's characteristics. This adaptation ensures the algorithm can handle complex and overlapping class structures more effectively. Furthermore, EFHSNN introduces a novel distance metric that captures the similarity between patterns more accurately. Instead of relying on the conventional Euclidean distance, the proposed algorithm employs a modified version of the Minkowski distance. This alternative distance metric enhances the algorithm's ability to measure the dissimilarity between patterns, resulting in improved

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

classification outcomes. To validate the effectiveness of EFHSNN, extensive experiments were conducted using four popular datasets (Glass, Liver, Pima, and Monks-3—recognized datasets available in the UCI Machine Learning Repository). The performance of EFHSNN was compared against FHSNN and other state-of-the-art pattern classifiers. Results through experiments demonstrated that EFHSNN consistently outperformed both FHSNN and other widely used classifiers in terms of accuracy.

The paper is organized into distinct portions, commencing with a survey of relevant studies described in Section 2. While Part 4 offers a detailed analysis of the datasets and the results achieved, Section 3 elaborates on the suggested structure and methodology. Section 5 evaluates conclusions and suggests prospective following research areas in its conclusion.

2. Literature Survey

The domain of Machine Learning has seen significant progress in the area of pattern recognition and classification. Most researchers have focused on supervised learning-based methods, which strive to achieve maximum feature separation between classes. As a result, developing a fuzzy algorithm poses a challenge. In 2001, U.V. Kulkarni suggested using an FHSNN [1] for recognizing handwritten characters by combining indistinct features, and the research paper revealed a precision rate of 72.55% utilizing the proposed method. Moreover, U.V. Kulkarni and colleagues [2] introduced a General Fuzzy Hypersphere Neural Network (GFHSNN) that combines unsupervised and supervised techniques in a single-pass training phase. This algorithm has been found to outperform its predecessor. Additionally, the computational complexity of the membership function utilized in the GFHSNN algorithm was significantly lower than that of the GFMM algorithm. As a result, a decrease in training phase duration was observed in the GFHSNN algorithm.

An accuracy of 72.35% was achieved on the IRIS dataset when the algorithm was evaluated. In [3], D. D. Doye et al. proposed a modified version of FHSNN called MFHSNN, which showed an 88.7% accuracy on the Marathi digits classification. P. M. Patil et al. [4] presented FHSNN in 2002, which produced fewer values for patterns near the hyper line segment and outperformed the GFMM approach on the FISHER IRIS data with a mean accuracy of 72.55%. They also proposed a Modular Fuzzy Hypersphere Neural Network (MFHSNN) as an extension of the FHSNN algorithm [5], which used one class features neglecting removal and overlap test, leading to decreased training time. During assessing the algorithm's performance, a precision of 72.35% was attained on the IRIS dataset. In a correlated study, D. D. Doye et al. [3] introduced a modified variation of FHSNN, referred to as MFHSNN, which achieved an accuracy of 88.7% on the classification of Marathi digits. P. M. Patil et al. 2002 [4] represented an FHSNN that generated fewer values for designs adjacent to the hyper line segment and recorded a mean precision of 72.55% on the FISHER IRIS dataset, surpassing the GFMM method. Additionally, they suggested a Modular Fuzzy Hypersphere Neural Network (MFHSNN) as an extension of the FHSNN algorithm [5], which solely used characteristics of a single class without overlap testing and removal, resulting in a reduction in training duration. Moreover, it demonstrated excellent generalization and testing time compared to FHSNN. Additionally, because it expands HSs without doing an overlap test, an additional computation in FMN and FHSNN algorithms, it was found to recognize patterns faster than FMN, FNN, and FHSNN. The Fisher Iris dataset was used for evaluation, resulting in an accuracy of 52.51%. Finally, In their paper, P. M. Patil et al. [6] presented an enhanced version of GFHSNN called Modular GFHSNN (MGFHSNN), which integrated unsupervised and supervised learning techniques into a unified approach for clustering, classification, and a combination of both.

This method enabled a significant degree of parallel processing and demonstrated an accuracy of 72.65% on the Fisher Iris dataset. An FHCNN was developed by B. M. Krishna Kanth et al. [7] integrated classification and clustering methods to differentiate various cancer diseases, resulting in an impressive accuracy of 94.12%. In another study, M. H. Kondekar et al. [8] introduced the Extended FHSNN (EFHSNN), which employed the Manhattan distance measure and achieved a 100% recognition rate on the PolyU HRF fingerprint database while also decreasing the training and recall time. S. S. Chowhan et al. [9] presented an MFHSNN, an extension of FHSNN, for iris recognition, demonstrating better generalization, training, and recall time. The proposed MFHSNN approach was tested on the CASIA dataset of 756 images. D. N Sonar et al. and B. M. Krishna Kanth et al. [10] proposed a method named PFHSNN for lung cancer classification in their work.

The learning phase implemented a pruning technique, which relied on the confidence factor of individual hyperspheres. This approach was an extension of the FHSNN algorithm, introducing a pruned method immediately to the learning phase to obtain a smaller network size. The performance evaluation of the PFHSNN algorithm, JSRT, achieved a high accuracy of 91.66% with better recall time and training time. In 2019, a hybrid approach based on a convolutional neural network and supervised fuzzy clustering was designed for numeral recognition [14]. Similarly, A new model based on fuzzy min-max was introduced by Liu et al. [15] for data classification. Arun Kulkarni et al. [16] suggested a neural network based on fuzzy clustering for detecting patterns. Rashmi Patil et al. [17][18] developed a melanoma analysis system using neural networks and transfer learning. Further, D. T. Mane et al. [19] modified the membership function by taking weighted Euclidean distance and applying the tanh function to the maximum distance by radius ratio, which improved the model's overall performance than the previous approaches.

The Fuzzy hypersphere neural network has been a topic of research for a significant amount of time. Recent research has focused on enhancing the parallelism of the model and minimizing inference time. While these advancements are beneficial from an implementation perspective, they do not address the inherent limitations of the algorithm itself, indicating a need for further improvement.[20] introduces a fuzzy version of the k-nearest neighbor (KNN) algorithm, which incorporates fuzzy logic to handle uncertain or imprecise data in the classification process. The membership assignments generated for categorized samples typically have appealing characteristics. In other words, a sample that has been wrongly categorized will not belong to any class close to one, whereas a sample that has been correctly classified will belong to the correct class close to one. The unbounded fuzzy hypersphere neural network (UFHSNN) model, presented by [21], is a supervised classifier trained in one iteration since the expansion parameter is not used. [22] The size or expansion parameter of hyper boxes or hyperspheres impacts models like FMN, FHSNN, and others. These models require repeated training to determine the optimal value of the expansion parameter that achieves high accuracy in classification with the least number of hyper boxes/hyperspheres.

The parameter for expansion is manually set by the user within the range of 0 to 1. Consequently, input patterns are scanned many times using different expansion parameter values. In addition, when adjusting to new input patterns or classes, the parameter value needs to be readjusted by retraining the current patterns. The designed approach, which is referred to as the online adaption capacity, trains the network in a single pass rather than repeatedly scanning the input information. In conclusion, this literature survey has explored the use of fuzzy set theory for addressing uncertainty in pattern categorization, specifically focusing on the Fuzzy Hypersphere Neural Network (FHSNN). While FHSNN is a supervised clustering technique that shows promise in pattern classification, it has limitations in handling complex datasets with overlapping classes. This paper introduces the Enhanced Fuzzy Hypersphere Algorithm (EFHSNN) to address these limitations.

3. Proposed Architecture

The Minkowski distance is a more flexible distance metric that can capture complex relationships and is more robust to outliers than the Euclidean distance. Hence, it is used in the membership function of the proposed pattern classification technique. Minkowski distance is more flexible and can capture more complex relationships between the features of your data. The Euclidean distance assumes that all features are equally important and have the same impact on the distance calculation. However, in many cases, some features may be more important than others, and the Minkowski distance allows you to adjust the impact of each feature by changing the value of the p parameter, i.e., the power parameter. An updated Minkowski distance function is suggested; along with that, an activation function is also introduced to maintain the membership equation's fuzziness and avoid overfitting. Based on information theory concepts, this membership function is intended to highlight some qualities over others. One way to express the suggested membership function is as follows:

$$D_j = \sqrt[3]{(x_1 - y_1)^3 + (x_2 - y_2)^3 + \cdots + (x_n - y_n)^3} \tag{1}$$

$$d(Pattern_k, O, r) = \tanh\left(\frac{R_j}{D_j}\right) \forall\, (R_j\,, D_j(R, D) \tag{2}$$

Suppose there is an input pattern P with k elements, denoted as pattern = pattern 1, pattern 2, pattern 3..., pattern k..., pattern n. Let C be an array of centroids for a given cluster, represented as C = o1, o2, o3..., oj..., on, and D be the individual distances among kth input patterns. Furthermore, let R be the radius of each centroid, and let max stands for the maximum value between all groups. The calculation of the hyperbolic tangent (tanh) activation function is as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

Where x is the ratio of Radius and Distance, and the exponential function is given by e. The output of the tanh function is within negative 1 to positive 1. Since the fuzziness is to be maintained, all negative values are considered. For that, the Minkowski membership function is enhanced using the following formula:

$$D_j = \sqrt[3]{\sum_{i=1}^{k} feature\_importance[i] \, * \, (x_n - c_n)^3} \tag{4}$$

Where, let x_n denote the nth feature importance value, which was determined by giving a test sample. Let Cn denote the considered centroid cluster's feature value. Additionally, let feature_importance represent the weight of the feature that was calculated while training. It is determined by the values of the Gini Index, which can be computed using the formula.

$$G = \sum_{i=1}^{C} f_i(1 - f_i) \tag{5}$$

Let i, denote the considered feature's value which is unique. Suppose K, concerning the training data, is the total count of distinct categories, and fi represents the occurrence of a distinct value of the feature. The feature needs to be continuous for the formula to be directly applicable. To deal with this, the continuous values are binned to obtain discrete values that can be used in the formula. To preserve the fuzzy nature of the equation, the tanh activation function is introduced, which restricts the output values to the range of 0 to 1.

Additionally, the weighted Minkowski function prevents overfitting and improves the algorithm's performance on testing data. It accomplishes it by assigning minimum weight to features with a high variance that may negatively impact the class prediction while assigning a higher weight to crucial features. By doing so, the proposed membership function relies solely on relevant features to make predictions, addressing the issues in the EFHSNN membership function. Better results are observed with the Minkowski distance compared to the Euclidean distance because it is more resistant to the influence of outliers. When calculating distances with the Euclidean distance, the squared differences between features can amplify the effects of outliers.

On the other hand, the Minkowski distance, particularly with a more considerable value of the p parameter, can reduce the impact of outliers by taking the pth root of the absolute differences. As a result, the Minkowski distance provides better results when dealing with datasets that contain outliers. In a clustering problem, the goal is to group similar data points into clusters. One popular approach is to use a hypersphere-based clustering algorithm, where each cluster is represented by a hypersphere (a higher-dimensional equivalent of a sphere) that encompasses the data points that belong to that cluster.

The first step for this algorithm is to initialize the hyperspheres. Initially, there are no hyperspheres, so new hyperspheres are created. To do this, a set of initial centroids are chosen, acting as the centers of the hyperspheres. The average location of all the data points in a cluster is known as the centroid. To ensure that a centroid is chosen from each class, k initial centroids are randomly selected, one from each class. In step 2 of the hypersphere-based clustering algorithm, each input pattern is processed individually to assign them to their respective clusters.

Before starting to process a pattern, the pattern is checked if it has been visited before. If it has been visited, the pattern is skipped, and the following pattern is considered. This check uses a visited array, which tracks whether a pattern has been processed. This step is essential because it prevents the algorithm from processing the same pattern multiple times, which can lead to incorrect results and slower processing times. By skipping the patterns that have already been processed, computation time is saved, and it is ensured that each pattern is assigned to a cluster only once. Once the initial centroids are chosen, new hyperspheres for each centroid are created. The hypersphere will have a center as the centroid and an initial radius of zero since it does not yet encompass any data points.

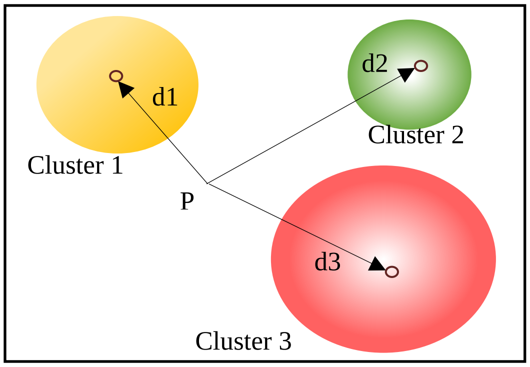

Fig. 1 Computing the distance with centroids (step 2.2)

Step 2.2 involves computing the distance between the input pattern and the centroids of its class. This is done to determine which centroid is closest to the input pattern, which will be used to assign the pattern to its respective cluster using Euclidean Distance. Once the distances are computed, they are sorted in ascending order, ensuring that the input pattern is assigned to the closest centroid first. This reduces the likelihood of misclassification and speeds up the processing of the input pattern.

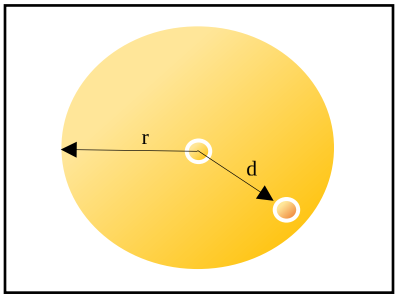

Fig. 2 Existing hyperspheres cover the input pattern (step 2.3)

Step 2.3 of the algorithm checks whether existing hyperspheres cover the input pattern. A hypersphere is assigned to the corresponding cluster if it covers the input pattern. If any hypersphere does not cover the input pattern, the algorithm proceeds to the following steps to determine whether a new hypersphere needs to be created. This step reduces the computation required to assign the input pattern to its cluster by avoiding the creation of unnecessary hyperspheres.

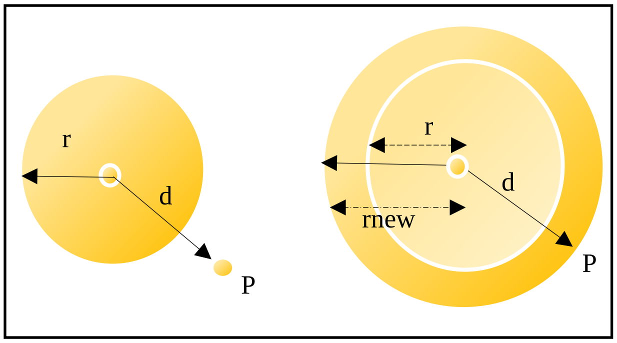

Fig. 3 Expansion criteria (step 2.4)

Step 2.4 is executed when no current hyperspheres cover the current input pattern. In this step, the algorithm checks whether an expansion of the existing hyperspheres is needed to cover the input pattern. If the expansion criterion is satisfied, the algorithm checks whether the expansion of the hypersphere will result in any overlap with the centroids of other classes. Overlap between the centroids of different classes is not allowed in the algorithm, as it can lead to incorrect clustering results. However, overlap between the hyperspheres of the same class is allowed as it can improve the clustering accuracy and reduce the fragmentation of clusters. By checking for overlap between the expanded hypersphere and the centroids of other classes, the algorithm ensures that the clustering remains accurate and that there is no ambiguity in assigning input patterns to their respective clusters.

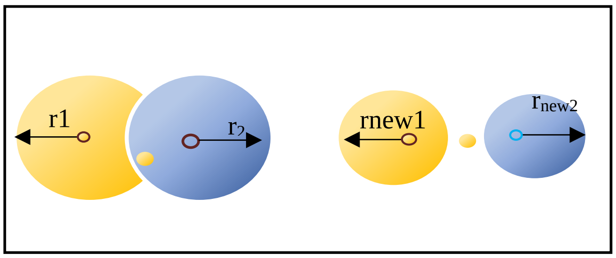

Fig. 4 Creation of new hypershere in overlapping scenario (step 2.5)

Step 2.5 is executed when the expansion criterion is not satisfied and overlap with the existing hyperspheres is detected. In such a scenario, the algorithm creates a new hypersphere centered at the current input pattern with a radius of 0. The count of the number of hyperspheres is also incremented, ensuring that each input pattern is assigned to a cluster even if the existing hyperspheres cannot be expanded to cover the pattern without causing overlap. This helps to accurately cluster the pattern without compromising the clustering accuracy.

Step 2.6 is executed after the creation of a new hypersphere. In this step, using the membership function, the algorithm checks whether the newly created hypersphere overlaps with the centroids of other classes. The membership function evaluates the degree to which an input pattern is a cluster member; it bases its calculation on the distance between the pattern and the cluster centroid. If the newly created hypersphere overlaps with the centroids of other classes, the algorithm modifies the radius of the overlapped cluster.

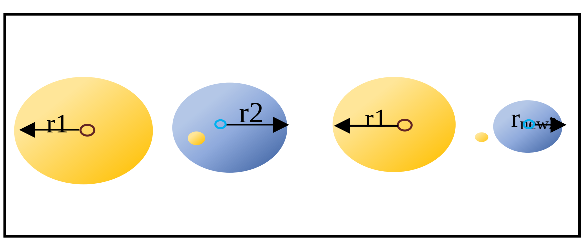

Fig. 5 Overlapping scenario with the centroids of other classes (step 2.6)

Step 2.7 When an existing cluster's radius is modified to avoid overlap, some input patterns may be excluded from the cluster due to the shrinking radius. Although they were formerly components of the overlapped cluster, these patterns are currently past the new radius.

All the above steps are repeated until all input patterns have been correctly clustered and there are no unassigned patterns. To ensure that these patterns are correctly assigned to their respective clusters, the algorithm repeats the same procedure for them by marking them as false so that they can be reevaluated and correctly assigned to their respective clusters based on the modified radii of the hyperspheres.

Finally, in Step 4, the membership function is employed to assess the class of testing input patterns. The membership function determines each cluster's membership degree based on the distance between a testing input pattern and each cluster's centroids. The testing input pattern is then assigned to the cluster with the highest association level, which becomes the predicted class of the input pattern. These steps are crucial for accurately clustering input patterns and predicting new testing data. The algorithm can accurately classify input patterns and predict new data by repeating the clustering procedure until every input pattern has been assigned to a cluster and using the membership function to evaluate the degree of membership of testing data to each cluster.

4. Experiment Results

4.1. Dataset Description

The evaluation of the model encompassed prominent datasets for classifying patterns - Liver, Pima, Glass, and Monks-3. Following is a succinct synopsis of each of these datasets:

4.1.1. Pima

The Pima Indians Diabetes dataset is a well-known dataset in machine learning for binary classification tasks, particularly for predicting the onset of diabetes based on specific diagnostic measurements. The dataset has 768 samples, eight input features, and a binary class label indicating whether diabetes is present or not.

4.1.2. Liver

The Liver Disorders dataset is a public medical dataset that contains 345 instances and seven attributes. The goal is to predict whether a patient has a liver disorder (hepatitis) or not based on various clinical and blood test data.

4.1.3. Glass

The Glass Identification dataset is a multivariate dataset used for pattern classification tasks. It consists of 214 samples of different types of glass, each described by its refractive index and the amounts of 7 different chemical elements in the glass. Predicting the type of glass based on these features is the aim of the classification problem.

```
Pseudo Code for the EFHSNN methodology:
  // K random input patterns
      Set the value of cnt as 0
      Set the value of Overlap as False
       Initialize all visited elements to False
 // Loop through all patterns until all are visited at least once
Repeat:
   For each pattern i from 1 to n:
       If visited[i] is not True, do the following:
   For each cluster center j from 1 to count:
       Calculate the distance between the pattern ρ[i]c and the
       cluster center Ccj and store it in Dst[j]
        Sort the Dst array in ascending order
  If mem(ρ[i]c, Cc) equals 1:
      Set visited[i] to True
 End If
   If spreading out conditions satisfy, then:
    For each m from 1 to k:
    For each l from 1 to cntk:
 If the distance between Cc and Cmk is less than
             or equal to (Radius c + Radius mk), then:
    Set Overlap to True
   Break out of the inner loop
 End If
 End For
 End For
If Overlap is True:
      Set radnew equal to radoriginal
       Set visited[i] to True
       Set radius[j] equal to Distance[j]
  Else:
      Set C[cnt]ck equal to ρ[i]c
      Set radius to 0
      Increment the value of cnt by 1
End If
      Set radius[Overlap] to (|ρ[i] - C[Overlap]| - δ)
For each element j in the set of overlapped patterns:
      Set visited[j] to False
End For
End For
End If
End for
    Run until all elements in visited are True
```

4.1.4. Monks-3

The Monks-3 dataset is a synthetic dataset commonly used for evaluating pattern recognition algorithms. It consists of 18 Boolean attributes and a binary class label (positive or negative). The dataset has 554 instances in total, which are divided into three subsets: training (124 instances), validation (432 instances), and testing (216 instances). The goal is to classify each instance into one of the two classes based on the attribute values. The Monks-3 dataset is known to be a challenging dataset because the target concept is non-linear and not easily separable.

The previously mentioned datasets were used to compare the new methodology with 12 standard supervised pattern classification models. The accuracy evaluation metric is employed, and a 5-fold cross-validation method is adopted. Accuracy indicates the ratio of total accurately predicted testing samples to the total amount of testing samples. The proposed algorithm could handle overfitting and improve its overall performance using the weighted and modified Minkowski's distance equation.

Table 1 compares several machine learning techniques on Monks-3, Liver, Glass, and Pima datasets. The techniques include SHNN, MLP, KNN, PNN, RBF, DKP, RBF-R, RBF-N, RBF-WTA, CSFHSNN, CSFHSNN-Rule2, MFHSNN, and newly mentioned techniques. The performance of each technique is evaluated based on its accuracy percentage. According to the table, the proposed technique achieved the highest accuracy on the Pima dataset with 78%, followed by MFHSNN and CSFHSNN with 76.17% and 75.5% accuracy, respectively. On the Liver dataset, again, the Proposed obtained the highest accuracy of 71.25%, while the SHNN technique came in second with an accuracy of 71%. For the Glass dataset, the Proposed technique achieved the highest accuracy of 82%, followed by PNN and CSFHSNN with 75.9% and 75% accuracy, respectively.

Finally, for the Monks-3 dataset, SHNN achieved an exceptional accuracy of 100%, followed by RBF and DKP with 97.5% and 99% accuracy, respectively. Overall, the table offers a helpful comparison of how different machine learning methods perform on diverse datasets.

Table 1. Performance metrics of existing models

Method	MLP	PNN	KNN	DKP	RBF	RBF-WTA	RBF-N	RBF-R	CSFHSNN	CSFHSNN-Rule2	MFHSNN	SHNN	Proposed EHSNN
Pima	67.5	70.5	73.2	74.7	71	73.8	72.1	75.3	75.5	75.5	72.39	76.17	78
Liver	64.2	65.3	66.6	65.5	53.8	61	62.8	62.2	68.1	68.1	65.8	71	71.25
Glass	52.8	70.2	62.4	70.4	38.7	69.1	66.3	66.1	75.9	70.9	75	80	82
Monks-3	94.9	96.8	97.1	89.6	97.5	68.6	95.8	99	87.1	85.7	85	100	100

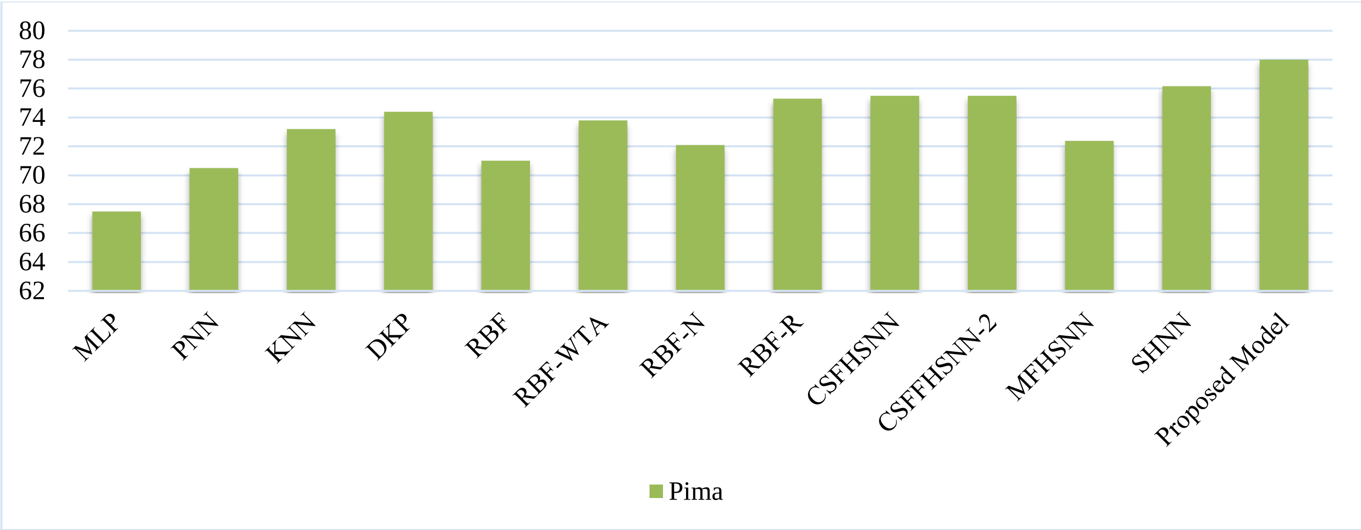

Fig. 6 5-Fold cross-validation of PIMA dataset

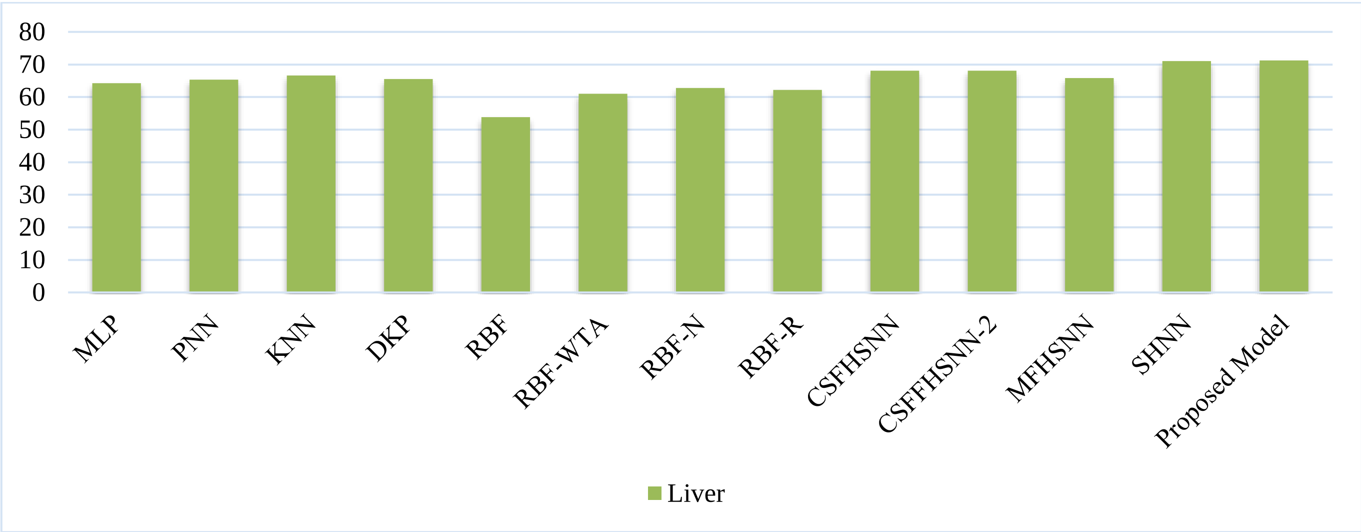

Fig. 7 5-Fold cross-validation of Liver dataset

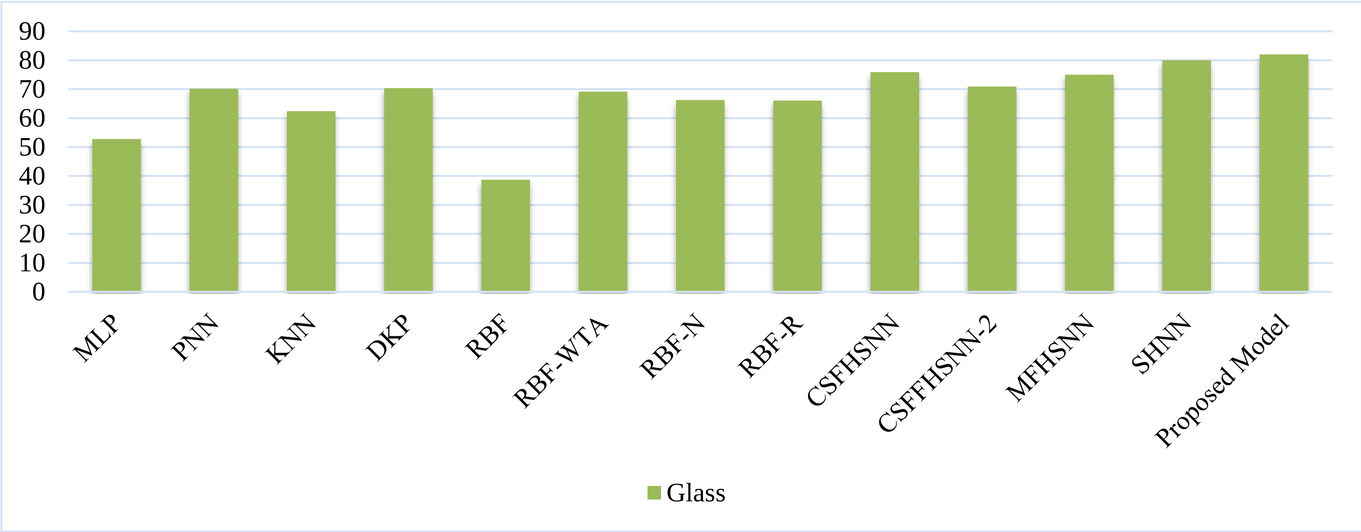

Fig. 8 5-Fold cross-validation of Glass dataset

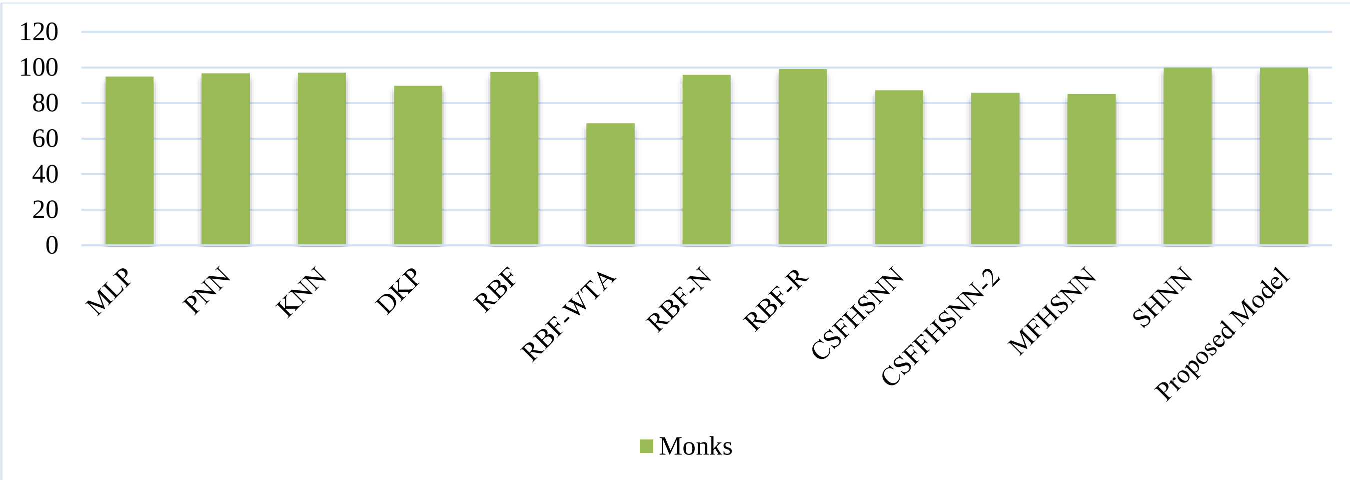

Fig. 9 5-Fold cross-validation of monks-5 dataset

Table 2. PIMA metric evaluation

Label	Class 0	Class 1	Micro Avg	Weighted Avg
F1-score	0.83	0.66	0.74	0.77
Recall	0.87	0.61	0.74	0.78
Precision	0.80	0.70	0.79	0.77
Support	125	67	192	192

Table 3. Liver metric evaluation

Label	Class 0	Class 1	Micro Avg	Weighted Avg
F1-score	0.84	0.29	0.60	0.68
Recall	0.94	0.2	0.59	0.74
Precision	0.80	0.5	0.64	0.71
Support	52	20	72	72

Table 4. Glass metric evaluation

Label	C0	C1	C2	C3	C4	C5	Micro avg	Weighted Avg
F1 Score	0.93	0.82	0.4	0.54	0.66	1.0	0.72	0.81
Recall	1.0	0.73	0.5	0.75	1.0	1.0	0.78	0.82
Precision	0.87	0.93	1.0	0.42	0.5	1.0	0.78	0.87
Support	14	19	04	04	02	07	50	50

Table 2 to Table 5 shows the label-wise metric evaluation.

With reference to the confusion matrix in Figure 12, Table 2 displays the assessment metrics for the PIMA dataset, which include the Recall, Precision, F1 Score and Support for two classes, Class 0 and Class 1. The recall, precision, f1 score and support for Class 0 are 0.87, 0.80, 0.83 and 125. The recall, precision, f1 score and support for Class 0 are 0.61, 0.70, 0.66 and 67. Recall that the precision and support of the model had micro-average F1 scores of 0.74, 0.74, 0.76, and 192, respectively. Recall, precision, support, and weighted-average F1 score for the model were 0.77, 0.78, 0.77, and 192, respectively. Overall, the model's accuracy rates were quite good.

Table 3 shows the liver's evaluation metrics, including F1 score, recall, precision, and support for two classes. Class 0 had a reasonably high accuracy rate; the F1 score was 0.84, the recall was 0.94, and the precision was 0.80, with a support of 52. For Class 1, the F1 score was lower at 0.29, the recall was 0.2, and the precision was 0.5, with support of 20. Overall, the MicroAvg row shows an F1 score, recall, and precision of 0.60, 0.59, and 0.64, respectively, with support of 72. The WeightAvg row provides a weighted average F1 score, recall, and precision of 0.68, 0.74, and 0.71, respectively, with support of 72. Check the confusion matrix for the liver in Fig 11.

Table 4 summarizes the Glass dataset's metric evaluation. It provides F1-score, recall, precision, and support for each of the six classes (Class 0 to Class 5). The micro-average F1-score is 0.72, and the recall and precision are 0.78. The weighted-average F1-score is 0.81, and the recall and precision are 0.82 and 0.87, respectively. Fig 12.

Table 5. Monks-5 metric evaluation

Label	Class 0	Class 1	Micro Avg	Weighted Avg
F1-score	1.0	1.0	1.0	1.0
Recall	1.0	1.0	1.0	1.0
Precision	1.0	1.0	1.0	1.0
Support	204	228	432	432

Table 5 shows the metric evaluation for the Monks-5 dataset. The table includes the F1 score, recall, precision, and support for two classes, labeled as Class 0 and Class 1. For both classes, the F1 score, recall, and precision are perfect, with a score of 1.0. The support for Class 0 is 204, while the support for Class 1 is 228. The MicroAvg and WeightAvg metrics are also perfect, with a score of 1.0. Overall, the model achieved perfect accuracy on the Monks-5 dataset.

Based on Figure 10, it can be inferred that the model's performance is relatively better in correctly identifying negative instances (true negatives) compared to positive instances (true positives).

Actual Values \ Predicted Values	Class 0	Class 1
Class 0	0.86	0.14
Class 1	0.37	0.63

Fig. 10 Confusion matrix for pima dataset

Actual Values \ Predicted Values	Class 0	Class 1
Class 0	0.94	0.56
Class 1	0.78	0.22

Fig. 11 Confusion matrix for liver dataset

Actual Values \ Predicted Values	Class 0	Class 1
Class 0	1	0
Class 1	0	1

Fig. 12 Confusion matrix for pima dataset

Actual Values \ Predicted Values	0	1	2	3	4	5
0	1	0	0	0	0	0
1	0.053	0.74	0	0.11	0.11	0
2	0.25	0	0.25	0.5	0	0
3	0	0.25	0	0.75	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1

Fig. 13 Confusion matrix for glass dataset

The false positive rate is moderate (0.14), indicating that a relatively small percentage of negative instances are being misclassified as positive. However, the high false negative rate (0.37) indicates that a sizable fraction of positive events is mistakenly labelled negative. Based on Figure 11, it can be inferred that the model has classified 22% of the positive instances correctly. This indicates that the model has some capability to identify positive instances. The model performed well in correctly identifying negative instances, with an accuracy rate of 94%. This implies that the model has a high ability to recognize negative instances correctly. The model misclassified 56% of the negative instances as positive. This suggests a relatively high rate of false positives, indicating that a significant portion of negative instances was incorrectly identified as positive. The model misclassified 78% of the positive instances as negative. This indicates a high rate of false negatives, suggesting that a large proportion of positive instances were incorrectly identified as negative. Figure 12 indicates that the model classified all the positive events properly. This indicates a perfect performance in identifying positive instances. The model also correctly classified all the negative instances. This suggests a perfect performance in identifying negative instances. There were no negative instances incorrectly identified as positive. This implies that the model did not produce any false positive results. There were no positive instances incorrectly identified as negative. This indicates that the model did not produce any false negative results.

5. Future Scope

The proposed EFHSNN has the potential for future scope in several ways. First, it improves pattern classification accuracy, which is a fundamental problem in machine learning and data mining. The EFHSNN algorithm overcomes the limitations of FHSNN in handling complex datasets with overlapping classes. Second, the algorithm adapts to the dataset's characteristics, making it more flexible and able to handle a broader range of datasets. Third, the new distance metric introduced in the EFHSNN algorithm captures the similarity between patterns more accurately. The experimental results demonstrate that EFHSNN outperforms FHSNN and other classifiers' classification accuracy, achieving an average accuracy improvement of 5% over FHSNN on four benchmark datasets. The EFHSNN algorithm features hold great promise for researchers and practitioners in machine learning and data mining. Furthermore, the development of more advanced classifiers that build on the EFHSNN algorithm could lead to significant improvements in pattern classification accuracy and pave the way for further innovations in the field.

6. Conclusion

Fuzzy algorithms are computational methods that utilize fuzzy set theory to address uncertainty and imprecision in data and decision-making processes. The paper addresses the problem of pattern classification, which is a fundamental challenge in machine learning and data mining. It focuses on improving the pattern classification accuracy of existing techniques, specifically the FHSNN, which utilizes fuzzy set theory for handling uncertainty in pattern categorization. However, FHSNN has limitations when it comes to complex datasets with overlapping classes. The paper proposes an Enhanced Fuzzy Hypersphere Algorithm (EFHSNN) to overcome these limitations. The key improvements introduced by EFHSNN lie in the modified membership function and the new distance metric. The modified membership function is designed to adapt to the dataset's characteristics, allowing for more precise capture of underlying patterns and uncertainties. EFHSNN can better handle complex datasets with overlapping classes by tailoring the membership function, leading to enhanced classification accuracy. In addition, EFHSNN introduces a new distance metric that accurately measures the similarity between patterns. Instead of relying on the traditional Euclidean distance, the algorithm employs a modified version of the Minkowski distance. This updated distance metric accounts for the complexities present in the dataset and provides a more reliable measure of similarity, thereby improving the accuracy of pattern classification. To validate the effectiveness of EFHSNN, the paper conducts extensive experiments on four popular datasets. The proposed algorithm is compared against FHSNN and other state-of-the-art classifiers.

The results of these experiments demonstrate that EFHSNN outperforms both FHSNN and other widely used pattern classifiers in terms of accuracy. The suggested model was evaluated on the Liver, Glass, Pima, and Monks-3 benchmark datasets, achieving accuracies of 78%, 71.25%, 82%, and 100%, respectively. In comparison, existing methods achieved accuracies of 76.17%, 71%, 80%, and 100% on the same datasets. On average, EFHSNN achieves a 5% improvement in accuracy over FHSNN across the four datasets. This significant improvement showcases the superiority of EFHSNN in pattern classification tasks. Moreover, it is worth noting that the proposed algorithm is not limited to a specific domain. It applies to various tasks, including audio recognition, image recognition, and natural language processing. This broad applicability highlights the potential of EFHSNN to enhance pattern classification performance across different fields.

In conclusion, the Enhanced Fuzzy Hypersphere Algorithm (EFHSNN) offers notable advancements over existing techniques by incorporating a modified membership function and a new distance metric. These improvements enable EFHSNN to handle complex datasets with overlapping classes more effectively, resulting in improved pattern classification accuracy. The experimental evaluations conducted in the paper and the average 5% accuracy improvement over FHSNN provide strong evidence of the algorithm's superiority. Furthermore, the algorithm's versatility allows for its application in various domains, underscoring its potential for enhancing pattern classification in diverse fields such as audio recognition, image recognition, and natural language processing.

References

[1] U.V. Kulkarni, and T.R. Sontakke, “Fuzzy Hypersphere Neural Network Classifier,” *10th IEEE International Conference on Fuzzy Systems, Melbourne,* vol. 2, pp. 1559-1562, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[2] U.V. Kulkarni, D.D. Doye, and T.R. Sontakke, T. R., “General Fuzzy Hypersphere Neural Network,” *Proceedings of the International Joint Conference on Neural Network*, Honolulu, pp. 2369-2374, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[3] D.D. Doye et al., “Speech Recognition Using Modified Fuzzy Hypersphere Neural Network,” *Proceedings of the International Joint Conference on Neural Networks (IJCNN’02)*, Honolulu, Hawaii, vol. 1, pp. 65-68, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[4] P.M. Patil et al., “Recognition of Handwritten Characters Using Modified Fuzzy Hyperline Segment Neural Network*,” 12th IEEE International Conference on Fuzzy Systems*, pp. 1418-1422, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[5] P.M. Patil, U.V. Kulkarni, and T.R. Sontakke, “Modular Fuzzy Hypersphere Neural Network,” *12th IEEE International Conference on Fuzzy Systems,* FUZZ '03, vol. 1, pp. 232-236, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[6] P.M. Patil et al., “Modular General Fuzzy Hypersphere Neural Network,” *17th IEEE International Conference on Tools with Artificial Intelligenc*e, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[7] B. B. M. Krishna Kanth, U. V. Kulkarni, and B. G. V. Giridhar, "Prediction of Cancer Subtypes Using Fuzzy Hypersphere Clustering Neural Network," *IJCSNS International Journal of Computer Science and Network Security*, vol. 11, no. 2, pp. 173-178, 2011. [Google Scholar] [Publisher Link]

[8] M. H. Kondekar, Dr. U. V. Kulkarni, and S. S. Chowhan, "Fingerprint Recognition Using Extended Fuzzy Hypersphere Neural Network," *Journal of Computing*, vol. 3, no. 4, pp. 101-105, 2011. [Google Scholar] [Publisher Link]

[9] S Chowhan et al., "Iris Recognition Using Modified Fuzzy Hypersphere Neural Network with Different Distance Measures," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 6, pp. 130-134, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[10] D N Sonar, and U V Kulkarni. "Pruned Fuzzy Hypersphere Neural Network (PFHSNN) for Lung Cancer Classification," *International Journal of Computer Applications*, vol. 157, no. 7, pp. 36-39, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Dipti S.Charjan, and Mukesh A.Pund. "Pattern Discovery for Text Mining using Pattern Taxonomy," *International Journal of Engineering Trends and Technology*, vol. 4, no. 10, pp. 4550-4555, 2013. [Google Scholar] [Publisher Link]

[12] A.B. Kulkarni et al., "U.V. Class-Specific Fuzzy Hypersphere Neural Network," *Procedia Computer Science*, vol. 143, pp. 285-294, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Abhimanyu Singh, and Yogeshkaswan, "Pattern Recognition Using IR," *SSRG International Journal of Electronics and Communication Engineering*, vol. 1, no. 2, pp. 5-7, 2014. [CrossRef] [Publisher Link]

[14] D. T. Mane, and U. V. Kulkarni, "A Novel Fuzzy Convolutional Neural Network for Recognition of Handwritten Marathi Numerals," *International Journal of High Performance Computing and Networking*, vol. 15, no. 3-4, pp. 158-169, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] Jinhai Liu et al., "Semi-Supervised Fuzzy Min–Max Neural Network for Data Classification," *Neural Processing Letters*, vol. 51, pp.1445-1464, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Arun Kulkarni, and Nikita Kulkarni, "Fuzzy Neural Network for Pattern Classification," *Procedia Computer Science*, vol. 167, pp. 2606-2616, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Rashmi Patil, and Sreepathi Bellary, "Machine Learning Approach in Melanoma Cancer Stage Detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3285-3293, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Rashmi Patil, and Sreepathi Bellary, "Transfer Learning Based System for Melanoma Type Detection," *Revue d'Intelligence Artificielle*, vol. 35, no. 2, pp. 123-130, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Deepak T. Mane et al., "Pattern Classification Using Supervised Hypersphere Neural Network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 8, pp. 1-8, 2022. [CrossRef] [Publisher Link]

[20] James M. Keller et al., "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580-585, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[21] M. S. Mahindrakar, and U. V. Kulkarni, "Unbounded Fuzzy Hypersphere Neural Network Classifier," *Journal of the Institution of Engineers (India): Series*, pp. 1335–1343, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Manisha Mahindrakar Mahindrakar, and U.V. Kulkarni, "Optimized Fuzzy Hypersphere Neural Network with Online Adaptation Capability," *Applied Computational Technologies. ICCET 2022. Smart Innovation, Systems and Technologies*, vol. 303, pp. 60-80, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] D. T. Mane, J. P. Kshirsagar, and U. V. Kulkarni, "Modified Quick Fuzzy Hypersphere Neural Network for Pattern Classification Using Supervised Clustering," *Soft Computing and Signal Processing*, vol. 1118, pp. 227–235, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] D. T. Mane, and U. V. Kulkarni, "Modified Fuzzy Hypersphere Neural Network for Pattern Classification Using Supervised Clustering," *Elsevier Procedia Computer Science*, vol. 143, pp. 295- 302, 2018. [CrossRef] [Google Scholar] [Publisher Link]