# Comparative Study of Fuzzy k-Nearest Neighbor and Fuzzy C-means Algorithms

Pradeep Kumar Jena
National Institute of Science and Technology,
Berhampur, Odisha, India

Subhagata Chattopadhyay
Bankura Unnayani Institute of Engineering,
Bankura-722146, West Bengal, India

## ABSTRACT

Fuzzy clustering techniques handle the fuzzy relationships among the data points and with the cluster centers (may be termed as cluster fuzziness). On the other hand, distance measures are important to compute the load of such fuzziness. These are the two important parameters governing the quality of the clusters and the run time. Visualization of multidimensional data clusters into lower dimensions is another important research area to note the hidden patterns within the clusters. This paper investigates the effects of cluster fuzziness and three different distance measures, such as Manhattan distance (MH), Euclidean distance (ED), and Cosine distance (COS) on Fuzzy c-means (FCM) and Fuzzy k-nearest neighborhood (FkNN) clustering techniques, implemented on Iris and extended Wine data. The quality of the clusters is assessed based on (i) *data discrepancy factor* (i.e., DDF, proposed in this study), (ii) *cluster size*, (iii) its *compactness*, (iv) *distinctiveness*, (v) *execution time* taken, and (vi) *cluster fuzziness (m)* values. The study observes that FCM handles the cluster fuzziness better than FkNN. MH distance measure yields best clusters with both FCM and FkNN. Finally, best clusters are visualized using a Self Organizing Map (SOM).

## General terms:

Fuzzy clustering algorithms, comparisons, datasets, distance measures

## Keywords:

Fuzzy clusters; FkNN; FCM; Cluster fuzziness; Data discrepancy factor (DDF)

# 1 INTRODUCTION

Clusters are defined as the groups of similar data points devoid of any predefined class labels. Clustering is a process of successfully partitioning a dataset into groups, where one group must be different from the other. It is an unsupervised process as the algorithms learn from observations rather than examples, which is seen in classification. Thus, clustering is useful to (i) explore the hidden pattern of any given dataset and (ii) model the data. The popularity of clustering techniques in machine learning is due to its inherent ability to handle different types of (i) attributes in a multidimensional data, (ii) noisy data, and (iii) users having no domain knowledge.

Clusters are of two types, such as crisp and fuzzy. In crisp clusters, the cluster boundaries are well-defined and within the boundary a data point is grouped according to the crisp similarity it has with respect to the representative data or cluster center. Some popular crisp clustering techniques are K-means [1], K-medoid [2], Agglomerative and divisive [3] etc. On the other hand, in fuzzy clusters, the cluster boundary is ill-defined as the data points inside the clusters are chosen according to its degrees of belongingness (i.e. fuzzy memberships) with the clusters. Hence, fuzzy clusters are popular in partitioning the real-world data where the data-data relationships are usually subjective and non-linear in nature [4]. There are several fuzzy clustering techniques available, e.g. Fuzzy c-means (FCM) [5], Fuzzy k-nearest neighbor (FkNN) [6], Entropy-based fuzzy clustering (EFC) [7], Fuzzy ISODATA [8] and so forth. This paper, however, focuses on FCM and FkNN techniques.

In both crisp and fuzzy clustering techniques, cluster centers play the key roles in grouping the data points, because these pose to be the most ideal representative data of the respective clusters. Cluster centers also provide information of the pattern stored within the cluster. These are also useful to measure compactness and distinctiveness of the clusters. Compactness denotes how closely the data points are located with respect to the cluster centers. Distinctiveness measures how far the clusters are lying from each other. A good clustering algorithm must be able to produce compact and distinct clusters.

Similarity measure between the representative cluster center and the random data points (to be clustered) is the initial technique to iteratively cluster the similar data points into and exclude the dissimilar data. Distance measures are the most useful techniques to compute such dissimilarity. There are several distance measures, such as Euclidean (ED), Manhattan (MH), Cosine (COS), Mahalanobis, Hamming, and so on [9]. Generally, the distances between two multidimensional data points are calculated attribute-by-attribute [10]. Detail discussion of all the available techniques is beyond scope of this work. The second focus of this study is to investigate how three distance measures, such as ED, MH, and COS influence the overall clustering performances.

Cluster visualization is an important method to directly display the clusters for interpreting the size, shape, compactness, distinctiveness etc. It is a challenge in case of clusters having multidimensional data points. The third focus of this study is to showcase the best clusters, obtained through the FCM and FkNN techniques using ED, MH and COS distances on a Self-organizing Map (SOM) [11].

# 2 RELATED RESEARCH

Fuzzy clustering techniques are quite popular in various research domains, such as engineering, economics and commerce, biometry and imaging, medical sciences and so on. This paper focuses on FCM and FkNN techniques, applied in various domains. Some recent studies on these two techniques are described below.

## 2.1 Works related to FkNN:

FkNN has been successfully used in various domains of science, such as materials science [12], banking and finance [13] [14],

biometry [15], healthcare [16] and so on. Detail discussion of all the studies are out of scope, however, some obtained from Google scholars, Medline, DBLP, and PubMed are briefly described below.

A comparative study between FkNN and back propagation neural network (BPNN) had been conducted to quantify duck color [12]. The objective of the study was to inject both speed and accuracy for such quantification. The study highlighted that FkNN is faster and more accurate than BPNN.

A bankruptcy prediction model was developed by FkNN approach, where, the size of the neighborhood and fuzzy strength were adaptively specified by both continuous and binary particle swarm optimization techniques [13] [14]. The proposed model was then compared with five other existing classifiers. It was observed that, the model is a powerful early warning system.

An extension of FkNN, called as Fuzzy few-Nearest Neighbor (Ff-NN) had been used to develop a personal authentication system for exit/entry authorization [15]. Twenty six different types of features had been considered for the study. The study concluded that Ff-NN could recognize a person with 88% accuracy, when compared to single-NN (79.2% accuracy).

FkNN has also been used in healthcare, such as in cardiology [16]. An attempt had been made to classify arrhythmias using FkNN, Multiplayer perceptron (MLP) with steepest descent and momentum back propagation and MLP with scaled conjugate gradient back propagation. Output of each classifier was combined with a Mamdani's fuzzy logic controller. The study observed 98% accuracy in the classification task.

## 2.2 Works related to FCM:

There are several studies existing on FCM clustering technique in heterogeneous domains, such as electrical engineering [17], communication engineering [18], image processing [19], bioinformatics and genetics [20], comparison with other techniques [21], healthcare (especially, in mental health) and so forth [4] [10][22] [23]. Below, some recent studies have been showcased.

FCM had been used to cluster multimodal interconnected electricity system (e.g., IEEE 39-bus) [17]. In the said study, a modified similarity measure was considered the group the nodes. The study revealed that the proposed algorithm would be able to appropriately classify the interconnected power system.

A hybrid of Quantum-behaved particle swarm optimization (QPSO) and FCM had been used to detect unwanted intrusions in the network [18]. Gradient descent of FCM had been used to import stronger global search capacity and preventing local minimum issues with FCM. The study confirmed the robustness of the said technique.

FCM had been used in image study to segment MRI images as manual segmentations are highly time-taking processes. It had been seen that to segment the images of multiple sclerosis, FCM-based segmentation worked faster and accurately [19]. FCM had also been tested to segment cDNA microarray image, which consists of thousands of gene sequences and some inherent artifacts when printed on glass slides [20]. The objective of the study was to remove the influence of artifacts and it was successfully accomplished with FCM technique.

A comparative study between FCM and EFC (and its extensions) had been performed [21]. It had been seen that FCM performed best while tested on three standard data sets, such as Iris, Wine and Olitos.

FCM had also been used to screen psychotic disorders based on the Brief Psychiatric Rating Scale (BPRS)-F2. The studies observed that it would be able to cluster such disorders quite effectively, i.e., without outliers and begetting desired number of clusters. The cluster center information of the best clusters obtained by FCM and EFC and its extensions were then extended to develop two Sugeno-Takagi type fuzzy logic controllers. The inference (i.e., diagnoses) thus obtained, were optimized using a binary-coded GA with encouraging results [22] [23].

Based on the available literature, the identified research scopes are as follows.

1) To measure the performance of a clustering algorithm based on the discrepancies of data points lying within and outside the clusters. In this study, Data discrepancy factor (DDF) has been proposed (see equation 8).

2) To study how various distance measures influence the overall clustering performance, especially deciding the size of the cluster, DDF, cluster compactness, cluster-to-cluster distances, and the algorithm run time. In this study, three popular distance measures have been chosen, such as Manhattan (MH), Euclidean (ED), and Cosine (COS).

3) To experiment how the parameter called 'cluster fuzziness' used in both FCM and FkNN influences the cluster patterns, and finally

4) To visualize best clusters obtained from various 'clustering technique-distance measure-cluster fuzziness' combinations using a SOM.

Rest of the paper is organized as follows.

- Section 3 describes the detail methodology of the study.
- Results are shown and discussed in section 4.
- Finally, the paper is concluded in section 5.

# 3 MATERIALS AND METHOD

In this study two standard datasets, such as 'Iris' and 'Extended Wine' data have been used to test and compare the performances of FCM and FkNN.

## 3.1 Data sets: overview

**Iris** is a set of total 150 data, each having four attributes, such as 'septal' length and breadth and 'pedal' length and breadth [24]. The dataset is divided into three class labels (e.g., *iris setosa; iris versicolor; and iris verginica*) each having equal data distributions, i.e., first 50 belongs to *iris setosa*, next 50 goes to *iris versicolor*, and the remaining 50 data belongs to *iris verginica*).

**Wine** (extended), on the other hand, is comparatively a larger dataset [25]. It is composed of total 204 data with three class labels (according to the classes of Wine) and thirteen attributes, such as Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavonoids, Nonflavonoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of dilutes wines and Proline. The original data distribution is as first 85, next 71, and final 48 data points inside the first, second, and the third clusters, respectively.

The methodology adopted is as follows.

1) Developing FkNN and FCM algorithms on Matlab 9 taking ED, MH and COS distances,

2) Testing the algorithms with various values of the 'cluster fuzziness' parameter to note the clustering outcome, and finally

3) Best sets of clusters, obtained through various combinations FkNN, FCM, three distance measures, and cluster-fuzziness values are then visualized by a SOM.

## 3.2 FkNN algorithm:

It works by assigning class membership to the multidimensional data points by measuring the distance to its k-nearest neighbor (kNN). It is worth noting that, here, kNN has been extended into FkNN by computing the 'fuzzy' distances among the data points that define the cluster fuzziness (i.e. 'm' in equation 1 and 2). The implementation algorithm is as follows,

*Step-1:* computing distances from data points to labeled samples

*Step-2:* If kNN have not yet found then to

include data point

else, if a labeled sample is closer to the data point than any other kNN then go to

*Step-3:* replacing the farthest with the new one

*Step-4:* to compute the fuzzy membership and

to repeat steps 1-4 for the next labeled sample.

$$u_{ij} = \frac{\sum_{j=1}^{K} u_{ij}\left(\frac{1}{\left\|x - x_j\right\|^{2/(m-1)}}\right)}{\sum_{j=1}^{K}\left(\frac{1}{\left\|x - x_j\right\|^{2/(m-1)}}\right)} \quad (1)$$

In equation 1, '$u_{ij}$' denotes the membership of the input '$x_{il}$' for the *j*-th cluster (*j* varies from 1 to *K*) based on the fuzzy distance to its kNN. '*k*' denotes the data dimension which let's assume is varying from 1 to M. The notation '*m*' denotes the fuzzy weight of the distance or fuzzy relationship when calculating each of the kNN's contribution to the membership value. In case the value of '*m*' increases, the neighbors are more evenly weighted and their relative distances from the data point that is to be classified will have less effect on each other and *vice versa*. It is important to note that, '*m*' has been varied carefully and the performances of the algorithms are then noted to obtain the optimum '*m*' values for FkNN and FCM and the distance combinations.

## 3.3 FCM algorithm:

The steps of this algorithm implementation are as follows,

*Step-1:* to assume '*K*' number of clusters of 'N' multiple dimensional data points, where, $2 \leq K \leq N$

*Step-2:* data normalization to avoid biasing factors

*Step-3:* to choose 'cluster fuzziness' $m>1$

*Step-4:* initialize membership matrix '$u_{ijk}$' such that $Sum(u_{ijk} = 1.0)$

*Step-5:* computing cluster center for the *j*-th cluster (see equation 6)

*Step-6:* computing distance between '*x*' (data point to be clustered) and the j-th cluster center

*Step-7:* update fuzzy membership matrix according to the distance (*D*) measure (refer to equation 7)

to repeat steps 5-7 until the membership matrix becomes stable.

$$C_{jk} = \frac{\sum_{i=1}^{N} u_{ijk}^{m} x_{ik}}{\sum_{i=1}^{N} u_{ijk}^{m}} \quad (2)$$

$$u_{ijk} = \frac{1}{\sum_{k=1}^{K} (D_{ijk} / D_{ick})^{2/m-1}} \quad (3)$$

In this study, FkNN and FCM algorithms are developed in Matlab 9 by taking three different distance measures, such as Euclidean (ED), Manhattan (MH) and Cosine (COS). The objective is to check how these measures influence the overall clustering task. Equations 4-7 show the ED, MH, COS and ED-norm, respectively.

$$ED = \sqrt{\sum_{i=1,k=1}^{N,M} \left(x_{ik} - x_{ki}\right)^2} \quad (4)$$

$$MH = \sum_{i=1,k=1}^{N,M} \left|x_{ik} - x_{ki}\right| \quad (5)$$

$$COS = 1 - \frac{x_{ik}^{\ t} . x_{ki}}{\left\|x_{ik}\right\| \left\|x_{ki}\right\|} \quad (6)$$

In the above equations, '*N*' and '*M*' denote total number of data points and the data-dimension. The notations '$x_{ik}$' and '$x_{ki}$' refer to M-dimensional data points (where, $i \neq k$)**.** The superscript'*t*' seen in equation 6 denotes the transpose. In equation 6, the denominator is the product of ED norm of vector '$x_{ik}$' and '$x_{ki}$'. The ED norm of say '$x_{ik}$' is expressed as,

$$x_{ik} = \sum_{i=1,\, j=1}^{N,M} x_{ik}^{\ 1/2} \quad (7)$$

It is worth noting that ED norm represents the length of the vector.

## 3.4 Proposed parameters to measure clustering performances:

Clustering performance of FCM and FkNN with three different distance measures (i.e. MH, ED and COS) and several values of cluster fuzziness (the 'm' parameter shown in equations 1,2 and 3) are compared based on the following parameters.

**a) Size of the clusters (CL_SIZE):** The size is determined by the total number of data points lying within each cluster. It is important to note that while developing the algorithms, the movement of each data point has been tracked to see any positional discrepancy among the data points within the clusters.

**b) Data discrepancy factor (DDF):** Data discrepancy is measured by noting the positional discrepancies among the data points during clustering. It is computed by adding the number of (i) 'wrong' data points grouped inside (*WI*), (ii) the 'correct' data points lying outside (*WO*) of any $k^{th}$ cluster and (iii) number of data points, which could not be clustered i.e. the outliers (OL) when matched with the representative data ($C_k$). Finally, it is expressed as a percentage of the total number of data points (*N*). Ideally the *DDF* must be 0%, i.e. all the data points are clustered as it should be and there is nil outlier. Its significance is to evaluate the 'under' and 'over' fitting of the data. An example of DDF computation is displayed in table 1 in the following section.

$$DDF = \frac{1}{C_k}\left[(WI + WO + OL)\right] \times 100 \qquad (8)$$

In this paper it is further examined how different values of cluster fuzziness (i.e., '*m*') influence the cluster size and thereby the DDF (refer to section 4.1.2.1 for detail).

**c) Cluster compactness (CL_COMP):** it is measured by calculating the average Euclidean distance of all data points with respect to the cluster centers (see equation 9). In this equation, '$x_0$' denotes the cluster center and '$x_i$' are the data points (where 'i' varies from 1 to N) present in the neighborhood. Good clusters should be compact in nature.

$$C' = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(x_o - x_i)^2} \qquad (9)$$

**d) Cluster distinctiveness (CL_DIST):** it measures the pair wise Euclidean distance between the cluster centers (see equation 10). Less distance determines the extent of overlapping, which is not desirable i.e. good clusters must not overlap each other.

$$D(C_i, C_j) = \left(x_0 - x_0^*\right)^2 \quad \text{where, } i \neq j \qquad (10)$$

In this equation, $x_0$ and $x_0^*$ are the cluster centers of the clusters $C_i$ and $C_j$.

**e) Run time (ALGO_RUN):** it is the execution time measured for the complete running of each algorithm in a P4 computer with 1GB RAM and 3.0 GHz processor. It is worth noting that in this paper the effects of various 'm' values are studied on run time for several distance-algorithm combinations (see section 4.1.2.2 for detail).

Finally, best clusters obtained using the technique-distance combinations are visualized using a Self-organizing map (SOM).

## 3.5 Cluster visualization with SOM:

SOM works in three phases – (i) Competition, (ii) Cooperation and (iii) Updating. It follows the principle of unsupervised learning, i.e., learning by observations. 'Competition' layer is consisted of a number of neurons equal to the number of input vectors ($X_i$) having random connection strengths (*W* [0, 1]). Euclidean distance is calculated between each neuron and the input vector with the help of connection weights or '*W*', iteratively. The notation'*t*' is the number of iterations that varies from 1 to τ. The minimum ED is considered in selecting the winner neuron (*n*). In the 'cooperation' phase, the neighborhood of '*n*' is decided by computing the lateral Euclidean distances among the remaining neurons. Gaussian function is considered as the neighborhood function for making the neighborhood search (see equation 11). Finally, the weights of the winner neurons with the neurons of its neighborhood are 'updated' iteratively with a learning parameter, 'η' [0, 1] (see equation 12) till the network becomes stable, i.e., no further updating of weights is required (i.e. the weights are converged) or the maximum number of iterations (τ) has been reached. It is important to mention that, with iterations, the required learning and its effects on the load of weight updating diminishes. In other words, the neighborhood shrinks with iterations (see equation 13 and figures 3, 4 and 5).

$$h_{j,n(xi)}(t) = \exp\left(-\frac{D^2_{j,n(x_i)}}{2\sigma_i^2}\right) \qquad (11)$$

$$W_j^i(t+1) = W_j^i(t) + \eta(t)h_{i,n(X_i)}(t)[X_i - W_j^i(t)] \qquad (12)$$

$$\sigma_t = \sigma_0 \exp\left(-\frac{t}{\tau}\right) \qquad (13)$$

At the end, in order to map higher dimensional data (i.e., the input) into 2-dimension (i.e., the output), the ED of each *'n'* is measured from the origin of the higher dimensional space. This information is used to draw a number of circular arcs, which is now equal to the number of *'n'* in a 2-D space, keeping the center of origin (0,0) (26). All *'n'* are now located in 2-D space and the *'n'to'n'* distances are kept same as it was in the higher dimensional space. It preserves the topological information of the original data points. The SOM algorithm has been developed in Matlab 9.

# 4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, experimental results are shown and discussed as per the objectives of the study. *The first* objective is to define the cluster size in terms of DDF computation. *The second* objective is to explore how various distance measures (i.e. MH, ED, and COS) could influence the performance of FkNN and FCM algorithms. This paper has proposed six parameters, such as (i) Cluster size (CL_SIZE), (ii) Data discrepancy factor (DDF), (iii) Cluster compactness (CL_COMP), (iv) Cluster distinctiveness (CL_DIST), (v) the execution/run time of the algorithm (ALGO_RUN) and (vi) the cluster fuzziness '*m*' to test and compare the performances of the algorithms. *The third objective* of the study is to test the effects of cluster fuzziness (i.e., the parameter '*m*') on clustering performance and the amount of time taken for execution. The *fourth objective* is to visualize the best clusters obtained with various technique-distance-cluster fuzziness combinations using a SOM.

## 4.1 Performance measure:

The performance of FCM and FkNN with various distance measures, such as MH, ED, and COS and chosen cluster fuzziness (m) are examined and compared. It is important to note that DDF and cluster fuzziness (m) are the major emphasis of this paper. Hence, these two concepts have been discussed separately in detail.

### *4.1.1 According to DDF computation:*

One of the cluster quality measures is the *DDF* computation. It is calculated using equation 8. The paper argues that it is the most important measure among all other measures to judge the performance of any clustering technique. Conventionally, good clustering is assessed by counting total number of data points within a cluster [27]. If the number equals to the number of desired data points, the cluster is said to be perfect [27]. This paper proposes that the goodness of the clustering techniques must not be judged based on only the data count inside a cluster, rather the goodness of a cluster must be tested by summing up the data points which are (i) present within a cluster where it should not be and vice versa and (ii) not clustered i.e. outliers (OL). While clustering, the iterative entry and exit of the data points has been carefully monitored. Table 1 shows an example of the proposed vs. conventional way of DDF calculation.

**Table 1. A sample of *DDF* calculation on *Iris* data, *FCM* algorithm with *MH* measure.**

| # Cluster | Data points | Target | Observed | # Wrong data points | OL | Proposed DDF (%) | Conventional DDF (%) |
|---|---|---|---|---|---|---|---|
| 1 | 1- 50 | 50 | 50 | 00 | 00 | {(0+4+6+0)/ 150}x100= 6.66% | {(0+1+1+0)/1 50}x100=1.3 % |
| 2 | 51 – 100 | 50 | ***49*** | *04**(107,120,134,135)*** | 00 | | |
| 3 | 101 - 150 | 50 | ***51*** | *06**(51,53,57,71,78,87)*** | 00 | | |

In this example, in cluster 1 first 50 data points are present. But, cluster 2 and 3 has one less and one more data, respectively. So based on the data count inside the clusters i.e. by the conventional way, the data discrepancy is only 2 and accordingly the DDF is 1.3%. However, by monitoring the individual data label it is seen that four data points, such as 107, 120, 134, and 135 have entered cluster 2, but these must be inside cluster 3 as per the Iris classification. Similarly, 51, 53, 57, 71, 78, and 87 these six data points are lying inside cluster 3, while they must be inside cluster 2. According to the proposed DDF it is now enhanced to 6.66% from 1.3% and reflects the clustering performance in a much holistic way. In this paper, DDF for all combinations (algorithm used, distance measures chosen, and the values of '*m*' parameter) are computed to critically assess the performance of the clustering techniques.

### *4.1.2 According to cluster fuzziness ('m'):*

One objective of this paper is to check how the cluster fuzziness parameter (i.e. '*m*') influences the clustering performance in terms of DDF and the run time. Figures 1 to 4 shows the influence of 'm' parameter on DDF. Figures 5 to 8 shows the effect of 'm' on the run time.

#### *4.1.2.1. Effect of 'm' on DDF:*

The objective is to test how a fuzzy clustering algorithm might handle the fuzziness among the data points which is true for real-life data. The paper proposes that the algorithm that is able to handle maximum fuzziness is robust. With this concept, from figures 1(a) to (d), it may be noted that FCM with MH distance is able to handle maximum fuzziness (m = 12) to get the best clusters with least DDF (6.66%) in Iris data. On the other hand, FkNN performs better in the extended Wine data (m = 7.5; DDF = 6.66%) with MH distance.

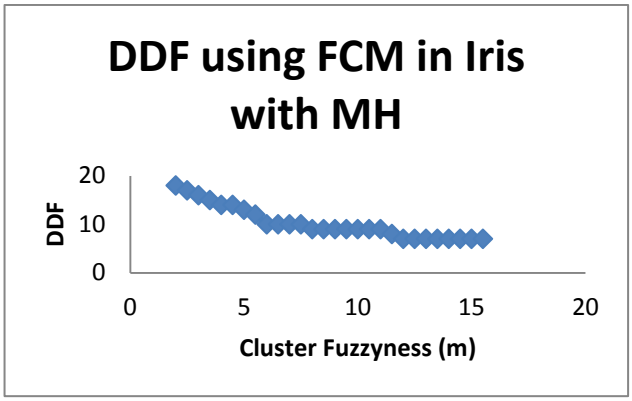


**Fig.1(a). DDF vs. 'm' (FCM in Iris with MH distance).**

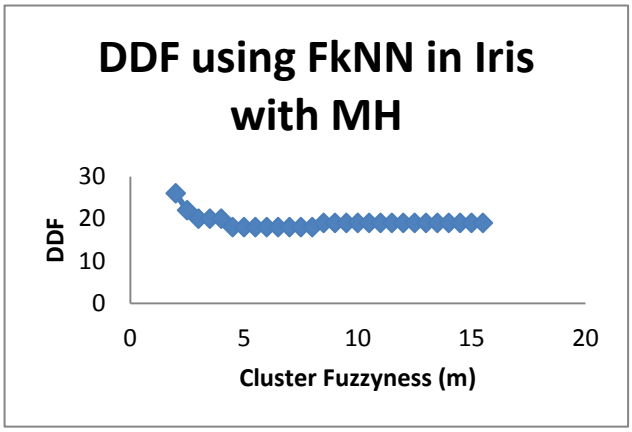


**Fig.1(b). DDF vs. 'm' (FkNN in Iris with MH distance).**

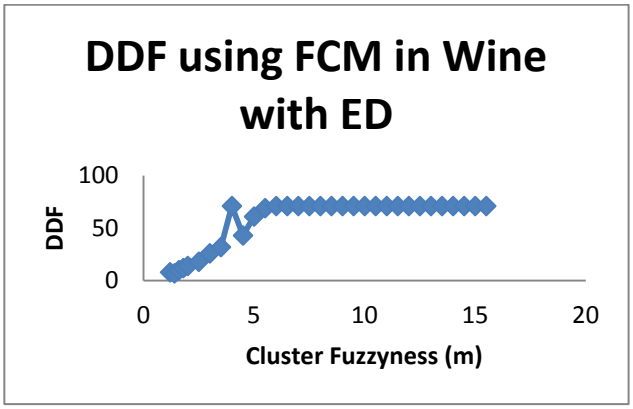


**Fig.1(c). DDF vs. 'm' (FCM in extended Wine with ED).**

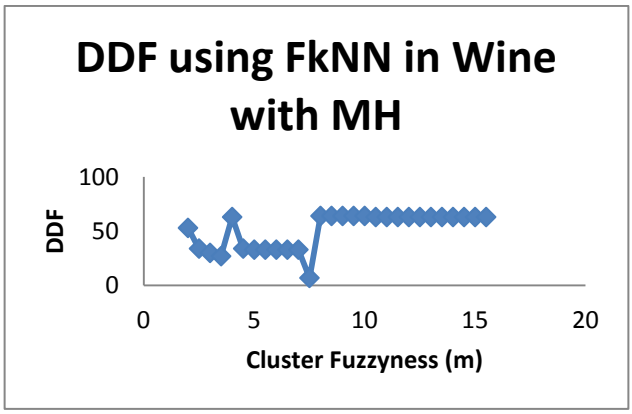


**Fig.1(d). DDF vs. 'm' (FkNN in extended Wine with MH distance)**

### *4.1.2.2. Effect of 'm' on run or execution time:*

The impact of cluster fuzziness (*m*) on run time has also been tested and displayed in figure 5 to 8. It may be noted that run time increases with increment of fuzziness, but interestingly such increment is not monotonous. There are also some decrements as seen in figure 2(a) to (d). The plausible reason could be that initially the algorithm is able to take care of the higher fuzziness, but later it is unable to do so. So, the paper argues that FCM manages the fuzziness best in extended Wine data with ED measure.

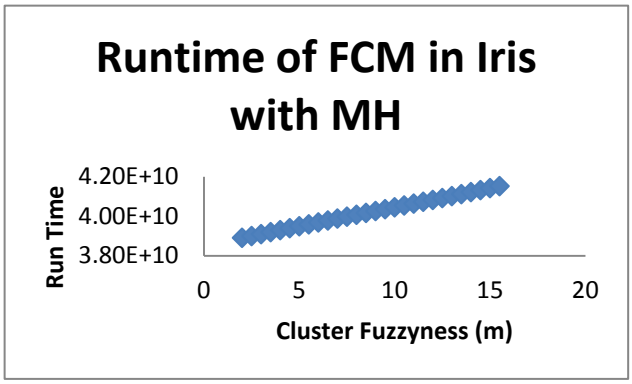


**Fig.2(a). Run time vs. 'm' (FCM on Iris data with MH distance).**

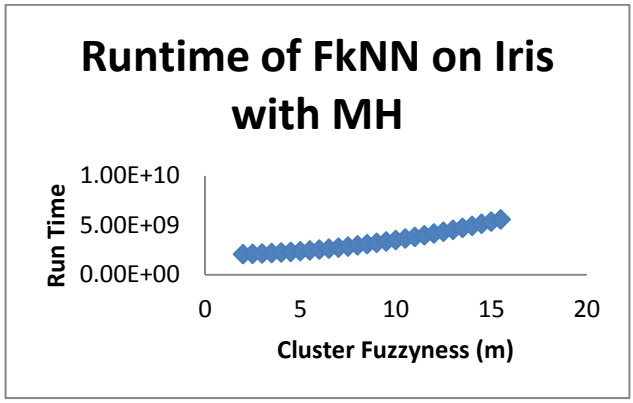


**Fig.2(b). Run time vs. 'm' (FkNN on Iris data with MH distance).**

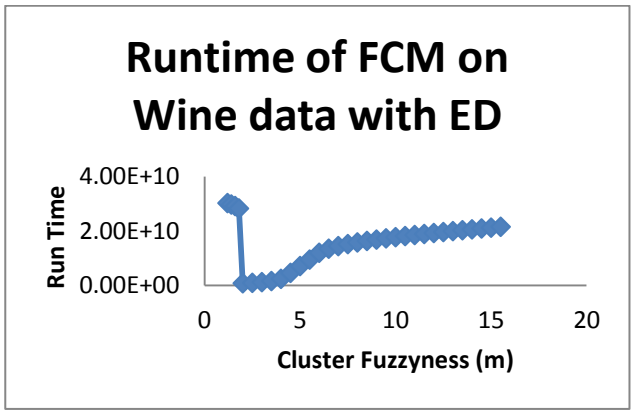


**Fig.2(c). Run time vs. 'm' (FCM on extended Wine data with ED).**

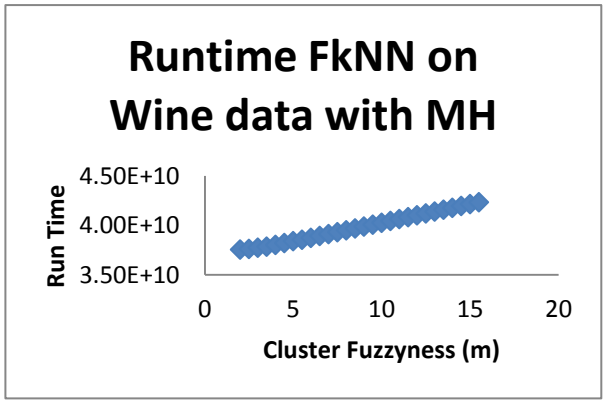


**Fig.2(d). Run time vs. 'm' (FkNN on Extended Wine data with MH distance).**

Summarily, FCM handles the cluster fuzziness better than FkNN algorithm on both Iris and extended Wine datasets with MH and ED measures.

### *4.1.3 According to other parameters:*

Now, the quality of FkNN and FCM clustering techniques based on four other parameters, such as (i) cluster size (CL_SIZE), (ii) cluster compactness (CL_CMP), (iii) inter cluster distance (CL_DST), and (iv) Execution/run time (ALGO_RUN). Table 2 and 3 show the results obtained by the FkNN and FCM algorithms, respectively on Iris and the extended Wine dataset.

**Table 2. Performance of FkNN with various distance measures according to the clusters.**

| Data | Distance | CL_SIZE | DDF | CL_CMP | CL_DIST | ALGO_RUN | 'm' |
|---|---|---|---|---|---|---|---|
| **Iris** | **MH** | *50,64,36* | *0+2+16+0* | 0.486,0.588,1.25 | 3.455,4.746,1.421 | **2.30e+09** | 4.5 |
| | ED | 50,64,36 | 0+2+16+0 | 0.486,0.588,1.25 | 3.455,4.746,1.421 | 1.07e+10 | 4.5 |
| | COS | 52,61,37 | 2+7+17+0 | 0.767,0.617,1.149 | 3.226, 3.483, 1.417 | 8.75e+09 | 2 |
| **Wine** | **MH** | *90,64,50* | *0+7+0+0* | *180.866, 139.042, 92.088* | 550.493,410.175,140.653 | 7.86e+10 | 7.5 |
| | ED | 63,89,52 | 24+6+0+0 | 380.11,177.098,88.546 | *258.023,321.124,64.562* | 7.35e+10 | 9.5 |
| | COS | 97,57,50 | 0+14+0+0 | 223.865,228.093,136.835 | 575.069,170.086,405.091 | 6.77e+10 | 9.5 |

**Table 3. Performance of FCM with various distance measures according to the clusters.**

| Data | Distance | CL_SIZE | DDF | CL_CMP | CL_DIST | ALGO_RUN | 'm' |
|---|---|---|---|---|---|---|---|
| **Iris** | **MH** | *50,49,51* | *0+3+4+0* | 0.486,0.768,0.847 | 3.455,4.397,1.0 | 4.08e+10 | 12 |
| | ED | 50,52,48 | 0+6+4+0 | 0.486,0.724,0.90 | 3.455,4.397,1.0 | 3.26e+10 | 8 |
| | COS | 50,36,64 | 0+25+11 | 0.504,1.499,0.670 | 3.911,4.840,1.162 | 1.34e+10 | 8 |
| **Wine** | MH | 86,53,65 | 1+7+0+0 | 189.278,167.900,70.837 | 550.493,410.175,140.453 | 5.40e+10 | 1.1 |
| | **ED** | 86,52,66 | 1+6+0+0 | 189.278,171.129,69.764 | 550.493,410.175,140.653 | 5.95e+10 | 1.5 |
| | COS | 65,86,53 | 1+7+0+0 | 250.429,103.896,129.090 | 597.034,310.051,287.054 | 9.370e+10 | 1.1 |

From tables 2 and 3, it may be noted that,

- On *Iris* data:
  a) FCM generates best sized clusters, i.e. CL_SIZE (50,49,51) using *MH* distance and the corresponding DDF is 6.66%.
  b) Most compact (CL_CMP) are seen with *MH* distance in both FkNN and FCM.
  c) Most distinct clusters are produced with *MH* and ED by FCM and FkNN, respectively.
  d) Least run time is noted in FkNN with *MH* distance for obvious reasons.
  e) The highest levels of cluster fuzziness (i.e. '*m*') have been handled by FCM algorithm with *MH* and ED compared to FkNN.
- On extended *Wine* data,
  a) FCM with ED produces best sized clusters (86,52,66), DDF is 6.66%.
  b) FkNN produces the best clusters which are also most compact and distinct with MH distance.
  c) Least run time is noted in FCM with MH for obvious reasons.
  d) Most cluster fuzziness has been handled by FkNN algorithm with MH, ED, and COS compared to FCM.

Summarily, from this experiment, the following technique-distance combinations have most efficiently handled the Iris and extended Wine datasets.

- On Iris data: (i) FkNN with MH distance with '*m*' = 4.5 and (ii) FCM with MH distance with '*m*' = 12. FCM is better in terms of lowest DDF value (6.66%) and handling the highest level of cluster fuzziness (i.e. '*m*' = 12).
- On extended Wine data: (i) FkNN with MH distance with '*m*' = 7.5 and (ii) FCM with ED with '*m*' = 1.5. FkNN and FCM are found equally efficient in terms of DDF values computed (3.43%), but FkNN can handle the higher level of cluster fuzziness.

## 4.3. Visualization of best clusters using a SOM:

Finally, the best clusters of Iris and extended Wine data obtained by the FkNN and FCM and the best distance combinations have been plotted as Convergence vs. Weight change using SOM. Figure 3(a) through (c) shows the three best clusters, *iris setosa*, *iris versicolor* and *iris verginica*, respectively obtained by FCM with MH distance. Similarly, figure 4(a) through (c) shows the best iris clusters obtained by FkNN with MH. Likewise, best clusters obtained by FCM and FkNN on extended Wine data are displayed in figure 5(a) to (c) and figure 6(a) to (c), respectively. The figures show the iterative convergence of the weights attribute wise based on the learning and gradual neighborhood compression. In these figures, the arrays of data pints converging at '0' denote the attributes of the datasets.

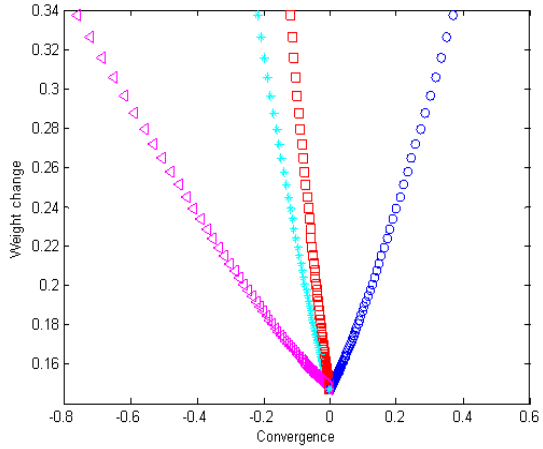


**Fig.3.(a) *iris setosa* (FCM with MH and 'm' = 12, DDF = 6.66%).**

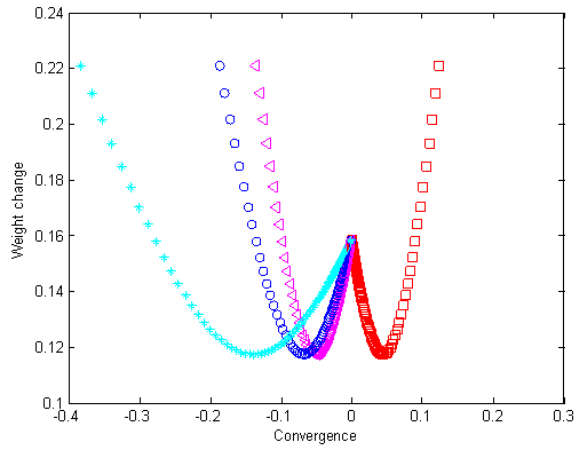


**Fig.3.(b) *iris versicolor* (FCM with MH and 'm' = 12, DDF = 6.66%).**

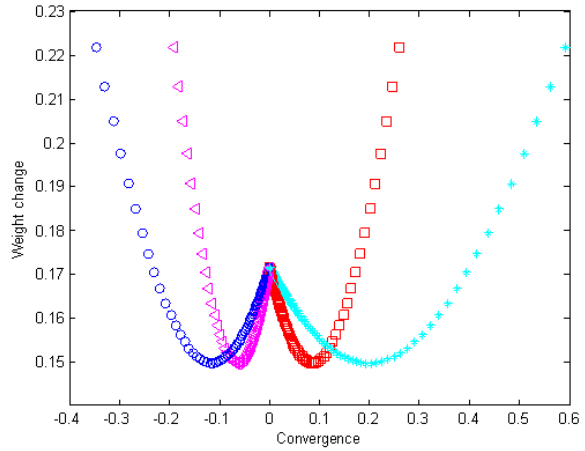


**Fig.3.(c) *iris verginica* (FCM with MH and 'm' = 12, DDF = 6.66%).**

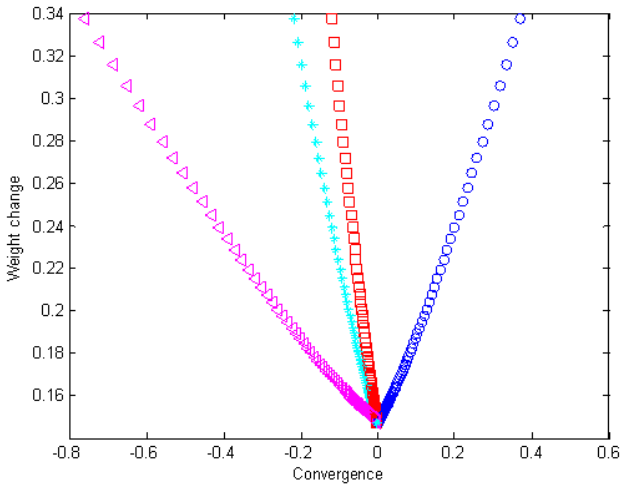


**Fig.4.(a) *iris setosa* (FkNN with MH and 'm' = 4.5).**

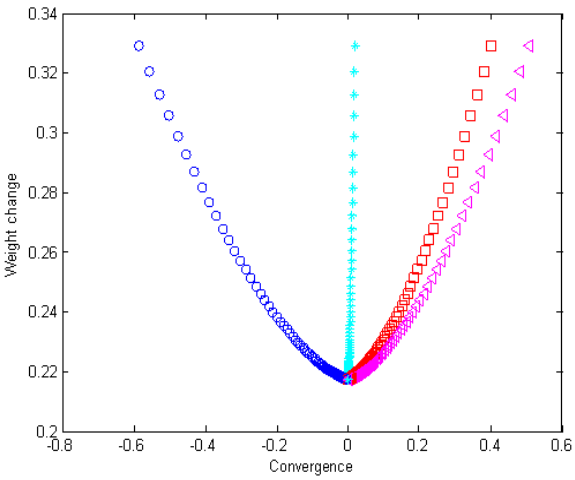


**Fig.4.(b) *iris versicolor* (FkNN with MH and 'm' = 4.5).**

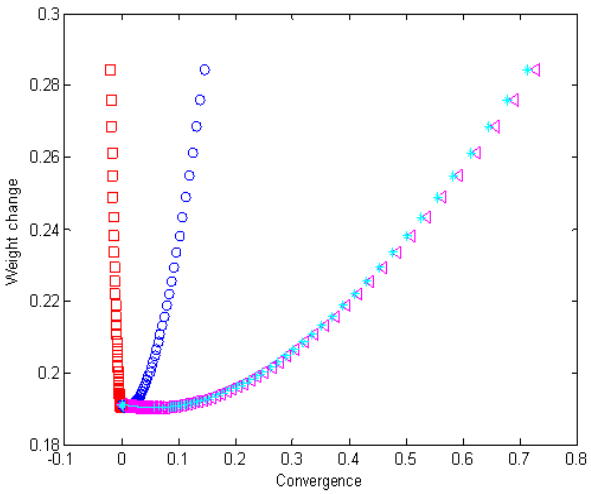


**Fig.4.(c) *iris verginica* (FkNN with MH and 'm' = 4.5).**

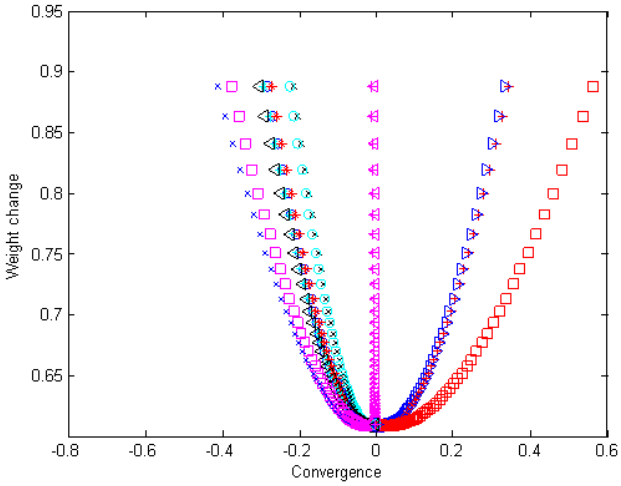


**Fig.5.(a) Wine cluster-1 (FCM with ED and 'm' = 1.5).**

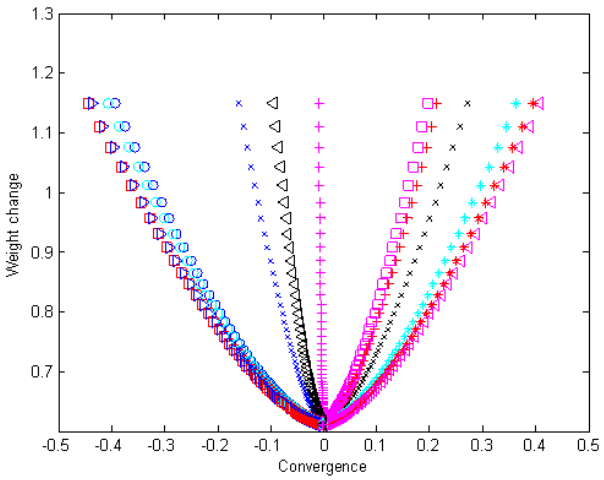


**Fig.5.(b) Wine cluster-2 (FCM with ED and 'm' = 1.5).**

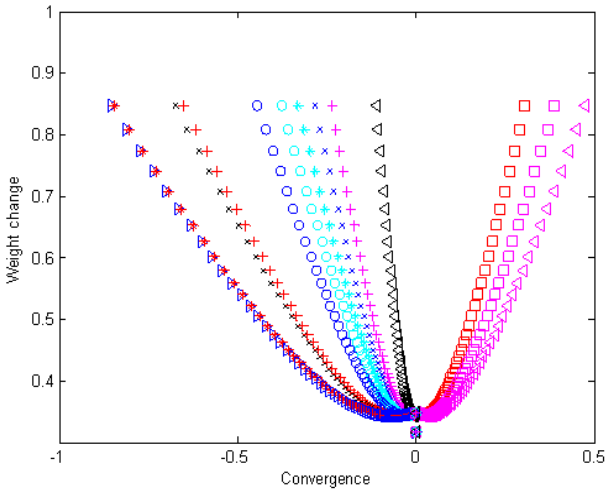


**Fig.5.(c) Wine cluster-3 (FCM with ED and 'm' = 1.5).**

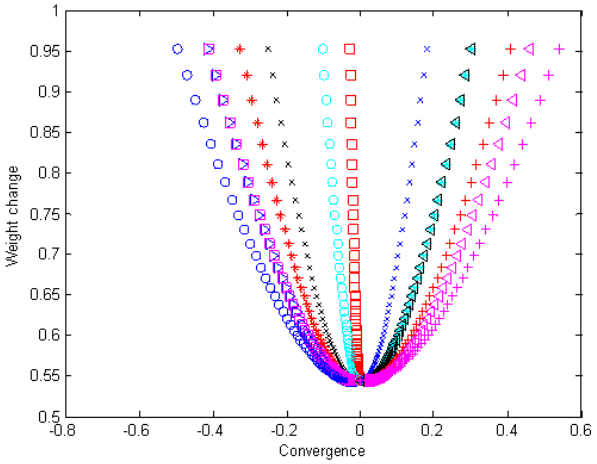


**Fig.6. (a) Wine cluster-1 (FkNN with MH and 'm' = 7.5).**

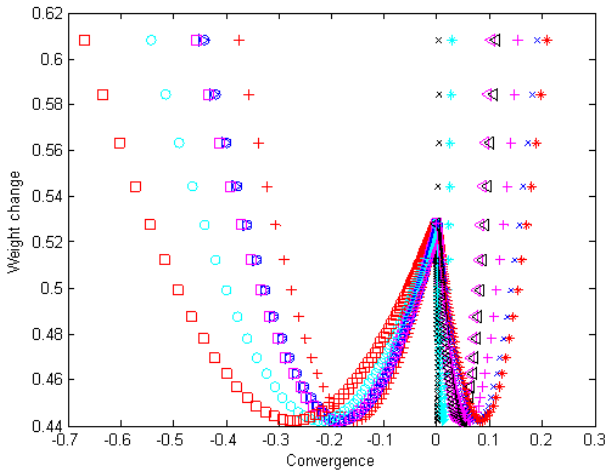


**Fig.6. (b) Wine cluster-2 (FkNN with MH and 'm' = 7.5).**

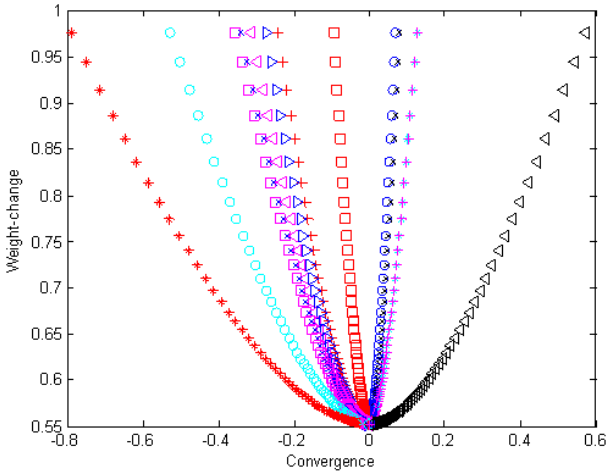


**Fig.6. (c) Wine cluster-3 (FkNN with MH and 'm' = 7.5).**

## 5. CONCLUSIONS AND FUTURE WORK

The paper at first investigates how different distance measures and the cluster fuzziness influence clustering performances of FCM and FkNN algorithms on Iris and extended Wine datasets. It then compares the performance based on (i) cluster size (CL_SIZE), (ii) DDF, (iii) cluster compactness (CL_CMP), (iv) inter cluster distance (CL_DST), and (v) Execution/run time (ALGO_RUN). Based on the experimental outcomes, the study concludes that,

- FCM produces best sized clusters on Iris and extended Wine data with MH and ED, respectively. Respective DDF values are 6.66% and 3.43%.
- Both FkNN and FCM produce compact clusters with MH and ED on Iris data. FkNN with MH produces most compact extended Wine clusters.
- On Iris data, FCM produces most distinct clusters with MH and ED. On extended Wine data, FCM with MH distance produces most distinct clusters.
- The fastest of all combinations is the FCM with MH (2.30e+09).
- FCM is able to handle maximum cluster fuzziness (m = 12) with MH distance measure on Iris data. On the other hand, on extended Wine FkNN can maximally handle the cluster fuzziness (m = 9.5) with ED and COS.
- Hence, FCM is found as a better fuzzy clustering approach, compared to FkNN.
- Finally, using SOM, best clusters could be successfully visualized.

The contribution of this study is the in-depth analysis of the clustering steps and visualization of the best clusters, obtained. The role of cluster fuzziness (the 'm' parameter) has been examined thoroughly on the clustering performance in terms of cluster size and DDF as well as the run time taken. It is seen that the relationship between '*m*' and the DDF is much non-linear as the algorithm tries to accommodate the fuzzy relationships while clustering with intermittent successes and failures (see figure 1(c)). From this example it may be inferred that fuzzy clustering algorithms can handle the said fuzziness up to a certain extent until it learns how to handle it further. Such information might be helpful to the researchers those use fuzzy clustering techniques on real-life complex data.

The limitation of the work is that the said algorithms might be tested further on real-life complex data, e.g. medical or business data. This constitutes author's future work.

## 6. REFERENCES

[1] MacQueen, J. B. (1967), *Some Methods for classification and Analysis of Multivariate Observations.* s.l. : University of California Press. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297.

[2] Theodoridis S., Koutroumbas K. (2006), *Pattern Recognition.* 3rd. s.l. : Elsevier, p. 635.

[3] Ester M., Kriegel H-P., Sander J., Xu X. (1996), *A density-based algorithm for discovering clusters in large spatial databases with noise.* . Han J., Fayyad U.M., Simoudis E. [ed.] s.l. : AAAI Press. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 226-231.

[4] Chattopadhyay S., Pratihar D. K., De Sarkar S. C. (2007), *Some studies on fuzzy clustering of psychoses data*. International Journal of Business Intelligence and Data Mining, Vol. 2, pp. 143-159.

[5] Bezdek J. C., Ehrlich R., Full W. (1984), *FCM: The fuzzy c-means clustering algorithm.* 2-3, Computers & Geosciences, Vol. 10, pp. 191-203 .

[6] Keller J. M., Gray M. R., Givens (jr.) J. A. (1985), *A fuzzy K-Nearest Neighbor Algorithm.*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, pp. 580-586.

[7] Yao J., Dash M., Tan S. T. (2000), *Entropy-based fuzzy clustering and fuzzy modeling.* Fuzzy Sets and Systems, Vol. 113, pp. 381-388.

[8] Dunn J. C. (1973), *A fuzzy relative of ISODATA process and its use in detecting compact well-separated clusters.* Journal of Cybernet, Vol. 3, pp. 32-57.

[9] Han J., Kamber M. *Data mining: concepts and techniques.* s.l. : Morgan Kaufmann, 2006.

[10] Chattopadhyay S., Ray P., Chen H.S., Lee M.B., Chiang H.C. (2008), *Suicidal risk evaluation using a similarity-based classifier.* Tang et al. [ed.] Chengdu China : Springer-Verlag Berlin Heidelberg, Advanced Data Mining and Applications (ADMA). pp. 51-61.

[11] Kohonen, T. *Self-organizing maps.* New York : Springer-Verlag , 1997. ISBN:3-540-62017-6.

[12] Tu D.C., Zhao J. H., Liu M. H., Shen J., Yu F. (2010) *Preliminary Study on Quantification of Duck Color Based on Fuzzy K – Nearest Neighbor Method.* Applied Mechanics and Materials, Vol. 39, pp. 210-215. DOI: 10.4028/www.scientific.net/AMM.39.210.

[13] Chen H-L., Liu D-Y., Yang B., Liu J., Wang G., Wang S-J. (2010), *An Adaptive Fuzzy k-Nearest Neighbor Method Based on Parallel Particle Swarm Optimization for Bankruptcy Prediction.* Cao L., Srivastava J. Huang J. Z. [ed.] s.l. : Springer-Verlag Berlin Heidelberg, 2011. PAKDD 2011, Part 1. LNAI 6634. pp. 249-264.

[14] Chen H-L., Yang B., Wang G., Liu J., Xu X., Wang S-J., Liu D-Y. (2010), *A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method.* Knowledge-Based Systems. DOI: doi:10.1016/j.knosys.2011.06.008.

[15] Arai Y., Lien N.T.H., Ishigaki K., Satoh H., Hayashi T., Dong F., Hirota K. (2010), *Fuzzy few-Nearest Neighbor Method with a Few Samples for Personal Authentication.* Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 14, pp. 167-178.

[16] Arif M., Akram M. U., Afsar F. A. (2009), *Arrhythmia Beat Classification Using Pruned Fuzzy K-Nearest Neighbor Classifier.* Malacca, Malaysia : IEEE Computer Society. International Conference of Soft Computing and Pattern Recognition. DOI:http://doi.ieeecomputersociety.org/10.1109/SoCPaR.2009.20.

[17] Wang H-M., Kim J-H, Jung D-Y., Lee S-M., Lee S-H. (2011), *Power interconnected system clustering with advanced fuzzy C-mean algorithm.* Journal of Central South

University of Technology, Vol. 18, pp. 190-195. DOI: 10.1007/s11771-011-0679-5.

[18] Wang H., Zhang Y., Li D. (2010), *Network intrusion detection based on hybrid Fuzzy C-mean clustering.* Yantai, Shandong : IEEE Xplore, 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). pp. 483 - 486.

DOI: 10.1109/FSKD.2010.5569762.

[19] Ramathilagam S., Pandiyarajan R., Sathya A., Devi R., Kannan S. R. (2011), *Modified fuzzy c-means algorithm for segmentation of T1-T2-weighted brain MRI.* 2011, Journal of Computational and Applied Mathematics, Vol. 235. DOI: 10.1016/j.cam.2010.08.033.

[20] Li Z. Y., Weng G. R. (2011), *Segmentation of cDNA Microarray Image Using Fuzzy c-Mean Algorithm and Mathematical Morphology*, Key Engineering Materials, Vol. 464, pp. 159-162.

[21] Chattopadhyay S., Pratihar D. K., De Sarkar S. C. *Performance studies of some similarity-based fuzzy clustering algorithms.* 2007, International journal of Performability Engineering, Vol. 2, pp. 191-200.

[22] Chattopadhyay S., Pratihar D. K., De Sarkar S. C. (2009), *Fuzzy logic-based Screening and Prediction of Adult Psychoses.* IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, Vol. 39, pp. 381-387.

[23] Chattopadhyay S., Pratihar D. K., De Sarkar S. C. (2008), *Developing Fuzzy Classifiers to Predict the Chance of Occurrence of Adult Psychoses*, Knowledge based Systems, Vol. 20, pp. 479-497.

[24] Fisher R. A. (1936), *The use of multiple measurements in taxonomic problems.*, Annals of Eugenics, Vol. 7, pp. 179-188.

[25] Forina M., Armanino C., Castino M., Ubigli M. (1990), *Chemometrical investigation on four red wines from a single cultivar grown in the Piedmont region.* Analyst, Vol. 115, pp. 907-910. DOI: 10.1039/AN9901500907.

[26] Dutta P., Pratihar D. K. *Some studies on mapping methods.* (2006), International Journal of Business Intelligence and Data Mining, Vol. 1(3), pp. 347-370.

[27] Panda S., Sahu S., Jena P.K., Chattopadhyay S. (2012), *Comparing Fuzzy-C means and K-means Clustering Techniques: a Comprehensive Study*. In Proceedings of 2nd International Conference on Computer Science, Engineering & Applications, by D.C. Wyld, J. Zizka, D. Nagamalai (Eds.), Advances in Intelligent and Soft Computing (AISC) Vol. 166, pp. 451-460. DOI: 10.1007/978-3-642-30157-5_45, 25-27 May, New Delhi India.