

# **PERFORMANCE ANALYSIS OF K-MEANS AND FCM ON SHAPE AND DENSITY VARYING CLUSTERS**

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

**DATA SCIENCE**

Submitted by:

**Aakash Verma**

**2K21/DSC/01**

Under the supervision of

**Dr. Sonika Dahiya**

**Assistant Professor**



**DEPARTMENT OF SOFTWARE ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

June, 2023

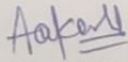
DEPARTMENT OF SOFTWARE ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi – 110042

**CANDIDATE'S DECLARATION**

I, Aakash Verma, Roll No. 2K21/DSC/01, student of Master of Technology (Data Science), hereby declare that the Major Project-II Dissertation titled "**Performance analysis of K-means and FCM on shape and density varying clusters**" which is submitted by me to the Department of Software Engineering, Delhi Technological University, Delhi in partial fulfillment of requirement for the award of degree of Master Of Technology (Software Engineering) is original and not copied from any source without proper citation. This work has not been previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 31-05-, 2023

  
Aakash Verma  
(2K21/DSC/01)

**DEPARTMENT OF SOFTWARE ENGINEERING  
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi – 110042**

**CERTIFICATE**

I hereby certify that the project entitled "**Performance analysis of K-means and FCM on shape and density varying clusters**" which is submitted by Aakash Verma (2K21/DSC/01) to Department of Software Engineering, Delhi Technological University, Shahbad Daultapur, Delhi in partial fulfilment of requirement for the award of the degree of Master of Technology in Data Science, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any degree or diploma to this university or elsewhere.

Place: Delhi

Date:                   , 2023

*Sonika  
31/05/2023*

**Dr. Sonika Dahiya**

**SUPERVISOR**

Assistant Professor

**Dept. of Software Engineering**

**DEPARTMENT OF SOFTWARE ENGINEERING**  
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Bawana Road, Delhi – 110042

**ACKNOWLEDGMENT**

I am very thankful to **Dr. Sonika Dahiya** (Assistant Professor, Department of Software Engineering) and all the faculty members of the Department of Software Engineering at DTU. They all provided us with immense support and guidance for the project. I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities, and environment which allowed us to work without any obstructions. I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

**AAKASH VERMA**  
**(2K21/DSC/01)**

## **ABSTRACT**

This study conducts a comprehensive performance analysis of K-means and Fuzzy C-means (FCM) clustering algorithms based on various distance metrics, including Euclidean, Manhattan, Mahalanobis, Minkowski, and Cosine distances. Clustering algorithms are essential tools for organizing data into meaningful groups. K-means and FCM are widely used algorithms in this context, with K-means focusing on crisp partitions and FCM providing fuzzy partitions. By employing different distance metrics, we explore how these algorithms perform under diverse similarity measures, capturing various aspects of data dissimilarity. Through extensive experimentation on benchmark datasets, we evaluate the clustering quality and computational efficiency of K-means and FCM algorithms using each distance metric. The evaluation metrics include intra-cluster distance, inter-cluster distance, silhouette coefficient, and clustering stability. Additionally, we analyze the runtime performance of the algorithms to assess their computational efficiency and scalability.

The results of our analysis provide valuable insights into the performance characteristics of K-means and FCM algorithms when applied with different distance metrics. We identify scenarios where one algorithm outperforms the other, shedding light on the suitability of each algorithm and distance metric combination for specific data characteristics and clustering objectives. This study contributes to the existing body of knowledge by offering a comprehensive comparison of K-means and FCM algorithms based on diverse distance metrics, including Euclidean, Manhattan, Mahalanobis, Minkowski, and Cosine distances. The findings of this analysis can guide researchers and practitioners in selecting the most suitable algorithm and distance metric combination for their clustering tasks, leading to improved clustering accuracy and efficiency.

# INDEX

<b>Content</b>	<b>Page Number</b>
<b>Declaration</b>	ii
<b>Certificate</b>	iii
<b>Acknowledgement</b>	iv
<b>Abstract</b>	v
<b>Index</b>	vi
<b>List of Figures</b>	viii
<b>List of Tables</b>	ix
<b>CHAPTER 1 INTRODUCTION</b>	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Objective	3
1.5 Thesis Structure	3
<b>CHAPTER 2 LITERATURE REVIEW</b>	5
2.1 Related Work	5
<b>CHAPTER 3 METHODOLOGY</b>	8
3.1 Clustering	8
3.2 Clustering Algorithm	9
3.2.1 K-means Algorithm	9
3.2.2 Fuzzy C-means Algorithm	10
3.3 Distance Measures	12
3.4 Dataset Description	18
<b>CHAPTER 4 EXPERIMENTAL ANALYSIS</b>	22
5.1 Circular and Elliptical Dataset Analysis	22
5.2 Spherical Dataset Analysis	26
5.3 Moon Dataset Analysis	27

<b>CHAPTER 5</b>	<b>DISCUSSION OF RESULTS</b>	28
6.1	Silhouette Analysis	29
6.2	Computation Efficiency Analysis	30
<b>CHAPTER 6</b>	<b>CONCLUSION AND FUTURE WORK</b>	31
<b>REFERENCES</b>		32

## LIST OF FIGURES

Figure 3. 1: 2-Dimensional Spherical Dataset .....	19
Figure 3. 2: 2-Dimensional Half and Full moon Dataset .....	19
Figure 3. 3: 2-Dimensional circular and elliptical Dataset .....	20
Figure 3. 4: 2-Dimensional circular and spherical Dataset .....	21
Figure 4. 1: Result of Clustering .....	23
Figure 4. 2: Result of Clustering .....	24
Figure 4. 3: Result of Clustering .....	25
Figure 4. 4: Result of Clustering .....	26
Figure 4. 5: Result of Clustering .....	27



## **LIST OF TABLES**

Table 4.1 : Silhouette Score comparison for K-means and FCM .....	29
Table 4.2: Computational time comparison for K-means and FCM.....	30

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

Clustering algorithms are heavily used in data analysis to find important groupings and trends within datasets. K-means and fuzzy C-means (FCM), two well-known clustering algorithms, have received a lot of attention due to their effectiveness and simplicity [1]. On their performance characteristics or applicability for datasets containing clusters of various sizes and densities, however, little research has been done. Clusters in real-world datasets frequently differ in their density distributions as well as their spatial layouts [2]. While some clusters may have consistent densities, others may exhibit substantial variations or overlap with other clusters. Additionally, clusters can have spherical, elongated, or irregular structures, among other shapes. These variances present difficulties for clustering algorithms, necessitating a thorough performance evaluation to comprehend their behavior and constraints. The performance of the K-means and FCM algorithms on datasets made up of clusters with various forms and densities is the main objective of this work [3]. We seek to provide insights into the strengths and shortcomings of these algorithms in difficult clustering scenarios by assessing their effectiveness in capturing and distinguishing clusters with varied features.

Shape variation describes the various geometric forms and arrangements that clusters can display inside a dataset[4]. Clusters can be round, elongated, or irregular, among other shapes. While elongated clusters have a stretched shape along one or more dimensions, spherical clusters are characterized by an approximately equal distance between data points and a central point. The boundaries of irregular clusters can be complicated and tangled and lack a clear geometric structure. The usage of clustering algorithms capable of accurately capturing and differentiating diverse cluster shapes is required due to the occurrence of different cluster shapes.

The term "density variation" describes variations in the distribution or concentration of data points inside clusters [5]. Data points can be evenly distributed throughout a cluster if it has uniform density. Alternatively, clusters might have changing densities, in which the distribution of data points varies over the cluster's various sections. When two or more clusters overlap, they share common regions or data points, which makes it more difficult to assign certain data points

to certain clusters. Clustering techniques have difficulties due to density variation since they must be flexible enough to tolerate different amounts of data point concentration within clusters. Clustering algorithms that can effectively handle clusters with diverse shapes and densities must be able to recognise the unique properties and structures of the data. Since they tend to generate spherical clusters around centroids, distance-based algorithms like K-means may struggle with extended or irregular clusters. In order to accommodate clusters with different densities and forms, fuzzy clustering techniques, like Fuzzy C-means, allow data points to have partial memberships in several clusters.

Determining the best technique for a given application requires a thorough understanding of the behavior and constraints of clustering algorithms on datasets with clusters of different density and shape. It also makes it possible to create brand-new algorithms or improve upon existing ones so they can better tackle these difficulties [6]. In order to shed light on the strengths, limitations, and applicability of various clustering scenarios, this study seeks to give a thorough performance analysis of clustering algorithms on datasets with shape and density variable clusters.

## **1.2 Motivation**

Real-world datasets frequently include clusters of various densities and forms. These variances can appear in a variety of applications, including anomaly detection, customer segmentation, and picture analysis. Gaining accurate and trustworthy results from clustering algorithms requires an understanding of how they operate on such complicated datasets. The popular clustering algorithms K-means and FCM are known for being straightforward and efficient. However, little research has been done on how they work with clusters of varied shapes and densities. These algorithms' strengths, weaknesses, and applicability for various clustering scenarios will be assessed using datasets with a variety of cluster characteristics. Algorithmic improvements may result from examining how K-means and FCM perform on clusters with different cluster shapes and densities. Researchers can create new algorithms or change existing ones to better effectively address these difficulties by understanding the constraints and weaknesses of these algorithms in particular clustering settings. This performance analysis may act as a spark for additional study and advancements in the clustering analysis discipline. By focusing on clusters that change in shape and density, this performance analysis adds to the body of knowledge on clustering

methods. Future research and comparisons involving different clustering algorithms and datasets with comparable properties might use the findings and understandings from this analysis as a guide and benchmark.

### **1.3 Problem Statement**

Data analysis requires clustering as a fundamental step, yet real-world datasets frequently have clusters of different sizes and densities. Although the K-means and FCM algorithms are frequently used for clustering, little research has been done on how well they function with clusters that vary in shape and density. Therefore, The performance analysis aims to address the following research questions:

- How well do the K-means and FCM algorithms perform on datasets that have irregular, spherical, and elongated clusters?
- Which method performs better at capturing clusters with various shapes and densities, K-means or FCM?
- Which distance metrics, such as Euclidean, Manhattan, Mahalanobis, Minkowski, and cosine, produce the best results for each algorithm on clusters with a variety of shapes and densities?
- How do the algorithms deal with clusters that have different densities, such as uniform, variable, and overlapping densities?

### **1.4 Objective**

The objective is to learn more about the benefits, drawbacks, and restrictions of the K-means and FCM algorithms when used on datasets having irregular clusters in terms of shape and density. These research topics will be addressed by this performance analysis, which will offer recommendations for choosing the best algorithm for particular clustering tasks and datasets with different cluster characteristics.

### **1.5 Thesis Structure**

Chapter 1 briefly takes you through the introduction of the topic thereafter section 1.2 shows the motivation behind the project, section 1.3 gives the objectives of the project. Chapter 2 takes you

through the summary of previously related studies. Chapter 3 gives the complete information related to implementation & methodology, it also explains the different clustering algorithms, different types of distance metrics and dataset description. Chapter 4 shows the experimental analysis that has been gathered and evaluation of those results is described in Chapter 5. Chapter 6 gives the conclusion on the complete idea of the topics and results; it also discusses the future scopes related to the idea. Finally, some references that have been taken are mentioned.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Related work

1. The K-means algorithm, which tries to reduce the within-cluster sum of squared distances, is briefly described at the beginning of the paper [7]. Although it is often used in K-means, the Euclidean distance metric may not always be appropriate for all sorts of data. As a result, the author examines how K-means performs with the following distance metrics: Euclidean, Manhattan, Minkowski, and Mahalanobis. The outcomes demonstrate that the selection of the distance metric has a significant impact on the K-means performance. When the clusters are close together and well-separated, the Euclidean distance performs well. It suffers, nevertheless, with datasets that contain overlapping or unevenly sized clusters. In these circumstances, the Manhattan distance, which computes the sum of absolute differences, performs better. It is more resistant to outliers and functions well with irregularly distributed data.
2. Similar data or objects are gathered into one group through a type of unsupervised learning called clustering. The display of these items is possible in an n-dimensional Euclidean space [8]. The performance of K-means and FCM, two distinct clustering approaches, is reviewed and examined in this work. These algorithms are used on a dataset that includes information on patients who have undergone surgery for breast cancer. When K-means and FCM are used with various distance measurements, it is discovered that classification accuracy, efficiency, and precision are all negatively impacted. When used with Manhattan, the K-means algorithm is shown to be more effective in clustering, whereas FCM performs better when Euclidean distance is used.
3. This paper compares the impacts of several distance functions used in k-means clustering and studies their effects. One method that has been suggested for application in the field of data mining is clustering [9]. The idea behind clustering is to group things according to

some common qualities so that they are comparable to one another but not to objects in other clusters.

4. In "Distance Metrics and Clustering Methods for Mixed-type Data," Alexander Foss and colleagues (2019) revealed that they used both theoretical and empirical analysis to determine the most successful methods for clustering mixed-type data [10]. The team presents a critical analysis of the benefits and drawbacks of the strategies mentioned in the literature using both theoretical and empirical evaluations. There are suggestions for future research areas as well as guidelines on how to approach various scenarios. Data clustering for mixed interval and categorical scales is still a difficult problem. They suggest that to find the best methods for clustering mixed-type data, they should combine theoretical and empirical analyses. There are suggestions for future research areas as well as guidelines on how to approach various scenarios.
5. The technique of categorizing a collection of tangible or intangible objects into various groups of related objects is known as cluster analysis. The key to solving the clustering problem is determining the optimal classification number of a data set, or whether it can be partitioned successfully. By adopting the division of intra-class compactness and inter-class separation, whose minimum reflects the best clustering, a validity function of the fuzzy C-means (FCM) clustering algorithm is proposed [11]. Through the use of simulation experiments, the suggested validity function for the FCM clustering algorithm is compared to the well-known usual validity functions. The conclusions demonstrate that the suggested validity function may successfully divide the data set.
6. Title The k-means Algorithm: A Comprehensive Survey and Performance Evaluation  
Performance comparisons were undertaken in terms of the aforementioned metrics for experimental analysis using k-means, x-means, limited k-means, k-prototype, and kernel k-means. With the KDD Cup datasets, k-means fared the best while x-means had the lowest accuracy. In the Wisconsin dataset, constrained-k-means performed best. The constrained-k-means appeared to perform consistently in terms of ARI score [12]. These algorithms are appropriate for datasets with varied attribute compositions. Using all three

datasets with mixed data, the kernel-k-means algorithm consistently outperformed k-prototype in terms of ARI score comparison.

7. In this study, Author presents a clustering approach that can effectively handle clusters of different densities. The DBSCAN method underwent a straightforward change by the approach. Calculating the maximum density permitted within each cluster and using Minpts to regulate the lowest density permitted within each cluster are the objectives. The maximum density in each cluster is calculated using the k-nearest neighbour approach, and the density of the  $i$ th neighbour is used to determine whether or not a point can be allocated to the current cluster. The  $i$ th neighbor's density must be larger than or equal to Minpts. According to the experimental data shown here[13], the suggested approach is more effective than the DBSCAN algorithm in handling datasets with different densities of clusters. The suggested algorithm has a temporal complexity of  $O(n \log n)$ , the same as the DBSCAN algorithm.
8. The number of clusters must be supplied into the minimal spanning tree (MST)-based clustering method known as LDPMST. To extract clusters of any shape, we suggest using a user-input-free density-based clustering approach termed UIFDBC. We offer a closest neighbor variable density clustering (NNVDC) technique that can identify various densities, forms, and sizes despite noise [14]. The findings demonstrate that, in terms of finding arbitrary shaped clusters on tested datasets, our strategy is generally superior to those of its competitors. The study in this paper focuses on the (EBRBS), which demonstrated excellent robustness in the event of sensor failure and shown promising performance when compared to common benchmark activity recognition (AR) models.



## CHAPTER 3

### METHODOLOGY

#### 3.1 Clustering

Unsupervised machine learning fundamentally uses clustering to look for hidden patterns or structures in data [15]. Without using predetermined class labels or target variables, it entails assembling related objects or data points based on their intrinsic similarities. Data is divided into clusters using clustering algorithms, with each cluster representing a collection of related data points. The main objective of clustering is to maximize intra-cluster similarity while minimizing inter-cluster similarity. In other words, compared to objects in separate clusters, objects in the same cluster should be more similar to one another. Clustering algorithms help find significant subsets or categories within the data by locating clusters, which can yield insightful information and make various types of data analysis easier. Customer segmentation, picture and document analysis, anomaly detection, social network analysis, and recommendation systems are just a few of the domains where clustering is used. It can support decision-making processes, identify groups with similar behaviors or preferences, and reveal hidden structures in large datasets.

There are various clustering algorithms, each having unique traits and presumptions. K-means, hierarchical clustering, density-based clustering (like DBSCAN), and fuzzy clustering (like Fuzzy C-means) are some of the most widely used methods[16]. Different techniques are used by these algorithms to establish clusters and determine how similar or dissimilar data points are.

Due to the lack of labeled data and the arbitrary nature of determining what makes a "good" cluster, clustering is a difficult operation. A clustering algorithm's performance is influenced by a variety of elements, including the choice of distance or similarity measures[17], the algorithm's scalability, the right choice of cluster size, and the capability to handle various data kinds and cluster forms.

## 3.2 Clustering Algorithms

Unsupervised machine learning techniques called clustering algorithms are used to put comparable data points together based on underlying patterns or similarities. Without using labels or goal values that have already been established, these algorithms seek to identify the underlying structure in the data.

### 3.2.1 K-means Algorithm

One of the most popular clustering methods is the K-means algorithm. Data is split into K clusters based on similarity using this quick and effective unsupervised learning technique[18]. The algorithm's goal is to reduce the sum of squared distances between data points and the cluster centroids that are allocated to them.

---

#### Algorithm 1 Pseudo code of K-means Algorithm

---

1. **Select** the number of clusters, K.
  2. **Initialize** the centroids randomly or using a specific initialization method.
  3. **Repeat until convergence:**
    - a. Assign each data point to the nearest centroid:
      - Compute the distance between each data point and all centroids.
      - Assign each data point to the centroid with the minimum distance.
    - b. **Update** the centroids:
      - For each centroid, compute the mean of all data points assigned to it.
      - Update the centroid coordinates to the computed means.
  4. **Return** the final centroids and the cluster assignments.
- 

The K-means algorithm has a number of crucial characteristics:

The clustering goal function, which is the sum of the squared distances between the data points and the assigned centroids, is optimized by an iterative process. The algorithm is sensitive to the original centroid's location and may not always produce outcomes that are globally optimal. The selection of K (the number of clusters) is critical, and figuring out the best value frequently involves subject expertise or evaluation measures. K-means can handle big datasets and is computationally efficient.

K-means is frequently utilized in many different applications, such as data preparation, document clustering, image compression, and consumer segmentation. When used on datasets with well-separated, spherical-shaped clusters, the technique can give effective clustering results despite its simplicity. K-means has restrictions, though. It is less suited for datasets with varied cluster forms or fluctuating cluster densities since it assumes that clusters are spherical and of equal size. Additionally, it is susceptible to outliers, and noise or overlapping clusters may impair its performance. Variations of the K-means approach, such as K-means++, which enhances centroid initialization, and K-medoids, which employs medoids rather than centroids to accommodate non-Euclidean distance metrics, have been created to get around some of these drawbacks. A well-liked and effective clustering method is the K-means algorithm, which divides data into K clusters based on the minimizing of distances between data points and cluster centroids.

### 3.2.2 Fuzzy C-means Algorithm

The K-means technique is extended to meet the needs of fuzzy or probabilistic clustering by the Fuzzy C-means (FCM) algorithm, a popular clustering algorithm[19]. By allowing data points to have various degrees of participation in numerous clusters, FCM offers a more adaptable and nuanced depiction of cluster membership.

---

#### Algorithm 2 Pseudo code of Fuzzy C-means Algorithm

---

**1. Initialize:**

- Set the number of clusters, K.
- Set the fuzziness coefficient, m.
- Set the maximum number of iterations, max\_iter.
- Initialize the membership matrix, U, randomly or using a specific initialization method.

**2. Repeat until convergence** or reaching the maximum number of iterations:

a. **Update** the cluster centroids:

- For each cluster, calculate the centroid as the weighted mean of data points using membership values:

$centroid_k = \frac{\sum(u_{ik}^m * x_i)}{\sum(u_{ik}^m)}$ , for each data point  $x_i$  assigned to cluster k.

b. **Update** the membership matrix:

- For each data point, calculate the membership values for each cluster using the Euclidean distance:

$u_{ik} = 1 / \sum((\text{dist}(x_i, \text{centroid}_k) / \text{dist}(x_i, \text{centroid}_j)) ^ (2 / (m-1))),$  for each cluster k and j.

c. **Check** for convergence by calculating the objective function or comparing the change in membership matrix.

d. **Increment** the iteration counter.

3. **Return** the final cluster centroids and the membership matrix.

---

The FCM algorithm has a number of significant characteristics:

Soft clustering is possible with it, allowing data points to partially belong to several clusters. Compared to strict clustering algorithms like K-means, FCM offers a more flexible representation of cluster membership. The algorithm seeks to maximise membership degrees while balancing the maximisation of membership degrees and the minimising of distances between data points and cluster centroids. Numerous domains, including pattern recognition, picture segmentation, bio informatics and data mining, have used FCM extensively. It is especially helpful when working with datasets where individual data points may have conflicting or redundant cluster allocations.

FCM does, however, have its limitations. The choice of the fuzziness parameter (usually represented as "m") impacts the fuzziness of the clustering results, and it is sensitive to the initial fuzzy membership values. It could involve a lot of processing and necessitate careful parameter optimisation for best results. The K-means technique can be extended to provide fuzzy or probabilistic clustering using the Fuzzy C-means (FCM) algorithm. It offers a more adaptable and nuanced depiction of cluster membership, allowing data points to to some extent belong to numerous clusters. When dealing with unclear or overlapping cluster assignments, FCM provides a useful substitute to conventional hard clustering methods that are widely employed in a variety of applications.

### 3.3 Distance Metrics

Distance measures are important because they help clustering algorithms assess how similar or different data points are [20]. Different distance measures use different methods to quantify distance or dissimilarity, which causes cluster formation to vary. The following list of frequent distance measures for clustering:

#### 1. Euclidean Distance:

The Euclidean distance formula[21], which can be written as follows, is used to compute it:

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2} \quad \dots\dots\dots(1)$$

- $d(x, y)$  represents the Euclidean distance between points  $x$  and  $y$ .
- $x_i$  and  $y_i$  are the values of the  $i$ -th dimension of points  $x$  and  $y$ , respectively.
- $\sum$  represents the sum over all dimensions.

#### 2. Manhattan Distance:

The formula for the Manhattan distance, sometimes referred to as the City Block or L1 distance, is as follows:

$$d(x, y) = \sum |x_i - y_i| \quad \dots\dots\dots(2)$$

The Manhattan distance computes the total of the absolute differences between the two points' corresponding dimensions[22]. By adding up the absolute differences in the coordinates of the two points along each dimension, it calculates the distance between them. The streets in Manhattan are laid out in a grid-like pattern, and the shortest path between any two points runs along the city blocks, hence the name "Manhattan distance."

Applications including image processing, classification, and clustering frequently employ the Manhattan distance. When the data has a grid-like or block structure, it is extremely helpful. The Manhattan distance is distinct from the Euclidean distance in that it only takes into account the distance travelled along the axis, not the direct line connecting the two places. It is a metric that satisfies the triangle inequality and can be applied to situations in which various dimensions have distinct scales or units.

### 3. Mahalanobis Distance

The Mahalanobis distance formula is as follows:

$$d(x, y) = \sqrt{(x - y)'S^{-1}(x - y)} \quad \dots\dots(3)$$

When computing the distance between two points, the Mahalanobis distance takes the data's correlation and variance into account[23]. It calculates the separation between two points by taking the data distribution's shape and direction into account. The Mahalanobis distance can take into consideration the various scales and correlations of the data along each dimension by incorporating the covariance matrix.

When working with data that has correlated characteristics or when there are distinct scales along different dimensions, the Mahalanobis distance is quite helpful. Applications like anomaly detection, clustering, and classification frequently employ it. Outliers or anomalous data points that significantly vary from the data's normal distribution can be found using the Mahalanobis distance.

It's crucial to remember that the covariance matrix  $S$  must be positive definite in order to calculate the Mahalanobis distance. Computational problems could result from a unique

or poorly-conditioned covariance matrix. Alternative distance metrics or regularisation approaches may be used in such circumstances.

#### 4. Minkowski Distance:

The equation for Minkowski distance is as follows:

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{(1/p)} \quad \dots\dots(4)$$

The Minkowski distance is a generalised distance metric with special cases for a number of different distance measurements[24]. The Manhattan distance (L1 norm) is what the Minkowski distance reduces to when  $p = 1$ , and the Euclidean distance (L2 norm) is what it reduces to when  $p = 2$ . Different amounts of focus can be placed on the distinct dimensions while calculating the distance by changing the value of  $p$ .

In many different applications, including clustering, classification, and regression, the Minkowski distance is frequently utilised. By modifying the value of  $p$  in accordance with the features of the data or the particular needs of the study, it provides for flexibility in calculating distances. When working with data, for instance, that has many scales or dimensions and varied levels of relevance, To account for these factors, the Minkowski distance with a suitable value of  $p$  can be chosen.

#### 5. Chebyshev Distance

The following is the equation for the Chebyshev distance, often known as the maximal norm:

$$d(x, y) = \max(|x_i - y_i|) \quad \dots\dots(5)$$

The biggest absolute difference between two points' related dimensions is measured by the Chebyshev distance[25]. Instead of adding or average the discrepancies, it determines the distance by taking into account the biggest difference along any dimension. In a multidimensional space, it represents the length of the longest feasible straight line between two points.

When working with data that has distinct scales or when the dimensions have variable levels of relevance, the Chebyshev distance is especially helpful. It is frequently employed in applications including decision-making algorithms, image processing, and pattern recognition. In situations where outliers or extreme values are of interest, the Chebyshev distance offers a metric that is sensitive to the largest difference in any dimension.

#### 6. Hamming Distance:

The equation for Hamming distance is as follows:

$$d(x, y) = \sum(x_i \neq y_i) \quad \dots\dots(6)$$

A metric for determining the difference between two binary vectors of identical length is the Hamming distance[26]. It counts the number of locations where the related vector elements diverge. In other words, it keeps track of how many bits must be flipped in order to change one vector into another.

The Hamming distance is frequently utilised in a variety of applications, including binary pattern matching, DNA sequence analysis, and error detection and repair. When working with category or binary data, where the presence or absence of specific attributes is of importance, it is especially useful. It's crucial to remember that the Hamming distance is only appropriate for vectors of the same length and was created primarily for binary or categorical data. It does not consider the magnitude or intensity of differences between elements.

#### 7. Cosine Distance:

The following is the equation for cosine distance, commonly referred to as cosine similarity:

$$d(x, y) = 1 - (\text{cosine\_similarity}(x, y)) \quad \dots\dots(7)$$

To calculate the cosine similarity, the equation is as follows:

$$\text{cosine\_similarity}(x, y) = (x \cdot y) / (||x|| \cdot ||y||) \quad \dots\dots(8)$$



Based on the cosine of the angle between two vectors, the cosine distance calculates how dissimilar they are[27]. It is frequently used to establish if two documents, text corpora, or sets of highly dimensional data are comparable or dissimilar. The cosine similarity scales from -1 to 1, with 1 denoting that the vectors are exactly the same and -1 denoting that they are completely unrelated.

In situations where the magnitude or scale of the vectors is unimportant, the cosine distance is useful. When working with high-dimensional data, where the vector representation represents the presence or absence of specific attributes, it is especially helpful. The cosine distance measures the direction of the vectors rather than their magnitude, in contrast to other distance measurements. It's crucial to remember that the cosine distance is not a standard geometric measure of distance but rather a measure of dissimilarity. When performing tasks like clustering, classification, or information retrieval, the cosine distance is frequently used.

#### 8. Canberra distance:

The equation for Canberra distance is as follows:

$$d(x, y) = \sum(|x_i - y_i| / (|x_i| + |y_i|)) \quad \dots\dots(9)$$

Calculating the distance between two vectors using the Canberra distance is a common task. By accounting for the size of each element, it calculates the normalized sum of absolute differences between comparable vector elements. where working with data that has different scales or where the ratio between numbers is crucial, it is especially helpful. In applications like pattern recognition, text mining, and information retrieval, the Canberra distance is frequently employed[28]. When comparing sparse or high-dimensional data, where the presence or absence of specific traits might have a big impact, it is especially well suited.

It is crucial to remember that the vector zeroes have an impact on the Canberra distance. For a specific element, the contribution to the distance calculation is zero if both  $x_i$  and  $y_i$  are zero. The Canberra distance works well for representations of sparse data because of this property. It is important to note that the Canberra distance might be asymmetric and impacted by outliers, which means that  $d(x, y)$  may not always be equal to  $d(y, x)$ .

Because of this, it is crucial to take these things into account when employing the Canberra distance for data analysis activities.

9. Jaccard distance:

The equation for Jaccard distance is as follows:

$$d(x, y) = 1 - (|x \cap y| / |x \cup y|) \quad \dots\dots(10)$$

An indicator used to determine how diverse two sets are is the Jaccard distance. As the Jaccard similarity coefficient's complement, it quantifies dissimilarity[29]. The ratio of the sizes of two sets' intersection and union is known as the Jaccard similarity coefficient. Applications like data mining, information retrieval, and pattern recognition frequently use the Jaccard distance. Comparing sets that reflect categorical or binary data makes use of it particularly well. The Jaccard distance spans from 0 to 1, with 0 denoting perfect similarity between the sets and 1 denoting total dissimilarity. The Jaccard distance is symmetric, i.e.,  $d(x, y) = d(y, x)$ . This is significant. It is frequently used to evaluate the dissimilarity between collections of objects or characteristics in clustering, classification, and recommendation systems.

### 3.4 Dataset Description

A synthetic dataset is one that has been manufactured artificially and is based on rules or algorithms rather than being gathered from actual observations. To investigate and assess methods, models, and systems, synthetic datasets are frequently used in a variety of domains, including machine learning, data analysis, and computer simulations. In order to recreate particular settings or occurrences that might not be readily available or apparent in real-world data, synthetic datasets are produced. They give researchers the ability to regulate and shape many facets of the data generating process to meet their study goals.

Models, methods, and simulations in mathematics can be used to create synthetic datasets. These techniques specify the links, patterns, or rules found in the data and create samples in accordance with those findings. To evaluate the effectiveness of clustering algorithms, for instance, datasets with well-known cluster structures may be produced in clustering experiments. Different traits, including cluster shapes, density fluctuations, noise levels, and class imbalances, can be seen in synthetic datasets. Specific scenarios, such as linearly separable or overlapping clusters, skewed distributions, or outliers, can be represented by these features.

In this report we used Synthetic datasets made to imitate data points scattered over or around a spherical form in two dimensions are referred to as two-dimensional spherical datasets. These datasets are frequently used to test algorithms in fields where spherical structures are important, including as data visualisation, clustering, and pattern recognition. Clusters that are shaped like spheres or have a spherical organisation can be included in two-dimensional spherical datasets. These clusters might have different sizes and densities, as well as overlap or not overlap. Clustering algorithms' capacity to recognise and distinguish spherical clusters can be evaluated using such datasets. Different spherical shapes, such as whole, partial, or overlapping spheres, can be seen in two-dimensional spherical datasets. The dimensions, radii, and orientation of these shapes can change. To develop customised spherical structures that meet their experimental demands, researchers can adjust these properties.

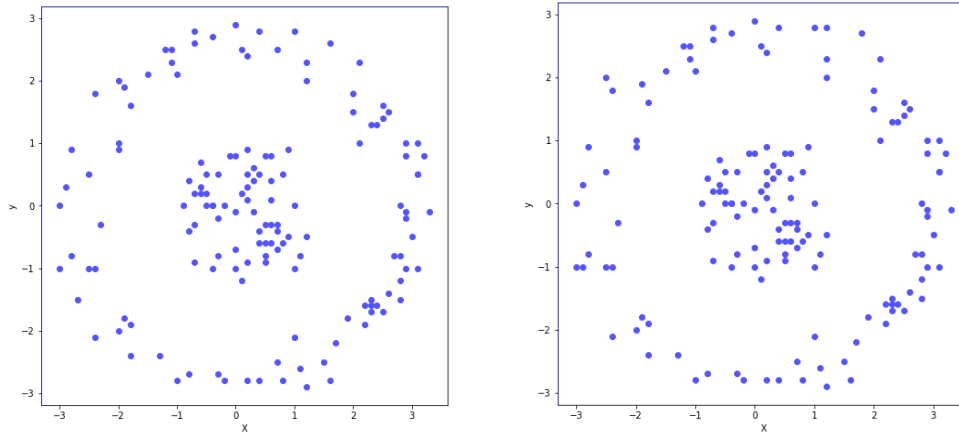


Fig.1 2-Dimensional Spherical Dataset

The term "half and full moon dataset generation in 2D" describes the process of creating artificial datasets with the appearance of a half moon or a full moon in two dimensions. These datasets are frequently used in pattern recognition and machine learning applications to assess how well algorithms handle curved or non-linear data distributions. The data points are arranged in a half-moon pattern in a half-moon dataset. The half moon's two sides each symbolise a separate class or cluster, with one side representing one. Given that the two classes are frequently not linearly separable, this dataset presents difficulties for classification tasks. A full moon dataset has data points that are arranged in a circular or spherical pattern similar to a full moon. Concentric circles or spheres that each represent a different class or cluster can make up this dataset. Full moon datasets are frequently used to test clustering algorithms and find spherical or circular patterns.

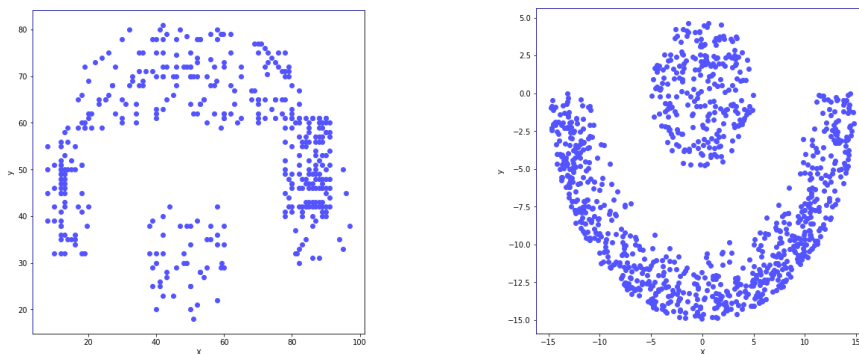


Fig.2 2-Dimensional Half and Full Moon Dataset

Elliptical 2D datasets are made up of data points that are arranged in an elliptical pattern in a two-dimensional space. These datasets are frequently used in pattern recognition, data analysis, and machine learning to assess how well algorithms handle data distributions with elliptical features. The geometry of elliptical 2D datasets is elliptical and can vary in size, aspect ratio (the ratio of major to minor axis lengths), and orientation. In the two-dimensional space, the elliptical shape can be rotated at an angle or lined up with the axes.

Elliptical-shaped data point clusters may be seen in synthetic datasets. The shapes, densities, and orientations of these clusters might vary. Clustering algorithms' capacity to recognise and distinguish elliptical clusters is evaluated using elliptical datasets.

Different levels of overlapping and separability across clusters or classes might be present in datasets. The degree to which data points from various clusters overlap or share a common location is referred to as overlapping. The term "separability" describes how easily an algorithm can separate the clusters.

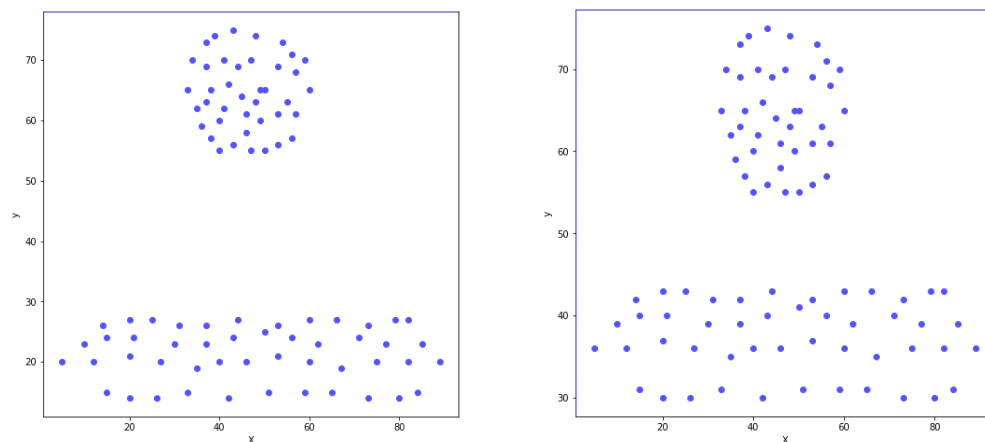


Fig.3 2-Dimensional Circular and Elliptical Dataset

Circular shaped 2D datasets are artificial datasets with a circular distribution of data points in a two-dimensional environment. These datasets are frequently used to assess how well algorithms handle data distributions with circular shapes in machine learning, data analysis, and pattern

recognition. The data points in circular-shaped 2D datasets are dispersed evenly across the circle's perimeter or inside of it, giving the datasets a perfectly round shape. Depending on the specified radius, the circle's size can change.

Circular-shaped data point clusters may be seen in synthetic datasets. Within the circular region, these clusters might be of different shapes, densities, and orientations. Clustering algorithms' capacity to recognise and distinguish circular clusters is evaluated using circular datasets. Different degrees of overlapping and separability between clusters or classes can be found in circularly formed datasets. The degree to which data points from various clusters overlap or share a common location is referred to as overlapping. The term "separability" describes how easily an algorithm can separate the clusters. Synthetic datasets can imitate real-world situations by including noise and outliers. Outliers are data points that dramatically differ from the majority of data points, whereas noise refers to random fluctuations or inaccuracies in data points. Outliers and noise increase the dataset's complexity and put algorithms' robustness to the test.

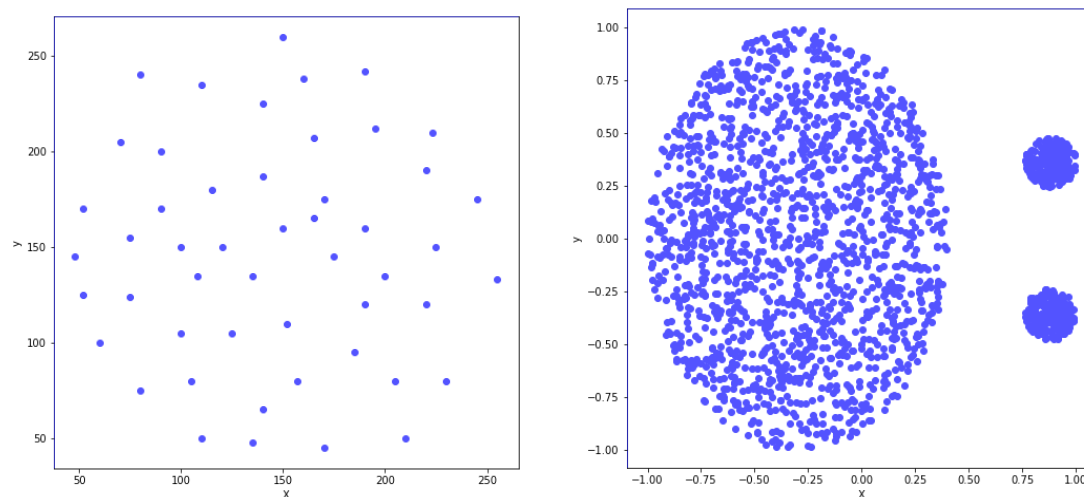


Fig.4 2-Dimensional Circular And Spherical Dataset

## CHAPTER 4

### EXPERIMENTAL RESULTS

The effectiveness of the two clustering techniques along with four distance metrics Kmeans and FCM with Euclidean, manhattan, minkowski, canberra distance metrics is analyzed in this section. The tolerance of the two-dimensional fabricated data sets is first examined. four clustering techniques with varied data density, nonlinearity, and form. The effectiveness of the clustering process is then evaluated using real test data sets with known data labels that were gathered from open databases. The results of the clustering are then visualized using the clustering techniques. All clustering techniques are used with the same beginning weight values for wic and termination criteria for every test data set.

#### 4.1 Simulated Data

In order for the real data points in each cluster and the clustering outcomes to be visually examined and validated, artificial data are manufactured in a two-dimensional plane. First, the impact of altered distances between two clusters is assessed. Simulated clusters in Fig. 1 include an elliptic cluster and a circular cluster. These two clusters' separation is expressed as

$$R_{distance} = \frac{D_i}{r_A + \lambda_B} \quad \dots\dots(11)$$

where  $D_i$  is the separation between the two clusters' centroids. The semiminor axis of the elliptic cluster is denoted by  $\lambda_B$ , and the radius of the circular cluster is denoted by  $r_A$ .

Two circular clusters (A and B) with varying densities are simulated in order to assess the impact of changes in cluster density. 50 data points are located inside a set 12-point radius in Cluster A. B Cluster identically distributed data points with different radiuses larger than 12. The ratio of density for two clusters is described by

$$R_{density} = \frac{N_B/r_B}{N_A/r_A} = \frac{r_A}{r_B} \quad \dots\dots(12)$$

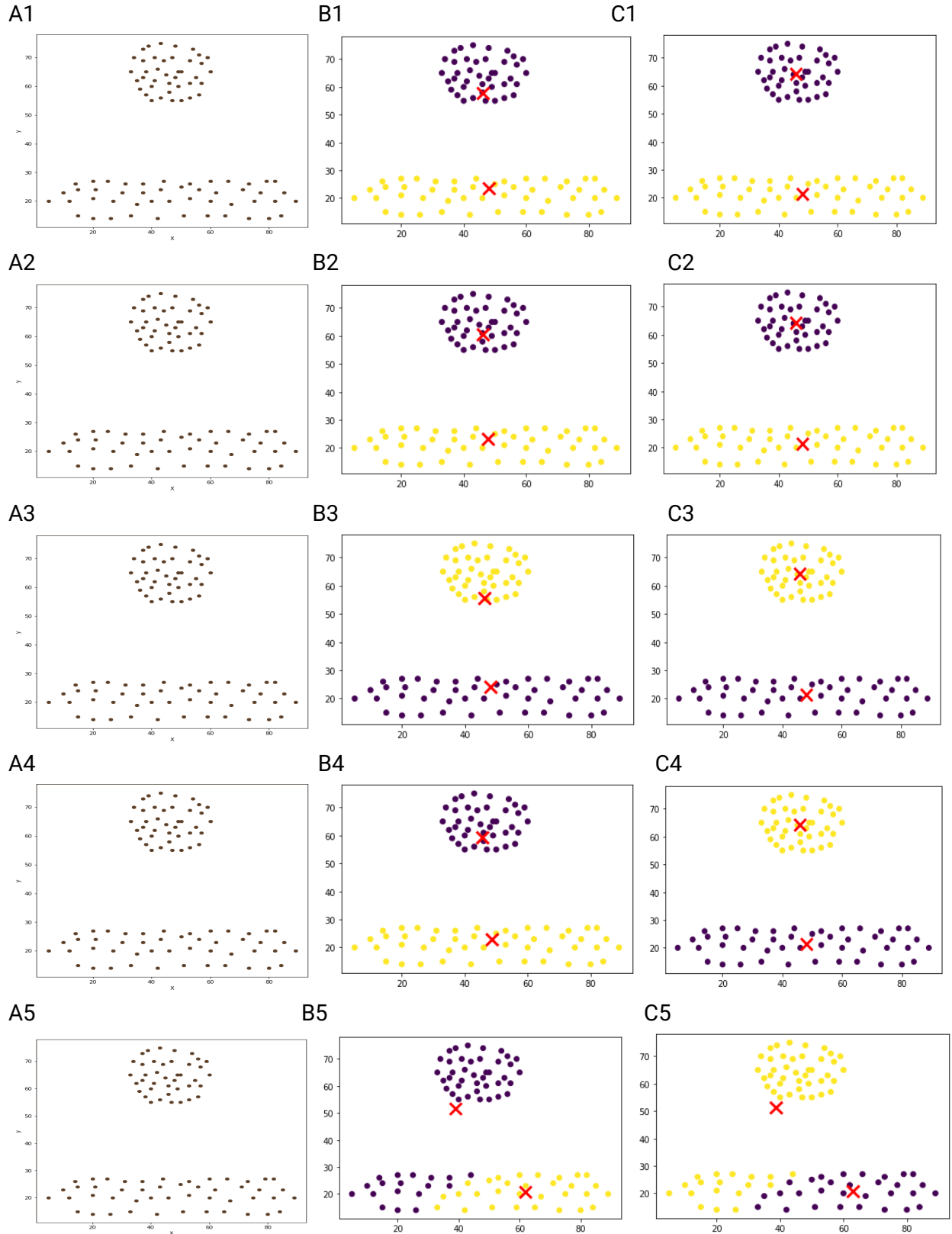


Fig .5 Result of Clustering, (A1)-(A5) Normal Dataset,(B1)-(B5)Result of FCM, (C1)-(C5) Result of K-means Using 5 (Euclidean, Manhattan, Minkowski, Cosine, Canberra) Distance Metrics.



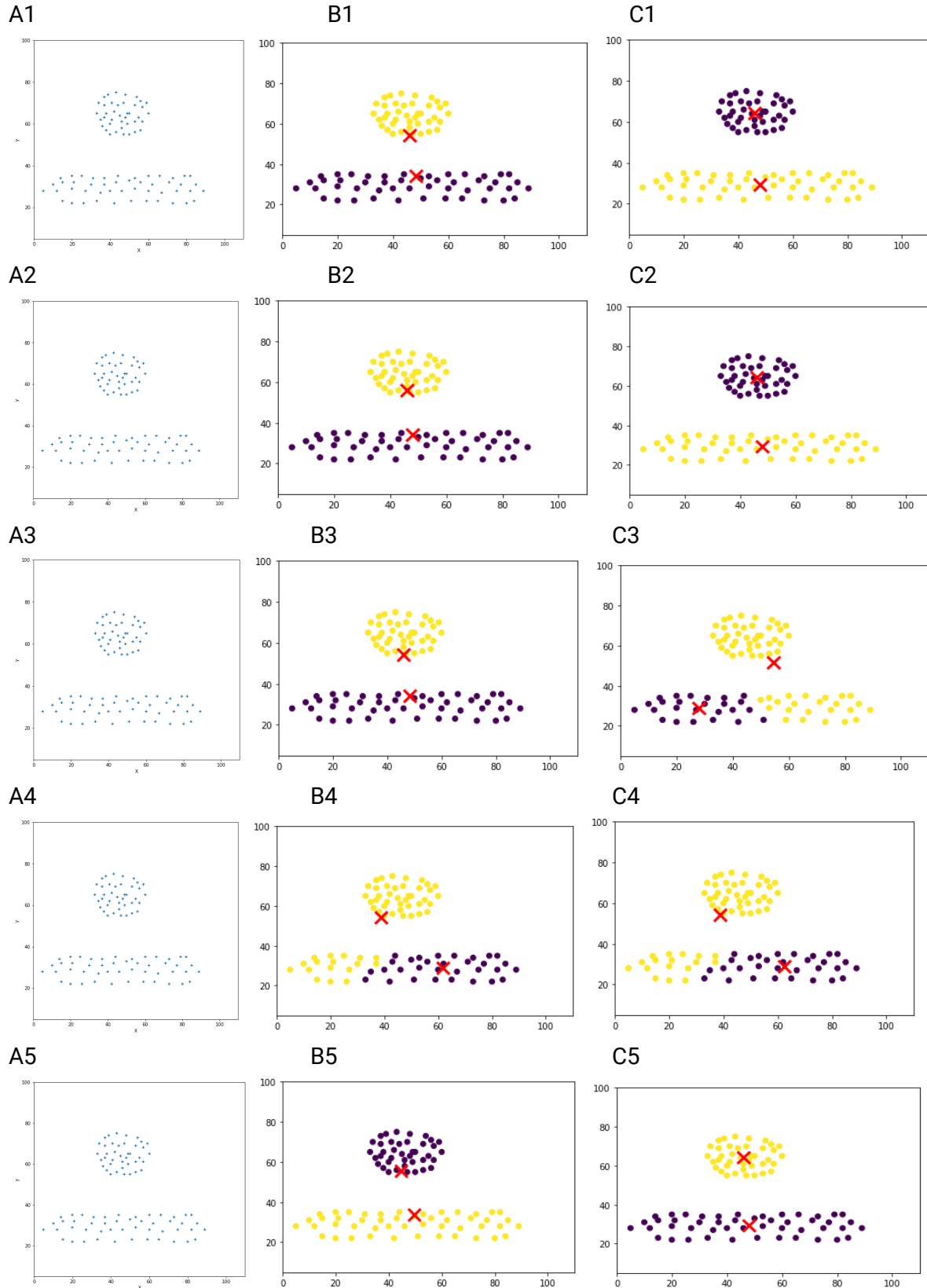


Fig.6 Result of Clustering, (A1)-(A5) Normal Dataset,(B1)-(B5)Result of FCM, (C1)-(C5) Result of K-means Using 5 (Euclidean, Manhattan, Minkowski, Cosine, Canberra) Distance Metrics. .

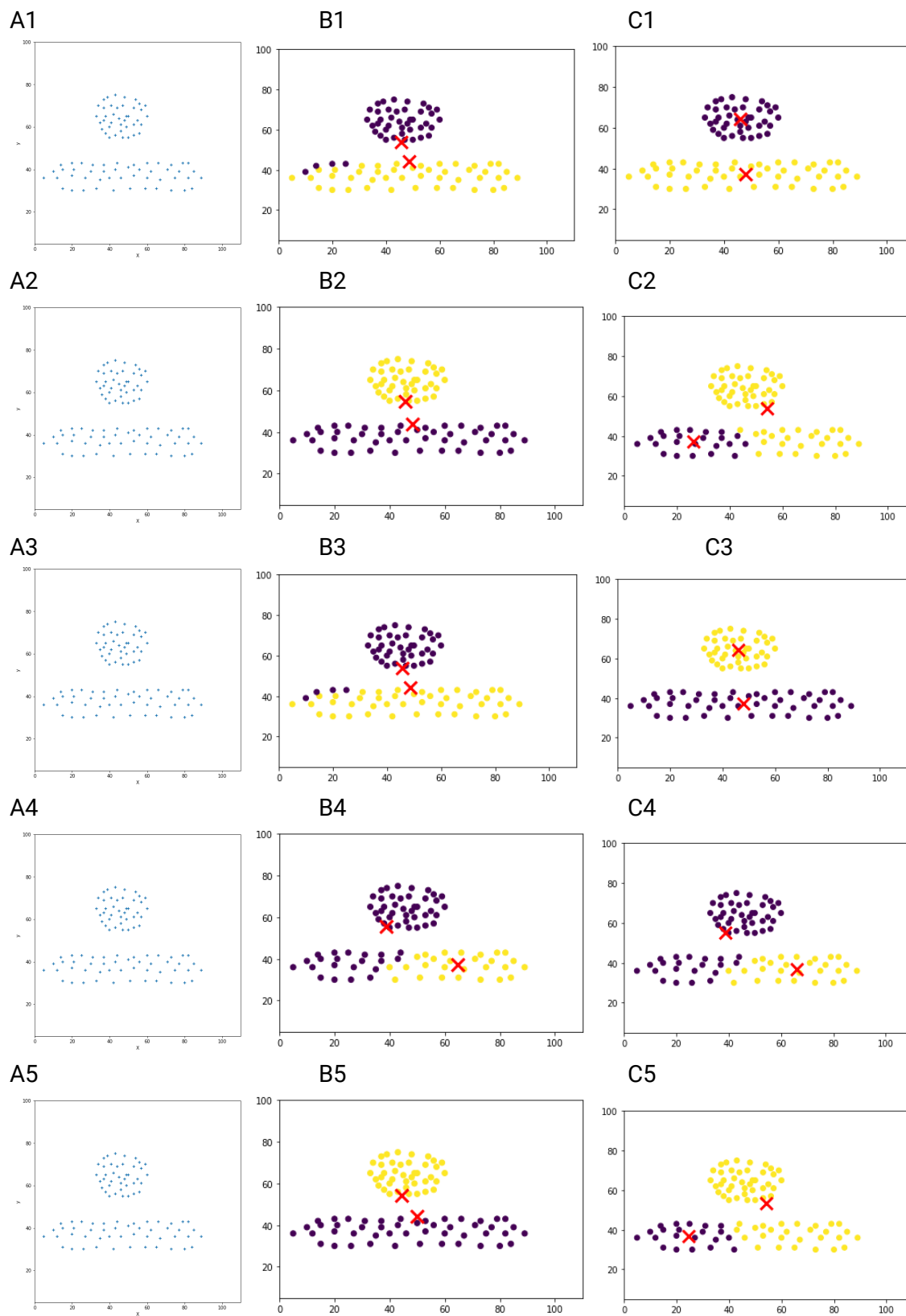


Fig.7 Result of Clustering, (A1)-(A5) Normal Dataset,(B1)-(B5)Result of FCM, (C1)-(C5) Result of K-means Using 5 (Euclidean, Manhattan, Minkowski, Cosine, Canberra) Distance Metrics. .

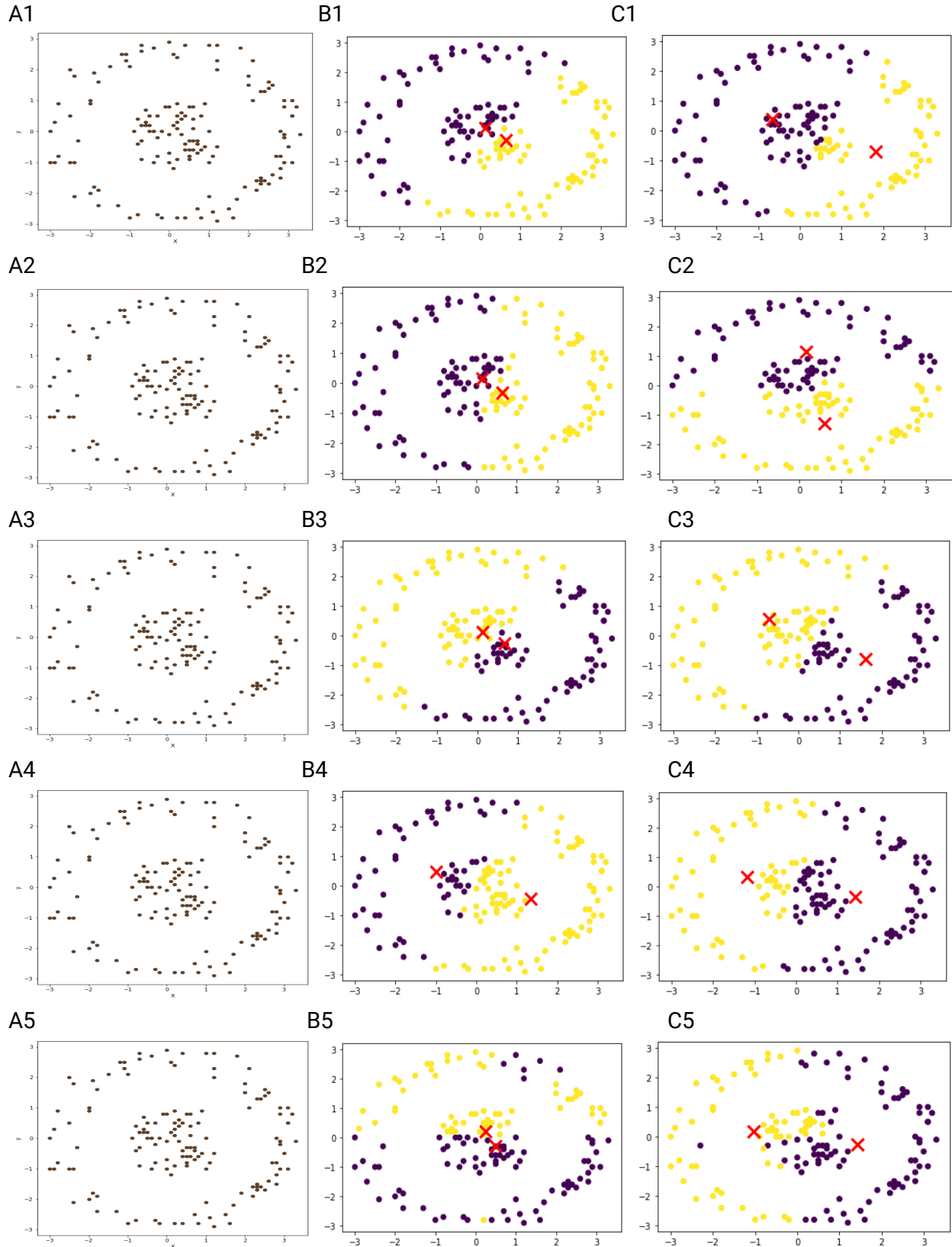


Fig.8 Result of Clustering, (A1)-(A5) Normal Dataset,(B1)-(B5)Result of FCM, (C1)-(C5) Result of K-means Using 5 (Euclidean, Manhattan, Minkowski, Cosine, Canberra) Distance Metrics. .

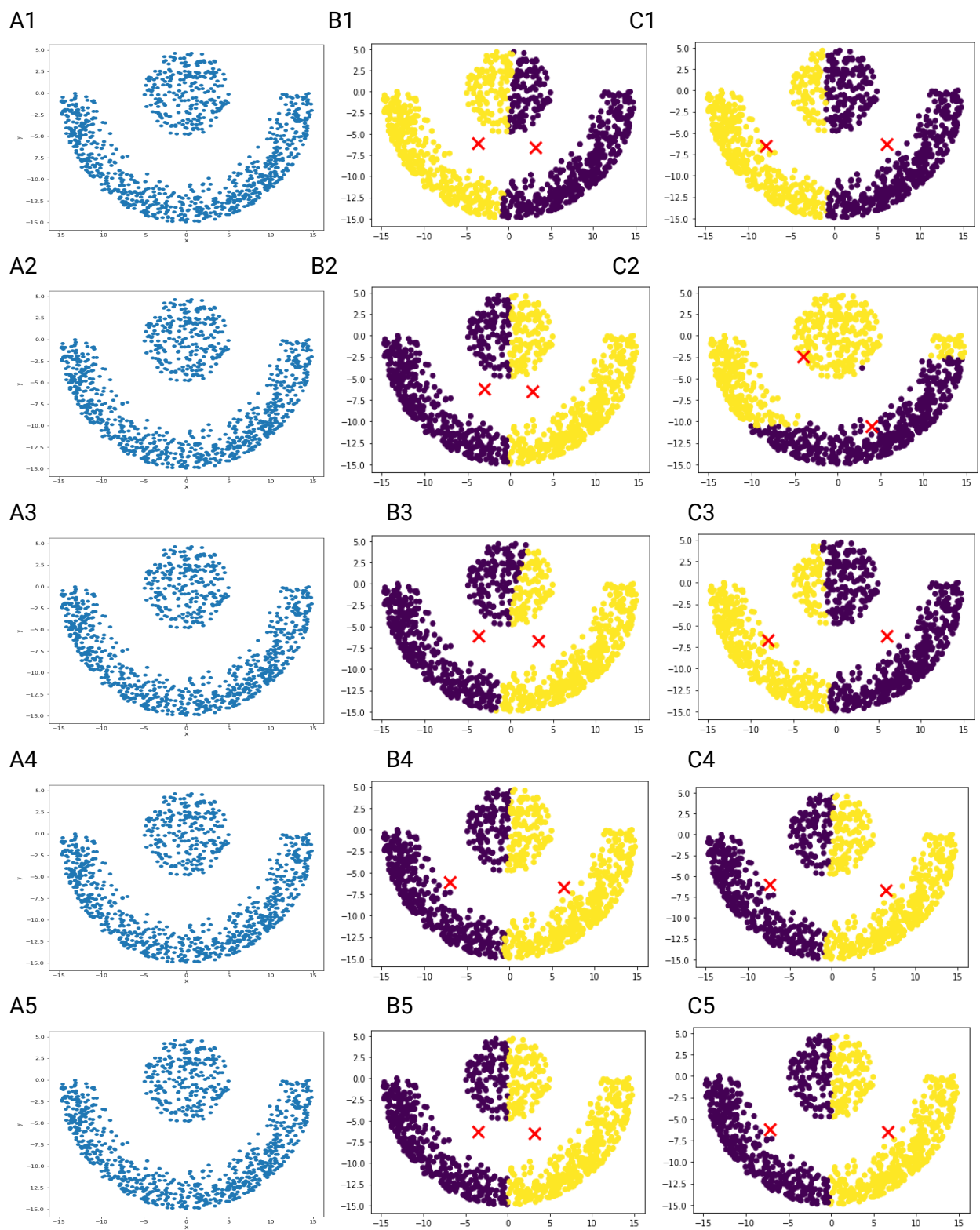


Fig.9 Result of Clustering, (A1)-(A5) Normal Dataset,(B1)-(B5)Result of FCM, (C1)-(C5) Result of K-means Using 5 (Euclidean, Manhattan, Minkowski, Cosine, Canberra) Distance Metrics. .

## CHAPTER 5

### DISCUSSION OF RESULT

To analyze the performance of K-means and FCM algorithms using Different Distance metrics such as Euclidean, Manhattan, Minkowski, Cosine and Canberra. We considered the following aspects.

- **Cluster Quality:**

By analyzing metrics like the Silhouette Coefficient, we evaluate the clustering's quality. These metrics assess the performance of the clustering process overall, the separation between clusters, or the compactness of the clusters.

- **Visual Inspection:**

The quality of the clustering results can be determined by plotting the clusters and visually inspecting them. The visual separation and compactness of clusters produced by various distance measurements can be compared.

- **Computational Efficiency:**

We have extensive datasets, comparing the computing efficiency of various distance measurements. Some distance measures could call for greater calculation time and resources than others.

- **Cluster Stability:**

By repeatedly executing the clustering algorithms with various random initializations and evaluating the consistency of the outcomes, we assessed the stability of the clusters. Consistent clusters throughout various runs suggest more dependable and stable outcomes.

## 5.1 Cluster Quality

Silhouette Score is generated for all the dataset on both K-means and Fuzzy C-means Clustering using five distance metrics such as Euclidean, Manhattan, Minkowski, Cosine and Canberra.

Based on the Score obtained by performing the clustering. It is observed that for K-means using Euclidean distance metrics outperforms in comparison with all the other distance metrics.

Also, FCM using Manhattan distance metrics outperforms in comparison with all other distance metrics. For, Circular and Elliptical Shaped Dataset on performing clustering algorithm FCM and K-means using Euclidean, Minkowski and Cosine distance metrics showing same silhouette score, as shown below.

Table 5.1 Silhouette Score Based On Clustering

Distance	Dataset	K-means	FCM	Best
Euclidean	Circular and Elliptical	0.6307	0.6307	Both
	Two Spherical	0.3258	0.3213	K-means
	Half and full moon	0.4277	0.4228	K-means
Manhattan	Circular and Elliptical	0.4276	0.6307	FCM
	Two Spherical	0.2903	0.3111	FCM
	Half and full moon	0.3523	0.4224	FCM
Minkowski	Circular and Elliptical	0.6307	0.6307	Both
	Two Spherical	0.3069	0.3213	FCM
	Half and full moon	0.4234	0.4228	K-means
Cosine	Circular and Elliptical	0.4126	0.4126	Both
	Two Spherical	0.3138	0.3148	Both
	Half and full moon	0.4233	0.4231	K-means

## 5.2 Computational Efficiency

The effectiveness or speed at which a computing algorithm or process completes its tasks is referred to as computation efficiency. It is a gauge of how quickly an algorithm can carry out a task or solve a problem utilizing the time and memory that are at its disposal. Computational efficiency for clustering algorithms like K-means or FCM can be measured by observing how long it takes for the algorithm to converge or reach a stopping point. This can be assessed by noting the algorithm's execution time, which is commonly expressed in seconds or milliseconds.

Table 5.2 Comparison of Computational Efficiency

Distance	Dataset	K-means	FCM	Best
Euclidean	Circular and Elliptical	0.002	0.047	K-means
	Two Spherical	0.003	0.273	K-means
	Half and full moon	0.039	5.443	K-means
Manhattan	Circular and Elliptical	0.002	0.0317	K-means
	Two Spherical	0.003	0.227	K-means
	Half and full moon	0.045	5.996	K-means
Minkowski	Circular and Elliptical	0.001	0.060	FCM
	Two Spherical	0.002	0.256	K-means
	Half and full moon	0.0005	5.714	K-means
Cosine	Circular and Elliptical	0.024	0.044	K-means
	Two Spherical	0.025	0.3105	K-means
	Half and full moon	0.368	1.206	K-means
Canberra	Circular and Elliptical	0.0	0.054	K-means
	Two Spherical	0.002	0.243	K-means
	Half and full moon	0.001	3.589	K-means

## Chapter 6

### Conclusion and Future Work

This work explores the Performance analysis of K-means and FCM with the help of different distance metrics such as Euclidean, Manhattan, Minkowski, Cosine, Canberra. However, the clustering based on the shape is still a challenging task when the clusters are unknown. The manual selection of centroids achieves better results than the automatic approach of selection. Also, Euclidean distance is perfect for most of the cases while computed with K-means Algorithm. Manhattan distance is perfect for most of the cases while using the FCM algorithm. Based on the computational efficiency it is observed that the computation timing is fast for K-means algorithms using any distance metrics on any type of dataset.

The future work for the performance analysis of K-means and FCM with different distance metrics encompasses various aspects such as evaluation on diverse datasets, comparison with other algorithms, parameter tuning, scalability analysis, visualization techniques, robustness analysis, real-world applications, hybrid approaches, interpretability, integration of domain knowledge, computational efficiency, and handling of noise and outliers. By addressing these research directions, we can advance our understanding of clustering algorithms and their applicability in different contexts, leading to improved clustering techniques and better solutions for data analysis problems.



## Bibliography

- [1] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, arXiv preprint arXiv:2001.05566 (2020).
- [2] N. Dhanachandra, K. Manglem, Y. J. Chanu, Image segmentation using k-means clustering algorithm and subtractive clustering algorithm, *Procedia Computer Science* 54 (2015) 764–771.
- [3] A. Dik, K. Jebari, A. Bouroumi, A. Ettouhami, A new fuzzy clustering by outliers, *Journal of Engineering and Applied Sciences* 9 (2014) 372–377.
- [4] W. Cai, S. Chen, D. Zhang, Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation, *Pattern recognition* 40 (2007) 825–838.
- [5] A. A.-h. Hassan, W. M. Shah, M. F. I. Othman, H. A. H. Hassan, Evaluate the performance of k-means and the fuzzy c-means algorithms to formation balanced clusters in wireless sensor networks., *International Journal of Electrical & Computer Engineering* (2088-8708) 10 (2020).
- [6] N. Singh, A. G. Mohapatra, G. Kanungo, Breast cancer mass detection in mammograms using k-means and fuzzy c-means clustering, *International Journal of Computer Applications* 22 (2011) 15–21.
- [7] S. Madhukumar, N. Santhiyakumari, Evaluation of k-means and fuzzy c-means segmentation on mr images of brain, *The Egyptian Journal of Radiology and Nuclear Medicine* 46 (2015) 475–479.
- [8] A. Singh, A. Yadav, A. Rana, K-means with three different distance metrics, *International Journal of Computer Applications* 67 (2013).
- [9] A. B. Amina Dik, 1Abdelaziz El moujahid, A. Ettouhami, Weighted distances for fuzzy clustering, *Applied Mathematical Sciences* 8 (2014) 147 – 156.
- [10] S. Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping k-means clustering method for medical applications, *Expert Systems with Applications* 67 (2017) 12–18.
- [11] S. Barak, T. Mokfi, Evaluation and selection of clustering methods using a hybrid group medm, *Expert Systems with Applications* 138 (2019) 112817.
- [12] P. K. Mishro, S. Agrawal, R. Panda, A. Abraham, A novel type-2 fuzzy c-means clustering for brain mr image segmentation, *IEEE Transactions on Cybernetics* (2020) 1–12.
- [13] S. Tyagi, S. Malhotra, D. Kumar, V. S. Verma, A. Bhardwaj, Mammographic image segmentation with modified fcm based clustering algorithm, *AIP Conference Proceedings* 2214 (2020) 020028.

- [14] P. Kavitha, P. V. Saraswathi, Segmentation for content based satellite image retrieval using fuzzy clustering, *International Journal of Advanced Science and Technology* 29 (2020).
- [15] R. Loochach, K. Garg, Effect of distance functions on simple k-means clustering algorithm, *International Journal of Computer Applications* 49 (2012) 7–9. doi:[10.5120/7629-0698](https://doi.org/10.5120/7629-0698).
- [16] D. Bora, D. Gupta, Effect of different distance measures on the performance of k-means algorithm: An experimental study in matlab, *arXiv* 5 (2014).
- [17] M. K. Arzoo, A. Prof, K. Rathod, K-means algorithm with different distance metrics in spatial data mining with uses of netbeans ide 8. 2, *Int. Res. J. Eng. Technol* 4 (2017) 2363– 2368.
- [18] A. Singh, A. Yadav, A. Rana, K-means with three different distance metrics, *International Journal of Computer Applications* 67 (2013).
- [19] A. Román y Zubeldia, Implementación de pruebas para una hipótesis sobre la aplicación de distancia euclidiana para realizar agrupamientos en espacios multidimensionales, 2018.
- [20] M. Bora, D. Jyoti, D. Gupta, A. Kumar, Effect of different distance measures on the performance of k-means algorithm: an experimental study in matlab, *arXiv preprint arXiv:1405.7471* (2014).
- [21] Y. Oike, M. Ikeda, K. Asada, A high-speed and low-voltage associative co-processor with exact hamming/manhattan-distance estimation using word-parallel and hierarchical search architecture, *IEEE Journal of Solid-State Circuits* 39 (2004) 1383–1387.
- [22] D. Androutsos, K. Plataniotiss, A. N. Venetsanopoulos, Distance measures for color image retrieval, in: *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, volume 2, IEEE, 1998, pp. 770–774.
- [23] V. Kumar, J. K. Chhabra, D. Kumar, Performance evaluation of distance metrics in the clustering algorithms, *INFOCOMP* 13 (2014) 38–52.
- [24] H.-D. Cheng, X. H. Jiang, Y. Sun, J. Wang, Color image segmentation: advances and prospects, *Pattern recognition* 34 (2001) 2259–2281.
- [25] D. Bora, A. Gupta, F. Khan, Comparing the performance of  $l^*a^*b^*$  and hsv color spaces with respect to color image segmentation, *arXiv* (2015).

- [26] S. Sural, G. Qian, S. Pramanik, Segmentation and histogram generation using the hsv color space for image retrieval, in: Proceedings. International Conference on Image Processing, volume 2, IEEE, 2002, pp. II–II.
- [27] D. Kaur, A comparative study of various distance measures for software fault prediction, arXiv preprint arXiv:1411.7474 (2014).
- [28] J. C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well separated clusters, *Journal of Cybernetics* 3: 32-57 (1973).
- [29] O. Moh'd Alia, R. Mandava, M. E. Aziz, A hybrid harmony search algorithm for mri brain segmentation, *Evolutionary Intelligence* 4 (2011) 31–49.
- [30] P.-Y. Zhou, K. C. Chan, A model-based multivariate time series clustering algorithm, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2014, pp. 805–817.
- [31] K.-L. Wu, Analysis of parameter selections for fuzzy c-means, *Pattern Recognition* 45 (2012) 407–415.