ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

Comparative Analysis of Kmeans Technique on Non Convex Cluster

Gummadi Venkata Nikhil Sai[1]; Robby Aulia Tubagus[2]; Vasala Rohith[3]; Haritha Donavalli[4]
[1,2,3]Department of CSE, [4]Professor, Department of CSE
Koneru Lakshmaiah Educational Foundation,
Vaddeswaram, AP, India

Abstract:- Clustering algorithms play a critical role in data analysis by grouping similar data points to reveal hidden pat- terns and structures. This study investigates the performance of several clustering algorithms using two distinct datasets: moons and circles. The primary focus is on evaluating and comparing the execution times of these algorithms to determine their efficiency and effectiveness in handling different types of data distributions. Through a series of experiments and performance measurements, this paper aims to provide a detailed analysis of each algorithm's computational efficiency and suitability for various clustering tasks. The findings are expected to offer practical insights into the selection and application of clustering methods, contributing to enhanced data analysis techniques and informed decision-making in diverse fields.

Keywords:- *Non-Convex Optimization, K-Means, Machine Learning, Comparative Analysis.*

I. INTRODUCTION

In recent years. the data produced has different variety and complexity and because of that we need to select and specify how to treat them differently, the need for an efficient and effective methods are urgently needed[1]. One well know algorithm used in various fields of research and industry is clustering, where they have ability to group evaluated data points in a cluster and possibly uncover hidden pattern[2], [3]. Such that it can help to increasing the insights for data driven-decision making and increasing the understability of the data distribution and structure.

The distribution of data in clustering is divided into two type: Convex and Non-Convex, these data distribution are affecting the performance of clustering algorithms such as k means. here we consider the shapes are the form where the data are clustered and grouped together forming a form of shape. Convex distribution is if we draw a line between any two points within the shape and it still remain entirely within the shapes, this distribution is relatively straight- forward for k means to handle[4]. Whereas, Non-Convex distribution is if two line draw together are disconnected, this distribution is a significant challenges for k means to handle, which assumes the data distribution to be convex, this limitation affecting accuracy of cluster which k means produce[5], [6].

Our paper explores the performance of several techniques, that helping k means algorithm to achieve higher performa in handling Non-Convex distribution, focusing on two distinct dataset from scikit learn: Moons[7] and Circles[8]. This research comparing execution time and execution accuracy for ten selected technique under k means clustering algorithm. By conducting a comprehensive evaluation of these techniques, this study aims to provide the insights to help to choose the technique for increasing k means performa for handling Non-Convex distribution.

II. RELATED WORKS

Salim and Ismail(1983) Their paper explain the convergence of k means, presenting a generalized convergence theorem for k means algorithm. They demonstrated k means limitation to Non-Convex distribution, as this algo- rithm is prone to converging to local minima rather than the global optimum. This limitation underscores the need for advanced initialization techniques, such as to set the initial centroids strategically to avoiding poor local minima and enhancing outcomes.

Xu and Wunsch(2005) They conducting the comprehensive analysis of the clustering algorithms focusing on Non-Convex distributions. Addressing the shortcoming of k means in handling Non-Convex data structure also high- lighting their effectiveness to handle such limitation.

Bhagav and Pavar(2016) In their review paper they summarized the impact of Non-Convexity for various existing algorithms, even in the presence of missing and noisy data. Their review underscores the robustness of selected algorithms to manage complexities associated with Non- Convex distribution.

III. METHODOLOGY

- *K Means Algorithm*

A common method for clustering datasets is the k-means algorithm, which divides the dataset into k clusters, each of which has a single data point that corresponds to the cluster with the closest mean[12]. Steps of the k-means Algorithm:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

- *Initialization*: Choose k data points at random to serve as the dataset's initial centroids. This step can also be improved using the k-means++ initialization to spread out the initial centroids more effectively.
- *Assignment Step*: Determine the distance to each centroid for each data point. The most common distance metric used is the Euclidean distance: The distance between xi and cj is given by:

$$\text{distance}(x_i, c_j) = \sqrt{\sum_{d=1}^{D} (x_{i,d} - c_{j,d})^2} \quad (1)$$

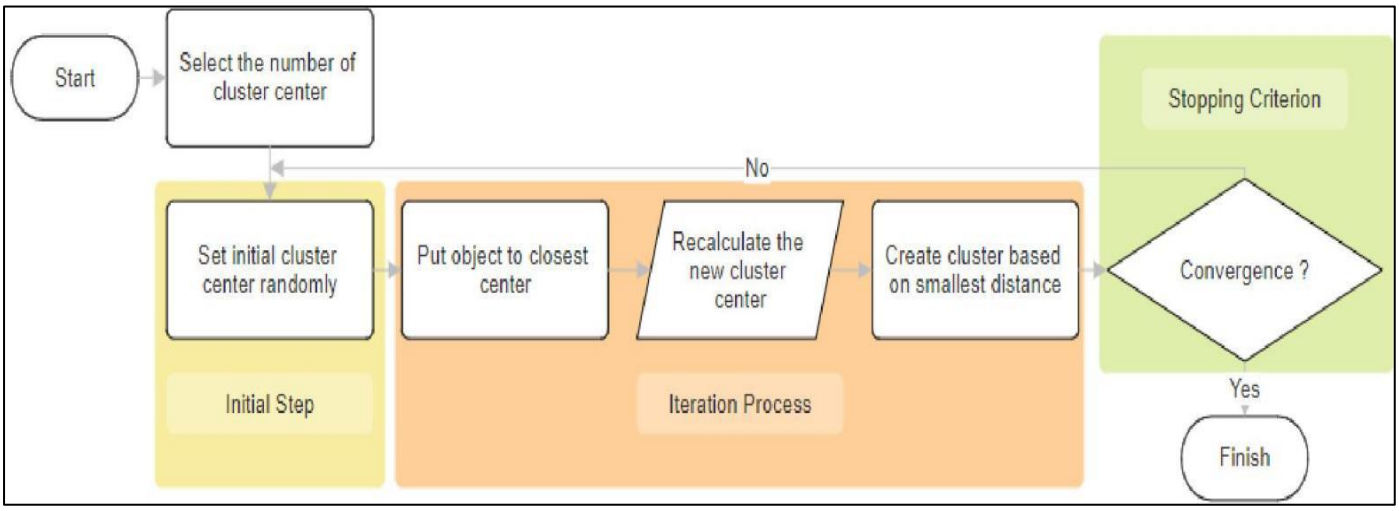

Fig 1 K means Flow Chart

$$\sqrt{\sum_{d=1}^{D} (x_i^d - c_j^d)^2} \quad (2)$$

Assign every data point to the closest centroid's cluster.

- *Update Step*:

For each cluster, update the centroid by calculating the mean of all data points assigned to that cluster. The centroid c_j of a cluster C_j is given by:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \quad (3)$$

Where:

✓ $|C_j|$ is the number of points in cluster C_j.

✓ $\sum_{x_i \in C_j} x_i$ is the sum of all points x_i in cluster C_j.

- *Convergence Check* :

Check if the centroids have changed significantly. If not, the algorithm has converged. Another convergence criterion can be if the cluster as- signments no longer change, or if a predefined maximum number of iterations is reached.

➢ *Why Non-Convex not suit for k-means*

K-means clustering, a widely used algorithm for partitioning datasets into distinct clusters, faces significant challenges when applied to non-convex shapes. These challenges stem from its inherent characteristics, particularly in the initial step, iteration process, and stopping criterion.

The initial step of K-means involves selecting initial centroids either randomly or through strategies like k-means++. This step presumes that clusters are approximately spherical (convex) and of similar size, which does not hold true for non-convex shapes such as crescents or rings. Consequently, the initial placement of centroids may not accurately represent the actual structure of non- convex clusters, leading to suboptimal clustering outcomes. For example, a centroid placed within a crescent-shaped cluster might not effectively capture the cluster's unique form, causing the algorithm to misinterpret the true cluster configuration.

Every data point is assigned by K-means to the closest centroid throughout the iteration phase, and the centroids are then recalculated using the average of the assigned points. This method is ineffective for non-convex geome- tries, but it performs well for convex clusters when a central point may accurately represent the cluster. In non- convex clusters, points that are close in Euclidean space may belong to different non-convex regions, whereas points farther apart might actually belong to the same cluster.K-means thus fails to capture the underlying structure of the data and may wrongly split a non-convex cluster into multiple convex regions or combine several non-convex clusters into a single group.

The stabilization of centroids, or the algorithm reaching a maximum number of iterations, is the basis for the K- means stopping criterion, which states that point assignments to clusters cease to vary. This criterion assumes that the clusters identified during the iterations are accurate. However, for non-convex shapes, K-means may converge to a local minimum where the final clusters do not accurately

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

reflect the data's structure. This local minimum can trap the algorithm into producing clusters that do not align with the true non-convex nature of the data.

➢ *K-Means Techniques*

The various techniques for enhancing the K-means algorithm focus on addressing its inherent limitations, particularly in managing complex clustering scenarios. These techniques make substantial improvements by optimizing key aspects such as initial centroid placement, iteration processes, and stopping criteria. For instance, k-means++ Initialization enhances the initial step by probabilistically selecting more dispersed cluster centers, which helps avoid poor local minima and improves clustering accuracy. Genetic Algorithms for K-means and Simulated Annealing in- troduce advanced strategies for initial centroid placement, using evolutionary and probabilistic methods to explore a wider range of potential solutions and thereby enhance the chances of finding a global optimum.

During the iteration process, techniques such as Spectral Clustering and Kernel K-means transform the data into higher-dimensional or similarity-based spaces, making it easier to identify and separate complex clusters that traditional K-means might struggle with. This transformation helps K-means handle intricate and non-convex shapes by modifying the data representation before clustering. By enabling data points to have differing degrees of membership in various clusters, fuzzy C-means enhances the iteration process and offers a more flexible and sophisticated method of cluster assignment.

In terms of stopping criteria, methods like Mini-Batch K- means introduce efficiency by converging centers through iterative updates with mini-batches of data, making the algorithm scalable to larger datasets. The EM Algorithm for Gaussian Mixture Models employs convergence of the log- likelihood function as a stopping criterion, offering a more comprehensive measure of model fit, especially in complex distributions. Techniques like Constraint-Based Clustering ensure that the algorithm respects additional constraints during both the assignment and updating phases, improv- ing clustering accuracy while adhering to specific require- ments.

Overall, these optimization techniques refine K-means by enhancing its initial conditions, iteration process, and stopping criteria. They enable the algorithm to better man- age complex data structures, handle overlapping clusters, and improve efficiency and scalability, resulting in more accurate and robust clustering outcomes for non-convex and intricate datasets.

Table 1 Comparison of K-Means Optimization Techniques

Technique	Initial Step	Iteration Process	Stopping Criteria
k-means++ Initialization [13]	Probabilistic selection of initial centers to spread out[1]	Assign points to nearest center, update centers to mean	Centroid convergence, assignment stability, max iterations
Genetic Algorithms for k-means [14]	Generate initial population of cluster centers	Apply selection, crossover, mutation; evaluate fitness	Convergence of fitness scores, max generations
Simulated Annealing for k-means [15]	Random initial solution	Generate neighbor solutions; probabilistically accept/reject[2]	Convergence of cost function, min temperature, max iterations
Hierarchical K-means (Bisecting K-means) [16]	Start with one cluster	Split clusters iteratively using k-means	Desired number of clusters achieved
Constraint-Based Clustering [17]	Initialize centers considering constraints	Modify k-means to respect constraints during assignment and updating	Convergence while respecting constraints, max iterations
Spectral Clustering [18]	Construct similarity matrix	Eigenvalue decomposition; apply k-means to lower-dimensional representation[3]	Convergence in k-means step, max iterations
Kernel K-means [19]	Apply kernel function to transform data[4]	Perform k-means in transformed space	Convergence in kernel space, max iterations
Fuzzy C-means [20]	Initialize membership values[5]	Update centers and membership values iteratively	Convergence of membership values, max iterations
EM Algorithm for GMM [21]	Initialize Gaussian parameters	Expectation step (E-step),[6] Maximization step (M-step)[7]	Convergence of log-likelihood, parameter change below threshold, max iterations
Mini-Batch K-means [22]	Random or k-means++ initialization	Use mini-batches[8]to update centers iteratively	Convergence of centers, max mini-batch iterations

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

- The k-means++ algorithm improves the initial selection of cluster centers to enhance convergence and accuracy.
- Simulated annealing probabilistically accepts solutions that are worse than the current solution to escape local minima.
- Eigenvalue decomposition is used to reduce the dimensionality of data by finding new axes (principal components) that capture the most variance.
- Kernel functions are used to transform data into a higher-dimensional space where it may be easier to separate clusters.
- In fuzzy c-means, each data point has a degree of belonging to each cluster, represented by membership values rather than a hard assignment.
- Expectation step (E-step) is where the probability of each data point belonging to each Gaussian component is calculated.
- Maximization step (M-step) is where the parameters of the Gaussian components are updated based on the probabilities calculated in the E-step.
- Mini-batches are small random subsets of the dataset used in each iteration to update cluster centers, making the algorithm faster and scalable to large dataset.

IV. COMPARATIVE ANALYSIS

This section presents a comparative study of different clustering algorithms on non-convex data. By evaluating each method based on both clustering performance and ex- ecution time, we aim to provide insights into the strengths and limitations of each approach. This comparison will help in understanding which algorithms are best suited for different types of data and computational requirements.

➢ *Non-Convex Clusters Dataset*

In this subsection, we explore datasets that contain non-convex clusters, which pose a significant challenge for tra-ditional clustering algorithms. Non-convex clusters do not conform to a single shape, such as a circle or ellipse, and often require more sophisticated techniques to accurately identify and separate them. Two prominent examples of non-convex clusters are circular clusters and moon-shaped clusters.

- *Circular Clusters:*

Circular clusters are designed to form ring-like shapes, which are inherently non-convex. These clusters are particularly challenging for algorithms like K-Means, which assume convex shapes by trying to minimize the distance within each cluster. Circular clusters are useful for testing the capability of clustering algorithms to handle complex geometric arrangements. In our exper- iments, we generated circular clusters to assess how well each algorithm can manage these non-traditional cluster shapes. The dataset consists of data points arranged in concentric circles, providing a clear test for the algorithm's ability to detect and separate clusters with non-linear boundaries.

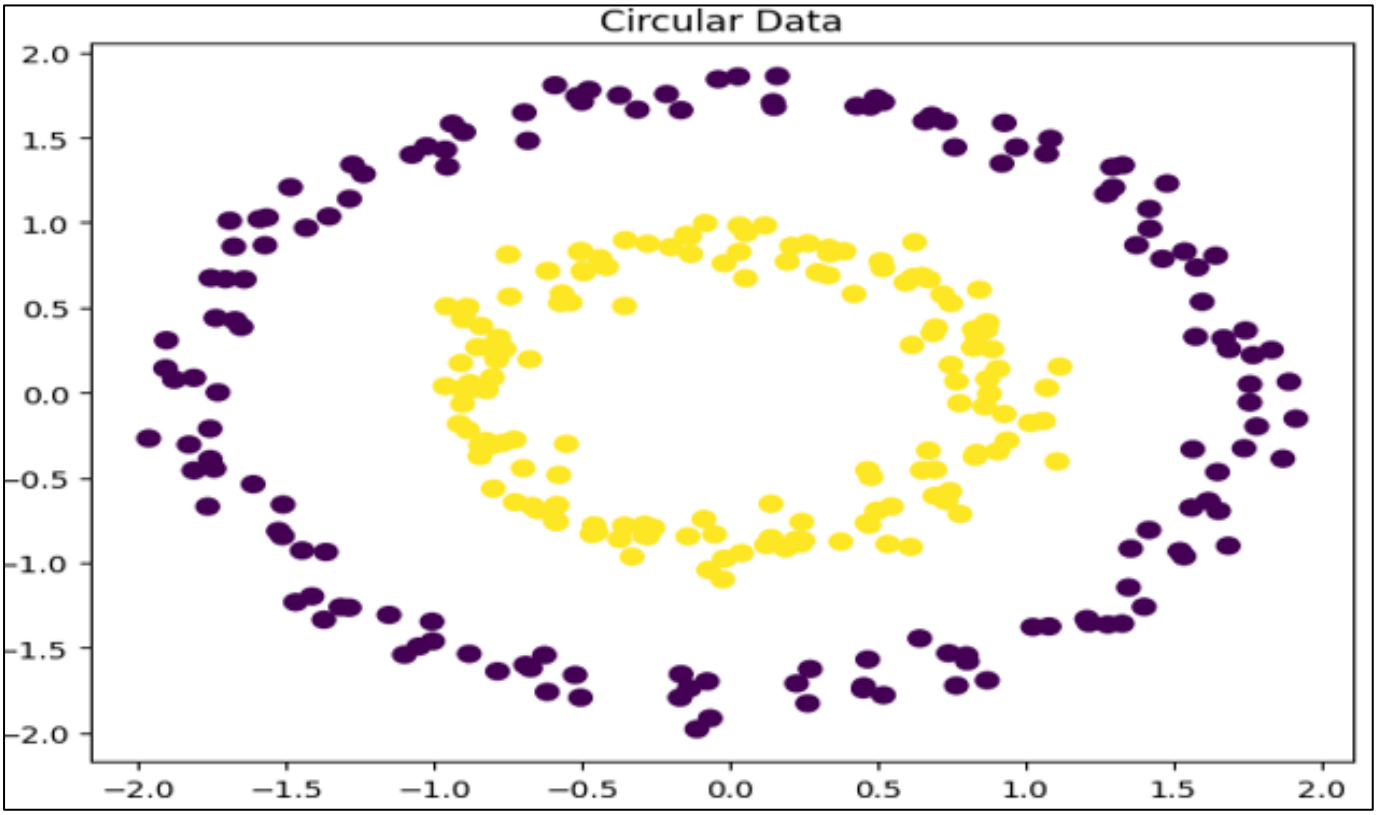

Fig 2 Circular Cluster

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

- *Moon-Shaped Clusters:*

The moon-shaped clusters, also called the "two moons" dataset, are made up of two crescent-shaped clusters that interlock. This dataset is widely used in the literature to benchmark clustering algorithms, particularly those designed to handle non- convex shapes. The intertwined nature of the clusters poses a substantial challenge for algorithms that rely on distance- based metrics assuming linear separability. The moon- shaped clusters are an excellent test for evaluating the robustness and flexibility of clustering algorithms. For our study, we utilized a synthetic two moons dataset to com- pare the performance and accuracy of different clustering techniques in identifying and correctly classifying the non- convex structures inherent in the data.

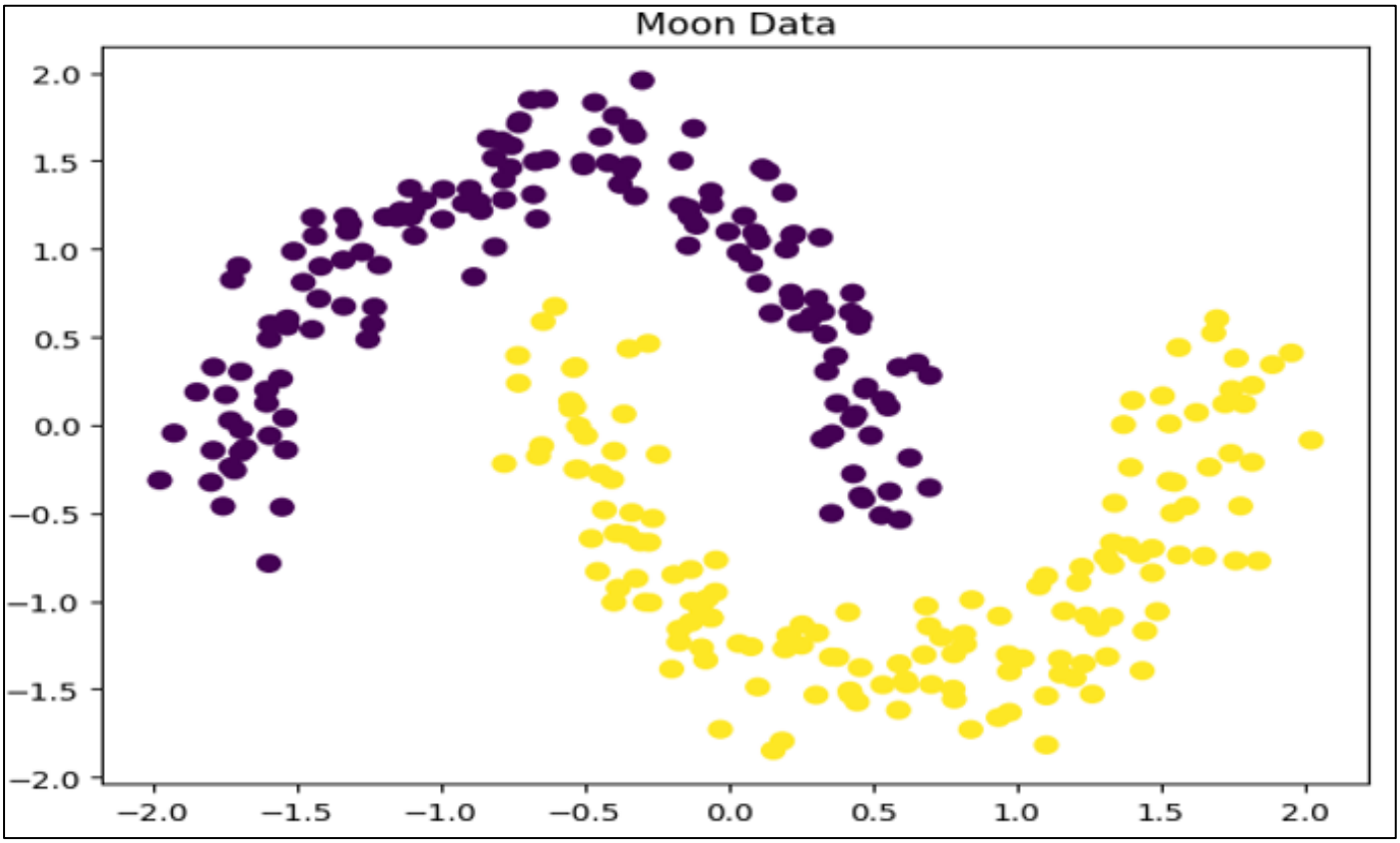

Fig 3 Moon Cluster

➢ *Accuracy Calculation*

To evaluate the accuracy of the clustering algorithms, we use a mapping-based accuracy metric. This involves mapping each cluster label to the most frequent true label within that cluster. The accuracy is then computed as the proportion of correctly mapped labels to the total number of data points. The equation for accuracy (A) is given by:

$$A = \frac{1}{N} \sum_{i=1}^{N} \delta(y_i, \hat{y}_i) \tag{4}$$

Where N is the total number of data points, yi is the true label of the i -th data point, y^i is the mapped cluster label of the i -th data point, and δ is the Kronecker delta function defined as:

$$\delta(y_i, \hat{y}_i) = \begin{cases} 1 & \text{if } y_i = \hat{y}_i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This accuracy metric provides a straightforward way to quantify the effectiveness of clustering algorithms by considering the most representative label for each cluster.

This table presents the accuracy results of various cluster- ing algorithms on two types of non-convex clusters: circular and moon-shaped. By comparing the accuracies, we can determine which algorithms are better suited for handling complex, non-linear cluster structures.

➢ *Execution Time for Moons and Circles Datasets*

This subsection presents the execution times of different clustering algorithms when applied to two distinct datasets: moons and circles. The moons dataset consists of 300 samples with a two-moon structure, while the circles dataset features 300 samples arranged in concentric circles.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

Table 2 Accuracy Results of Clustering Algorithms on Circular and Moon-Shaped Clusters

Clustering Algorithm	Circular	Moon-Shaped
Standard K-means	0.5000	0.8467
K-means++ Initialization	0.5033	0.8467
Simulated Annealing K-means	0.5033	0.8467
Hierarchical K-means	0.5033	0.8467
Constraint-Based Clustering	0.5000	0.8467
Spectral Clustering	1.0000	0.9733
Kernel K-means	0.5067	0.8700
Gaussian Mixture Models	0.5033	0.8467
Mini-Batch K-means	0.5133	0.8467
Fuzzy C-Means	0.5000	0.8500
Genetic Algorithm K-means	0.5000	0.8467

Table 3 Execution Times for Clustering Algorithms on the Moons and Circles Datasets

Clustering Algorithm	Circles (s)	Moons (s)
Standard K-means	0.0086	0.0096
K-means++ Initialization	0.0090	0.0119
Simulated Annealing K-means	4.8480	4.9387
Hierarchical K-means	0.0137	0.0102
Constrained K-means	0.0029	0.0023
Spectral Clustering	0.0649	0.0405
Kernel K-means	0.0521	0.0341
EM Algorithm for GMM	0.1077	0.0049
Mini-Batch K-means	0.0176	0.0085
Fuzzy C-means	0.0882	0.0029
Genetic Algorithm K-means	0.7073	0.7177

To evaluate the computational efficiency of each clustering algorithm, we measure the execution time (T) taken to complete the clustering process for both datasets. The execution time for an algorithm A on a dataset D is given by:

$$T_A(D) = \text{end\_time}_A(D) - \text{start\_time}_A(D) \quad (6)$$

Where $\text{start\_time}_A(D)$ is the recorded time just before the algorithm A begins processing the dataset D, and $\text{end\_time}_A(D)$ is the recorded time just after the algorithm completes processing. The following table summarizes the execution times for each algorithm on both datasets. This comparison helps to understand how each clustering method scales with differ- ent data structures and how their computational demands vary. The results highlight both the efficiency and potential trade-offs of various clustering approaches in practice.

Overall, algorithms like Standard K-means and Constrained K-means are among the fastest, while Simulated Annealing and Genetic Algorithm K-means tend to be slower due to their more complex nature. The choice of algorithm may depend on the specific requirements for accuracy versus computational efficiency.

V. RESULTS

Spectral Clustering consistently outperformed other algorithms in terms of accuracy, achieving a perfect score on the circular dataset (1.0000) and a very high score on the moon-shaped dataset (0.9733). Kernel K-means also showed strong performance, especially on the moon- shaped dataset, where it achieved an accuracy of 0.8700. Most other algorithms, including Standard K-means and its variants, demonstrated similar and slightly lower accuracy, with values clustering around 0.8467 for the moon-shaped dataset and closer to 0.5000 for the circular dataset. This suggests that Spectral and Kernel K-means are particularly well-suited for handling non-convex shapes, while tradi- tional methods may struggle with these types of data.

In terms of execution time, traditional algorithms like Standard K-means and Constrained K-means were the fastest, completing clustering tasks in milliseconds. More complex methods, such as Simulated Annealing K-means, were significantly slower, requiring several seconds to com- plete the same tasks. Spectral Clustering and Kernel K- means, despite their higher accuracy, had moderate exe- cution times, balancing both accuracy and computational efficiency.

VI. FUTURE ENHANCEMENT

Future enhancements to this study could include exploring different types of non-convex clusters beyond the circular and moon-shaped datasets analyzed here. For instance, extending the analysis to more complex shapes such as spirals, polygons, or even irregular real-world clusters would offer a broader understanding of algorithm performance across diverse non-convex structures. Additionally, tuning the parameters of these algorithms, such as the number of clusters, initialization methods, or distance metrics, could yield significant improvements in both accuracy and ef-

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1507

ficiency. Systematic parameter optimization could further enhance the effectiveness of clustering algorithms when applied to non-convex datasets.

VII. CONCLUSION

The choice of a clustering algorithm should be guided by the specific requirements of the task. For applications where accuracy in non-convex shapes is critical, Spectral and Kernel K-means are recommended despite their higher computational cost. Conversely, for tasks requiring rapid execution, traditional methods like Standard K-means may be preferable, albeit with potentially lower accuracy.

REFERENCES

[1]. Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016). Robust ran- dom cut forest based anomaly detection on streams. Proceedings of the 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research, 48, 2712-2721. Available at: https://proceedings.mlr.press/v48/guha16.html.

[2]. Aggarwal, C. C., & Reddy, C. K. (2014). Data clustering: Algorithms and applications. CRC Press. (pp. 3, 5). Available at: https://people. cs.vt.edu/~reddy/papers/DCBOOK.pdf.

[3]. Xu, R., & Wunsch, D. C. (2009). Clustering. Wiley-IEEE Press. (p. 2). ISBN: 9780470276808. DOI: 10.1002/9780470382776.

[4]. Har-Peled, S. (2012). Geometric approximation algorithms. American Mathematical Society. (p. 10). Available at: https://graphics.stanford. edu/courses/cs468-06-fall/Papers/01%20har-peled%20notes.pdf.

[5]. Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In Proceedings of the 17th International Conference on Machine Learning (pp. 727-734).

[6]. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, pp. 281-297).

[7]. Scikit-learn developers. (2024). sklearn.datasets.make_moons. In Scikit-learn documentation. Available at https://scikit-learn.org/ stable/modules/generated/sklearn.datasets.make_moons.html.

[8]. Scikit-learn developers. (2024). sklearn.datasets.make_circles. In Scikit-learn documentation. Available at https://scikit-learn.org/ stable/modules/generated/sklearn.datasets.make_circles.html.

[9]. Selim SZ, Ismail MA. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Trans Pattern Anal Mach Intell. 1984 Jan;6(1):81-7. doi: 10.1109/tpami.1984.4767478. PMID: 21869168.

[10]. Rui Xu and D. Wunsch, "Survey of clustering algorithms," in IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, May 2005, doi: 10.1109/TNN.2005.845141.

[11]. Bhargav, Sushant & Pawar, Mahesh. (2016). A Review of Clustering Methods forming Non-Convex clusters with, Missing and Noisy Data. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGI- NEERING. 3. 39-44.

[12]. S. Lloyd, "Least squares quantization in PCM," in IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, March 1982, doi: 10.1109/TIT.1982.1056489.

[13]. Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advan- tages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8. 1027-1035. 10.1145/1283383.1283494.

[14]. K. Krishna and M. Narasimha Murty, "Genetic K-means algo- rithm," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 29, no. 3, pp. 433-439, June 1999, doi: 10.1109/3477.764879.

[15]. Ingber, L. (1993). Simulated annealing: Practice versus theory. Math- ematical and Computer Modelling, 18(11), 29-57. DOI: 10.1016/0895-7177(93)90204-C.

[16]. Steinbach, Michael & Karypis, George & Kumar, Vipin. (2000). A Comparison of Document Clustering Techniques. Proceedings of the International KDD Workshop on Text Mining.

[17]. Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). "Constrained k-means clustering with background knowledge." Proceedings of the Eighteenth International Conference on Machine Learning (ICML).

[18]. Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). "On spectral clustering: Analysis and an algorithm." Advances in Neural Information Process- ing Systems (NIPS).

[19]. Dhillon, I. S., Guan, Y.,& Kulis, B. (2004). "Kernel k-means: Spectral clustering and normalized cuts." Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[20]. Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2–3), 191-203. ISSN 0098-3004. DOI: 10.1016/0098-3004(84)90020-7.

[21]. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22. ISSN 0035-9246. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[22]. Sculley, D. (2010). "Web-scale k-means clustering." Proceedings of the 19th International Conference on World Wide Web.1177-1178. DOI: 10.1145/1772690.1772862.