Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4976-4991 Received 25-11-2025; Accepted 31-01-2026; Published online 06-02-2026

Hybrid IDK_means++: Integrating Particle Swarm Optimization for Robust and Accurate K_means Initialization

Fadi Yamout fadi.yamout@liu.edu.lb
Lebanese International University,
School of Arts & Sciences,
Computer Science Information Technology,
Beirut, Lebanon.

Corresponding Author: Fadi Yamout

Abstract

Clustering is a method of grouping items based on shared qualities. Clustering can be used for market segmentation, crime analysis, urban landscape quality assessment, and pattern discovery in large datasets. Although a few clustering methods have linear time complexity but lower accuracy, clustering strategies are computationally costly. The K_means algorithm is a clustering algorithm that randomly select K centroids at the initial stages, the results are affected by these initial selections. To overcome this problem, an IDK_means algorithm was introduced to determine these initial centroids. IDK_means has shown better results than the K_means algorithm. This paper extends the IDK_means algorithm by improving centroid selection using Particle Swarm Optimization (PSO). PSO replaces the averaging of centroids used in the IDK_means with finding the best fit for each centroid within its cluster. IDK_means++ is tested on many datasets. The experiments were also repeated using different clusters. We applied Euclidean distance and cosine similarity and assessed the results using the Silhouette Coefficient, Davies–Bouldin Index and Calinski–Harabasz Index, where IDK_means algorithm never outperformed IDK_means++. The paper indicates that incorporating PSO into IDK_mean gave better results for all types of datasets.

Keywords: Clustering, IDK_means, K_means, Machine learning, Silhouette Coefficient, Davies–Bouldin Index, Calinski–Harabasz Index.

1. INTRODUCTION

Clustering techniques organize data into meaningful groups, where data within the same group are more like each other than data in different groups. Clustering is used in image analysis, bioinformatics, market research, text mining, and social network analysis [1, 2]. It identifies hidden patterns to understand the relation among the data [3]. One of the essential clustering techniques is the K_means, which is most widely used due to its simplicity and computational efficiency [4]. However, K_means' performance depends on the initial selection of centroids, which are selected

Citation: Fadi Yamout. Hybrid IDK_means++: Integrating Particle Swarm Optimization for Robust and Accurate K_means Initialization. **Advances in Artificial Intelligence and Machine Learning.** 2026;6(1):276. https://dx.doi.org/10.54364/AAIML.2026.61276

at random. This random selection causes the algorithm to converge to local minima, resulting in unstable clusters that do not accurately reflect the data distribution [5, 6]. To overcome these weaknesses, new techniques have been introduced, such as K_means++ [7, 8]. Other methods were proposed to derive the initial centroids directly from the data structure [9, 10]. One of those techniques is IDK_means, which produces better clusters than the standard K_means algorithm. The choice of centroids can yet be improved in a number of ways.

In this paper, we present IDK_means++, an extension of IDK_means that uses Particle Swarm Optimization (PSO) [11, 12] to improve centroid selection at the initial stages. The algorithm is tested on multiple datasets using Euclidean distance [13, 14] and cosine similarity [15, 16]. To assess the results, the Silhouette Coefficient [17], Davies–Bouldin Index [18] and Calinski–Harabasz Index [19] are used. Section 2 of the following sections describes the clustering techniques used and the PSO technique. Section 3 introduces the new method, IDK_means++. Section 4 covers the experimental design for the planned experiments. Section 5 covers the experiments and discusses the results. The study is concluded in Section 6 with recommendations for further investigation.

2. CLUSTERING AND PSO TECHNIQUES

Clustering partitions unlabeled data into groups based on similarity [2, 3, 20]. Among many clustering techniques, K_means remains one of the most common because it si simple and efficiency on large datasets [21, 22]. This section describes the K_means and IDK_means techniques [9, 10] and the Particle Swarm Optimization (PSO) [11]. All of those will form the new technique, IDK_means++.

2.1 K_means Algorithm

The K_means algorithm, introduced initially by MacQueen in 1967 [4], is a partitioning method that minimizes the sum of squared distances between each data point and its designated centroid while dividing the data into K clusters. K_means selects K data points as centroids, then repeats the following steps:

- Each data point is assigned to the nearest centroid
- Centroids are recomputed as the average of the data points assigned to each cluster.

These steps repeat until the newly generated centroids become stable. K_means has two disadvantages: First, it depends on the initial centroid selection, and second, it requires specifying the number of clusters K in advance [5, 6]. Randomly selecting K data points as centroids can lead to unstable solutions. This has motivated the development of new clustering techniques with improved initialization processes, such as K_means++ [6], and other deterministic or metaheuristic-based approaches [23, 24].

2.2 IDK_means Algorithm

The IDK_means algorithm [9, 10] was introduced as an improvement to K_means, where it determines the initial centroid based on the dataset rather than selecting it randomly. FIGURE 1 shows the LCTO algorithm that clusters the dataset and is consequently used by IDK_means in FIGURE 2, to derive the centroids for the K_means algorithm.

- Convert the dataset into a vector format table
- Compute the sum of 1s for each column
- Remove the items with sum of 1s is equal to 1
- Repeat until all clusters belong to the same cluster
  - Select column with lowest sum
  - Create a cluster with the object found in the column
  - Remove the column

Figure 1: LCTO pseudocode

- Perform LCTO
- Divide the cluster generated by LCTO into k parts and generate k centroids of each part.
- Use the derived k centroids as initial centroids for k-means

Figure 2: IDK_means pseudocode

This approach has shown more stable and accurate results. IDK_means computing the new centroids by averaging the data points, which was the primary motivation for developing IDK_means++, where Particle Swarm Optimization [12] replaces the averaging process.

2.3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) [12] is a computational technique derived from the social dynamics of the social behavior of birds, employed to solve complex optimization problems. Potential solutions in PSO are regarded as particles moving in a multidimensional space. The particles renovate their positions and velocities based on Personal best (pbest), which is the best result uncovered by the particle so far, and based on the Global best (gbest), which is the best solution discovered by the entire swarm.

This method allows PSO to balance exploration and exploitation as it moves into the search space [1, 25]. PSO has been applied to clustering by treating cluster centroids as particles [26–28]. Compared to other techniques, PSO is simple, uses a small number of parameters, and shows fast convergence [29]. These characteristics make it a good base for the hybrid IDK_means++ approach introduced in this study.

3. PROPOSED TECHNIQUE IDK_MEANS++

Forming centroids by simple averaging of members of a preliminary cluster, as IDK_means does [9, 10], may not always produce the best results. Therefore, we introduce IDK_means++, which is a hybrid clustering method that augments the deterministic initialization of IDK_means with a Particle Swarm Optimization (PSO) stage [12]. PSO is employed to locate the most representative centroid that minimizes the distance to all records within a preliminary cluster. This hybrid method produces higher-quality initial centroids for K_means refinement.

3.1 Justification for using PSO

The K_means clustering algorithm [4] relies on selecting the initial centroids which will converge to local optima. For this reason, we believe that integrating the Particle Swarm Optimization (PSO) [11, 12] with clustering minimizes intra-cluster dispersion while maximizing inter-cluster separation. In the proposed IDK_means++ algorithm, PSO replaces the averaging of centroids used in IDK_means [9, 10]. This hybrid approach reduces the computational efficiency of K_means while benefiting from PSO's ability to escape poor local solutions.

We first give an overview of the new technique, then describe the complete algorithm step-by-step, discuss parameter choices, and finally summarize the expected advantages and limitations.

3.2 Overview

The proposed IDK_means++ has two main steps. The first step partitions the dataset into k clusters based on the transactions, as IDK_means does. The second step uses PSO to find a centroid that minimizes the cluster's fitness. FIGURE 3 shows the IDK_means++ pseudocode. This PSO uses the same distance/similarity metric as the outer experiment (Euclidean distance or cosine similarity), so when the outer experiment uses cosine similarity [15, 16], PSO also uses it. When it uses Euclidean distance [13, 14], PSO optimizes using Euclidean distance. After initialization, standard K_means runs to convergence from the PSO-chosen starting centroids. IDK_means++ therefore combines three ideas: First, it is deterministic, data-derived preliminary clusters (from IDK_means) to avoid random seeding. Second, it uses PSO search within each preliminary cluster to find a centroid that best represents the cluster under the chosen metric. Third, it uses standard K_means refinement to finalize cluster assignment and centroids.

- Perform LCTO
- Use PSO to generate the k centroids
- Use the derived k centroids as initial centroids for k-means

Figure 3: IDK_means++ pseudocode

3.3 Advantages and Limitations

PSO produces centroids by selecting the most representative seeds for each cluster. Experimental evidence shows that this method produces better Silhouette Coefficient [17], Davies–Bouldin Index [18], and Calinski–Harabasz Index [19], suggesting it performs better at clustering. However, PSO makes things more challenging, specifically when performing with large or high-dimensional datasets. But you can reduce this overhead by running PSO actions in parallel across clusters. PSO also has the disadvantage of requiring several hyperparameters. While the default settings usually work well, some datasets may require special tuning to achieve the best results.

3.4 Summary

IDK_means++ is an improvement on IDK_means [9, 10]. It adds a PSO technique [11, 12] to compute cluster centroids. This replaces the averaging of centroids used by IDK_means and K_means algorithms [4]. It also keeps costs down and makes the first solution much better by running PSO within each cluster instead of on the complete dataset. When used with K_means, this technique showed more accurate and reliable clusters.

4. EXPERIMENTAL DESIGN AND ANALYSIS

This section presents the datasets, similarity measures, and evaluation metrics used to evaluate the performance of IDK_means [9, 10] and the proposed IDK_means++. All experiments were conducted in the same environment to ensure a fair and consistent comparison. For each dataset, clustering was performed by varying the number of clusters k from 2 to 5, and each experiment was repeated using both the Euclidean distance [13, 14] and cosine similarity [15, 16]. The clustering quality was assessed using the Silhouette Coefficient [17], Davies–Bouldin Index [18], and Calinski–Harabasz Index [19], and the results were summarized in tables for each k.

4.1 Dataset

For the experiments in this paper, we have used a few datasets from the UCI Machine Learning Repository [30], specifically the Car Evaluation, Flare2 (also known as the Solar Flare dataset), and Mushroom. The Car Evaluation dataset consists of 1728 records with 6 attributes. The information in the dataset is related to structural and safety aspects of automobiles. The Flare2 dataset (a version of the Solar Flare dataset) consists of 1389 measurements of solar activity with 10 features that capture sunspot characteristics and flare class indicators. The Mushroom dataset has 8124 records, 22 mushroom attributes, and a binary class indicating edibility or toxicity. All datasets were downloaded from the UCI repository, and the details are shown in TABLE 1

For each dataset, clustering was applied sequentially for all k =2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12, and Silhouette Coefficient [17], Davies–Bouldin Index [18] and Calinski–Harabasz Index [19], were computed for the corresponding metrics of IDK_means [9, 10] and IDK_means++ under the different distance measures described in the next section.

Table 1: Dataset used for experiments.

Dataset	Instances	Attributes	Description
Car	1728	6	Car characteristics such as price, safety, and capacity
Flare2	1389	10	Sunspot and solar activity characteristic
Mushroom	8124	22	Physical features of a mushroom, such as cap shape, odor, and gill color

4.2 Similarity and Distance Measures

Two measures were adopted to assess the similarity between data points and cluster centroids: Euclidean and cosine similarity. Both were implemented to evaluate the consistency and robustness of the proposed technique across different data types.

4.2.1 Euclidean Distance

Euclidean distance is one of the most widely used metrics in clustering algorithms [13, 14]. For two d-dimensional vectors $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$, the Euclidean distance is defined as:

$$\mathrm{dist}_{\mathrm{Euc}}(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

It determines the distance between two points. In clustering, the Euclidean distance tends to work best when the clusters are compact and spherical.

4.2.2 Cosine Similarity

Cosine similarity measures vector orientation rather than magnitude [15, 16]. It is defined as:

$$\mathrm{sim}_{\cos}(x, y) = \frac{x \cdot y}{\| x \| \| y \|}$$

where x and y are the dot product, and $\| x \|$ and $\| y \|$ are the vector magnitudes. The corresponding cosine similarity is:

$$\mathrm{dist}_{\cos}(x, y) = 1 - \mathrm{sim}_{\cos}(x, y)$$

Cosine similarity is particularly suitable for high-dimensional, sparse data, where the presence or absence of items is more significant than their quantities.

4.3 Evaluation Metrics

4.3.1 Silhouette Coefficient

To assess clustering quality, the Silhouette Coefficient [17] was used as the standard metric. The Silhouette Coefficient is a comprehensive measure that captures both cluster cohesion and separation. For a given record x:

$$s(x) = \frac{b(x) - a(x)}{\max(a(x), b(x))}$$

where $a(x)$ is the average distance between the records in the same cluster and x, and b(x) is the smallest average distance between the records in other clusters and x. The overall Silhouette Coefficient is the average $s(x)$ across all points. The range of the values of $s(x)$ are between -1 and $+1$. For $s(x)$ close to 1, the points are well-clustered. For $s(x)$ near 0, the clusters overlap. For $s(x) < 0$, the points are possibly misclassified. Higher Silhouette scores indicate better clustering quality and well-separated groups.

4.3.2 The Davies–Bouldin Index (DBI)

DBI [18] is an internal clustering validation metric that measures the average similarity between every cluster and its most identical one. The similarity is the ratio of intra-cluster dispersion to inter-cluster separation. Smaller DBI values indicate better clustering performance. The equation for DBI is:

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{S_i + S_j}{M_{ij}}\right)$$

where S_i is the average distance between data points in cluster i and its centroid, and M_{ij} is the distance between the centroids of clusters i and j. Because DBI relies on distance computations, it is based on the distance metric.

4.3.3 The Calinski–Harabasz Index (CHI)

CHI [19] evaluates the clustering characteristic by comparing the distribution between clusters to that within clusters. Higher CHI values indicate better clusters, which have high inter-cluster separation and low intra-cluster variance. The CHI index is defined as:

$$\mathrm{CHI} = \frac{Tr(B_k)/k - 1}{Tr(W_k)/n - k}$$

where k is the number of clusters, n is the total number of data points, $Tr(B_k)$ is the trace of the between-cluster scatter matrix, and $Tr(W_k)$ is the trace of the within-cluster scatter matrix. The CH index is computed using squared Euclidean distances and is independent of other distance metrics, such as cosine or Jaccard similarity. For this reason, in our experiments, we use the CHI index as a complementary variance-based validation measure.

4.3.4 Why Both Metrics Are Used

If we use Silhouette alone, the assessment can be biased towards convex clusters; therefore, I have added DBI and CHI to provide greater weight to clustering quality. DBI focuses on how compact the clusters are, whereas CHI assesses the global variance structure. Using all those metrics gives a better evaluation of clustering performance.

4.4 Experimental Setup

For the PSO [12] component in IDK_means++, the swarm size S is set to 30, and the maximum iterations T is set to 200. Inertia weight W, which controls the momentum of particle movement, is set to 0.7; the most used range is 0.6-0.8, which is considered balanced behavior. The cognitive coefficient c_1, which is the particle's tendency to move toward its own best-known, is set to 1.5; a standard PSO setting is between 1.0 and 2.0. The social coefficient c_2, which is the particle's tendency to move toward the swarm's global best, is set to 2.5; most studies recommend a value between 1.5 and 3.0. Each run of the program is run for every K in [2 .. 12], and two results are produced, one for Euclidean distance and one for cosine.

5. EXPERIMENTS

The experiments were done with K ranging from 2 to 12. For each value of k, two independent experiments were conducted under the same conditions as IDK_means [9, 10] to ensure fair comparison. For each value of k, the algorithm iteratively optimized cluster assignments using the Particle Swarm Optimization (PSO) approach [12] integrated within IDK_means++ for centroid refinement. The PSO was configured with a fixed number of particles and iterations for consistency across datasets. The leading performance indicators used were the Silhouette Coefficient [17], Davies–Bouldin Index [18], and Calinski–Harabasz Index [19], to evaluate how well data points are assigned to their clusters [17].

5.1 Euclidean Distance Vs Cosine Similarity Experimental Results

FIGURE 4 – FIGURE 6, and TABLE 2 compare the results obtained with K_mean and IDK_means++ across all datasets. The experiments were run using cosine similarity and Euclidean distance.

TABLE 2 shows the results of the experiments on K_means and IDK_means++ using the Silhouette Coefficient, Davies–Bouldin Index (DBI), and Calinski–Harabasz Index (CHI) when Euclidean distance and cosine similarity were used. The results indicate that cosine similarity always gives better clustering quality than Euclidean distance. The table shows that cosine similarity performs better than Euclidean distance, with higher silhouette coefficient values and lower DBI values. As for the CHI values, the numbers are almost the same for both cosine similarity and Euclidean distance. Given these results, we will use cosine similarity in our experiments.

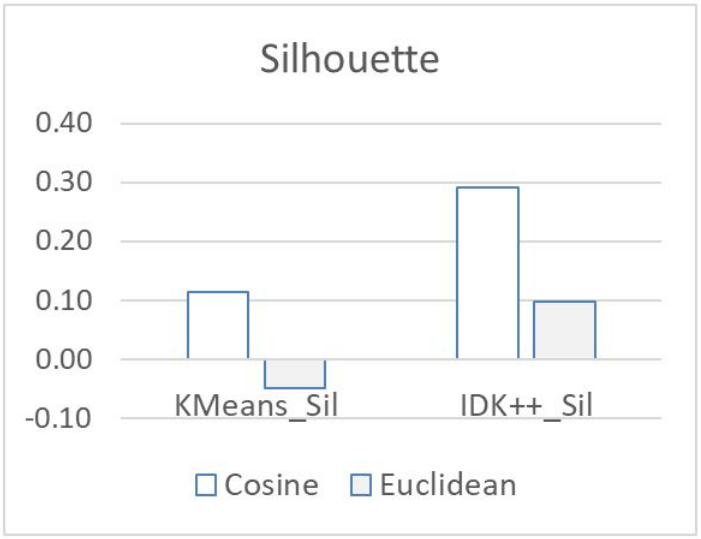

Figure 4: Euclidean vs cosine using Silhouette

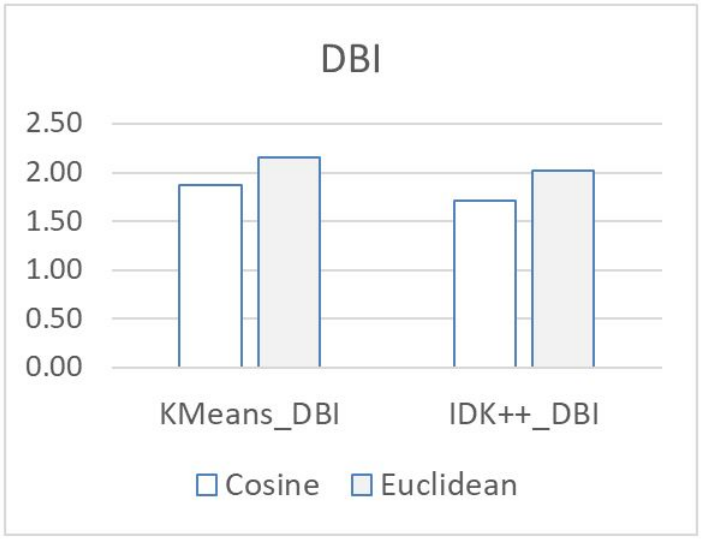

Figure 5: Euclidean vs cosine using DBI

5.2 Clustering Experimental Results using Silhouette Coefficient

FIGURE 7 and TABLE 3, compare IDK_means++ to the baseline using Cosine distance. The experiments are run on all datasets with different clusters.

Table 3 shows the Silhouette Coefficient for K_means, IDK_means, and IDK_means++ for values of K ranging from 2 to 12. While no single method performs best across all values of K, IDK_means++ performs well across most clustering configurations. For K greater than 5, IDK_means++ yields the highest Silhouette scores. The baseline K_means was unstable across different values of K, with

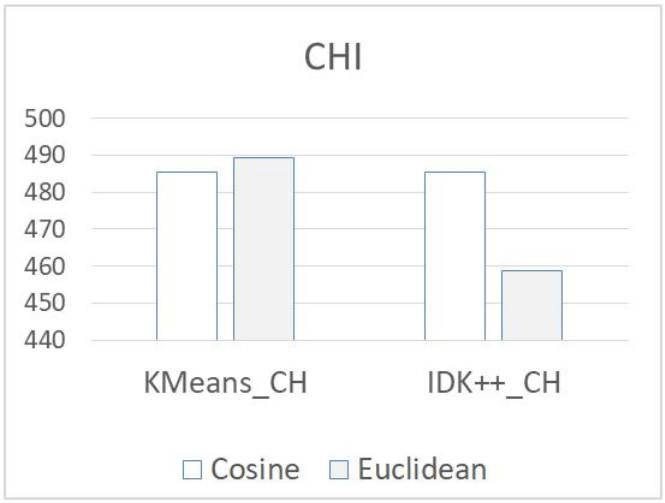

Figure 6: Euclidean vs cosine using CHI

Table 2: Euclidean vs cosine

	K_means	IDK_means++
Cosine/ Silhouette	0.115	0.292
Euclidean/ Silhouette	−0.049	0.098
Cosine/ DBI	1.875	1.709
Euclidean/ DBI	2.162	2.024
Cosine/ CHI	485	485
Euclidean/ CHI	489	458

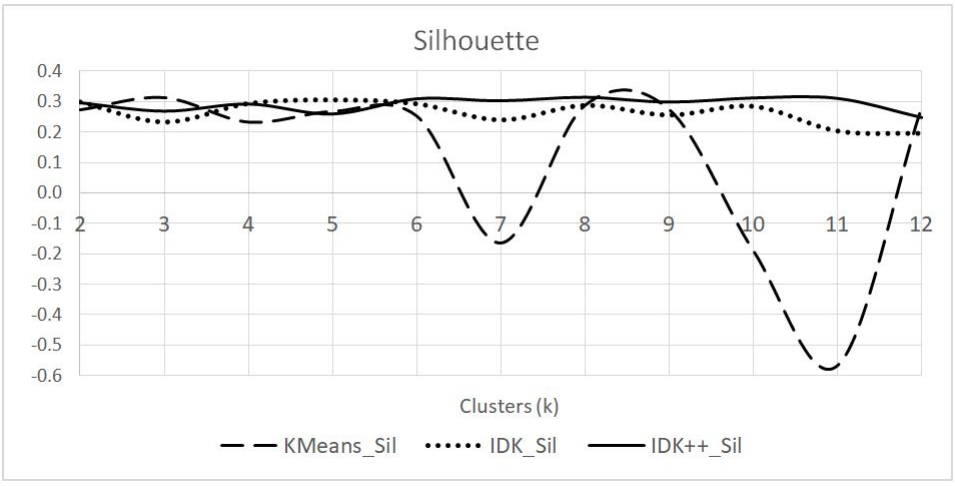

Figure 7: Graphs of Silhouette results for all clusters

Table 3: Table of Silhouette results for all clusters

K	K_means	IDK_means	IDK_means++	K	K_means	IDK_means	IDK_means++
2	0.273	0.299	0.297	8	0.288	0.288	0.315
3	0.313	0.233	0.269	9	0.275	0.256	0.299
4	0.233	0.294	0.292	10	−0.189	0.285	0.312
5	0.268	0.306	0.260	11	−0.567	0.204	0.311
6	0.253	0.293	0.310	12	0.282	0.196	0.248
7	−0.164	0.240	0.303				

some negative Silhouette scores across several configurations (e.g., K = 7, 10, 11), indicating poor cluster assignments. IDK_means improves stability compared to K_means; however, its performance remains generally lower than that of PSO-based IDK_means++. These results demonstrate that combining PSO with K_means improves clustering performance, particularly as the number of clusters increases.

5.3 Clustering Experimental Results using Davies–Bouldin Index

FIGURE 8 and TABLE 4, show the clustering performance of IDK_means and IDK_means++ for many clusters, where the experiments were done on various datasets.

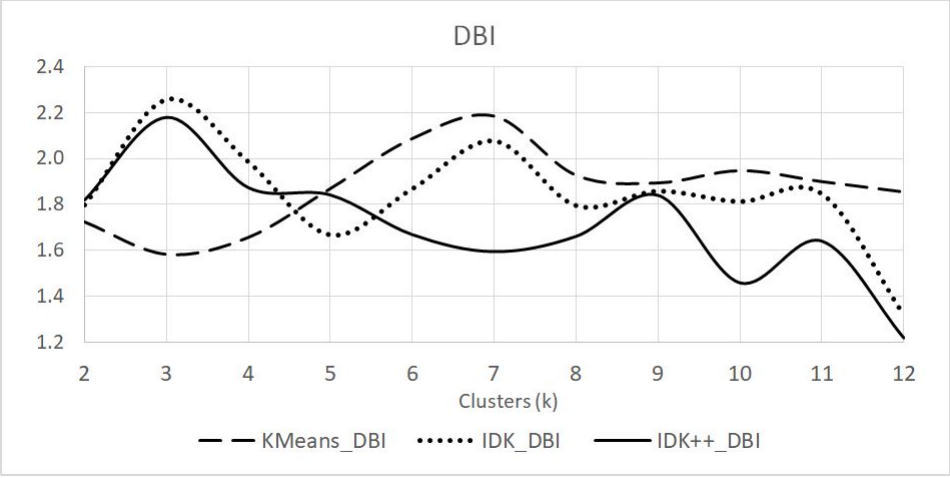

Figure 8: Graphs of DBI results for all clusters

TABLE 4 shows the Davies–Bouldin Index (DBI) values for K_means, IDK_means, and IDK_means++ with K varying from 2 to 12. Since lower DBI values indicate better cluster separation and compactness, the results show that IDK_means++ performed well mainly for larger numbers of clusters. From K = 6 to 12, DBI values were the lowest for IDK_means++ in most cases, outperforming K_means and IDK_means. While K_means performed well for small values of K, its performance degraded as K increased, whereas IDK_means++ gave well-separated clusters. These results demonstrate that adding PSO to K_means improves clustering performance.

Table 4: Table of DBI results for all clusters

K	K_means	IDK_means	IDK_means++	K	K_means	IDK_means	IDK_means++
2	1.724	1.798	1.819	8	1.927	1.795	1.661
3	1.583	2.256	2.179	9	1.894	1.858	1.840
4	1.657	1.985	1.873	10	1.947	1.813	1.459
5	1.869	1.667	1.841	11	1.899	1.846	1.641
6	2.087	1.868	1.669	12	1.855	1.320	1.220
7	2.184	2.076	1.595				

5.4 Clustering Experimental Results using Calinski–Harabasz Index

FIGURE 9 and TABLE 5, present the clustering results K in [2 .. 12] for various datasets.

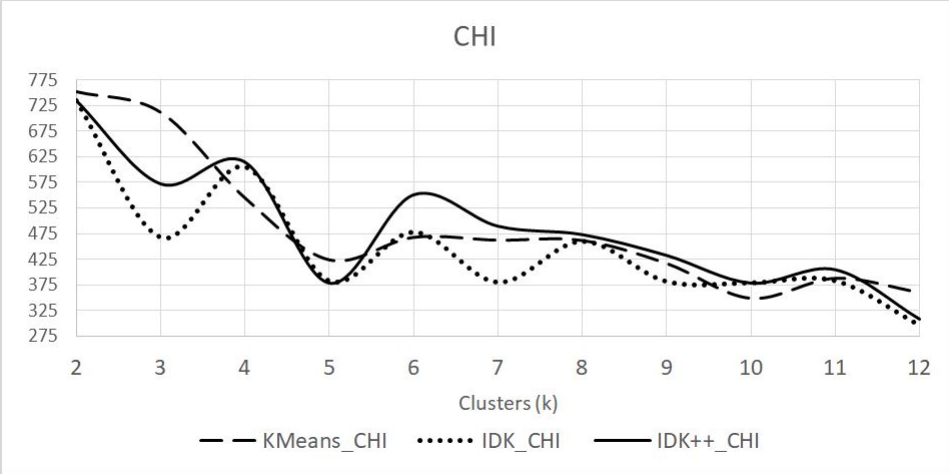

Figure 9: Graphs of CHI results for all clusters

Table 5: Table of CHI results for all clusters

K	K_means	IDK_means	IDK_means++	K	K_means	IDK_means	IDK_means++
2	752.193	734.832	734.835	8	461.744	460.194	473.228
3	712.057	468.019	572.298	9	416.965	381.582	432.563
4	544.964	605.358	614.761	10	348.886	379.106	378.599
5	423.902	382.774	378.624	11	387.895	383.124	404.777
6	467.867	478.036	550.907	12	360.309	296.125	308.245
7	462.339	380.023	489.574				

TABLE 5 shows the results of the experiments of K_means, IDK_means, and IDK_means++ using the Calinski–Harabasz Index (CHI). The experiments were run for all cluster sizes (K) from 2 to 12; higher CHI values indicate better clusters. We can see from the table that IDK_means++ achieves higher or similar CHI scores to both K_means and IDK_means, mainly for K = 4, 6, 7, 8, 9, and 11.

While K_means attains the highest score for minimal values of K, its performance declines rapidly as the number of clusters increases. Overall, the PSO-based centroid refinement in IDK_means++ yields more stable, well-separated clustering solutions across different values of K.

5.5 Overall Performance Trends Across K

For all evaluated values of K, the experimental results show that IDK_means++ provides more robust, stable clustering performance than both standard K_means and IDK_means. K_means achieves good results in a few cases for very small values of K; its performance degrades as the number of clusters increases. IDK_means performed better than K_means; however, averaging the centroids does not improve performance for complex cluster structures. As for IDK_means++, the results were much better for moderate to large values of K since it achieved higher Silhouette and Calinski-Harabasz scores and lower Davies-Bouldin values in most cases. The results show that PSO-based optimization improved centroid placement and clustering performance.

5.6 Advantages of the Proposed Method

The main advantage of using the new technique is that, by incorporating Particle Swarm Optimization, the proposed method overcomes K_means' sensitivity to random initialization and enhances IDK_means' averaging-based centroid selection. The results showed better centroid placement, improved cluster compactness, and stronger separation between clusters. Furthermore, IDK_means++ demonstrated better performance across different clusters and various datasets.

5.7 Limitations

Although IDK_means++ outperformed the baseline, it requires significant computational overhead due to its PSO-based optimization. This overhead may be a problem for very large datasets and even real-time applications. It should also be noted that the performance depends on PSO parameters, which may require tuning to achieve optimal results across different datasets.

6. CONCLUSION

This paper introduces IDK_means++, a hybrid clustering technique that outperforms the baseline by using the global optimization capability of Particle Swarm Optimization (PSO). The new technique overcomes some of the limitations of K_means, such as its sensitivity to random centroid initialization and the tendency to converge to suboptimal local minima [4, 5]. Replacing the average of centroids used in IDK_means [9, 10] with PSO [12] centroid optimization, the new technique performed significantly better. Experiments on many datasets and varying numbers of clusters (K) demonstrate that IDK_means++ outperforms both K_means and IDK_means. The performance was assessed using the Silhouette Coefficient [17],Davies–Bouldin Index [18] and Calinski–Harabasz Index [19]. The best results were achieved on high-dimensional and sparse transactional datasets. These results approve hpw efficient it is to combine PSO into centroid initialization. Future work

will evaluate IDK_means++ on additional data domains and investigate parallel and distributed implementations to improve scalability further.

Funding details

No grant was received for this research.

Conflict of Interest Statement

Regarding the publication of this article, the author assert that there is no conflict of interest.

7. ACKNOWLEDGEMENT

The Lebanese International University is acknowledged by the author for its ongoing support of innovation and research.

References

[1] Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. 4th ed. Netherlands: Elsevier. 2022.

[2] Jain AK. Data Clustering: 50 Years Beyond K-means. Pattern Recognit Lett. 2010;31:651-666.

[3] Aggarwal CC, Reddy CK. Data Clustering: Algorithms and Applications. 1st Edition. FL: CRC Press; 2014.

[4] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967:281-297.

[5] Lloyd SP. Least Squares Quantization In PCM. IEEE Trans Inf Theor. 1982;28:129-137.

[6] Arthur D, Vassilvitskii S, " K_means++: The Advantages of Careful Seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). 2007:1027-1035.

[7] Celebi ME, Kingravi HA, Vela PA. A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. Expert Syst Appl. 2013;40:200-210.

[8] Fränti P, Sieranoja S. K-Means Properties on Six Clustering Benchmark Datasets. Appl Intell. 2018;48:4743-4759.

[9] Yamout F. Enhancing the K-Means Algorithm Using Cluster Adjustment. In: International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, USA; 2023:307-311.

[10] Khansa HA, Yamout F, Salam F. Deriving Centroids for K-Means algorithm. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE. 2018:266-269. .

[11] Shi Y, Eberhart R. A Modified Particle Swarm Optimizer. In 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). IEEE. 1998:69-73.

[12] Kennedy J, Eberhart R. Particle Swarm Optimization. In Proceedings of ICNN'95-International Conference on Neural Networks. IEEE. 1995;4:1942-1948.

[13] Halkidi M, Batistakis Y, Vazirgiannis M. On Clustering Validation Techniques. J Intell Inf Syst. 2001;17:107-145.

[14] Leskovec J, Rajaraman A, Ullman JD. Mining of Massive Datasets. 3rd ed. Cambridge University Press. 2020.

[15] Srivamsi D, Deepak OM, Praveena MA, Christy A. Cosine similarity-based Word2Vec model for biomedical data analysis. In 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE. 2023:1400-1404.

[16] Veras MB, Sarker B, Aridhi S, Gomes JP, Macêdo JA, et al. On the design of a similarity function for sparse binary data with application on protein function annotation. Knowledge-Based Systems. 2022;238:107863.

[17] Rousseeuw PJ. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. J Comput Appl Math. 1987;20:53-65.

[18] Davies DL, Bouldin DW. A Cluster Separation Measure. IEEE Trans Pattern Anal Mach Intell. 1979;PAMI-1:224-227.

[19] Calinski T, Harabasz J. A Dendrite Method for Cluster Analysis. Commun Stat. 1974;3:1-27.

[20] Xu D, Tian Y. A Comprehensive Survey of Clustering Algorithms. Ann Data Sci. 2015;2:165-193.

[21] Ester M, Kriegel H, Sander J, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise. In: Proceedings of the 2nd International Conference Knowledge Discovery and Data Mining (KDD). ACM. 1996;96:226-231.

[22] Djouzi K, Beghdad-Bey K. A review of clustering algorithms for big data. In2019 International Conference on Networking and Advanced Systems (ICNAS). IEEE. 2019:1-6.

[23] Harifi S, Khalilian M, Mohammadzadeh J, Ebrahimnejad S. Using Metaheuristic Algorithms to Improve k-Means Clustering: A Comparative Study. Revue d'Intelligence Artificielle. 2020;34.

[24] Hospedales T, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence. 2021;44:5149-5169.

[25] Liu M, Zhu Q, Yun H, Chun H. Enhanced PSO-based clustering algorithm with hybrid approach for population replacement and empty cluster correction. Egyptian Informatics Journal. 2025;32:100814.

[26] Niknam T, Amiri B. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied soft computing. 2010;10:183-197.

[27] Naik A, Satapathy SC, Parvathi K. A comparative analysis of results of data clustering with variants of particle swarm optimization. International conference on swarm, evolutionary, and Memetic computing. Cham: Springer International Publishing. 2013:180-192.

[28] Yamout F, Rahal M, Ghaddar A. EPSO-SCR: Energy-Efficient Pso-Based Clustering Approach for Redundancy Reduction In WSNs. In 2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE. 2024;539-545.

[29] Ayubi S, Muyeba MK, Baraani A, Keane J. An algorithm to mine general association rules from tabular data. Information Sciences. 2009;179:3520-3539.

[30] https://archive.ics.uci.edu/