Black Hole Clustering: Gravity-Based Approach with No Predetermined Parameters

RESEARCH PAPER

BELAL K. ELFARRA
MAMOUN A. A. SALAHA
WESAM M. ASHOUR

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

Clustering is a fundamental technique in data mining and machine learning, aiming to group data elements into related clusters. However, traditional clustering algorithms, such as K-means, suffer from limitations such as the need for user-defined parameters and sensitivity to initial conditions.

This paper introduces a novel clustering algorithm called Black Hole Clustering (BHC), which leverages the concept of gravity to identify clusters. Inspired by the behavior of masses in the physical world, gravity-based clustering treats data points as mass points that attract each other based on distance. This approach enables the detection of high-density clusters of arbitrary shapes and sizes without the need for predefined parameters. We extensively evaluate BHC on synthetic and real-world datasets, demonstrating its effectiveness in handling complex data structures and varying point densities. Notably, BHC excels in accurate prediction of the number of clusters and achieves competitive clustering accuracy rates. Moreover, its parameter-free nature enhances clustering accuracy, robustness, and scalability. These findings represent a significant contribution to advanced clustering techniques and pave the way for further research and application of gravity-based clustering in diverse fields. BHC offers a promising approach to addressing clustering challenges in complex datasets, opening up new possibilities for improved data analysis and pattern discovery.

CORRESPONDING AUTHOR:
Belal K. ELFarra
Faculty of Engineering, Islamic University of Gaza, P.O. Box 108, Gaza, Palestine
fbelal@iugaza.edu.ps

KEYWORDS:
Clustering; gravity-based clustering; density-based clustering; machine learning; data mining

TO CITE THIS ARTICLE:
ELFarra, BK, Salaha, MAA and Ashour, WM 2024 Black Hole Clustering: Gravity-Based Approach with No Predetermined Parameters. *Data Science Journal*, 23: 27, pp. 1–13. DOI: https://doi.org/10.5334/dsj-2024-027

DOI: 10.5334/dsj-2024-027

I. INTRODUCTION

Clustering is a data analysis technique used to group similar data points together based on certain features or characteristics. It is used for pattern recognition, data compression, anomaly detection, recommendation systems, image segmentation, customer segmentation, and genomics analysis. However, it faces challenges, such as selecting the right number of clusters, handling irregular cluster shapes, scalability issues, sensitivity to outliers, and the need for appropriate evaluation methods. Researchers continually work on improving clustering algorithms to address these challenges and make clustering more effective for various applications.

However, commonly employed clustering algorithms like K-means and expectation maximization (EM) face challenges such as dependence on user-defined parameters and sensitivity to initial conditions. In high-dimensional data, determining the optimal number of clusters (k) in K-means can pose a particularly challenging task (Cai et al. 2023; Ghazal et al. 2021).

To overcome these limitations, density-based clustering (Ester, et al., 1996) techniques have emerged, defining clusters based on regions of high densities separated by regions of low densities. Among these techniques, gravity-based clustering stands out as a variant that exploits the concept of gravity to detect clusters. By treating data points as mass points that attract each other based on distance, gravity-based clustering forms high-density clusters where points are closely related (Huang et al. 2019; Kuwil et al. 2020).

The inspiration for gravity-based clustering stems from the role of gravity in the physical world, where it governs the behavior of masses in the universe (Cadiou et al. 2020). Researchers have leveraged this concept to develop innovative clustering algorithms capable of effectively identifying clusters in diverse datasets. Gravity-based clustering finds applications in various fields, including time series analysis, astrophysics, and data mining (Jankowiak et al. 2017), enabling the identification of clusters with arbitrary shapes and sizes, making it particularly valuable for datasets characterized by varying point densities.

This paper introduces the Black Hole Clustering (BHC) algorithm, a novel approach that harnesses the concept of gravity to classify unsupervised datasets and autonomously predict cluster numbers, eliminating the need for predefined parameters. BHC demonstrates robust performance in predicting cluster numbers and consistently achieves competitive accuracy rates. Comparative evaluations against established clustering methods consistently highlight BHC's superiority across various scenarios. When applied to real-world datasets from diverse domains, BHC consistently proves its effectiveness, emphasizing its reliability and potential for further research and practical applications. This contribution advances the field of clustering algorithms capable of handling complex data structures.

The remaining sections of this paper are organized as follows: The next section is dedicated to the literature review, where we begin with a discussion on the statement of the problem, followed by an exploration of related works. Subsequently, we delve into the methodology, starting with the proposed idea and then presenting the proposed algorithm. The experiments and results sections showcase the outcomes of our research. Finally, we conclude the paper by summarizing the key findings and contributions of our study, along with highlighting potential directions for future research.

II. LITERATURE REVIEW

A. STATEMENT OF THE PROBLEM

The problem addressed in this research is the development of a clustering approach that does not rely on predetermined parameters and can identify clusters with non-linear boundaries. The proposed solution leverages the concept of black holes to model the clustering of data points. The goal is to determine the number of clusters and perform clustering for each data point without using any parameters. The identification of the cluster center is a significant challenge, and the datapoints with max dens are suggested as the efficient cluster centroids. The objective function evaluates the contribution of every variable to achieve optimized clustering, and the centroids get relocated to find the optimum grouping, such that the data points within a cluster are closest to their centroid.

B. RELATED WORKS

In the field of clustering, Xu and Wunsch (Xu & Wunsch II 2005) provided a comprehensive review of various algorithms that aim to identify clusters in data. These algorithms can be classified into distribution-based, hierarchical-based, density-based, and grid-based approaches, with the choice depending on the characteristics and prior knowledge of the dataset (Xu et al. 2016; Liu, et al., 2007; Liu & Hou 2016; Louhichi et al. 2017). However, the challenge arises when dealing with big data, which is often heterogeneous and challenging to exploit.

Clustering methods offer a promising solution to tackle the complexities of big data. Density-based clustering methods, in particular, are widely used due to their ability to handle large databases and effectively handle noisy data (Ester et al. 1996; Hai-Feng et al. 2023). One such algorithm is DBSCAN, which has been extended to include variants such as OPTICS, ST-DBSCAN, and MR-DBSCAN (Ankerst et al. 1999; Birant & Kut 2007; He et al. 2011). While these methods perform well on spatial data, they have limitations when applied to high-dimensional data. Subspace clustering algorithms, like DENCLUE and CLIQUE, address this issue by detecting clusters within low-dimensional subspaces of high-dimensional data (Hinneburg & Keim 1998; Agrawal et al. 1998). However, DENCLUE suffers from slow execution time due to its hill-climbing method, which slows down convergence to local maxima.

Several new clustering algorithms have been proposed to address the limitations of existing methods. The Multi-Elitist PSO algorithm combines particle swarm optimization with clustering (Das et al. 2008), while PSO-Km integrates PSO with the K-means method (Dhawan & Dai 2018). An improved method (Elfarra et al. 2013) uses the concept of gravity to discover clusters in data, where each data point is attracted to the closest point with higher gravity. However, this method requires the specification of two predetermined parameters.

Another notable clustering algorithm is Density Peak Clustering (DPC), which identifies cluster centers based on their density and assigns points to clusters accordingly (Rodriguez & Laio 2014). Several improved versions of DPC, such as MDPC, PPC, FDP Cluster, and DPCG, have been proposed (Cai et al. 2018; Ni et al. 2019; Yan et al. 2016; Xu et al. 2016). However, these algorithms tend to select high-density points as initial cluster centers, which may lead to incorrect assignments or treat low-density points as noise.

The Shared Nearest Neighbors (SNN) algorithm addresses the issue of multiple-density clusters by considering the number of shared neighbors between objects (Jarvis & Patrick 1973). However, identifying clusters without significant separation zones may not be accurate, as the k-nearest neighbors based on distance may not be at the same level as the object (Ertöz et al. 2003). Although the SNN and DPC algorithms have been integrated into SNN-DPC, accurately identifying clusters without evident separation zones remains a challenge, and user input regarding the number of clusters or center points is often required (Liu et al. 2018).

In summary, the field of clustering algorithms offers various approaches to address the challenges posed by big data. From density-based methods like DBSCAN and OPTICS to subspace clustering algorithms like DENCLUE and CLIQUE, each approach has its strengths and limitations (Chen et al. 2022). Newer algorithms, such as Multi-Elitist PSO, PSO-Km, and gravity-based methods, strive to improve clustering performance. However, accurately identified clusters without evident separation zones and the need for user-specified parameters remain ongoing challenges in the field.

III. METHODOLOGY

A. PROPOSED IDEA

Our clustering approach utilizes the concept of black holes to model data points, akin to the gravitational force exerted by black holes in space. In our approach, we designate prototypes as black holes that attract nearby data points. Each data point generates gravity for each link between itself and any data point that is identified as its nearest neighbor. By selecting prototypes with the highest gravity, we attract the nearest data points, which, in turn, pull their nearest data points towards the prototypes, resulting in the formation of clusters.

The gravity of a data point X is determined by the number of data points Y that consider X as their nearest neighbor. Our objective is to determine the optimal number of clusters and classify each data point without relying on predefined parameters.

B. CHALLENGES AND NOVEL ALGORITHMS

Challenges may arise when two data points designate each other as their nearest neighbor, leading to the complete separation of these points from the cluster. This situation causes the cluster to split into two new clusters. Figure 1 provides a clear example of such a case, where datapoint 1 designates datapoint 2 as its nearest neighbor, and vice versa.

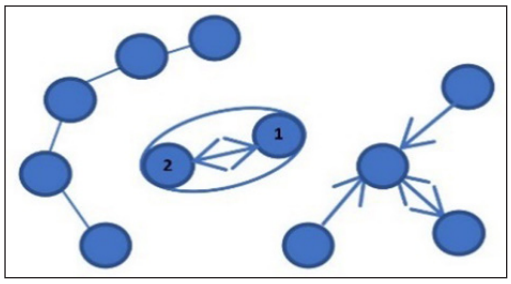

Figure 1 Reciprocal nearest neighbor relationship between datapoint 1 and datapoint 2.

Another challenge arises when a data point, X, has nearby neighbors that are closely grouped together, leading to the splitting of the cluster into two separate clusters. This situation is illustrated in Figure 2, where data point 1 identifies data point 4 as its nearest neighbor, while data point 2 identifies data point 3 as its nearest neighbor. As a result, data point 3 forms new connections with other data points that are unrelated to the neighbors of data point 4, leading to the formation of two isolated subgroups. Ideally, these isolated subgroups should be classified as a single cluster rather than multiple new subclusters.

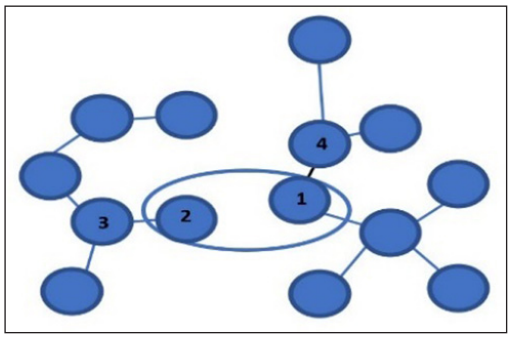

Figure 2 Formation of isolated subgroups within a cluster due to neighbor relations.

To effectively tackle these challenges without introducing additional parameters, we introduce two innovative algorithms, namely "Move_data_points" and "Shrink."

The Move_data_points algorithm is designed to optimize the relationships between data points. It achieves this by relocating the second, third, fourth, and fifth nearest neighboring points (referred to as fifth-level neighbors) either to the given data point or one of its adjacent points. This adjustment serves to refine the connections among neighboring points and enhance the overall clustering structure. For example, in Figure 2, datapoint 2 will have connections with datapoints 1, 3, and 4, resulting in a single cluster.

However, it's worth noting that this approach can occasionally lead to the unintended merging of clusters, particularly when noise points act as connectors between distinct clusters. In response to this challenge, we present the "Shrink" algorithm as a complementary solution. The Shrink algorithm operates by transforming data points to positions closer to their nearest neighbors if the distance between them exceeds twice the mean distance between neighboring points. This strategic relocation effectively brings noisy points into closer proximity to their respective nearest clusters while simultaneously disrupting any spurious connections that may exist between clusters.

Together, these two algorithms, Move_data_points and Shrink, work in tandem to refine the clustering results and mitigate potential challenges arising from noisy or poorly connected data points.

As a final step, we evaluate the effectiveness of our black hole clustering approach by comparing it to other widely used clustering algorithms such as K-means, DBSCAN, and OPTICS. The proposed black hole clustering approach offers a promising alternative method for clustering without the need for predefined parameters. This method excels at identifying clusters with non-linear boundaries and can be applied to various data types, including high-dimensional data. Further research can explore the efficiency and effectiveness of this method and its potential for real-world applications.

C. PROPOSED ALGORITHM

The BHC-Clustering algorithm starts by loading the dataset and creating a matrix called Z. It then calculates the Euclidean distance between each pair of data points in Z, resulting in a distance matrix called d_{ecu}. The distances are sorted in ascending order, and the indices of the sorted distances are stored in an $S_{indices}$ array. By using this matrix, we can determine the fifth-level neighbors datapoints for each row-datapoint. The next step is to apply the Shrink function to Z matrix using the distance matrix d_{ecu} and the $S_{indices}$ array. As mentioned before, we use the Shrink algorithm to eliminate or reduce the impact of noise datapoints. This step modifies the positions of the data points in Z matrix and creates a modified matrix called X. The algorithm continues by repeatedly iterating through the points in X matrix that have not been moved. It identifies the parent data point (P_{dp}), which is a datapoint that is marked as the nearest datapoint and as large as possible, with the highest recurrence in the first column of the $S_{indices}$ array, and then we move the associated data points in X using the "Move_data_points"-algorithm. This process continues until all data points have been moved. Finally, the algorithm returns the modified X matrix, which represents the clustered data points based on the BHC-Clustering approach.

Algorithm 1: BHC-Clustering

***Input*:** *A dataset Z* with d dimension

***Output*:** *A matrix X of clustered data*

foreach X_i & $X_j \epsilon Z$, *calculate Euclidian distance*:

$$d_{ecu} \leftarrow \sqrt{\sum_{k=1 \rightarrow d} \left(X_i[k] - X_j[k]\right)^2}$$

$S_{indices} \leftarrow$ *the indices of the sorted distances*

$X \leftarrow Shrink(Z, d_{ecu}, S_{indices})$

While *exists points not moved, loop:*

$P_{dp} \leftarrow argmax(i, count(S_{indices}[1], i))$

$X \leftarrow Move\_data\_points(X, P_{dp})$

return *X*

The "Move_data_points" algorithm operates on a dataset and performs the following steps without explicitly referring to individual data points:

For each data point in the dataset, designate a specific data point, referred to as P_{dp}, as its nearest neighbor.

Iterate through the dataset again, and for each data point encountered, set its nearest neighbor to P_{dp}.

Consider P_{dp} as the second, third, fourth, and fifth nearest neighbor for each data point in the dataset. Update the data points accordingly by assigning P_{dp} as their nearest neighbor.

Finally, return the modified dataset after applying these updates.

In summary, the "Move_data_points" algorithm operates on a dataset, establishing P_{dp} as the nearest neighbor for each data point, and extends this association to the second, third, fourth, and fifth nearest neighbors. The algorithm then updates the data points based on these assignments before returning the modified dataset.

The Shrink algorithm takes a dataset Z, a distance matrix d_{ecu}, and the indices of the nearest neighbors for each data point as input. It performs the following steps:

First, it calculates the mean distance between each data point and its nearest neighbor and stores these mean distances in a variable called mean_dist. Then, for each data point, it checks if the distance between the current data point and its nearest neighbor is greater than three times the mean distance. If this condition is true, it moves the current data point to the position of its nearest neighbor.

Algorithm 2: Move_data_points(X, P_{dp})

Input: *Parent data point P_{dp}, a dataset X*

Output: *A matrix X with repositioned datapoints*

foreach *j ∈ X such that P_{dp} is its nearest point*:

  foreach *i ∈ X such that j is its nearest point*:

    $i \leftarrow P_{dp}$

      foreach *point i in X such that j is in (2nd, 3rd, 4th, 5th) nearest point*:

    $i \leftarrow P_{dp}$

  $j \leftarrow P_{dp}$

return *X*

In summary, the Shrink algorithm adjusts the positions of data points based on their distances to their nearest neighbors. It ensures that any data point with a distance significantly larger than the mean distance is moved closer to its nearest neighbor. This process enhances the clustering results by bringing scattered points, which may act as outliers, closer to their neighbors.

Source code available on Google Colap at https://colab.research.google.com/drive/1gVMiNf4K-PyUCdqqFDQk-5xP_fogHkLaC.

Algorithm 3: Shrink (X, d_{ecu}, $S_{indices}$)

Input: *A dataset X to be shrink, a matrix d_{ecu} represents datapoints euclidian distances, $S_{indices}$ indices of nearest datapoints for each one*

Output: *A matrix X of shrunken data*

Compute the mean distance between each point in X and its nearest point:

$$mean\_dist \leftarrow (1/|X|) * \Sigma_i \, min_j \, || X_i - X_j ||$$

foreach *j ∈ X*:

  If $d_{ecu}[j, S_{indices}[j, 0]] > 2 * mean\_dist$:

    $X[j] \leftarrow X[S_{indices}[j, 0]]$

return *X*

D. COMPLEXITY ANALYSIS

In Algorithm 1, BHC-Clustering, the primary factors contributing to time complexity are the nested loops used for distance calculations and data shrinking. Specifically, the time complexity is $O(n^2*d)$ due to these nested operations. While in the Move_data_points algorithm, it involves iterating over points and potentially reassigning them, resulting in a time complexity of $O(n^2)$. In the Shrink algorithm, the primary time-consuming tasks are computing mean distances and shrinking points. The overall time complexity is $O(n^2*d)$.

It's noteworthy that the dominant factor in the time complexity of these algorithms is the nested loops, which encompass iterating over the dataset and evaluating distances between data points. This leads to a combined time complexity of $O(n^2*d)$ for these algorithms.

In summary, the nested loops involved in dataset traversal and distance computations are the key contributors to the overall time complexity of these algorithms, resulting in a common time complexity of $O(n^2*d)$.

IV. EXPERIMENTS AND RESULTS

We conducted various experiments using synthetic and real-world datasets to demonstrate the effectiveness of BHC-Clustering and compare it against K-means, DBSCAN, OPTICS, and BIRCH algorithms. Synthetic datasets included both Gaussian and non-Gaussian data, while real-world datasets were also utilized. The experiments were performed using Python version 3.8 on a Windows 10 Education system with an Intel Core i5-10500H CPU running at 2.50GHz and 16 gigabytes of memory. The synthetic datasets were two-dimensional and had varying numbers of true clusters.

In these experiments, we employed BHC-Clustering to determine the optimal number of clusters and then compared the results with the aforementioned clustering methods. The goal was to evaluate the performance of each method in clustering the data and assess the effectiveness of BHC-Clustering in particular.

A. SYNTHETIC DATA SETS

We conducted our experiments using two synthetic data sets. The first data set contained two clusters, while the second data set consisted of 15 clusters. Initially, we applied our proposed method to automatically determine the number of clusters, and our approach successfully predicted the correct number of clusters.

To compare the performance of our proposed algorithm, BHC-Clustering, we also utilized several other popular clustering algorithms, namely K-means, DBSCAN, OPTICS, and Birch. We applied these algorithms to the data sets and evaluated their results against our proposed method.

Figures 3 and 4 depict the clustering results obtained from applying the mentioned algorithms to the 2D-synthetic data sets. Specifically, when utilizing K-means clustering (Figures 3(a) and 4(a)), the algorithm exhibited unsatisfactory performance across the datasets. It incorrectly merged half of each cluster with half of the others, resulting in inaccurate classifications.

Similarly, the application of DBSCAN (Figures 3(b) and 4(b)) showed unsatisfactory performance across the datasets, leading to an incorrect number of clusters. This resulted in the misclassification of data points and the formation of spurious clusters. Evidence of this can be observed from the presence of misclassified data points and the existence of scattered points that should have been grouped together in coherent clusters.

Likewise, OPTICS (Figures 3(c) and 4(c)) also demonstrated poor performance. It frequently led to an excessive increase in the number of clusters and caused the fragmentation of clusters into multiple smaller clusters in certain cases. As a result, OPTICS consistently produced inaccurate classifications across the datasets. On the other hand, the Birch algorithm yielded significantly better results in clustering, as evident in Figures 3(d) and 4(d). However, it classified the data set into 14 clusters instead of the expected 15 clusters.

To provide a comprehensive comparison, Figure 5 presents the performance of our proposed BHC-Clustering approach against the aforementioned algorithms. The results clearly demonstrate that our proposed method outperformed the other algorithms in terms of clustering accuracy and overall performance.

Our contribution lies in the accurate prediction of the number of clusters in the data. Determining the optimal number of clusters is a crucial step in clustering analysis, as it directly affects the quality of the results. Traditional clustering algorithms, such as K-means, often require the number of clusters to be specified in advance, which can be challenging, especially when

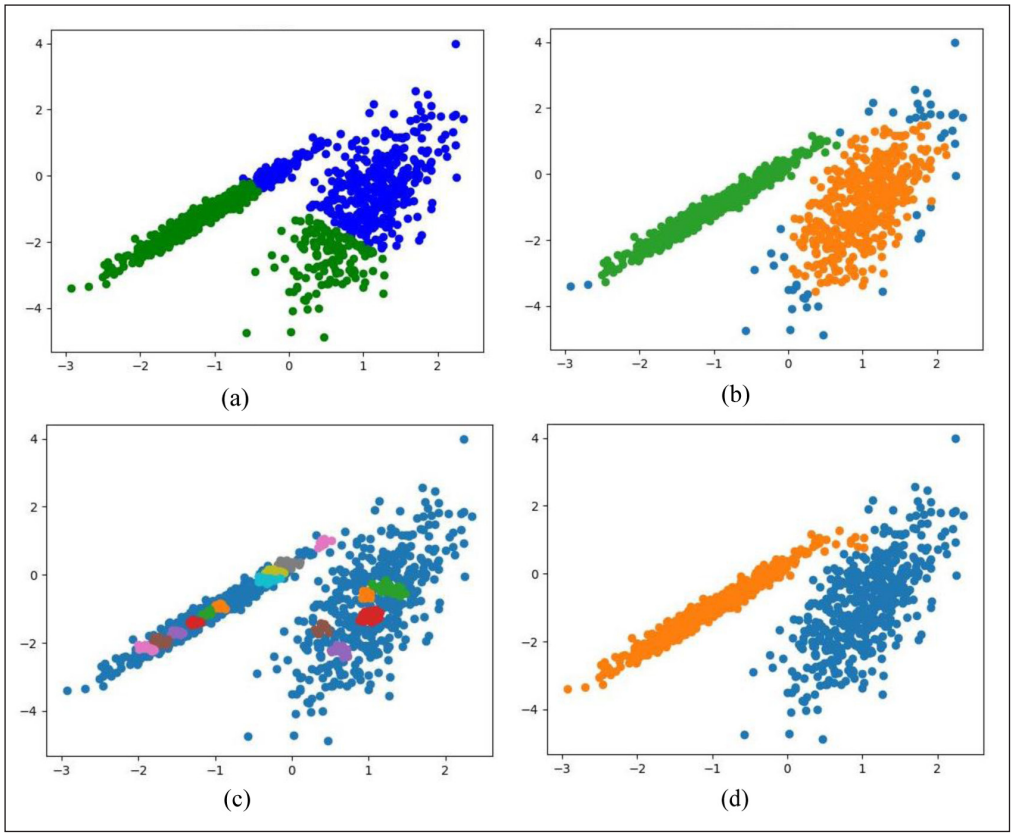

Figure 3 Comparison of clustering algorithms on 2D-synthetic data sets with two clusters **(a)** K-means clustering results, **(b)** DBSCAN clustering results, **(c)** OPTICS clustering results, and **(d)** Birch clustering results.

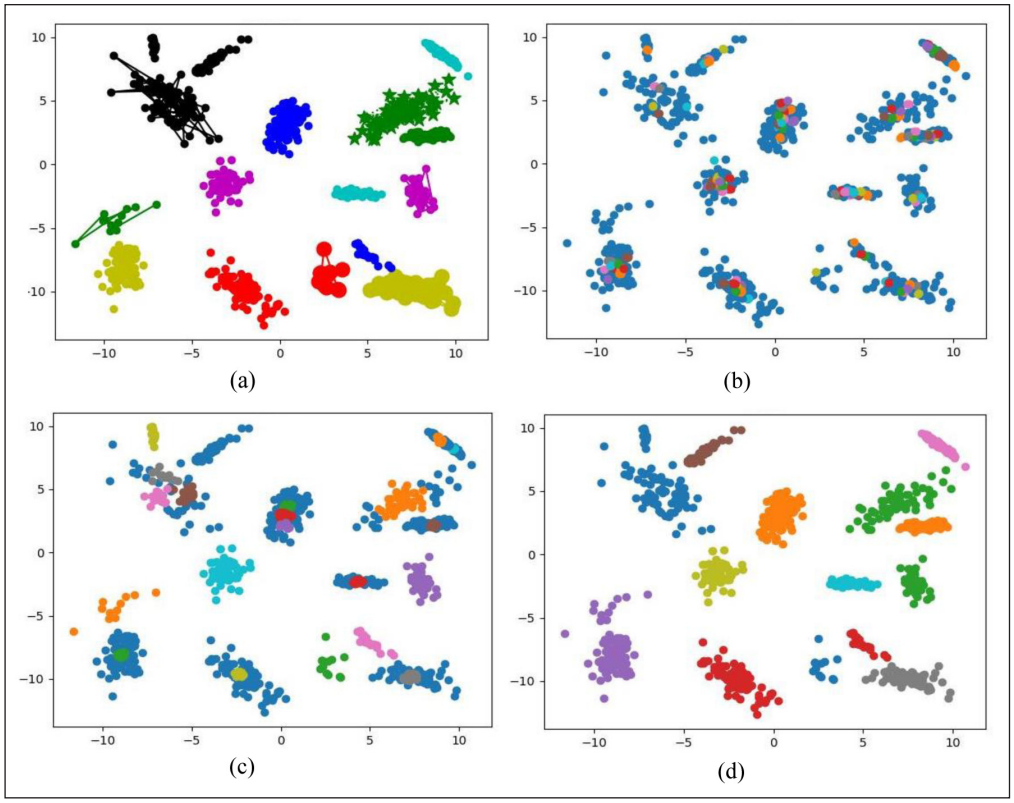

Figure 4 Comparison of clustering algorithms on 2D- synthetic data sets with 15 clusters **(a)** K-means clustering results, **(b)** DBSCAN clustering results, **(c)** OPTICS clustering results, and **(d)** Birch clustering results.

working with unfamiliar or complex datasets. This advancement in cluster prediction enhances the accuracy and reliability of clustering results. It enables us to uncover the underlying structure of the data more effectively. Moreover, our approach reduces the burden on users by automating the process of selecting the number of clusters, making it more accessible and efficient for various applications in data analysis and machine learning.

Overall, the experiments conducted on synthetic data sets provide valuable insights into the performance and suitability of BHC-Clustering for different clustering tasks.

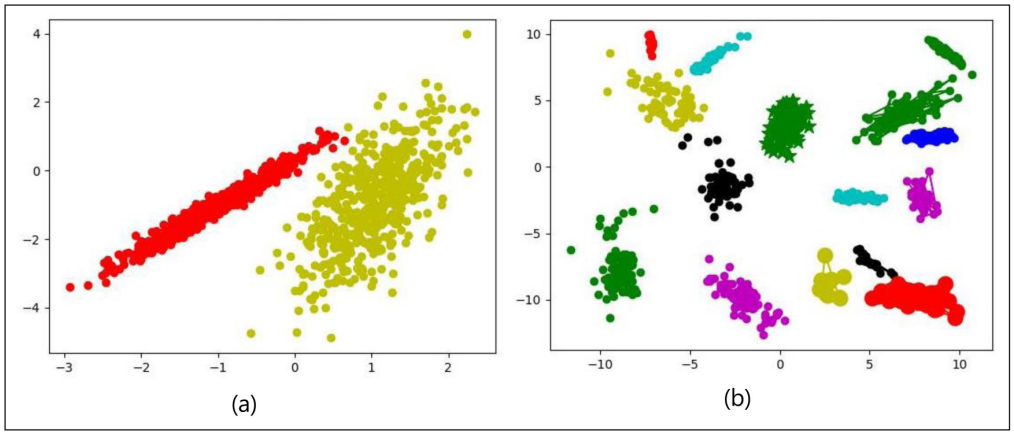

Figure 5 Performance comparison of BHC-Clustering against other algorithms.

B. REAL WORLD DATA SETS

In addition to the synthetic data sets, we also tested BHC-Clustering on real-world datasets to assess its performance. Table 1 summarizes the characteristics of these real datasets. They serve as practical benchmarks for evaluating BHC-Clustering in complex real-world scenarios.

Table 1 Characteristics of real-world datasets.

DATASET (DS)	NUMBER OF INSTANCES	CLASSES	DIMENSION
Iris Plants	150	3	4
Wine	178	3	13
Breast Cancer (BC)	569	2	30
Seeds-Dataset (SD)	210	3	7
Glass Identification (GI)	214	6	9

The first data set, Iris Plants, consists of 150 instances and belongs to three different classes. It has four dimensions, capturing various features of iris plants. This dataset is one of the earliest datasets used in the literature on classification methods and is widely used in statistics and machine learning. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other (Fisher 1988).

The second dataset, Wine, contains 178 instances and is divided into three classes. It is a high-dimensional dataset with 13 dimensions, representing different chemical properties of wines. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. This dataset is selected for its complexity, allowing us to assess the proposed BHC algorithm's performance in handling high-dimensional data with distinct attributes (Aeberhard et al. 1991).

The third data set, Breast Cancer (BC), is composed of 569 instances and has two classes. It is a particularly challenging data set due to its high dimensionality, with 30 different attributes related to breast cancer diagnosis. These attributes are derived from digitized images of fine needle aspirates (FNA) of breast masses, describing characteristics of cell nuclei within the images (Wolberg et al. 1995). The selection of this dataset was motivated by its high dimensionality and real-world relevance, rendering it a valuable testbed for our clustering algorithm in a healthcare context.

The fourth data set, Seeds-Dataset (SD), contains measurements of the geometrical properties of kernels belonging to three different varieties of wheat. A soft X-ray technique and GRAINS package were used to construct all seven real-valued attributes. The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa, and Canadian, 70 elements each, randomly selected for the experiment. High quality visualization of the internal kernel structure was detected using a soft X-ray technique. It is non-destructive and considerably

cheaper than other more sophisticated imaging techniques like scanning microscopy or laser technology. The images were recorded on 13x18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin (Charytanowicz et al. 2012).

The fifth dataset is the Glass dataset, providing a more realistic scenario with 214 instances and 10 attributes. Each instance in this dataset represents a unique piece of glass, and the class attribute indicates the type of glass based on the manufacturing process. There are six distinct types of glass, representing different manufacturing techniques (German 1987). The study of the classification of types of glass was motivated by a criminological investigation. At the scene of the crime, the glass left can be used as evidence, making this dataset particularly relevant for forensic and investigative applications.

These real-world data sets serve as practical benchmarks to assess the effectiveness and applicability of BHC-Clustering in diverse and complex real-world scenarios. The subsequent sections will present the experimental results and comparisons for each of these data sets.

The results presented in Table 2 demonstrate the accuracy rates of various clustering algorithms, namely BHC-Clustering, DBSCAN, OPTICS, and K-means, applied to five real-world data sets: Iris, Wine, Breast Cancer, Seeds-Dataset, and Glass.

Table 2 Predicted and actual number of classes and accuracy rates of clustering algorithms on real-world datasets.

DS	# OF CLASSES		ACCURACY %			
	PRED.	ACT.	BHC-CLUST.	DBSCAN	OPTICS	K-MEANS
Iris	3	3	90.7	66	67	24
Wine	3	3	62	33	67	16
BC	2	2	70.3	63	72	85
SD	3	3	63	28	18	26
GI	6	6	76.2	23.8	16	45

C. EVALUATION METRICS

To assess the BHC algorithm's effectiveness, we utilized confusion matrices as our primary evaluation tool. A confusion matrix is a valuable resource in clustering and unsupervised learning. It aids in gauging how effectively data points are grouped into clusters by comparing assigned cluster labels to actual cluster memberships. This matrix tallies the instances that were correctly and incorrectly assigned to clusters, offering insights into the algorithm's performance.

The diagonal values within the matrix represent correctly clustered instances. To compute the overall accuracy, we divided these diagonal values by the total number of instances. For visual clarity, Figure 6 illustrates the BHC algorithm's classification of the Iris dataset. The accompanying confusion matrix reveals that out of a total of 150 instances, 136 were correctly classified, resulting in an accuracy rate of 90.7%.

D. EXPERIMENTAL RESULTS AND EVALUATION

Notably, the proposed method achieved remarkable success in accurately predicting the number of clusters, as indicated by the column "Pred." It consistently achieved a perfect prediction rate, highlighting its significance in effectively determining the true number of clusters.

For the Iris data set, BHC-Clustering achieved an accuracy rate of 90.7%, outperforming the other algorithms. DBSCAN and OPTICS had relatively lower accuracy rates of 66% and 67%, respectively, while K-means performed poorly with an accuracy rate of only 24%.

In the case of the Wine data set, BHC-Clustering achieved a moderate accuracy rate of 62%, surpassing DBSCAN (33%) and K-means (16%) but falling behind OPTICS (67%). It is worth noting that none of the algorithms achieved high accuracy on this particular data set.

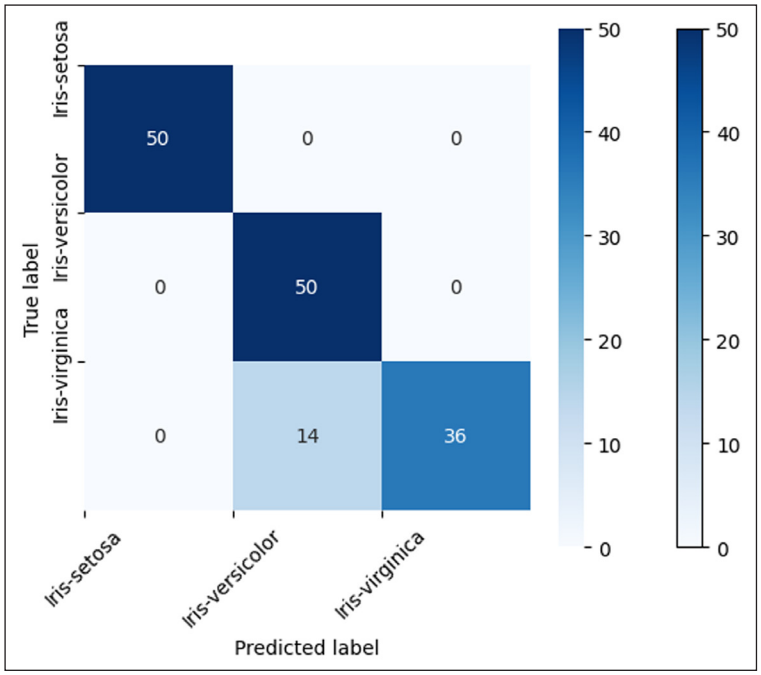

Figure 6 Confusion matrix for Iris dataset clustering using BHC algorithm.

In the case of the Wine data set, BHC-Clustering achieved a moderate accuracy rate of 62%, surpassing DBSCAN (33%) and K-means (16%) but falling behind OPTICS (67%). It is worth noting that none of the algorithms achieved high accuracy on this particular data set.

For the Breast Cancer (BC) data set, BHC-Clustering achieved a decent accuracy rate of 70.3%. DBSCAN (63%) and OPTICS (72%) also performed reasonably well, but K-means excelled with an accuracy rate of 85%.

For the Seed-dataset (SD), BHC-Clustering achieved a moderate accuracy rate of 63%. However, it outperformed the other algorithms in this case as well. DBSCAN and OPTICS had significantly lower accuracy rates of 28% and 18%, respectively, while K-means performed slightly better with an accuracy rate of 26%.

In the Glass Identification (GI) data set, BHC-Clustering achieved an accuracy rate of 76.2%, demonstrating its effectiveness in clustering this particular data set. DBSCAN and OPTICS had lower accuracy rates of 23.8% and 16%, respectively, while K-means performed relatively better with an accuracy rate of 45%.

Overall, the results suggest that BHC-Clustering exhibits competitive performance compared to the other algorithms in terms of clustering accuracy. However, the performance varies depending on the data set, indicating the importance of considering the characteristics and complexity of the data when selecting a suitable clustering algorithm. The proposed method's success in accurately predicting the number of clusters, demonstrates its potential for enhancing the clustering process.

V. CONCLUSION

The proposed BHC-Clustering method has been extensively investigated and applied to synthetic and real-world datasets. This approach utilizes the concept of black holes to attract nearby data points and form clusters. The method exhibits robust performance in accurately predicting the number of clusters and achieving competitive clustering accuracy rates.

Comparative evaluations against popular clustering algorithms, such as K-means, DBSCAN, OPTICS, and BIRCH, demonstrate that BHC-Clustering outperforms K-means and achieves comparable or superior results compared to DBSCAN and OPTICS. Although BIRCH shows promise, it has lower accuracy on one of the datasets.

Furthermore, the application of BHC-Clustering on real-world datasets, including Iris, Wine, Breast Cancer, Seeds-Dataset, and Glass, showcases its effectiveness across different domains. It demonstrates varying levels of performance, depending on the characteristics of the dataset. The findings emphasize the reliability and effectiveness of BHC-Clustering as a

clustering approach and encourage further research to refine the method, assess its efficiency, and explore its applicability in diverse applications. Overall, BHC-Clustering offers a promising alternative for clustering tasks, providing accurate cluster prediction and competitive clustering accuracy on a variety of datasets, including real-world scenarios.

However, it is important to acknowledge the challenge posed by the algorithm's complexity, which scales as $O(n^{2}*d)$, particularly when confronted with multiple noise points in real-world scenarios. The need for further research in this direction is evident. Future work in this field should focus on:

Complexity enhancement: Addressing the complexity of the BHC-Clustering method $O(n^{2}*d)$ to improve its efficiency and scalability, especially when dealing with large datasets and intricate cluster structures.

Noise handling: Developing advanced mechanisms to enhance the algorithm's ability to identify and manage multiple noise points effectively. This will bolster its applicability in noisy, real-world environments and ensure more efficient clustering outcomes.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Belal K. ELFarra orcid.org/0009-0008-8187-598X
Faculty of Engineering, Islamic University of Gaza, P.O. Box 108, Gaza, Palestine

Mamoun A. A. Salaha orcid.org/0009-0001-3870-3225
Faculty of Engineering, Islamic University of Gaza, P.O. Box 108, Gaza, Palestine

Wesam M. Ashour
Faculty of Engineering, Islamic University of Gaza, P.O. Box 108, Gaza, Palestine

REFERENCES

Aeberhard, M, Stefan, M and **Forina, M** 1991. Wine. *UCI Machine Learning Repository*. DOI: https://doi.org/10.24432/C5PC7J

Agrawal, R, Gehrke, J, Gunopulos, D and **Raghavan, P** 1998. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 27(2): 94–105. DOI: https://doi.org/10.1145/276305.276314

Ankerst, M, Breunig, MM, Kriegel, H P and **Sande, J** 1999. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record,* 28(2): 49–60. DOI: https://doi.org/10.1145/304181.304187

Birant, D and **Kut, A** 2007. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering,* 60(1): 208–221. DOI: https://doi.org/10.1016/j.datak.2006.01.013

Cadiou, E, Sarzi, M and **Dubois, Y** 2020. Gravitational clustering of stars and gas in galaxy simulations. *Monthly Notices of the Royal Astronomical Society*, 496(4): 4986–5001.

Cai, B, Huang, G, Yong, X, Jing, H, Huang, GL, Ke, D, et al. 2018. Clustering of multiple density peaks. In: *22nd Pacific–Asia Conference, PAKDD 2018*, Melbourne, Australia on 3–6 June 2018, 413–425. DOI: https://doi.org/10.1007/978-3-319-93040-4_33

Cai, J, Hao, J, Yang, H, Zhao, X, Yang, Y, et al. 2023. A review on semi-supervised clustering. *Information Sciences,* 632: 164–200. DOI: https://doi.org/10.1016/j.ins.2023.02.088

Charytanowicz, M, Jerzy, N, Piotr, K, Piotr, K, Szymon, L, et al. 2012. *Seeds. UCI Machine Learning Repository*. DOI: https://doi.org/10.24432/C5H30K

Chen, X, Wu, H, Lichti, D, Han, X, Ban, Y, Li, P, Deng, H, et al. 2022. Extraction of indoor objects based on the exponential function density clustering model. *Information Sciences,* 607: 1111–1135. DOI: https://doi.org/10.1016/j.ins.2022.06.032

Das, S, Abraham, A and **Konar, A** 2008. Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. *Pattern Recognition Letters,* 29(5): 688–699. DOI: https://doi.org/10.1016/j.patrec.2007.12.002

Dhawan, AP and **Dai, S** 2018. Clustering and pattern classification. In: Dhawan, AP, Huang, HK, and Kim, DS (eds.), *Principles and Advanced Methods in Medical Imaging and Image Analysis.* Singapore: World Scientific. pp. 229–265. DOI: https://doi.org/10.1142/9789812814807_0010

Elfarra, BK, El Khateeb, TJ and **Ashour, WM** 2013. BH-centroids: A new efficient clustering algorithm. *Work,* 1(1): 15–24. DOI: https://doi.org/10.14257/ijaiasd.2013.1.1.02

Ertöz, L, Steinbach, M and **Kumar, V** 2003. Finding clusters of different sizes shapes and densities in noisy high dimensional data. In: *The 2003 SIAM International Conference on Data Mining*, San Francisco, CA on 1–3 May 2003, pp. 47–58. DOI: https://doi.org/10.1137/1.9781611972733.5

Ester, M, Kriegel, HP, Sander, J and **Xu, X** 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: The 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon on 2–4 August 1996, pp. 226–231.

Fisher, RA 1988. Iris. *UCI Machine Learning Repository*. DOI: https://doi.org/10.24432/C56C76

German, B 1987. Glass Identification. *UCI Machine Learning Repository*. DOI: https://doi.org/10.24432/C5WW2P

Ghazal, T, Hussain MZ, Said, RA and **Nadeem, A** 2021. Performances of K-means clustering algorithm with Different Distance Metrics. *Intelligent Automation and Soft Computing*, 30(2): 735–742. DOI: https://doi.org/10.32604/iasc.2021.019067

Hai-Feng, Y, Xiao-Na, Y, Jiang-Hui, C, Yu-Qing, Y, et al. 2023. An in-depth exploration of LAMOST unknown spectra based on density clustering. *Research in Astronomy and Astrophysics*, 23(5). DOI: https://doi.org/10.1088/1674-4527/acc507

He, Y, Tan, H, Luo, W, Mao, H, Ma, D, Feng, S, Fan, J et al. 2011. Mr-dbscan: An efficient parallel density-based clustering algorithm using mapreduce. In: 2011 IEEE 17th International Conference on Parallel and Distrubuted Systems,Tianan, Taiwan on 7–9 December 2011, pp. 473–480. DOI: https://doi.org/10.1109/ICPADS.2011.83

Hinneburg, A and **Keim, DA** 1998. An efficient approach to clustering in large multimedia databases with noise. In: KDD '98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY on 27–31 August 1998, pp. 58–65.

Huang, Y, Yang, H and **Zhang, L** 2019. A novel clustering algorithm based on gravity. *Journal of Ambient Intelligence and Humanized Computing*, 10(6): 2461–2470.

Jankowiak, M, Kaczmarek, M, Wozniak, M and **Wojciechowski, K** 2017. Gravity-based clustering of time series data. *Information Sciences*, 385-386: 52–64.

Jarvis, RA and **Patrick, EA** 1973. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C–22(11): 1025–1034. DOI: https://doi.org/10.1109/T-C.1973.223640

Kuwil, FH, Atila, Ü, Abu-Issa, R and **Murtagh, F** 2020. A novel data clustering algorithm based on gravity center methodology. *Expert Systems with Applications*, 156: 113435. DOI: https://doi.org/10.1016/j.eswa.2020.113435

Liu, P, Zhou, D and **Wu, N** 2007. VDBSCAN: varied density based spatial clustering of applications with noise. In: *International Conference on Service Systems and Service Management*, Chengdu, China on 9–11 June 2007, pp. 1–4. DOI: https://doi.org/10.1109/ICSSSM.2007.4280175

Liu, R, Wang, H and **Yu, X** 2018. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Information Sciiences*, 450: 200–226. DOI: https://doi.org/10.1016/j.ins.2018.03.031

Liu, W and **Hou, J** 2016. Study on a density peak based clustering algorithm. In: *7th International Conference on Intelligent Control and Information Processing (ICICIP)*, Siem Reap, Cambodia on 1–4 December 2016, pp. 60–67. DOI: https://doi.org/10.1109/ICICIP.2016.7885877

Louhichi, S, Gzara, M and **Ben-Abdallah, H** 2017. Unsupervised varied density based clustering algorithm using spline. *Pattern Recognition Letters*, 93: 48–57. DOI: https://doi.org/10.1016/j.patrec.2016.10.014

Ni, L, Luo, W, Zhu, W and **Liu, W** 2019. Clustering by finding prominent peaks in density space. *Engineering Applications Of Artifical Intelligence*, 85: 727–739. DOI: https://doi.org/10.1016/j.engappai.2019.07.015

Rodriguez, A and **Laio, A** 2014. Clustering by fast search and find of density peaks. *Science*, 344(6191): 1492–1496. DOI: https://doi.org/10.1126/science.1242072

Wolberg, W, Mangasarian, O, Street, N and **Street, W** 1995. Breast Cancer Wisconsin (Diagnostic). *UCI Machine Learning Repository*. DOI: https://doi.org/10.24432/C5DW2B

Xu, R and **Wunsch, D, II** 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3): 645–678. DOI: https://doi.org/10.1109/TNN.2005.845141

Xu, X, Ding, S, Du, M and **Xue, Y** 2016. DPCG: An efficient density peaks clustering algorithm based on grid. *International Journal of Machine Learnning and Cybernetics*, 9: 743–754. DOI: https://doi.org/10.1007/s13042-016-0603-2

Yan, Z, Luo, W, Bu, C and **Ni, L** 2016. Clustering spatial data by the neighbors intersection and the density difference. In: *UCC'16: 9th International Conference on Utility and Cloud Computing*, Shanghai, China on 6–9 December 2016, pp. 217–226.

TO CITE THIS ARTICLE:
ELFarra, BK, Salaha, MAA and Ashour, WM 2024 Black Hole Clustering: Gravity-Based Approach with No Predetermined Parameters. *Data Science Journal*, 23: 27, pp. 1–13. DOI: https://doi.org/10.5334/dsj-2024-027

Submitted: 10 June 2023
Accepted: 16 November 2023
Published: 07 May 2024

Data Science Journal is a peer-reviewed open access journal published by Ubiquity Press.