Vol. 6, No. 1 | January – June 2023

Harris' Hawks Optimization-Tuned Density-based Clustering

Muhammad Shoaib Omar[1], Syed Muhammad Waqas[2], Kashif Talpur[3*], Sumra Khan[4], Shakeel Ahmad[3]

Abstract:

Clustering is a machine learning technique that groups data samples based on similarity and identifies outliers with distinct features. Density-based clustering outperforms other methods because it can handle arbitrary shapes of clustering distributions. However, it has a limitation of requiring empirical values for the cluster center and the nominal distance between the cluster center and other data points. These values affect the accuracy and the number of clusters obtained by the algorithm. This paper proposes a solution to optimize these parameters using Harris' hawks optimization (HHO), an efficient optimization technique that balances exploration and exploitation and avoids stagnation in later iterations. The proposed HHO-tuned density-based clustering achieves better performance as compared to other optimizers used in this work. This research also provides a reference for designing efficient clustering techniques for complex-shaped datasets.

Keywords: *Machine learning; density-based clustering; metaheuristic algorithm; Harris' hawk optimization; clustering.*

1. Introduction

Unlabeled data is being generated exponentially by today's information systems, which demands accuracy and high efficiency for analysis, in order to draw logical conclusions. Clustering is the most studied and broadly used learning technique of its kind. It endeavors to divide a dataset into numerous usually separate subsets where each one is called a cluster. Through this partition, each cluster may assimilate to some prospective categories, which the clustering algorithm is not aware of prior to working on it. Based on the diverse learning approaches, researchers have considered many types of clustering algorithms, based on different features. These methods can be characterized into density-based, portioning-based, hierarchy-based, model-based, and grid-based approaches [1].

A density-based clustering algorithm groups data points based on cut-off distance and minimum number of points in each radius. In 1996, an algorithm was proposed by Martin Ester et. al [2] as a clustering algorithm with arbitrary cluster shapes without specifying the number of clusters beforehand, and they named it density-based spatial clustering of applications with noise (DBSCAN). DBSCAN, by design, has several advantages when compared to its counterparts; as it does

[1] Department of Computer Science, Bahria University, Karachi, Pakistan
[2] Department of Computer Science, Muhammad Ali Jinnah University, Karachi, Pakistan
[3] Department of Science and Engineering, Solent University, Southampton, United Kingdom
[4] Department of Information Technology, Salim Habib University, Karachi, Pakistan
Corresponding Author: [kashif.talpur@solent.ac.uk]

not need to define the number of clusters as prior information, and it can work with datasets with varying densities. DBSCAN algorithm is a novel technique to deal with unlabeled data as it promises great success.

Initially, using DBSCAN, several past works have focused on clustering spatial data, however, its success has been noticed in various other data complexities [3,4,5]. Because it groups data points with respect to density, clusters in DBSCAN are considered to be the regions with dense data points, and these clusters are separated by the regions having low density, or noise. This core functionality of this algorithm makes it effective in identifying clusters with unusual arbitrary shapes, as well as, determining noise or outliers [6]. DBSCAN performs clustering with the help of two user-defined parameters: epsilon (ε) which is the cluster radius and minimum points (minPts) to be present in that radius. Despite its efficacy in clustering applications, DBSCAN faces same limitations as other clustering methods, such as the determination of optimal values for user-defined parameters (ε and minPts) is difficult and problem specific. This demands a significant amount of personal experience or several experimental trials, hence limiting its extensive use.

To address the said problem, different approaches have been proposed in the literature. Lai et. al [7] used multi-verse optimizer (MVO) algorithm for automatic selection of ε and minPts parameters. The authors claimed to have achieved improved clustering performance of DBSCAN while employed on three public datasets. In this research, the authors first improved MVO and then used it for finding the optimum range of values for the DBSCAN parameters. Researchers in [8] proposed the unsupervised patterns in multi-dimensional data, also known as data cubes, using genetic algorithm (GA). While tuning GA parameters (mutation and crossover), the researchers utilized fuzzy inference mechanisms: Mamdani's rules and Takagi-Sugeno's rules. When compared with the existing DBSCAN, they found the GA-tuned DBSCAN achieved better clustering quality on six datasets of data cubes, for OLAP (online analytical processing) purposes. Another successful application of metaheuristic algorithms on DBSCAN parameter optimization is done by Zhu et. al [9] where the authors used harmony search (HS) optimization algorithm for the similar purposes. The experiments of this new clustering scheme, on six clustering problems of varying complexity, showed superior performance than the canonical DBSCAN, based on Rand index (RI) and Jaccard coefficient (JC) evaluation metrics. These significant research efforts have developed and extended the applicability of DBSCAN by benefiting from optimization capability of metaheuristic algorithms that select more reasonable values of clustering parameters from an optimal range.

Metaheuristic algorithms have proved to be powerful optimization techniques, while implemented over a range of optimization problems. By mimicking intelligent behaviors from nature, researchers have developed efficient search mechanisms. The well-established metaheuristic algorithms are particle swarm optimization (PSO) [10], artificial bee colony (ABC) [11], genetic algorithms (GA) [12], and ant colony optimization (ACO) [13]. However, some of the latest counterparts have also been effective in research, such as grey wolf optimization (GWO) [14], Archimedes optimization algorithm (AOA) [15], honey badger algorithm (HBA) [16], and chaos game optimization (CGO) [17]. These and many others are used in data mining tasks [18], wireless sensors node localization [19], power flow optimization in smart grids [20], stock price prediction [21], etc. Most of these metaheuristic algorithms are based on collective intelligence displayed by a group of search agents, which makes them simple, efficient, and ready to deploy on any optimization problem. Harris' hawks optimization (HHO) is a recent induction in the paradigm of optimization methods, developed by Heidari et. al paradigm of optimization methods, developed by Heidari et. al [22]. HHO maintains a robust global search strategy by integrating balanced exploration and

exploitation tools. Its performance is rightfully validated by several research works, such as engineering design optimization [23], satellite imaging breakdown threshold optimization [24], drug design and discovery [25] and wireless sensor node localization [26], and many other constrained and unconstrained problems [7,27,28,29].

Keeping in view the outstanding global search ability of HHO and its effective exploration and exploitation strategies, we utilize it to improve clustering performance of DBSCAN. We optimize the DBSCAN parameters (ε and minPts) using HHO, since the manual parameter selection process is based on trial and error which is considered as an ineffective approach. The optimal parameters result in high clustering accuracy. The rest of the paper is organized as follows. The upcoming section explains about materials and methods like fundamentals of DBSCAN and HHO. Section \ref{sec:MaterialsAndMethods} elaborates on the methodology of HHO for DBSCAN parameter optimization. The experimental settings and results are discussed in Section \ref{sec:Results}, whereas Section \ref{sec:Conclusion} duly concludes the findings of this research.

2. Materials and Methods

2.1. DBSCAN Clustering Algorithm

DBSCAN (Density-based spatial clustering of applications with noise), introduced by Martin et al. in 1996 [2], is a density-based clustering algorithm, which identifies clusters by density of objects found in dense regions. The benefits of this technique are that it can identify clusters of arbitrary shapes, clusters within a cluster (nested clusters) as well as outliers (the points not belonging to any cluster). This algorithm is mainly based on two integer value parameters: ε and minPts, where ε is the maximum radius of the neighborhood and minPts are the minimum number of points in that radius. A user has to provide values of these two parameters based on some past experience or on hit and trial basis. These two parameters play an important role in accuracy of the clusters being identified by the algorithm. Let's say if ε is a larger number, it will include many points in a cluster. Some of the points might occur there, which may not actually belong to that cluster. On the contrary, if it is initialized with a small value, it may create more clusters as compared to the correct number of required clusters. The parameter minPts also acts in same manner. DBSCAN also identifies outliers, the points that do not belong to any cluster. There are some core definitions related to the DBSCAN, which are:

- ε-neighborhood: Points within the radius of the center point ε.
- Core point: A point within its radius has at least minimum points (minPts).
- Boundary point: A point that is not a center point and has less than minimum points in its neighborhood but at least has one center point within its radius.
- Noise points: Points that are neither midpoints nor boundary points and do not belong to any cluster.
- Direct-density reachable: If point *q* is within the radius and *p* is the core point, then point *q* is directly density accessible from point *p*.
- Density-Reachable: Point *q* is the density reachable of point *p* if point *q* is within the radius and point *p* is not a center point. Point *q* is connected to point *p* through other points (q_1, q_2, q_3, ... q_n).
- Density-Connectivity: If points *p* and *q* are densities accessible from center point *o*, then points *p* and *q* are connected to each other relative to point *o*.
- Cluster: Defined as the largest set of densely connected points.

The algorithmic steps of DBSCAN are illustrated in Algorithm 1.

Algorithm 1: Pseudo-code of DBSCAN

Procedure DBSCAN(Dataset X, ϵ, minPts)

Step-1: Explore all core points in the eld space X using ϵ and minPts and add it to the set of core objects. A core point is when $|N_\epsilon(x_i)| >$ minPts

Step-2: Select any core object from the set and make a cluster with other core points using ε and minPts until all core points are visited and no more core point is left.

Step-3: Add unvisited border points in the current cluster one-by-one. But border point cannot extend the cluster further as it is not a core point.

Step-4: If a point is neither core nor a border point, mark it as noise and put it in a separate set, and label it.

Step-5: When all border points are added to the cluster; pick another core point for the set and repeat above steps until a cluster is made.

return Labeled data space of cluster index.

End procedure

2.2. Harris' Hawks Optimization

In 2019, Ali Asghar et al. [22] presented an optimization technique after observing the hunting behavior of Harris' hawks, which is called Harris' hawks optimization (HHO). This algorithm exhibits the hunting behavior of the Harris' hawks. For these hawks, the primary tactic for hunting prey is the "raid". This strategy is also known as the "seven-kill strategy". In this strategy, sometimes the surprised prey is caught within a few seconds or otherwise, these attacks are kept in continuity till the prey is caught. This search approach may include several brief and smart dives from different directions within seconds and minutes to confuse the exhausted prey that it is now and then is caught by the attacking hawk.

The Harris' hawks can apply a variety of attacking styles keeping in view the hunting environment, escaping behavior, and energy of their prey. There may be some hard and soft besieges while attacking the prey till it is finally caught. These abruptly changing attacking behavior of the hawks are beneficial in a way that this strategy brings rabbits to a complete exhaustion and become defenselessness.

The HHO algorithm consists of three phases, in which these attacks are taken place. These are:

2.2.1. Exploration Phase

Firstly, in the exploration phase, hawks wait in the hunting field and use their extended and strong visual power to locate a prey (rabbit). Sometimes the rabbits are explored very easily and other times, they have to wait for hours to locate one. If the hawks are unable to locate any rabbit for a certain period of time, they change their place and use random perching in other locations to look for the prey. This strategy in the form of mathematical representation is depicted in Eq. 1:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| \text{ if } q \geq 0.5 \\ X_{rabbit}(t) - r_1|X_m(t) - r_3(UB - r_4LB)| \text{ otherwise} \end{cases} \quad (1)$$

where $X_{rabbit}(t)$ is the current optimal solution whereas the next possible solution in terms of hawk's position is denoted by X(t+1). X(t), $X_{rand}(t)$, and $X_m(t)$ show the positions of all hawks (total solutions-set), randomly selected hawks, and average positions, respectively. The lower and upper bounds of the search space are given by LB and UB, whereas r_1-r_4 represent random variables between [0-1].

2.2.2. The Transition from Exploration to Exploitation

This is the second phase of HHO algorithm, which takes place between exploration and exploitation. When a prey is found, then different exploitation behaviors are used to reduce its escaping energy so that some of the hawks are able to catch it easily. In HHO, it is called energy component, formulated in Eq. 2:

$$E = 2E_0\left(1 - \frac{t}{t_{max}}\right) \quad (2)$$

where initial energy of the rabbit is denoted by E_0, and t and t_{max} represent current and maximum number of iterations respectively.

2.2.3. Exploitation Phase

The hawks bout on their target using different attacking techniques as required in that situation. The situation depicts the escaping pattern of the prey and attacking style of the hawks, and energy of the rabbits as well. There are four likely strategies in the HHO algorithms. These are as depicted in Eqs. 3 - 6:

- **Soft besiege:** If chance of escape ≥0.5 and absolute energy ≥0.5:

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \\ \Delta X(t) = X_{rabbit}(t) - X(t), J = 2(1 - r_5) \quad (3)$$

where $\Delta X(t)$ is the difference between the current position of a hawk X(t) and rabbit X_{rabbit} and r_5 being a random variable.

- **Hard besiege:** If chance of escape ≥0.5 and absolute energy <0.5:

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \quad (4)$$

- **Soft besiege with rapid progressive dives:** If chance of escape <0.5 and absolute energy ≥0.5:

$$X(t+1) = \begin{cases} Y \text{ if } F(Y) < F(X(t)),\ Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \\ Z \text{ if } F(Z) < F(X(t)),\ Z = Y + S \times LevyFlight(D) \end{cases} \quad (5)$$

where D and S denote problem dimensions and a random vector of size D, respectively. LevyFlight(D) is Levy Flight function.

- **Hard besiege with rapid progressive dives:** If chance of escape <0.5 and absolute energy <0.5:

- $$X(t+1) = \begin{cases} Y \text{ if } F(Y) < F(X(t)),\ Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \\ Z \text{ if } F(Z) < F(X(t)),\ Z = Y + S \times LevyFlight(D) \end{cases} \quad (6)$$

The step-by-step procedure of HHO algorithm can be defined in Algorithm 2.

Algorithm 2: Pseudo-code of HHO algorithm

Procedure HHO(Dataset X, ϵ, minPts)

Initialization of population X_i, (i=1, 2,..., N)

While t< t_{max}

Calculate fitness values of hawks and best location of rabbit

For each hawk $X_i(t)$

If E≥1 **Then**

Exploration and position update via Eq. 1

Else

Exploitation and position update via Eqs. 3-6

End For

End While

return Optimum solution X_{rabbit}

End procedure

As can be seen in Algorithm 2, it first initializes the population and then enters the exploration phase using Eq. 1 until it finds some prey in the search space. It will keep on searching and will wait until it finds a prey. After locating the prey, it will transit from the exploration to the exploitation phase using Eq. 2. Then it will make some attacking movements using four strategies as mentioned using Eq. 3 to Eq. 6 called soft and soft besieges and soft/ hard besieges with progressive dives.

2.3. HHO-Tuned DBSCAN

In this research, HHO and DBSCAN are integrated, in order to enhance clustering capability of the DBSCAN algorithm. Here, DBSCAN is integrated with HHO in such a way that it achieves highest accurate values of ε and minPts. As can be seen from Algorithm 3 the DBSCAN algorithm is called within HHO. HHO provides the values of best-fit rabbit and its energy as candidate values of ε and minPts, which are evaluated and then passed to HHO. The provided values are

compared to each other and convergence of the HHO is analyzed. At the point where it converges at global minima and with lowest clustering error, it is our desired values for ε and minPts.

The HHO is used to search DBSCAN parameters for optimal clustering accuracy, as shown in Algorithm 3.

Algorithm 2: Pseudo-code of HHO-DBSCAN algorithm

Procedure HHO-DBSCAN(Solution Size N, Maximum Iterations t_{max})

Initialization of population X_i (i = 1, 2,..., N)

While $t<t_{max}$

For each $X_i(t)$

$x = X_i(t)$

$\varepsilon = x_1$ and MinPts = x_2

Calculate fitness of DBSCAN with given parameters

End For

For each hawk $X_i(t)$

If $E \geq 1$ Then

Exploration and position update via Eqs. 1 and 2

Else

Exploitation and position update via Eqs. 3-6

End For

End While

Return Optimum solution X_{rabbit}

End procedure

3. Experimental Results

To evaluate the HHO-tuned DBSCAN performance, we used a computing machine with a 1TB hard drive, 32 GB RAM, and a Core i7 processor, on a Windows 10 PC, the code was developed in MATLAB R2016a. For experiments, we used three real-world datasets namely Seed, Segment, and Iris. And, four synthetic datasets namely Flame, Path-based, D31, and Compound.

3.1. HHO-DBSCAN on Real-World Datasets

Three University of California machine learning datasets [30] are chosen: Iris has three classes and four attributes; Seeds has three classes and seven attributes; and Segment has seven classes and nineteen attributes. The results of HHO-DBSCAN are compared with improved multi-verse optimizer (IMVO2) from Lai et al. [7] which is used for searching best value for ε parameter. To evaluate IMVO2 search ability, the researchers chose 500 points at random from each interval [ε_{min}, ε_{max}] as the value of ε, and used it in DBSCAN. Whereas, our proposed method HHO is being more effective as the results indicated that it shows the highest accuracy in every case as compared to IMVO2. HHO-tuned DBSCAN achieves outstanding results, serving as a model for developing efficient clustering approaches for complex clustering-shaped datasets.

Table 1 presents comparison results of HHO-DBSCAN and other counterparts (metaheuristic improved versions of DBSCAN i.e., PSO-DBSCAN [31], ACO-DBSCAN [32], and IMVO-DBSCAN [7]) with respect to accuracy metric on real-world and synthetic datasets. It can be observed from the results that HHO-DBSCAN outperformed the other improved versions of DBSCAN, as it scores high in each evaluation. The indices of evaluation metric are almost near to 100\% in each case, which speaks of efficacy of the proposed method.

In Fig. 1, the trend of optimal accuracy of DBSCAN clustering for Seeds, Iris, and Segment is consistent with the tendency of clustering accuracy in their subsets, when minPts takes different values. Also, the optimal parameter minPts is the same, which are 18, 3, and 3 respectively in IMVO2. The optimal minPts is selected and then optimized for the parameter ε. The results are shown in Table 1, which indicates that the ε interval

Table 1. Accuracy comparison between HHO-DBSCAN and others based on optimized values for DBSCAN parameters.

Datasets	Classes	IMV02			ACO-DBSCAN			PSO-DBSCAN			HHO-DBSCAN		
		$minPts$	$[\epsilon_{min}, \epsilon_{max}]$	Accuracy	$minPts$	ϵ	Accuracy	$minPts$	ϵ	Accuracy	$minPts$	ϵ	Accuracy
Seeds subset	Kama, Canadian	18	[0.25177,0.25322]	80.00%	20	4.5	87.29%	25	5.49	90.05%	37	3.5	95.05%
Seeds	Rose, Kama, Canadian	18	[0.25264,0.25322]	70.48%	28	2.79	88.00%	36	2.07	87.52%	38	1.99	89.52%
Iris subset	Versicolor, virginica	3	[0.40001,0.41231]	74.00%	30	3.99	90.40%	32	4.01	90.48%	38	3.83	99.47%
Iris	Setosa, versicolor, virginica	3	[0.40001,0.41231]	80.67%	23	1.16	75.78%	16	5.00	80.44%	10	3.16	84.66%
Segment subset	2, 3,...,7	3	[0.16195,0.16296]	54.14%	37	1.48	80.12%	37	2.48	80.77%	38	1.53	76.42%
Segment	1, 2, 3,...,7	3	[0.16195,0.16296]	52.20%	19	10	82.71%	36	7.10	70.27%	31	5	66.02%
Results on Synthetic Datasets													
Compound	2	18	[0.25177,0.25322]	90.73%	15	0.59	80.76%	20	1.76	90.32%	37	3.5	96.07%
D31	31	18	[0.25264,0.25322]	96.10%	17	0.82	82.78%	19	1.23	95.34%	38	1.99	99.00%
Flame	6	3	[0.40001,0.41231]	87.19%	4	0.56	86.21%	22	1.63	96.45%	38	3.83	99.88%
Path-based	2	3	[0.40001,0.41231]	93.57%	5	0.23	89.62%	26	2.01	96.76%	10	3.16	95.33%

corresponds to the optimal clustering accuracy of all the samples in a dataset. Furthermore, the interval of ε corresponding to the optimal clustering accuracy of all the samples in the dataset overlaps with the range of ε examined by IMVO2 to its subset in Table 1. On the other hand, when the proposed technique HHO-DBSCAN is applied to Seed Subsets, Seeds, Iris Subsets, Iris, Segment Subsets, and Segments, the accuracy obtained is 95.05\%, 89.52\%, 99.47\%, 84.66\%, 76.42\%, and 66.02\%, respectively. The results show that by using known label samples to tune the DBSCAN parameters, we can mine unknown label samples, which is unattainable with supervised learning methods.

Table 2. Evaluation of HHO-DBSCAN on real-world datasets

Datasets	Classes	Purity	NMI	RI	ARI
Seeds subset	Kama, Canadian	0.9505	0.6915	0.7024	0.7099
Seeds	Rose, Kama, Canadian	0.8952	0.5778	0.5873	0.5822
Iris subset	Versicolor, virginica	0.9948	0.8542	0.8834	0.8906
Iris	Setosa, versicolor, virginica	0.8466	0.7195	0.7233	0.7063
Segment subset	2, 3,...,7	0.7642	0.9939	0.9799	0.9887
Segment	1, 2, 3,...,7	0.6602	0.8834	0.9014	0.9112

We can identify the optimal clustering accuracy of DBSCAN by incorporating HHO to optimize the parameters of DBSCAN. This helps us to choose a more suitable ε value for DBSCAN clustering. Fig. 1 depicts a graph of various datasets on IMVO2 and the proposed method with minPts ranging from 1 to 40 on the x-axis and accuracy on the y-axis. The graphs indicate accuracy of the proposed method is higher than IMVO2-DBSCAN, so we can conclude that our method is reliable, as it achieves the highest accuracy in every dataset. DBSCAN parameters determination and optimization for a dataset and its subset is presented in Table 2. In Fig. 1, the best DBSCAN clustering accuracy trend for Seeds, Iris, and Segment is consistent with the clustering accuracy trend of their subsets where minPts is provided with different values. Moreover, the optimal minPts parameters are the same, 18, 3, and 3 in IMVO2, respectively.

On the other hand, when the proposed technique HHO-DBSCAN is applied to Seed Subsets, Seeds, Iris Subsets, Iris, Segment Subsets, and Segments, the accuracy obtained is 95.05\%, 89.52\%, 99.47\%, 84.66\%, 76.42\%, and 66.02\%, respectively. The results show that by using known label samples to tune the DBSCAN parameters, we can mine unknown label samples.

It can be seen from Table 2 that purity is more than 90\% i.e., 0.9505 and 0.9948 in Seeds Subset and Iris Subset, which clearly speaks of the results validity. The NMI, RI and

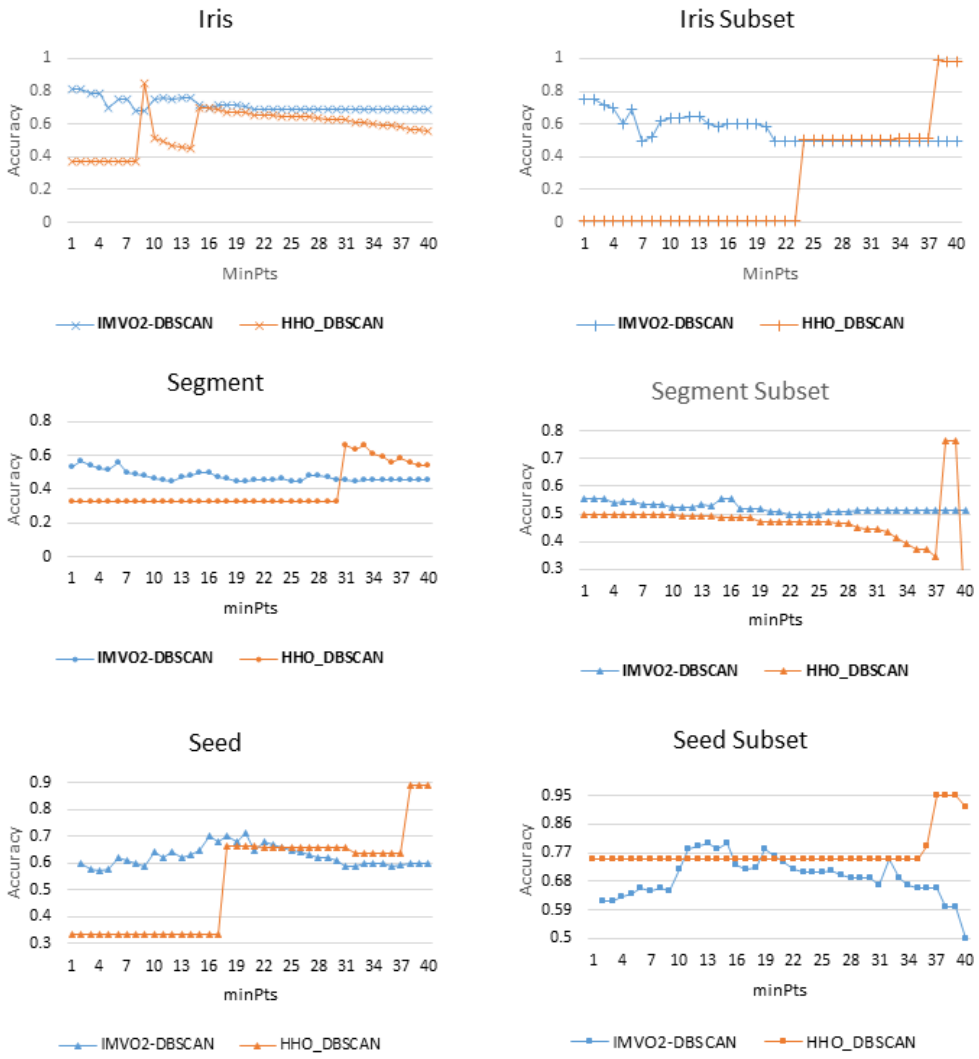

Fig. 1. Accuracy comparison between IMVO2-DBSCAN and HHO-DBSCAN based on optimized values for minPts parameter.

ARI of Segment Subset are 0.9939, 0.9799, and 0.9887, respectively. For the Segment, values of NMI, RI, and ARI are 0.8834, 0.9014, and 0.9112, which are also promising. However, NMI, RI, and ARI indices of Seeds and Seeds Subsets are not as expected, which shows that the method did not perform well on this dataset because it consists of 19 attributes and 7 classes of diverse natures. Since this dataset belongs to segments of a picture, hence it contains a high variance and dispersion among the values. The diverse nature of these values and increased number of classes, the results generated are not as expected.

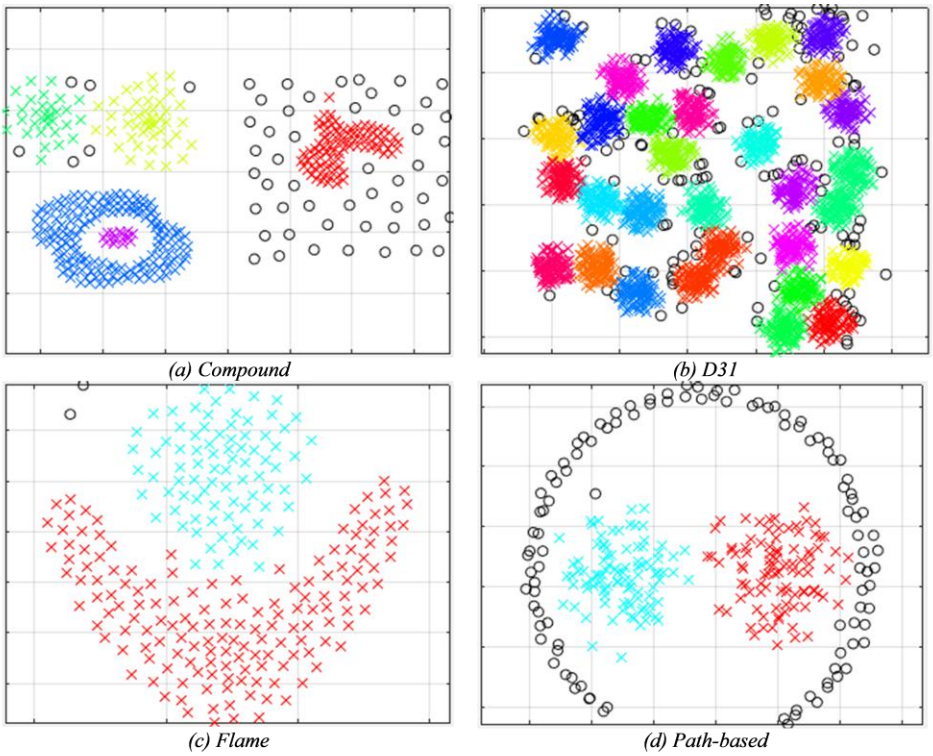

Fig. 2 Clusters of synthetic datasets

3.2. HHO-DBSCAN on Synthetic Datasets

The proposed method is then implemented on four synthetic datasets including Compound, D31, Flame, and Path-based. Here, Compound dataset has 788 points with 2 dimensions and 7 clusters. Whereas the D31 dataset which is made up of 31 similar 2-D Gaussian distributions. The Path-based dataset contains 300 points and has 2 dimensions and contains 6 clusters. The Flame dataset has 240 data points, 2 dimensions, and contains 2 clusters. The clustering results of HHO-DBSCAN are evaluated on synthetic datasets based on Purity, NMI, RI, and the ARI evaluation measures.

On Compound, HHO-DBSCAN achieves the purity of 0.9607, NMI value.9510, RI value 0.9863 and, the ARI value 0.9755. Overall, it achieves better results by using an efficient optimization method for finding ε parameter on different minimum parameter (minPts) ranging from 1 to 40. The clustering results of HHO-DBSCAN on Compound are shown in Fig.2(a), from which it can be inferred that the outcomes of projected solution are outperforming. The D31 is made up of 31 similar 2-D Gaussian distributions. While applying HHO-DBSCAN, we get the highest purity value of 0.9900 for the D31 dataset. We can also suggest from the Fig.2(b) that the proposed approach is reliable and produces the best results.

The clustering results of HHO-DBSCAN on the Flame dataset are shown in Fig.2(c) which clearly illustrate that this strategy for selecting DBSCAN parameters is the most efficient method available. HHO provides enhanced exploration and exploitation capabilities, which aids in avoiding solution stagnation in subsequent rounds. Table 2 presents that using the Flame dataset, we attain purity of 0.9988 and NMI of 0.9714, while the RI and ARI are 0.9949 and 0.9717. Fig.2(d) demonstrates clustering results on the Path-based dataset when HHO-DBSCAN is applied

to that data the highest results we achieve as Purity, NMI, RI, and ARI are 0.9533, 0.9144, 0.9438, and 0.9398 respectively.

From the overall results, it can be concluded that HHO is an efficient method for achieving the optimal values for the parameters for DBSCAN. The application of HHO to optimize DBSCAN parameters, allows us to discover the best clustering accuracy. The proposed algorithm optimally finds values of ε and minPts parameters to create clusters with maximum accuracy. The combination of DBSCAN and HHO algorithms enhanced the process of clustering with accuracy improvement.

4. Conclusion

DBSCAN is a clustering technique which has achieved appreciation from researchers and practitioners, mainly due to its ability to detect outliers and clustering efficiency for the data points distributed in arbitrary manner. However, manual selection of the DBSCAN parameters (ε and minPts) is a cumbersome job, as it requires past experience, as well as, several trials. This highly affects accuracy of the algorithm, since every dataset or clustering problem demands separate efforts to find best suitable DBSCAN parameters. This demands an effective approach to selecting these parameters in an automatic manner. Using metaheuristic algorithms, for this purpose, has overall resolved the problem, but with the existence of numerous optimization methods, the choice of the efficient one is crucial. This study applied an efficient metaheuristic algorithm HHO which has already generated comparatively better optimization results for a variety of optimization problems.

Both of the algorithms are successfully integrated to achieve the objective of automatically getting values of minPts and Eps. These values are metaheuristically calculated by optimization algorithm HHO. The results and comparison show that the calculated values are more accurate as compared to previous methods proposed y different researchers. The evaluation metrics used in this study indicate that the proposed method outperforms its counterpart methods. Moreover efficiency of the proposed model is also tested with other alike methods found in literature.

Based on insightful comparative analysis performed for HHO-DBSCAN and one of the recently introduced IMVO2-DBSCAN methods, it can be asserted that HHO is able to optimize DBSCAN parameters more efficiently, as it helps DBSCAN achieve clustering accuracy. Although the proposed method performs comparatively better for various datasets however, its performance is observed low in datasets having diverse data points. The lack of smoothness of data due to dispersion caused the proposed method to drop its efficiency, since it fails to identify the values of ε and minPts correctly.

Moreover, remaining in our limited scope, the proposed solution is for now applied to real-world and synthetic datasets however, it can also be applied to industry/engineering problems to testify to its performance. Moreover, since DBSCAN, sometimes fails to perform properly in large-scale and high-dimensional datasets, so in future DBSCAN and HHO can also be improved to work with large and high-dimensional datasets.

REFERENCES

[1] Bie, R., Mehmood, R., Ruan, S., Sun, Y., Dawood, H.: Adaptive fuzzy clustering by fast search and find of density peaks. Personal and Ubiquitous Computing 20, 785-793 (2016)

[2] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, p. 226-231. AAAI Press (1996).

[3] Ali, T., Asghar, S., Sajid, N.A.: Critical analysis of dbscan variations. In: 2010 international conference on information and emerging technologies, pp.1-6. IEEE (2010).

[4] Birant, D., Kut, A.: St-dbscan: An algorithm for clustering spatial{temporal data. Data & knowledge engineering 60(1), 208-221 (2007).

[5] Viswanath, P., Babu, V.S.: Rough-dbscan: A fast hybrid density based clustering method for large data sets. Pattern Recognition Letters 30(16), 1477-1488 (2009).

[6] Andrade, G., Ramos, G., Madeira, D., Sachetto, R., Ferreira, R., Rocha, L.: G-dbscan: A gpu accelerated algorithm for density-based clustering. Procedia Computer Science 18, 369-378 (2013).

[7] Lai, W., Zhou, M., Hu, F., Bian, K., Song, Q.: A new dbscan parameters determination method based on improved mvo. Ieee Access 7, 104,085-104,095 (2019).

[8] Rad, M.H., Abdolrazzagh-Nezhad, M.: Data cube clustering with improved dbscan based on fuzzy logic and genetic algorithm: Designing and improving data cube clustering. Information Technology and Control 49(1), 127-143 (2020).

[9] Zhu, Q., Tang, X., Elahi, A.: Application of the novel harmony search optimization algorithm for dbscan clustering. Expert Systems with Applications 178, 115,054 (2021).

[10] Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science, pp. 39-43. IEEE (1995).

[11] Karaboga, D., et al.: An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer . . . (2005).

[12] Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press (1992).

[13] Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on evolutionary computation 1(1), 53-66 (1997).

[14] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software 69, 46-61 (2014).

[15] Hashim, F.A., Hussain, K., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Applied Intelligence 51, 1531-1551 (2021).

[16] Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S., Al-Atabany, W.: Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. Mathematics and Computers in Simulation 192, 84-110 (2022).

[17] Talatahari, S., Azizi, M.: Chaos game optimization: a novel metaheuristic algorithm. Artificial Intelligence Review 54, 917-1004 (2021).

[18] Dhaenens, C., Jourdan, L.: Metaheuristics for data mining: survey and opportunities for big data. Annals of Operations Research 314(1), 117-140 (2022).

[19] Tsai, C.W., Tsai, P.W., Pan, J.S., Chao, H.C.: Metaheuristics for the deployment problem of wsn: A review. Microprocessors and Microsystems 39(8), 1305-1317 (2015).

[20] Papadimitrakis, M., Giamarelos, N., Stogiannos, M., Zois, E., Livanos, N.I., Alexandridis, A.: Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications. Renewable and Sustainable Energy Reviews 145, 111,072 (2021).

[21] Shahvaroughi Farahani, M., Razavi Hajiagha, S.H.: Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. Soft computing 25(13), 8483-8513 (2021).

[22] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. Future generation computer systems 97, 849-872 (2019).

[23] Gupta, S., Abderazek, H., Yildiz, B.S., Yildiz, A.R., Mirjalili, S., Sait, S.M.: Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems. Expert Systems with Applications 183, 115,351 (2021).

[24] Jia, H., Lang, C., Oliva, D., Song, W., Peng, X.: Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation. Remote sensing 11(12), 1421 (2019).

[25] Houssein, E.H., Hosney, M.E., Oliva, D., Mohamed, W.M., Hassaballah, M.: A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. Computers & Chemical Engineering 133, 106,656 (2020).

[26] Houssein, E.H., Saad, M.R., Hussain, K., Zhu, W., Shaban, H., Hassaballah, M.: Optimal sink node placement in large scale wireless sensor networks based on harris' hawk optimization algorithm. IEEE Access 8, 19,381-19,397 (2020).

[27] Rodriguez-Esparza, E., Zanella-Calzada, L.A., Oliva, D., Heidari, A.A., Zaldivar, D., Perez-Cisneros, M., Foong, L.K.: An efficient harris hawks-inspired image segmentation method. Expert Systems with Applications 155, 113,428 (2020)

[28] Ewees, A.A., Abd Elaziz, M.: Performance analysis of chaotic multi-verse harris hawks

optimization: a case study on solving engineering problems. Engineering Applications of Artificial Intelligence 88, 103,370 (2020).

[29] Fan, Q., Chen, Z., Xia, Z.: A novel quasi-reflected harris hawks optimization algorithm for global optimization problems. Soft Computing 24, 14,825-14,843 (2020).

[30] Dua, D., Graff, C.: UCI machine learning repository (2017). URL http://archive.ics.uci.edu/ml.

[31] Alswaitti, M., Albughdadi, M., Isa, N.A.M.: Density-based particle swarm optimization algorithm for data clustering. Expert Systems with Applications 91, 170-186 (2018).

[32] Jiang, H., Li, J., Yi, S., Wang, X., Hu, X.: A new hybrid method based on partitioning-based dbscan and ant clustering. Expert Systems with Applications 38(8), 9373-9381 (2011).