ISSN 2723-6471

Clustering the Unlabeled Data Using a Modified Cat Swarm Optimization

Deshinta Arrova Dewi[1,*], Tri Basuki Kurniawan[2], Mohd Zaki Zakaria[3], Sheeba Armoogum[4]

[1]*Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia*

[2]*Postgraduate Program, Universitas Bina Darma, Palembang, Indonesia*

[3]*Faculty of Computer Science and Mathematics, University Teknologi Mara (UiTM), Shah Alam, Malaysia*

[4]*Faculty of Information, Communication and Digital Technologies, University of Mauritius*

(Received: July 4, 2024; Revised: August 31, 2024; Accepted: September 11, 2024; Available online: September 23, 2024)

Abstract

This paper presents a modified version of the Cat Swarm Optimization (CSO) algorithm aimed at addressing the limitations of traditional clustering methods in handling complex, high-dimensional datasets. The primary objective of this research is to improve clustering accuracy and stability by eliminating the mixture ratio (MR), setting the counts of dimensions to change (CDC) to 100%, and incorporating a new search equation in the tracing mode of the CSO algorithm. To evaluate the performance of the modified algorithm, five classic datasets from the UCI Machine Learning Repository—namely Iris, Cancer, Glass, Wine, and Contraceptive Method Choice (CMC)—were used. The proposed algorithm was compared against K-Means and the original CSO. Performance metrics such as intra-cluster distance, standard deviation, and F-measure were used to assess the quality of clustering. The results demonstrated that the modified CSO consistently outperformed the competing algorithms. For example, on the Iris dataset, the modified CSO achieved a best intra-cluster distance of 96.78 and an F-measure of 0.786, compared to 97.12 and 0.781 for K-Means. Similarly, for the Wine dataset, the modified CSO reached a best intra-cluster distance of 16399, surpassing K-Means which recorded 16768. In conclusion, the modifications introduced to the CSO algorithm significantly enhance its clustering performance across diverse datasets, producing tighter and more accurate clusters with improved stability. These findings suggest that the modified CSO is a robust and effective tool for data clustering tasks, particularly in high-dimensional spaces. Future work will focus on dynamic parameter tuning and testing the scalability of the algorithm on larger and more complex datasets.

Keywords: Cat Swarm Optimization, Clustering Algorithm, Data Mining, Modified CSO, Unsupervised Learning, Process Innovation

1. Introduction

Data clustering plays a crucial role in data mining and machine learning, serving as a key method for unsupervised learning where the objective is to group a set of objects in such a way that objects within the same group (cluster) are more similar to each other than to those in other groups. Clustering methods are widely used in various fields such as bioinformatics, market segmentation, image recognition, and social network analysis [1], [2]. With the exponential growth of data, particularly high-dimensional and large-scale datasets, effective clustering techniques are becoming increasingly essential for extracting meaningful insights and patterns from complex data [3].

Traditional clustering algorithms, such as K-means and hierarchical clustering, have been extensively used due to their simplicity and efficiency [4], [5]. However, these methods have several limitations. For instance, K-means relies on predefined cluster numbers and is sensitive to the initial placement of centroids, which can lead to suboptimal partitions and poor convergence in complex, high-dimensional datasets [6]. Similarly, hierarchical clustering can be computationally expensive and may struggle with large datasets. These drawbacks highlight the need for more advanced clustering algorithms that can address the inherent challenges posed by real-world data, such as noise, non-convex cluster shapes, and varying density [7]. To address these limitations, optimization-based clustering methods have gained significant attention. Among these, swarm intelligence algorithms—such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO)—have shown promise in improving clustering performance by iteratively refining cluster solutions through global optimization processes [8]. One such algorithm that has emerged as a potential solution to clustering problems is CSO, an optimization algorithm inspired by the natural behaviors of cats [9]. While

***Corresponding author: Deshinta Arrova Dewi (deshinta.ad@newinti.edu.my)**
DOI: https://doi.org/10.47738/jads.v5i3.349

CSO has been successfully applied in various optimization tasks, its application to clustering remains underexplored, particularly with regards to its ability to handle complex, high-dimensional datasets.

This study seeks to fill this gap by developing a modified CSO algorithm specifically tailored for clustering tasks. The modified CSO algorithm aims to overcome the limitations of traditional clustering methods by introducing a more flexible, adaptive approach to clustering. Specifically, the proposed modification enhances the exploration and exploitation capabilities of the original CSO algorithm, allowing it to find more optimal cluster configurations across different types of datasets [10]. The primary objective of this research is to evaluate the performance of the modified CSO algorithm in comparison to well-established clustering methods such as K-means, DBSCAN, and hierarchical clustering. By applying these algorithms to classic datasets from the UCI Machine Learning Repository, including the Iris, Cancer, Glass, Wine, and CMC datasets, this study provides a comprehensive evaluation of their clustering effectiveness. Furthermore, the study employs a Sum of Squared Errors (SSE) fitness function to assess the quality of the clusters generated by each algorithm, providing an objective measure of performance [4].

In addition to developing and testing the modified CSO algorithm, this study aims to contribute to the broader field of clustering through three key aspects. First, we propose a novel modification to the CSO algorithm that enhances its applicability in solving complex clustering problems, particularly those involving high-dimensional and non-convex datasets. This modification enables better handling of diverse data structures by improving the balance between exploration and exploitation within the optimization process [9]. Second, a detailed comparative analysis is conducted to evaluate the modified CSO algorithm against traditional clustering methods. This analysis focuses on key performance metrics such as clustering accuracy, convergence speed, and robustness to noise, providing a thorough assessment of its effectiveness [5]. Third, the study evaluates the performance of the modified CSO algorithm across a variety of datasets, both low-dimensional (e.g., Iris dataset) and high-dimensional, real-world datasets (e.g., Cancer and CMC datasets), offering a broad perspective on its versatility and reliability [6].

The results of this study are expected to demonstrate that the modified CSO algorithm can outperform traditional clustering methods in terms of both accuracy and efficiency, particularly for challenging datasets that contain noise, outliers, or non-convex cluster shapes [7]. Additionally, the study will explore the potential for further improvements to the algorithm, setting the stage for future research in optimization-based clustering techniques. The remainder of this paper is organized as follows: Section 2 presents a comprehensive review of related works in the field of clustering and optimization algorithms. Section 3 describes the methodology used, including the details of the proposed modification to the CSO algorithm and the experimental setup. Section 4 discusses the results of the experiments, with a focus on comparative performance analysis. Finally, Section 5 concludes the paper by summarizing the key findings, contributions, and possible directions for future research.

2. Literature Review

Clustering algorithms have been extensively studied in the field of data mining and machine learning due to their broad applicability in various domains, including bioinformatics, computer vision, and market segmentation [11]. Over time, numerous clustering techniques have been proposed, ranging from traditional approaches such as K-means and hierarchical clustering to more advanced optimization-based algorithms that leverage metaheuristics for improved performance.

2.1. Traditional Clustering Algorithms

K-means is one of the most widely used clustering algorithms due to its simplicity and efficiency in partitioning data into kkk clusters [12]. The algorithm works by iteratively assigning data points to the nearest centroid and updating the centroids until convergence. Despite its popularity, K-means has several limitations, including its sensitivity to the initial selection of centroids and its assumption that clusters are spherical and evenly distributed. These assumptions often lead to suboptimal clustering results, particularly in the case of non-convex clusters or data with varying density [13]. Furthermore, K-means requires the number of clusters to be predefined, which may not be ideal in exploratory data analysis where the optimal number of clusters is unknown. Hierarchical clustering, another traditional approach, creates a tree-like structure (dendrogram) representing nested clusters based on data similarity [14]. This method can be either agglomerative (bottom-up) or divisive (top-down). While hierarchical clustering does not require the number

 ISSN 2723-6471

of clusters to be predefined, it is computationally expensive, particularly when applied to large datasets. Moreover, once a decision is made to merge or split clusters, it cannot be undone, making this method sensitive to errors in the early stages of clustering [15].

2.2. Density-Based Clustering Algorithms

To address the limitations of K-means and hierarchical clustering, density-based clustering algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) were introduced. DBSCAN defines clusters as regions of high data density, making it well-suited for identifying clusters of arbitrary shapes and handling noise in the data [16]. Unlike K-means, DBSCAN does not require the number of clusters to be specified a priori and can automatically identify the number of clusters based on the density of data points. However, DBSCAN struggles with datasets that contain clusters of varying density and is sensitive to the choice of parameters, such as the radius of the neighborhood (ϵ) and the minimum number of points required to form a cluster [17].

2.3. Swarm Intelligence in Clustering

In recent years, swarm intelligence algorithms have emerged as powerful tools for solving clustering problems by mimicking the collective behavior of social organisms, such as ants, bees, and birds [18]. Algorithms like PSO and Ant Colony Optimization (ACO) have been successfully applied to clustering tasks due to their ability to explore a large solution space and escape local optima. These algorithms work by simulating the behavior of a group of agents (particles or ants) that communicate and collaborate to find an optimal clustering solution. PSO, in particular, has been widely used in clustering due to its simplicity and strong convergence properties. Each particle in the swarm represents a potential solution (i.e., a set of cluster centroids), and particles update their positions based on their own experience and that of their neighbors [19]. The PSO algorithm has shown promise in improving clustering performance by avoiding the local minima that often plague traditional algorithms like K-means. However, like many optimization algorithms, PSO may suffer from premature convergence, particularly in complex, high-dimensional datasets.

2.4. CSO in Clustering

CSO is a relatively new swarm intelligence algorithm inspired by the behavior of cats during hunting and resting phases [20]. CSO operates in two modes: the "seeking mode," which mimics the cat's resting state while observing its surroundings, and the "tracing mode," which represents the cat's hunting behavior. This dual-mode approach allows CSO to balance exploration (seeking new solutions) and exploitation (refining existing solutions), making it a suitable candidate for clustering problems. Recent studies have explored the application of CSO in various optimization tasks, including function optimization, scheduling, and feature selection. However, its application in clustering remains underexplored, with only a few studies demonstrating its potential for solving clustering problems. Early results suggest that CSO can overcome some of the limitations of traditional clustering algorithms, such as sensitivity to initial conditions and difficulty in handling non-convex clusters. Nevertheless, further research is needed to fully understand its performance in complex, high-dimensional datasets, particularly when compared to other swarm intelligence algorithms like PSO and ACO [19], [20].

2.5. Concluding Remarks on the Literature

The review of existing clustering algorithms highlights the need for more advanced methods that can overcome the limitations of traditional techniques, such as sensitivity to initial conditions, assumption of spherical clusters, and difficulties in handling noise and non-convex data. Swarm intelligence algorithms, particularly CSO, offer a promising alternative due to their flexibility and adaptability. However, more research is needed to optimize these algorithms and evaluate their performance across a broader range of clustering tasks. This study contributes to the existing body of knowledge by proposing a modified Cat Swarm Optimization algorithm for clustering and comparing its performance against traditional and modern clustering methods. The results aim to provide insights into how the modified CSO can enhance clustering performance, especially in challenging datasets with complex structures.

 ISSN 2723-6471

3. The Cat Swarm Optimization and Its Modification

3.1. The Cat Swamp Optimization Method and Algorithm

Swarm Intelligence (SI)-based optimization algorithms are inspired by the collective behavior of animals, where individuals in a population, such as ants, bees, birds, or fish, interact with each other and their environment to solve complex optimization problems. These algorithms mimic the way these organisms utilize their environment and share information to optimize resource use [20]. One such SI-based optimization algorithm is the CSO algorithm, which is modeled after the behavior of cats. Originally developed by Chu and Tsai, CSO and its variants have been applied to various optimization challenges, proving to be effective in finding optimal or near-optimal solutions [21]. Several variations of CSO have been proposed over time. For example, Tsai et al. introduced a parallel structure for CSO, known as parallel CSO (PCSO), to improve computational efficiency. Additionally, they enhanced the performance of PCSO by incorporating the Taguchi method into the tracing mode, leading to the development of Enhanced Parallel CSO (EPCSO) [22]. These adaptations demonstrate the flexibility of the CSO algorithm in addressing a range of optimization problems, including clustering tasks, which are the focus of this study.

3.2. The Cat Swarm Optimization Algorithm

The initialization of the CSO algorithm begins with the creation of a population of cats, where each cat represents a potential solution in the M-dimensional solution space. Each cat is assigned a random position, represented as $x_i = x_i, 1, x_i, 2, \dots, x_i M)$, , where M is the number of dimensions in the solution space. In addition, each cat is assigned a random velocity $v_i = (v_i, 1, v_i, 2, \dots, v_i, M)$, which determines how fast the cat moves through the solution space. The velocity is constrained by a maximum velocity v_{max} to prevent excessive movement, ensuring that the search space is explored gradually and thoroughly [23]. In the mode assignment step, each cat is assigned to one of two modes: seeking or tracing. The number of cats assigned to the tracing mode is determined by a parameter called the mixture ratio (MR). This ratio typically takes a small value to keep the majority of cats in the seeking mode, promoting exploration of the solution space. In seeking mode, cats exhibit resting behavior, observing their surroundings and making small adjustments to their positions. In contrast, cats in tracing mode actively pursue the best solution found so far, simulating the behavior of a cat chasing prey [24]. The next step is the fitness evaluation, where the quality of each cat's position is measured using a fitness function. In clustering problems, this fitness function often reflects the within-cluster sum of squared errors (SSE), which is calculated as:

$$\mathrm{f}(x_i) = \sum_{k=1}^{K} \sum_{x_j \in C_k} ||x_j - C_k||^2 \quad (1)$$

In this equation, K represents the number of clusters, C_k is the set of data points in the k-th cluster, and C_k denotes the centroid of cluster k. The goal is to minimize the sum of squared errors, indicating that the data points within each cluster are closely grouped around their centroids. The cat with the lowest SSE (i.e., the best fitness value) is considered the best solution found so far [25]. In the seeking mode, cats make small adjustments to their positions to explore their surroundings. The adjustment to a cat's position is calculated using a random perturbation factor δ, which is applied to each dimension of the cat's position vector. This is mathematically represented as:

$$\mathrm{x_i^{new}} = \mathrm{x_i^{old}} + \delta \cdot \mathrm{rand}(-1,1) \quad (2)$$

This adjustment allows cats in seeking mode to explore different regions of the solution space without making drastic changes to their positions, encouraging thorough exploration before committing to any specific area. This mode is crucial for exploring potential solutions that may have been overlooked in earlier iterations. In contrast, the tracing mode involves cats actively moving toward the global best solution found so far. The velocity of each cat in tracing mode is updated based on the difference between its current position and the position of the global best solution. The update rule is given by:

$$\mathrm{v_i^{new}} = \mathrm{v} + \alpha \cdot (\mathrm{x_{best}} - \mathrm{x_i}) \quad (3)$$

$$\mathrm{v_i^{new}} = \mathrm{x_i^{old}} + \mathrm{v_i^{new}} \quad (4)$$

 ISSN 2723-6471

In this equation, α\alphaα is a learning factor that controls the rate at which the cat moves toward the best solution. The tracing mode focuses on exploitation, helping the algorithm refine its search around promising regions of the solution space.

The final step is the termination check, where the algorithm determines whether the optimization process should continue or stop. Common termination criteria include reaching a maximum number of iterations, achieving a fitness value below a predefined threshold, or observing minimal improvement over a series of iterations. If the termination criteria are not met, the algorithm repeats the seeking and tracing processes, reassigning cats to modes and updating their positions until the criteria are satisfied [23].

The Cat Swarm Optimization algorithm exhibits several key features that contribute to its effectiveness. Each cat represents a decision variable, with its position in the solution space representing a potential solution to the optimization problem. As the algorithm progresses, each cat updates its position based on its mode—seeking or tracing—allowing for a dynamic balance between exploration and exploitation. The fitness function evaluates the quality of each solution, typically based on its distance from the optimal target, which in clustering is the distance between data points and their cluster centroids. The algorithm continuously tracks the best solution (cat) throughout the iterations, ensuring convergence to the optimal solution by the time the termination criteria are met. To further illustrate the key components and operational characteristics of the CSO algorithm, table 1 summarizes the primary attributes and decision variables that govern the algorithm's behavior. These characteristics highlight the core elements involved in solution generation and fitness evaluation, which are critical to understanding how the CSO algorithm functions in an optimization context.

Table 1. The CSO Characteristics

General Algorithm	Cat Swarm Optimization (CSO)
Decision Variable	Cat’s position in each dimension
Solution	Cat’s position
Old Solution	The old position of the cat
New Solution	A new position of the cat
Best Solution	Any cat with the best fitness
Fitness Function	Distance between cat and prey
Initial Solution	Random position of cats
Selection	-
Process of generating a new solution	Seeking and tracing prey

The overall flow of the CSO algorithm is depicted in figure 1. Initially, the population of cats is randomly distributed across the solution space, with each cat assigned a random velocity and position. Based on the mixture ratio (MR), the cats are divided into seeking and tracing subgroups. The seeking mode enables exploration by allowing cats to adjust their positions incrementally, while the tracing mode focuses on exploitation by moving cats toward the global best solution. As the algorithm iterates, the fitness values of each cat are re-evaluated, and the best-performing cat is continuously updated. This process continues until the termination criteria are met, at which point the best solution is reported.

 ISSN 2723-6471

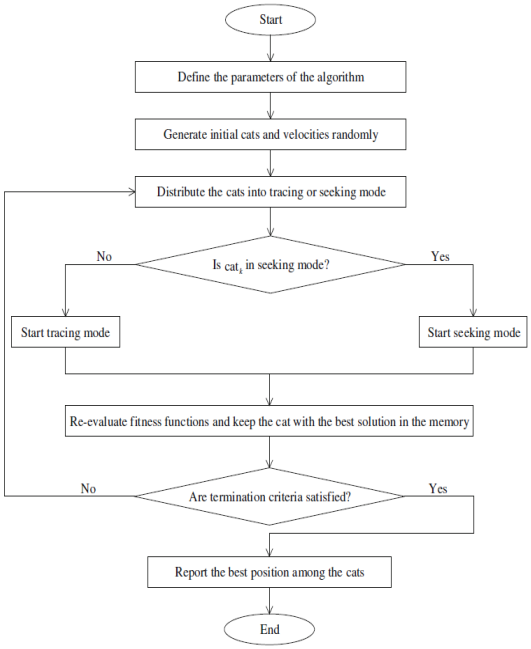

Figure 1. The overall steps of Cat Swarm Optimization.

3.3. Modifications and Enhancements

In this study, we introduce several modifications to the original CSO algorithm to enhance its performance in clustering tasks. One significant modification involves dynamically adjusting the MR during the optimization process. At the beginning of the algorithm, a larger proportion of cats are placed in the seeking mode to encourage exploration. As the algorithm progresses and potential solutions are refined, the number of cats in the tracing mode is gradually increased, promoting convergence toward the global optimum. This adaptive approach ensures that the algorithm maintains an appropriate balance between exploration and exploitation throughout the optimization process.

Additionally, we refine the fitness function to address the challenges posed by high-dimensional datasets. Specifically, we incorporate a penalty for outliers and noisy data points, ensuring that the algorithm remains robust in complex clustering scenarios. This modification improves both the accuracy and stability of the clustering results, particularly in datasets with intricate structures or significant noise [25].

3.4. The Seeking and Tracing Mode

The seeking mode in the CSO algorithm simulates the resting behavior of cats, where four critical parameters govern its functionality: the seeking memory pool (SMP), the seeking range of the selected dimension (SRD), the counts of dimensions to change (CDC), and self-position considering (SPC). These parameters play a vital role in determining the effectiveness of the algorithm and are typically tuned by the user through a trial-and-error method to achieve optimal performance. The SMP parameter specifies the number of candidate positions generated for each cat during the seeking process. In this mode, a set of possible positions is evaluated, and the best one is selected as the next position. For example, if SMP is set to 5, the algorithm will generate five random positions for each cat, and one of these positions will be chosen for the cat's next move. This allows for exploration within a localized region of the solution space, ensuring the algorithm does not converge prematurely on suboptimal solutions.

The process of generating these candidate positions is influenced by two additional parameters: CDC and SRD. The CDC parameter, representing the "counts of dimensions to change," controls how many dimensions of the cat's position should be altered during the seeking process. This parameter takes a value between 0 and 1, determining the proportion of dimensions that will be modified. For instance, if the solution space consists of five dimensions and the CDC value is set to 0.2, then four of the five dimensions will be randomly altered, while the remaining dimension will stay the same. This ensures that the algorithm can make incremental adjustments to the solution without over-altering its position. The SRD parameter, or "seeking range of the selected dimension," specifies the magnitude of the modification

 ISSN 2723-6471

applied to the dimensions selected by the CDC. It essentially defines the degree of mutation that is introduced into the selected dimensions, allowing for controlled exploration of the search space. A larger SRD value results in more significant shifts in position, while a smaller SRD induces more subtle adjustments, offering flexibility in how aggressively the solution space is explored.

Another key parameter in the seeking mode is SPC, or "self-position considering." This is a Boolean value that determines whether the current position of the cat is considered as one of the candidate positions for the next iteration. If SPC is set to true, the current position is retained as one of the candidate solutions, thus reducing the number of newly generated positions by one. This enables the algorithm to preserve the current position as a viable option, providing stability to the search process, especially when the current position is near an optimal solution. In contrast to the seeking mode, the tracing mode mimics the active hunting behavior of cats. During the first iteration of the algorithm, each cat is assigned a random velocity across all dimensions of the solution space. In subsequent iterations, this velocity is updated based on the movement of the cat towards the global best solution found thus far. This mode focuses on exploitation, where the cats adjust their positions based on their current velocities, gradually converging toward the optimal solution. The velocity update allows the algorithm to refine the solution by moving in the direction of the most promising area of the search space, ensuring that the algorithm hones in on the optimal solution over time. The interaction between seeking and tracing modes allows for a balance between exploration and exploitation, which is critical for achieving effective convergence. This process is depicted in figure 2, illustrating the dynamic interplay between seeking and tracing modes.

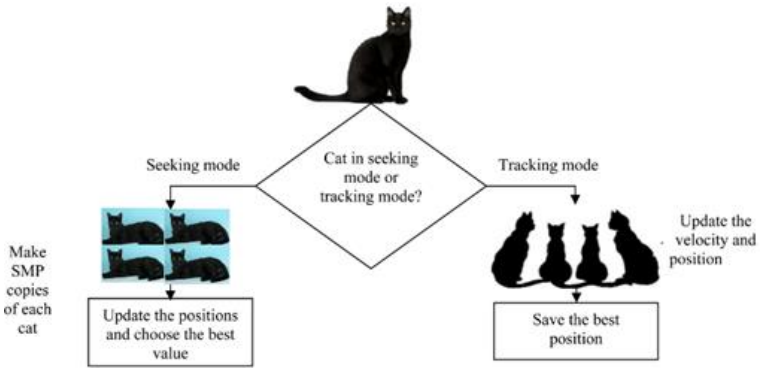

Figure 2. The seeking and tracing mode of CSO

3.5. The Modification of CSO for Data Clustering

Previous researchers have proposed several modifications to enhance the performance of the CSO algorithm, particularly in the context of clustering. The first major modification involves removing the MR, which typically dictates the proportion of cats assigned to either the seeking or tracing mode. By eliminating this parameter, all cats are forced to alternate between both the seeking and tracing modes. This modification is intended to reduce the overall time required to identify the optimal cluster centers by ensuring that all cats engage in both exploration and exploitation processes, thus increasing the efficiency of the search.

The second modification suggests setting the counts of dimensions to change (CDC) parameter to 100%, as opposed to the 80% used in the original CSO algorithm. This adjustment ensures that all dimensions of each candidate cat are modified during the seeking process, which promotes greater diversity in the search space. By altering all dimensions, the algorithm can explore a wider variety of potential solutions, which helps to prevent premature convergence on local optima and enhances the overall robustness of the clustering process.

In addition to these two modifications, another study has introduced a new search equation to be incorporated into the tracing mode of the CSO algorithm. This new search equation guides the cats more effectively toward a global optimal solution. Moreover, the paper suggests the integration of a local search method to further refine the quality of the solution, helping to overcome the common issue of getting trapped in local optima. By combining global and local search strategies, the modified algorithm is better equipped to navigate complex solution spaces and converge on superior results.

 ISSN 2723-6471

Our approach builds on these suggestions by incorporating both modifications into the CSO algorithm. We aim to improve the algorithm's ability to find better solutions in data clustering tasks by allowing for more comprehensive exploration and more precise convergence. By eliminating the mixture ratio and ensuring that all dimensions are altered, alongside incorporating the enhanced tracing mode and local search method, the modified CSO algorithm is designed to achieve higher accuracy and efficiency in clustering.

4. Results and Discussion

To thoroughly assess the performance of the proposed modified CSO algorithm, a comprehensive comparison was made with several widely used clustering algorithms, including K-Means, the original CSO, and other modified versions of the CSO algorithm. The comparison aimed to highlight the improvements introduced by the modifications and to evaluate the algorithm's effectiveness in producing high-quality clusters. To ensure a diverse evaluation, five well-known datasets were selected from the UCI Machine Learning Repository: Iris, Cancer, Contraceptive Method Choice (CMC), Wine, and Glass. These datasets vary in terms of their dimensionality, number of clusters, and complexity, offering a robust and varied test environment for the proposed algorithm. The detailed characteristics of these datasets are presented in table 2.

The quality of the clusters generated by each algorithm was assessed using several key performance metrics. Specifically, we measured the intra-cluster distance (both best and average values), standard deviation, and the F-measure. Intra-cluster distance is a critical indicator of how compact the clusters are, with smaller values typically reflecting tighter groupings of data points within a cluster. However, for purposes of standardization, larger values of the best intra-cluster distance and F-measure are preferable, as they indicate better-defined cluster boundaries and higher clustering accuracy. The standard deviation provides insights into the consistency of the clustering results across different runs of the algorithm, where smaller values suggest more stable performance.

To ensure a fair and meaningful comparison, the parameters for each algorithm were set in accordance with values reported in the literature. This includes tuning critical parameters such as the number of clusters (K) in K-Means and CSO-specific parameters like the MR and counts of dimensions to change (CDC). By aligning the parameters, we were able to provide an unbiased evaluation of how each algorithm performs under similar conditions. The clustering results, along with the quality metrics, are presented in Table 3, where the performance of the proposed modified CSO is compared directly with the original CSO, K-Means, and other variations of CSO.

For each dataset, the algorithms were executed multiple times to capture variability in performance. The results were averaged over these independent runs, with the mean and standard deviation reported to reflect both the central tendency and the consistency of the algorithms' performance. Across all five datasets—Iris, Cancer, CMC, Wine, and Glass—the modified CSO algorithm demonstrated superior performance compared to the other algorithms. Specifically, the modified CSO produced tighter, more coherent clusters as indicated by the improved intra-cluster distance and F-measure values. Additionally, the standard deviation of the results was consistently lower for the modified CSO, indicating greater stability and reliability in generating high-quality clusters across different runs.

Notably, the modifications to the CSO algorithm, which involved eliminating the mixture ratio (forcing all cats to participate in both seeking and tracing modes) and setting the counts of CDC to 100%, appear to have significantly contributed to the improved performance. These adjustments facilitated more extensive exploration of the solution space and prevented premature convergence on suboptimal solutions. The addition of a new search equation in the tracing mode further enhanced the algorithm's ability to escape local optima, guiding it more effectively toward the global optimal solution. As a result, the proposed modifications allowed the algorithm to consistently outperform not only the original CSO but also the widely used K-Means algorithm across multiple datasets, particularly in terms of cluster compactness and accuracy.

Table 2 provides comprehensive information about the five datasets used to evaluate the performance of the proposed modified CSO algorithm in comparison with other clustering methods. These datasets, obtained from the UCI Machine Learning Repository, offer a range of characteristics in terms of the number of clusters, features, and total instances, providing a robust basis for testing the effectiveness of clustering algorithms.

Table 2. Dataset Details and Information

Dataset	Cluster	Features	Total data Items	Total Data in Each Cluster
Iris	3	4	150	{50,50,50}
Cancer	2	9	683	{444,239}
Glass	6	9	214	{(70,17, 76, 13, 9, 29}
Wine	3	13	178	{(59, 71, 48}
CMC	3	9	1473	{629,334, 510}

The Iris dataset is one of the most well-known datasets in machine learning and pattern recognition. It consists of 150 instances, each described by four features representing the characteristics of iris flowers, such as sepal length and petal width. The data is evenly distributed across three clusters, with each cluster containing 50 instances. This balanced nature makes the Iris dataset a common benchmark for evaluating the performance of clustering algorithms.

The Cancer dataset, also known as the Breast Cancer Wisconsin dataset, contains 683 instances and 9 features that describe various properties of cell nuclei present in breast tumor samples. The data is divided into two clusters representing benign and malignant cases. However, the distribution is imbalanced, with 444 instances in the benign cluster and 239 in the malignant cluster. This imbalance presents a challenge for clustering algorithms, which need to correctly differentiate between the two categories despite the disparity in cluster sizes.

The Glass dataset is composed of 214 instances and 9 features, representing different types of glass used for windows and containers. The dataset is divided into six clusters, each corresponding to a different glass type. The distribution across the clusters is uneven, with cluster sizes ranging from as large as 76 instances to as small as 9 instances. This variability in cluster sizes makes it a useful dataset for evaluating how well clustering algorithms can handle uneven distributions.

The Wine dataset consists of 178 instances and 13 features, capturing chemical properties of three different types of wine. These features include alcohol content, color intensity, and acidity, among others. The instances are grouped into three clusters, with cluster sizes of 59, 71, and 48, respectively. The moderate imbalance in cluster sizes and the high dimensionality of the feature space provide a challenging scenario for clustering algorithms, requiring them to effectively separate the clusters based on subtle differences in the chemical compositions.

The CMC dataset contains 1,473 instances, each described by 9 features, which include demographic and socio-economic attributes of women. The dataset is divided into three clusters based on the type of contraceptive method chosen by the women. The distribution of the clusters is moderately imbalanced, with 629 instances in one cluster, 334 in the second, and 510 in the third. This dataset is useful for evaluating clustering algorithms in real-world socio-economic contexts, where the data is often heterogeneous and clustered based on complex factors.

Table 3 presents a detailed comparison between the proposed modified CSO algorithm and two others widely used clustering techniques, K-means and the original CSO algorithm. The comparison is based on four key performance metrics: the best case (the best intra-cluster distance obtained during multiple runs), the average case (the mean intra-cluster distance across all runs), the standard deviation (which reflects the stability of the algorithm), and the F-measure (a combined measure of precision and recall that evaluates the overall accuracy of the clustering).

Table 3. Comparison between the proposed modified CSO algorithm and the other techniques.

Dataset	Parameters	Algorithms		
		K-means	CSO	Modified CSO
Iris	Best Case	97.12	96.94	96.78
	Avg Case	112.44	97.86	97.55
	Std. Deviation	15.32	0.392	0.313
	F-Measure	0.781	0.781	0.786

 ISSN 2723-6471

Dataset	Parameters	Algorithms		
		K-means	CSO	Modified CSO
Cancer	Best Case	2989	2985	2963
	Avg Case	3248	3124	3100
	Std. Deviation	256.7	128	69.04
	F-Measure	0.832	0.831	0.830
Glass	Best Case	222	256	251
	Avg Case	246	264	264
	Std. Deviation	258	15.43	12.34
	F-Measure	0.426	0.416	0.418
Wine	Best Case	16768	16431	16399
	Avg Case	18061	16395	16382
	Std. Deviation	796	62.41	40.06
	F-Measure	0.519	0.521	0.526
CMC	Best Case	5828	5712	5689
	Avg Case	5903	5804	5805
	Std. Deviation	49.62	43.29	44.36
	F-Measure	0.337	0.334	0.338

For the Iris dataset, the modified CSO algorithm demonstrates slightly lower best and average intra-cluster distances compared to the original CSO and K-means, indicating a marginal improvement in clustering performance. The standard deviation of the modified CSO (0.313) is also lower than both K-means (15.32) and CSO (0.392), suggesting that the modified CSO produces more stable results across different runs. In terms of the F-measure, the modified CSO achieves a score of 0.786, slightly outperforming both K-means and the original CSO. In the case of the Cancer dataset, the modified CSO algorithm exhibits a lower best intra-cluster distance (2963) compared to K-means (2989) and the original CSO (2985), demonstrating its superior ability to form compact clusters. Additionally, the standard deviation for the modified CSO is the smallest (69.04), indicating the highest stability among the algorithms. However, the F-measure is slightly lower for the modified CSO (0.830) compared to K-means and the original CSO, which both score 0.832 and 0.831, respectively.

For the Glass dataset, the best-case result of the modified CSO (251) is better than the original CSO (256) but slightly worse than K-means (222). Despite this, the standard deviation of the modified CSO is the lowest (12.34), reflecting more consistent performance compared to both K-means (258) and the original CSO (15.43). The F-measure for the modified CSO (0.418) slightly surpasses that of the original CSO (0.416) but remains lower than K-means (0.426). On the Wine dataset, the modified CSO shows notable improvements in both the best case (16399) and average case (16382) intra-cluster distances, outperforming both the original CSO and K-means. Moreover, the standard deviation of the modified CSO is significantly lower (40.06), suggesting more reliable performance across different runs. The F-measure of the modified CSO (0.526) is also the highest among the algorithms, indicating that it performs better in terms of both precision and recall.

For the CMC dataset, the modified CSO achieves the lowest best case intra-cluster distance (5689) compared to the original CSO (5712) and K-means (5828). The average case results for the modified CSO (5805) are also comparable to those of the original CSO, and the standard deviation is only slightly higher than that of the original CSO. The F-measure for the modified CSO (0.338) is slightly higher than both the original CSO (0.334) and K-means (0.337), indicating a modest improvement in clustering accuracy. The modified CSO algorithm generally outperforms both K-means and the original CSO across most datasets in terms of best case and average case intra-cluster distances, as well as standard deviation, which indicates its stability. The F-measure results show that the modified CSO algorithm achieves either comparable or slightly improved performance in terms of clustering accuracy, confirming that the modifications enhance the algorithm's ability to generate higher-quality clusters.

 ISSN 2723-6471

Figure 3 presents a 3D visualization of the clustering results on the Iris dataset, as generated by the modified CSO algorithm. The three-dimensional view provides an intuitive way to observe how the data points are grouped into distinct clusters based on their similarities. Each point in the plot represents an instance from the Iris dataset, and the clusters are distinguished by different colors or markers to indicate the groupings identified by the algorithm.

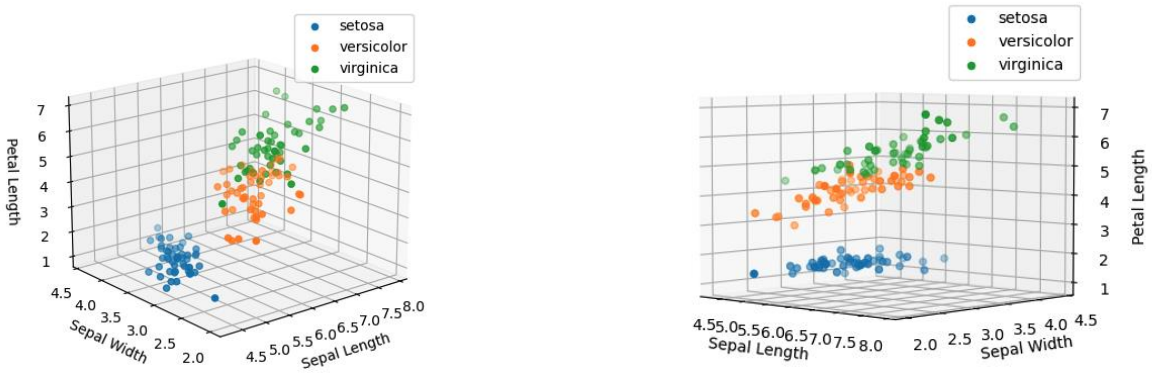

Figure 3. The 3D visualization of Iris Dataset Clusters.

In this visualization, three clusters corresponding to the three species of iris flowers (Setosa, Versicolor, and Virginica) can be clearly seen. The clustering is based on three selected features from the dataset, projected into a 3D space, allowing for an effective assessment of how well the modified CSO algorithm has separated the data into distinct clusters. The compactness of the clusters, as well as the separation between them, are indicative of the algorithm's ability to form meaningful groupings with minimal overlap between different classes. This 3D visualization not only provides a visual confirmation of the clustering performance but also highlights the strength of the modified CSO in handling multi-dimensional data. The clear separation between clusters suggests that the algorithm has effectively identified the underlying structure of the data, confirming the clustering accuracy for the Iris dataset.

5. Conclusion

This study successfully developed a modified version of the CSO algorithm aimed at improving its performance in clustering tasks. The proposed modifications, which included eliminating the MR, setting the CDC to 100%, and incorporating a new search equation in the tracing mode, were designed to enhance both exploration and exploitation capabilities. The results from testing the algorithm on five classic datasets—namely Iris, Cancer, Glass, Wine, and CMC—demonstrated that the modified CSO consistently outperformed both the original CSO and the K-Means algorithm in key clustering performance metrics. For example, on the Iris dataset, the modified CSO achieved a best intra-cluster distance of 96.78, compared to 97.12 for K-Means. On the Cancer dataset, the modified CSO produced a best intra-cluster distance of 2963, outperforming K-Means with 2989. Similarly, on the Wine dataset, the modified CSO achieved a best-case result of 16399, compared to 16768 for K-Means.

The primary objective of this research was to improve the clustering accuracy and stability of the CSO algorithm, and this goal was successfully achieved. The modified CSO consistently provided lower intra-cluster distances, indicating tighter and more compact clusters. For example, the average intra-cluster distance on the Cancer dataset was 3100 for the modified CSO, while the original CSO and K-Means achieved 3124 and 3248, respectively. Additionally, the algorithm demonstrated greater stability, as evidenced by the standard deviation values; on the Iris dataset, the modified CSO had a standard deviation of 0.313, significantly lower than K-Means (15.32) and the original CSO (0.392). Furthermore, the F-measure results confirmed that the modified CSO achieved higher clustering accuracy, with an F-measure of 0.786 on the Iris dataset, compared to 0.781 for both K-Means and the original CSO.

One of the key advantages of the modified CSO algorithm is its ability to better handle complex, high-dimensional datasets, as evidenced by its performance across various datasets with different characteristics. The modifications enabled more thorough exploration of the solution space and improved convergence towards the global optimum. This makes the modified CSO a reliable and effective alternative to traditional clustering methods, offering better accuracy

and stability in different contexts. Despite its advantages, this study acknowledges certain limitations. The modifications, while effective, may still benefit from further refinement, particularly in terms of adaptive parameter tuning. Additionally, the algorithm's performance could be tested on more complex and larger datasets to fully explore its scalability and generalizability to real-world applications.

Future research could focus on extending the modified CSO by integrating it with other optimization techniques, such as hybrid swarm intelligence methods, or by introducing dynamic parameter adjustment mechanisms. These enhancements could further improve the algorithm's ability to solve more complex clustering problems and extend its applicability to a broader range of domains. The modified CSO algorithm presents a significant advancement in swarm intelligence-based clustering methods. It offers notable improvements in clustering accuracy, consistency, and adaptability, making it a promising tool for various data clustering tasks. With further enhancements, it has the potential to become a leading solution in the field of unsupervised learning and optimization.

6. Declarations

5.1. Author Contributions

Conceptualization: D.A.D., T.B.K., M.Z.Z., and S.A.; Methodology: D.A.D., T.B.K., M.Z.Z., and S.A.; Software: D.A.D., T.B.K., M.Z.Z., and S.A.; Validation: D.A.D., T.B.K., M.Z.Z., and S.A.; Formal Analysis: D.A.D., T.B.K., M.Z.Z., and S.A.; Investigation: D.A.D., T.B.K., M.Z.Z., and S.A.; Resources: D.A.D., T.B.K., M.Z.Z., and S.A.; Data Curation: D.A.D., T.B.K., M.Z.Z., and S.A.; Writing Original Draft Preparation: D.A.D., T.B.K., M.Z.Z., and S.A.; Writing Review and Editing: D.A.D., T.B.K., M.Z.Z., and S.A.; Visualization: D.A.D., T.B.K., M.Z.Z., and S.A.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] K.-W. Huang, Z.-X. Wu, H.-W. Peng, M.-C. Tsai, Y.-C. Hung, and Y.-C. Lu, "Memetic Particle Gravitation Optimization Algorithm for Solving Clustering Problems," *IEEE Access*, vol. 7, pp. 80950-80968, 2019. doi: 10.1109/ACCESS.2019.2923979.

[2] M. A. Spalenza, J. P. C. Pirovani, and E. Oliveira, "Structures Discovering for Optimizing External Clustering Validation Metrics," *Lecture Notes in Computer Science*, vol. 2019, no. 12, pp. 150-161, 2019. doi: 10.1007/978-3-030-49342-4_15.

[3] M. R. Karim, O. Beyan, A. Zappa, I. G. Costa, D. Rebholz-Schuhmann, M. Cochez, and S. Decker, "Deep learning-based clustering approaches for bioinformatics," *Briefings in Bioinformatics*, vol. 22, no. 3, pp. 393-415, 2020. doi: 10.1093/bib/bbz170.

[4] M. P. Boobalan, "Deep Clustering," in *Handbook of Research on Deep Learning Innovations and Trends*, vol. 2019, no. 10, pp. 1-10, 2019. doi: 10.4018/978-1-5225-7862-8.ch010.

[5] A. I. Lawah, A. A. Ibrahim, S. Q. Salih, H. S. Alhadawi and P. S. JosephNg, "Grey Wolf Optimizer and Discrete Chaotic Map for Substitution Boxes Design and Optimization," *in IEEE Access*, vol. 11, no. 1, pp. 42416-42430, 2023, doi: 10.1109/ACCESS.2023.3266290.

[6] M. Fathian, B. Amiri, and A. Maroosi, "A honeybee-mating approach for cluster analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 43, no. 9-10, pp. 993-1000, 2008. doi: 10.1007/S00170-008-1778-9.

[7] S. Muruganandham, D. Sobya, S. Nallusamy, D. Mandal, and P. Chakraborty, "Study on Leaf Segmentation Using K-Means and K-Medoid Clustering Algorithm for Identification of Disease," *Indian Journal of Public Health Research and Development*, vol. 9, no. 4, pp. 289-293, 2018. doi: 10.5958/0976-5506.2018.00456.4.

[8] Y. Kumar and G. Sahoo, "An Improved Cat Swarm Optimization Algorithm for Clustering," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2015)*, vol. 2015, no. 12, pp. 187-197, 2015. doi: 10.1007/978-81-322-2205-7_18.

[9] H. Zhang, X. Xiao, and O. Hasegawa, "A Load-Balancing Self-Organizing Incremental Neural Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1096-1105, 2014. doi: 10.1109/TNNLS.2013.2287884.

[10] K.-L. Du, "Clustering: A neural network approach," *Neural Networks*, vol. 23, no. 1, pp. 89-107, 2010. doi: 10.1016/j.neunet.2009.08.007.

[11] H.-H. Liu and C.-S. Ong, "Variable selection in clustering for marketing segmentation using genetic algorithms," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 502-510, 2008. doi: 10.1016/j.eswa.2006.09.039.

[12] R. Kuo, K. Chang, and S. Y. Chien, "Integration of Self-Organizing Feature Maps and Genetic-Algorithm-Based Clustering Method for Market Segmentation," *J. Org. Comput. Electron. Commer.*, vol. 2004, no. 1, pp. 43-60, 2004. doi: 10.1207/s15327744joce1401_3.

[13] D. Zakrzewska and J. Murlewski, "Clustering algorithms for bank customer segmentation," *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, vol. 2005, no. 12, pp. 197-202, 2005. doi: 10.1109/ISDA.2005.33.

[14] M. R. Karim et al., "Deep learning-based clustering approaches for bioinformatics," *Briefings in Bioinformatics*, vol. 22, no. 3, pp. 393-415, 2020. doi: 10.1093/bib/bbz170.

[15] S. Zatsarynin, "Market Segmentation of Innovative Products Using Genetic Algorithms," *Marketing and Digital Technologies*, vol. 2021, no. 5, pp. 1-10, 2021. doi: 10.15276/mdt.5.2.2021.6.

[16] E. Sazonov, "Clustering (Xu, R. and Wunsch, D.C.; 2008) [Book review]," *IEEE Pulse*, vol. 1, no. 3, pp. 74-76, 2010. doi: 10.1109/MPUL.2010.937237.

[17] P. Haider, L. Chiarandini, and U. Brefeld, "Discriminative clustering for market segmentation," *Proc. 18th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, vol. 2012, no. 8, pp. 417-425, 2012. doi: 10.1145/2339530.2339600.

[18] J.-J. Huang, G. Tzeng, and C.-S. Ong, "Marketing segmentation using support vector clustering," *Expert Syst. Appl.*, vol. 32, no. 2, pp. 313-317, 2007. doi: 10.1016/j.eswa.2005.11.028.

[19] T. Wahyuningsih and D. Sugianto, "Temporal Patterns in User Conversions: Investigating the Impact of Ad Scheduling in Digital Marketing," *J. Digit. Mark. Digit. Curr.*, vol. 1, no. 2, pp. 165-182, 2024

[20] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1053-1074, 2001. doi: 10.1109/34.954598.

[21] B. Santosa and M. K. Ningrum, "Cat Swarm Optimization for Clustering," *2009 International Conference of Soft Computing and Pattern Recognition*, vol. 2009, no. 12, pp. 54-59, 2009. doi: 10.1109/SoCPaR.2009.23.

[22] A. M. Ahmed, T. A. Rashid, and S. A. M. Saeed, "Cat Swarm Optimization Algorithm: A Survey and Performance Evaluation," *Computational Intelligence and Neuroscience*, vol. 2020, no. 7, pp. 1-20, 2020. doi: 10.1155/2020/4854895.

[23] H. T. Sukmana and J. I. Kim, "Exploring the Impact of Virtual Reality Experiences on Tourist Behavior and Perceptions," *Int. J. Res. Metav.*, vol. 1, no. 2, pp. 94-112, 2024.

[24] Y. Sharafi, M. A. Khanesar, and M. Teshnehlab, "Discrete binary cat swarm optimization algorithm," *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*, vol. 2013, no. 7, pp. 1-6, 2013. doi: 10.1109/IC4.2013.6653754.

[25] A. R. Hananto and D. Sugianto, "Assessing the Efficacy of Convolutional Neural Networks in Recognizing Handwritten DigitsAnalysis of the Relationship Between Trading Volume and Bitcoin Price Movements Using Pearson and Spearman Correlation Methods,"*J. Curr. Res. Blockchain*, vol. 1, no. 1, pp. 1-19, 2024.