



OPEN

Augmented weighted K-means grey wolf optimizer: An enhanced metaheuristic algorithm for data clustering problems

Manoharan Premkumar^{1✉}, Garima Sinha², Manjula Devi Ramasamy³, Santhoshini Sahu⁴, Chithirala Bala Subramanyam⁵, Ravichandran Sowmya⁶, Laith Abualigah^{7,8,9,10,11,12,13,14} & Bizuwork Derebew^{15✉}

This study presents the K-means clustering-based grey wolf optimizer, a new algorithm intended to improve the optimization capabilities of the conventional grey wolf optimizer in order to address the problem of data clustering. The process that groups similar items within a dataset into non-overlapping groups. Grey wolf hunting behaviour served as the model for grey wolf optimizer, however, it frequently lacks the exploration and exploitation capabilities that are essential for efficient data clustering. This work mainly focuses on enhancing the grey wolf optimizer using a new weight factor and the K-means algorithm concepts in order to increase variety and avoid premature convergence. Using a partitional clustering-inspired fitness function, the K-means clustering-based grey wolf optimizer was extensively evaluated on ten numerical functions and multiple real-world datasets with varying levels of complexity and dimensionality. The methodology is based on incorporating the K-means algorithm concept for the purpose of refining initial solutions and adding a weight factor to increase the diversity of solutions during the optimization phase. The results show that the K-means clustering-based grey wolf optimizer performs much better than the standard grey wolf optimizer in discovering optimal clustering solutions, indicating a higher capacity for effective exploration and exploitation of the solution space. The study found that the K-means clustering-based grey wolf optimizer was able to produce high-quality cluster centres in fewer iterations, demonstrating its efficacy and efficiency on various datasets. Finally, the study demonstrates the robustness and dependability of the K-means clustering-based grey wolf optimizer in resolving data clustering issues, which represents a significant advancement over conventional techniques. In addition to addressing the shortcomings of the initial algorithm, the incorporation of K-means and the innovative weight factor into the grey wolf optimizer establishes a new standard for further study in metaheuristic clustering algorithms. The performance of the K-means clustering-based grey wolf

¹Department of Electrical & Electronics Engineering, Dayananda Sagar College of Engineering, Kumaraswamy Layout, Bengaluru, Karnataka 560078, India. ²Department of Computer Science and Engineering, Jain University, Ramanagaram, Bengaluru, Karnataka, India. ³Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India. ⁴Department of Computer Science & Engineering, GMR Institute of Technology, Rajam, Srikakulam, Andhra Pradesh, India. ⁵Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, India. ⁶Department of Electrical and Electronics Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India. ⁷Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan. ⁸Artificial Intelligence and Sensing Technologies (AIST) Research Center, University of Tabuk, 71491 Tabuk, Saudi Arabia. ⁹Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan. ¹⁰MEU Research Unit, Middle East University, Amman 11831, Jordan. ¹¹Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon. ¹²School of Engineering and Technology, Sunway University Malaysia, 27500 Petaling Jaya, Malaysia. ¹³College of Engineering, Yuan Ze University, Taoyuan, Taiwan. ¹⁴Applied science research center, Applied science private university, Amman, 11931, Jordan. ¹⁵Department of Statistics, College of Natural and Computational Science, Mizan-Tepi University, Tepi Bushira, Ethiopia. ✉email: mprem.me@gmail.com; bizuworkd@mtu.edu.et

optimizer is around 34% better than the original grey wolf optimizer algorithm for both numerical test problems and data clustering problems.

Keywords Computational intelligence, Data mining, Grey wolf optimizer, K-means clustering, Optimization algorithm

Basic concepts of data clustering

An unsupervised learning technique called clustering separates a database into clusters of identical matters by reducing the resemblance among objects in diverse groups and exploiting the similarities between entities in the identical cluster^{1–3}. Clustering has been a crucial tool for data analysis in many disciplines, including intrusion detection, data mining, bioinformatics, and machine learning systems. Clustering is also used in various fields, including social network analysis, robotics, and networks. Hierarchical clustering, mixed clustering, learning network clustering, and partition clustering can all be used to group data into clusters^{4–6}. The main objective of clustering techniques is to make clusters more homogeneous and heterogeneous. Partitional clustering techniques previously ran into issues such as responsiveness to the starting center points, local optima trap, and lengthy run times. Clustering separates a set of datasets with d -dimensions into k different groups. Every division is known as a cluster C_i . Each cluster's members share several traits in common, although there is little overlap between them^{7–9}. In this situation, clustering's objective is to identify the separate groups and allocate objects depending on how closely they resemble the appropriate groups. The absence of initial tags for observations is the primary distinction between the clustering and classification methods. However, classification techniques use predetermined classifications to which objects are given, clustering groups of objects without previous knowledge^{10–13}.

Numerous research efforts on data clustering have been offered throughout the past decades. To cluster a dataset, there are various solutions to the clustering problem. These methods primarily use complicated network approaches, K -means, and its improved variants, metaheuristic algorithms, and other methods^{14–18}. One of the most well-known of these methods is the K -means algorithm, which attempts to partition a complete dataset into k clusters by randomly selecting k data points as starting cluster centers. The K -means method, however, is delicate to the choice of beginning points and might be unable to group huge databases. Many experts have concentrated on swarm intelligence algorithms to address the drawbacks of the K -means technique, which can do a simultaneous search in a complicated search space to avoid a premature convergence trap^{3,15,19–24}. Researchers also focused on merging metaheuristic techniques with conventional clustering techniques to lessen such limitations.

Literature review

Metaheuristics algorithms are population-based and imitate the shrewd behaviour of socially organized creatures. Glowworm and crow search-based clustering have been proposed for data clustering. These techniques depicted the clustering solutions as swarms of creatures. Then, to quickly cover the search space, these strategies use clever intensification and diversification techniques. Although metaheuristic algorithms reduced classic clustering techniques' execution times, they still had drawbacks²⁵. The shortcomings of Particle Swarm Optimization (PSO) and its competitors can be summarized as follows: lack of developed memory elements and the diversity of populations²⁶. The PSO and its variants use a single optimal solution stored in the solution space to reposition the members of the swarm, which can cause them to become stuck in local minima. These shortcomings caused PSO and its variants to obtain solutions with low quality and convergence speed, which accounts for the birth of numerous other algorithms reported in the literature^{13,27,28}.

The authors of Ref.²⁹ suggested a genetic algorithm (GA) for clustering that exploits a region-based crossover mechanism. It finds the best preliminary center for the k -means algorithm. The chromosomes translate the clusters of centroids, and during the crossover, chromosome pairs exchange several centroids located in the same area of space. According to experimental research, the region-based crossover outperforms a random exchange of centroids. The authors of Ref.² suggested a differential evolution (DE) algorithm integrating the k -means technique. The local search and initial population are conducted using the k -means algorithm. The population vectors set the cluster centroids. In an additional effort to eliminate the redundant nature of the centroid encoding, The authors of Ref.³⁰ reported a technique that hybridized the k -means algorithm and gravitational search algorithm. The k -means was used to improve the generation of the initial population; one individual was generated using k -means, and the remaining individuals were generated randomly. A data clustering approach based on the Gauss chaotic map-using PSO was presented in Ref.³¹. Sequences produced by the Gauss chaotic drift were used to replace the random elements affecting the velocity update's cognitive and social components. A cooperative artificial bee colony (CABC) approach for data clustering was proposed in Ref.³², in which every bee contributes to creating the optimal solution. The appropriate solution for each bee is thought to replace every solution of the optimal solution. The authors of Ref.^{33,34} used representative points, typically not centroids, to indicate potential solutions, and, as with centroids, a dataset partition was created by allocating data to the cluster closest to the representative point.

Many metaheuristic algorithms have recently been reported in addition to the above-discussed algorithms for numerical and real-world engineering design optimization problems, including data clustering. For instance, ant colony optimization³⁵, firefly algorithm^{36,37}, flower pollination algorithm³⁸, grey wolf optimizer (GWO)^{39–42}, Jaya algorithm⁴³, Teaching–learning based optimization (TLBO) algorithm⁴⁴, Rao algorithm⁴⁵, political optimizer⁴⁶, whale optimization algorithm (WOA)⁴⁷, Moth flame algorithm (MFO)⁴⁸, multi-verse optimizer (MVO)⁴⁹, Salp swarm algorithm (SSA)^{50,51}, spotted hyena optimizer⁵², butterfly optimization⁵³, lion optimization⁵⁴, fire-works algorithm⁵⁵, Cuckoo search algorithm⁵⁶, bat algorithm⁵⁷, Tabu search⁵⁸, harmony search algorithm⁵⁹,

Newton–Raphson optimizer⁶⁰, reptile search algorithm⁶¹, slime mould algorithm^{62,63}, harris hawk optimizer⁶⁴, Chimp optimizer⁶⁵, artificial gorilla troop optimizer⁶⁶, atom search algorithm⁶⁷, marine predator algorithm^{68,69}, sand cat swarm algorithm⁷⁰, equilibrium optimizer^{71,72}, Henry gas solubility algorithm (HGSA)⁷³, resistance–capacitance algorithm⁷⁴, arithmetic optimization algorithm⁷⁵, quantum-based avian navigation optimizer⁷⁶, multi trail vector DE algorithm^{10,77}, arithmetic optimization algorithm⁷⁸, starling murmuration optimizer⁷⁹, atomic orbit search (AOS)⁸⁰, subtraction-average-based optimizer⁸¹, etc. are reported for solving optimization problems. In conclusion, these new algorithms and their improved variations based on different metaheuristic computing algorithms yield greater results than before^{82–84}. A comparative study to show the recent efforts in using metaheuristic algorithms for data clustering is listed in Table 1.

The authors of Ref.⁸⁸ proposed an improved version of the firefly algorithm by hybridizing the exploration method and a chaotic local search approach. The improved firefly algorithm was practically validated for routinely choosing the optimal dropout rate for the regularization of the neural networks. In order to maximize the local and global characteristics collected from each of the handwritten phrase representations under consideration, a hierarchical feature selection framework based on a genetic algorithm has been developed in Ref.⁸⁹. The authors of Ref.⁹⁰ have reviewed the PSO algorithm and its variants for medical disease detection. The overfitting problem was addressed in Ref.⁹¹ by using the sine-cosine algorithm to determine an appropriate value for the regularisation parameter dropout. According to the literature review, swarm strategies are currently being utilized effectively in this domain, although their future application has not yet been fully explored. The effective extreme gradient boosting classification algorithm, which is used to classify the characteristics obtained by the convolutional layers, was used to substitute the fully connected layers of a standard convolution neural network in order to increase classification accuracy⁹². Furthermore, to support the suggested research, a hybrid version of the arithmetic optimization method is constructed and used to optimize the extreme gradient boosting hyperparameters for COVID-19 chest X-ray pictures. To solve the problem of early convergence, this study introduces a novel variant known as the adaptive seagull optimization algorithm. The performance of the suggested algorithm is improved by increasing the seagulls' inclination towards exploratory behaviour⁹³. Quasi-random sequences are employed for the population initialization in place of a random distribution in order to increase the variety and convergence factors. The authors of Ref.⁹⁴ proposed an enhanced PSO algorithm that uses pseudo-random sequences and opposing rank inertia weights instead of random distributions for initialization to improve convergence speed and population diversity. The authors also introduced a new initialization population approach that uses a quasi-random sequence to initialize the swarm and generates an opposing swarm using the opposition-based method. For fifteen UCI data sets, the suggested technique optimized feed-forward neural network weight.

Research gaps

The GWO is one of the well-known metaheuristic algorithms⁹⁵. This algorithm draws inspiration from the hunting behaviour of grey wolves and the hierarchical leadership model. The GWO has been implemented, and the results have been encouraging enough to warrant additional research. Investigators can improve the issues of low precision and slow convergence speed⁹⁶. As a result, studies utilized various techniques to increase optimizers' efficiency and address optimization issues. For instance, an improved GWO was suggested to adjust the recurrent

Algorithm	Year	Inspiration	Remarks
GA with gene rearrangement ²⁹	2009	GA with gene rearrangement is reported, i.e. a new crossover operator is introduced to improve the exploitation	This algorithm has been tested only for image clustering
CABC ³²	2010	Inspired by the foraging behaviour of honey bees	Six real-time datasets are used to test the algorithm. The comparison was made between GA and PSO
Gravitational search algorithm ³⁰	2011	Inspired by the gravity law and mass interactions	The comparison was made between PSO and GA only. Detailed analysis is not available
DE ²	2013	Inspired by Darwin's theory of evolution	The comparison is carried out between different variants of DE
PSO hybridized with magnetic charge system search ²⁷	2015	Hybrid PSO with magnetic charge system searches and is inspired by electromagnetic theory	Validation is carried out for very few benchmark datasets
Glowworm optimization algorithm ²⁵	2017	The swarm's movement of glowworms is determined by their distance from one another and by a luminous quantity	Detailed analysis is not carried out for the clustering problems
Symbiotic organism search algorithm ⁸⁵	2019	Inspired by the symbiotic interaction implemented to survive and propagate	Ten datasets are used to validate the algorithm and compared with PSO and GA
GWO ³⁹	2020	Inspired by the social hierarchy and hunting behaviour of the grey wolves	Limited datasets are used for validation
MFO ⁸⁶	2021	Inspired by the moth's intelligence, i.e., transverse orientation to navigate in nature	Original MFO is applied, and it gets trapped by local optima
Aquila optimizer ⁸⁷	2022	Hybridized with the arithmetic optimization algorithm. The Aquila optimizer has inspired the behaviours during the finding of the prey	It has been applied for text and data clustering problems. But computation complexity is high
Chaos-based PSO ³¹	2022	Inspired by PSO and Gaussian Chao map	It has been applied for auto labelling in recognition of human activity

Table 1. Summary of a few metaheuristic algorithms applied to data clustering.

neural network's parameters. To allow faster GWO convergence, chaotic GWO was introduced. Researchers have also employed numerous techniques to enhance GWO^{97–99}. The literature review has been extensively augmented to underscore the existing research gaps within the context of optimization algorithms applied to data clustering, with a specific focus on the limitations of current metaheuristic algorithms, including the traditional GWO. Despite GWO's proven effectiveness in various optimization tasks, its application to data clustering reveals critical shortcomings, primarily its struggle with premature convergence and its inability to maintain a balance between exploration and exploitation phases. These limitations significantly affect the quality of clustering outcomes, especially in complex datasets with high dimensionality or noise. Moreover, while existing studies have explored numerous enhancements to GWO and other metaheuristic algorithms, there remains a distinct gap in the literature regarding the integration of these algorithms with classical clustering techniques, such as K-means, to address these specific challenges. This gap highlights the need for innovative approaches that can leverage the strengths of both metaheuristic optimization and traditional clustering methods to achieve superior clustering performance. The proposed *K*-means Clustering-based Grey Wolf Optimizer (KCGWO) aims to fill this gap by introducing a hybrid algorithm that combines the adaptive capabilities of GWO with the efficiency of *K*-means clustering. This combination is designed to enhance diversity, prevent premature convergence, and ensure a more effective balance between the exploration of new solutions and the exploitation of known good solutions. However, the literature review reveals that while there are various attempts to improve the clustering process through algorithmic enhancements, the specific approach of blending GWO with *K*-means, complemented by a dynamic weight factor to adjust exploration and exploitation dynamically, is notably absent. This research gap signifies an opportunity for the KCGWO to contribute significantly to the field, offering a novel solution that addresses both the limitations of traditional GWO in clustering tasks and the need for more effective hybrid algorithms. By clarifying these gaps and positioning the KCGWO within this context, the revised related works section establishes a strong foundation for the significance and novelty of the proposed research.

Need for the proposed algorithm

In addition to the metaheuristic algorithm, the primary goal of the data mining process is to gather data from a big data set. The data can then be translated into a clear format for further usage. Clustering is a popular experimental data analysis tool. Objects are arranged using clustering so that each cluster contains more comparable objects. As discussed earlier, various cluster methods have been created to group the data. *K*-means is an example of a partitioning clustering algorithm because it operates based on the cluster centroid^{15,19}. Numerous uses of the *K*-means cluster have been documented. In addition to enhancing the reliability of wireless sensor networks, *K*-means clustering was also used for image segmentation. Additionally, as an unsupervised learning technique, the *K*-means cluster has been frequently utilized to categorize data with no labels. The primary objective is to propose another variant of GWO called KCGWO to solve complex optimization problems, including data clustering problems. In this study, the KCGWO is proposed as an advanced solution to the inherent limitations of the GWO in addressing data clustering challenges. The GWO, while innovative in mimicking the social hierarchy and hunting tactics of grey wolves, exhibits deficiencies in exploration and exploitation—key factors for effective clustering. The proposed KCGWO method enhances GWO by incorporating the *K*-means algorithm and introducing a dynamic weight factor, aiming to improve the algorithm's performance significantly. The methodology of KCGWO unfolds in two pivotal enhancements over the traditional GWO. First, the integration of the *K*-means algorithm serves as an initial refinement step. Before the optimization process, *K*-means is applied to the dataset to establish a preliminary grouping of data points. This step ensures that the starting positions of the grey wolves (solutions) are closer to potential optimal solutions, thereby enhancing the exploration phase of GWO. The initial clustering helps in guiding the wolves towards promising areas of the search space from the onset. Second, a dynamic weight factor is introduced to adjust the influence of exploration and exploitation dynamically throughout the optimization process. This weight factor varies the wolves' movements, allowing for a more flexible search strategy that can adapt based on the current state of the search. It enables the algorithm to maintain a balance between exploring new areas and exploiting known promising regions, thus preventing premature convergence to suboptimal solutions. The performance of KCGWO was evaluated through extensive testing on numerical benchmarks and real-world datasets, demonstrating its superior capability to efficiently navigate the solution space and locate optimal cluster centers swiftly. This effectiveness is attributed to the synergistic combination of *K*-means for initial solution enhancement and the dynamic weight factor for maintaining an optimal balance between exploration and exploitation. Overall, KCGWO represents a significant advancement in solving data clustering problems, offering a robust and reliable method that overcomes the limitations of GWO. Its innovative approach to integrating *K*-means with a dynamic adjustment mechanism ensures high-quality solutions, making it a valuable tool for data analytics and clustering applications.

The primary contributions of this study are discussed as follows:

- A new variant of GWO called KCGWO based on *K*-means clustering algorithm and weight factors in the position update is proposed.
- Formulation fitness function for data clustering problem of a machine learning systems.
- The performance of the KCGWO is validated using 10 numerical test functions and data clustering problems using eight real-world data sets with different dimensions.
- The performance comparison is made with other well-known algorithms based on the statistical data analysis and statistical Friedman's ranking test (FRT).

The paper has been structured as follows. Section "Data clustering and problem statement" discusses the data clustering concepts and the formulation of the fitness function for the same problem. Section "Proposed K-means clustering grey wolf optimizer" comprehensively presents the formulation of the proposed KCGWO based on the K-means clustering algorithm; in addition, the basic concepts of GWO are also discussed. The results are comprehensively discussed in Section "Results and discussions", and Sect. "Conclusions" concludes the paper with a future study.

Data clustering and problem statement

The basic objective of data mining techniques is to obtain features from huge volumes of data. Such techniques use data processing techniques to find interesting patterns in huge amounts of data. Clustering, classifications, detecting anomalies, detecting deviations, synthesizing, and regression are a few examples of data analysis techniques. Data clustering is dividing a set of information into smaller groups where the similarities between the individuals in each group are high while those between the data in other groups are low. Distance metrics like Euclidean distance, Chord distance, and Jaccard index are used to assess how similar individuals of a subset are to one another. In principle, clustering algorithms can be divided into two groups: partitional and hierarchical, depending on how clusters are created and maintained⁸⁶. A tree that depicts a sequence of clusters is produced in hierarchical clustering with no background knowledge of the number of groups or dependence on the initial state. Nevertheless, because they are static, an entity allocated to one cluster cannot be moved to another. Hierarchical algorithms' main drawback is this. The incompetent clustering of overlapping clusters could also be due to a lack of planning regarding the number of clusters. On the other hand, partitional clustering divides items into clusters of a predetermined size. Various techniques for partitional clustering aim to increase the dissimilarity of members belonging to distinct clusters while attempting to reduce the difference between objects in each cluster^{27,100-102}.

Typically, the Euclidean distance is used to measure similarity. In this work, the distance between any two objects (o_i and o_j) inside the cluster is also determined using the Euclidean distance measure. Typically, it could be expressed as follows¹⁰³:

$$D(o_i, o_j) = \|o_i - o_j\| = \sqrt{\sum_{m=1}^d (o_{im} - o_{jm})^2}, \quad (1)$$

where o_i and o_j denote two distinct objects inside the cluster, d denotes the number of features for the entity, and partition clustering can be transformed into an optimization model based on the similarity metric, and this model can be explained as follows:

$$\text{Minimize}_{Z,W} : f(Z, W) = \sum_{k=1}^K \sum_{i=1}^n w_{ik} D(x_i, z_k), \quad (2)$$

Subjected to:

$$\begin{cases} \sum_{k=1}^K w_{ik} = 1, & i = 1, 2, \dots, n; \\ w_{ik} \in \{0, 1\}, & \forall i \in \{1, 2, \dots, n\}, k \in \{1, 2, \dots, K\} \end{cases} \quad (3)$$

where n denotes the sample size, K denotes the cluster size, and x_i signifies the coordinates of the i th object in the current datasets. The term w_{ik} indicates whether the i th object is clustered into the k th cluster or not, and $D(x_i, z_k)$ indicates the length between the i th object and the center of the k th cluster. Noteworthy is the fact that the following is used to observe the same.

$$W = \{w_{ik} | i = 1, 2, \dots, n, k = 1, 2, \dots, K\} \quad (4)$$

A sample's partition criteria determine the amount of w_{ik} in Eq. (3). Obtain an object partition that meets Eq. (3) for given a sample set $X = \{x_1, x_2, \dots, x_n\}$.

$$\begin{cases} \sum_{i=1}^K C_i = X; \\ C_i \cap C_j = \phi, & \forall i, j \in \{1, 2, \dots, K\} \wedge i \neq j; \\ C_i \neq \phi, & \forall i \in \{1, 2, \dots, K\} \end{cases} \quad (5)$$

where C_i ($i = 1, 2, \dots, K$) is the i th cluster's object set, and the following equation can be used to identify its members:

$$\begin{cases} C_i = \{x_k | \|x_k - z_i\| \leq \|x_k - z_p\|, x_k \in X\}, & p \neq i, p = 1, 2, \dots, K; \\ z_i = \frac{1}{|C_i|} \sum_{x_k \in C_i} x_k, & i = 1, 2, \dots, K \end{cases} \quad (6)$$

where z_i is frequently employed in the k -means clustering method, symbolizes a new center of cluster i , and $\| \cdot \|$ indicates the Euclidean distance between any two items in the subset.

Proposed K-means clustering-based grey wolf optimizer

This section briefs the original concepts of the basic Grey Wolf Optimizer (GWO) with its mathematical modeling. The proposed K-means Clustering-based Grey Wolf Optimizer (KCGWO) is discussed comprehensively.

Grey wolf optimizer

The grey wolf optimization algorithm is the most contemporary breakthrough in the field of metaheuristic optimization and was initially devised in Ref.⁹⁵. GWO mimics the hunting actions of grey wolves in the wild, a supporting approach they use to chase their prey. The framework of the GWO seems quite distinct compared to other meta-heuristic optimization in that it uses three optimal specimens as the basis for a complex search procedure. These three optimal specimens are an alpha wolf α that serves as the pack leader, a beta wolf β that provides support to the leader, and a delta wolf δ that follows the leader and the loyal wolves. The last kind of wolf is termed omega wolf ω . Such wolves have varying degrees of responsibility and can be arranged in a hierarchy, with α being the highest level and the first solution, β , δ , and ω representing the second, third, and final solutions, correspondingly. Thus, the three wolves mentioned above serve as inspiration for omegas. All species of wolves employ the three separate coefficients utilized to implement the encircling process to attempt to encompass the prey when they have located it. Three wolves evaluate the potential location of prey during the iterative search strategy. Based on Eqs. (7), (8), the positions of the wolf are updated during the optimization procedure.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_P(t) - \vec{X}(t) \right|, \quad (7)$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \cdot \vec{D}, \quad (8)$$

where t is the current iteration, \vec{C} and \vec{A} are coefficient vectors, \vec{X}_P signifies the prey's position, and \vec{X} signifies the wolf position. The vectors \vec{C} and \vec{A} are as follows:

$$\vec{A} = 2a \cdot \vec{r}_1 - \vec{a}, \quad (9)$$

$$\vec{C} = 2 \cdot \vec{r}_2, \quad (10)$$

where \vec{r}_1 and \vec{r}_2 signify random vectors in the range [0, 1], and factor a linearly falls from 2 to 0 with the number of iterations. The wolf at a location can change its position about the prey using the aforementioned updating algorithms. By changing the random parameters \vec{A} and \vec{C} , it may be made to move to any location in the continuous space close to prey. The GWO considers that the prey's position is likely in the alpha, beta, and delta positions. During searching, the best, second-best, and third-best individuals found so far are recorded as alpha, beta, and delta. Omega wolves, on the other hand, change their sites in accordance with alpha, beta, and delta wolf populations.

$$\left. \begin{aligned} \vec{D}_\alpha &= \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta &= \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta &= \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \end{aligned} \right\}. \quad (11)$$

The location vectors for α , β , and δ are, respectively, \vec{X}_α , \vec{X}_β , and \vec{X}_δ . The vectors \vec{C}_1 , \vec{C}_2 , and \vec{C}_3 were produced randomly, and \vec{X} indicates the current position vector. The distances between the position of the current person and those of alpha, beta, and delta are calculated, respectively, by Eq. (11). To determine the present person's final position matrices, the following are described.

$$\left. \begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha \right) \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \left(\vec{D}_\beta \right) \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \left(\vec{D}_\delta \right) \end{aligned} \right\}, \quad (12)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \quad (13)$$

where \vec{A}_1 , \vec{A}_2 , and \vec{A}_3 denote vectors that are randomly created, and t signifies the current iteration. The regulating factor that modifies the coefficient \vec{A} is variable a . This tactic aids the population in deciding whether to pursue or flee its prey. As a result, if $|a|$ has a value greater than 1, the wolf is trying to find new search spaces. However, the wolf could pursue and attack the prey if the value of $|a|$ is smaller than 1. The grey wolf starts to prevent any motion of the prey from attacking it once the hunting is accomplished adequately. This technique is accomplished by lowering the value of a , which is in the range of 2 and 0. The value of a also decreases the value of \vec{A} , which now falls between $[-1, 1]$. The pseudocode of the GWO is provided in Algorithm 1.

Initialize random population X_n ($n = 1, 2, \dots, dim$), X_α , X_β , and X_δ
 Initialize a , A , and C vectors
 Compute fitness value
While ($i < Max_it$)
 For each population
 Update search position by Eq. 13
 End For
 Update a , A , and C vectors
 Compute the fitness value
 Update X_α , X_β , and X_δ
 $i = i + 1$
End while
 Return X_α

Algorithm 1. Pseudocode of grey wolf optimizer.

K-means clustering-based grey wolf optimizer

In addition to the significance of optimization techniques, data analysis is a key research area. Clustering has thus been utilized as one of the data exploration approaches to gain a general understanding of the data's architecture. K-means is the most used unsupervised algorithm. Data points in each group resemble each other much more than those in other clusters. The method does a sequence of operations to identify unique subsets, which are discussed below.

- The number of subsets is the primary criterion for the K-means, which in data mining starts with the initial set of random centroids chosen for each cluster.
- The following step involves determining the Euclidean distance from the center to every data point in specific information set to connect each data with the closest point.
- Continue performing until there is no modification in the centroids if K centres shift during the iteration.

The algorithm attempts to minimize the squared error function or objective function presented in Eq. (14).

$$\sum_{i=1}^C \sum_{j=1}^{C_i} \|x_i^j - C_j\|^2, \quad (14)$$

where x_i^j signifies the data points of i th cluster, C_j denotes the size of the cluster center, and $\|x_i^j - C_j\|$ represents the Euclidean distance between C_j and x_i^j . The initialization of the proposed KCGWO is similar to the original version of GWO. K-means is utilized to separate the grey wolf population into three groups. The objective function value is then determined for each cluster/population individually¹⁰⁴. The population has been divided into three clusters based on a random integer. If the random number is greater than 0.5, KCGWO uses population clusters based on the fitness values of each cluster. All the clusters' fitness values are compared within the condition. The population position is equal to cluster position 1, position 2, or position 3 based on the conditions provided in the pseudocode. However, KCGWO operates on the actual population without clustering if the random value is less than or equal to 0.5. Therefore, this feature can be utilized with different methods, but it needs to be evaluated to ensure it functions well. However, K-means are utilized in this study to enhance the effectiveness of GWO. The proposed KCGWO tries to compute the fitness for each population after selecting a particular population with/without clustering until it discovers the best fitness. Equations (7)–(12) determine the optimum search agents. Equation (13) is then used to update each position. However, the weightage is not provided for the wolf hierarchy. Therefore, in Eq. (13), weight factors are introduced to improve the solution quality¹⁰⁵. The modified position update equation is provided in Eq. (15).

$$\vec{X}(t+1) = \frac{3\vec{X}_1 + 2\vec{X}_2 + \vec{X}_3}{6}, \quad (15)$$

The variables a , \vec{A} and \vec{C} are updated for the subsequent iteration. As a result, the iteration's best fit is chosen. Finally, the best fitness and position are returned. Figure 1 illustrates the flowchart of the proposed KCGWO algorithm. Algorithm 2 depicts the pseudocode of the KMGWO algorithm.

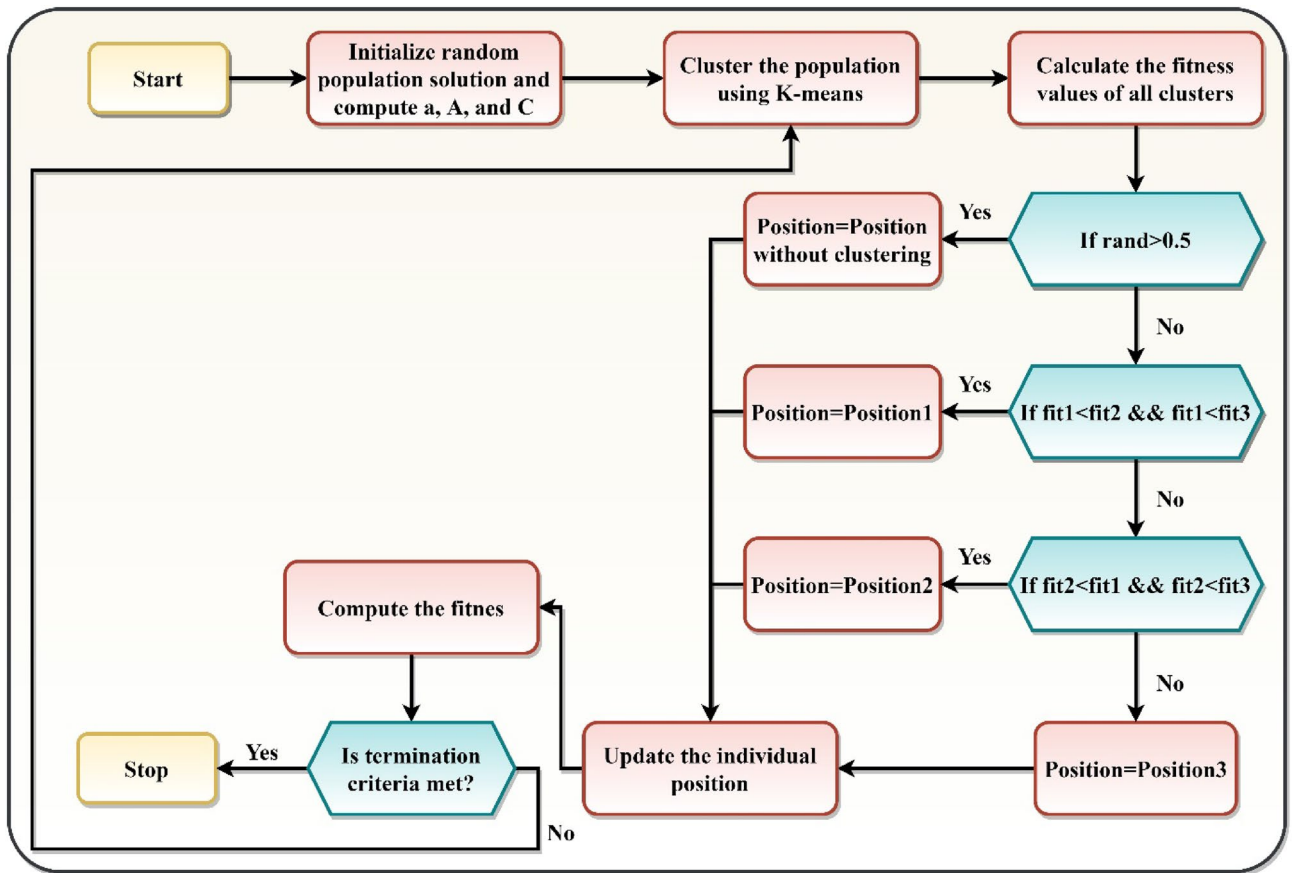


Figure 1. Flowchart of the proposed algorithm.

Application of the proposed KCGWO to data clustering

A crucial stage in every metaheuristic approach is solution encoding. Each solution (grey wolf) represents all the cluster centers. These solutions are first produced randomly. However, the best position at each iteration of the KCGWO serves as a guide for the remaining grey wolves.

Each answer is an array of size $d \times k$, with d being the total number of characteristics for each data and k being the total clusters. Figure 2 displays a pack of grey wolves representing the solutions. The fitness function is the total intra-cluster distance. The fitness function must be minimized to discover the best cluster centers using KCGWO. It is preferred to reduce the sum of intra-cluster distances⁹⁶. In Eq. (16), the cluster center is defined, and Eq. (17) defines the distances between cluster members.

$$Y_j = \frac{1}{n_j} \sum_{x_p \in C_j} x_p, \tag{16}$$

$$\text{Distance}(x_p - y_j) = \sqrt{\sum_{i=1}^a (x_p - y_j)^2} \tag{17}$$

where y_j denotes cluster center, x_p denotes the position of the p th cluster member, a denotes the number of features of the dataset, n_j denotes the members in the cluster j , and C_j denotes the cluster member j .

$X_1^{1,1}$	$X_1^{1,2}$...	$X_1^{1,d-1}$	$X_1^{1,d}$...	$X_1^{k,1}$	$X_1^{k,2}$...	$X_1^{k,d-1}$	$X_1^{k,d}$
$X_2^{1,1}$	$X_2^{1,2}$...	$X_2^{1,d-1}$	$X_2^{1,d}$...	$X_2^{k,1}$	$X_2^{k,2}$...	$X_2^{k,d-1}$	$X_2^{k,d}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$X_n^{1,1}$	$X_n^{1,2}$...	$X_n^{1,d-1}$	$X_n^{1,d}$...	$X_n^{k,1}$	$X_n^{k,2}$...	$X_n^{k,d-1}$	$X_n^{k,d}$

Figure 2. Illustration of solution encoding⁹⁶.

```

Initialize random population  $X_n$  ( $n = 1, 2, \dots, dim$ ),  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
Initialize  $a$ ,  $A$ , and  $C$  vectors
By Eq. 14, the population cluster is created
Compute fit1, fit2, and fit3
    If rand>0.5
        If fit1<fit2 && fit1<fit3
            Position=Position1
        ElseIf fit2<fit1 && fit2<fit3
            Position=Position2
        Else
            Position=Position3
        End If
    Else
        Position=Position
    End If
While ( $i < Max\_it$ )
    For each population
        Update search position by Eq. 15
    End For
    Update  $a$ ,  $A$ , and  $C$  vectors
    Compute the fitness value
    Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
     $i = i + 1$ 
End while
Return  $X_\alpha$ 

```

Algorithm 2. Pseudocode of the proposed KCGWO.

Computational complexity

The computational complexity of the KCGWO is discussed as follows: (i) The proposed algorithm necessitates $O(N \times dim)$, where N denotes the number of search agents, i.e. population size and dim denotes the problem dimension, (ii) the control parameters of KCGWO necessitates $O(N \times dim)$, (iii) the position update of the KCGWO necessitates $O(N \times dim)$, and (iv) fitness values of each population and cluster necessitate $O(N \times dim \times n)$, where n denotes the number of clustered population. Based on discussions, the complexity of KCGWO for each iteration is $O(N \times dim \times n)$, and finally, the total complexity of the proposed KCGWO algorithm is $O(N \times dim \times n \times Max_it)$, where Max_it denotes the maximum of iterations.

Results and discussions

The original GWO is improved by employing the K-means clustering concept along with the weight factor, and it has been tested using 10 benchmark numerical functions, which have both unimodal and multimodal features. In addition, the performance is also validate for data clustering problems. The performance of the proposed KCGWO is compared with four other algorithms, such as MFO, SSA, MVO, ASO, PSO, JAYA, and the original GWO algorithm. The population size is 30, and the maximum number of iterations is 500 for all selected algorithms. All the algorithms are implemented using MATLAB software installed on a laptop with an i5 processor, a 4.44 GHz clock frequency, and 16 GB of memory. The algorithms are executed 30 times individually for a fair comparison.

Results for numerical optimization functions

The details of the selected benchmark functions are recorded in Table 2. The functions F1-F4 have unimodal features with 30 dimensions, F5-F7 have multimodal features with 30 problem dimensions, & F8-F10 have multimodal features with very low dimensions. The purpose of selecting the listed benchmark function is to analyze the exploration and exploitation behaviour of the developed KCGWO algorithm. The statistical measures, such

Function	Dim	Range	f_{\min}
$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[-100, 100]	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[-32, 32]	0
$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$f_8(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_9(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[-5, 5]	0.398
$f_{10}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

Table 2. 10 benchmark test functions for validation.

as minimum (Min), Mean, maximum (Max), and Standard Deviation (STD) of all designated algorithms, are recorded in Table 3.

Classifying functions F01–F04 as unimodal test scenarios with a single global best is appropriate. Such test sets can be used to look into the general exploitation potential of the proposed KCGWO approach. Findings of the proposed KCGWO and other approaches, as well as their Min, Max, Mean, and STD, are shown in Table 2. The associated tables' higher outcomes are noted. The optimization techniques are then ordered based on their average values. The average rank is also calculated to determine the approaches' overall ranking. All Tables include a summary of the findings of the statistical analysis. The best results are emphasized in bold face in all tables. For each unimodal function, individual ranking is provided to examine the performance of the proposed algorithm. The proposed algorithm stands first out of all selected algorithms for all four unimodal functions.

The F1–F04 shows that the KCGWO can arrive at capable making with a suitable exploitation capability. This seems to be due to the effectiveness with which the suggested K-means clustering concept and weight factors can boost the GWO's tendencies for exploration and exploitation. As a result, the mechanisms make the algorithm more likely to produce smaller fitness and higher stability index values. This tool helps explore new locations close to the recently discovered results. Because of this, it was found that the new algorithmic changes have improved how GWO handles unimodal test cases. Assessing the exploration potential using multimodal functions (F5–F10) is reasonable. Table 2 shows that KCGWO can investigate highly competitive solutions for the F5–F10 test scenarios. The KCGWO can produce optimal results for all test functions compared to other approaches. According to the results, KCGWO can outperform all selected algorithms in multimodal instances. Additionally, statistical analyses show that, in 95% of evaluations, KCGWO outcomes are superior to those of other approaches. Compared to GWO, the accuracy is increased based on the STD index.

In particular, when the objective problems (F5–F8) involve several local optima, KCGWO's outperformance demonstrates a sufficient explorative nature. This is due to the effectiveness with which the K-means clustering structure can boost the GWO's performance for exploration and exploitation. Lower stability index values can encourage wolves to make more exploratory jumps. This feature might be seen when KCGWO requires investigating previously unexplored regions of the issue landscape. The weight factors have helped GWO achieve a delicate balance between its local and global search inclinations. According to the findings, the recommended K-means searching steps increase the GWO's exploration capability. Additionally, the KCGWO's update mechanism can lessen the likelihood of the KCGWO entering local optima. The exploratory propensity of KCGWO is hence advantageous. The computational complexity of the proposed algorithm is assessed by recording the RunTime (RT). The RT values for each function by all selected algorithms are recorded in Table 4. The average values of RT are also provided, and based on the mean RT value, the original GWO has less RT value, and the RT value of KCGWO is slightly greater than the GWO, which is due to the fact that the introduction of the K-means clustering mechanism. At the same time, the weight factor does not impact the proposed algorithm's computational complexity.

Figure 3 shows all selected algorithms' convergence characteristics for handling F1–F10 functions. All selected algorithms consistently outperform for a benchmark and have excellent convergence outfits in the original publication. Figure 3 also offers a convergence timeframe. It pinpoints the times when KCGWO performs better than GWO. According to Fig. 3, KCGWO eventually converges to superior outcomes. A large number of iterations allows KCGWO to approximate more precise solutions close to the optimum solutions. Additionally, rapid convergence patterns may be seen when comparing the curves of KCGWO and its original version. This pattern demonstrates that KCGWO can emphasize more exploitation and local search in the latter stages. These plots suggest that the KCGWO can successfully increase all wolves' fitness and promise to exploit improved results. In order to visualize the stability analysis, the boxplots are also plotted and shown in Fig. 4. From Fig. 4, it is detected that the stability of the KCGWO is better than all selected algorithms.

To further assess the performance of the proposed algorithm, the statistical non-parametric test, Friedman's Ranking Test (FRT), has been conducted, and the average FRT values of all algorithms are logged in Table 5.

Functions	Metrics	KCGWO	GWO	MFO	MVO	SSA	JAYA	AOS	PSO
f_1	Min	0.00E+00	2.65E-29	3.00E+02	8.70E-01	3.41E-08	0.887679	2.84E-80	899.5347
	Max	1.74E-158	8.74E-28	5.42E+02	1.73E+00	6.13E-07	72.8437	3.79E-76	11,137.22
	Mean	<i>0.00E+00</i>	1.82E-28	3.24E+02	1.70E+00	1.90E-07	1.454592	3.81E-80	1032.142
	STD	1.01E-158	4.51E-28	1.33E+02	4.88E-01	3.00E-07	0.234375	3.859375	0.052083
	Rank	1	3	7	5	4	6	2	8
f_2	Min	0.00E+00	9.51E-18	1.48E+01	5.95E-01	1.35E+00	1.17E-02	7.67E-55	1.67E+01
	Max	0.00E+00	2.98E-16	2.63E+01	8.89E-01	4.58E+00	1.75E-01	3.73E-53	4.30E+01
	Mean	<i>0.00E+00</i>	6.38E-17	2.42E+01	6.93E-01	3.00E+00	1.02E-01	1.72E-53	2.40E+01
	STD	0.00E+00	1.54E-16	6.11E+00	1.50E-01	1.61E+00	9.90E-02	3.17E+00	4.17E-02
	Rank	1	3	8	5	6	4	2	7
f_3	Min	2.65E-99	3.85E-07	1.38E+04	1.66E+02	6.98E+02	9.35E+03	2.85E+04	7.62E+03
	Max	9.44E-78	8.42E-04	4.61E+04	2.19E+02	2.26E+03	1.34E+04	5.31E+04	3.98E+04
	Mean	<i>4.84E-96</i>	5.16E-05	2.17E+04	1.76E+02	1.02E+03	1.06E+04	3.20E+04	3.51E+04
	STD	5.45E-78	4.72E-04	1.68E+04	2.83E+01	8.22E+02	1.51E-01	3.34E+00	1.20E-01
	Rank	1	2	6	3	4	5	7	8
f_4	Min	0.00E+00	2.02E-07	5.19E+01	1.09E+00	8.02E+00	1.65E+01	1.07E+01	6.90E+01
	Max	2.96E-84	6.98E-07	7.18E+01	2.13E+00	1.36E+01	3.80E+01	7.70E+01	7.50E+01
	Mean	<i>3.50E-87</i>	2.22E-07	6.58E+01	1.74E+00	9.02E+00	1.70E+01	3.58E+01	7.16E+01
	STD	1.71E-84	2.81E-07	1.02E+01	5.27E-01	2.97E+00	5.73E-02	3.16E+00	5.21E-02
	Rank	1	2	7	3	4	5	6	8
f_5	Min	0.00E+00	9.66E-13	1.16E+02	8.15E+01	4.48E+01	1.25E+00	0.00E+00	1.32E+02
	Max	0.00E+00	5.40E+00	1.41E+02	1.30E+02	5.77E+01	1.26E+01	0.00E+00	1.75E+02
	Mean	<i>0.00E+00</i>	1.59E-12	1.36E+02	1.26E+02	4.58E+01	7.20E+00	0.00E+00	1.63E+02
	STD	0.00E+00	3.12E+00	1.32E+01	2.70E+01	7.20E+00	8.85E-02	3.01E+00	4.17E-02
	Rank	1	3	6	5	8	4	2	7
f_6	Min	8.88E-16	1.00E-13	2.00E+01	1.10E+00	1.78E+00	2.02E+01	4.44E-15	2.00E+01
	Max	8.88E-16	1.11E-13	2.00E+01	2.16E+00	3.35E+00	2.02E+01	4.44E-15	2.00E+01
	Mean	<i>8.88E-16</i>	1.04E-13	2.00E+01	1.21E+00	2.89E+00	2.02E+01	4.44E-15	2.00E+01
	STD	0.00E+00	5.43E-15	5.34E-05	5.86E-01	8.06E-01	9.90E-02	2.91E+00	5.73E-02
	Rank	1	3	6	4	5	8	2	7
f_7	Min	0.00E+00	0.00E+00	6.98E+00	8.38E-01	4.94E-03	1.03E+00	0.00E+00	5.85E+00
	Max	0.00E+00	1.23E-02	1.01E+02	9.33E-01	1.96E-02	1.09E+00	0.00E+00	6.77E+01
	Mean	<i>0.00E+00</i>	<i>0.00E+00</i>	4.36E+01	8.42E-01	1.23E-02	1.08E+00	0.00E+00	6.28E+00
	STD	0.00E+00	7.12E-03	4.73E+01	5.37E-02	7.32E-03	7.81E-02	2.86E+00	6.25E-02
	Rank	1	2	8	5	4	6	3	7
f_8	Min	-1.03163	-1.03163	-1.03163	-1.03163	-1.03162	-1.0316	-1.0316	-1.0316
	Max	-1.03163	-1.03163	-1.03163	-1.03163	-1.03152	-1.0316	-1.0316	-1.0316
	Mean	<i>-1.03163</i>	<i>-1.03163</i>	<i>-1.03163</i>	<i>-1.03163</i>	-1.03162	-1.0316	-1.0316	-1.0316
	STD	0	6.62E-08	0	6.21E-07	5.64E-05	1.93E-05	2.35E-10	1.57E-16
	Rank	1	5	1	6	8	7	4	3
f_9	Min	0.397887	0.397888	0.397887	0.397887	0.39793	0.39823	0.39789	0.39789
	Max	0.397887	0.397898	0.397887	0.397888	0.399332	0.39918	0.39790	0.39789
	Mean	<i>0.397887</i>	0.397889	<i>0.397887</i>	<i>0.397887</i>	0.398836	0.39872	0.39789	0.39789
	STD	0	5.64E-06	0	9.32E-08	0.000711	4.75E-04	4.99E-06	0.00E+00
	Rank	1	6	1	4	8	7	5	1
f_{10}	Min	-10.5364	-10.5351	-10.5364	-10.5361	-4.96056	-4.7879	-10.5248	-3.8354
	Max	-10.5364	-10.532	-2.80663	-10.5355	-2.91529	-2.3747	-3.8349	-2.4217
	Mean	<i>-10.5364</i>	-10.5341	-10.5364	-10.536	-3.09438	-2.7739	-5.1264	-3.8354
	STD	2.81E-15	0.001572	4.46279	0.00034	1.132679	1.29E+00	3.55E+00	8.16E-01
	Rank	1	4	3	2	7	8	5	6
Average rank	1	3.3	5.3	4.2	5.8	6	3.8	6.2	

Table 3. Results obtained for 10 benchmark functions by all selected algorithms. Significant values are in bold and italic.

Functions	KCGWO	GWO	MFO	MVO	SSA	JAYA	AOS	PSO
f_1	0.276	0.1875	0.3073	0.4948	0.1927	0.2844	3.8594	0.3521
f_2	0.0833	0.0521	0.1198	0.1146	0.0885	0.1198	2.8490	0.1469
f_3	0.1823	0.1771	0.1563	0.2188	0.1563	0.2292	0.2240	0.2552
f_4	0.0833	0.0469	0.0625	0.099	0.0625	0.1667	0.1302	0.1042
f_5	0.1094	0.0885	0.1094	0.1563	0.1927	0.1254	0.1356	0.1563
f_6	0.0625	0.0469	0.0729	0.1198	0.1198	0.1667	0.1302	0.1042
f_7	0.0781	0.0625	0.0833	0.1042	0.1458	0.1719	0.1927	0.2292
f_8	0.0677	0.0781	0.1094	0.1615	0.0729	0.1347	0.1451	0.1289
f_9	0.0365	0.0781	0.0677	0.125	0.0677	0.0598	0.0747	0.0909
f_{10}	0.0521	0.026	0.0469	0.0677	0.0521	0.0789	0.1151	0.0994
Average RT	0.10312	0.08437	0.11355	0.16617	0.1151	0.15375	0.7856	0.16673

Table 4. RT values of each test functions.

Based on the observation, the proposed algorithm attains the top of the table with an average FRT of 1.383, followed by GWO, AOS, MVO, SSA, MFO, JAYA, and PSO.

These statistics indicate that the K-mean clustering approach and modified position update equation based on the weight factors can enhance the search functionality of GWO. Comparing the suggested KCGWO to existing approaches with superior convergence characteristics can be more effective.

Results for data clustering problems

The suggested clustering approach was thoroughly assessed using eight datasets. Few datasets used are synthetic, and others are drawn from real-time benchmark data. Table 6 provides a summary of the traits of these datasets¹⁰⁶. The features (dimensions), the total number of samples, and the number of clusters in each dataset are recorded in Table 6. The type of the problems is also mentioned. The dataset is selected based on the type and number of samples.

The performance of the proposed KCGWO for clustering is initially compared with standalone K-means clustering algorithms and the Gaussian Mixture Model (GMM). The non-linear, unsupervised t-distributed Stochastic Neighbor Embedding (t-SNE) is typically employed for data analysis and high-dimensional data visualization. For all the selected datasets, t-SNE plots obtained by KCGWO, GMM, and K-means are plotted in Figs. 5, 6, 7, 8, 9, 10, 11 and 12. Figure 5a displays the emission data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figure 5b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster centre of each cluster found by the K-means algorithm is also demonstrated in Fig. 5c.

Figure 6a displays the HTRU2 data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figure 6b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 6c. Figure 7a displays the Wine data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figures 7b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 7c.

Figure 8a displays the Breast cancer data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figure 8b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 8c. Figure 9a displays the Sonar data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figures 9b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 9c.

Figure 10a displays the WDBC data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figure 10b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 10c. Figure 11a displays the Iris data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. Figure 11b and c show the t-SNE plots obtained by the GMM and K-means algorithms. The cluster center of each cluster found by the K-means algorithm is also demonstrated in Fig. 11c.

Figure 12a displays the 2022 Ukraine-Russia war data distribution obtained by the KCGWO between various dimensions. It also shows how well the high-dimensional data are distributed in 2-dimensions. According to Figs. 5, 6, 7, 8, 9, 10, 11 and 12, in data with convex-shaped clusters, KCGWO has been capable of recognizing clusters and discriminating overlap among clusters quite effectively. This demonstrates how clearly defined the differences between the clusters are. According to Figs. 5, 6, 7, 8, 9, 10, 11 and 12, KCGWO could cluster most of the data points accurately despite the high density and large scatter of sample points in the dataset. This shows that KCGWO is resistant to high data volume and dispersion. Additionally, it has been demonstrated

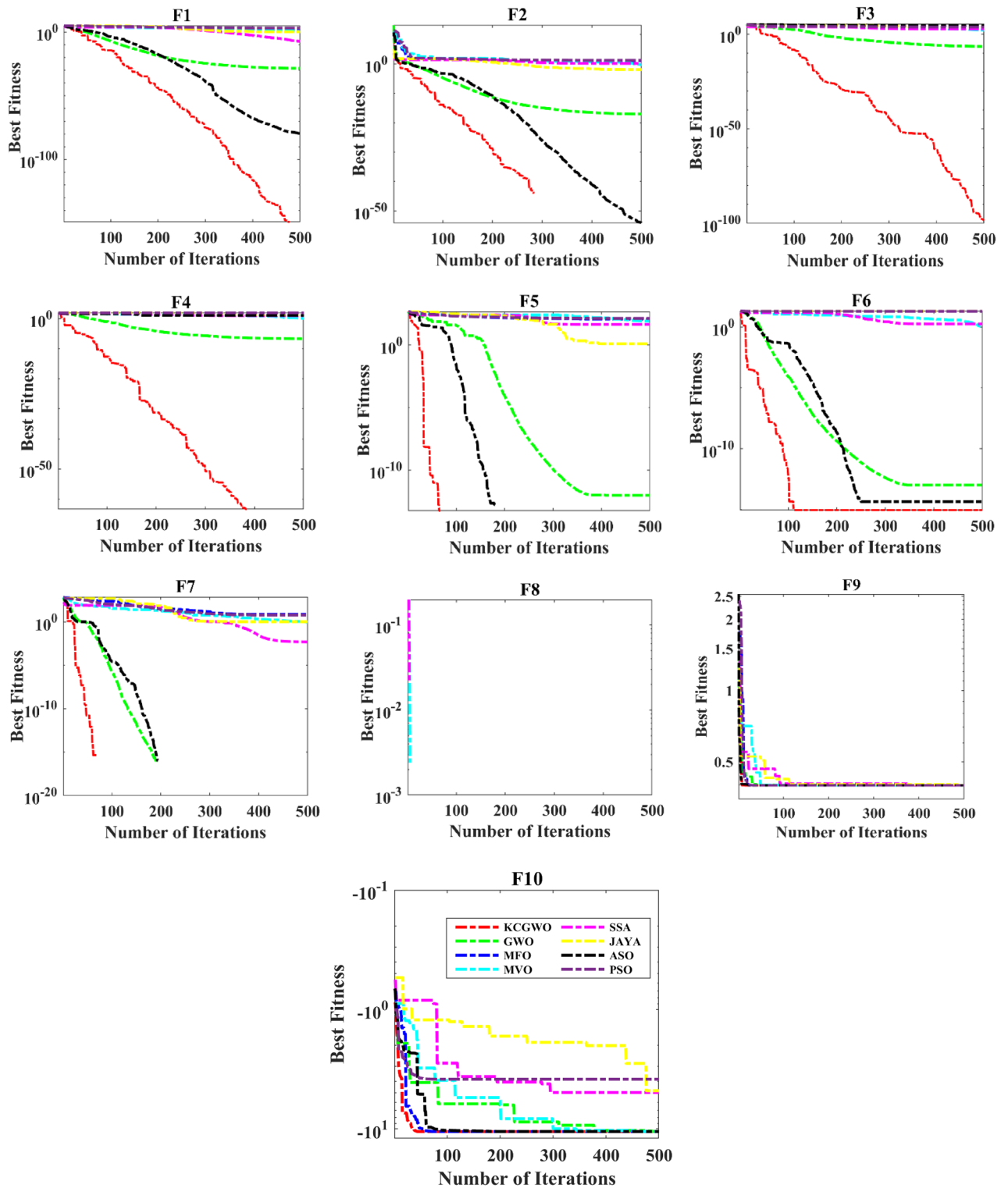


Figure 3. Convergence curve obtained by all algorithms.

that KCGWO performs effectively when dealing with circular clusters in difficult datasets. KCGWO successfully identifies the majority of the curved regions in these datasets. Due to the utilization of the Euclidean distance measure for clustering, the proposed KCGWO has not completely distinguished all of the clusters in the data.

In order to prove the performance of the proposed KCGWO, two additional metrics, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE), are recorded in Table 7. The average MAE and MSE values obtained by KCGWO with respect to GMM and K-means are listed in Table 7. Based on the average values, it is observed that the performance of the KCGWO with respect to K-means is better than GMM. Based on

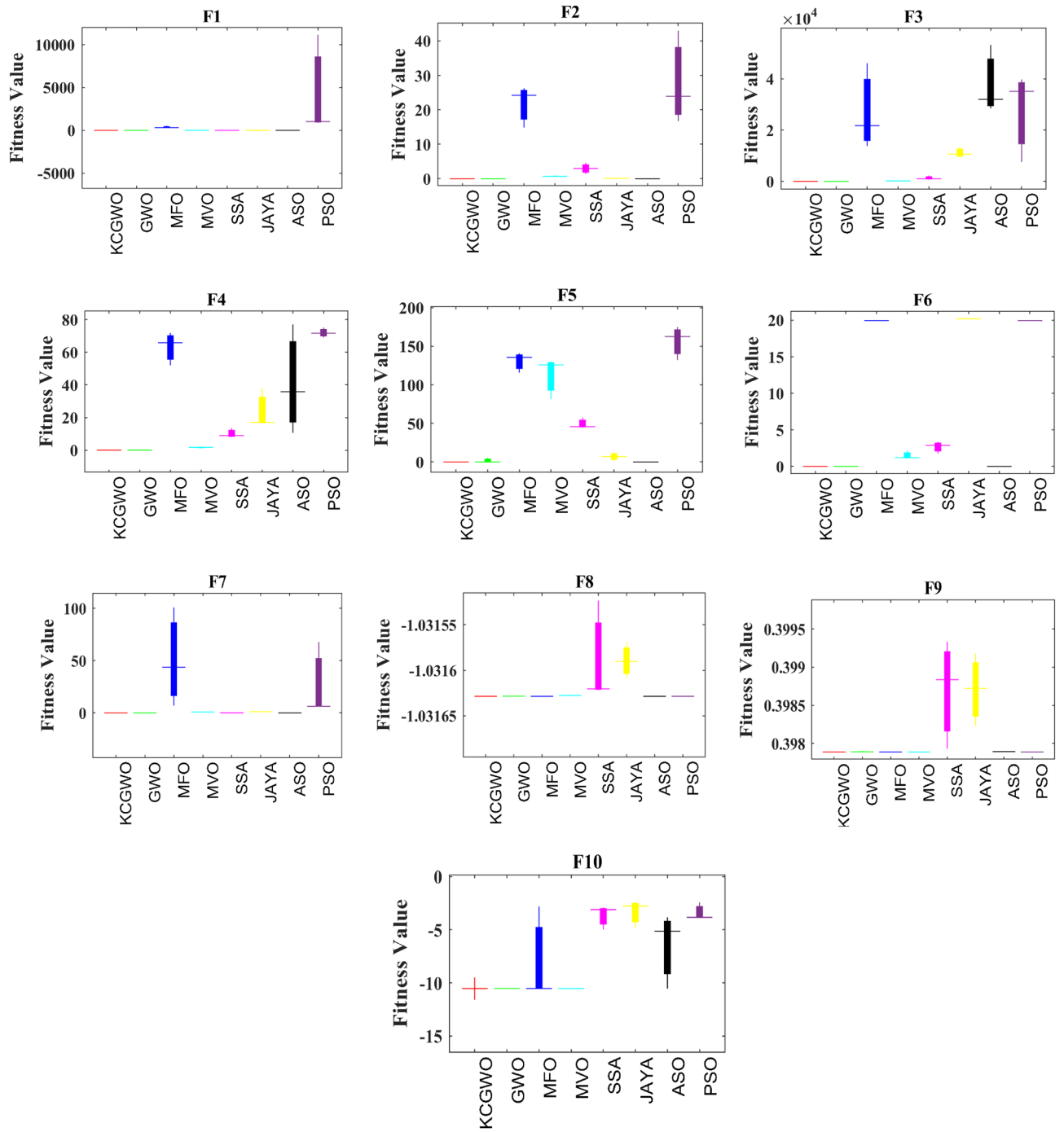


Figure 4. Boxplot analysis of all selected algorithms.

the comparison of the GMM with respect to K-means, it is observed that GMM is performing better than the K-means clustering algorithm. The results show that KCGWO produced results with more accuracy than GMM and K-means. One way to look at this enhancement is due to the population distribution by K-means and the weight factors, which avoids early convergence and strikes a compromise between global and local searches. Conversely, the proposed KCGWO has significantly enhanced effectiveness in data with significant overlapping and difficulty. As a result, KCGWO outperformed GMM and K-means and improved the ability to identify non-linear clusters.

Further, to have a fair comparison, the performance of the proposed algorithm is also compared with other metaheuristic algorithms, such as GWO, MFO, MVO, and SSA, in terms of the statistical measures, such as Min, Mean, Max, and STD. For all algorithms, the population size is carefully chosen as the number of clusters multiplied by 2, and the iteration count is 500. Table 8 recorded all the statistical measures of all selected algorithms and datasets. It is noticed from Table 8 that the KCGWO can able to attain the best Min values for all datasets.

Functions	KCGWO	GWO	MFO	MVO	SSA	JAYA	AOS	PSO
f_1	1.000	3.000	7.000	5.667	4.000	5.333	2.000	8.000
f_2	1.000	3.000	7.667	5.000	6.000	4.000	2.000	7.333
f_3	1.000	2.000	7.000	3.000	4.000	5.333	7.333	6.333
f_4	1.000	2.000	7.000	3.000	4.000	5.333	6.333	7.333
f_5	1.000	2.333	7.333	4.667	4.333	6.000	2.667	7.667
f_6	1.667	4.333	7.333	4.667	1.333	6.000	3.000	7.667
f_7	2.000	2.000	7.667	4.667	6.000	4.333	2.000	7.333
f_8	1.833	5.000	1.833	6.000	7.333	7.667	4.000	2.333
f_9	2.000	5.333	2.000	4.333	7.667	7.333	5.333	2.000
f_{10}	1.333	3.667	3.000	2.667	6.000	7.333	5.333	6.667
Average FRT	1.383	3.267	5.783	4.367	5.067	5.867	4.000	6.267

Table 5. FRT values of all algorithms for each test functions. Significant values are in bold.

Datasets	Dimensions	Samples	Cluster	Type
Emission	4	7384	6	Real-world
HTRU2	4	17,898	6	Artificial
Wine	14	178	2	Artificial
Breast cancer	30	569	2	Real-world
Sonar	60	208	2	Artificial
WDBC	30	568	2	Real-world
Iris	4	150	3	Real-world
2022 Ukraine Russia war	11	194	4	Real-world

Table 6. Details of the selected dataset¹⁰⁶.

The proposed algorithm can converge to the global optima and find the best solution. Except for the WDBC dataset, the proposed algorithm's maximum values are better. The mean and STD values obtained by the proposed algorithm are better than any other algorithms for all selected datasets. It means that the reliability of KCGWO is better than any other selected algorithms for all selected datasets. For each dataset, the ranking is provided based on the Min values, and the average rank values are also logged in Table 8. Based on the mean rank values, KCGWO stands first, followed by SSA, GWO, MFO, and MVO.

The following ratios of sequential errors describe the convergence rate given an undetermined optimal value, which is typically the case in data clustering applications:

$$\text{Convergence Rate}(CR) = \frac{|f_{i+1} - f_i|}{|f_i - f_{i-1}|}, \quad (18)$$

where f_i denotes the fitness value during the current iteration, f_{i+1} denotes the fitness value during the next iteration, and f_{i-1} denotes the fitness value during the previous iteration. The logarithmic CR plot measures the dynamic fitness change all over the iteration. The curves of logarithmic convergence curves are illustrated in Fig. 13 to visualize the effect on the various datasets. Comparatively to the other configurations, such as GWO, MFO, MVO, and SSA, using K-means clusters with weight factors in GWO has produced a good convergence that avoids the local optimum trap, with the lowest MAE and MSE values occurring at iteration 500. The adopted mechanism in the GWO algorithm maintained a reasonable balance between them and produced suitable population patterns for exploration and exploitation. In addition to the convergence curve, the boxplot analysis is also made to prove the reliability of the algorithms selected. All the algorithms are executed 30 times. The boxplots are plotted and illustrated based on the recorded values in Fig. 14.

From Fig. 14, it is clearly evident that the reliability of the KCGWO is superior to all the selected algorithms. The computational time necessary by the algorithm to find the overall optimal solution is known as the time to best solution. The RT of an algorithm is the sum of all computations performed until its stopping criterion stops it. Therefore, the RT is recorded for the selected algorithms and recorded in Table 9.

Similar to numerical optimization problems, the average RT values are provided in Table 9, and based on the mean RT value, the original GWO has less RT value, and the RT value of KCGWO is slightly greater than the GWO, which is due to the fact that the introduction of the K-means clustering mechanism. At the same time, the weight factor does not impact the proposed algorithm's computational complexity. It is clear from the prior comparisons and discussions that the improvisation of GWO performance with K-Means clustering and weight factors has accomplished its objectives and improved the original GWO algorithm. The new adjustments enabled

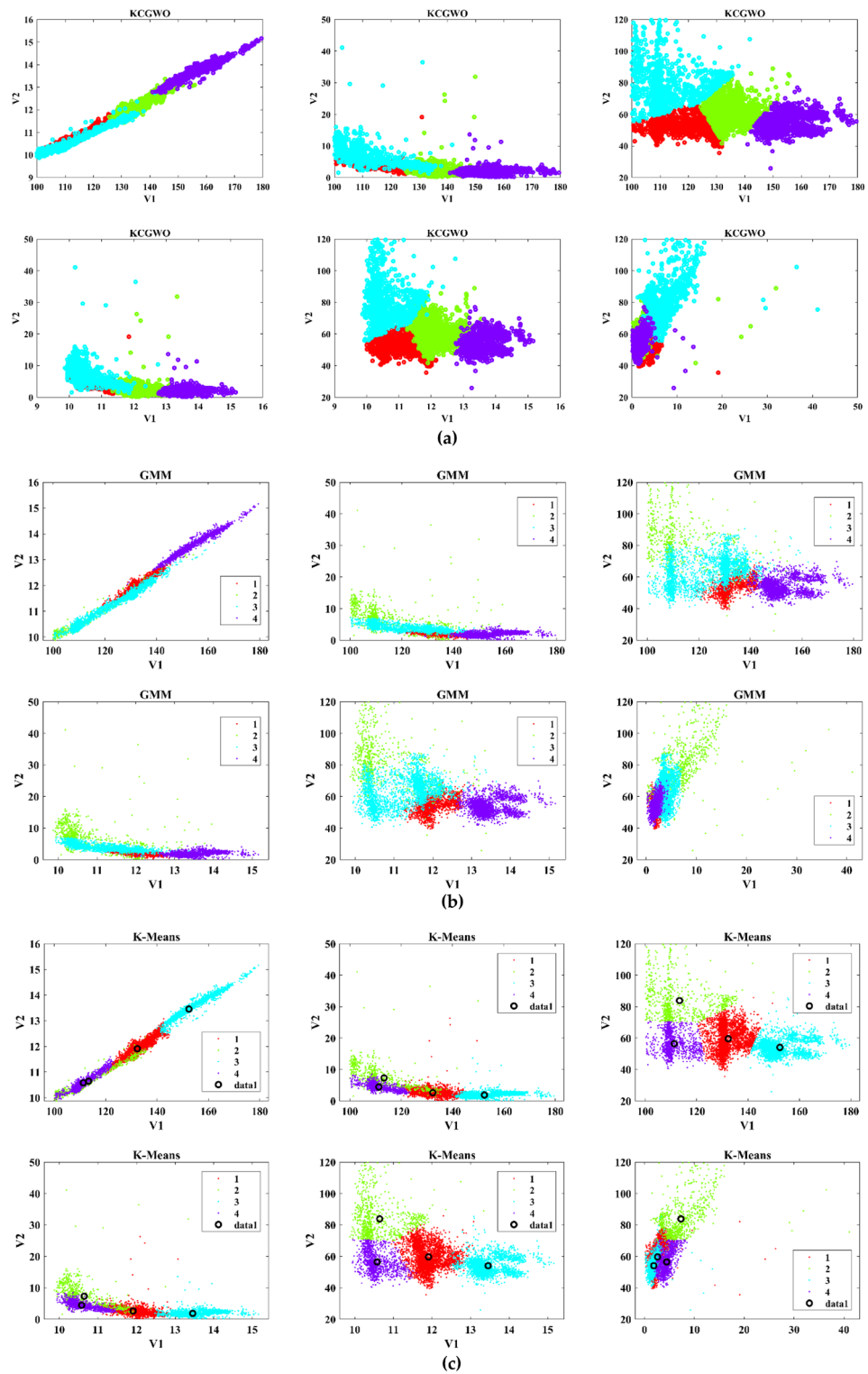


Figure 5. T-SNE plots of emission data: (a) KCGWO, (b) GMM, (c) K-means.

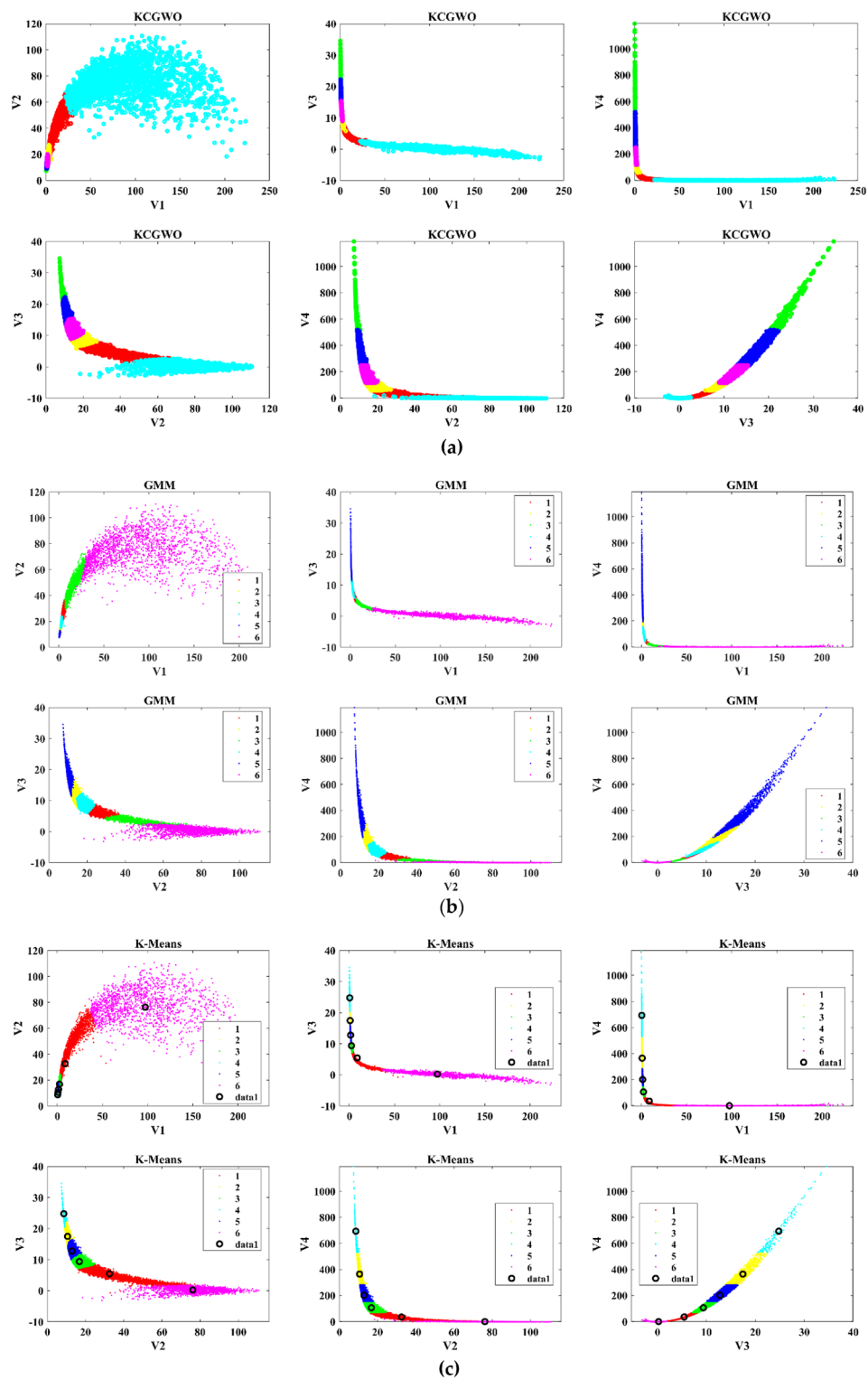


Figure 6. T-SNE plots of HTRU2 data; (a) KCGWO, (b) GMM, (c) K-means.

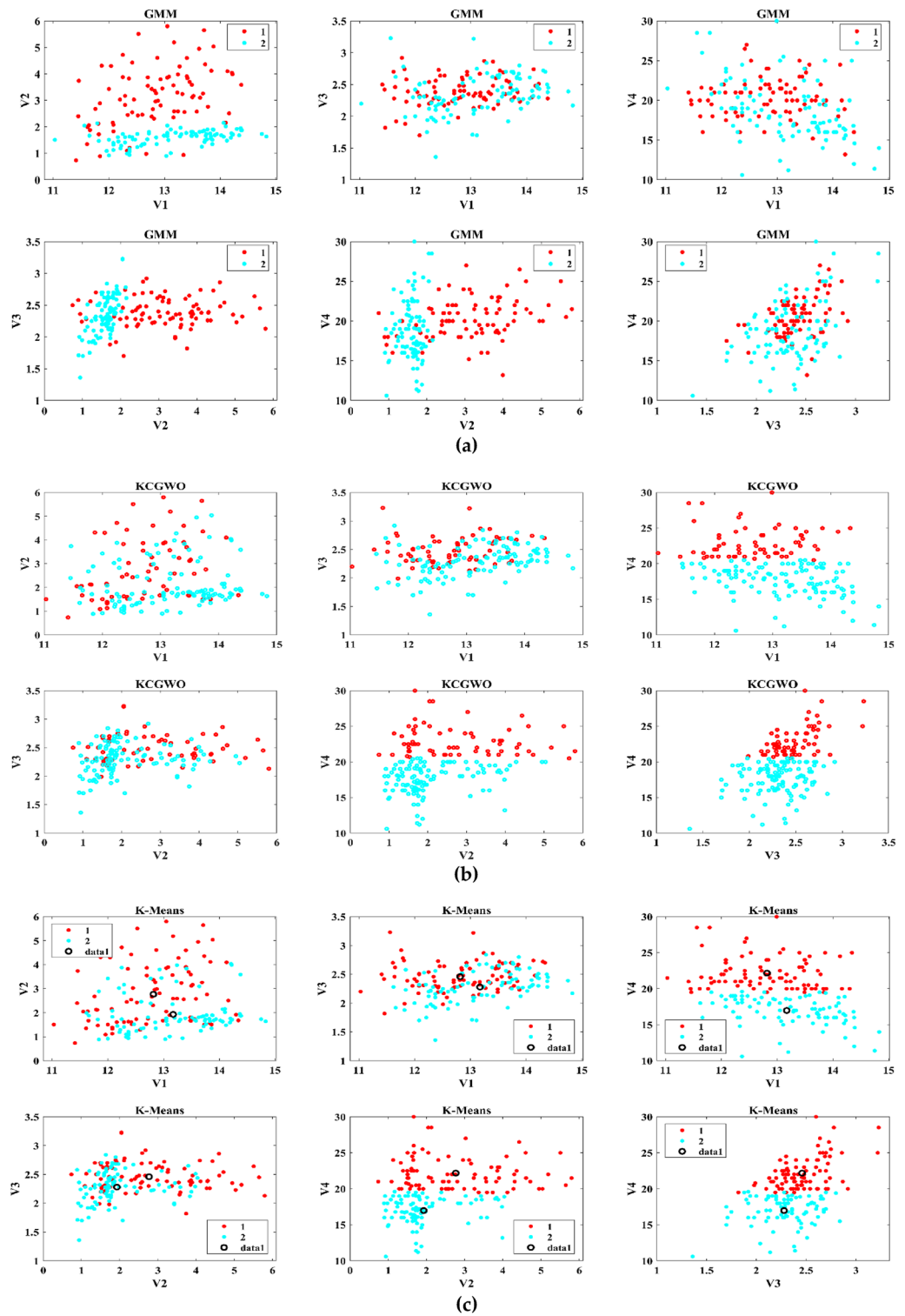


Figure 7. T-SNE plots of Wine data; (a) KCGWO, (b) GMM, (c) K-means.

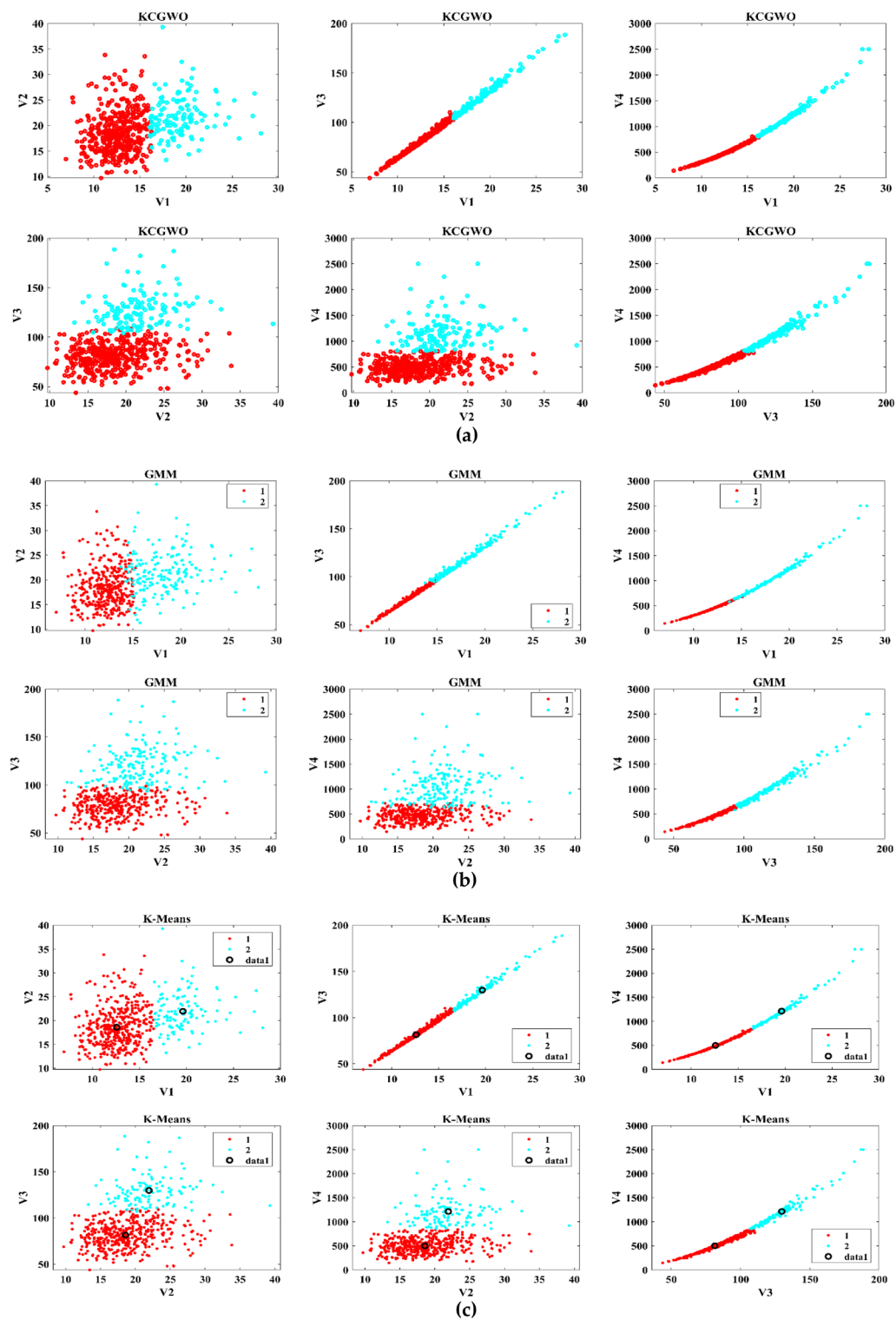


Figure 8. T-SNE plots of Breast cancer data; (a) KCGWO, (b) GMM, (c) K-means.

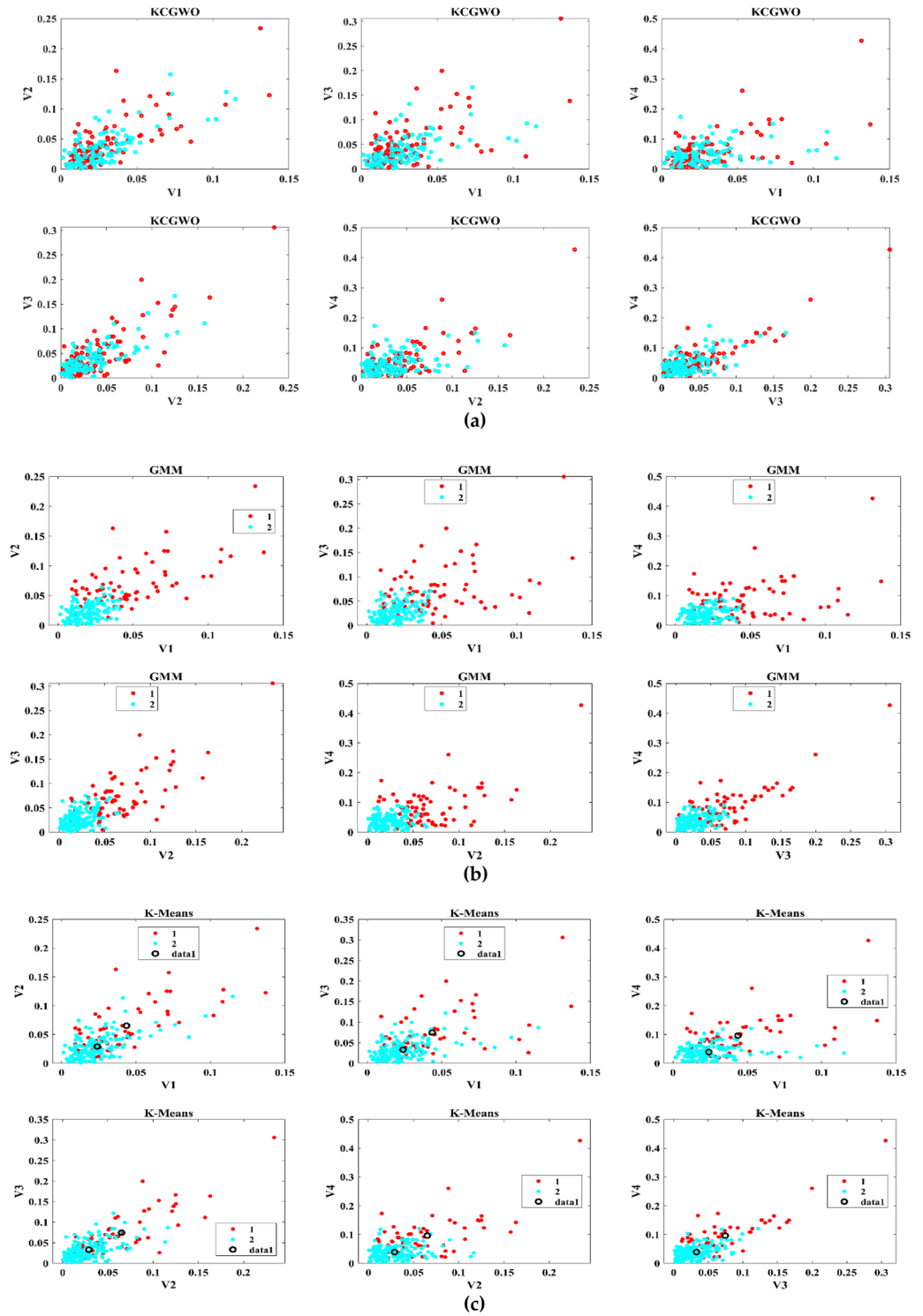


Figure 9. T-SNE plots of Sonar data; (a) KCGWO, (b) GMM, (c) K-means.

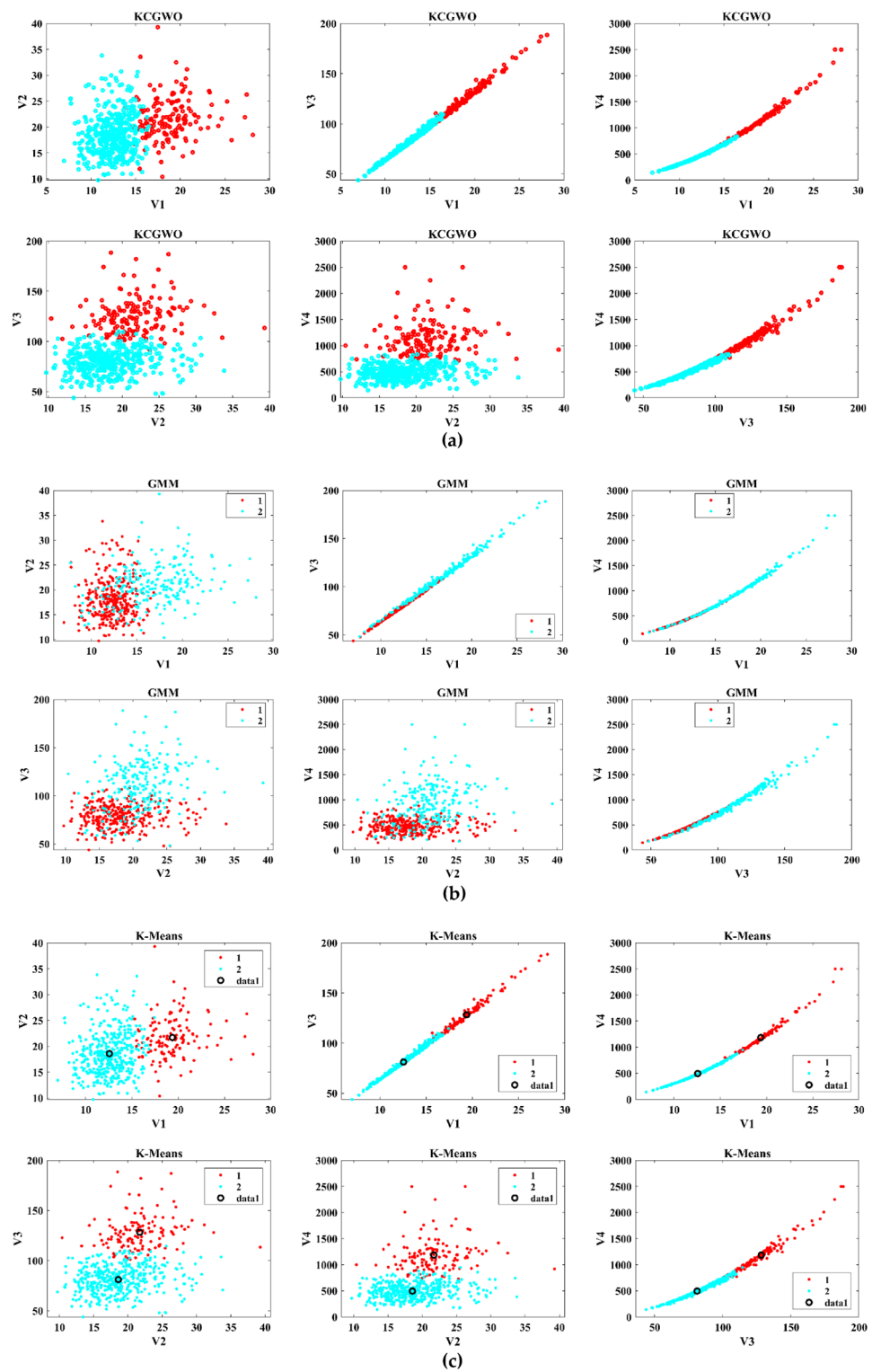


Figure 10. T-SNE plots of WDBC data; (a) KCGWO, (b) GMM, (c) K-means.

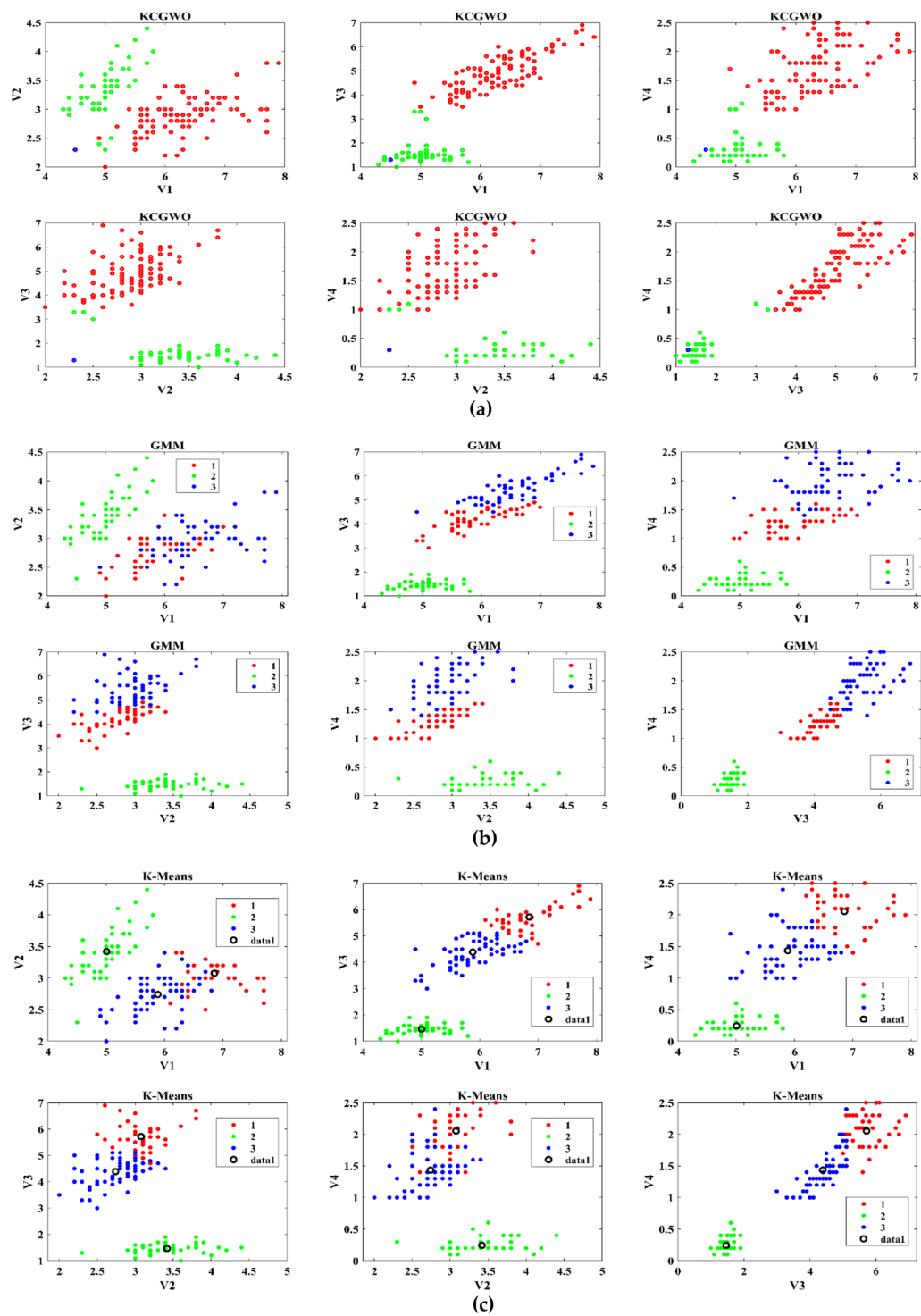


Figure 11. T-SNE plots of iris data; (a) KCGWO, (b) GMM, (c) K-means.

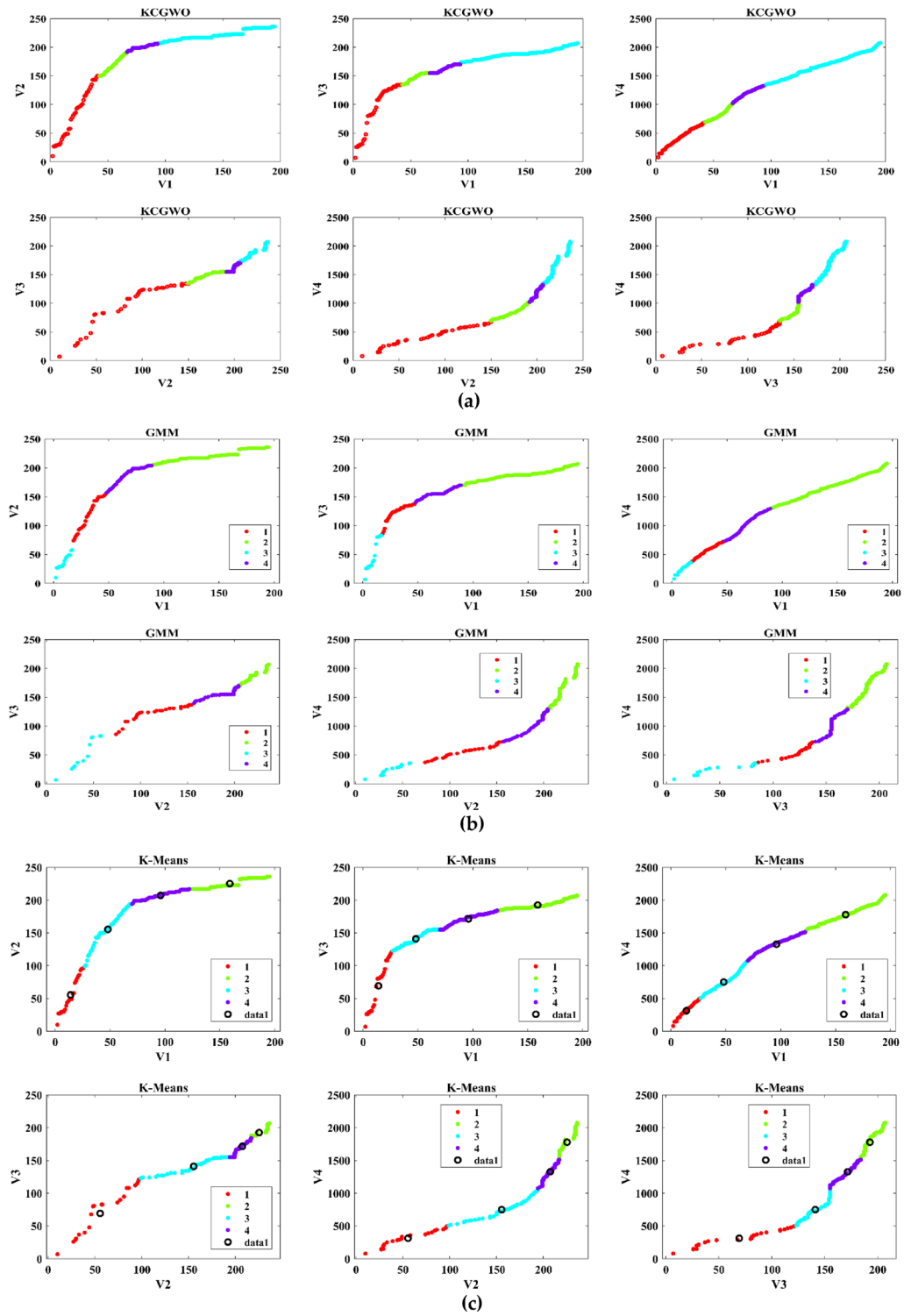


Figure 12. T-SNE plots of 2022 Ukraine-Russia war data; (a) KCGWO, (b) GMM, (c) K-means.

Dataset	MAE				MSE			
	KCGWO vs GMM	KCGWO vs K-means	GMM vs K-means	KCGWO vs K-means (centers)	KCGWO vs GMM	KCGWO vs K-means	GMM vs K-means	KCGWO vs K-means (centers)
Emission	0.6889	1.1532	0.9347	333.5077	0.9378	1.7564	1.4076	12.6256
HTRU2	1.9047	1.1920	1.3160	39,612.27	5.5686	2.5295	3.1712	96.1929
Wine	0.3876	0.1292	0.3483	0.3215	0.3876	0.1292	0.3483	0.2505
Breast cancer	0.0845	0.0194	0.1039	12.2918	0.0845	0.0194	0.1039	760.4808
Sonar	0.4615	0.5769	0.2115	0.0358	0.4615	0.5769	0.2115	0.0027
WDBC	0.8576	0.0334	0.8313	17.8233	0.8576	0.0334	0.8313	3384.228
Iris	0.7600	0.8000	1.0667	0.6654	1.4933	1.5733	2.1333	1.2273
2022 Ukraine Russia war	0.9536	0.8299	0.8351	275.3480	1.3557	0.9948	1.5567	254,083.1
Average values	0.7623	0.5917	0.7059	5031.533	1.3933	0.9516	1.2204	32,292.26

Table 7. MAE and MSE values obtained by KCGWO, GMM, and K-means.

KCGWO to defeat numerous original and other selected algorithms, presenting KCGWO as a global optimizer and an efficient data clustering technique that can be applied in industrial applications.

Discussions

While the KCGWO introduces significant improvements to the conventional GWO, enhancing its applicability to data clustering tasks, it is not without its limitations. These constraints, inherent to the methodology and application context, warrant consideration for future research and practical implementation. KCGWO's performance is partly contingent on the initial clustering obtained from the K-means algorithm. This dependence means that the quality of KCGWO's outcomes can be affected by the initial positioning of centroids in K-means, which is sensitive to the chosen initial points. If the K-means algorithm converges to a local optimum during its initialization phase, KCGWO may start from a less advantageous position, potentially impacting the overall optimization process. The introduction of a dynamic weight factor in KCGWO, while beneficial for balancing exploration and exploitation, adds complexity in terms of parameter tuning. The performance of KCGWO can be sensitive to the settings of this weight factor alongside other algorithm parameters. Finding the optimal configuration requires extensive experimentation and can be computationally demanding, especially for large-scale problems or datasets with high dimensionality. Although KCGWO is designed to explore and exploit the solution space efficiently, the computational overhead introduced by the integration of K-means and the dynamic weight adjustment mechanism can increase the algorithm's computational complexity. This may limit the scalability of KCGWO to very large datasets or real-time clustering applications where computational resources or time are constrained. While empirical tests have demonstrated KCGWO's effectiveness on various datasets, its ability to generalize across all types of data distributions remains a concern. The algorithm's performance on datasets with complex structures, high dimensionality, or noise could vary, and its robustness in these scenarios has not been fully explored. The K-means component of KCGWO may not be inherently robust against noise and outliers, as K-means tends to be influenced by these factors. Consequently, KCGWO's performance could be degraded in datasets where noise and outliers are prevalent, affecting the quality of the clustering outcomes.

Addressing these limitations presents paths for future work, including the development of strategies to reduce dependence on initial clustering quality, adaptive parameter tuning mechanisms to mitigate sensitivity issues, and enhancements to computational efficiency. Additionally, further research could explore the incorporation of noise and outlier handling techniques to improve the robustness of KCGWO across diverse and challenging data environments.

Conclusions

This study advances data clustering and optimization through the development of an innovative approach, integrating the GWO with K-Means clustering, further augmented by a dynamic weight factor mechanism. This integration not only contributes to the theoretical framework of swarm intelligence methods but also demonstrates practical applicability in enhancing data clustering outcomes. The theoretical implications of this research are underscored by the systematic incorporation of a traditional clustering algorithm with a contemporary

Functions	Metrics	KCGWO	GWO	MFO	MVO	SSA
Emission	Min	4.671E+04	9.651E+04	1.314E+05	1.681E+05	1.268E+05
	Max	4.731E+04	1.038E+05	1.365E+05	2.280E+05	1.465E+05
	Mean	4.731E+04	1.038E+05	1.365E+05	2.280E+05	1.465E+05
	STD	7.841E+02	1.006E+04	7.692E+03	5.345E+04	1.442E+04
	Rank	1	2	4	5	3
HTRU2	Min	4.215E+05	1.377E+06	1.390E+06	1.600E+06	1.369E+06
	Max	4.461E+05	1.806E+06	1.400E+06	1.684E+06	1.395E+06
	Mean	4.461E+05	1.806E+06	1.400E+06	1.684E+06	1.395E+06
	STD	2.174E+04	3.448E+05	1.335E+04	7.098E+04	1.742E+04
	Rank	1	3	4	5	2
Wine	Min	383.0677	564.2516	574.3487	582.2759	551.9577
	Max	385.4381	588.2155	597.7833	691.7798	587.9902
	Mean	385.4381	588.2155	597.7833	691.7798	587.9902
	STD	3.817709	24.16611	23.8235	124.3448	32.72063
	Rank	1	3	4	5	2
Breast cancer	Min	7.984E+04	8.823E+04	1.411E+05	1.507E+05	1.403E+05
	Max	8.124E+04	1.040E+05	1.412E+05	1.605E+05	1.405E+05
	Mean	8.124E+04	1.040E+05	1.412E+05	1.605E+05	1.405E+05
	STD	1.753E+03	1.637E+04	1.440E+02	1.392E+04	2.119E+02
	Rank	1	2	4	5	3
Sonar	Min	15.77131	21.42128	20.34382	19.43976	19.00758
	Max	17.16814	24.3906	20.88503	22.09095	19.71072
	Mean	17.16814	24.3906	20.88503	22.09095	19.71072
	STD	1.60654	2.033542	0.431548	3.517527	0.417388
	Rank	1	5	4	3	2
WDBC	Min	1.563E+05	2.133E+05	2.684E+05	2.805E+05	2.647E+05
	Max	2.348E+05	2.741E+05	2.726E+05	3.037E+05	2.681E+05
	Mean	2.348E+05	2.741E+05	2.726E+05	3.037E+05	2.681E+05
	STD	1.129E+05	5.706E+04	3.622E+03	2.202E+04	3.404E+03
	Rank	1	2	4	5	3
Iris	Min	96.68885	192.0871	292.3649	294.3163	284.2368
	Max	99.58939	248.1072	325.407	335.4001	285.4755
	Mean	99.58939	248.1072	325.407	335.4001	285.4755
	STD	4.793102	39.83272	32.09701	38.01879	0.960675
	Rank	1	2	4	5	3
Ukraine Russia war	Min	5.565E+04	2.284E+05	2.188E+05	2.317E+05	2.146E+05
	Max	5.776E+04	2.915E+05	2.582E+05	2.812E+05	2.152E+05
	Mean	5.776E+04	2.915E+05	2.582E+05	2.812E+05	2.152E+05
	STD	1.810E+03	5.386E+04	2.336E+04	5.189E+04	8.246E+02
	Rank	1	4	3	5	2
Average rank		1.000	2.875	3.875	4.750	2.500

Table 8. Statistical results obtained for clustering problem by all selected algorithms. Significant values are in bold.

optimization technique, enriching the metaheuristic algorithm landscape. This methodology offers a new perspective on achieving a balance between exploration and exploitation in swarm-based algorithms, a pivotal factor in their efficiency and effectiveness for complex problem-solving. From a practical perspective, the introduction of the KCGWO represents a significant advancement towards more accurate and efficient data clustering

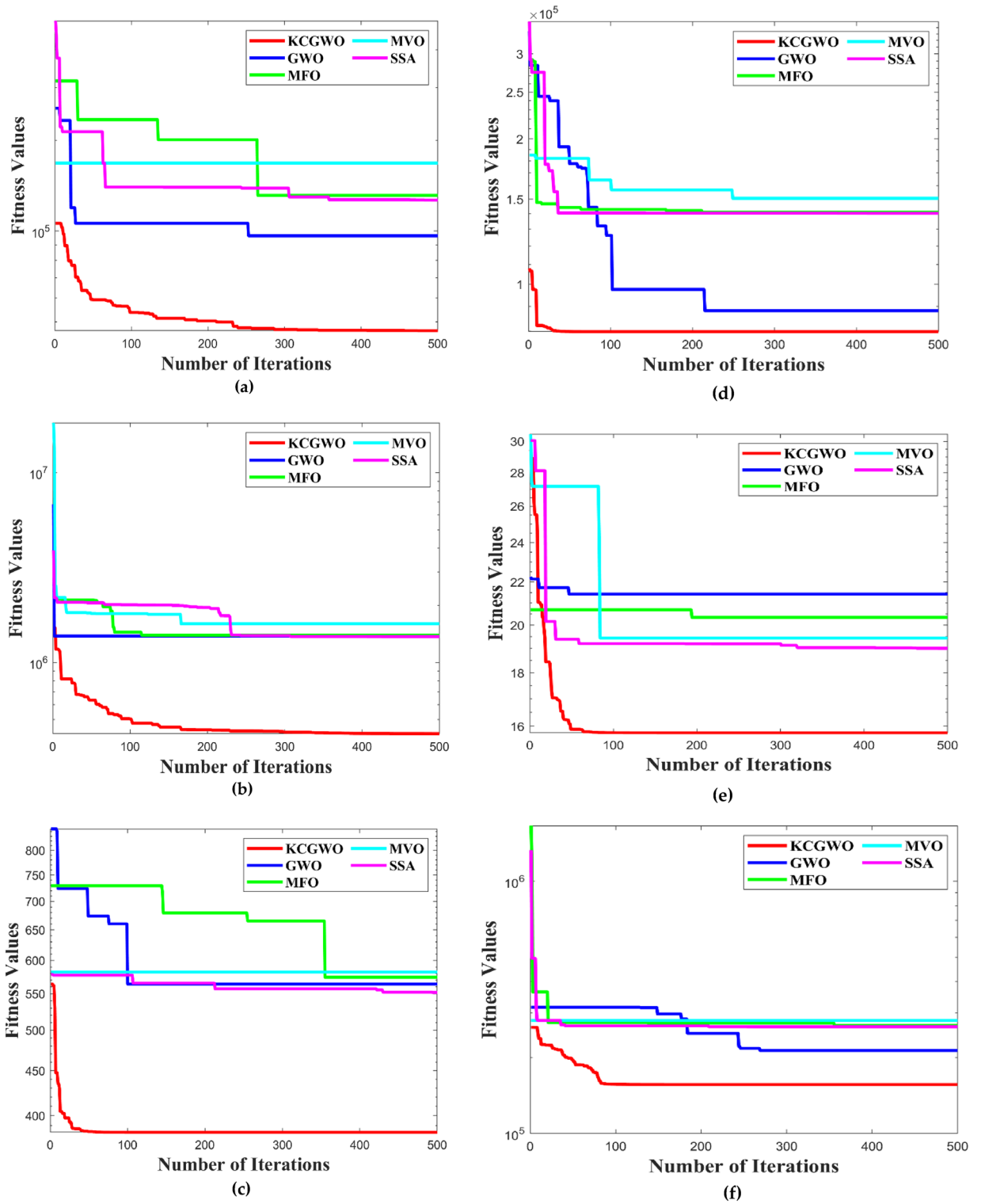


Figure 13. Convergence curves obtained by all algorithms: (a) Emission, (b) HTRU2, (c) Wine, (d) Breast cancer, (e) Sonar, (f) WDBC, (g) Iris, (h) 2022 Ukraine-Russia war.

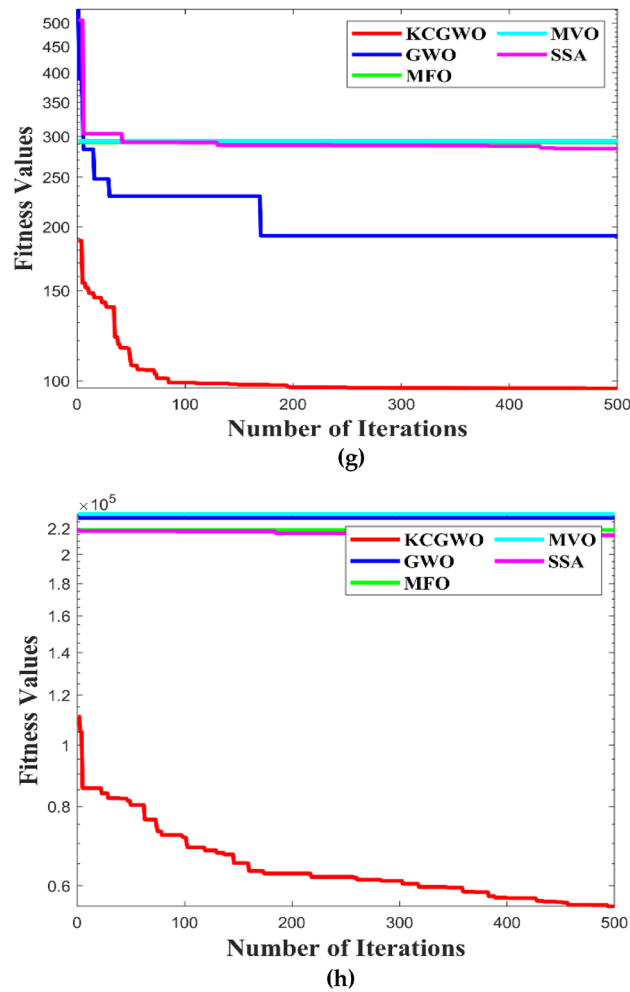


Figure 13. (continued)

solutions. By ingeniously adjusting swarm movements based on initial positions and integrating weight factors, the method exhibits enhanced diversity and an improved ability to escape local optima. These features are essential for applications demanding precise data segmentation, such as image recognition, market segmentation, and biological data analysis.

The contributions of this research extend beyond theoretical enhancement, offering tangible benefits to sectors reliant on data analytics. The improved exploration and exploitation dynamics of KCGWO result in faster convergence rates and superior clustering outcomes, rendering it an invaluable asset for processing large datasets with intricate structures. This is particularly pertinent in the Big Data context, where rapid and accurate clustering of large data sets can significantly influence decision-making processes and resource management.

In summary, the KCGWO algorithm marks a notable academic contribution to the discourse on optimization algorithms and facilitates its application across various practical scenarios. Its adaptability and efficiency herald new possibilities for addressing data-clustering challenges in diverse fields, signalling a new era of optimization solutions that are robust and responsive to the dynamic requirements of data analysis.

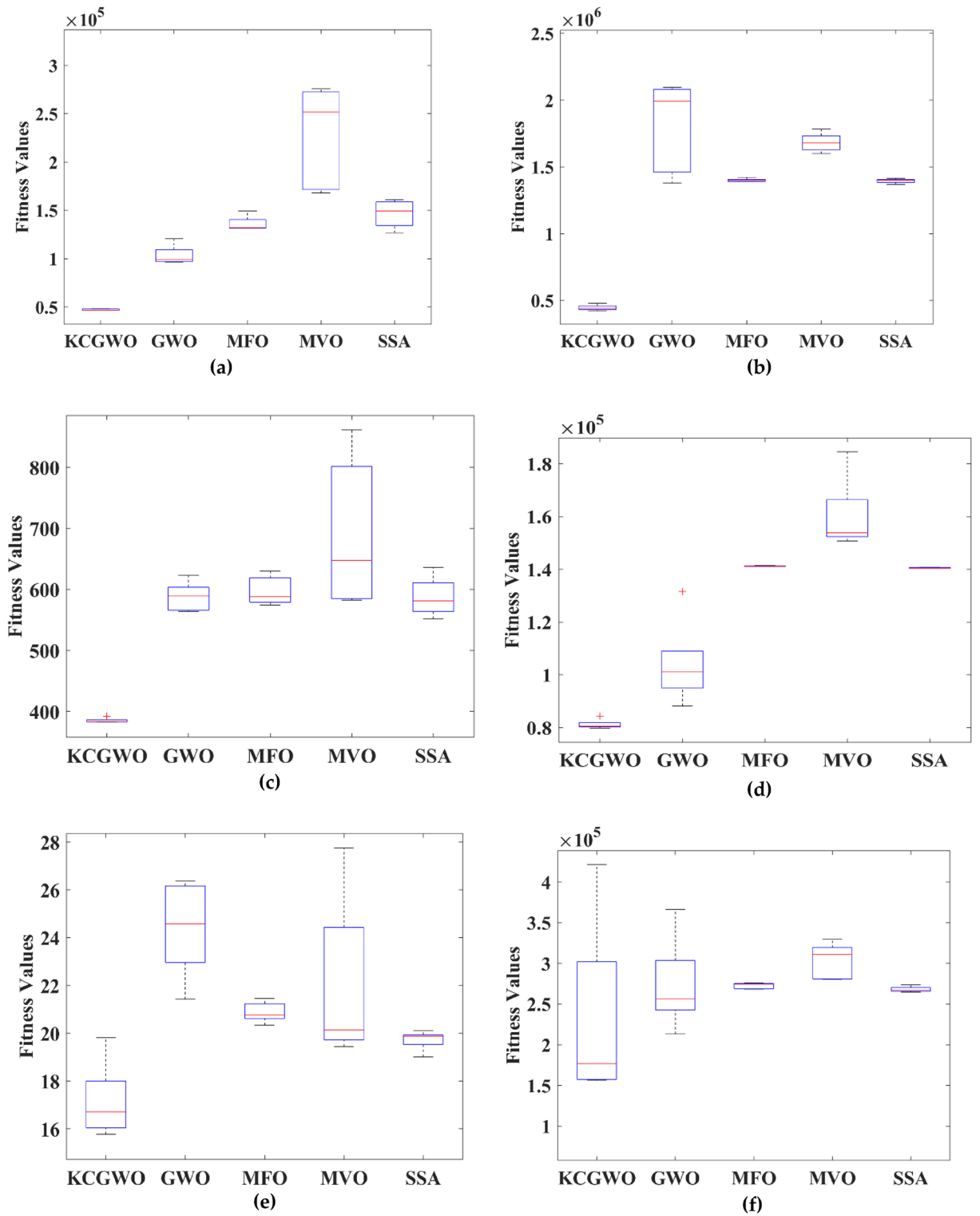


Figure 14. Boxplots obtained by all algorithms; (a) Emission, (b) HTRU2, (c) Wine, (d) Breast cancer, (e) Sonar, (f) WDBC, (g) Iris, (h) 2022 Ukraine-Russia war.

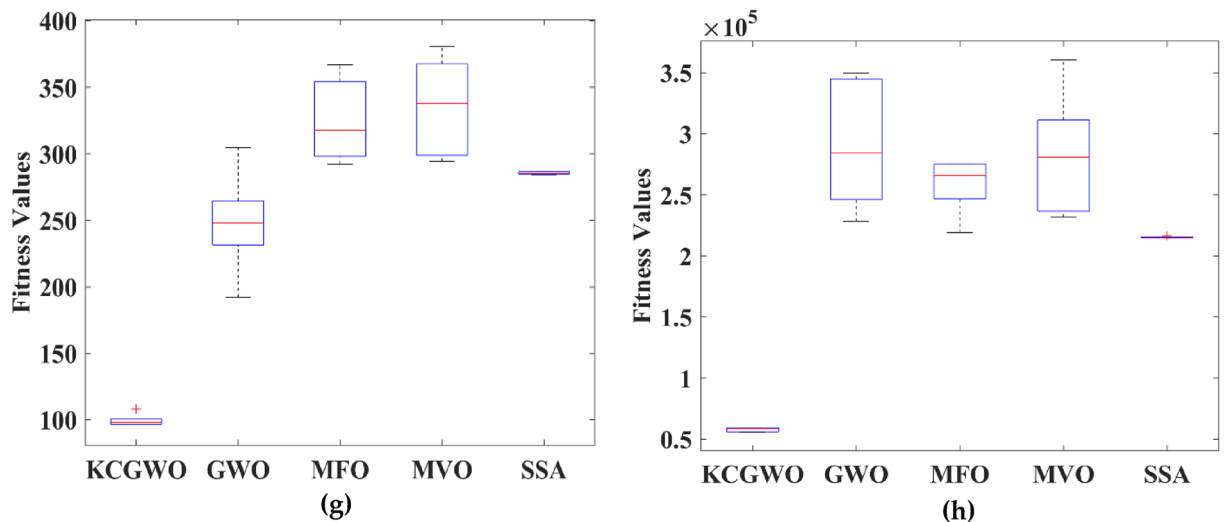


Figure 14. (continued)

Functions	KCGWO	GWO	MFO	MVO	SSA
Emission	2.145	2.047	2.569	2.602	2.554
HTRU2	3.419	3.217	3.987	4.001	3.984
Wine	1.265	1.193	1.547	1.698	1.654
Breast cancer	2.413	2.341	2.974	3.047	2.998
Sonar	2.987	2.801	3.014	3.142	3.131
WDBC	2.545	2.471	2.698	2.774	2.709
Iris	1.016	0.998	1.262	1.411	1.345
Ukraine Russia war	1.169	1.045	1.335	1.406	1.347
Average RT	2.120	2.014	2.423	2.510	2.465

Table 9. RT values of all algorithms for the clustering problem. Significant values are in bold.

Data availability

The dataset used in this paper is available in open source at <https://archive.ics.uci.edu/datasets?Task=Clustering>. All other data is included in the paper, and no additional data has been used in this study.

Received: 7 December 2023; Accepted: 26 February 2024

Published online: 05 March 2024

References

- Xiang, W.-L., Zhu, N., Ma, S.-F., Meng, X.-L. & An, M.-Q. A dynamic shuffled differential evolution algorithm for data clustering. *Neurocomputing* **158**, 144–154. <https://doi.org/10.1016/j.neucom.2015.01.058> (2015).
- Martinović, G.; Bajer, D. Data Clustering with Differential Evolution Incorporating Macromutations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8297 LNCS, 158–169. https://doi.org/10.1007/978-3-319-03753-0_15/COVER (2013).
- Nanda, S. J. & Panda, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm Evol. Comput.* **16**, 1–18. <https://doi.org/10.1016/j.swevo.2013.11.003> (2014).
- Kumar, Y. & Kaur, A. Variants of bat algorithm for solving partitional clustering problems. *Eng. Comput.* **38**, 1973–1999. <https://doi.org/10.1007/S00366-021-01345-3> (2022).
- Abualigah, L., Diabat, A. & Geem, Z. W. A comprehensive survey of the harmony search algorithm in clustering applications. *Appl. Sci.* **10**, 3827. <https://doi.org/10.3390/app10113827> (2020).
- Selvaraj, S. & Choi, E. Swarm intelligence algorithms in text document clustering with various benchmarks. *Sensors* **21**, 3196. <https://doi.org/10.3390/s21093196> (2021).
- Fujita, K. A clustering method for data in cylindrical coordinates. *Math. Probl. Eng.* <https://doi.org/10.1155/2017/3696850> (2017).
- Nguyen, H. H. Clustering categorical data using community detection techniques. *Comput. Intell. Neurosci.* <https://doi.org/10.1155/2017/8986360> (2017).
- Ma, J., Jiang, X. & Gong, M. Two-phase clustering algorithm with density exploring distance measure. *CAAI Trans. Intell. Technol.* **3**, 59–64. <https://doi.org/10.1049/TRIT.2018.0006> (2018).
- Hadikhani, P.; Lai, D.T.C.; Ong, W.H.; Nadimi-Shahraki, M.H. Improved Data Clustering Using Multi-Trial Vector-Based Differential Evolution with Gaussian Crossover. In *Proc. of the GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference; Association for Computing Machinery, Inc., pp. 487–490 (2022).*

11. Amiri, B., Fathian, M. & Maroosi, A. Application of shuffled frog-leaping algorithm on clustering. *Int. J. Adv. Manuf. Technol.* **45**, 199–209. <https://doi.org/10.1007/S00170-009-1958-2> (2009).
12. Ghany, K. K. A., AbdelAziz, A. M., Soliman, T. H. A. & Sewisy, A. A. E. M. A hybrid modified step whale optimization algorithm with Tabu search for data clustering. *J. King Saud Univ. Comput. Inf. Sci.* **34**, 832–839. <https://doi.org/10.1016/J.JKSUCI.2020.01.015> (2022).
13. Bouyer, A. & Hatamlou, A. An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Appl. Soft Comput.* **J. 67**, 172–182. <https://doi.org/10.1016/J.ASOC.2018.03.011> (2018).
14. Aljarah, I. & Ludwig, S. A. A new clustering approach based on glowworm swarm optimization. *IEEE Congress Evolut. Comput. CEC* **2013**(2013), 2642–2649. <https://doi.org/10.1109/CEC.2013.6557888> (2013).
15. Mai, X., Cheng, J. & Wang, S. Research on semi supervised K-means clustering algorithm in data mining. *Cluster Comput.* **22**, 3513–3520. <https://doi.org/10.1007/S10586-018-2199-7> (2019).
16. Jacques, J. & Preda, C. Functional data clustering: A survey. *Adv. Data Anal. Classif.* **8**, 231–255. <https://doi.org/10.1007/S11634-013-0158-Y> (2014).
17. Shirkorshidi, A.S.; Aghabozorgi, S.; Wah, T.Y.; Herawan, T. Big Data Clustering: A Review. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8583 LNCS, 707–720, https://doi.org/10.1007/978-3-319-09156-3_49 (2014).
18. Reynolds, A. P., Richards, G., de La Iglesia, B. & Rayward-Smith, V. J. Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *J. Math. Model. Algorithms* **5**, 475–504. <https://doi.org/10.1007/S10852-005-9022-1> (2006).
19. Hartigan, J. A. & Wong, M. A. Algorithm AS 136: A K-means clustering algorithm. *Appl. Stat.* **28**, 100. <https://doi.org/10.2307/2346830> (1979).
20. Jain, A. K., Murty, M. N. & Flynn, P. J. Data clustering: A review. *ACM Comput. Surv.* **31**, 264–323. <https://doi.org/10.1145/331499.331504> (1999).
21. Kao, Y. T., Zahara, E. & Kao, I. W. A hybridized approach to data clustering. *Expert Syst. Appl.* **34**, 1754–1762. <https://doi.org/10.1016/J.ESWA.2007.01.028> (2008).
22. Nasiri, J. & Khiyabani, F. M. A whale optimization algorithm (WOA) approach for clustering. *Cogent Math. Stat.* **5**, 1483565. <https://doi.org/10.1080/25742558.2018.1483565> (2018).
23. Besharatnia, F., Talebpour, A. & Aliakbary, S. An improved grey wolves optimization algorithm for dynamic community detection and data clustering. *Appl. Artif. Intell.* <https://doi.org/10.1080/08839514.2021.2012000> (2021).
24. Singh, T. A novel data clustering approach based on whale optimization algorithm. *Expert Syst.* **38**, e12657. <https://doi.org/10.1111/EXSY.12657> (2021).
25. Isimeto, R., Yinka-Banjo, C., Uwadia, C.O., Alienyi, D.C. An enhanced clustering analysis based on glowworm swarm optimization. In *IEEE 4th International Conference on Soft Computing and Machine Intelligence, ISCM 2017, 2018-January*, 42–49, <https://doi.org/10.1109/ISCM.2017.8279595> (2018).
26. Zhang, L. *et al.* Overlapping community-based particle swarm optimization algorithm for influence maximization in social networks. *CAAI Trans. Intell. Technol.* <https://doi.org/10.1049/CIT2.12158> (2023).
27. Kumar, Y. & Sahoo, G. Hybridization of magnetic charge system search and particle swarm optimization for efficient data clustering using neighborhood search strategy. *Soft Comput.* **19**, 3621–3645. <https://doi.org/10.1007/s00500-015-1719-0> (2015).
28. Cura, T. A particle swarm optimization approach to clustering. *Expert Syst. Appl.* **39**, 1582–1588. <https://doi.org/10.1016/j.eswa.2011.07.123> (2012).
29. Chang, D. X., Zhang, X. D. & Zheng, C. W. A genetic algorithm with gene rearrangement for K-means clustering. *Pattern Recogn.* **42**, 1210–1222. <https://doi.org/10.1016/j.patcog.2008.11.006> (2009).
30. Hatamlou, A.; Abdullah, S.; Nezamabadi-Pour, H. Application of Gravitational Search Algorithm on Data Clustering. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6954 LNAI, 337–346, https://doi.org/10.1007/978-3-642-24425-4_44/COVER (2011).
31. Lin, B.Y.; Kuo, C.N.; Lin, Y. Da A Clustering-Based Gauss Chaotic Mapping Particle Swarm Optimization for Auto Labeling in Human Activity Recognition. In *Proc. 2021 International Conference on Technologies and Applications of Artificial Intelligence, TAAI 2021*, 238–242, <https://doi.org/10.1109/TAAI54685.2021.00052> (2021).
32. Zou, W., Zhu, Y., Chen, H. & Sui, X. A clustering approach using cooperative artificial bee colony algorithm. *Discrete Dyn. Nat. Soc.* <https://doi.org/10.1155/2010/459796> (2010).
33. Jinyin, C., Huihao, H., Jungan, C., Shanqing, Y. & Zhaoxia, S. Fast Density clustering algorithm for numerical data and categorical data. *Math. Probl. Eng.* <https://doi.org/10.1155/2017/6393652> (2017).
34. Lv, L. RFID data analysis and evaluation based on big data and data clustering. *Comput. Intell. Neurosci.* <https://doi.org/10.1155/2022/3432688> (2022).
35. Dorigo, M., Maniezzo, V. & Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst Man Cybern. B (Cybernetics)* **26**, 29–41. <https://doi.org/10.1109/3477.484436> (1996).
36. Johari, N. F., Zain, A. M., Noorfa, M. H. & Udin, A. Firefly algorithm for optimization problem. *Appl. Mech. Mater.* **421**, 512–517. <https://doi.org/10.4028/WWW.SCIENTIFIC.NET/AMM.421.512> (2013).
37. Tadeipalli, Y. *et al.* Content-based image retrieval using Gaussian-Hermite moments and firefly and grey wolf optimization. *CAAI Trans. Intell. Technol.* **6**, 135–146. <https://doi.org/10.1049/CIT2.12040> (2021).
38. Mohammadzadeh, H. & Gharehchopogh, F. S. A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study email spam detection. *Comput. Intell.* **37**, 176–209. <https://doi.org/10.1111/COIN.12397> (2021).
39. Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H. & Mirjalili, S. Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowl. Inf. Syst.* **62**, 507–539. <https://doi.org/10.1007/s10115-019-01358-x> (2020).
40. Premkumar, M., Jangir, P., Santhosh Kumar, B., Alqudah, A. & Sooppy Nisar, M. K. Multi-objective grey wolf optimization algorithm for solving real-world BLDC motor design problem. *Comput. Mater. Continua* **70**, 2435–2452. <https://doi.org/10.32604/CMC.2022.016488> (2022).
41. Premkumar, M., Sowmya, R., Umashankar, S. & Jangir, P. Extraction of uncertain parameters of single-diode photovoltaic module using hybrid particle swarm optimization and grey wolf optimization algorithm. *Mater. Today Proc.* **46**, 5315–5321. <https://doi.org/10.1016/J.MATPR.2020.08.784> (2021).
42. Xavier, F. J., Pradeep, A., Premkumar, M. & Kumar, C. Orthogonal learning-based gray wolf optimizer for identifying the uncertain parameters of various photovoltaic models. *Optik (Stuttg)* **247**, 167973. <https://doi.org/10.1016/J.IJLEO.2021.167973> (2021).
43. Venkata Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **7**, 19–34. <https://doi.org/10.5267/j.ijiec.2015.8.004> (2016).
44. Rao, R. V. & Patel, V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *Int. J. Ind. Eng. Comput.* **3**, 535–560. <https://doi.org/10.5267/J.IJIEC.2012.03.007> (2012).
45. Rao, R. V. Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *Int. J. Ind. Eng. Comput.* **11**, 107–130. <https://doi.org/10.5267/j.ijiec.2019.6.002> (2020).
46. Premkumar, M.; Sowmya, R.; Jangir, P.; Siva Kumar, J.S.V. A New and Reliable Objective Functions for Extracting the Unknown Parameters of Solar Photovoltaic Cell Using Political Optimizer Algorithm. In *Proc. of the 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020* (2020).

47. Premkumar, M. & Sumithira, R. Humpback whale assisted hybrid maximum power point tracking algorithm for partially shaded solar photovoltaic systems. *J. Power Electron.* **18**, 1805–1818. <https://doi.org/10.6113/JPE.2018.18.6.1805> (2018).
48. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **89**, 228–249. <https://doi.org/10.1016/j.knsys.2015.07.006> (2015).
49. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513. <https://doi.org/10.1007/s00521-015-1870-7> (2016).
50. Premkumar, M., Kumar, C., Sowmya, R. & Pradeep, J. A novel Salp swarm assisted hybrid maximum power point tracking algorithm for the solar photovoltaic power generation systems. *Automatika* <https://doi.org/10.1080/00051144.2020.1834062> (2021).
51. Premkumar, M., Ibrahim, A. M., Kumar, R. M. & Sowmya, R. Analysis and simulation of bio-inspired intelligent Salp swarm MPPT method for the PV systems under partial shaded conditions. *Int. J. Comput. Digit. Syst.* **8**, 2210–3142. <https://doi.org/10.12785/ijcds/080506> (2019).
52. Dhiman, G. & Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **114**, 48–70. <https://doi.org/10.1016/j.advengsoft.2017.05.014> (2017).
53. Arora, S. & Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* <https://doi.org/10.1007/s00500-018-3102-4> (2019).
54. Yazdani, M. & Jolai, F. Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm. *J. Comput. Des. Eng.* <https://doi.org/10.1016/j.jcde.2015.06.003> (2016).
55. Sudhakar Babu, T., Prasanth Ram, J., Sangeetha, K., Laudani, A. & Rajasekar, N. Parameter extraction of two diode solar PV model using fireworks algorithm. *Solar Energy* **140**, 265–276. <https://doi.org/10.1016/j.solener.2016.10.044> (2016).
56. Yang, X.-S.; Deb, S. Cuckoo Search via Levy Flights. In *Proc. of the World Congress on Nature & Biologically Inspired Computing (NaBIC); IEEE: Coimbatore, India*, pp. 210–214 (2010).
57. Yang, X. S. A new metaheuristic bat-inspired algorithm. *Stud. Comput. Intell.* **284**, 65–74. https://doi.org/10.1007/978-3-642-12538-6_6 (2010).
58. Prajapati, V.K.; Jain, M.; Chouhan, L. Tabu Search Algorithm (TSA): A Comprehensive Survey. In *Proc. of 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things, ICETCE 2020*, 222–229. <https://doi.org/10.1109/ICETCE48199.2020.9091743> (2020).
59. Geem, Z. W., Kim, J. H. & Loganathan, G. V. A new heuristic optimization algorithm: Harmony search. *Simulation* **76**, 60–68. <https://doi.org/10.1177/003754970107600201> (2001).
60. Sowmya, R., Premkumar, M. & Jangir, P. Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. *Eng. Appl. Artif. Intell.* **128**, 107532. <https://doi.org/10.1016/j.engappai.2023.107532> (2024).
61. Kailasam, J. K., Nalliah, R., Muthusamy, S. N. & Manoharan, P. MLBRSA: Multi-learning-based reptile search algorithm for global optimization and software requirement prioritization problems. *Biomimetics* **8**, 615. <https://doi.org/10.3390/BIOMIMETICS8080615> (2023).
62. Premkumar, M. *et al.* MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting. *IEEE Access* **9**, 3229–3248. <https://doi.org/10.1109/ACCESS.2020.3047936> (2021).
63. Kumar, C., Raj, T. D., Premkumar, M. & Raj, T. D. A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters. *Optik (Stuttg)* **223**, 165277. <https://doi.org/10.1016/j.ijleo.2020.165277> (2020).
64. Heidari, A. A. *et al.* Harris hawks optimization: Algorithm and applications. *Future Gen. Comput. Syst.* **97**, 849–872. <https://doi.org/10.1016/j.future.2019.02.028> (2019).
65. Khishe, M. & Mosavi, M. R. Chimp optimization algorithm. *Expert Syst. Appl.* **149**, 113338. <https://doi.org/10.1016/j.eswa.2020.113338> (2020).
66. Abdollahzadeh, B., Soleimani Gharehchopogh, F. & Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *In. J. Intell. Syst.* **36**, 5887–5958. <https://doi.org/10.1002/INT.22535> (2021).
67. Irudayaraj, A. X. R. *et al.* Distributed intelligence for consensus-based frequency control of multi-microgrid network with energy storage system. *J. Energy Stor.* **73**, 109183. <https://doi.org/10.1016/j.est.2023.109183> (2023).
68. Jangir, P., Buch, H., Mirjalili, S. & Manoharan, P. MOMPA: Multi-objective marine predator algorithm for solving multi-objective optimization problems. *Evolut. Intell.* **2021**, 1–27. <https://doi.org/10.1007/S12065-021-00649-Z> (2021).
69. Sowmya, R. & Sankaranarayanan, V. Optimal vehicle-to-grid and grid-to-vehicle scheduling strategy with uncertainty management using improved marine predator algorithm. *Comput. Electr. Eng.* **100**, 107949. <https://doi.org/10.1016/j.compeleceng.2022.107949> (2022).
70. Wang, X., Liu, Q. & Zhang, L. An adaptive sand cat swarm algorithm based on cauchy mutation and optimal neighborhood disturbance strategy. *Biomimetics* **8**, 191. <https://doi.org/10.3390/BIOMIMETICS8020191> (2023).
71. Premkumar, M. *et al.* Multi-objective equilibrium optimizer: Framework and development for solving multi-objective optimization problems. *J. Comput. Des. Eng.* **9**, 24–50. <https://doi.org/10.1093/JCDE/QWAB065> (2022).
72. Houssein, E. H. *et al.* An efficient multi-thresholding based COVID-19 CT images segmentation approach using an improved equilibrium optimizer. *Biomed. Signal Process. Control* **73**, 103401. <https://doi.org/10.1016/j.bspc.2021.103401> (2022).
73. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gen. Comput. Syst.* <https://doi.org/10.1016/j.future.2019.07.015> (2019).
74. Ravichandran, S., Manoharan, P., Jangir, P. & Selvarajan, S. Resistance-capacitance optimizer: A physics-inspired population-based algorithm for numerical and industrial engineering computation problems. *Sci. Rep.* **13**, 1–40. <https://doi.org/10.1038/s41598-023-42969-3> (2023).
75. Premkumar, M. *et al.* A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: Diversity analysis and validations. *IEEE Access* **9**, 84263–84295. <https://doi.org/10.1109/ACCESS.2021.3085529> (2021).
76. Zamani, H., Nadimi-Shahraki, M. H. & Gandomi, A. H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **104**, 104314. <https://doi.org/10.1016/j.engappai.2021.104314> (2021).
77. Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S. & Faris, H. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Appl. Soft Comput.* **97**, 106761. <https://doi.org/10.1016/j.asoc.2020.106761> (2020).
78. Cao, L., Chen, H., Chen, Y., Yue, Y. & Zhang, X. Bio-inspired swarm intelligence optimization algorithm-aided hybrid TDOA/AOA-based localization. *Biomimetics* **8**, 186. <https://doi.org/10.3390/BIOMIMETICS8020186> (2023).
79. Zamani, H., Nadimi-Shahraki, M. H. & Gandomi, A. H. Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Comput. Methods Appl. Mech. Eng.* **392**, 114616. <https://doi.org/10.1016/j.cma.2022.114616> (2022).
80. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **93**, 657–683. <https://doi.org/10.1016/j.apm.2020.12.021> (2021).
81. Trojovský, P. & Dehghani, M. Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* **8**, 149. <https://doi.org/10.3390/BIOMIMETICS8020149> (2023).

82. Hassan, M. H., Kamel, S., Selim, A., Khurshaid, T. & Domínguez-garcía, J. L. A modified Rao-2 algorithm for optimal power flow incorporating renewable energy sources. *Mathematics* **9**, 1532. <https://doi.org/10.3390/MATH9131532> (2021).
83. Peraza-Vázquez, H. *et al.* A bio-inspired method for mathematical optimization inspired by *Arachnida salticidae*. *Mathematics* **10**, 102. <https://doi.org/10.3390/MATH10010102> (2021).
84. Shaban, H. *et al.* Identification of parameters in photovoltaic models through a Runge Kutta optimizer. *Mathematics* **9**, 2313. <https://doi.org/10.3390/MATH9182313> (2021).
85. Zhou, Y., Wu, H., Luo, Q. & Abdel-Baset, M. Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowl. Based Syst.* **163**, 546–557. <https://doi.org/10.1016/j.knsys.2018.09.013> (2019).
86. Singh, T. *et al.* Data clustering using moth-flame optimization algorithm. *Sensors* **21**, 4086. <https://doi.org/10.3390/S21124086> (2021).
87. Abualigah, L. & Almotairi, K. H. Dynamic evolutionary data and text document clustering approach using improved Aquila optimizer based arithmetic optimization algorithm and differential evolution. *Neural Comput. Appl.* **2022**, 1–33. <https://doi.org/10.1007/S00521-022-07571-0> (2022).
88. Bacanin, N. *et al.* Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* **9**, 2705. <https://doi.org/10.3390/MATH9212705> (2021).
89. Malakar, S., Ghosh, M., Bhowmik, S., Sarkar, R. & Nasipuri, M. A GA based hierarchical feature selection approach for handwritten word recognition. *Neural Comput. Appl.* **32**, 2533–2552. <https://doi.org/10.1007/S00521-018-3937-8/TABLES/12> (2020).
90. Pervaiz, S., Ul-Qayyum, Z., Bangyal, W. H., Gao, L. & Ahmad, J. A systematic literature review on particle swarm optimization techniques for medical diseases detection. *Comput. Math. Methods Med.* <https://doi.org/10.1155/2021/5990999> (2021).
91. Bacanin, N. *et al.* Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. *Sci. Rep.* **12**, 1–20. <https://doi.org/10.1038/s41598-022-09744-2> (2022).
92. Zivkovic, M. *et al.* Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images. *Electronics* **11**, 3798. <https://doi.org/10.3390/ELECTRONICS11223798> (2022).
93. Bangyal, W.H., Shakir, R., Rehman, N.U., Ashraf, A., Ahmad, J. An improved seagull algorithm for numerical optimization problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13968 LNCS, 297–308, doi:https://doi.org/10.1007/978-3-031-36622-2_24/COVER (2023).
94. Bangyal, W. H. *et al.* An improved particle swarm optimization algorithm for data classification. *Appl. Sci.* **13**, 283. <https://doi.org/10.3390/APPI13010283> (2022).
95. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007> (2014).
96. Ahmadi, R., Ekbatanifard, G. & Bayat, P. A modified grey wolf optimizer based data clustering algorithm. *Appl. Artif. Intell.* **35**, 63–79. <https://doi.org/10.1080/08839514.2020.1842109> (2020).
97. Faris, H., Aljarah, I., Al-Betar, M. A. & Mirjalili, S. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **30**, 413–435. <https://doi.org/10.1007/S00521-017-3272-5> (2017).
98. Rezaei, F. *et al.* An enhanced grey wolf optimizer with a velocity-aided global search mechanism. *Mathematics* **10**, 351. <https://doi.org/10.3390/MATH10030351> (2022).
99. Rasappan, P., Premkumar, M., Sinha, G. & Chandrasekaran, K. Transforming sentiment analysis for e-commerce product reviews: Hybrid deep learning model with an innovative term weighting and feature selection. *Inf. Process. Manag.* **61**, 103654. <https://doi.org/10.1016/J.IPM.2024.103654> (2024).
100. Kumar, Y. & Sahoo, G. A two-step artificial bee colony algorithm for clustering. *Neural Comput. Appl.* **28**, 537–551. <https://doi.org/10.1007/s00521-015-2095-5> (2015).
101. Kushwaha, N., Pant, M., Kant, S. & Jain, V. K. Magnetic optimization algorithm for data clustering. *Pattern Recogn. Lett.* **115**, 59–65. <https://doi.org/10.1016/j.patrec.2017.10.031> (2018).
102. Singh, H. *et al.* An enhanced whale optimization algorithm for clustering. *Multimed. Tools Appl.* **2022**, 1–20. <https://doi.org/10.1007/S11042-022-13453-3> (2022).
103. Yang, X.; Luo, Q.; Zhang, J.; Wu, X.; Zhou, Y. Moth Swarm Algorithm for Clustering Analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10363 LNAI, 503–514, https://doi.org/10.1007/978-3-319-63315-2_44/COVER (2017).
104. Mohammed, H. M., Abdul, Z. K., Rashid, T. A., Alsadoon, A. & Bacanin, N. A new K-means grey wolf algorithm for engineering problems. *World J. Eng.* **18**, 630–638. <https://doi.org/10.1108/WJE-10-2020-0527/FULL/XML> (2021).
105. Gao, Z. M. & Zhao, J. An improved grey wolf optimization algorithm with variable weights. *Comput. Intell. Neurosci.* <https://doi.org/10.1155/2019/2981282> (2019).
106. Dua, D. and G. UCI machine learning repository: Data sets. <https://archive.ics.uci.edu/ml/datasets.php> (Accessed 16 March 2023).

Acknowledgements

The authors would like to acknowledge the blind reviewers for their constructive comments to improve the paper.

Author contributions

Conceptualization, P.M.; data curation, P.M., G.S. and M.D.R.; formal analysis, P.M., G.S., M.D.R., S.S., B.S.C., L.A., and S.R.; funding acquisition, Premkumar M; investigation, P.M., G.S. and M.D.R. and S.S.; methodology, P.M.; project administration, L.A. and S.R.; resources, M.D.R., S.S. and L.A.; software, P.M., S.R., and G.S.; validation, P.M., G.S. and M.D.R.; visualization, P.M., S.S. and B.S.C.; writing—original draft, P.M., S.R. and G.S.; writing—review & editing, M.D.R., S.S., B.S.C., L.A. and S.R.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.P. or B.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024, corrected publication 2024