




An Improved Water Flow Optimizer for Data Clustering

Prateek Thakral¹ · Yugal Kumar^{1,2} 

Received: 5 October 2023 / Accepted: 8 June 2024

© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd. 2024

Abstract

Recently, various meta-heuristic algorithms have been considered to allocate the data into different clusters based on similar information. These algorithms have obtained state of the art clustering results compared to traditional algorithms and proven their capability in the field of data clustering. This work presents an improved version of the water flow optimizer, called the IWFO algorithm for effective cluster analysis. The proposed IWFO algorithm handles the performance issues associated with the water flow optimizer algorithm such as random initialization, unbalanced search mechanism and local optima. The random initialization issues are handled through the gaussian map that can generate the initial population systematically. The search mechanism of the WFO algorithm is enhanced using the combination of non-linear functions and the previous best solution. The local optima issue is alleviated by using a neighbourhood search mechanism. The efficacy of the proposed IWFO algorithm is evaluated using benchmark clustering datasets and results are compared with popular clustering algorithms. The simulation results are assessed using intra-cluster distance (intra), standard deviation (SD), rank, accuracy rate (AR) and detection rate (DR) parameters. Some statistical tests are also performed to validate the efficiency of the proposed IWFO algorithm. The proposed IWFO algorithm improves the clustering results (average accuracy rate of more than 7%) compared to the original WFO.

Keywords Clustering · Cluster analysis · Meta heuristics · Water flow optimizer

Introduction

Clustering is a well-known data analysis method that can be used to arrange the data into different clusters. Similar data are placed in the same cluster, while dissimilar data are put in another cluster. The dissimilarity between data is calculated using the distance function [1]. In literature, clustering can be used in a variety of fields including text mining, social network analysis, data exploration, medical science, finance, and multimedia data [2, 3]. Additionally, the subcategories of clustering are (i) hard clustering and (ii) soft clustering. In hard clustering, the data can be assigned to a single cluster, while, in soft clustering, data

can belong to several clusters depending on probability function value [4]. The main issue associated with clustering is the quality of partition. Based on the optimal partition (i.e., cluster centroids), a dataset is divided into numerous clusters, and the clustering techniques are utilized to calculate optimal centroids for obtaining optimal partitioning (clusters). These partitions are validated based on internal, external, and relative cluster validation measures [5]. These validation methods have defined the quality of the clustering algorithm. The internal validation is based on cluster creation, such as compactness, separation, and connectivity. The closeness between the data within a cluster can be used to define compactness. If a cluster displays the bare minimum of compactness, it is sufficient. The distance between two or more clusters is used to calculate separation, and it can be on the extreme side. The identical cluster data is used to describe the connectivity. External validation is described by comparing the clustering result against the class labels that are mentioned in the dataset. It includes various performance indicators like purity, rand index, entropy, etc. The relative validation evaluates the clustering structure through

✉ Yugal Kumar
yugalkumar.14@gmail.com

Prateek Thakral
18.prateek@gmail.com

¹ Jaypee University of Information Technology, Solan,
Himachal Pradesh, India

² School of Technology Management and Engineering,
NMIMS, Chandigarh Campus, Chandigarh, India

various user-defined parameter values that are provided by the algorithms [6].

In literature, different clustering algorithms based on diverse methodologies have been presented in the literature for addressing clustering problems and also considered the various viewpoints for solving these problems [7–12]. Further, the clustering techniques are divided into five categories such as partitional, hierarchical, grid, density, and model-based [13]. Additionally, each technique has several advantages and disadvantages. Recently, the research community has focused on grouping uncertain and high-dimensional data [14, 15]. Despite this, current developments in clustering can be described as fuzzy, evolutionary, meta-heuristic, and multimedia clustering [16, 17]. A few clustering studies also presented novel distance functions to improve the results of clustering. Some studies also focus on validation metrics to evaluate the effectiveness of clustering algorithms [11]. However, no one approach can more effectively handle the clustering problem with a wide range of datasets. Every algorithm has several advantages and disadvantages. According to a thorough literature review, it is identified that partitional clustering can be used more frequently than other clustering techniques, like the hierarchical, grid, and model-based clustering techniques due to being less time-consuming [18]. Partitional clustering determines a distinct set of clusters and allocates the data to a particular cluster based on distance measures. Additionally, prior knowledge of clusters (K) is the prerequisite for this clustering. K -Means is one of the popular partitional clustering algorithms [19], and it is important to anticipate the number of clusters. In turn, the clustering process can become more extensive and even more difficult to compute the optimal partitioning for the given dataset. Further, it is noticed that clustering results do not converge on the global optima [20]. Therefore, to solve the aforementioned problem, either cluster information should be provided beforehand, or the number of clusters should be calculated automatically to achieve optimal partitioning. Some meta-heuristic algorithms have been reported for handling partitional clustering in the literature. A few of these algorithms are summarized as PSO [21, 22], MOA [23, 24], CSSA [25], BH [26], BA [27, 28], ABC [29, 30], ACOA [31], BB-BC [32], BBO [33], etc. These algorithms have balanced search capabilities and produce prominent clustering results. However, these algorithms have some other drawbacks such as population diversity, trade-off, convergence rate, and sometimes trapped in local optima [34]. The aforementioned weaknesses of meta-heuristic algorithms can be eliminated with the assistance of additional meta-heuristic algorithms. To achieve superior clustering results, the weak point of one meta-heuristic algorithm is substituted by the strong point of another meta-heuristic algorithm. For example, the slow convergence rate

of the PSO algorithm is handled by hybridizing the PSO with the k -harmonic mean [35]. Similarly, chaotic maps are incorporated into PSO to accelerate convergence speed [36]. To increase the performance of k -means and limit the effect of initial centroids on final clustering results, the k -mean and PSO algorithms are combined [37]. ABC trade-off problem is handled via a self-adaptive system [37, 38]. The BB-BC local optima problem is solved by integrating chaotic maps [39]. Based on approximation functions, these methods also provide near-to-optimal solutions for clustering issues. However, the No Free Lunch theorem [81] states that no single clustering approach can be used to solve all clustering problems as well as applicable to all datasets. Hence, there is a scope to develop a new clustering algorithm that can generate a more optimal solution for clustering problems and is also applicable to a wide range of datasets. As a result, a new algorithm for obtaining optimal solutions and solving large-scale clustering problems can be devised. Recently, a new meta-heuristic algorithm, named water flow optimizer (WFO) has been presented to handle a variety of constrained and unconstrained optimization problems [40, 41]. The hydraulic processes of water particles inspired this method, which describes the flow of water from highland to lowland. Further, the laminar and turbulent flows are taken into account to devise the stochastic search operators for the optimization process. From an optimization viewpoint, the water flow can be classified into two types such as either to maximize or minimize an objective function that can be designed to solve the problems. Second, an iterative process can be used to find the best solution and convergence behaviour of an algorithm. Hence, this work aims to examine the efficacy of the WFO algorithm for solving clustering problems. However, before implementing the WFO in the clustering field, several modifications are integrated into the WFO algorithm to make it more effective and generate optimal clustering results.

Motivation and Objectives of the Work

This research work aims to examine the efficacy of the WFO algorithm for handling clustering problems. Clustering can be described as an optimization problem with constraints and it determines the groups of similar data objects. The data in groups are allocated using a distance function and the goodness of the groups is assessed using the centroids. Hence, the main objective of the WFO algorithm is to compute optimal centroids to group the data objects into respective clusters. However, some improvements are proposed and integrated into WFO before its implementation. The reason for these improvements is to overcome the issues associated with the WFO algorithm. In the literature, it is mentioned that WFO has better optimization capability, but this algorithm also suffers from several shortcomings [42,

Table 1 Summarizes the works reported on clustering using meta-heuristic algorithms

References	Meta-heuristic(s)	Datasets	Neighboring search	Objective function	Clustering type	Year
[44]	Capuchin search algorithm	Hill-Valley, dermatology, iris, wine, balance, <i>E. coli</i> , TAE, seeds, CMC, and Hungarian	Chameleon swarm algorithm	Intra cluster distance	Hard clustering	2024
[45]	Genetic algorithm, particle swarm optimization, gradient evolution	Wine, WBC, Tae, vehicle, pima, iris, breast, liver, banknote, audit, fertility, seed, Haberman, and vertebral	*	Global cluster compactness and fuzzy separation	Multi-objective hard clustering	2024
[46]	Grey wolf optimizer	Iris, wine, breast cancer, sonar, WDBC	*	Euclidean distance	Hard clustering	2024
[47]	Electrical search algorithm	Iris, wine, seeds, hepatitis C virus	*	Euclidean distance	Hard clustering	2023
[48]	Interactive Autodidactic school (IAS) algorithm	Iris, glass, blood, seeds, speech, breast, CMC, dermatology, wine, vowel, hepatitis, balance scale, ORL, Libras, lung cancer	Chaotic maps	Euclidean distance and sum of squared error	Hard clustering	2023
[49]	Leaders and followers optimization, differential evolution	Glass, iris, wine, yeast, aggregation, compound, path-based, spiral, flame, Jain, R15, D31	*	Euclidean distance	Hard clustering	2023
[50]	Affinity propagation, improved equilibrium optimizer	Colon, SRBCT, central nervous system, ALL-AML-2, and ALL-AML-4,	Crisscross strategy	Intra cluster similarity	Automatic clustering	2023
[51]	Chimp optimization algorithm, generalized Normal distribution algorithm, opposition-based learning	Iris, wine, cancer, blood, CMC, path-based, flame, aggregation	*	Euclidean distance	Hard clustering	2023
[52]	Enhanced Whale Optimization algorithm, Tabu search	Iris, wine, cancer, CMC, LR, glass, ISOLET, thyroid	Tabu search-based neighborhood strategy	Sum of squared error	Hard clustering	2023
[53]	Firefly algorithm	Obesity, segment, hepatitis, vehicle, <i>E. coli</i> , glass, CMC and mammographic	Variable neighbourhood search	Euclidean distance	Hard clustering	2022
[54]	Grey wolf optimizer algorithm, label propagation algorithm	Heart, <i>E. coli</i> , horse, cancer, balance, dermatology, credit, cancer-int, diabetes	Improved local search technique	Homogeneity criterion	Community detection and hard clustering	2022
[55]	Enhanced cat swarm optimization	Iris, wine, glass, CMC, LD, cancer, vowel, thyroid	Step division-based neighbourhood search	Sum of squared error	Hard clustering	2022
[56]	Electromagnetic clustering algorithm	Gas, human activity recognition, vowel, thyroid, iris, IONO, crude oil, CMC	*	Intra cluster distance	Hard clustering	2022
[57]	Improved Particle swarm optimization	Iris, wine, breast cancer, car evaluation, Statlog, yeast	Multi starter search	Sum of squared error	Hard clustering	2022

Table 1 (continued)

References	Meta-heuristic(s)	Datasets	Neighbouring search	Objective function	Clustering type	Year
[58]	Dynamic parameters enabled harmony search optimization algorithm	Artificial Data1, Data2, Data3, iris, glass	*	Euclidean distance	Automatic clustering	2022
[59]	Sine-cosine algorithm, k-prototypes algorithm	CMC, Statlog, thyroid disease, teaching assistant evaluation, credit approval, flags, dermatology, census income, heart disease, CVR	*	Euclidean distance, hamming distance	Hard clustering	2022
[60]	Improved bat algorithm	Iris, glass, wine, ionosphere, control, vowel, balance, crude oil, CMC, LD, WBC, thyroid	Q-learning-based neighborhood search mechanism	Sum of squared error	Hard clustering	2022
[61]	Learning-automata, artificial Jellyfish search algorithm, marine predator algorithm	CMC, bank note authentication, glass, iris, LD, wine, breast cancer, divorce predictors, hepatitis C virus, blood transfusion service center	*	Sum of squared error	Hard clustering	2022
[62]	Symbiotic organism search, K-means	Breast, compound, flame, glass, iris, Jain, path-based, spiral, thyroid, two moons, wine, yeast	*	DB index	Automatic clustering	2022
[63]	Firefly algorithm, self-organized neural network	Iris, wine, cancer, CMC, vowel	*	Cartesian distance	Hard clustering	2022
[64]	Genetic-based metaheuristic encircle algorithm, dynamic K-modes	Yeast, wine, letter, WDBC, glass, aggregation, D31, R15, spiral, Jain, flame, iris, breast, fault	*	Chi-square, frequency-based measure	Hard clustering	2022

*Neighbouring search mechanism is not adopted in concerned research paper

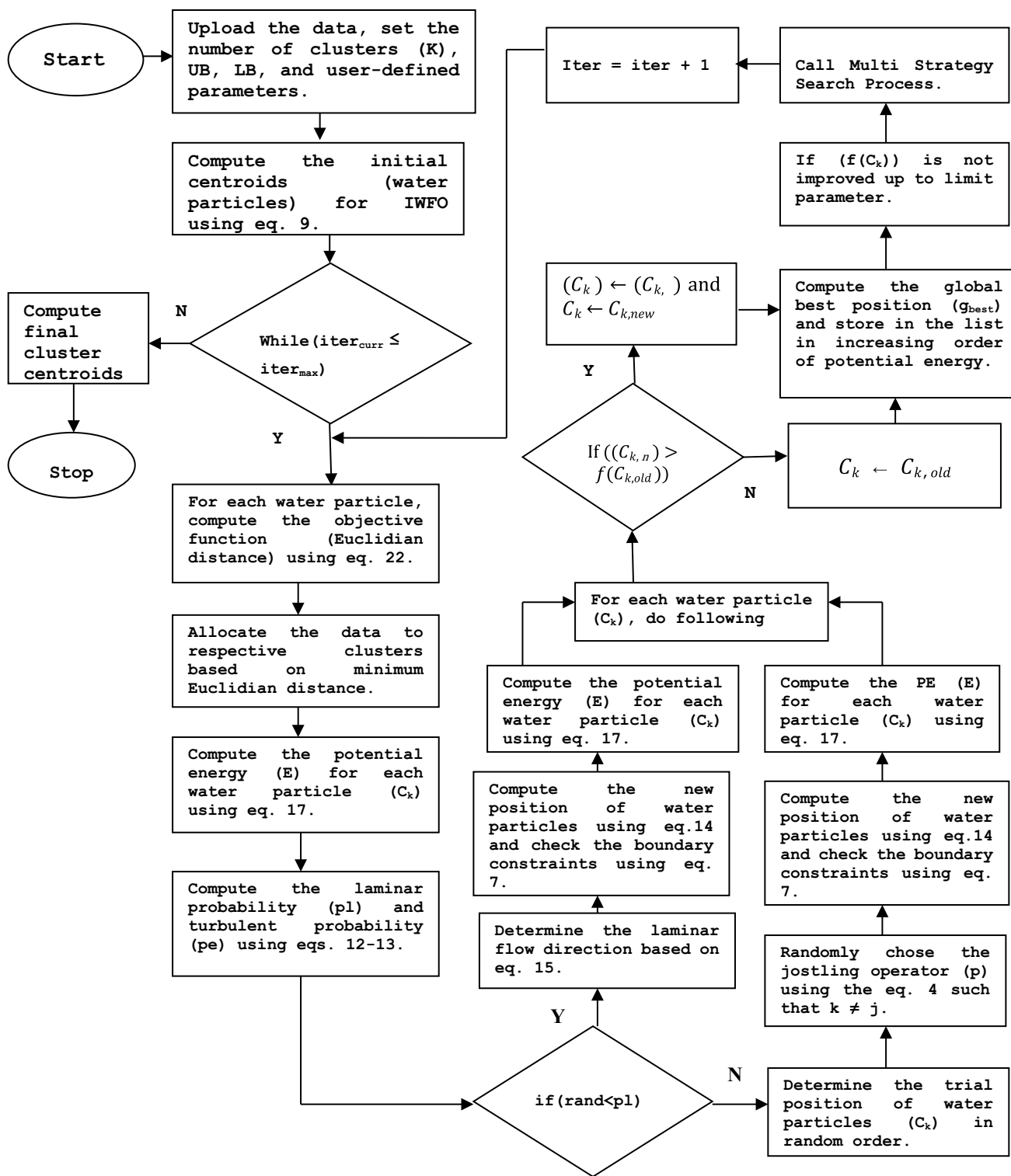


Fig. 1 Flowchart of the proposed IWFO clustering algorithm

43]. These shortcomings are expressed as—(i) a random distribution function is utilized to seed the population of WFO in a pseudo-chaotic manner. Hence, the process of finding the optimal solution is sometimes blind. (ii) Further, the

laminar flow (pl) operator and turbulent flow (pe) operator consist of constant values, which cannot adequately balance the exploration and exploitation ability of the algorithm. (iii)

It is also observed that the laminar flow phase includes a parallel one-way search strategy that may result in search holes, and in turn, WFO may fall into local optima. The aforementioned shortcomings of the WFO are handled through the gaussian chaotic map, an improved balance mechanism and a search strategy method. The main objectives of the work are highlighted below.

- The random distribution function is utilized to generate the initial population of the WFO algorithm in search space. This distribution function cannot generate a uniform population. Hence, a gaussian chaotic map is utilized to generate the population in a more systematic manner instead of random.
- The exploration and exploitation ability of the WFO algorithm cannot be well balanced due to constant values of the laminar flow (pl) operator and turbulent flow (pe) operator. The constant values of the laminar flow (pl) operator and turbulent flow (pe) operator are substituted by dynamic values that can be changed in each iteration. Hence, an improved balance mechanism is designed to improve the exploration and exploitation ability of the WFO algorithm.
- The one-way strategy is adopted in the laminar flow phase of the WFO algorithm to search the candidate solutions and in turn, an algorithm may stuck in local optima. This issue is resolved by using a neighbourhood search process.
- These improvements are integrated into the WFO algorithm for generating more optimal solutions, called the improved WFO (IWFO) algorithm. The proposed algorithm is adopted for solving the clustering problems. The task of this algorithm is to determine the optimal centroid for a given dataset.
- The efficacy of the proposed IWFO algorithm is evaluated using several well-known clustering datasets downloaded from the UCI repository based on SSE, AR and DR rates. The simulation results are compared with conventional as well as meta-heuristic algorithms. The findings stated that the proposed WFO algorithm obtains superior clustering results compared to other algorithms.

The rest of the manuscript is organized as related works based on meta-heuristic algorithms for clustering are summarized in "[Literature Review](#)". The information on the original water flow optimizer is discussed in "[Water Flow Optimizer](#)". "[Proposed Improved Water Flow Optimizer Algorithm \(IWFO\)](#)" presents the proposed improved water flow optimizer for clustering problems. The findings of the IWFO algorithm are discussed in "[Experimental Results](#)". The outcomes of this work are mentioned in "[Conclusion](#)".

Literature Review

The latest works on partitional clustering are discussed in this section.

Qtaish et al. [44] presented a hybrid capuchin search algorithm (HCSA) to deal with the local optima and initialization issues of the K-means clustering algorithm. Further, the chameleon swarm (CS) algorithm is adopted to strengthen the search mechanisms of the CSA algorithm. In addition, the aforementioned combination of the CS-CSA is used to generate the initial centroids for the K-means algorithm, called HCSA. The sixteen datasets are considered to evaluate the performance of the HCSA based on well-known clustering metrics. The simulation results are compared with nine meta-heuristic algorithms including k-means. The results revealed that the combination of CS-CSA-K-means successfully overrides the issues of K-means.

Kuo et al. [45] combined the three algorithms such as PSO, GA and GE with possibilistic fuzzy c-means (PFCM) for effective cluster analysis. Further, this study also integrates Atanassov's intuitionistic fuzzy sets (IFSs) to PFCM, called PIFCM. These algorithms are summarized as MOGA-PIFCM, MOPSO-PIFCM and MOGE-PIFCM. The performances of these algorithms are evaluated using fifteen standard clustering datasets. The simulation results are assessed using well-known clustering metrics. The results showed that the MOGE-PIFCM algorithm outperforms other clustering algorithms in terms of validation indices.

Premkumar et al. [46] introduced a K-means-based grey wolf optimizer (KCGWO) for handling the clustering problems. In KCGWO, the K-means algorithm is used to enhance the optimization capabilities of the traditional GWO. This integration aims to improve the diversity and convergence rate of the GWO algorithm. A new weight factor is also added in GWO to improve its performance. The performance of the KCGWO is assessed over a set of benchmark clustering datasets and results are examined using well-defined metrics. The results stated that KCGWO achieves more stable clustering results compared to other algorithms. This integration also obtains optimal centroids.

Demirci et al. [47] presented a new meta-heuristic algorithm called the electrical search algorithm (ESA) to solve the clustering problems. This algorithm is inspired by the movement of electricity. Further, the initialization process is defined through the special structures called poles. The search mechanism is described by the movement of electrons. The four benchmark datasets such as iris, wine, seeds and hepatitis C virus are considered to examine the performance of the ESA algorithm and the results are compared with seven existing meta-heuristic algorithms including K-means. This work also considers the Friedman

Signed Rank and post hoc Wilcoxon tests to evaluate the efficacy of ESA. The results showed that the ESA improves the clustering results in a significant manner compared to other algorithms.

Gharehchopogh and Khargoush [48] presented an improved version of the interactive autodidactic school (IAS) algorithm to solve the data clustering problems. To improve the exploitation process and also generate better populations, some chaotic maps are integrated into the ISA algorithm. The working of the ISA algorithm is described by three operators -individual training sessions, group training sessions, and new student challenges. The performance of the proposed chaotic ISA algorithm is examined over twenty clustering datasets. The results are evaluated using the best, average and worst solutions, and compared with state of art meta-heuristic algorithms. It is revealed that the Chebyshev chaotic function-based IAS algorithm obtains superior clustering results than other algorithms.

Ezgi combined the LaF and DE algorithms to discover the optimal centroids for data clustering problems [49]. The weak exploitation process of the LaF algorithm is improved using the DE/best/1 mutation operator. The performance of the proposed LaF-DE algorithm is evaluated using twelve clustering datasets based on the SSE and accuracy parameters. The results showed that the proposed LaF-DE provides better clustering results with eight datasets out of twelve. It is also noticed that the proposed algorithm also obtains satisfactory clustering results with the rest of the dataset compared to most of the algorithms being compared.

Duan et al. [50] handled the automatic clustering problem in high dimensional data by an improved affinity propagation based on an optimization algorithm. Initially, the dimensionality of data is reduced using the t-distributed stochastic neighbour method. Further, an improved equilibrium optimizer is utilized for optimizing the preference selection. The local search and convergence efficiency are enhanced using the crisscross strategy. Finally, the performance of the above-mentioned combination is assessed using seven high-dimensional datasets and results are compared with well-known four clustering algorithms based on NMI and RI. It is stated that the performance of the improved affinity propagation is significantly improved using the above-mentioned improvements.

An effective data clustering algorithm based on the chimp optimization algorithm (ChOA), generalized normal distribution algorithm (GNDA), and opposition-based learning (OBL) is reported for handling clustering problems [51]. Three different clustering algorithms are proposed for solving clustering problems and these algorithms are ChOA(I) and ChOA(II) based on chaotic maps, the combination of ChoAGNDA-OBL, and SO-ChOAGNDA. The performance of these algorithms is evaluated using five clustering datasets based on SSE and error rate. The

simulation results are compared with a wide variety of meta-heuristic clustering algorithms. It is analyzed that the SO-ChOAGNDA algorithm obtains lower SSE and error rates compared to other algorithms.

Singh et al. [52] presented an enhanced whale optimization algorithm (EWOA) for handling the clustering problems effectively. The whale optimization algorithm (WOA) suffers from local optima, convergence rate and trade-off issues. The trade-off issue of WOA is addressed through searching behaviour water wave optimization algorithms. The local optima and convergence issues are resolved through the neighbourhood mechanism and tabu search algorithm. The eight benchmark datasets are considered to examine the performance of the EWOA based on average intra-cluster distance and f-measure. The results showed that superior clustering results are obtained by the EWOA with most of the datasets.

A variable neighbourhood strategy-based firefly algorithm (VNS-FA) is presented for effective data clustering [53]. It is observed that the firefly algorithm (FA) converges on premature solutions due to a lack of exploitation capability. Further, variable neighbourhood strategy (VNS) is incorporated into FA to resolve the aforementioned issues. The efficacy of the proposed VNS-FA is evaluated using eight well-known clustering datasets. The results are assessed using intra-cluster distance, internal CH metric, entropy and F-measure parameters. The results stated that the proposed VNS-FA method obtains superior results with most of the datasets.

An improved grey wolf's optimization (IGWO) algorithm is presented for clustering and dynamic social networks [54]. This work aims to improve the accuracy rate of clustering problems. To achieve this, a label propagation algorithm is integrated into the GWO algorithm. The performance of the IGWO is examined using six well-known datasets based on NMI, intra-cluster distance, and error rate parameters. The results confirmed that the proposed IGWO algorithm obtains a better NMI rate than other clustering algorithms. It is also seen the hat proposed IGWO algorithm also gets a minimum error rate and intra-cluster distance than other algorithms.

A cat-based meta-heuristic algorithm is reported for addressing the partitional clustering [55]. Before implementing this algorithm, several modifications are inculcated into the cat algorithm in terms of tradeoff mechanism between local and global searches, diversity issues and premature convergence. In turn, an improved search mechanism, accelerated velocity equation and neighborhood mechanism are introduced for handling the aforementioned issues. The efficiency of the cat algorithm is examined over eight clustering datasets based on intra-cluster distance and f-measure parameters. It is analyzed that the proposed cat algorithm has a minimum intra-cluster distance with most of the dataset while obtaining a better f-measure rate than other algorithms.

Kushwaha et al. [56] presented an electromagnetic field optimization (EFO) method for resolving the issues of the k-mean algorithm. The k-mean algorithm suffers from poor selection of initial centroid and in turn traps in local optima. To overcome this issue of the k-mean algorithm, the EFO algorithm is utilized for generating the optimal initial centroid for the k-mean algorithm. It is also reported that the EFO algorithm may not stuck in local optima due to attraction and repulsion mechanisms. Several well-known datasets are considered for assessing the efficacy of the proposed clustering algorithm based on normalized mutual information (NMI), rand index (RI), and purity. The results showed that the proposed algorithm gets far better clustering results than the same class of algorithms.

To perform the clustering in massive data, Hashemi et al. [57] designed an updated PSO method. The proposed method utilizes the multi-start pattern reduction mechanism to decrease the calculation time for clustering. The clustering time is reduced using a reduction operator while the multi-start operator is utilized for ensuring population diversity and local minima. The six clustering datasets are considered to evaluate the performance of the proposed method based on accuracy and execution time. The results confirmed that better clustering results are obtained by the proposed PSO method.

Prior information on the cluster is the main prerequisite for partitional clustering, so the research community is paying close attention to automatic partitional clustering. The aforementioned problem of partitional clustering was also taken into account by Zhu et al. [58] who presented the AC-DPHS dynamic parameter-based HS algorithm for automatic data clustering. In the proposed AC-DPHS algorithm, the parameter is modified dynamically instead of static. The efficiency of the proposed AC-DPHS is assessed using five datasets. The results are evaluated using ARI, FM, and PBM parameters. It is stated that the proposed algorithm obtains superior results than other algorithms being compared.

A hybrid k-prototype clustering method based on enhanced SCA is developed by Kuo and Wang [59]. SCA algorithm is utilized to compute the optimal weight of attributes as well as initial attribute selection. Furthermore, the k-prototype method incorporates different mutation strategies, like Gaussian, Cauchy, levy, and single-point to produce better clustering outcomes. The ten datasets are chosen from the UCI repository to examine the efficacy of the k-prototype algorithm based on accuracy and Cohen kappa. The findings showed that the proposed k-prototype algorithm produces better clustering results than other algorithms.

A meta-heuristic clustering approach based on the behaviour of micro-bats is proposed by Kaur and Kumar

[60]. Several modifications are designed to resolve the issues related to the algorithms such as convergence rate, local optima, and trade-off issues. The elitist process handles the slow convergence rate, the initialization issue is handled by a collaborative approach. The effectiveness of the proposed bat-based metaheuristic algorithm is evaluated using several healthcare and non-healthcare datasets utilizing well-known performance metrics like intra-cluster distance, standard deviation, accuracy, and rand index. The results of the proposed approach are compared to conventional clustering algorithms and it is claimed that the proposed approach achieves a better accuracy rate than conventional algorithms.

A learning automata-based hybrid MPA and JS algorithm is presented for handling the data clustering problem effectively [61]. The MPA and learning automata are utilized for enhancing proficiency and reducing the complexity of the JS algorithm. MPA is applied to memorize the best solution achieved so far while learning automata is used to improve the learning strategy of the JS algorithm. The efficiency of the hybrid MPA-JS algorithm is evaluated using ten clustering datasets based on SSE, average execution time and CHC. The results are compared with several state of art meta-heuristic algorithms and it is found that the proposed hybrid MPA-JS algorithm gets better results than other algorithms.

To handle the automatic clustering problems, Ikotun and Ezugwu [62] hybridized the symbiotic organisms search algorithm with K-means. A global threshold function is also utilized for computing the outliers in the dataset and further, such data points are eliminated from the dataset. Moreover, a three-way mutation mechanism is also designed and integrated into the symbiotic organisms search algorithm to improve the performance of the aforementioned hybridization. The efficacy of the hybrid SOS-KM algorithm is examined over forty-two datasets based on DB index, CS index and computational time. The findings stated that the hybrid SOS-KM algorithm gets superior results for most of the dataset in terms of DB index, and CS index.

A hybrid algorithm based on a firefly algorithm (FA) and self-organizing map (SOM) is presented for the clustering task, called FA-SOM [63]. In the proposed FA-SOM algorithm, initial cluster centroids are selected using the FA. Further, the weight of SOM is optimized using optimal cluster centroids generated by FA. The six clustering datasets are utilized for evaluating the performance of the FA-SOM based on SSE parameters. A statistical test is also adopted to investigate the efficacy of the proposed FA-SOM algorithm. The findings stated that the proposed FA-SOM algorithm obtains superior clustering results than other methods.

To achieve better computational time, Suryanarayana et al. [64] developed a dynamic k-mode clustering algorithm for effective cluster analysis. The PSO algorithm is adopted

to compute the optimal centroid for the k-Mode algorithm. Further, a frequency-based technique is also utilized for updating the modes and decreasing the cost function for clustering tasks. The efficacy of the dynamic k-mode algorithm is evaluated using six well-known clustering datasets using accuracy, f-measure and NMI parameters. The results demonstrated that the proposed dynamic k-mode algorithm obtains more accurate clustering results than others (Table 1).

Water Flow Optimizer

Recently, a new meta-heuristic algorithm, called WFO based on the water flow theory has been developed for solving global optimization problems [40]. This algorithm is inspired by the shapes of water flow which is described through laminar and turbulent flows. Hence, the WFO algorithm also comprises two different evolutionary operators (laminar and turbulent) as stated above. In turn, the WFO algorithm imitates the hydraulic phenomenon of water particles i.e. flowing from highland to lowland by using two operators. The optimization process of the WFO is designed by using the laminar and turbulent operators. The optimization problems are described as either minimization or maximization problems. So, an objective function can be defined either in terms of minimization or maximization and this objective function is being solved by the optimization steps of the WFO algorithm. As WFO is inspired by the flow of water from highland to lowland, hence there exists a similar pattern among water flow and searching of the solutions in optimization problems. So, in WFO, a water particle can act as a possible solution, the position of the water particle corresponds to the solution value, and the objective function is described in terms of the potential energy of the water particle. The description of the laminar and turbulent operators is mentioned below.

Laminar Operator

The working of this operator is inspired by a laminar flow of water and this flow specified that water particles should be moved in parallel straight lines as mentioned in Fig. 1. Further, the particle velocities can differ depending on the viscosity of water. The particles that are far from obstacles or walls, can move faster than those closer to obstacles and walls. Mathematically, this behaviour of water flow is demonstrated using Eq. (1), called the laminar operator.

$$y_t(i) = x_t(i) + R \times \bar{V} \forall t = \{1, 2, 3, 4, \dots, m\} \quad (1)$$

In Eq. (1), $y_t(i)$ described as the moving position of t th water particle after i th iteration, $x_t(i)$ corresponds to the position of t th water particle in i th iteration, R is a random number in the range of $[0, 1]$, called water coefficient and \bar{V} defines a vector value corresponding to a common motional direction (dimension) and m is the total number of water particles. The direction of common motional using Eq. (2).

$$\bar{V} = x_e(i) - x_o(i) \text{ such that } (e \neq f, (f(x_e(i)) \leq f(x_o(i)))) \quad (2)$$

In Eq. (2), $x_e(i) \leq x_o(i)$ describes the potential energy of water particles that can be used to select the best water particle in the i th iteration. If e th particle energy is lower than o th particle energy, then the best particle is $x_e(i)$, otherwise $x_o(i)$.

Turbulent Operator

The working of this operator is inspired by the turbulent flow of water and this can be described as a contingent pushing of water particles. In turn, an inconsistent motion can disrupt the bonding of water particles, and produce the fast flow of water to destabilize the obstruction and cause a local oscillation. In turn, amplitude is generated for oscillation, and the amplitude can grow with time. A shearing force is produced due to the aforementioned process, known as torque, which is responsible for swirling water particles. If the dimension of the problem to be solved can be described through a layer of water, then the dimension transformation of problems can be viewed as an irregular motion of water particles in turbulent flow. Mathematically, it can be understood as randomly selecting the dimension and position of particles during oscillation. Finally, the behaviour of the turbulent operator is described through Eq. (3).

Table 2 Description of the datasets

Sr. no.	Data sets	Cluster (K)	Dimension (D)	Instance (N)
1	Iris	3	4	150
2	Glass	6	9	214
3	Wine	3	13	178
4	Ionosphere	2	34	351
5	Control	6	60	600
6	Vowel	6	3	871
7	Balance	3	4	625
8	Crude oil	3	5	511
9	CMC	3	9	1,473
10	LD	2	6	416
11	WBC	2	9	699
12	Thyroid	3	5	215

$$y_t(i) = (x_t^d(i), p, x_t^{d+1}(i), p, x_t^{d+2}(i), p, x_t^{d+3}(i), \dots, p, x_t^{d+n}(i)) \quad (3)$$

where $d = \{1, 2, 3, \dots, n\}$

In Eq. (3), 'p' corresponds to the jostling operator, and it can be described through Eq. (4).

$$p = \begin{cases} \gamma(x_t^d(i), x_j^d(i)), & \text{if } r < pe \\ \vartheta(x_t^d(i), x_j^{d+1}(i)), & \text{otherwise} \end{cases} \quad (4)$$

In Eq. (4), j describes the index of the randomly chosen particle (e) such as $j \in \{1, 2, 3, \dots, s\}$ and $i \neq j$. ' $d+1$ ' defines a dimension that can be chosen randomly such that $d+1 \neq d$. pe can be defined as an eddying parameter and r is a random number in the range of $[0, 1]$. Further, it is observed that the eddy shape is similar to the Archimedean spiral and it can be expressed using the Eq. (5).

$$\gamma(x_t^d(i), x_j^d(i)) = x_t^d(i) + \varphi \times \theta \times \cos(\theta) \quad (5)$$

In Eq. (5), θ is a random value in the range of $[-\pi, \pi]$ and φ corresponds to the shearing force between t th and e th particles. The shearing force (φ) between t th and e th particles is determined using Eq. (6).

$$\varphi = |x_t^d(i), x_j^d(i)| \quad (6)$$

Further, the general behaviour of water particles is described through a transformation function and this function is summarized in Eq. (7).

$$\vartheta(x_t^d(i), x_j^{d+1}(i)) = (ub^d - lb^d) \frac{x_j^{d+1}(i) - lb^{d+1}}{ub^{d+1} - lb^{d+1}} \quad (7)$$

In Eq. (7), ub^d denotes the upper bound in d th dimension, lb^d denotes the lower bound of the d th dimension. $x_j^{d+1}(i)$ denotes the j th index of e th particle in dimension $(d+1)$.

Evolutionary Rule

This subsection discusses that the water flow can be either laminar or turbulent. The distinction between laminar and turbulent flows can be made with the help of the Reynolds number. It is stated that a threshold function is defined to determine the flow of water. If the flow of water is less than a threshold, then it can be considered as laminar, otherwise turbulent. Further, laminar flow probability can be denoted through (P_l) , whereas, turbulent flow probability is defined using $(1 - (P_l))$. Moreover, laminar probability can be described as a control parameter and it is a random number in the range of $[0, 1]$. The flow of water is determined using P_l . However, it is also discussed that water can be flowed

from highland to low land, in turn, the position of water particles may change. Hence, this behaviour of water can be characterized using the evolutionary rule. As per this rule, the moving position of the water particles can be changed according to their potential energy, if potential energy is less than the threshold, a new moving position is calculated for the water particles otherwise, there is no change in the current position. This behaviour is illustrated using Eq. (8).

$$X_t(i+1) = \begin{cases} y_t(i), & \text{if } f(y_t(i)) \leq f(x_t(i)) \\ x_t(i), & \text{otherwise} \end{cases} \quad (8)$$

Proposed Improved Water Flow Optimizer Algorithm (IWFO)

This section presents the improved water flow optimizer algorithm for solving the hard clustering problems. It is observed that several limitations are associated with the water flow algorithm such as (i) random distribution function is utilized to generate the initial population, (ii) adequately less balance between exploration and exploitation mechanisms, and (iii) due to one-way strategy adopted in laminar flow phase, WFO may be stuck in local optima. Firstly, these aforementioned limitations of the WFO algorithm are handled through a logistic chaotic map-based function for generating the initial population, the balance between exploration and exploitation mechanisms is enhanced using improved solution search equations, and the local optima issue is handled through a multi-strategy search process. The proposed improvements are discussed below.

Chaotic Map-Based Distribution Function

In WFO, a random function is applied to generate the initial population. But, a *rand* (\cdot) function provides a random number between 0 and 1. Further, it cannot provide random numbers in uniform order. Hence, the population cannot be generated uniformly throughout the search space. The initial population for WFO is computed using Eq. (9).

$$x = lb + rand(N, D) \cdot [ub - lb] \quad (9)$$

In Eq. (9), x denotes the initial population of water particles, lb and ub denote the lower and upper bounds of the search space, N represents the total number of water particles and D can be defined as the dimension of the search space. Suppose, the number of water particles (N) is 4, dimension (D) is 3 and lb for the search space is given as $(0, 10, 25)$ and ub is given as $(30, 50, 60)$. Now, a *rand*(\cdot) function is used to generate four numbers in the range of $[0, 1]$. These numbers are integrated with Eq. 9 for generating the initial population of the water particles and the initial population

Table 3 Comparison of the simulation results of the proposed IWFO algorithm and other standard existing clustering algorithms using intra, SD and rank parameters

Datasets	Measure	Existing well-known clustering algorithms										Proposed IWFO
		K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	WFO		
Iris	Intra	9.20E+01	9.86E+01	1.01E+02	1.08E+02	1.21E+02	1.25E+02	9.68E+01	1.15E+02	9.52E+01	9.21E+01	
	SD	2.67E+01	4.67E-01	1.31E+00	6.63E+00	5.23E+00	1.46E+01	4.22E+00	3.76E+01	2.80E+00	3.20E+00	
	Rank	2	5	6	7	9	10	4	8	3	1	
Glass	Intra	3.79E+02	2.76E+02	2.19E+02	3.29E+02	3.62E+02	2.82E+02	6.64E+02	3.75E+02	2.26E+02	2.03E+02	
	SD	7.05E+01	1.86E+01	3.36E+00	1.14E+01	1.21E+01	4.14E+00	6.89E+01	1.03E+01	1.23E+01	9.98E+00	
	Rank	9	4	2	6	7	5	10	8	3	1	
Wine	Intra	1.81E+04	1.64E+04	1.62E+04	1.69E+04	1.58E+04	1.66E+02	1.67E+04	1.71E+04	1.65E+04	1.59E+04	
	SD	9.06E+02	8.55E+01	3.69E+01	4.74E+02	5.60E+01	7.84E+01	6.29E+01	5.66E+01	4.74E+01	4.34E+01	
	Rank	10	4	3	8	1	6	7	9	5	2	
Ionosphere	Intra	2.42E+03	1.01E+03	9.38E+02	1.11E+03	1.13E+03	1.00E+03	1.07E+03	1.33E+03	1.07E+03	9.04E+02	
	SD	4.55E+02	3.34E+02	4.48E+02	2.61E+02	3.17E+02	4.13E+02	2.99E+02	2.34E+02	1.21E+02	1.53E+01	
	Rank	10	3	2	7	8	4	6	9	5	1	
Control	Intra	1.01E+06	4.18E+04	2.39E+04	5.12E+04	5.23E+04	4.62E+04	2.38E+04	2.68E+04	2.52E+04	2.41E+04	
	SD	5.05E+03	1.02E+03	1.71E+02	1.32E+03	9.16E+02	1.58E+03	1.09E+02	1.78E+02	2.18E+02	1.42E+02	
	Rank	10	6	4	8	9	7	1	5	3	2	
Vowel	Intra	1.60E+05	1.58E+05	1.89E+05	1.70E+05	1.81E+05	1.59E+05	1.94E+05	1.96E+05	1.63E+05	1.56E+05	
	SD	4.52E+03	2.88E+03	2.58E+03	4.64E+03	2.86E+03	3.11E+03	2.44E+04	3.98E+03	3.26E+02	1.14E+02	
	Rank	4	2	7	6	8	3	9	10	5	1	
Balance	Intra	1.20E+05	6.20E+04	5.94E+04	6.61E+04	6.78E+04	6.91E+04	5.96E+04	6.02E+04	5.89E+04	5.78E+04	
	SD	9.28E+03	4.01E+03	7.56E+02	6.79E+02	5.25E+03	5.62E+03	3.72E+02	8.26E+02	5.17E+02	3.34E+02	
	Rank	10	9	3	6	7	8	4	5	2	1	
Crude oil	Intra	2.91E+02	2.86E+02	2.47E+02	2.81E+02	3.69E+02	2.83E+02	2.61E+02	2.89E+02	2.85E+02	2.63E+02	
	SD	2.63E+01	1.14E+01	4.71E+01	1.09E+01	2.33E+01	8.14E+00	1.17E+02	1.76E+01	1.90E+01	1.08E+01	
	Rank	9	7	1	4	10	5	2	8	6	3	
CMC	Intra	5.59E+03	5.85E+03	5.83E+03	5.94E+03	5.95E+03	5.77E+03	5.71E+03	5.79E+03	5.76E+03	5.64E+03	
	SD	4.68E+01	4.89E+01	1.23E+02	1.31E+02	8.69E+01	5.04E+01	2.86E+01	3.67E+01	2.74E+01	2.14E+01	
	Rank	1	8	7	9	10	5	3	6	4	2	
LD	Intra	1.17E+04	3.24E+03	3.25E+03	9.85E+03	1.15E+04	2.54E+03	1.13E+03	1.74E+03	1.68E+03	1.23E+03	
	SD	6.68E+02	2.88E+01	1.64E+01	8.20E+02	2.07E+03	4.18E+01	2.41E+01	1.52E+01	2.97E+01	1.27E+01	
	Rank	10	6	7	8	9	5	1	4	3	2	
WBC	Intra	1.93E+04	4.26E+03	3.37E+03	3.50E+03	3.73E+03	3.00E+03	2.96E+03	3.06E+03	3.02E+03	2.85E+03	
	SD	5.14E-12	2.08E+02	4.17E+01	2.12E+02	1.84E+02	2.25E+02	5.57E+02	1.98E+02	1.93E+02	1.46E+02	
	Rank	10	9	6	7	8	3	2	5	4	1	

Table 3 (continued)

Datasets	Measure	Existing well-known clustering algorithms									
		K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	WFO	Proposed IWFO
Thyroid	Intra	2.39E+03	1.11E+04	1.99E+03	1.98E+03	2.97E+03	1.22E+04	1.94E+03	1.39E+03	1.48E+03	1.27E+03
	SD	2.46E+02	2.71E+01	3.09E+01	2.23E+02	2.06E+01	3.26E+01	1.95E+02	2.28E+01	2.37E+01	1.12E+01
	Rank	7	9	6	5	8	10	4	2	3	1
Average ranking		7.67	6	4.5	6.75	7.83	5.92	4.42	6.58	3.83	1.5

is expressed as (24.6270, 27.4879, 29.0764), (27.2680, 19.3384, 56.8596) and (4.6826, 13.8287, 28.7677). By analyzing the population, it is noticed that the population cannot explore the entire search space effectively. On the other side, search space can be examined more systematically using chaotic maps rather than *rand()* function. This work employs the gaussian chaotic map for generating the initial population of water particles and it is computed using Eq. (10).

$$x = lb + c_k(N, D) \cdot [ub - lb] \tag{10}$$

Here, *x* denotes the population of water particles, *lb* and *ub* are lower and upper bounds, *N* is the number of water particles and *D* is the dimension, *c_k* denotes *k*th Gaussian map and it is computed using Eq. (11).

$$c_{k+1} = \mu \times e^{c_k \mu} \in [-1, 1] \tag{11}$$

In Eq. (11), *μ* is a user-defined parameter in between -1 to 1, but the optimal value of *μ* is -0.2, *c_k* denotes the *k*th Gaussian map, and *c_{k+1}* denotes the (*k* + 1)th chaotic map, and the value of *c₀* is set to 0.6.

Balancing Exploration and Exploitation Mechanisms

The exploration and exploitation mechanisms are an integral part of every meta-heuristic algorithm. The exploration can be understood as a global search, while exploitation can be described as a local search. However, these mechanisms should be balanced to achieve effective solutions. It is noticed that a weak local search cannot exploit then search space efficiently. On the other side, the global search is responsible for exploring the solution that can be found during the local search. This search also determines the capability of a local solution whether it can be acted as either a global solution or not. Moreover, the global search is also directed the search towards the global optima. To guide the solution towards global optima, the search process also knows the direction of the previously best solution. Hence, two nonlinear (sigmoid and tanh) functions are taken into consideration for obtaining a better tradeoff between the search mechanisms. In WFO, the search mechanisms are described by the laminar operator (*pl*) and turbulent operator (*pe*) which are summarized in Eqs. (12–13).

$$pl = (c_1 - c_2) \times \left(\frac{1}{1 + e^{-t_{curr}/t_{max}}} \right) \tag{12}$$

$$pe = c_2 \times \left(\frac{2}{1 + e^{-2t_{curr}/t_{max}}} - 1 \right) + c_1 \tag{13}$$

Here, *c₁* and *c₂* are two cognitive parameters whose values are 0.5 and 0.3. *t_{curr}* is the current iteration and *t_{max}* is

Table 4 Comparison of the simulation results of the proposed IWFO algorithm and other recently developed clustering algorithms using intra, SD and rank parameters

Dataset	Measure	Recent clustering algorithms									
		VS	MBOA	WOA	ICSO	TLBO	CS	GSA	LION	GWO	IWFO
Iris	Intra	9.89E+01	9.83E+01	9.84E+01	9.57E+01	9.69E+01	9.64E+01	9.79E+01	9.76E+01	9.98E+01	9.21E+01
	SD	1.21E+00	1.24E+00	1.06E+00	1.92E+00	2.88E+00	3.25E+00	3.19E+00	4.03E+00	3.12E+00	3.20E+00
	Rank	9	7	8	2	4	3	6	5	10	1
Glass	Intra	2.34E+02	2.31E+02	2.18E+02	2.26E+02	2.37E+02	2.41E+02	2.39E+02	2.38E+02	2.36E+02	2.03E+02
	SD	1.01E+01	1.27E+01	1.37E+01	1.10E+01	1.09E+01	1.12E+01	9.87E+00	1.07E+01	1.41E+01	9.98E+00
	Rank	5	4	2	3	7	10	9	8	6	1
Wine	Intra	1.83E+04	1.91E+04	1.84E+04	1.69E+04	1.68E+04	1.65E+04	1.70E+04	1.65E+04	1.66E+04	1.59E+04
	SD	6.90E+01	7.07E+01	4.83E+01	3.26E+01	2.94E+01	2.64E+01	2.74E+01	2.66E+01	1.94E+01	4.34E+01
	Rank	8	10	9	6	5	2	7	3	4	1
Ionosphere	Intra	2.25E+03	2.93E+03	2.45E+03	1.45E+03	1.11E+03	1.23E+03	2.83E+03	1.42E+03	1.58E+03	9.04E+02
	SD	3.67E+01	4.39E+01	3.69E+01	3.62E+01	2.35E+01	3.46E+01	4.04E+01	3.38E+01	2.98E+01	1.53E+01
	Rank	7	10	8	5	2	3	9	4	6	1
Control	Intra	3.15E+04	3.04E+04	2.99E+04	2.51E+04	2.55E+04	3.03E+04	3.13E+04	2.75E+04	2.63E+04	2.41E+04
	SD	1.17E+02	9.68E+02	8.47E+01	7.08E+01	4.83E+01	6.18E+01	5.37E+01	4.47E+01	6.65E+01	1.42E+02
	Rank	10	8	6	2	3	7	9	5	4	1
Vowel	Intra	1.59E+05	1.62E+05	1.58E+05	1.55E+05	1.57E+05	1.59E+05	1.60E+05	1.59E+05	1.61E+05	1.56E+05
	SD	4.52E+02	2.94E+02	2.54E+02	1.77E+02	1.78E+02	3.42E+01	3.87E+01	1.74E+01	2.09E+01	1.14E+02
	Rank	7	10	4	1	3	5	8	6	9	2
Balance	Intra	5.84E+04	5.96E+04	6.06E+04	5.39E+04	5.36E+04	5.79E+04	5.81E+04	5.86E+04	5.83E+04	5.78E+04
	SD	1.04E+02	1.81E+02	1.83E+02	2.17E+02	1.29E+02	2.19E+02	3.02E+02	2.98E+02	1.78E+02	3.34E+02
	Rank	7	9	10	2	1	4	5	8	6	3
Crude oil	Intra	2.81E+02	2.86E+02	2.83E+02	2.64E+02	2.59E+02	2.74E+02	2.71E+02	2.69E+02	2.68E+02	2.63E+02
	SD	1.54E+01	2.08E+01	1.62E+01	1.48E+01	1.37E+01	1.24E+01	1.75E+01	1.39E+01	1.50E+01	1.08E+01
	Rank	8	10	9	3	1	7	6	5	4	2
CMC	Intra	5.76E+03	5.21E+03	5.69E+03	5.32E+03	5.65E+03	5.85E+03	5.67E+03	5.70E+03	5.73E+03	5.64E+03
	SD	6.42E+01	5.60E+01	6.98E+00	7.45E+00	6.38E+00	2.35E+02	6.32E+01	9.47E+01	8.35E+01	2.14E+01
	Rank	9	1	6	2	4	10	5	7	8	3
LD	Intra	1.52E+03	1.32E+03	1.91E+03	1.41E+03	1.50E+03	1.39E+03	1.73E+03	1.33E+03	1.36E+03	1.23E+03
	SD	6.32E+01	6.52E+01	5.91E+01	2.95E+01	2.39E+01	5.62E+01	4.79E+01	2.68E+01	2.15E+01	1.27E+01
	Rank	8	2	10	6	7	5	0	3	4	1
WBC	Intra	4.09E+03	3.61E+03	3.76E+03	3.12E+03	2.99E+03	3.72E+03	2.92E+03	2.89E+03	3.13E+03	2.85E+03
	SD	4.33E+01	2.87E+01	2.26E+01	1.81E+01	2.35E+01	1.73E+01	8.36E+00	7.67E+00	1.65E+01	1.46E+02
	Rank	10	7	9	5	4	8	3	2	6	1

Table 4 (continued)

Dataset	Measure	Recent clustering algorithms									
		VS	MBOA	WOA	ICSO	TLBO	CS	GSA	LION	GWO	IWFO
Thyroid	Intra	1.96E+03	2.16E+03	1.67E+03	9.90E+02	1.68E+03	1.43E+03	1.86E+03	1.54E+03	1.39E+03	1.27E+03
	SD	1.81E+01	1.66E+01	2.09E+01	1.16E+01	1.48E+01	2.39E+01	1.90E+01	2.46E+01	1.73E+01	1.12E+01
	Rank	9	10	6	1	7	4	8	5	3	2
Average ranking		8.08	7.33	7.25	3.17	4	5.67	6.25	5.08	5.83	1.58

maximum iteration. If, $(pl > rand)$, the updated position of water particles (y) is computed by Eq. (14).

$$y_t(i) = \begin{cases} x_t(i) + r \times \vec{d}, & \text{if } rand < pl \\ x_t(i) + dt.(x_{best} - x_t(i)), & \text{otherwise} \end{cases} \quad (14)$$

In Eq. (14), $x_t(i)$ is the current position of the water particle in i th iteration, x_{best} is the best position, r is a random number in between $[0, 1]$, \vec{d} is the direction of water particles and dt is a decrement operator inspired by the whale optimization algorithm. The direction of water particles is computed using Eq. (15), while dt is determined by Eq. (16).

$$\vec{d} = x_t(i) - x_s(i) \text{ such that } (t \neq s, f(x_t(i)) \leq f(x_s(i))) \quad (15)$$

Here, $x_t(i)$ and x_s denote the t th and s th water particles in i th iteration, $f(x_t(i))$ and $f(x_s(i))$ is the potential energy of t th and s th water particles. Further, the potential energy of the particle (x_t) is computed using the Eq. (17)

$$dt = \frac{1}{1 - \cos(\pi l)} \quad (16)$$

In Eq. (16), l is a random number in the range of $[0.6, 1]$. Further, the potential energy ($f(x_t(i))$) is computed using the Eq. (17).

$$f(x_t(i)) = \frac{\sum_i^n SSE(x_t(i))}{\sum_{i=1}^n SSE(x_t(i))} \quad (17)$$

Here, $f(x_t(i))$ is the potential energy of the t th water particle (x_t), and $SSE(x_t(i))$ is the sum of the squared error. It can be defined as the sum of the squared error of the given particle divided by the sum of the squared error of all particles.

Neighborhood Search Process

This subsection discusses the neighbourhood search process to overcome the single search strategy of WFO. This process also handles the local optima issue of the WFO algorithm effectively. In the literature, neighbourhood search mechanisms have been employed to overcome local optima issues [65, 66]. This work proposes three methods to overwhelm the local optima- (i) Gaussian local search strategy, (ii) Mutation strategy, and (iii) Crossover strategy.

Gaussian local search strategy: It is observed that local search is responsible for finding the optimal solutions by exploring the search space and this solution is called local optimal solution. On the other side, it is also noticed that local search sometimes converges quickly and in turn solution sticks in local optima. Hence, to avoid the solu-

Table 5 Simulation results of the proposed IWFO and other standard existing clustering algorithms using AR and DR parameters

Dataset	Parameter	KM	PSO	ACO	ABC	DE	GA	BB-BC	BAT	WFO	Proposed IWFO
Iris	AR	0.673	0.833	0.789	0.887	0.842	0.741	0.868	0.904	0.908	0.961
	DR	0.696	0.857	0.794	0.892	0.868	0.773	0.882	0.907	0.914	0.968
Glass	AR	0.519	0.537	0.374	0.489	0.481	0.49	0.555	0.484	0.692	0.717
	DR	0.538	0.572	0.384	0.509	0.492	0.511	0.586	0.504	0.687	0.743
Wine	AR	0.739	0.711	0.746	0.773	0.741	0.729	0.766	0.787	0.858	0.873
	DR	0.752	0.737	0.781	0.793	0.762	0.749	0.794	0.813	0.891	0.896
Ionosphere	AR	0.712	0.648	0.607	0.644	0.63	0.601	0.626	0.621	0.708	0.789
	DR	0.728	0.647	0.612	0.664	0.653	0.618	0.634	0.632	0.724	0.804
Control	AR	0.597	0.412	0.395	0.356	0.393	0.467	0.394	0.668	0.754	0.801
	DR	0.614	0.452	0.437	0.396	0.417	0.492	0.421	0.695	0.783	0.819
Vowel	AR	0.763	0.753	0.775	0.796	0.698	0.745	0.813	0.832	0.781	0.886
	DR	0.846	0.795	0.806	0.832	0.747	0.79	0.856	0.859	0.896	0.914
Balance	AR	0.85	0.898	0.743	0.767	0.75	0.78	0.797	0.868	0.887	0.913
	DR	0.863	0.904	0.772	0.783	0.776	0.824	0.835	0.879	0.896	0.932
Crude oil	AR	0.655	0.765	0.591	0.568	0.665	0.632	0.636	0.632	0.664	0.796
	DR	0.684	0.773	0.645	0.587	0.681	0.654	0.649	0.654	0.675	0.823
CMC	AR	0.357	0.514	0.369	0.416	0.437	0.403	0.447	0.426	0.595	0.613
	DR	0.455	0.598	0.465	0.489	0.466	0.435	0.512	0.487	0.671	0.686
LD	AR	0.522	0.541	0.529	0.499	0.52	0.493	0.502	0.531	0.565	0.685
	DR	0.635	0.587	0.564	0.549	0.648	0.59	0.613	0.654	0.634	0.746
Cancer	AR	0.698	0.721	0.749	0.786	0.654	0.691	0.67	0.792	0.781	0.836
	DR	0.751	0.748	0.773	0.824	0.678	0.715	0.698	0.836	0.816	0.868
Thyroid	AR	0.638	0.689	0.649	0.644	0.658	0.632	0.639	0.658	0.669	0.738
	DR	0.661	0.692	0.654	0.661	0.697	0.643	0.674	0.703	0.783	0.786

tion stuck in local optima, this work explores the Gaussian local search strategy for computing the local optimal solution. This process is summarized below.

$$x_t'(i) = (1 - \epsilon) \times x_i(i) + \tau_c \tag{18}$$

In Eq. (18), $x_t'(i)$ represents the new position of the water particle, $x_i(i)$ denotes the current position of the water particle that can be responsible for local optima, τ_c is a Gaussian chaotic variable which is computed using the Eq. (19).

$$\tau_c = lb^d + c_k(ub^d - lb^d) \times g_{best}^d \text{ where } d \in (1, 2, 3, \dots, D) \tag{19}$$

In Eq. (19), lb^d and ub^d denote the lower and upper constraints in the d th dimension, g_{best}^d denotes the best position of water particles, c_k denotes the logistic chaotic map that can be computed using Eq. (11).

Mutation-based strategy: The local optima issue arises due to similar populations in successive iterations. In turn, there is no change in the allocation of data objects to the respective clusters. Hence, this work also explores the capability of the mutation operator to generate a diverse population to avoid the local optima condition. The new

position of water particles is generated by considering the three previous personal best positions of water particles in a random fashion such as x_e, x_f, x_g where, $e \neq f \neq g$ and this behaviour is depicted using Eq. (20).

$$x_t'(i) = x_e(i) + c_k \times (x_f(i) - x_g(i)) \tag{20}$$

In Eq. (20), c_k denotes the gaussian chaotic map computed using Eq. (11) and it is also utilized to control the chaos of mutation parameter.

Crossover-based strategy: This strategy is derived from the concept of the crossover operator. The new position is generated by performing the two-point crossover between two randomly chosen gbest water particles and it is expressed in Eq. (21).

$$x_t'(i) = \begin{cases} x_i(i) + c_k(x_p - x_q) & \text{if, rand} < (f(x_p) || f(x_q)) \\ x_i(i) + c_k \times \left(1 - \frac{t_{curr}}{t_{max}}\right) & \text{otherwise} \end{cases} \tag{21}$$

In Eq. 21, $x_t'(i)$ denotes the new position of water particle, $x_i(i)$ denotes the previous position of the water particle, x_p , and x_q are two randomly chosen water particles from the list of gbest particles, $f(x_p)$ and $f(x_q)$ describe the potential

Table 6 Simulation results of the proposed IWFO and recent clustering algorithms based on AR and DR parameters using recent clustering algorithms

Dataset	Parameter	VS	MBOA	WOA	ICSO	TLBO	CS	GSA	LION	GWO	Proposed IWFO
Iris	AR	0.941	0.954	0.946	0.914	0.912	0.885	0.783	0.852	0.852	0.961
	DR	0.956	0.959	0.951	0.932	0.928	0.893	0.774	0.812	0.868	0.968
Glass	AR	0.589	0.593	0.684	0.691	0.695	0.689	0.664	0.681	0.678	0.717
	DR	0.602	0.625	0.693	0.726	0.742	0.726	0.684	0.713	0.709	0.743
Wine	AR	0.697	0.7031	0.684	0.732	0.725	0.803	0.79	0.878	0.792	0.873
	DR	0.737	0.717	0.703	0.754	0.767	0.824	0.818	0.908	0.837	0.896
Ionosphere	AR	0.633	0.646	0.611	0.687	0.694	0.735	0.752	0.769	0.746	0.789
	DR	0.649	0.683	0.629	0.725	0.737	0.746	0.762	0.779	0.753	0.804
Control	AR	0.578	0.603	0.620	0.713	0.728	0.698	0.674	0.738	0.814	0.801
	DR	0.629	0.634	0.673	0.747	0.734	0.722	0.694	0.762	0.833	0.819
Vowel	AR	0.640	0.569	0.587	0.653	0.649	0.835	0.847	0.851	0.815	0.886
	DR	0.678	0.581	0.618	0.682	0.652	0.866	0.858	0.897	0.842	0.914
Balance	AR	0.713	0.720	0.693	0.786	0.810	0.917	0.849	0.855	0.793	0.913
	DR	0.732	0.756	0.728	0.816	0.856	0.938	0.887	0.896	0.845	0.932
Crude oil	AR	0.623	0.652	0.635	0.746	0.734	0.704	0.761	0.805	0.748	0.796
	DR	0.647	0.663	0.658	0.779	0.752	0.713	0.792	0.834	0.768	0.823
CMC	AR	0.411	0.442	0.424	0.468	0.465	0.571	0.547	0.534	0.516	0.613
	DR	0.456	0.482	0.443	0.501	0.483	0.614	0.582	0.558	0.548	0.686
LD	AR	0.509	0.507	0.514	0.530	0.532	0.659	0.631	0.662	0.615	0.685
	DR	0.568	0.549	0.583	0.571	0.609	0.689	0.654	0.713	0.648	0.746
Cancer	AR	0.792	0.853	0.834	0.918	0.921	0.820	0.728	0.789	0.824	0.836
	DR	0.826	0.884	0.876	0.946	0.952	0.841	0.786	0.841	0.872	0.868
Thyroid	AR	0.604	0.594	0.621	0.682	0.714	0.691	0.678	0.724	0.706	0.738
	DR	0.647	0.628	0.655	0.697	0.748	0.748	0.729	0.759	0.739	0.786

energy of p th and q th particles, c_k denotes the gaussian chaotic map which is computed using Eq. 11, t_{curr} denotes the current iteration and t_{max} denotes the maximum number of iterations.

Algorithmic Steps of the IWFO Algorithm

This subsection presents the algorithmic steps of the proposed IWFO algorithm for clustering problems. The IWFO algorithm aims to determine the optimal centroids for the given dataset. The working of the IWFO algorithm is defined through six major steps. These steps are (i) Initialization, (ii) Objective Function and Data Allocation, (iii) Laminar Flow, (iv) Turbulent Flow, (v) Evolving and updation. The descriptions of these steps are mentioned below.

- (i) Initialization: This step corresponds to user-defined parameters of the IWFO algorithm. The parameters include the number of particles (clusters (K)), dimension (D), laminar probability (pl), eddying probability (pe), lb , ub , and maximum iteration. The initial position of the water particles is computed by

Eq. (10). Further, initial positions are also described as the initial centroids for cluster problems.

- (ii) Objective function and data allocation: This step defines a problem-related objective function. For clustering problems, the objective function is defined using a distance function and this function is used to allocate the data to clusters. So, in this work, an objective function is defined in terms of the Euclidean distance and the allocation of data to respective clusters is done based on the minimum Euclidean distance. This function is defined in Eq. (22).

$$D(x_i, c_j) = \sqrt{\sum_{m=1}^d (x_{im} - c_{jm})^2} \quad (22)$$

Here, $D(x_i, c_j)$ is the Euclidean distance between data (x_i) and centroids (c_j), d denotes the dimension of data, and it is expressed as $m = 1, 2, 3, \dots, d$.

- (iii) Laminar flow: When data is allocated to respective clusters, the next step is to compute the direction of water flow. It can be achieved by comparing the lami-

Table 7 Depicts the execution time of clustering algorithms (in seconds)

Dataset	Clustering algorithm										
	KM	DE	ACO	PSO	ABC	GA	BB-BC	BAT	WFO	Proposed IWFO	
CMC	11.31	19.25	17.4	14.58	14.97	16.1	16.73	15.7	13.6	14.23	
Glass	11.19	15.19	15.9	14.01	16.22	16.8	14.98	14.7	15.3	16.11	
crude	9.15	14.56	14.2	14.02	13.79	13.2	12.79	14.1	11.3	12.09	
Liver	13.76	15.96	15.7	14.07	12.69	13.7	15.67	11.9	12.2	13.01	
Cancer	13.09	15.88	14.3	13.34	13.98	13.3	14.09	13.1	13.2	13.83	
Thyroid	17.47	15.09	15.3	15.24	15.95	14.5	14.73	14.6	14.5	14.89	
Ionosphere	10.48	12.46	12.8	14.11	13.63	11.8	11.56	11.1	11.5	12.61	
balance	10.37	14.72	13.8	13.51	13.58	13.2	13.1	14	11	11.33	
Iris	14.91	19.12	16.4	14.52	10.29	11	10.98	11.9	12.2	11.03	
Wine	13.52	15.11	14.9	14.3	11.73	12.2	11.65	13	11.6	12.17	
Vowel	11.24	16.05	16	14.47	14.27	14.8	14.18	13.8	11.4	12.52	
Control	12.72	15.86	14.6	14.95	14.48	15.4	14.66	14.2	14.9	15.81	

Table 8 Depicts the execution time of clustering algorithms (in seconds)

Dataset	Clustering algorithm									
	VS	MBOA	WOA	ICSO	TLBO	CS	GSA	LION	GWO	Proposed IWFO
CMC	14.43	13.94	16	18.96	15.61	13.6	12.38	14.3	16.1	14.23
Glass	13.78	11.22	18.8	18.43	14.72	15.9	13.63	13.8	14.7	16.11
crude	12.02	14.86	14.2	13.05	12.96	13.7	13.05	12.7	15	12.09
Liver	13.74	14.01	13.6	12.86	12.56	14.8	14.21	13.9	13.4	13.01
Cancer	14.01	12.52	14.1	13.8	13.48	14	13.29	14.2	15.6	13.83
Thyroid	15.63	13.74	14.9	13.74	14.64	17	14.38	15.1	14.2	14.89
Ionosphere	12.63	11.89	11.3	11.97	12.02	10.3	11.04	10.9	11.1	12.61
balance	11.32	12.08	12.3	14.16	11.21	12.8	11.93	11.2	10.6	11.33
Iris	11.05	11.16	11.8	10.94	10.46	11	11.78	10.3	10.4	11.03
Wine	13.59	12.27	13.2	13.48	10.59	11.3	12.08	11.6	12.1	12.17
Vowel	12.11	9.78	12.7	12.29	12.17	13.7	12.47	11.5	11.8	12.52
Control	15.44	14.04	13.8	14.73	15.01	13.6	15.89	14.6	14	15.81

nar probability (pl) with a $rand()$ function such as $If (rand(.) < pl)$. If the condition is true, then water flow is defined as laminar flow, and the position of water particles is updated using the laminar flow mechanism. The process of the laminar flow is highlighted in Algorithm 1.

Algorithm 1 Steps for the laminar flow phase

1. Determine the laminar flow direction based on equation 15.
2. Compute the new position of water particles using the equation 14. Check boundary constraints, if these are not satisfied, generate the new position using equation 7.
3. Compute the potential energy ($f(C_k(i))$) of water particles (C_k) using the equation 17.

- (iv) Turbulent flow: If water flow is not defined as laminar, then it can be turbulent and the position of water particles is updated using the turbulent flow mechanism. This mechanism is mentioned in Algorithm 2.

Algorithm 2 Steps for the turbulent flow phase

-
1. Determine the trial position of water particles (C_k) in random order.
 2. Randomly choose the jostling operator (p) using equation 4 such that $k \neq j$.
 3. Compute the new position of water particles using the equation 3. Check boundary constraints, if these are not satisfied, generate the new position using equation 7.
 4. Compute the potential energy ($f(C_k(i))$) of water particles (C_k) using the equation 17.
-

Table 9 Average ranking of each algorithm based on execution time using all datasets

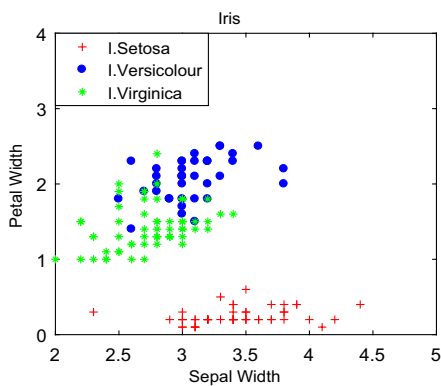
Algorithm	Average time	Average ranking
KM	12.43	1
DE	15.77	19
ACO	15.11	18
PSO	14.26	17
ABC	13.8	13
GA	13.83	14
BB-BC	13.76	12
BAT	13.5	11
VS	13.31	9
MBOA	12.63	2
WOA	13.89	15
ICSO	14.03	16
TLBO	12.95	5
CS	13.47	10
GSA	13.01	6
LION	12.85	4
GWO	13.25	8
WFO	12.72	3
Proposed IWFO	13.3	7

- (v) Evolving and Updation: This step corresponds to determining the final updated position of water particles (cluster centroids) after laminar and turbulent mechanisms. To determine the final position of water particles, the potential energy of new particles is compared with the previous potential energy of particles. If $(f(x_{t,new}(i)) > f(x_{t,old}(i)))$ is higher, then the new centroid is as $x_t(i+1) \leftarrow x_{t,new}(i)$, otherwise $x_t(i+1) \leftarrow x_{t,old}(i)$. This step also includes a limit operator for alleviating the local optima problem and it is set to 5. If, potential energy ($f(C_k(i))$) is not improved in five successive iterations, it is assumed that the algorithm sticks in local optima and the neighbourhood search procedure is called to compute the new position of the water particles. If the termination condition is not met, then steps (ii)-(v) are repeated. Otherwise, obtain the optimal position of water particles (optimal cluster centroids). The algorithmic steps of the proposed IWFO algorithm are listed in Algorithm 3, while the flowchart of the IWFO-based clustering algorithm is depicted in Fig. 1.

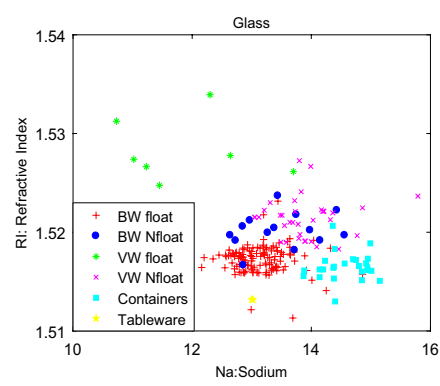
Algorithm 3 Proposed IWFO algorithm for cluster analysis

1. Upload the dataset and set the user-defined parameters of the proposed IWFO algorithm.
 2. Choose the initial population of the water particles (C_k) using the equation 9.
 3. While ($i_{curr} \leq i_{max}$), do following
 4. Compute the objective function (Euclidean distance) using the equation 22 and arrange the data (X_i) into different clusters (C_k) based on the minimum value of Euclidean distance.
 5. Compute the potential energy (E) for each water particle(C_k) using the equation 17.
 6. Compute the laminar probability (pl) and turbulent operator (pe) using the equations 12-13.
 7. If ($rand() < pl$)
 8. Call the algorithm 2 (Laminar Flow Phase)
 9. Else
 10. Call the algorithm 2 (Turbulent Flow Phase)
 11. For each water particle (C_k), do following
 12. If ($f(C_{k,new}) > f(C_{k,old})$)
 13. $f(C_k) \leftarrow f(C_{k,new})$
 14. $C_k \leftarrow C_{k,new}$
 15. Else
 16. $C_k \leftarrow C_{k,old}$
 17. Compute the global best position(g_{best}) and store in the list in increasing order of potential energy.
 18. If($f(C_k)$) is not improved up to limit parameter *local optima issue*/
 19. Call Multi-Strategy Search Process
 20. If($f(C_k) < 0.4$)
 21. Invoke chaotic local search strategy using equation 18.
 22. Else If($0.4 \leq f(C_k) < 0.7$)
 23. Invoke mutation-based strategy using equation 20.
 24. Else
 25. Invoke crossover-based strategy using equation 21.
 26. End if
 27. If maximum iteration is not reached, repeat the steps 4-27.
 28. Otherwise, obtain optimal centroids (C_k).
-

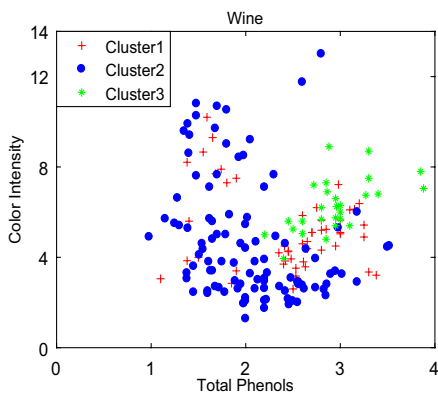
Fig. 2 Demonstrates the different clusters of data in a variety of datasets using the proposed IWFO algorithm



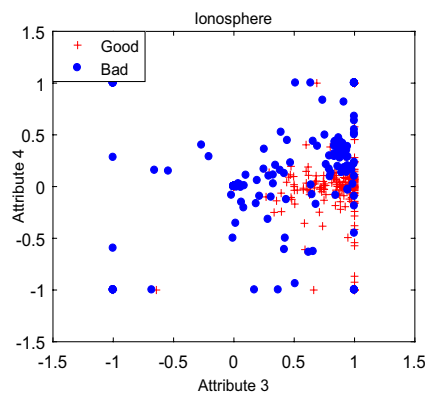
(a): Iris Dataset



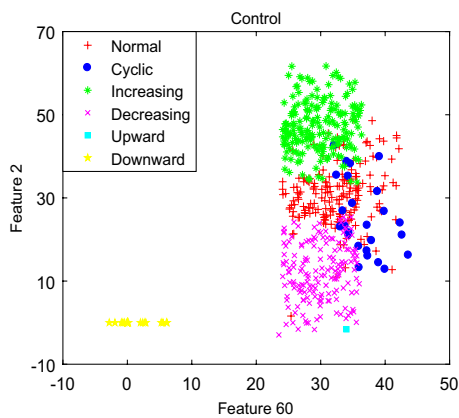
(b): Glass Dataset



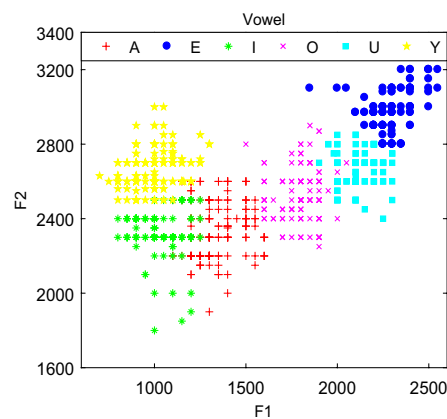
(c): Wine Dataset



(d): Ionosphere Dataset

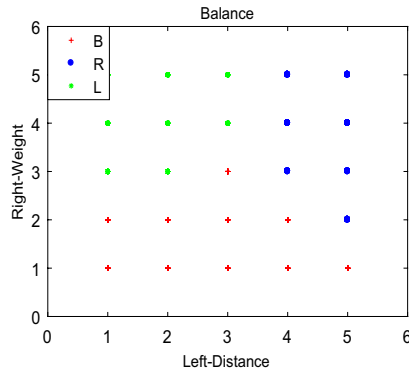


(e): Control Dataset

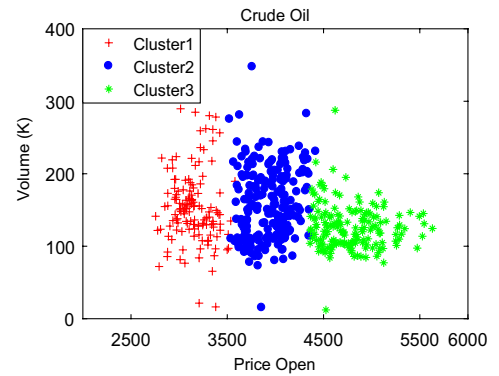


(f): Vowel Dataset

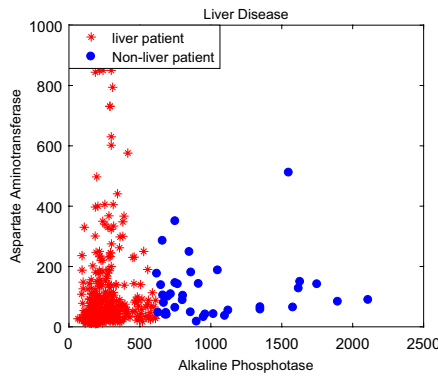
Fig. 2 (continued)



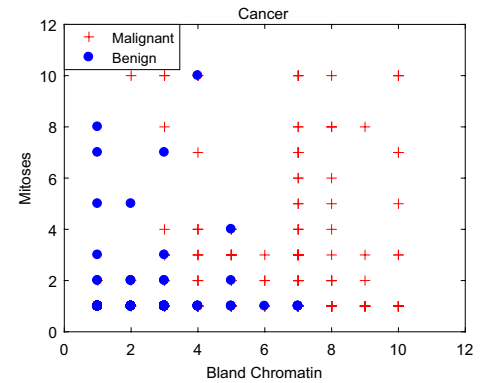
(g): Balance Dataset



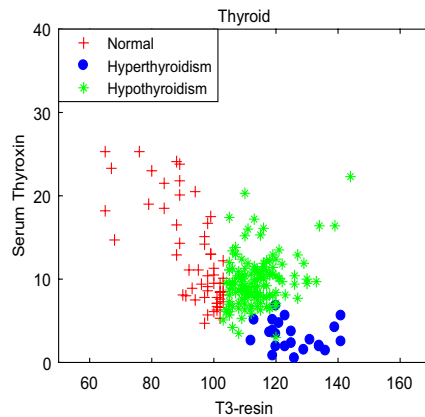
(h): Crude oil Dataset



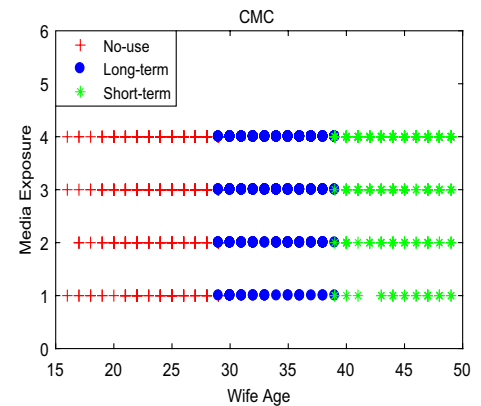
(i): LD Dataset



(j): Cancer Dataset



(k): Thyroid Dataset



(l): CMC Dataset

Experimental Results

This section presents the simulation result of the proposed IWFO algorithm and other algorithms. The efficacy of the IWFO algorithm is examined using twelve standard clustering datasets. These datasets are downloaded from the UCI repository and the description of these datasets is mentioned in Table 2. The simulation results of the proposed IWFO

algorithm are compared with a wide variety of meta-heuristic algorithms. These algorithms are described as K-means [67], PSO [21], ACO [68], ABC [69], DE [70], GA [71], BB-BC [72], BAT [73], VS [74], MBOA [75], WOA [46], ICSO [76], TLBO [77], CS [78], GSA [79], LION [80], WWO[82] and GWO [52]. The simulation results are evaluated using intra, SD, rank, AR and DR parameters. The results indicate an average of thirty separate runs. The intra-parameter can be defined as the sum of the distance between

Table 10 Statistical results of the Wilcoxon rank test using proposed IWFO and other clustering algorithms based on accuracy parameter

Algorithm	Sum	Mean	Median	Z value	p-value	Significance level	Proposed IWFO			H0
							Sum	Mean	Median	
GWO	8.899	0.740	0.770	2.8829	0.0039	0.05	9.608	0.800	0.798	Reject
Lion	9.138	0.760	0.779	2.8045	0.0050					Reject
GSA	8.704	0.730	0.740	3.0398	0.0024					Reject
CS	9.007	0.750	0.719	2.9614	0.0031					Reject
TLBO	8.579	0.710	0.720	2.5691	0.0102					Reject
ICSO	8.520	0.710	0.702	2.6476	0.0081					Reject
WOA	7.853	0.650	0.628	3.0398	0.0024					Reject
MBOA	7.836	0.650	0.624	2.8829	0.0039					Reject
VS	7.730	0.640	0.628	3.0398	0.0024					Reject
WFO	8.862	0.740	0.731	3.0600	0.0022					Reject
BAT	8.203	0.680	0.663	3.0398	0.0024					Reject
BB-BC	7.713	0.640	0.637	3.0398	0.0024					Reject
GA	7.404	0.620	0.632	3.0398	0.0024					Reject
DE	7.469	0.620	0.656	3.0398	0.0024					Reject
ABC	7.625	0.640	0.644	3.0398	0.0024					Reject
ACO	7.316	0.610	0.628	3.0398	0.0024					Reject
PSO	8.022	0.670	0.700	3.0600	0.0022					Reject
KM	7.723	0.640	0.664	3.0398	0.0024					Reject

data objects and respective cluster centres. This parameter indicates the closeness between data objects and cluster centres. Standard deviation (SD) is also computed for validating the intra-results. Further, AR denotes the accuracy rate while DR denotes the detection rate. The maximum iteration is set to 100. The parameter settings of the other algorithms except the proposed IWFO are recommended the same as reported in the corresponding literature. The parameters setting of the proposed IWFO are given as population size (P) is defined in terms of clusters (K), laminar probability (P_{la}) $\in (0.2, 0.5)$, eddying probability (P_e) $\in (0.5, 0.9)$, limit operator ($limit_{op}$) = 5, maximum iteration (max_iter) is set to 100. The experiment is conducted in a Matlab environment using a Core i5-based processor with 32 GB of RAM.

Simulation Results

This subsection discusses the simulation results of the proposed IWFO and other clustering algorithms using intra-cluster distance (intra), standard deviation (SD), rank, accuracy (AR) and detection rate (DR) parameters. Table 3 presents the simulation results of the proposed IWFO and other standard existing clustering algorithms based on intra, SD and rank parameters. The simulation results of the proposed algorithm are compared with the original WFO, K-means, PSO, ACO, ABC, DE, GA, BB-BC, and BAT algorithms. A variety of datasets are considered for evaluating the efficacy of the proposed IWFO and other clustering algorithms. It is seen that the proposed IWFO algorithm

obtains the minimum value of intra-parameter with most of the datasets. The proposed algorithm achieves minimum intra with iris ($9.21E+01$), glass ($2.03E+02$), ionosphere ($9.04E+02$), vowel ($1.56E+05$), balance ($5.78E+04$), cancer ($1.23E+03$), and thyroid ($1.27E+03$) except wine, control, crude oil, CMC and LD datasets. For the wine dataset, the DE algorithm obtains the minimum intra-value ($1.58E+04$) while the proposed IWFO algorithm obtains the second minimum intra-value ($1.59E+04$) among all algorithms. It is also seen that the BB-BC algorithm gets minimum intra values ($2.38E+04$ and $3.13E+03$) for the control and LD dataset, while the proposed IWFO algorithm achieves ($2.41E+04$ and $2.85E+03$) minimum intra values for both datasets. For the crude oil dataset, the ACO algorithm achieves minimum intra value ($2.47E+02$), whereas the proposed IWFO algorithm obtains ($2.63E+02$) value for intra parameter. For the CMC dataset, the K-means algorithm achieves minimum intra-cluster distance ($5.59E+03$), while the proposed IWFO algorithm obtains the second minimum intra-distance rate i.e. ($5.64E+03$) which is the second minimum among all algorithms. It is also noticed that for the LD dataset, the BB-BC algorithm achieves minimum intra-cluster distance ($1.13E+03$), while the proposed IWFO algorithm obtains second minimum intra-distance rate i.e. ($1.23E+03$).

SD is also an important parameter that can verify the performance of the proposed IWFO algorithm in successive iterations. By analyzing the SD parameter, it is revealed that the proposed IWFO algorithm obtains a minimum SD rate for most datasets compared to other algorithms.

Table 11 Illustrates the average ranking of each algorithm based on accuracy parameter

	KM	PSO	ACO	ABC	DE	GA	BB-BC	BAT	WFO	VS	MBOA	WOA	ICSO	TLBO	CS	GSA	LION	GWO	Proposed IWFO
	12.83	10.17	14.75	13.5	13.79	16.04	12.83	10.25	6.25	14.04	12.33	12.92	8.08	7.42	5.17	7.38	4.54	6.13	1.58

Table 12 Friedman test statistical results using intra-cluster distance (intra) parameter

Statistical value	p-value	DF	critical value	Hypothesis
114.6065	4.339e-16	18	28.8693	Rejected

The proposed IWFO algorithm gets minimum SD rate with wine (4.34E+01), ionosphere (1.53E+01), vowel (1.41E+01), balance (3.34E+02), crude oil (1.08E+01), CMC (2.14E+01), LD (1.27E+01), cancer (1.46E+02), and thyroid (1.12E+01). For the iris dataset, the ACO algorithm obtains a minimum SD rate (1.31E+00), while the proposed IWFO algorithm obtains a (3.20E+00) SD rate. For the glass dataset, the GA algorithm gets a minimum SD rate (4.14E+00) while the proposed IWFO algorithm achieves a (9.98E+00) SD rate. For the control dataset, the BB-BC algorithm achieves a minimum SD rate (1.09E+02), while the proposed algorithm obtains a (1.42E+02) SD rate. It is analyzed that the proposed algorithm achieves comparable SD results with the aforementioned datasets. Further, a rank parameter is also used to describe the ranking of each algorithm and this parameter is devised based on the minimum intra parameter. By analyzing the rank parameter, it is revealed that the proposed IWFO algorithm obtains the first rank with eight datasets (iris, glass, ionosphere, vowel, CMC, cancer, thyroid). For wine, control and LD datasets, the proposed algorithm obtains the second rank. For the crude oil dataset, the proposed algorithm gets third rank. The average ranking of the proposed IWFO algorithm is 1.5 using all datasets, while, the DE algorithm exhibits the worst rank (7.83) among all algorithms. It is also noticed that the average rank of the original WFO algorithm is 3.83. Hence, it is stated that the proposed algorithm obtains superior clustering results with most of the datasets.

Further, the performance of the proposed IWFO algorithm is also compared with the recently developed meta-heuristic algorithms. These algorithms are VS, MBOA, WOA, ICSO, TLBO, CS, GSA, LION and GWO. The results are evaluated using intra, SD and rank parameters. The intra-parameter is utilized to compute the compactness among the data objects within clusters. Table 4 presents the simulation results of the proposed IWFO and other algorithms based on the intra, SD and rank parameters. It is noticed that the proposed IWFO algorithm gets the minimum value of intra parameter with most of the datasets except vowel, balance, crude oil and thyroid datasets. The proposed IWFO algorithm obtains minimum intra values with iris (9.21E+01), wine (1.59E+04), glass (2.03E+02), ionosphere (9.04E+02), balance (5.78E+04), CMC (5.64E+03), cancer(1.23E+03), and LD (2.85E+03) except wine, control, crude oil and LD datasets. For vowel and thyroid datasets, the ICSO algorithm obtains minimum intra values (1.55E+05 and 9.90E+02)

while the proposed IWFO algorithm obtains (1.56E+05 and 1.27E+03) intra values. It is also seen that the TLBO algorithm gets minimum intra values (5.36E+04 and 2.59E+02) for balance and crude oil datasets, while the proposed IWFO algorithm achieves (5.78E+04 and 2.63E+02) intra values for both datasets. SD is also a significant parameter that can verify the performance of the proposed IWFO algorithm in successive iterations. By analyzing the SD parameter, it is revealed that the proposed IWFO algorithm obtains a minimum SD rate for most of the datasets except iris, wine, control and balance compared to other algorithms. The proposed IWFO algorithm gets a minimum SD rate with glass (9.98E+00), ionosphere (1.53E+01), vowel (1.41E+01), crude oil (1.08E+01), CMC (2.14E+01), LD (1.27E+01), cancer (1.46E+02), and thyroid (1.12E+01). For the iris dataset, the WOA algorithm obtains a minimum SD rate (1.06E+00), while the proposed IWFO algorithm obtains a (3.20E+00) SD rate. For the wine dataset, the TLBO algorithm gets a minimum SD rate (2.94E+01) while the proposed IWFO algorithm achieves a (4.34E+01) SD rate. For control and balance datasets, the VS algorithm achieves minimum SD rates (1.17E+02 and 1.04E+02), while the proposed algorithm obtains (1.42E+02 and 3.34E+02) SD rates. For the crude oil dataset, the ABC algorithm gets a minimum SD rate (1.09E+01), while the proposed IWFO algorithm obtains a (3.80E+01) SD rate. It is analyzed that the proposed algorithm achieves comparable SD results with the aforementioned datasets. Further, a rank parameter is also used to describe the ranking of each algorithm and this parameter is devised based on the minimum intra parameter. By analyzing the rank parameter, it is revealed that the proposed IWFO algorithm obtains the first rank with most of the datasets except vowel, CMC, balance, crude oil and thyroid datasets. The proposed algorithm obtains the second rank with vowel, crude oil and thyroid datasets. For CMC and balance datasets, the proposed algorithm obtains the third rank among all algorithms being compared. Hence, it is stated that the proposed algorithm obtains better clustering results with most of the datasets based on intra, SD and rank parameters.

The efficacy of the proposed IWFO algorithm is also assessed using AR and DR parameters. The simulation results of the proposed IWFO and other standard existing algorithms including the original WFO algorithm are reported in Table 5. It is seen that the proposed IWFO algorithm gets superior clustering results based on AR parameter compared to KM, PSO, ACO, ABC, DE, GA, BB-BC and BAT algorithms. The proposed IWFO algorithm obtains better AR results with most of the datasets such as iris (0.961), glass (0.717), wine (0.873), ionosphere (0.789), control

(0.801), vowel (0.878), balance (0.913), crude oil (0.796), CMC (0.613), LD (0.685), cancer (0.836), and thyroid (0.738). DR is also described as one of the potential parameters for analyzing the performance of the clustering algorithms. By analyzing the DR parameter, it is stated that the proposed IWFO algorithm gets superior results with most of datasets such as iris (0.968), glass (0.743), wine (0.896), ionosphere (0.804), control (0.819), vowel (0.904), balance (0.932), crude oil (0.823), CMC (0.646), LD (0.746), cancer (0.868), and thyroid (0.786). It is also observed that the proposed IWFO algorithm obtains better AR and DR rates as compared to the original WFO algorithm. Hence, it is stated that the proposed IWFO algorithm obtains better AR and DR results than similar classes of algorithms.

Moreover, the performance of the proposed IWFO algorithm is also compared with several recent clustering algorithms reported in the literature. The simulation results of the proposed IWFO and other algorithms based on AR and DR parameters are presented in Table 6. It is revealed that the proposed IWFO algorithm obtains better AR results with most of the datasets except wine, balance, control and cancer. The proposed algorithm achieves a higher accuracy rate with iris (0.961), glass (0.717), wine (0.873), ionosphere (0.789), control (0.801), vowel (0.878), balance (0.913), crude oil (0.796), CMC (0.613), LD (0.685), cancer (0.836), and thyroid (0.738). For the wine dataset, the LION algorithm obtains a higher AR rate (0.878) among all algorithms, while the proposed IWFO gets (0.873) AR rate. It is also noticed that the GWO algorithm obtains a higher AR rate (0.814) for the control dataset, whereas, the proposed algorithm obtains a (0.801) AR rate. For the balanced dataset, the CS algorithm achieves a superior AR rate (0.917), while the proposed algorithm gets (0.913) AR results. Furthermore, the TLBO algorithm achieves a better AR rate (0.921) for the cancer dataset, while the proposed algorithm obtains (0.836) AR rates. It is also analyzed that the proposed algorithm gets comparable AR results for the aforementioned datasets. By analyzing the DR parameter, it is said that the proposed IWFO algorithm obtains better DR results with most of the datasets except wine, balance, control and cancer. The proposed algorithm achieves higher DR results with iris (0.968), glass (0.743), wine (0.896), ionosphere (0.804), control (0.819), vowel (0.904), balance (0.932), crude oil (0.823), CMC (0.646), LD (0.746), cancer (0.868), and thyroid (0.786). For the wine dataset, the LION algorithm obtains a higher DR rate (0.908) among all algorithms, while the proposed IWFO gets a (0.896) DR rate. It is also noticed that the GWO algorithm obtains a higher DR rate (0.833) for the control dataset, whereas, the proposed algorithm obtains (0.819) DR rate. For the balance dataset, the CS algorithm

achieves a superior DR rate (0.938), while the proposed algorithm gets (0.932) DR results. Furthermore, the TLBO algorithm achieves a better DR rate (0.952) for the cancer dataset, while the proposed algorithm obtains (0.868) DR rates. Finally, it is stated that the proposed IWFO algorithm gets superior clustering with most datasets in terms of intra, SD, rank, AR and DR parameters.

Apart from the well-known clustering measure, several researchers also adopt the execution time for comparing the performance of the clustering algorithms. This work also compares the performances of the proposed IWFO and other clustering algorithms using the execution time parameters. Tables 7, 8 and 9 illustrate the execution time of each clustering algorithm using different datasets. The execution time is measured in seconds. Tables 7 and 8 present the execution time of the proposed IWFO and other clustering algorithms. It is analyzed that the proposed IWFO has a competitive execution time compared to other algorithms. Table 9 depicts the average execution time of each technique. It is found that the KM algorithm obtains minimum average execution times using all datasets and its rank is 1. The average ranking of the proposed IWFO algorithm is seven in the context of execution time. It is also said that the proposed IWFO has a better average execution time compared to most of the clustering algorithms.

Figure 2 illustrates the grouping of the data into different clusters based on the proposed IWFO clustering algorithm. This work considers the well-known clustering datasets such as iris, glass, wine, ionosphere, control, vowel, balance, crude oil, CMC, LD, cancer, and thyroid. The clustering results using optimal centroids are presented in Fig. 2. Hence, it is summarized that the proposed algorithm allocates the data to appropriate clusters more accurately.

Statistical Test Results

This subsection demonstrates the statistical results of the Wilcoxon signed-rank test and Friedman test. The significance of the statistical tests is to validate the performance of the newly proposed algorithm compared to existing algorithms. This work considers the Wilcoxon signed-rank test and the Friedman test to check the significant difference between the performances of the proposed IWFO and other existing clustering algorithms. Wilcoxon rank test is a non-parametric statistical test that can be used to check whether the samples belong to the same population or not. So, this test considers one pair of data and checks whether the mean rank is equal or not. In this work, the simulation results of the proposed IWFO are compared with eighteen clustering algorithms. So, as per the Wilcoxon rank test, eighteen pairs of data samples are designed to perform this test. Before conducting this test, two hypotheses are designed such as H_0 stands for no significant difference between the median/

mean values of data samples and H_1 stands for a significant difference between the mean/median values of the data samples. Further, the significance level is set to 0.05. The statistical results of the Wilcoxon rank test are presented in Table 10 using sum, mean, median, z-value, p-value and Hypothesis (H_0). It is analyzed that the p-value for each pair of the data samples is less than the z-value. Hence, the hypothesis (H_0) is rejected at the significance level of 0.05. So, it is concluded that a significant difference occurs between the performances of the proposed IWFO and other clustering algorithms.

The existence of the proposed IWFO algorithm is also validated using the Friedman statistical test. This test also requires two hypotheses such as H_0 and H_1 . H_0 claimed that the performances of the proposed IWFO and other clustering algorithms are similar. While, H_1 claimed that the performances of the proposed IWFO and other clustering algorithms are dissimilar. Firstly, this test computes the ranking of each algorithm using each dataset and further, an average ranking of each technique is computed. The Friedman test utilizes the Friedman statistics to obtain the average rank of each algorithm. Table 11 illustrates the average ranking of each algorithm using the accuracy parameter. It is seen that the proposed IWFO algorithm gets first rank (1.58) compared to other algorithms. Whereas, GA exhibits the worst rank (16.04) among all. It is noticed that the rank of the WFO algorithm is 6.25. So, it is claimed that the proposed IWFO is an outperformer and significantly different from other algorithms.

Moreover, Table 12 depicts the statistics of the Friedman test at a significance level of 0.05. The degree of freedom for this test is 18 and the critical value is 28.8693. Further, the statistical value for the Friedman test is 114.6065 and the p-value is $4.339e-16$. The p-value for the Friedman test is lower than the critical value. In turn, the null hypothesis (H_0) is rejected and the rejection of the null hypothesis stated that the performances of the proposed IWFO and other clustering algorithms are significantly dissimilar.

This work considers the two statistical tests (Wilcoxon signed-rank test and Friedman test) for validating the performance of the proposed IWFO in the clustering field. These tests aim to evaluate the performance of the proposed IWFO algorithm concerning other existing clustering algorithms and validate the proposed algorithm statistically. The findings of these tests stated that the proposed IWFO algorithm is significantly different from existing clustering algorithms. This existence is duly verified by the results of the aforementioned tests. Hence, it is said that the proposed IWFO algorithm is an effective algorithm for solving the clustering problems and it can be certified both experimentally and statistically in this work.

Conclusion

This work presents an improved WFO algorithm to handle the clustering problems. The source of inspiration for this algorithm is the flow of water. This work incorporates some improvements in the WFO to handle the shortcomings associated with it. These improvements are summarized as the initialization issue of the WFO is handled by a Gaussian-based chaotic map, nonlinear functions are considered for improving the balance between local and global searches, and the local optima issue is alleviated by a neighbourhood search mechanism. A set of twelve benchmark datasets is downloaded to examine the performance of the proposed IWFO algorithm. The results are evaluated using well-known clustering measures such as intra, rank, SD, AR and DR, and compared with popular clustering algorithms. The results showed that the proposed IWFO algorithm is superior to most of the clustering algorithms and also gets better results with most of the datasets. Further, two statistical tests are also adopted to validate the existence of the proposed IWFO algorithm

in the clustering field. The statistical results showed that the proposed IWFO algorithm is significantly different from other clustering algorithms. Hence, it is stated that both simulation and statistical results certify the efficacy of the IWFO algorithm in the clustering field. It can also be concluded that IWFO is one of the most efficient and robust clustering algorithms. The future perspective will explore the IWFO for spherical-shaped clusters, feature engineering, optimizing the weights, and multi-objective clustering.

Appendix 1: Comparative Illustrations of Simulation Results of the Proposed IWFO and Other Clustering Algorithms

Intra Cluster Distance Parameter (Intra)

See Fig. 3.

Intra Cluster Distance Parameter (Intra)

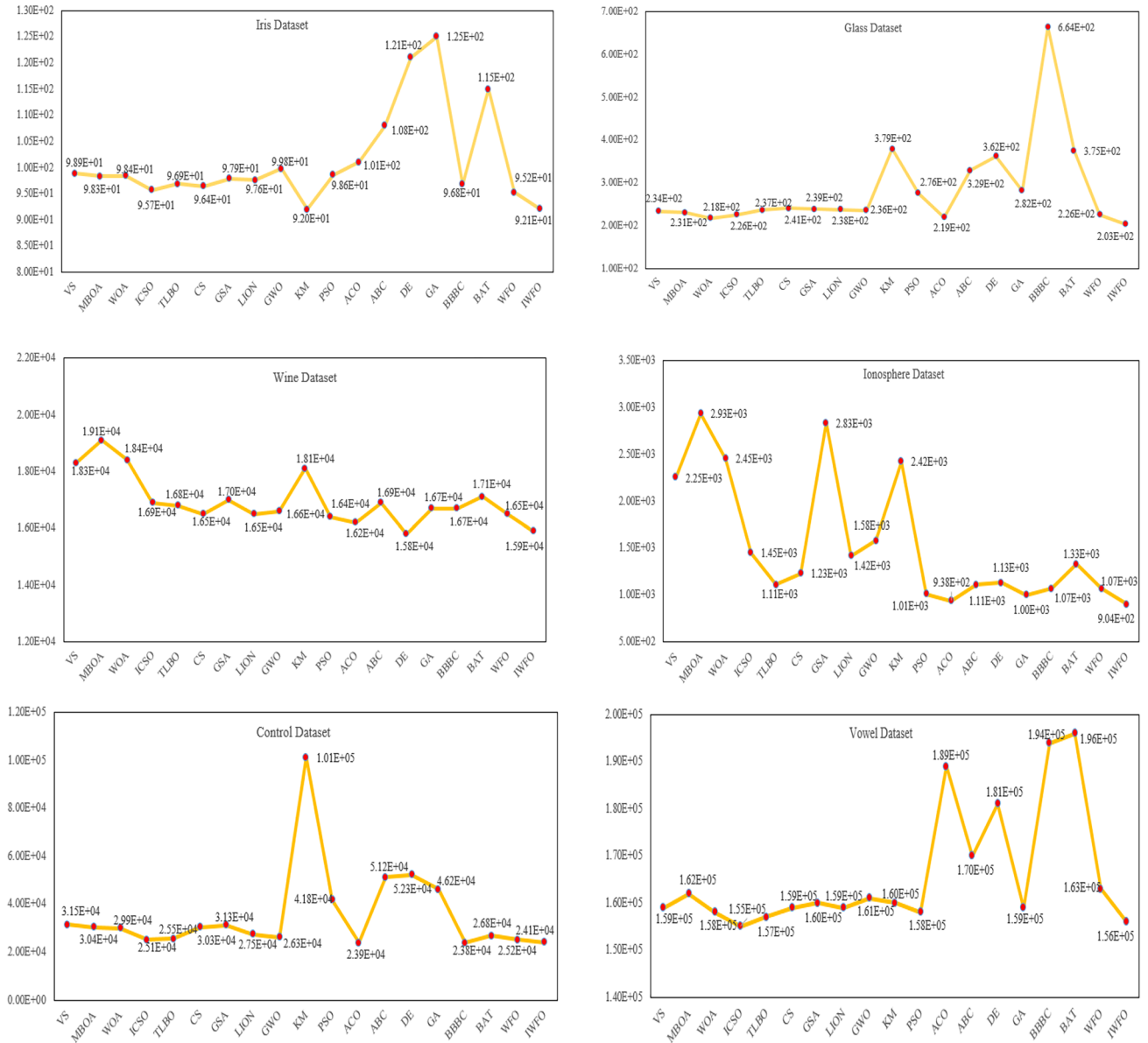


Fig. 3 Depicts the comparison of the intra cluster distance (Intra) parameter of the proposed IWFO and other clustering algorithms using benchmark clustering datasets



Fig. 3 (continued)

Accuracy Rate (AR) Parameter

See Fig. 4.

Accuracy Rate (AR) Parameter

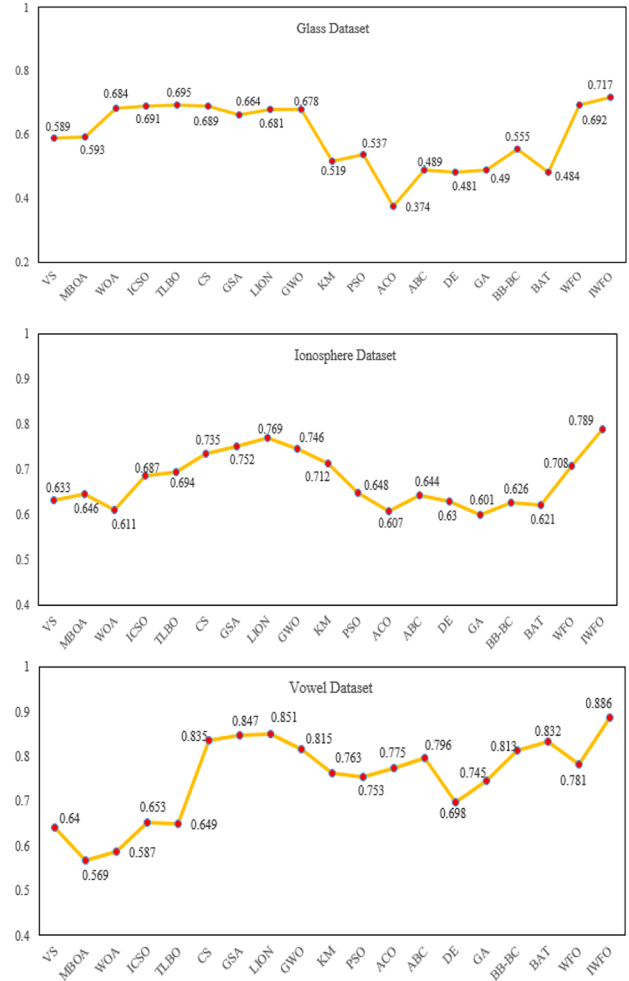
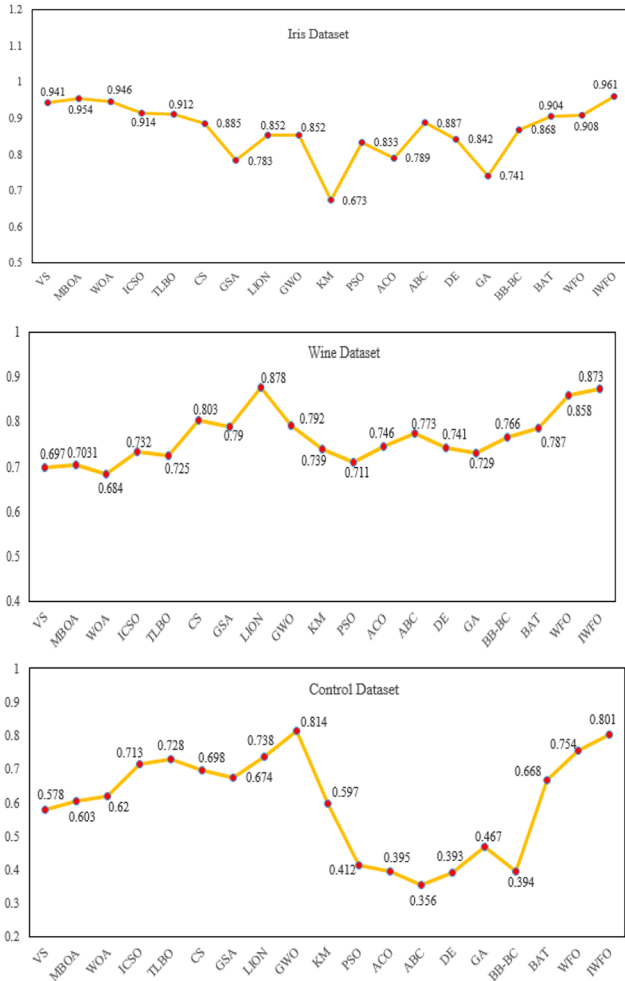


Fig. 4 Depicts the comparison of the accuracy rate (AR) parameter of the proposed IWFO and other clustering algorithms using benchmark clustering datasets

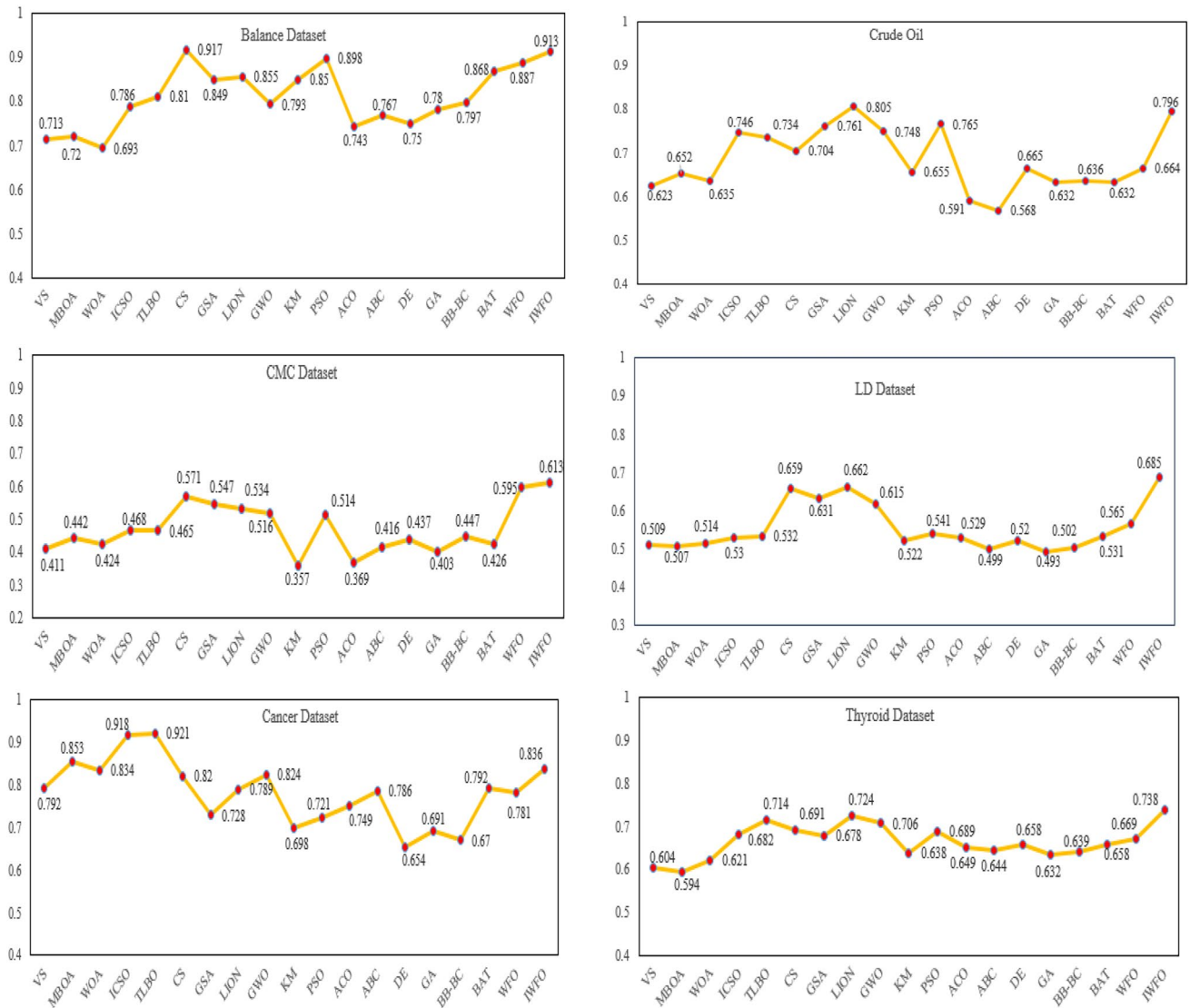


Fig. 4 (continued)

Detection Rate (DR) Parameter

See Fig. 5.

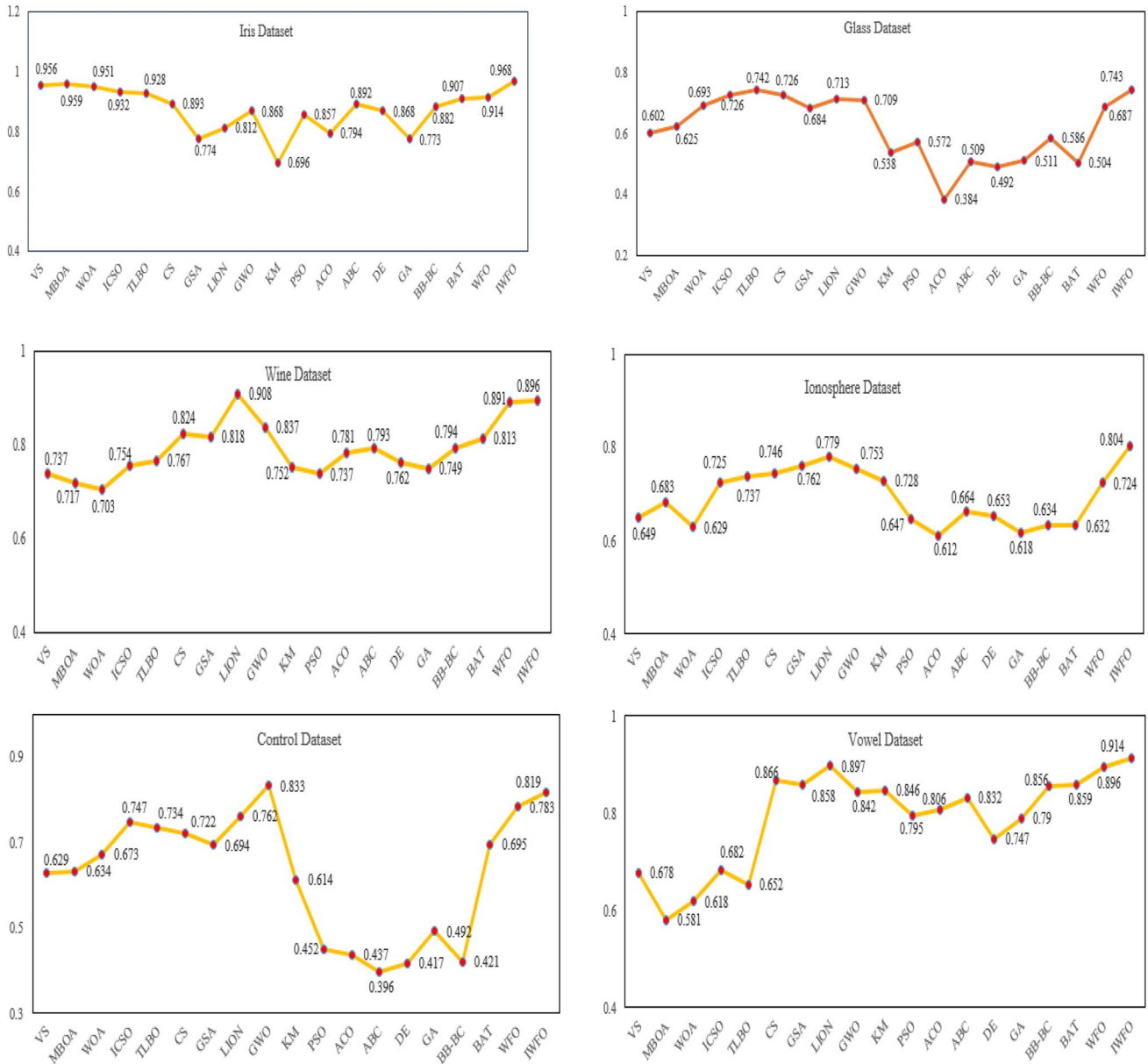


Fig. 5 Depicts the comparison of the detection rate (DR) parameter of the proposed IWFO and other clustering algorithms using benchmark clustering datasets

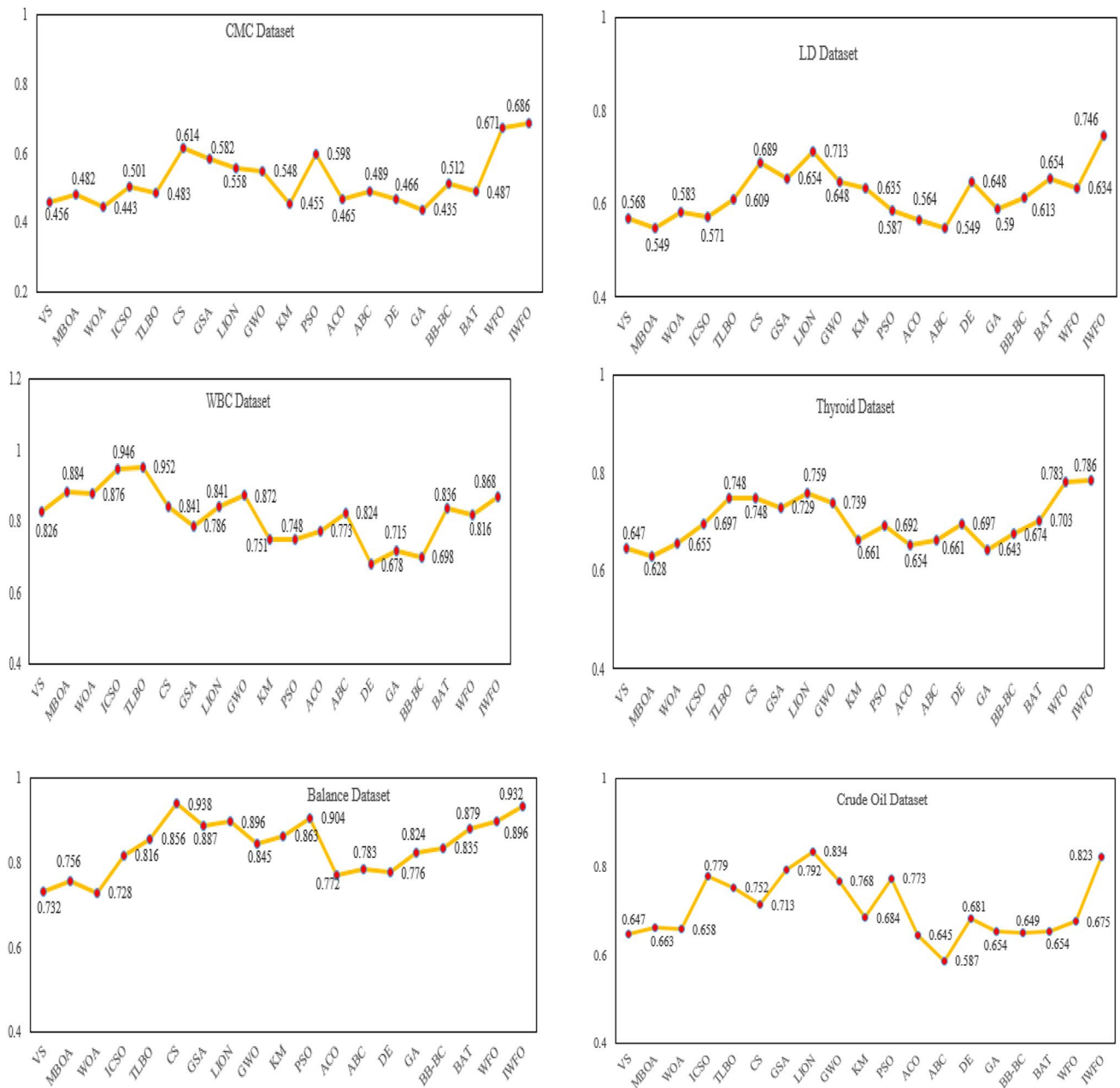


Fig. 5 (continued)

Author Contributions Prateek Thakral: Writing Original Draft, Software Implementation, Validation, Formal Analysis, Investigation; Yugal Kumar: Conceptualization, Methodology, Visualization, Data Curation, Review and Editing.

Funding No funding is received to perform this research work.

Data Availability The data is publicly available at UCI repository and can be available free of cost.

Declarations

Conflict of interest There is no competing of interests among authors.

Research involving human and/or animals Either human or animal are not involved in this research work.

Ethical and informed consent for data used The data is publicly available at UCI repository and can be available free of cost.

References

- Aggarwal CC, Reddy CK. Data clustering algorithms and applications. Londra: Chapman & Hall/CRC Data mining and Knowledge Discovery Series; 2014.
- Gan G, Ma C, Wu J. Data clustering: theory, algorithms, and applications. Society for Industrial and Applied Mathematics; 2020.
- Zhao Y, Karypis G. Data clustering in life sciences. *Mol Biotechnol.* 2005;31(1):55–80.
- Aggarwal CC, Reddy CK. An introduction to cluster analysis. 2013.
- G. V. P. S. D. a. S. D. Brock. CValid: an R package for cluster validation. *J Stat Softw.* 2008;25(4):1–22.
- M. Y. B. a. M. V. Halkidi. On clustering validation techniques. *J Intell Inf Syst.*
- Kaur A, Kumar Y. A new metaheuristic algorithm based on water wave optimization for data clustering. *Evol Intel.* 2022;15(1):759–83.
- Sahoo AJ, Kumar Y. Modified teacher learning based optimization method for data clustering. In: *Advances in signal processing and intelligent recognition systems.* Berlin: Springer International Publishing; 2014. p. 429–37.
- Kumar Y, Gupta S, Kumar D, Sahoo G. A clustering approach based on charged particles. In: *Optimization algorithms-methods and applications.* 2016. p. 245–63.
- Kaur A, Kumar Y. A multi-objective vibrating particle system algorithm for data clustering. *Pattern Anal Appl.* 2022;25(1):209–39.
- Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, Lin CT. A review of clustering techniques and developments. *Neurocomputing.* 2017;267:664–81.
- Özbakır L, Turna F. Clustering performance comparison of new generation meta-heuristic algorithms. *Knowl-Based Syst.* 2017;130:1–16.
- Patel KMA, Thakral P. The best clustering algorithms in data mining. In: *International Conference on Communication and Signal Processing (ICCSP).* 2016. p. 2042–6.
- Xu R, Wunsch DC II. BARTMAP: a viable structure for biclustering. *Neural Netw.* 2011;24(7):709–16.
- Jiang B, Pei J, Tao Y, Lin X. Clustering uncertain data based on probability distribution similarity. *IEEE Trans Knowl Data Eng.* 2011;25(4):751–63.
- Mukhopadhyay A, Maulik U, Bandyopadhyay S. A survey of multi objective evolutionary clustering. *ACM Comput Surv (CSUR).* 2015;47(4):1–46.
- Sevillano X, Alías F. A one-shot domain-independent robust multimedia clustering methodology based on hybrid multimodal fusion. *Multimed Tools Appl.* 2014;73(3):1507–43.
- Ezugwu AE, Ikotun AM, Oyelade OO, Abualigah L, Agushaka JO, Eke CI, Akinyelu AA. A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng Appl Artif Intell.* 2022;110: 104743.
- MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. 1967. p. 281–97.
- Jain AK, Murty MN, Flynn PJ. Data clustering: a review. *ACM Comput Surv (CSUR).* 1999;31(3):264–323.
- Cura T. A particle swarm optimization approach to clustering. *Expert Syst Appl.* 2012;39(1):1582–8.
- Jordehi AR. Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. *Appl Soft Comput.* 2015;26:401–17.
- Kushwaha N, Pant M, Kant S, Jain VK. Magnetic optimization algorithm for data clustering. *Pattern Recogn Lett.* 2018;115:59–65.
- Kumar Y, Sahoo G. A charged system search approach for data clustering. *Prog Artif Intell.* 2014;2(2):153–66.
- Hatamlou A. Black hole: a new heuristic optimization approach for data clustering. *Inf Sci.* 2013;222:175–84.
- Kaur A, Kumar Y. Neighborhood search based improved bat algorithm for data clustering. *Appl Intell.* 2022;52(9):10541–75.
- Kumar Y, Kaur A. Variants of bat algorithm for solving partitioned clustering problems. *Eng Comput.* 2021;1–27.
- Karaboga D. An idea based on honey bee swarm for numerical optimization, vol 200, p. 1–10. Technical report-tr06, Erciyes University, engineering faculty, computer engineering department. 2005.
- Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim.* 2007;39(3):459–71.
- Dorigo M, Birattari M, Stutzle T. Artificial ants as a computational intelligence technique. *IEEE Comput Intell Mag.* 2006;1(4):28–39.
- Erol OK, Eksin I. A new optimization method: big bang–big crunch. *Adv Eng Softw.* 2006;37(2):106–11.
- Ergezer M, Simon D, Du D. Oppositional biogeography-based optimization. In: *2009 IEEE international conference on systems, man and cybernetics.* IEEE; 2009. p. 1009–14.
- Osmani A, Mohasefi JB, Gharehchopogh FS. Sentiment classification using two effective optimization methods derived from the artificial bee colony optimization and imperialist competitive algorithm. *Comput J.* 2022;65(1):18–66.
- Bouyer A. An optimized k-harmonic means algorithm combined with modified particle swarm optimization and cuckoo search algorithm. *Found Comput Decis Sci.* 2016;41(2):99–121.
- Chuang LY, Hsiao CJ, Yang CH. Chaotic particle swarm optimization for data clustering. *Expert Syst Appl.* 2011;38(12):14555–63.
- Zhao M, Tang H, Guo J, Sun Y. Data clustering using particle swarm optimization. In: *Future information technology.* Berlin, Germany: Springer; 2014. p. 607–12.
- Liu F, Sun Y, Wang GG, Wu T. An artificial bee colony algorithm based on dynamic penalty and Lévy flight for constrained optimization problems. *Arab J Sci Eng.* 2018;43(12):7189–208.
- Du Z, Han D, Li KC. Improving the performance of feature selection and data clustering with novel global search and elite-guided artificial bee colony algorithm. *J Supercomput.* 2019;75:5189–226.
- Singh H, Kumar Y. Cellular automata based model for e-healthcare data analysis. *Int J Inf Syst Model Design (IJISMD).* 2019;10(3):1–18.
- Luo K. Water flow optimizer: a nature-inspired evolutionary algorithm for global optimization. *IEEE Trans Cybern.* 2021;52(8):7753–64.
- Matos Macêdo FJ, da Rocha Neto AR. A binary water flow optimizer applied to feature selection. In: *International Conference on Intelligent Data Engineering and Automated Learning.* Cham: Springer; 2022. p. 94–103.
- Said M, Shaheen AM, Ginidi AR, El-Sehiemy RA, Mahmoud K, Lehtonen M, Darwish MM. Estimating parameters of photovoltaic models using accurate turbulent flow of water optimizer. *Processes.* 2021;9(4):627.
- Cheng MM, Zhang J, Wang DG, Tan W, Yang J. A localization algorithm based on improved water flow optimizer and max-similarity path for 3D heterogeneous wireless sensor networks. *IEEE Sens J.* 2023.
- Qtaish A, Braik M, Albashish D, Alshammari MT, Alreshidi A, Alreshidi EJ. Optimization of K-means clustering method using hybrid capuchin search algorithm. *J Supercomput.* 2024;80(2):1728–87.

45. Kuo RJ, Hsu CC, Nguyen TPQ, Tsai CY. Hybrid multi-objective metaheuristic and possibilistic intuitionistic fuzzy c-means algorithms for cluster analysis. *Soft Comput*. 2024;28(2):991–1008.
46. Premkumar M, Sinha G, Ramasamy MD, Sahu S, Subramanyam CB, Sowmya R, et al. Augmented weighted K-means grey wolf optimizer: an enhanced metaheuristic algorithm for data clustering problems. *Sci Rep*. 2024;14(1):5434.
47. Demirci H, Yurtay N, Yurtay Y, Zaimoğlu EA. Electrical search algorithm: a new metaheuristic algorithm for clustering problem. *Arab J Sci Eng*. 2023;48(8):10153–72.
48. Harehchopogh FS, Khargoush AA. A chaotic-based interactive autodidactic school algorithm for data clustering problems and its application on COVID-19 disease detection. *Symmetry*. 2023;15(4):894.
49. Zorapacı E. Data clustering using leaders and followers optimization and differential evolution. *Appl Soft Comput*. 2023;132:109838.
50. Duan Y, Liu C, Li S, Guo X, Yang C. An automatic affinity propagation clustering based on improved equilibrium optimizer and t-SNE for high-dimensional data. *Inf Sci*. 2023;623:434–54.
51. Boroujeni SPH, Pashaei E. A hybrid chimp optimization algorithm and generalized normal distribution algorithm with opposition-based learning strategy for solving data clustering problems. 2023. arXiv preprint <http://arxiv.org/abs/2302.08623>.
52. Singh H, Rai V, Kumar N, Dadheech P, Kotecha K, Selvachandran G, Abraham A. An enhanced whale optimization algorithm for clustering. *Multimed Tools Appl*. 2023;82(3):4599–618.
53. Al-Behadili HNK. Improved firefly algorithm with variable neighborhood search for data clustering. *Baghdad Sci J*. 2022;19(2):0409–0409.
54. Besharatnia F, Talebpour A, Aliakbary S. An improved grey wolves optimization algorithm for dynamic community detection and data clustering. *Appl Artif Intell*. 2022;36(1):2012000.
55. Singh H, Kumar Y. An enhanced version of cat swarm optimization algorithm for cluster analysis. *Int J Appl Metah Comput (IJAMC)*. 2022;13(1):1–25.
56. Kushwaha N, Pant M, Sharma S. Electromagnetic optimization based clustering algorithm. *Expert Syst*. 2022;39(7): e12491.
57. Hashemi SE, Tavana M, Bakhshi M. A new particle swarm optimization algorithm for optimizing big data clustering. *SN Comput Sci*. 2022;3(4):1–16.
58. Zhu Q, Tang X, Elahi A. Automatic clustering based on dynamic parameters harmony search optimization algorithm. *Pattern Anal Appl*. 2022:1–17.
59. Kuo T, Wang KJ. A hybrid k-prototypes clustering approach with improved sine-cosine algorithm for mixed-data classification. *Comput Ind Eng*. 2022;169:108164.
60. Kaur A, Kumar Y. Neighborhood search based improved bat algorithm for data clustering. *Appl Intell*. 2022;52:10541–75.
61. Barshandeh S, Dana R, Eskandarian P. A learning automata-based hybrid MPA and JS algorithm for numerical optimization problems and its application on data clustering. *Knowl-Based Syst*. 2022;236: 107682.
62. Ikotun AM, Ezugwu AE. Improved SOSK-means automatic clustering algorithm with a three-part mutualism phase and random weighted reflection coefficient for high-dimensional datasets. *Appl Sci*. 2022;12(24):13019.
63. Mohammadi M, Mobarakeh MI. An integrated clustering algorithm based on firefly algorithm and self-organized neural network. *Prog Artif Intell*. 2022;11(3):207–17.
64. Suryanarayana G, Prakash KLNC, Mahesh PS, Bhaskar T. Novel dynamic k-modes clustering of categorical and non categorical dataset with optimized genetic algorithm based feature selection. *Multimed Tools Appl*. 2022;81(17):24399–418.
65. De Abreu LR, Araújo KAG, de Athayde Prata B, Nagano MS, Moccellini JV. A new variable neighbourhood search with a constraint programming search strategy for the open shop scheduling problem with operation repetitions. *Eng Optim*. 2022;54(9):1563–82.
66. Li W, Zhang Y, Sun Y, Wang W, Li M, Zhang W, Lin X. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Trans Knowl Data Eng*. 2019;32(8):1475–88.
67. Chowdhury K, Chaudhuri D, Pal AK. An entropy-based initialization method of K-means clustering on the optimal number of clusters. *Neural Comput Appl*. 2021;33(12):6965–82.
68. Jiang H, Yi S, Li J, Yang F, Hu X. Ant clustering algorithm with K-harmonic means clustering. *Expert Syst Appl*. 2010;37(12):8679–84.
69. Kumar Y, Sahoo G. A two-step artificial bee colony algorithm for clustering. *Neural Comput Appl*. 2017;28:537–51.
70. Kwedlo W. A clustering method combining differential evolution with the K-means algorithm. *Pattern Recogn Lett*. 2011;32(12):1613–21.
71. Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. *Pattern Recogn*. 2000;33(9):1455–65.
72. Bijari K, Zare H, Veisi H, Bobarshad H. Memory-enriched big bang–big crunch optimization algorithm for data clustering. *Neural Comput Appl*. 2018;29:111–21.
73. Senthilnath J, Kulkarni S, Benediktsson JA, Yang XS. A novel approach for multispectral satellite image classification based on the bat algorithm. *IEEE Geosci Remote Sens Lett*. 2016;13(4):599–603.
74. Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: vortex search algorithm. *Inf Sci*. 2015;293:125–45.
75. Wang G-G, Deb S, Cui Z. Monarch butterfly optimization. *Neural Comput Appl*. 2019;31(7):1995–2014.
76. Kumar Y, Singh PK. Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. *Appl Intell*. 2018;48(9):2681–97.
77. Kumar Y, Singh PK. A chaotic teaching learning based optimization algorithm for clustering problems. *Appl Intell*. 2019;49(3):1036–62.
78. Boushaki SI, Kamel N, Bendjehaba O. A new quantum chaotic cuckoo search algorithm for data clustering. *Expert Syst Appl*. 2018;96:358–72.
79. Han X, Quan L, Xiong X, Almeter M, Xiang J, Lan Y. A novel data clustering algorithm based on modified gravitational search algorithm. *Eng Appl Artif Intell*. 2017;61:1–7.
80. Chander S, Vijaya P, Dhyan P. Multi kernel and dynamic fractional lion optimization algorithm for data clustering. *Alex Eng J*. 2018;57(1):267–76.
81. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput*. 1997;1(1):67–82.
82. Sahoo RC, Kumar T, Tanwar P, et al. An efficient metaheuristic algorithm based on water flow optimizer for data clustering. *J Supercomput*. 2023. <https://doi.org/10.1007/s11227-023-05822-y>.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.