Neural Computing and Applications (2021) 33:6247–6306
https://doi.org/10.1007/s00521-020-05395-4

REVIEW

Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature

Absalom E. Ezugwu[1] · Amit K. Shukla[2] · Moyinoluwa B. Agbaje[1] · Olaide N. Oyelade[3] · Adán José-García[4] · Jeffery O. Agushaka[1]

Received: 5 August 2020 / Accepted: 24 September 2020 / Published online: 10 October 2020

Abstract
Cluster analysis is an essential tool in data mining. Several clustering algorithms have been proposed and implemented, most of which are able to find good quality clustering results. However, the majority of the traditional clustering algorithms, such as the K-means, K-medoids, and Chameleon, still depend on being provided a priori with the number of clusters and may struggle to deal with problems where the number of clusters is unknown. This lack of vital information may impose some additional computational burdens or requirements on the relevant clustering algorithms. In real-world data clustering analysis problems, the number of clusters in data objects cannot easily be preidentified and so determining the optimal amount of clusters for a dataset of high density and dimensionality is quite a difficult task. Therefore, sophisticated automatic clustering techniques are indispensable because of their flexibility and effectiveness. This paper presents a systematic taxonomical overview and bibliometric analysis of the trends and progress in nature-inspired metaheuristic clustering approaches from the early attempts in the 1990s until today's novel solutions. Finally, key issues with the formulation of metaheuristic algorithms as a clustering problem and major application areas are also covered in this paper.

Keywords Clustering algorithm · Automatic clustering · Taxonomy · Metaheuristic · Bibliometric analysis

1 Introduction

Data collection by any process, in any form and anyhow has, over the years, become a major necessity for the extraction of meaningful and tangible information across all domains, but especially in research, computer science, natural science, and engineering. However, without proper and definite analysis, these acquired data become meaningless and irrelevant. Because most of these data are in various arbitrary forms, their grouping might be difficult due to the lack of prior knowledge about the data object features. Without such grouping or classification, the data can become instances of unsupervised learning. Data clustering analysis, especially in the field of data mining, has over time played a vital role in organizing and classifying data appropriately. Simply put, data clustering or clustering analysis is a process that aims at grouping data into different and distinct classes, such that objects with similar attributes or characteristics are grouped in the same class called clusters. At the same time, dissimilar data points are separated into other, different clusters. These sets of clusters are generated in such a way that objects in one cluster are distinct and do not belong to other clusters. In other words, data clustering helps objects with similar characteristics or attributes to be meaningfully grouped together. Organizing the set of data points into their respective clusters should result in a high intra-cluster similarity and low inter-cluster similarity, as illustrated in Fig. 1.

Data clustering has been successfully applied and proven to be indispensable in diverse fields. Some areas in

✉ Absalom E. Ezugwu
ezugwua@ukzn.ac.za

[1] School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, KwaZulu-Natal, South Africa

[2] South Asian University, New Delhi 110021, India

[3] Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria

[4] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK

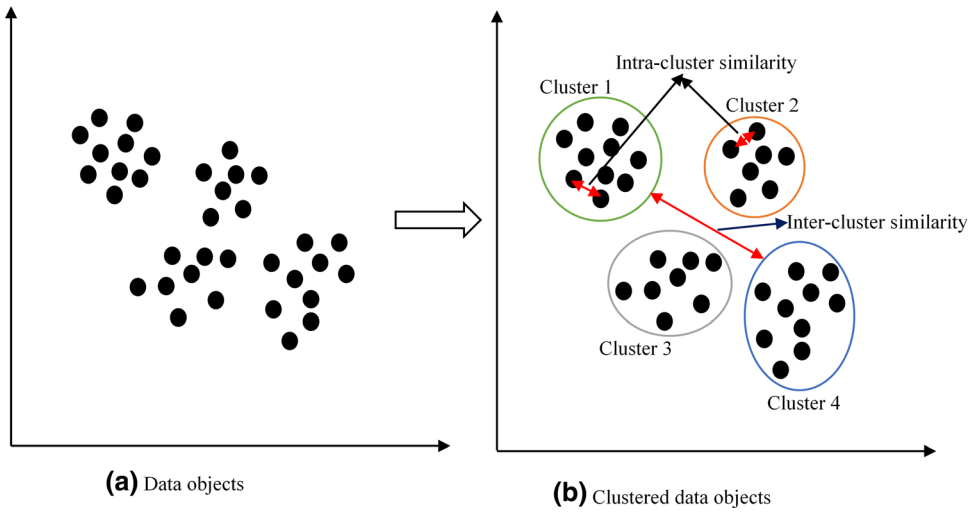

Fig. 1 Clustering example with intra- and inter-clustering illustrations

which clustering has found relevance are social network analysis, customer segmentation, market research, image segmentation, biological data analysis, data summarization, image retrieval, machine learning, data mining, and data analysis [4–10, 15, 82, 98, 118]. Jain [106] reviewed some of the principal methods of data clustering and also presented the evolution of and trends in data clustering that have been developed over the last few decades. According to Ramadas and Abraham [178], the process of clustering data occurs in seven different ways: specifically, data collection, initial screening of data, data representation, clustering tendency, clustering strategy, data validation, and clustering interpretation.

Data collection involves acquiring and gathering data from their various sources. Initial screening consists of the interchange of the different data that have been extracted from sources. The extracted data are then prepared and represented in such a way that they are made fit for a particular algorithm. Not all the extracted data would be useful; hence, the need to first verify whether or not there is a tendency for them to be clustered. On the one hand, if data or a group of data have been verified as suitable for being placed in a cluster, a clustering strategy chooses the right algorithm and parameters. After a particular algorithm or method has been selected, it is then used to examine and test the set of data manually. Finally, the resulting clustering solutions are interpreted, and further analyses are suggested and performed. On the other hand, some data are ambiguous, large, and complex, such that grouping or naming them is almost impossible. This is because the intrinsic natural properties are unknown or difficult to know, as found primarily in real-world problems where information regarding the data in view is unavailable. Due to their inability to handle the grouping of large datasets, and the limitations associated with clustering [13], the traditional method of clustering is exhausting and daunting. Therefore, automatic clustering techniques have emerged as a promising solution to these problems.

Automatic data clustering has the same outcome as the traditional clustering, with the advantage that we do not need to have any background information relating to the data objects in question. This then essentially accommodates real-world scenarios, which inevitably involve large and complex data requiring to be accurately partitioned into small clusters. Furthermore, real-life data sets are usually unlabeled, so the manual identification and classification of the data points is nearly impossible. With traditional clustering methods, the major problem has always been the difficulty in determining the optimal number of clusters and appropriate partitioning of the data objects. This problem of poor performance by traditional clustering approaches is linked to some degree to the inherent limitations in the algorithms. For example, traditional algorithms are mostly local search algorithms, and except for linear and convex optimization, they thus cannot guarantee global optimality, so results often depend on the initial starting points. Most traditional methods also tend to be problem-specific (for example, the k-mean algorithms, partitioning around medoids (PAM), Clustering for Large Applications (CLARA), and Cobweb clustering algorithm), and they struggle to cope with problems of discontinuity [13]. However, the recent shift toward automatic data clustering techniques implemented using nature-inspired metaheuristic approaches has helped to overcome these challenges in clustering analysis and has also offered several improvements in the methods of clustering [14, 75, 111].

Clustering methods can broadly be classified mainly into two categories of hierarchical or partitional clustering [178]. The hierarchical clustering algorithms are iterative-based clustering procedures, which generate outputs that are similar to a hierarchical tree or dendrogram that shows

a sequence of clustering with each of the clusters belonging to a partition of the data objects [38, 39]. By contrast, the partitional clustering algorithms start by decomposing the datasets into a set of disjoint clusters based on specific optimization criteria. Examples of the many data clustering algorithms that have been categorized and implemented over the years include the density-based algorithms [65], prototype-based algorithms [144], graph-based (hierarchical agglomerative clustering) methods [216], and hybrid algorithms [119]. Under the partitional clustering method, we find the well-known traditional clustering methods such as the k-means, fuzzy c-means, and simulated annealing; among which the k-means clustering algorithms and its variants dominate, due primarily to their implementation simplicity and the ease of adaption of natural methods for unsupervised learning. However, associated with the simplicity of these techniques are some drawbacks, which make them less scalable and robust. For example, as the number of features, attributes, and dimensionality of the data objects increases, most of the methods easily become entrapped into local optima [5, 8] and their effectiveness is highly dependent on the initial solution.

In recent times, several nature-inspired metaheuristic techniques (evolutionary algorithms, swarm intelligence, and stochastic, population-based algorithms) have been developed to mitigate some of the drawbacks of the traditional-based optimization methods. Among the well-known metaheuristic algorithms that have received much desired attention in the area of automatic clustering we note, in particular, the firefly algorithm (FA) [14], particle swarm optimization (PSO) [63], genetic algorithm (GA) [85], differential evolution (DE) [211], artificial bee colony (ABC) [115], symbiotic organisms search algorithm (SOS) [176], and teaching learning-based optimization (TLBO) [183], which have all played some significant role in different application domains [69–74] and have also become dominant problem-solving methods in other aspects of clustering analysis. These algorithms are referred to as nature-inspired metaheuristic algorithms because their development is inspired by concepts inherent in a natural occurring phenomenon. The metaheuristic techniques require some higher level of search procedures due to the trade-off balance between local search and randomization [79]. Since automatic data clustering is aimed at minimizing the similarity within a cluster and maximizing the dissimilarity between clusters, it has been classified as an optimization problem, and therefore, most metaheuristic approaches are judged to fit well into the context of the new clustering paradigm [129]. Our detailed review of the various proposed metaheuristic algorithms for clustering analysis is discussed in the subsequent sections of this paper. Figure 2 shows the taxonomy of clustering methods, starting from classical-based clustering methods to the most recent metaheuristic search-based clustering methods. It is interesting to note here that metaheuristic search algorithms are the most applied techniques, which are commonly used for the implementation of automatic clustering algorithms as identified in the study conducted by José-García and Gómez-Flores [111].

This paper presents an in-depth and systematic review of nature-inspired metaheuristic algorithms used for automatic clustering analysis. The focus of this paper is on the metaheuristic algorithms that have been employed to solve clustering problems over the last three decades. There have been some review studies on cluster analysis reported in the literature, some with an overwhelming citation by other researchers. For example, Nanda and Panda in 2014 presented a survey study on nature-inspired metaheuristic algorithms with a specific focus on partitional clustering methods. In his work, Jain [106] provided a brief overview of most of the well-known clustering methods, with the emphasis on the significant challenges and critical issues in designing clustering algorithms. Further, Jain also highlighted some of the emerging and useful research directions, which point to the idea of semi-supervised clustering, ensemble clustering, and simultaneous feature selection during data clustering, and large-scale data clustering, which to an extent are all trending and active clustering research areas. Jafar and Sivakumar [105], in their paper, gave a brief review on the application of biologically inspired data clustering technique with a focus on the ant-based clustering algorithms. Xu and Wunsch [233] presented a comprehensive survey of major clustering algorithms of data sets appearing in some research fields such as statistics, computer science, and machine learning. Berkhin [30] presented a more general overview of some clustering techniques, with his survey concentrating mainly on clustering algorithms from a data mining perspective. The survey study presented by Abbasi and Younis [1] focused on a general overview discussion of clustering algorithms, with application interests to wireless sensor networks, while Yu and Chong [237] gave a comprehensive survey study of clustering schemes for mobile ad hoc networks. A study that might be assumed to be more closely related to the current survey is the work of José-García and Gómez-Flores [111], whose research presented a review of sixty-five clustering methods based on nature-inspired metaheuristics that can be used for automatic clustering analysis. However, we strongly believe that the current study differs largely from these existing review works, because it is based on a 30-year detailed bibliometric analysis and an up-to-date systematic review of all nature-inspired metaheuristic algorithms for automatic data clustering problems.

Considering the considerable growth of interdisciplinary interests, and the dynamics in the application of clustering analysis to different research domains, it is obviously true to say that much has been achieved since the latest previous

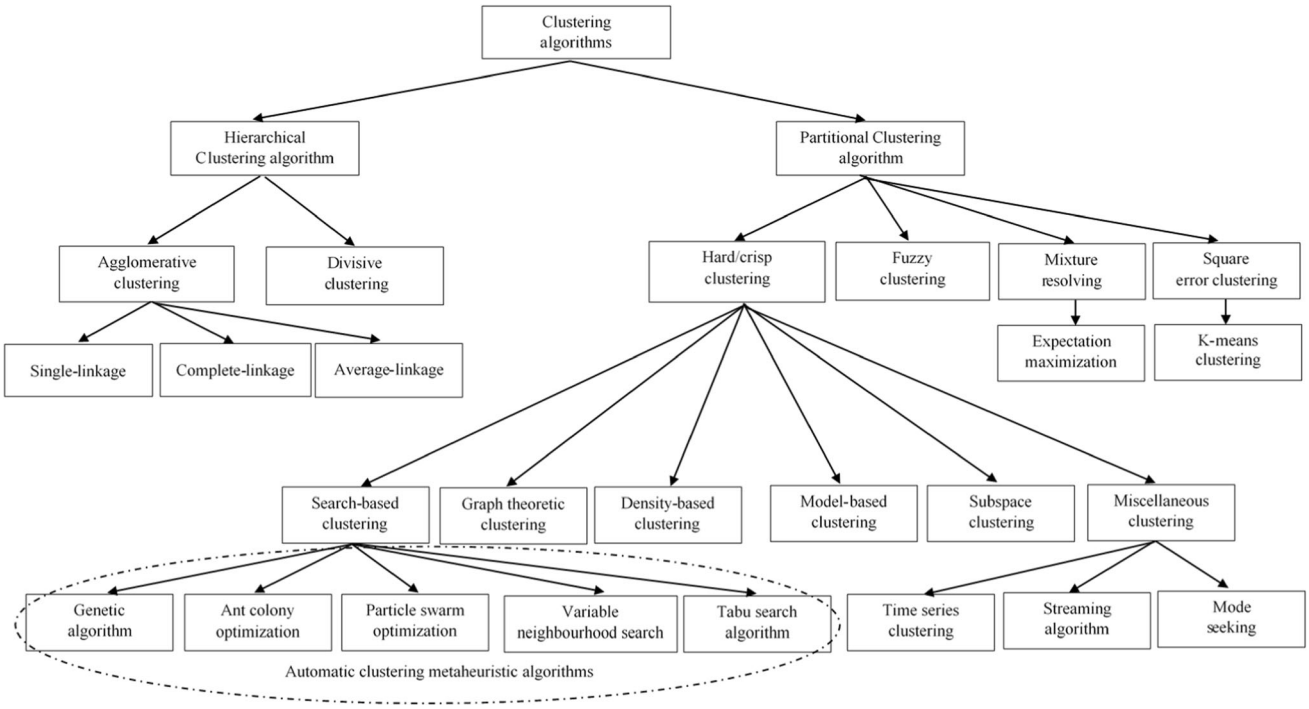

Fig. 2 Taxonomy of clustering algorithms

review of the literature by Jain in 2010. Therefore, so to speak, the main motivation of the current review is that, despite the broad base of research interest in the development and use of clustering algorithms, the available literature on the topic is remarkably segmented, which makes it extremely difficult for an applied researcher to become fully informed about the latest developments in this area [153]. Therefore, by keeping in mind the current research trends in the development of automatic clustering algorithms and their applications, the current study contributes in the following three ways: (1) a systematic review of all the nature-inspired metaheuristic algorithms used in both classical and automatic clustering methods, (2) an up-to-date bibliometric analysis of all clustering algorithms from the early 1990s to date, and (3) exploration of all the recent trends in clustering algorithms, open challenges, and further research directions.

The rest of the paper is organized as follows: Section 2 discusses the scientific background of data clustering analysis. A detailed bibliometric analysis of classical and nature-inspired metaheuristic algorithms is presented in Sect. 3. Section 4 offers a comprehensive review of the state-of-the-art nature-inspired metaheuristic clustering algorithms. Section 5 deals with the detailed discussion on clustering algorithms in general, while Sect. 6 covers the open challenges, further research directions, and trending application areas of clustering algorithms. Finally, the concluding remarks from the findings arising from the systematic review are presented in Sect. 7.

2 Scientific background

The clustering procedure involves the partitioning of N data vectors into a collection of K groups or clusters. Given a set of data vectors $X = \{x_1, x_1, \ldots, x_N\}$, group them such that "more similar" vectors are in the same cluster and those "less similar" are in different clusters. Depending on the underlying clustering technique being used to address the clustering problem, the definition of clustering may vary as we describe in the next subsections.

2.1 Number of possible clustering groups

Cluster analysis is an unsupervised machine learning technique aimed at discovering the natural grouping of unlabeled objects according to the similarity of measured intrinsic characteristics [134]. The two fundamental problems in automatic clustering are determining the optimal number of clusters and identifying all data groups correctly. In this sense, the number of combinations in assigning N objects into K clusters is represented as follows:

$$S(N,K) = \frac{1}{K!}\sum_{I=0}^{K}(-1)^{K-i}\binom{K}{i}i^N, \quad (1)$$

where $S(N,K)$ is known as the Stirling number of the second kind.

The problem is compounded by the fact that the number of clusters is usually unknown. The search space size for finding the optimal number of clusters is given as follows:

$$B(N) = \sum_{K=1}^{N} S(N, K), \tag{2}$$

where $B(N)$ is known as the Bell number.

Because the data clustering (or grouping) problem of finding an optimal solution is considered to be NP-hard when $K > 3$ [77], hence, even for moderate-sized problems, the clustering task could be computationally prohibitive [46].

2.2 Clustering techniques

The specialized literature on cluster analysis, as explained earlier, commonly classifies clustering techniques as either partitional or hierarchical [106, 134]. The details of these follow.

2.2.1 Partitional clustering

Partitional clustering can be performed in two different modes: either hard (or crisp) or fuzzy. On the one hand, hard clustering assumes that the membership of patterns among clusters is binary; thus, each pattern belongs to exactly one cluster. On the other hand, fuzzy clustering assigns different degrees of membership to the patterns for each cluster, thereby building a non-binary relationship between them.

Hard clustering divides a data set directly into a prespecified number of clusters, without a hierarchical structure [238], so that a data set X is partitioned into K nonoverlapping groups $C = \{c_1, c_1, \ldots, c_K\}$, such that the following three conditions should all be satisfied:

- $c_i \neq \emptyset,\ i = 1, 2, \ldots, K;$
- $\cup_{i=1}^{K} c_i = X;$
- $c_i \cap c_i = \emptyset,\ i, j = 1 \ldots, K$ and $i \neq j.$

Perhaps the most fundamental algorithm related to hard clustering is the k-means algorithm, which attempts to minimize the sum-of-squared-error criterion [94, 144]. Fifty years after its formulation, k-means is still popular and widely used because of its simplicity and low computational complexity [106]. However, at the beginning of the algorithm, a predefined number of clusters is required, which is unknown in several real-world clustering applications. Hence, the k-means algorithm has been extended to find the number of clusters automatically; some of these extended approaches include the X-means algorithm [172] and the G-means algorithm [92].

Fuzzy clustering may be defined in terms of fuzzy sets, in which each pattern may belong to more than one cluster simultaneously, with a certain degree of membership $u_j \in [0; 1]$. The membership value of the ith pattern in the jth cluster should satisfy both of the following two conditions:

- $\sum_{j=1}^{K} u_j(X_i) = 1, \quad i = 1, \ldots, N;$
- $\sum_{i=1}^{N} u_j(X_i) < N, \quad j = 1, \ldots, K.$

The most well-known fuzzy algorithm is fuzzy c-means [32], which is essentially a fuzzy extension of the k-means method.

2.2.2 Hierarchical clustering

Hierarchical clustering algorithms produce a hierarchy of clustering called a dendrogram (or tree structure), which represents the nested grouping of the objects in a data set. The procedure builds N successive clustering levels, in which the current clustering is based on the solution obtained at the previous level. Therefore, a hierarchical clustering does not require a priori knowledge about the number of clusters; however, the groups so obtained are static because the objects assigned to a given cluster cannot move to another one. Also, they are associated with arbitrary decision making and time complexity. Agglomerative and divisive approaches are the two main categories of hierarchical clustering, of which single-link and complete-link [106] algorithms, respectively, are the most well known.

2.3 Proximity measures

Clustering algorithms measure the proximity between objects to form groups [54]. The selection of the appropriate proximity measure is important because memberships are defined for every object in data set X. Depending on the kind of proximity measure, different groupings can be created [150]. A proximity measure can be either a distance (dissimilarity) or a similarity between a pair of objects, between an object and a prototype, or between a pair of prototypes. The most common proximity measures used in the automatic clustering techniques described herein are detailed below.

- The Minkowski metric [130], or L_p-norm, is a dissimilarity measure defined as

$$d_p(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{D}|x_i - y_i|^p\right)^{1/p}, \quad (3)$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are D-dimensional data vectors. Note that when $p = 2$, the Minkowski metric becomes the well-known Euclidean distance (or L_2-norm), which is denoted as $d_e(\boldsymbol{x},\boldsymbol{y})$. Two other common special cases of the Minkowski metric are the Manhattan distance (or L_1-norm), when $p = 1$, and the Chebyshev distance (or L_∞-norm), when $p \to \infty$, which is computed as

$$d_\infty(\boldsymbol{x},\boldsymbol{y}) = \max_{1\le i\le D}|x_i - y_i|. \quad (4)$$

- The similarity between two vectors x and y can be measured by the cosine of the angle between them:

$$\cos(x,y) = \frac{\boldsymbol{x}^{\mathbf{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}, \quad (5)$$

where $\|\cdot\|$ denotes the L_2-norm. There is a relationship between cosine similarity and Euclidean distance, such that the cosine similarity can be transformed into a dissimilarity measure as [52].

$$d_{cos(\boldsymbol{x},\boldsymbol{y})} = 1 - \cos(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2}d_e^2(\boldsymbol{x},\boldsymbol{y}). \quad (6)$$

3 Bibliometric analysis

Initially used in the library and information sciences, bibliometric or scientometric analysis has now become a separate independent research domain, having extended its roots into different technical fields. This type of analysis makes overt exclusive, intrinsic and hidden structures about the research area, based on the publication data (also known as bibliometric data). The technique provides not only a fruitful field where young researchers can kick-start their research journey, but it also provides useful insights into the field, thereby helping any researchers to make novel contributions. In the literature, one can find two types of bibliometric studies these days, being journal specific or research area specific. Some of the notable journal specific bibliometric studies have focused on soft computing [152], Applied Soft Computing [158], IEEE Transactions on Fuzzy Systems [236], and neurocomputing [109]. Journal-specific studies are centered on the publications of that specific journal. An alternative focus for bibliometric studies may be the whole research area, listing out all the bibliometric information. There are studies available in the areas of aggregation operators [33], real-time operating systems [207], computer methods and programs in biomedicine (Shukla, Merigó, Lammers, and Miranda, [208], fuzzy sets and systems [206], industry 4.0 [157], fuzzy decision making [34]), etc. This paper deals with a research area-based bibliometric study on automatic clustering algorithms (ACA), covering the background of clustering algorithms.

Apart from tabular analysis, we have used the VOS viewer [225] as the visualization analysis. This popular tool conducts the bibliographic coupling, co-citation analysis, and co-occurrence of author keywords analysis. Bibliographic coupling between two entities occurs when two papers or documents cite the same paper in the reference, while co-citation between two entities is when third, paper cites both entities. Co-occurrence of author keywords assigns a ranking to the keywords used most often by the authors in their documents.

Some performance indicators that we will be using for bibliometric discussion of the collected data include total papers (TP), count of the papers published for that entity, total citations (TC), count of the citations received by the entity, and the citations per paper (CPP), which is the citations received per paper for that entity. Here, an entity could be authors, country, organization, journal, etc. In case of journal analysis, the impact factor (IF) is used for qualitative evaluation, which is computed as the average citations of ACA papers received in that journal for the past 2–5 years.

3.1 Methodology and data statistics

The methodology and data statistics employed in the current study are solely dependent on the bibliometric data, which contain all the information regarding a publication, such as authors, journal, keywords, countries, document type, etc. This bibliometric data is generally indexed by Web of Science (WoS), Scopus, and Google Scholar, etc. In this paper, we have considered only the WoS database, as this platform indices only high-quality journals and ranked international conferences, thereby ensuring that the publications are of high quality.

This paper is based on quite recent automatic clustering approaches, which are a subset of the clustering approaches. Therefore, for the research methodology, we chose the keyword query to be "clustering algorithms" or "Automatic Clustering Algorithm" (ACA). Collectively we call the output of the search query as ACA. This query will return all those publications where any of these keywords appear in the title, abstract, or authors' keywords. The search was performed on April 19, 2020, which resulted in 5063 papers. However, because this was only 4 months into 2020, we have considered data only until December

2019. Therefore, the full-year range is 1989–2019. The refined query resulted in a total of 4875 publications. These publications were classified by the WOS as 13 document types, which are shown in Table 1. Some of the document types include articles (4751), proceeding papers (465), reviews (87), and early access (14). It should be noted that the total numbers for all these document types and the total percentage may be higher than expected (4875 for publications and 100% for percentage) as there could be a few publications which were classified as more than one document type.

The year-wise publication count, total publications (TP) is shown in Fig. 3. Since the growth in publications was increasing over the years, the maximum publications came in 2019 (TP = 577). We take the 5-year interval to be as follows: Y1 (1989–1994), Y2 (1995–1999), Y3 (2000–2004), Y4 (2005–2009), Y5 (2010–2014), and Y6 (2015–2019). The interval Y1 has a range of 6 years to make all other ranges to be 5 years. The maximum percentage of growth rate can be seen during the year range Y3 (169.43%), followed by Y4 (123.17%) and Y6 (74.70%). Interestingly, the lowest growth rate is observed during Y5. Irrespective of this variation, the overall growth has increased over the years, which confirms the growing interest in the research community to contribute to and evolve this ACA domain.

Corresponding to the data in Fig. 3, we have Fig. 4, which shows the number of citations received by all the papers in ACA. As would be obvious from the results for a number of publications, the highest citations were received in year 2019, with a citation count of 19,012. Apart from this, there are five other years for which the received citations were more than 10,000, i.e., 2018 (TC = 16,945), 2017 (TC = 15,445), 2016 (TC = 14,191), 2015 (TC = 12,802), and 2014 (TC = 11,295). To further explore the citation structure, Table 2 shows the citation structure of all the publications in ACA. From this table, we find that there is one paper (that is 0.02% of all the papers) which received more than 6000 citations. This paper is "Data clustering: A review" by Jain et al. [107] in the ACM computing surveys, which has received a total of 6154 citations since September 1999. There are three other papers (0.06% of all the papers), each with more than 3000 but less than 6000 citations. Specifically, these are "CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure" by Jakobsson and Rosenberg [108], "A tutorial on spectral clustering" by Von Luxburg [226], and "Data clustering: 50 years beyond K-means" by Jain [106]. In addition, two papers (0.04%) have between 2000 and 3000 citations, while nine papers (0.19%) have between 1000 and 2000 citations. Notably, Jain [106] is singularly the most highly cited author; contributing 9266 citations among the total of 148,134 citations. More details on the authors' contribution are discussed in Sect. 3.3.

Table 1 Document types in Web of Science (WoS)

Document types	Total number	Percentage (%)
Article	4751	97.45
Proceeding paper	465	9.53
Review	87	1.78
Early access	14	0.28
Editorial material	9	0.18
Letter	8	0.16
Correction	6	0.12
Meeting abstract	6	0.12
Note	4	0.08
Book chapter	3	0.06
News item	2	0.04
Database review	1	0.02
Software review	1	0.02

3.2 Source/journal analysis

In considering the development of a research field like ACA, the source of publications plays a very prominent role as the propagator or server of information. Moreover, it gives a straightforward direction to the young researchers. Table 3 lists the top 25 journals, ranked on the basis of number of publications in ACA. This table also contains the citations received by those published papers, citations per paper received, and IF of the journals.

As it turns out, the journal *Pattern Recognition* published the most papers in ACA (TP = 120), followed by *IEEE Access* (TP = 114), *BMC Bioinformatics* (TP = 105), and *Expert Systems with Applications* (TP = 104). These are the only four journals with more than 100 publications, although all the journals in the top 25 have published more than 20 papers in the field. With respect to influential journals as judged by the number of citations (TC), *Bioinformatics* is the most influential among the top 25, with a total citation count of 9330. Other influential journals in the list are *IEEE Transactions on Pattern Analysis* and *Machine Intelligence* (TC = 6059), *Pattern Recognition* (TC = 5993), and *Pattern Recognition Letters* (TC = 5633).

Since ACA covers the analysis of the clusters of the datasets, there are journals in the top 25 that are related to medical fields such as *BMC Bioinformatics* (TP = 105, TC = 3065), PLOS One (TP = 76, TC = 1098), *Bioinformatics* (TP = 67, TC = 9330), and *IEEE-ACM Transactions on Computational Biology and Bioinformatics*

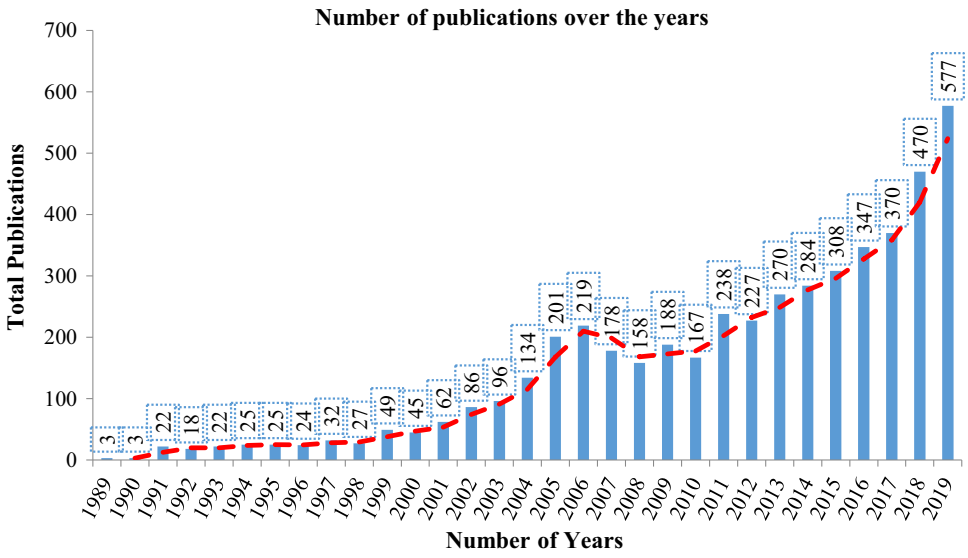

Fig. 3 Total number of publications over the years 1989–2019

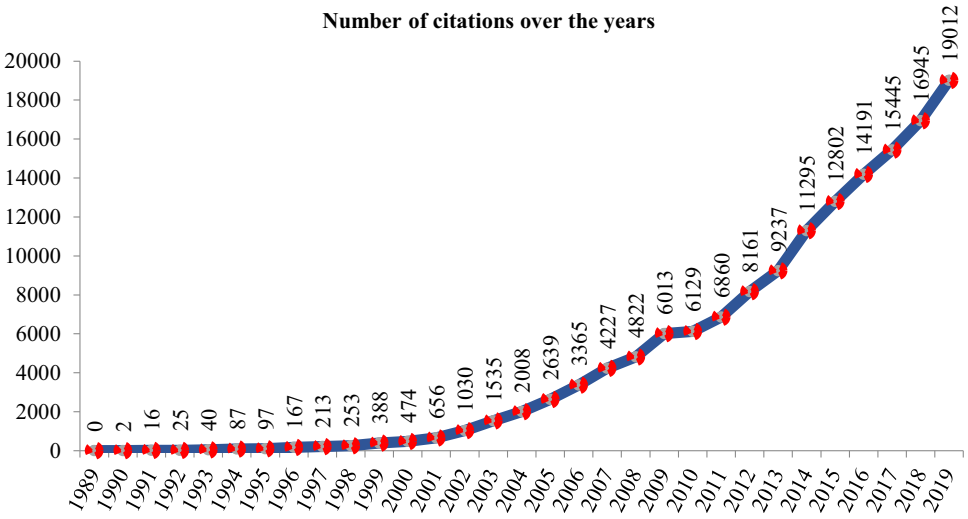

Fig. 4 Total number of received citations over the years from 1989 to 2019

Table 2 Citations' structure of publications in ACA

Number of citations	# of publications	% publications
≥ 6000	1	0.02
≥ 3000	3	0.06
≥ 2000	2	0.04
≥ 1000	9	0.19
≥ 500	21	0.43
≥ 250	52	1.06
≥ 100	166	3.39
≥ 50	294	6.03
≥ 25	4347	89.16
Total publications	4875	100

(TP = 22, TC = 382). That *Bioinformatics* is also the most influential journal, as discussed above, shows the importance of clustering approaches for data in the medical field. Interestingly, *Bioinformatics* has also received the second largest number of citations per paper (CPP = 139.25), after *IEEE Transactions on Pattern Analysis and Machine Intelligence* (CPP = 144.26). *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics* also has more than 100 citations per paper, i.e., 104.38. For comparison, while still falling into the top 25 for a number of publications, journals with less than 5 CPP are *IEEE access* (CPP = 2.12), *Mathematical Problems in Engineering* (CPP = 2.28), *Journal of Intelligent & Fuzzy Systems* (CPP = 3.89), *Intelligent Data Analysis* (CPP = 3.98), and *Wireless Personal Communications* (CPP = 4.21).

The next part of journal analysis is the bibliographic coupling of ACP publications. A minimum threshold of 10

Table 3 Top 25 journals publishing work in (ACA)

S. no.	Source/journals	TP	TC	CPP	IF
1	Pattern Recognition	120	5993	49.94	5.898
2	IEEE Access	114	242	2.12	4.098
3	BMC Bioinformatics	105	3065	29.19	2.970
4	Expert Systems with Applications	104	1864	17.92	4.292
5	Pattern Recognition Letters	89	5633	63.29	2.810
6	Neurocomputing	88	1204	13.68	4.072
7	Information Sciences	76	2150	28.29	5.524
8	PLoS One	76	1098	14.45	2.776
9	Applied Soft Computing	74	1509	20.39	4.873
10	Bioinformatics	67	9330	139.25	4.531
11	IEEE Transactions on Knowledge and Data Engineering	62	2392	38.58	3.857
12	Knowledge-Based Systems	51	824	16.16	5.101
13	Soft Computing	49	358	7.31	2.784
14	IEEE Transactions on Fuzzy Systems	46	1930	41.96	8.759
15	Knowledge and Information Systems	46	499	10.85	2.397
16	IEEE Transactions on Pattern Analysis and Machine Intelligence	42	6059	144.26	17.730
17	Intelligent Data Analysis	42	167	3.98	0.612
18	Fuzzy Sets and Systems	34	1126	33.12	2.907
19	Engineering Applications of Artificial Intelligence	30	756	25.20	3.526
20	Data Mining and Knowledge Discovery	29	2632	90.76	2.879
21	Wireless Personal Communications	29	122	4.21	0.929
22	Mathematical Problems in Engineering	29	66	2.28	1.179
23	Journal of Intelligent & Fuzzy Systems	28	109	3.89	1.637
24	IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics	24	2505	104.38	6.22
25	IEEE-ACM Transactions on Computational Biology and Bioinformatics	22	382	17.36	2.428

documents published by each journal was used. Of the 1344 journals identified, only 87 meet the set threshold; the bibliographic coupling data for the top 50 journals are plotted in Fig. 5. Here, the links between the journals represent the articles which most frequently refer to the common documents. The size of each node represents the number of publications associated with it, which implies that the bigger the node, the greater the number of coupled papers. This pattern can be verified from the results of Table 3. For example, as shown earlier, the most productive journals are *Pattern Recognition*, *IEEE Access* and *BMC Bioinformatics*, which produced the biggest nodes in Fig. 5.

Different colors are used in Fig. 5, so that different colors indicate sets of nodes that form clusters. Same color clusters are those journals wherein articles mostly cite the same documents. Of interest are the medical journals forming a cluster of blue color consisting of *BMC Bioinformatics, Bioinformatics, PLOS One, IEEE-ACM Transactions on Computational Biology and Bioinformatics,* and *Computers in Biology and Medicine*, etc. It is to be noted that some journal names are not visible, which may be either because their node is too small or because they are hiding behind a bigger node. This is just a limitation of VOSviewer. For instance, *Pattern Recognition Letters* (green node) should be the fifth biggest node according to the number of publications, but it is hiding behind the biggest node, which is *Pattern Recognition*. Some of the major journals in the red color clusters are *Pattern Recognition, Neurocomputing, IEEE Access,* and *IEEE Transactions on Knowledge and Data Engineering*. The green node clusters comprised of journals like *Pattern Recognition Letters, Applied Soft Computing*, *Expert Systems with Applications*, *IEEE Transaction on Fuzzy Systems*, etc. It seems that this cluster of journals cites papers mostly from the fields of artificial intelligence, machine learning optimization, etc.

Figure 6 shows the co-citation analysis among the journals that have published papers on ACA. Here, co-citation means that a link is established between two journals if a document has cited a publication from both journals. Specifically, in Fig. 6 we can see that biomedical

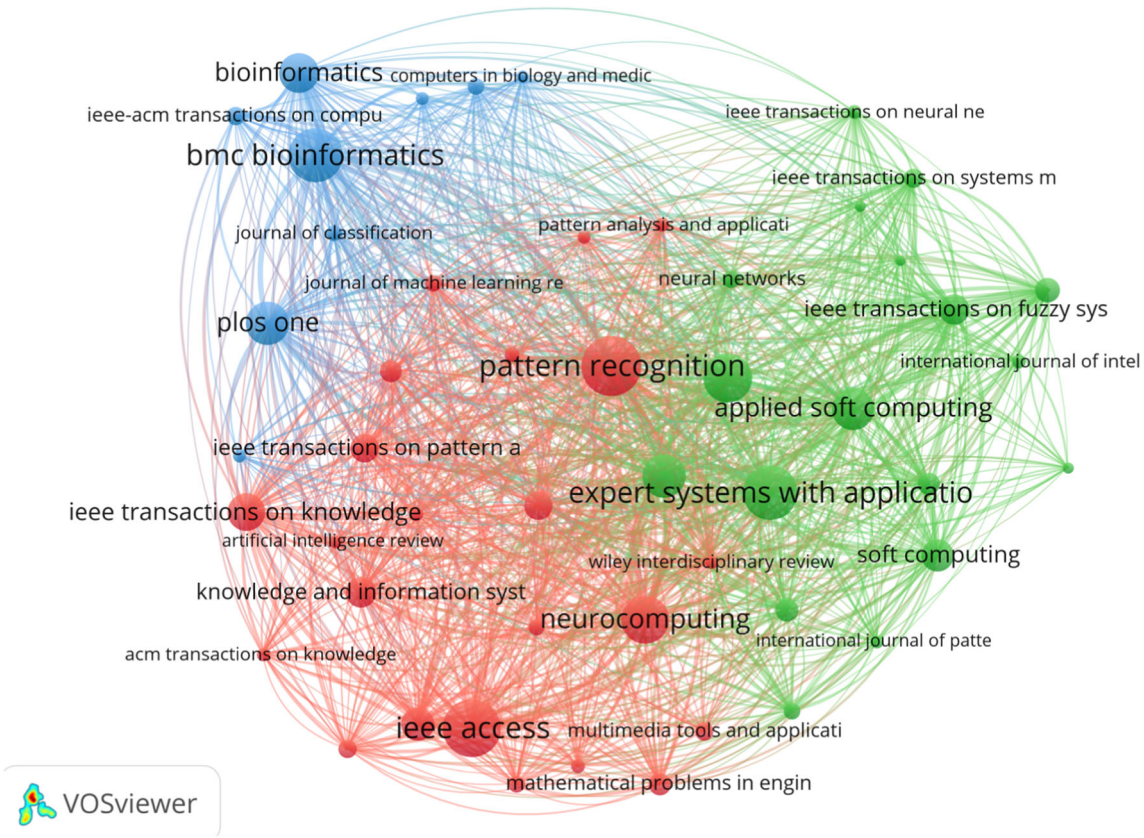

Fig. 5 Bibliographic coupling of the source publishing on ACA (color figure online)

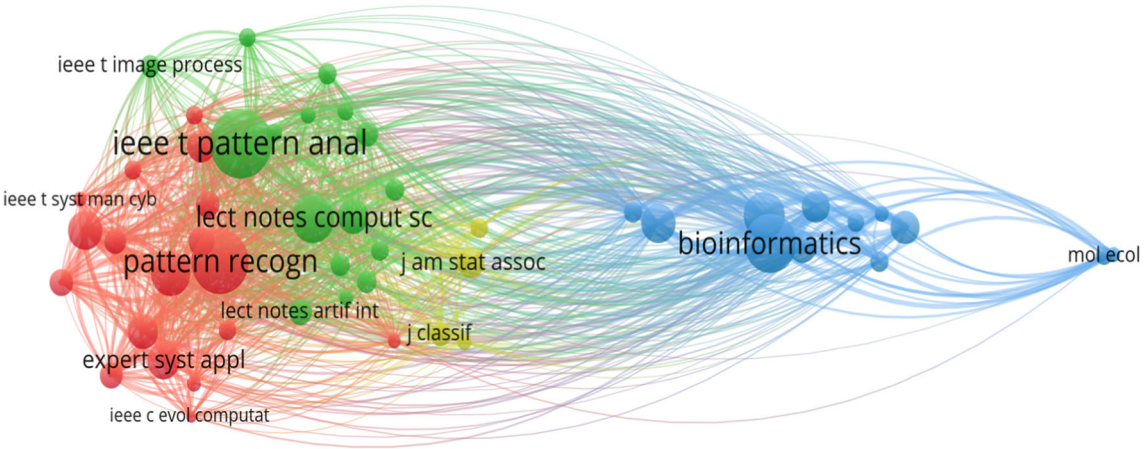

Fig. 6 Co-citation analysis among the journals publishing in ACA

journals like *Bioinformatics* are on the right side and have the links with the journals on the left, which are the ultimate sources of the algorithms on clustering. As clustering is more of a data mining/machine learning approach, the papers on its theory and implementation are published in journals like *Pattern Recognition, Expert Systems with Appli*cations, etc.

3.3 Authors' analysis

Author analysis is also important in that it highlights the most prominent authors who are publishing and getting recognition for their work in ACA. Scholars can then follow the work of the respective authors and become aware of developments from the previously published papers. Table 4 shows the top 25 authors publishing in ACA. On

the left side of this table, authors are sorted based on TP (productivity) and on the right side, according to TC (influence).

Bezdek is the most productive author in the ACA field with 26 publications and received a total of 2776 citations. With the next most publications, we find Jiao L. and Yang M-S., with 21 ACA papers each. However, Jiao stands second as he has received more citations (TC = 376). The same ranking approach has been followed for all the authors who have the same number of publications. Maulik is the fourth most productive author with 18 publications and an apparently higher CPP (26.94) than the above two authors. Bezdek has an even greater CPP of 106.77. There are two authors with 15 papers, namely Yang MS. (TC = 953, CPP = 63.53) and Wang S. (TC = 509, CPP = 33.93).

In terms of the most influential authors, Jain has an astonishing citation per paper of 1491.8, translating to 7459 citations from just 5 papers. This remarkable achievement is primarily due to the most cited clustering paper "Data clustering: A review," which was published in 1999 and has received 6154 citations over the 20 years up to 2019. Jain is followed by Von Luxburg and Bezdek with, respectively, 3481 and 2776 citations and 696.9 and 106.77 citations per paper.

The bibliographic coupling analysis involved two thresholds; authors should have a minimum of five documents with 10 citations. These criteria returned only 133 authors out of all the 13,970 authors. We selected the top 50 of these to plot the visual representation, as shown in Fig. 7. The bibliographic coupling among authors implies that these authors have cited the same set of papers in their publications. These sets of authors formed clusters as indicated by different colors in the figure. The most prominent authors in each of the colored clusters are Jiao L. (red cluster), Maulik (yellow cluster), Yang M-S. (green cluster), and Liang (blue cluster). One might have expected the author Bezdek to have been the largest node, but the data had saved as either Bezdek, James C or Bezdek, J. C. Thus, there were two entries as per the visualization, which is a limitation of the method. However, this mistake is corrected in the corresponding Table 4.

Table 4 Most productive and influential top 25 authors in ACA

S. no.	Author	TP	TC	CPP	Author	TC	TP	CPP
1	Bezdek, James C.	26	2776	106.77	Jain, AK	7459	5	1491.80
2	Jiao, Licheng	21	376	17.90	Von Luxburg, Ulrike	3483	5	696.60
3	Yang, Miin-Shen	21	229	10.90	Bezdek, James C.	2776	26	106.77
4	Maulik, Ujjwal	18	485	26.94	Yang, MS	953	15	63.53
5	Yang, MS	15	953	63.53	Karypis, G	838	6	139.67
6	Wang, Shitong	15	509	33.93	Kriegel, Hp	749	7	107.00
7	Yu, Jian	13	213	16.38	Zhao, Y	693	5	138.60
8	Li, Tao	12	649	54.08	Li, Tao	649	12	54.08
9	Chung, Fu-Lai	12	403	33.58	Krishnapuram, R	636	7	90.86
10	Bandyopadhyay, Sanghamitra	12	379	31.58	Wang, Shitong	509	15	33.93
11	Liang, Jiye	12	244	20.33	Baraldi, A	491	5	98.20
12	Kuo, R. J.	12	216	18.00	Maulik, Ujjwal	485	18	26.94
13	Wang, Jun	11	246	22.36	Masulli, Francesco	452	5	90.40
14	De Carvalho, Francisco De A. T.	11	233	21.18	Rovetta, Stefano	452	5	90.40
15	Wang, Wei	11	123	11.18	Das, Swagatam	414	8	51.75
16	Mukhopadhyay, Anirban	10	333	33.30	Chung, Fu-Lai	403	12	33.58
17	Zhao, Feng	10	262	26.20	Huang, Joshua Zhexue	382	7	54.57
18	Pedrycz, Witold	10	213	21.30	Leckie, Christopher	380	9	42.22
19	Ding, Shifei	10	192	19.20	Bandyopadhyay, Sanghamitra	379	12	31.58
20	Hung, Wen-Liang	10	75	7.50	Aghabozorgi, Saeed	379	5	75.80
21	Leckie, Christopher	9	380	42.22	Jiao, Licheng	376	21	17.90
22	Isa, Nor Ashidi Mat	9	206	22.89	Ng, Michael K.	354	6	59.00
23	Saha, Sriparna	9	117	13.00	Yu, J	342	5	68.40
24	Hou, Jian	9	111	12.33	Mukhopadhyay, Anirban	333	10	33.30
25	Zhang, Xianchao	9	99	11.00	Deng, Zhaohong	325	7	46.43

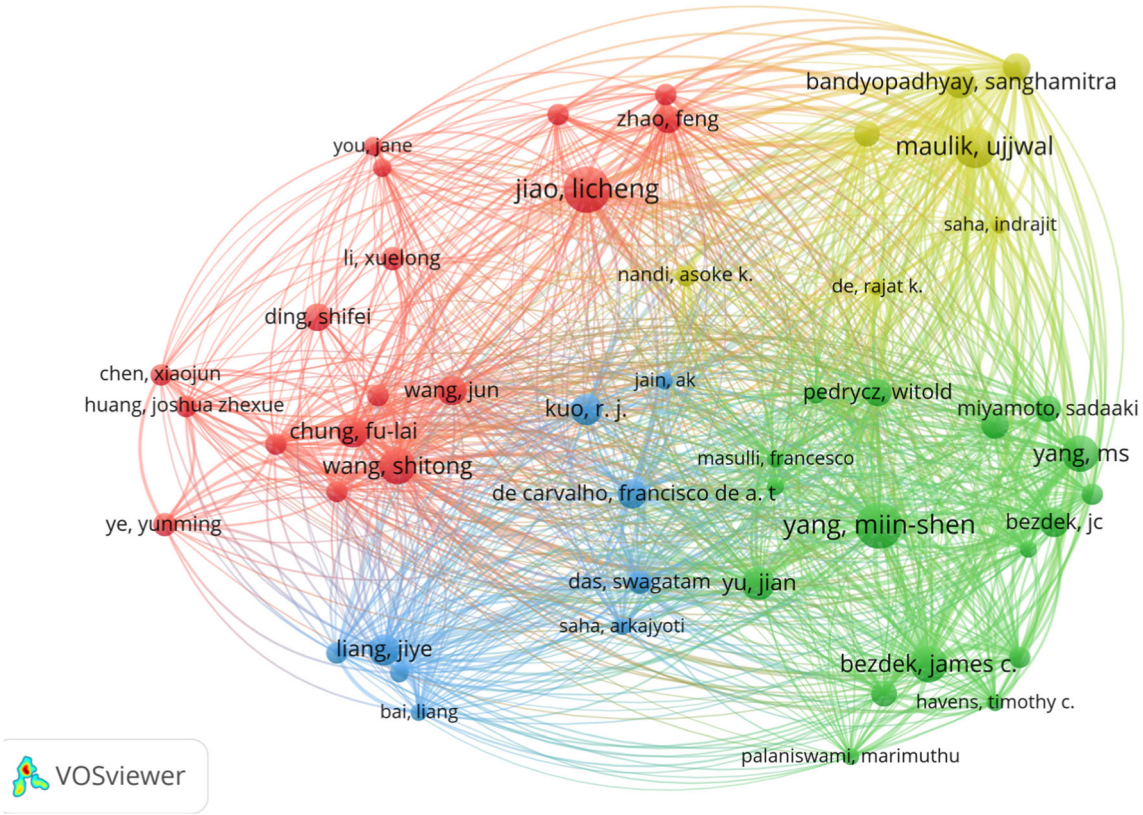

Fig. 7 Bibliographic coupling of the authors publishing on ACA (color figure online)

Figure 8 shows the co-citation analysis among the authors who have published in ACA. Co-citation analysis for authors implies that some third author's document has cited papers of the first two authors. It can be verified by the largest node in the figure (Jain, A.K.). Since the paper by Jain et al. [107], being a review paper, is the most cited paper and lies in the middle of all the authors with particularly strong links with Bezdek, Kohonen, etc.

3.4 Country and institution analysis

Country and the institution analysis are also very important in the bibliometric study as they indicate the regional source of the work in the ACA domain and the universities where the work is being performed. Tables 5 and 6 show the side-by-side comparison according to the most productive and most influential of the best 25 countries and institutions, respectively. These tables also contain the citations per publications for each country and institution.

China tops the list as the most productive country with 1160 publications followed by USA (TP = 1140), India (TP = 313), and England (TP = 249). Notably, third position India has 72.5% fewer publications than second position USA. There are only 15 countries with more than 100 publications in ACA. This trend can be verified by the contributions of the universities in these countries. The Chinese Academy of Sciences and Xiadin University in China top the list of most productive institutions with 78 and 49 publications, respectively. The Indian Statistical Institute stands at third position with 42 publications.

Although China (TC = 16,114) is the most productive country, USA stands at the top of the most influential nation in publications on ACA with 65,666 citations. The next two countries in the list, with more than 10,000 citations each, are India (TC = 12,326) and Germany (TC = 11,580). Once again, the USA has considerably more citation counts than the countries behind it. This almost stands to reason because the most cited author (Bezdek) is from the USA. More evidence is presented in Table 6, where the same effect can also be seen with the most influential institutions being from the USA, namely Michigan State University (TC = 10,834) and Ohio State University (TC = 6562). The institute from India, Indian Institute of Sciences, is at number three with 6501 citations. The next three universities are also from the USA: University of Michigan (TC = 4263), University of Washington (TC = 3900), and University of Missouri (TC = 3252).

Next, we present a graphical visualization of countries and institutions, corresponding to the two Tables 5 and 6.

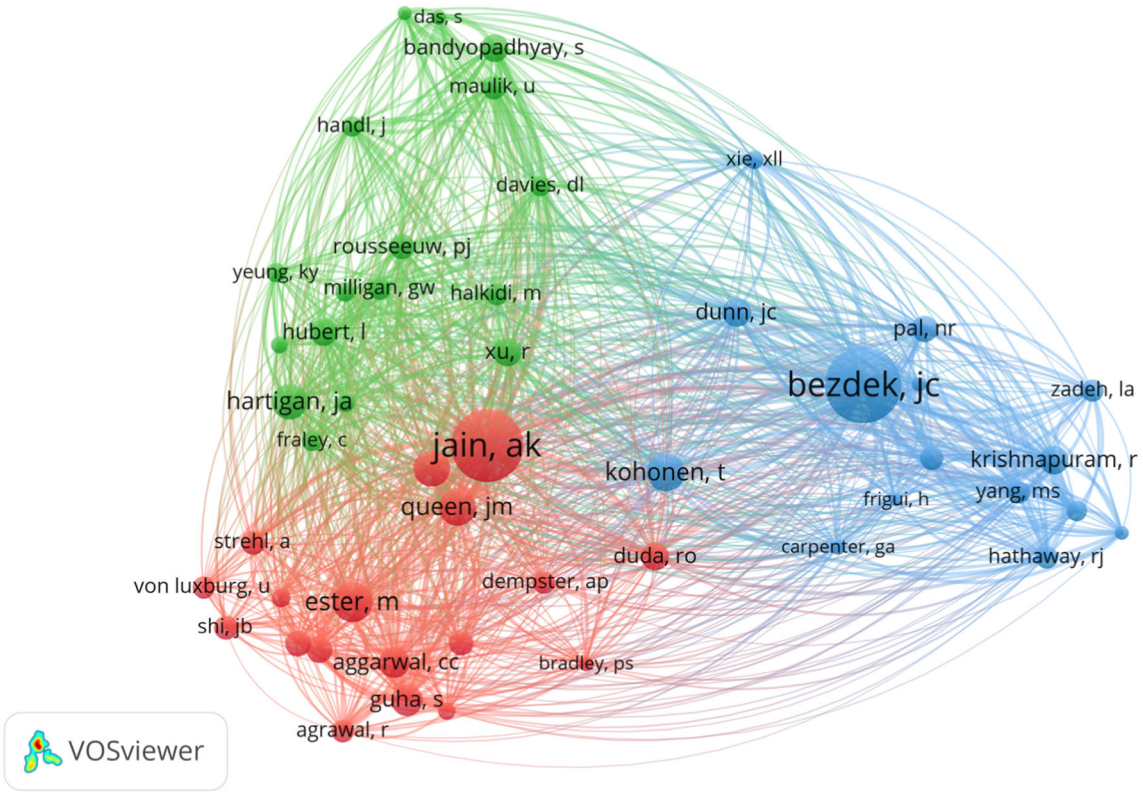

Fig. 8 Co-citation analysis among the authors publishing in ACA

Figures 9 and 10 show the bibliographical coupling of countries and institutions, respectively. As shown in Fig. 9, a bigger node represents the country with most publications, which can also be verified in Table 5. Here, the coupling means that the publications from these countries have cited the same references. Thus, there is a link between the countries. The same color nodes and links suggest that these countries publications have mostly cited the papers published in the corresponding colored node country. We can see four clusters here: China is prominent in the green color cluster containing South Korea, Turkey, India, Taiwan, Iran, Egypt, etc. The USA is prominent in the blue color cluster containing countries such as England, Norway, Austria, Ireland, and Scotland. Similar behavior can be seen in Fig. 10 corresponding to Table 5. The bibliographic coupling indicates that these universities have cited papers mostly from the linked or same cluster universities. In addition, this also revealed that authors seem to work and communicate within their geographic region and community.

3.5 Authors' keyword analysis

This section portrays the keywords most commonly used by the authors who have published papers in the domain of ACA. Figure 11 shows the analysis of co-occurrence of authors' keywords from papers publishing in ACA. As would be expected, "clustering" is the biggest node followed by "clustering algorithms," "data mining," "fuzzy clustering," etc. The keyword in each colored cluster signifies that they are mostly used together. The blue colored cluster has keywords like "clustering algorithms," "optimization," "feature selection," "partition algorithms," etc. Similarly, the green color cluster has keywords such as "clustering," "data mining," "classification," "subspace clustering,", and "microarray."

4 Metaheuristic clustering algorithms

Cluster analysis has over the years evolved to be an important and fast-growing area with widespread applications, but especially in data mining. Many studies have been carried out in this regard, and different approaches to data clustering have also been proposed and implemented. Nature-inspired metaheuristic algorithms are broadly categorized into three research fields, namely evolutionary algorithms (for example, evolutionary strategy, genetic algorithms, and differential evolution algorithms), swarm intelligence (for example, ant colony optimization, particle swarm optimization, firefly, cuckoo search and symbiotic organisms search algorithms), or stochastic, population-

Table 5 Top 25 most productive and influential countries in ACA

S. no.	Country	TP	TC	CPP	Country	TC	TP	CPP
1	Peoples R China	1160	16,114	13.89	USA	65,666	1140	57.60
2	USA	1140	65,666	57.60	Peoples R China	16,114	1160	13.89
3	India	313	12,326	39.38	India	12,326	313	39.38
4	England	249	8126	32.63	Germany	11,580	233	49.70
5	Canada	246	7377	29.99	England	8126	249	32.63
6	Germany	233	11,580	49.70	France	7797	183	42.61
7	Italy	218	6293	28.87	Canada	7377	246	29.99
8	Spain	203	4427	21.81	Italy	6293	218	28.87
9	Australia	200	5016	25.08	South Korea	5587	159	35.14
10	Taiwan	184	3154	17.14	Australia	5016	200	25.08
11	France	183	7797	42.61	Spain	4427	203	21.81
12	South Korea	159	5587	35.14	Taiwan	3154	184	17.14
13	Iran	156	2193	14.06	Greece	3074	86	35.74
14	Brazil	112	2021	18.04	Ireland	2806	14	200.43
15	Turkey	108	1812	16.78	Netherlands	2519	42	59.98
16	Japan	92	1350	14.67	Israel	2486	53	46.91
17	Greece	86	3074	35.74	Iran	2193	156	14.06
18	Singapore	82	2100	25.61	Singapore	2100	82	25.61
19	Poland	65	1133	17.43	Brazil	2021	112	18.04
20	Malaysia	57	1149	20.16	Belgium	1924	51	37.73
21	Switzerland	56	1808	32.29	Austria	1903	35	54.37
22	Israel	53	2486	46.91	Turkey	1812	108	16.78
23	Saudi Arabia	52	1746	33.58	Switzerland	1808	56	32.29
24	Belgium	51	1924	37.73	Portugal	1804	41	44.00
25	Finland	51	845	16.57	Saudi Arabia	1746	52	33.58

based, nature-inspired optimization algorithms. In the next subsections, various nature-inspired metaheuristic clustering analysis methods are presented systematically and discussed.

Driver and Kroeber, in 1932, were the first to introduce and use the concept of cluster analysis, by applying it in the field of anthropology. Later, Zubin applied it in psychology in 1938 and Tryon in 1939. Further, it was used for trait theory classification in personality psychology by Cattell in early 1943. Subsequently, cluster analysis has grown significantly due to its relevance in diverse areas. Jain [106] presented a review of the prominent clustering approaches that had been in existence for over five decades, indicating their evolution, and showing trends in data clustering techniques. Also, Fahad et al. [76] and Nerurkar et al. [164] gave reviews and comparisons of the different clustering techniques and their applications to big data. As mentioned in Sect. 1, clustering algorithms fall into the two main categories of hierarchical or partitional; all the clustering approaches that have been developed so far are designed based on these two classifications. These methods of clustering have emerged from the classical (traditional) methods, through to the evolutionary (natural evolving ones from biological occurrences) methods, and to those based on a group of particles called a swarm, up till those that are chemical-based, geographical-based, music-based, and even sport-based. Although these classes of algorithms operate by different modalities, there are nevertheless apparent similarities among their behavior. Next, we present the generalized algorithmic design framework for the two main classes of metaheuristics techniques.

4.1 Generalized metaheuristic algorithmic design

In general, the algorithmic design frameworks and procedures of the swarm intelligence optimization and evolutionary algorithms appear to be somewhat similar in terms of their design and solution representation. However, solution representation is a prerequisite for the overall performance of both optimization approaches. Similarly, the two broad classes of algorithms share the same design steps, which begins with the first step of random initialization of population size. The second step is the evaluation of the initialized population to identify the candidate individual or particles, which in this case would represent

Table 6 Top 25 most productive and influential universities in ACA

S. no.	Organization	TP	TC	CPP	Organization	TC	TP	CPP
1	Chinese Academy of Sciences	78	1696	21.74	Michigan State University	10,834	16	677.13
2	Xidian University	49	617	12.59	Ohio State University	6562	17	386.00
3	Indian Statistical Institute	43	1911	44.44	Indian Institute of Sciences	6501	9	722.33
4	Nanyang Technological University	42	1371	32.64	University of Michigan	4263	16	266.44
5	City University Hong Kong	42	1115	26.55	University of Washington	3900	28	139.29
6	Hong Kong Polytech University	41	853	20.80	University of Missouri	3252	18	180.67
7	Islamic Azad University	36	592	16.44	Korea University	3139	16	196.19
8	University of Sydney	34	908	26.71	University Toronto	2202	21	104.86
9	Beijing Jiaotong University	34	434	12.76	Cornell University	2191	7	313.00
10	Jadavpur University	32	1141	35.66	University Chicago	2042	5	408.40
11	University of Sao Paulo	31	754	24.32	Stanford University	2028	27	75.11
12	University of Illinois	31	705	22.74	MIT	2002	21	95.33
13	Chung Yuan Christian University	30	713	23.77	Harvard University	1931	20	96.55
14	Harbin Institute of Technology	30	486	16.20	Indian Statistical Institute	1911	43	44.44
15	Dalian University of Technology	30	354	11.80	Imperial College of Science, Tech. & Medicines	1837	10	183.70
16	University Elect Sci & Tech. China	29	537	18.52	University of Texas	1743	20	87.15
17	University of Washington	28	3900	139.29	Chinese Academy of Sciences	1696	78	21.74
18	Jiangnan University	28	446	15.93	University W Florida	1673	10	167.30
19	Wuhan University	28	261	9.32	CSIRO	1605	5	321.00
20	Stanford University	27	2028	75.11	Athens University Econ & Business	1560	7	222.86
21	Natl Taiwan University Sci & Technol	27	556	20.59	Nanjing University Aeronaut & Astronaut	1517	17	89.24
22	Shenzhen University	27	278	10.30	University of Maryland	1487	18	82.61
23	Tsinghua University	26	382	14.69	University of Minnesota	1428	25	57.12
24	Natl University Singapore	26	324	12.46	Nanyang Technol University	1371	42	32.64
25	Tianjin University	26	213	8.19	University S Florida	1349	15	89.93

the choice of solution. The third step is to generate a new population by modifying individual specified variation operators, and this is done iteratively, after which the new individuals are evaluated based on their fitness. An update is made depending on which of the candidate individual is better in terms of the problem-defined objective function. Generally, the selection of the best candidate solutions is made by comparing the previous solution with the current solutions with precedence often given to the current best solution. These steps are iteratively repeated until the termination condition is met, and the optimal best solutions are identified. The general algorithm design for both the swarm intelligence and evolutionary algorithms is shown in Algorithm listings 1 and 2.

Algorithm 1: A generalised evolutionary algorithmic representation (e.g. GA, DE, ES, etc.)

1: *Initialize* population randomly by generating initial population $P(0)$ and setting $i = 1$;
2: *Evaluate* the fitness of each candidate individual in $P(0)$;
3: **While** (termination condition) is not met **do**
4: *Modify* candidate individual using variation operators to form $P(i)$;
5: *Evaluate* the fitness of each candidate individual in $P(i)$;
6: *Select* parents from $P(i)$ and $P(i-1)$ based on their fitness;
7: *Update* candidate individual $P(i-1)$;
8: *Find* the best individual;
9: **End While**

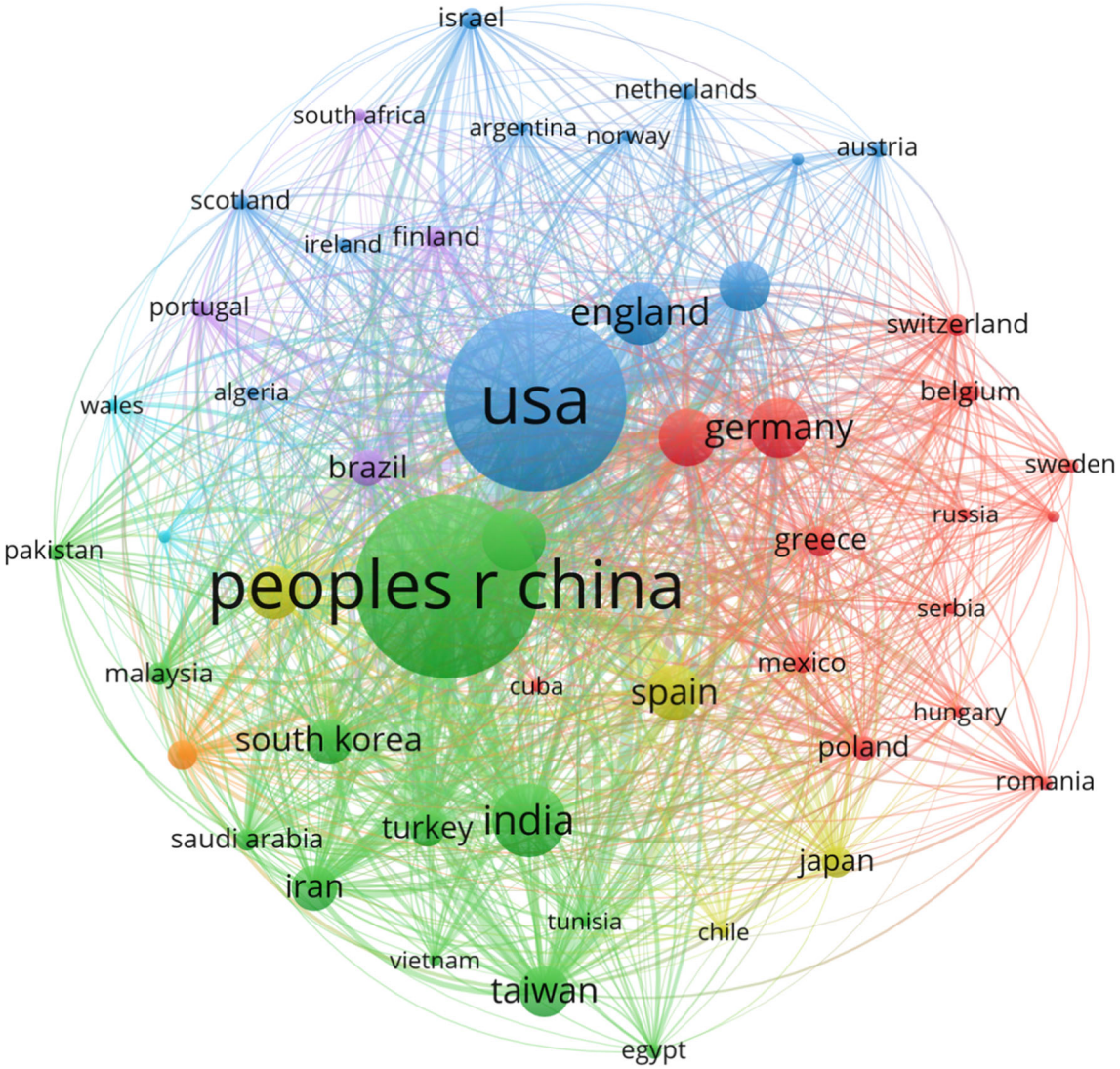

Fig. 9 Bibliographic coupling of the countries publishing on ACA

Algorithm 2 represents a generic framework design for the swarm intelligence-based algorithmic concept, which includes the PSO, ACO, SOS.

4.2 Metaheuristic solution representation and encoding scheme

Generally, for any optimization problem model design, a well-formulated solution representation is vital for the

Algorithm 2: A generalised swarm intelligence algorithmic representation (e.g. PSO, SOS, ACO, etc.)

1: *Initialize* population by randomly generating particles $P(0)$ and setting $i = 1$;
2: *Evaluate* each particle in $P(0)$;
3: **While** (termination condition) is not met do
4: *Modify* particles using specific variations operators;
5: *Evaluate* candidate particles $P(i)$;
6: *Replace* particles $P(i)$ with $P(i-1)$ based on their fitness;
7: *Find* best particle $P(i-1)$ for the next generation;
8: *Repeat* the worst particle $P(i)$;
9: **End While**

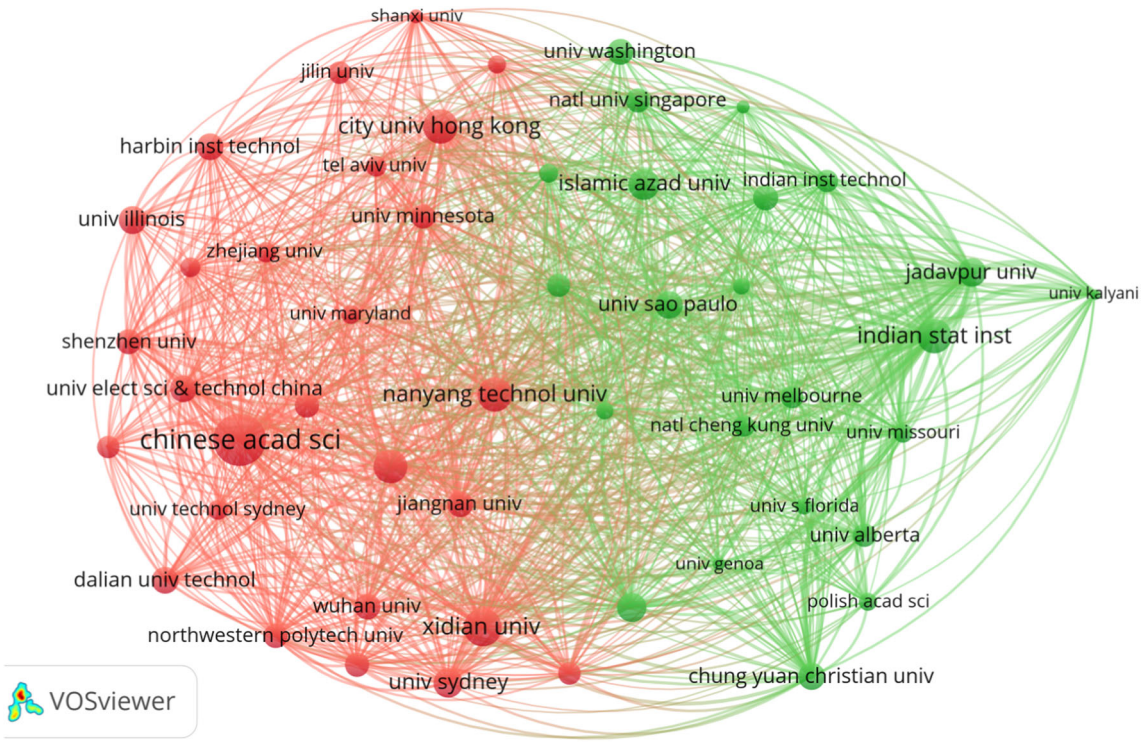

Fig. 10 Bibliographic coupling of the institutions publishing on ACA

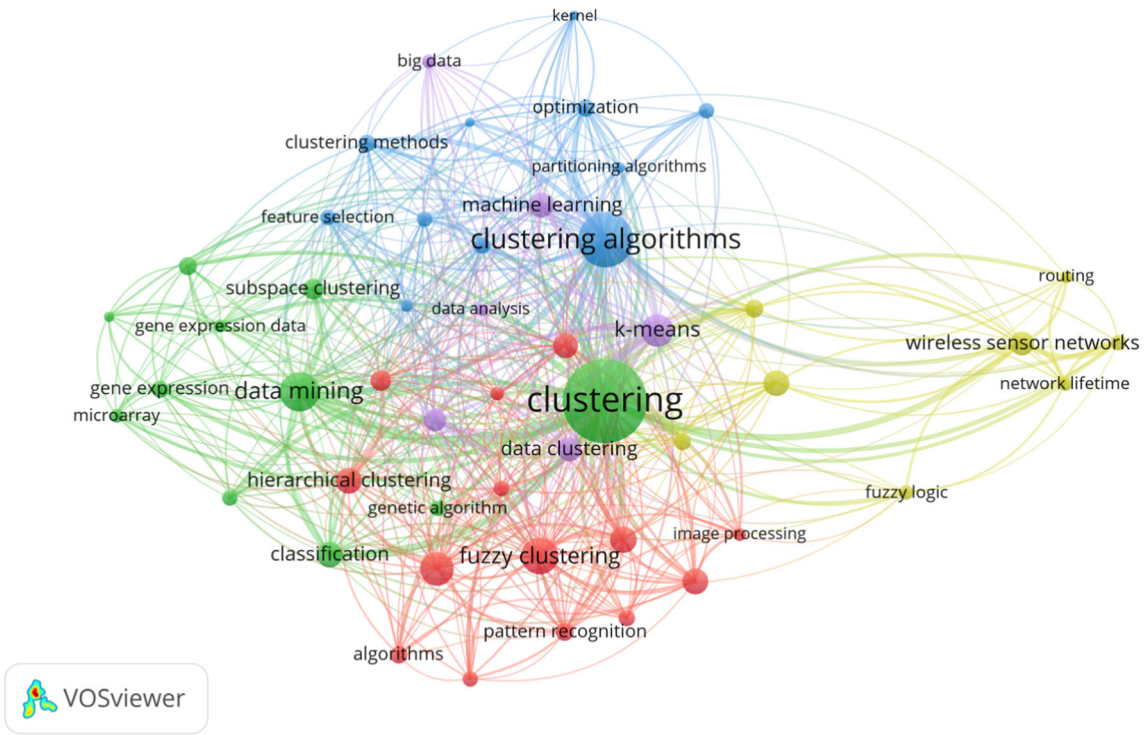

Fig. 11 Co-occurrence of authors' keyword analysis in ACA

successful performance and scalability of most nature-inspired metaheuristic algorithm application to the problem at hand. In other words, the successful application of metaheuristic techniques to solve any real-world problem is linked to a well-formulated solution representation. The solution representation for a candidate clustering problem

can be formulated as follows: Consider a dataset X, which contains n data points say $x_1, x_2, \ldots, x_n$, with d-dimensional attributes, features, variables, components [75]. Formally, this can be expressed in a vector form as, $X = \{x_1, x_2, \ldots, x_n\}$ to represent a set of n data points, each having d real-valued features. In other words, $x_i \in \mathbb{R}^d$, $\forall i = 1, 2, \ldots n$, $x_i = \{x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots x_{id}\}$, where x_{ij} denote all the features of x_i. Therefore, the population matrix can be initialized as follows:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i,1} & x_{i,2} & x_{i,j} & x_{i,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}$$

The population matrix, in this case, comprises strings which are composed of real numbers that encode the centers of the cluster partitions. Now let the individuals in the population encode the number of clusters k_i. The grouping of individuals in the population into a similar class is estimated based on the lower bound denoted by $k_{\min}$ and upper bounds denoted by $k_{\max}$ of the number of groups in the population. The number of clusters for each individual is evaluated using the following expression.

$$k_i = rand(0, 1) \times (k_{\min} + (k_{\max} - k_{\min}))$$

where the function $rand()$ denotes randomly generated numbers between 0 and 1.

Again, for a d-dimensional space, the length of a d-dimensional dataset is denoted as $d \times k_{\max}$, while an individual consists of a vector of real numbers of dimension $k_{\max} + k_{\max} \times d$. The length of i^{th} individual is given as $d \times k_i$. Note that the first $k_{\max}$ values are positive floating points numbers in [0, 1], and these values are similarly used to determine the suitability of the corresponding clusters for the data points classification. Afterward, the remaining $k_{\max}$ values are set aside for $k_{\max}$ clusters' centers, each d dimensional space [13, 75]. Also, note that the initial population for the metaheuristic algorithms is generated randomly. For example, in the case of PSO, FA, SOS, and IWO algorithms, a random number between [0, 1] is generated for each data points (in this case individual particles) position in the first part. In contrast, for the initial centroid, a data point is picked randomly for each cluster centroid.

4.3 Trend from traditional algorithms to metaheuristic algorithms

Traditional approaches to clustering do not require an in-depth or rigorous search process. Also called "heuristic" methods, these traditional approaches are instead driven by "trial and error" in order to find or discover solutions to a specific problem. There is, therefore, no guarantee that the optimal solutions will be found in a reasonable time because they are prone to be trapped within the search space. Xu and Tian [232] have provided a comprehensive review of clustering algorithms based on different characteristics. The k-means clustering algorithm [238], which is the most common and popular classical and partitional clustering algorithm, and also considered to be one of the top ten algorithms in data mining [231], has been in the forefront of these heuristic techniques, due to its ease of implementation, flexibility, and efficiency [5, 8]. However, the major limitations of the k-means stem from the method relying on the need for predetermined information about the data, as well as the initial solution. This dependence means the algorithm may be easily trapped within a local optimum [245]. Another classical clustering method is the fuzzy c-means (FCM) algorithm [32], which, although not as popular as the k-means algorithm, is the most common in the field of fuzzy clustering. In 2017, Sathapan and colleagues [198] carried out a literature study on numerous traditional clustering algorithms for uncertain data. Some of these traditional algorithms include the k-means, k-medoid, global kernel k-means, k-mode, u-rule, uk-means, and fuzzy c-means. Their study was motivated by the increasing need to deal with the complexities associated with real-world data. To overcome these challenges arising from the limitations, researchers have devised and come up with some other more productive and efficient approaches.

Various approaches have over time been implemented, which have been designed specially to handle high-dimensional and complex real-world problems. These are either evolutionary-based or swarm intelligence algorithms and are referred to as being "metaheuristic" because they require a higher heuristic search in finding optimal solutions. They are thus more significant in solving general problems, especially those of optimization and computational problems. Further, metaheuristic methods look out for the most promising (optimal) solution in their search space, thus keeping a balance between *diversification* and *intensification*, and they try as much as possible to prevent their search results from jumping into unpromising regions within the search space. It is noteworthy to mention here that the intensification feature helps metaheuristics methods to obtain the best value for decision variables, while the diversification feature makes them well suitable for problems with large search space.

Hussain et al. [103] provided a study on the metaheuristic methods that had been in use for about 33 years in the area of optimization, in which they also investigated the trend by which metaheuristic methods have grown over that time. Evolutionary-based algorithms (EAs) are

strategies that are based on the processes occurring during the natural evolution of different species [42], and are, more specifically, known as Darwin's concept of evolution of living things according to the "survival of the fittest" [178, 246]. These algorithms include the GA [85], or differential evolution (DE) [53, 211]. The two are similar in that the DE uses the same genetic functions of GA, namely crossover, mutation, and selection, but the GA depends more on the crossover function, while the DE is based more on the mutation factor. Evolutionary programming (EP) [217], which is one of the foundational approaches to modern methods, also uses the mutation and crossover operators. Other examples include teaching learning-based optimization, which is based on the actual learning experience of students in a learning environment [116]. Evolutionary algorithms usually have a few random individuals to initiate the search process, and then, the algorithm proceeds with the search for the best solution while being guided by rules [177]. Hruschka et al. [101] presented a study on evolutionary computation based on clustering algorithms. Akyol and Alatas [19] classified evolutionary algorithms into nine groups, based on the method and approach by which they are inspired, which is similar to the study carried out by Rajpurohit et al. [177] where a total of 176 metaheuristic algorithms were reviewed. Some of the metaheuristic algorithms, as mentioned earlier, are chemical-based, such as the artificial chemical reaction optimization algorithm (ACROA) [20], or biology-based, such as the bacterial colony optimization (BCO) [166], bacterial evolutionary algorithm (BEA) [52], and bacterial swarming (BS) [42]. Swarm intelligence (SIs) algorithms are based on the collective and social behavior of living creatures; they are also referred to as "population-based" algorithms [246]. Some of these SIs are, for instance, particle swarm optimization (PSO) [63], firefly algorithm (FA) [234], artificial bee colony (ABC) [115], invasive weed optimization (IWO) [151], cuckoo search (CS) [204], ant colony optimization (ACO) [55, 56], teaching learning-based optimization (TLBO) [183, 246], the FIFA World Cup [185], which is one of the new generation metaheuristic algorithms, and so forth. Recently, Molina et al. [155] presented a comprehensive study on the taxonomies of nature-inspired and bio-inspired optimization algorithms. The study includes several metaheuristic algorithms based on their source of inspiration and behavior of the particles or organisms.

Almost all of these metaheuristic algorithms have been effectively applied to solve clustering problems. In this section, we review published studies that have been done on both non-automatic and automatic clustering using nature-inspired metaheuristic algorithms. The categorization done in this section is according to the taxonomy study presented in [155].

4.4 Clustering with swarm intelligence-based algorithms

All swarm intelligence methods, also called population-based algorithms, are inspired by the collective intelligent interactions which emerge from the social behavior of a group of individual (organisms) or particles in an environment. It should be noted that, while not all population-based methods are swarm intelligence algorithms, all swarm intelligence algorithms are population-based, because they require the co-interaction of participating individuals to carry out an exhaustive search for possible and promising solutions. In this section, we review the clustering algorithms based on swarm intelligence.

4.4.1 Particle swarm optimization (PSO)

Merwe and Engelbret were the first to solve clustering problems using the PSO [224], whereby the particles that were randomly generated were mapped to one data vector, such that a data vector represents a cluster centroid. They proposed two new approaches using PSO for data clustering. The first approach was to show that PSO can be used to find centroids of a user-specified number of clusters and, secondly, extend the PSO with the k-means algorithm to seed the initial swarm, thus refining the newly formed clusters. Results showed that their approaches have better convergence, large inter-cluster distances, and smaller intra-cluster distances than the other compared methods from the literature. However, their proposed methods were not able to automatically determine the number of clusters, and also, the methods could not scale through on high-dimensional data.

Similar to the study in [224], Zhao et al. [241] proposed an improved performance of the PSO by integrating it with the k-means algorithm to avoid the hybrid PSO algorithm's performance being directly affected by the original clusters. Experimental results and comparison with the classical k-means algorithm showed that the improved PSO with k-means had superiority over classical k-means with respect to time. Nevertheless, the proposed method also could not determine the appropriate number of clusters automatically.

Chuang et al. [43] proposed a new strategy by combining the particle swarm optimization strategy with an acceleration strategy. The new chaotic optimization algorithm for data clustering was called ACPSO. The proposed method searches through arbitrary datasets for appropriate centroids, thus efficiently finding better solutions. They compared their results with six other algorithms, and the results for their proposed method showed its superiority over other compared methods in terms of robustness in

finding cluster centroids and time efficiency. ACPSO, however, cannot automatically identify clusters, as it requires the number of clusters to be defined or known a priori.

Silva Filho et al. [209] proposed two data clustering methods by hybridizing PSO with fuzzy c-means, called FCM-IDPSO and FCM2-IDPSO. Their methods were aimed at dynamically adjusting PSO parameters during execution, hence to provide the right balance between exploitation and exploration, while avoiding falling into local minima, and so providing better solutions. Their results, when compared with other existing methods and those based on the PSO, showed more accuracy and obtained better solutions. A limitation of their proposed model was that it could not automatically determine the number of clusters. The authors, however, proposed that their model should be extended so it could be able to obtain clusters automatically. In other words, they suggested that automatic data clustering is an active approach that could handle the shortfall.

Rana et al. presented a boundary-restricted adaptive particle swarm optimization, the BR-APSO algorithm, for data clustering [180]. The boundary restriction was introduced into the standard PSO to allow particles to go outside the boundaries of their search spaces. Still, it could also forcibly bring back particles that had gone outside the search space during the evaluation process. Experimental results showed that, when compared with seven other algorithms, the proposed adaptive PSO outdid them in terms of robustness, accuracy, and convergence speed. The BR-APSO was, nevertheless, unable to detect clusters automatically.

Another use of PSO for data clustering is found in the study carried out by Cura [48], where he applied a new approach of PSO to data clustering. His new approach followed the "gbest neighborhood topology" of the standard PSO algorithm, such that a new particle moves toward its previous best position and toward the best particle in the search space. Cura compared results from his study with those from three existing algorithms and could show that his proposed method outperformed the others in terms of robustness, effectiveness and easy tuning. Although his method solved the clustering problem in which the number of clusters was not known, it, however, did not give the distinct number of clusters present in the datasets used. Also, while implementing the method for an unknown number of clusters, robustness was reduced due to the large difference obtained between the best and worst fitness values.

Lashkari and Moattar [135] proposed an extended chaotic PSO, known as ECPSO, for data clustering. The proposed ECPSO used the purity index to evaluate the clustering solutions, and it was shown that it had more enhanced and intelligent operations than the standard PSO. The enhanced modifications of PSO included a chaotic initial population generation with a systematic migration procedure; thus, the ECPSO was able to improve the exploration ability and convergence rate of the original PSO. Simulation results were compared with those for four other algorithms and showed that ECPSO yielded more optimized solutions and a higher degree of purity than did the other algorithms. The authors mentioned that ECPSO was able to achieve this superiority because the number of clusters was predefined before the proposed method was trained on the chosen datasets. Hence, ECPSO was not able to automatically determine the number of clusters. The authors also suggested that for future works, the method could be integrated or combined with some other evolutionary algorithms for better efficiency.

Recently, Alswaitti et al. [22] implemented a density-based PSO, called DPSO, for data clustering. In trying to find a balance between intensification and diversification and address the issue of premature convergence with the classical PSO, they used a combination of a kernel-based density estimation technique that is associated with a new bandwidth estimation method and also estimated multi-dimensional gravitational learning coefficients. DPSO used the Dunn index to evaluate its effectiveness. Simulation results were compared with those from five other state-of-the-art algorithms, and DPSO showed better performance in terms of classification greater accuracy and cluster compactness, with lesser computational time than did the others.

Duan et al. [58] implemented a hybrid approach using the artificial bee colony (ABC) and PSO, called ABCPS, in order to establish a form of diversity within the swarm during exploration and to give fast convergence. ABCPS made use of the modified partition coefficient index (MOC), Fukuyama and Sugeno (FS) index, and weighted inter-intra (Wint) index. From simulation results, their proposed hybrid method outperformed three other compared methods, in terms of better solutions.

A clustering algorithm based on the hybridization of PSO with k-means, called IKPSO, was developed by Atabay et al. [26]. The authors exploited the simplicity and speed of the k-means and the generalization and effectiveness of PSO. Results were compared with those from the classical PSO and k-means algorithm and showed that IKPSO outperformed the other two in terms of accuracy and speed.

Similarly, Nayak et al. [161] hybridized an improved PSO with a genetic algorithm (GA) and k-means for cluster analysis (GA-IPSO-K-means). The improved PSO was used to determine the number of clusters; the GA being integrated to improve the quality of particles, while the k-means algorithm refined the solutions in the last search

phase to avoid premature convergence. Simulation results, when compared with results from other algorithms, showed that the proposed method outperformed the others in terms of fast convergence and optimal solution. The proposed method, however, was not able to choose the number of clusters automatically, neither could it find better fitness value. Furthermore, the number of clusters needed to be defined before training with the method.

Huang et al. [102] hybridized the PSO with ant colony optimization (ACO) for data clustering, calling their method ACOR-PSO. Their proposed method incorporated a continuous ACO with PSO to improve the searchability by looking into four models of hybridization, namely sequence approach, parallel approach, sequence approach with an enlarged pheromone-particle table and global exchange. Simulation results when compared with those from the k-means algorithm, classical PSO and classical ACO showed that the sequence approach with the enlarged pheromone-particle table is superior in effectiveness to other approaches because of the diversity of generation of new solutions that the pheromone table offers from the ACO, which in turn prevents entrapment within the local optima.

A dynamic clustering approach, called DCPSO, was developed by Omran et al. [167] and applied to image segmentation. DCPSO automatically obtained the appropriate number of clusters and simultaneously partitioned the clusters without the need for human interference. The algorithm starts by partitioning the dataset into a large number of clusters to reduce initial condition effects. Then, with the aid of a binary PSO, the clusters are chosen, and finally, the centroids of the selected clusters are refined by the k-means algorithm. Results showed that DCPSO was able to generate the appropriate number of clusters for the images used for the test.

A dynamic clustering using combinatorial PSO was presented by Masoud et al. [148]. The proposed model, called CPSOII, was implemented to automatically find the best number of clusters, as well as group or partition the data effectively. As a preprocessing step, the model used a renumbering approach and so extended the PSO operators in order to improve population diversity, quality of solutions, and convergence speed. Further, CPSOII used the variance ration criterion (VRC) and DB index to evaluate its performance. Experimental results showed that CPSOII, when compared with three other algorithms, achieved superiority in terms of effectiveness and robustness. The authors suggested that for future study, CPSOII could be integrated with multi-objective PSO to improve its performance.

In the study carried out by Ling et al. [138], their proposed method, called PLDC, was able to estimate the number of clusters automatically. Their method measured the local compactness of each cluster by the local density function, which makes the PSO drift toward maximizing compactness, thereby avoiding many clusters to be identified during evolution. A distance constraint that was based on local fitness mechanisms and a partition measurement were also incorporated to maintain diversity in the population and provide good performance. The performance of PLDC, when compared with six other state-of-the-art methods using the Rand index (RI), showed that it was able to precisely and appropriately determine the number of clusters, as well as achieve a better grouping. The method, however, could not handle outliers that were detected in the datasets.

Kuo and Zulvia [131] implemented an improved particle swarm optimization for automatic data clustering. Their proposed algorithm, called automatic data clustering using PSO (ACPSO), addressed two main issues in automatic clustering. The first section addressed the problem of determining the number of clusters, while the second section handled the representation of the cluster centroids. Further, they employed a sigmoid function to handle infeasible solutions and then used the k-means algorithm to adjust the cluster centroids. Their experimental results showed that their proposed ACPSO algorithm outperformed three other related algorithms when compared in terms of accuracy and consistency.

Nanda and Panda [160] proposed a multi-objective immunized PSO algorithm (MOIMPSO) to classify actions of 3D human models. Their proposed algorithm provided a suitable Pareto optimal archive for unsupervised problems by automatically evolving cluster centers and simultaneously optimizing two different objective functions. Also, from the Pareto optimal archive, a single best solution that satisfies users' requirements was provided. The resulting analysis showed that the performance of the proposed algorithm was better in terms of result accuracy and computational time when compared to results from other related algorithms.

A kernel-based modified PSO (kernel MEPSO) by Abraham et al. [3] and a multi-elitist PSO (MEPSO) by Das et al. [51] were developed and implemented for automatic clustering of complex data. The studies are similar, and the two sets of authors proposed two algorithms in which, instead of using the conventional square-measure distance approach, they adopted a kernel-induced similarity function. This adaptation enabled data that are non-separable in its original form to be clustered into homogenous groups in a high-dimensional feature space transformation. Comparison of the kernel_MEPSO with other algorithms showed that, for some test cases, it is statistically significantly superior to them.

Kao and Chen [113] implemented a hybrid PSO for automatic clustering for generalized cell formation, called

PSOAC. The method adopted an integer number and set of real numbers, which were then used to encode the number of machine cells, a discrete PSO was used to search for the number of machine cells, and a continuous PSO was used for the machine clustering. The method searched for the number of machine cells in two ways: either by random selection or by inheritance of past best results. The former option aids PSOAC from becoming trapped within local optima, and the latter allows the proposed method to exploit the best machine cell solution found, as well as reduce infeasible solutions from occurring, thus saving computational time. Experimental results showed that PSOAC was able to determine the number of clusters automatically, and it also assigned the most suitable routing process for each part. Further, results showed that PSOAC had more time efficiency with large-sized problems than did other compared methods from the literature.

Another study on a multi-objective approach to automatic data clustering was conducted by Abubaker et al. [11] by integrating a multi-objective PSO with multi-objective simulated annealing (SA) for automatic data clustering. The proposed method, called MOPSOSA, simultaneously optimized three cluster validity indices in order for the suitable number of clusters to be established, as well as for appropriate partitioning. The first validity index (DB index) focused on Euclidean distance, the second (Sym index) on the point symmetry distance, while the third (Conn index) was centered on short distance (i.e., a relative neighborhood graph concept). MOPSOSA addressed the issue of automatic identification of suitable clusters and partitioning of the identified clusters in the datasets. Comparing results from MOPSOSA with those from six other automatic clustering algorithms showed MOPSOSA's superiority in terms of accurate results, obtaining the correct number of clusters, handling overlapped datasets, datasets with various irregular shapes and datasets that contained many clusters.

A fast and high-performance PSO algorithm called MPREPSO was implemented by Tsai et al. to handle the time complexity associated with the classical PSO [221]. The proposed method adopted two other operators in addition to those of the classical PSO. The first, a pattern reduction operator, determined whether a pattern could be regarded as static, thus compressing it, and the second, a multi-start operator, improved the quality of the final results obtained. Experimental results showed that MPREPSO reduced the computational time and also provided better results when compared with results from five other existing algorithms.

Recently, Sharma and Chhabra [202] introduced a mutation operator into a hybrid PSO for sustainable automatic data clustering, calling their hybrid HPSOM. The proposed HMPSO was used to group data generated from different networks, which are usually dynamic and heterogeneous, and where the number of clusters is also unknown in advance. The HPSOM was further extended to AHPSOM, in order to generate and readjust the clusters automatically over the mobile network devices, thereby facilitating the generation of sustainable clusters. The performance of HPSOM was compared with some known evolutionary clustering methods. The effectiveness of AHPSOM was evaluated using cluster numbers, inter- and intra-cluster distances, ARI, and F-measure, as well as being compared with other state-of-art automatic clustering techniques. Results showed that the proposed algorithms were both superior to the competing algorithms, in terms of well-separated, compact, and sustainable clusters.

Rana et al. [179] proposed a hybrid sequential approach, which integrated PSO into a sequence with the k-means algorithm for data clustering. The proposed algorithm was able to handle the limitations of both the PSO and k-means algorithms and provide improved quality of clustering while also avoiding the solutions from being trapped by local optima. The authors used the quantization error, intra-cluster distance and inter-cluster distance to evaluate the quality of the clustering solutions. Further, in the proposed approach, the PSO was used to start the clustering search process due to its fast convergence rate. Then, the results from the PSO algorithm were fine-tuned by the k-means algorithm. The performance of the proposed hybrid sequential algorithm was compared with that of four other existing algorithms, and results showed that it generated better clustering solutions than the counter algorithms.

4.4.2 Firefly algorithm (FA)

The firefly algorithm has been widely and successfully applied to solve data sorting problems due to its several benefits, which include robustness, efficiency, ability to handle problems in different fields and domains, including those that are NP-hard, and versatility. Comprehensive reviews of the FA were carried out by Fister et al. in 2013 and 2014, which discuss the diverse areas and broad spectrum of real-world applications where the algorithm has been successfully applied with satisfactory results. In both works, the authors went so far as to suggest future directions for the algorithm, FA. Although the FA has been studied extensively and shown to have good track records across diverse domains, its implementation in data clustering and automatic data clustering scopes is, however, still very scanty. Few works were identified for this review on the application of the firefly algorithm to data clustering, and even fewer previous studies were found concerning its application to automatic data clustering. These two applications are discussed next.

A performance study on the firefly algorithm (FA) for data clustering was carried out by Senthilnath et al. [201]. They acknowledged the strengths of FA and applied classification error percentage (CEP) to generate optimal cluster centroids. The standard FA was implemented for data clustering by focusing primarily on the attractiveness, light absorption, population size, and distance; CEP was applied in order to check the method that generated the optimal number of clusters. Further, FA was compared with ABC, PSO, and nine other clustering methods. Results showed that the classification efficiency of FA is superior to others in terms of reliability, efficiency, excellent global performance, and robustness.

Hassanzadeh and Meybodi in 2012 presented a new hybrid approach based on FA and k-means for data clustering [95]. The proposed model called K-FA was implemented such that FA was first used to find cluster centroids for a user-specified number of clusters, and then, the FA was extended using the k-means algorithm. The extension of the algorithm was aimed at refining the cluster centroids that had been detected by FA; also, global optima were used to improve the standard FA. Experimental results showed that K-FA outperformed three other clustering algorithms in terms of better efficiency and a decrease in intra-cluster distances, which allowed the k-means method to have a proper initialization.

Banati and Bajaj in 2013 conducted a viability performance analysis of FA for data clustering. The proposed method, called FClust, which is centroid-based, adopted the flashing behavior of fireflies as the objective function of the clustering problem to obtain the optimal solution. The performance of FClust was evaluated using two statistical criteria, namely trace within criteria (TWR) and variance ratio criteria (VRC) [171]. A comparison of the simulation results of FClust with those from the standard PSO and DE showed that FClust achieved the best mean fitness and standard deviation values on the VRC measure. Further, the quality of solutions obtained by FClust was evaluated using the number of function evaluations via the run length distribution (RLD) approach [99]. RLD for FClust showed that when compared with results from the same algorithms as in Banati and Bajaj [28], it achieved the best function evaluation value and a faster convergence rate.

In 2015, Kaushik and Arora integrated FA with an improved genetic algorithm [120]; the hybrid was called FGA. The proposed model selects its initial population from a population pool, which is based on solutions from the firefly algorithm, i.e., the initial population is generated from the global best solutions of the firefly algorithm. FAG operates in two ways; first, the classical FA is applied to sets of a randomly selected initial population, which generates chromosomes of a set, and secondly, the chromosomes are then positioned in the mating pool from where they partake in the mutation and crossover operations of the genetic algorithm. Also, the initialization stage of FGA results in global optimization, which prevents the solutions from getting trapped within the local optima. The test results, when compared to the classical genetic algorithm and firefly algorithm, showed that FGA had better inter-cluster and intra-cluster distances and more satisfactory results.

Nayak et al. [162] implemented an improved FA, by incorporating a fuzzy c-means algorithm; the hybrids, called FAFCM and improved FAFCM, were used for real-world clustering datasets. These two hybrids addressed the shortfalls of the fuzzy c-means method: specifically, local optima entrapment and high sensitivity to initialization. FAFCM was designed with two stages: firstly a standard firefly algorithm with fuzzy c-means clustering, and secondly, an improved firefly algorithm with fuzzy c-means clustering. The first handled the limitations of the fuzzy c-means algorithm by minimizing the objective function, while the second phase refined the cluster centers that had been identified from the first phase, and it also helped in further minimization of the objective function. FAFCM performance was compared with that of three other clustering algorithms, and the results showed that FAFCM had consistent results over the test datasets, a faster convergence speed, as well as a minimized objective function, although the number of clusters was predefined before centroid assignment by FAFCM.

An efficient hybrid method based on a modified FA and a dynamic k-means algorithm for data clustering was developed by Sundararajan and Karthikeyan [213]. The proposed algorithm is called a hybrid modified firefly and dynamic k-means algorithm. The dynamic k-means algorithm was incorporated so that it could adequately find the optimal number of clusters during execution time, as well as improve the cluster quality and optimality. Since the model works well for a predefined number of clusters, at each iteration it determines new centroids by the cluster counter increasing by one until the required cluster quality is achieved. Experimental results showed that the proposed model found better clusters quality in less time with increased optimality, than did the compared algorithm.

4.4.3 Artificial bee colony (ABC)

Karaboga and Ozturk [117] implemented the classical artificial bee colony algorithm for data clustering, applying it to handle the problems that are present in the classification of some benchmark datasets. The datasets were divided into training and testing datasets, and the classification error percentage (CEP) was used to evaluate the percentage of the test datasets with incorrectly classified patterns. Further, the ABC was used to minimize the sum

of the Euclidean distance between two data points and centroids for all the training datasets. The performance of ABC was compared with that for the PSO and nine other clustering methods, and results showed that ABC was able to classify the datasets more successfully than other competing algorithms.

A hybrid between the discrete ABC with a greedy randomized adaptive search procedure called hybrid DABC-GRASP was proposed and implemented by Marinakis et al. [147] to optimize the clustering solution with known (user-defined) number of clusters. The algorithm comprised of two stages, firstly, feature selection and then clustering solution. In the proposed model, the feature selection is addressed by the discrete ABC, while the greedy randomized procedure solves the clustering phase. Experimental results showed that the hybrid DABC-GRASP had a better performance, with the largest number of correct clusters, than the other eleven compared algorithms, although the number of clusters was defined a priori.

Tran et al. [220] proposed a hybrid clustering method based on hybridizing an enhanced ABC with the k-means algorithm, the new approach being called EABCK. In the method, ABC was enhanced by a new mutation operator, which was guided by the best global solution obtained from the enhanced ABC alone, without the k-means algorithm (EABC). After that, the global best solution in each iteration was updated using the k-means algorithm. EABCK was compared with six other clustering algorithms, and the results showed that EABCK outperformed the others in terms of convergence speed and accuracy. Since the number of clusters was predefined by the method, the authors suggested that for future research, the proposed method should be applied to solve high-dimensional datasets, as well as an automatic clustering problem.

Similarly, the authors Karaboga and Ozturk [117], together with Ozturk et al. [168], devised, devised an improved binary ABC for dynamic clustering, called IDisABC. The discrete ABC has the shortfall of depending on measuring the similarity between binary vectors through the Jaccard coefficient; the IDisABC addresses the shortfall by using all the similarities to efficiently enhance the discrete ABC through the genetic components. The crossover and swap operators are then used on the newly generated solutions according to the similarity cases. The VI index and correct classification percentage (CCP) were used to evaluate the efficiency of the proposed model and the quality of clustering results. The performance analysis of the method was compared with that of five other clustering algorithms, and the results clearly show that IDisABC obtained the optimal number of clusters automatically, with good quality of solutions and fast convergence rates. Further, IDisABC was applied to image segmentation, where it was able to achieve optimal clusters with reasonable classifications for the set of images used.

Kuo et al. [129] integrated ABC with kernel clustering to devise a method called AKC-BCO. When their proposed algorithm was implemented to solve automatic data clustering, it determined the appropriate number of clusters as well as correctly assigning data points to clusters. They accomplished this by using a kernel function, which increased the clustering capability of the ABC algorithm. Experimental results showed that AKC-BCO was superior to the three other algorithms chosen for comparison in terms of faster convergence, no local optimum entrapment, and better and more stable clustering results. The AKC-BCO was further applied to a real-life case of a prostate cancer prognosis system, and results revealed that AKC-BCO clustered patients tested data appropriately and were also able to predict survival chances for patients diagnosed with the disease.

Similar to the study in Kuo et al. [129], Kuo and Zulvia [128] proposed improving the ABC by incorporating the k-means algorithm for automatic data clustering and this was further applied to customer segmentation [128]. The hybrid method, called iABC, improved the classical ABC by directing the movements of bees to better positions and providing better initial centroids for the clusters defined by the k-means algorithm. These centroids then provide the onlooker bees with an improved method of finding a better solution faster than was possible with the first onlooker bees' movement. Simulation results showed that iABC, which used the VI index to evaluate its effectiveness, was superior to seven other automatic clustering algorithms in terms of better and steadier solutions, although with a relatively high computational time. When the method was further applied to customer segmentation, results showed that iABC classified customers appropriately into ten different clusters so that organizations would be able to identify potential customers and design the most suitable marketing strategy to bring new customers onboard, thereby increasing profit.

4.4.4 Ant colony optimization (ACO)

Pacheco et al. [169] addressed the problem of automatic grouping of data by implementing an automatic clustering method based on the collective intelligence of ants in the ant colony optimization (ACO) algorithm. The proposed method, called Anthill, made use of adaptive strategies to speed up the process of building the solution. The silhouette index and visual inspection were used to evaluate the performance of the proposed model and also to assess the quality of the generated clusters. Experimental results on the proposed Anthill algorithm indicated excellent performance when compared to results from three other existing

methods, and it obtained significant partitioning of the found clusters.

Niknam et al. [165] proposed an efficient hybrid evolutionary algorithm, called ACO-SA, that combined ant colony optimization (ACO) and simulated annealing (SA) algorithms to solve the clustering analysis problem. The proposed model, which is applicable only when the number of clusters is known a priori, was intended to find optimal or near-optimal solutions for clustering problems. Simulation results of the ACO-SA showed that the hybrid algorithm outperformed the basic SA, ACO, and k-means, individually, for the partitional clustering problem in terms of robustness and efficiency.

Liu and Fu [142] proposed the ESacc clustering algorithm, which was based on ant colony optimization to solve unsupervised clustering. Their proposed method iteratively keeps the best solutions stochastically. The proposed method made use of the Dunn, Jaccard, Folks and Mallows, and Rand indices to evaluate the optimal number of clusters. Computational results obtained from ESacc were compared with those from the original Sacc algorithm, and it showed that ESacc had lesser run time, better clustering effect, more performance stability, and greater efficiency.

The paper presented by Boryczka [35] used a modification of Lumer and Faieta's algorithm for data clustering, called ant-based clustering algorithm (ACA). The approach mimics the clustering behavior that had been observed in real ant colonies. It improved clustering convergence and the spatial separation between clusters. Further, the algorithm was able to detect the number of clusters automatically without the prior need for information about the data objects. They used Euclidean distance, cosine measure, and Gower measure to evaluate the quality of the clustering solutions so obtained. Although ACA dealt with numerical databases, it did not require any information about the feature of the clusters or the number of clusters. Also, the ACA algorithm was able to obtain comparative clustering results when compared with another existing algorithm. The authors, however, suggested that the ACA algorithm could be hybridized with other metaheuristics to improve its performance and efficiency.

4.4.5 Symbiotic organism search (SOS)

A new swarm metaheuristic algorithm, called symbiotic organism search (SOS), was implemented for automatic data clustering by Zhou et al. [245]. SOS mimics the symbiotic interaction of organisms needed for survival and proliferation in an ecosystem; it was proposed to address the shortfalls of the k-means algorithm. SOS adopts three biological interaction phases, namely mutualism, commensalism, and parasitism. In the mutualism phase, organisms benefit from each other without either of them impeding the other; for commensalism, only one organism benefits from the interaction but does not cause any harm to the other; while in the parasitism phase, one organism benefits from the interaction while causing harm to the other. The simulation results from SOS were compared with those for seven other metaheuristic algorithms, and it was shown that SOS outperformed the others in terms of quality of solutions, accuracy, stability, and convergence speed. The computational time, however, was higher than for any of those compared, which is due to it having the most function evaluations.

4.4.6 Bacterial evolutionary algorithm (BEA)

Das et al. [52] proposed a bacterial evolutionary algorithm for automatic data clustering (ACBEA) in Das et al. [52]. The proposed method, according to the authors, was inspired by biological microbial evolution, which uses the operations of bacterial mutation (it mimics the process occurring at the genetic level in bacteria, which improve the chromosome parts) and gene transfer (information is exchanged between chromosomes in the population). The operators were then modified to handle the variable length of chromosomes that encode different clustering classifications. The CS index was used to evaluate the performance of the proposed algorithm, which was then compared with that of two other clustering algorithms, to show that ACBEA was superior in terms of result accuracy.

4.4.7 Grey wolf optimizer (GWO)

A grey wolf optimizer (GWO)-based automatic clustering for satellite image segmentation was proposed by Kapoor et al. [114]. The algorithm was further applied to two satellite images from New Delhi, and its performance was evaluated using the DB index, inter-cluster distance, and intra-cluster distance. Computational results showed that GWO is computationally efficient, and its accuracy is superior to those of the other three compared clustering algorithms. Furthermore, the result of the image segmentation showed that GWO reveals the growth of urbanization and infrastructure and a decrease in green forest vegetation in the surrounding areas of New Delhi.

4.4.8 Sine–cosine algorithm (SCA)

More recently, Elaziz et al. [64] proposed an automatic data clustering algorithm, called ASOCSA, which is based on the hybridization of the sine–cosine algorithm (SCA) with the atom search optimization (ASO). The main objective of the proposed algorithm is to automatically find the optimal number of centroids and their respective positions to minimize the CS index. To achieve this,

ASOCSA improves on the original ASO by adopting SCA as the local search operator. The effectiveness of the proposed clustering algorithm was evaluated using different validity indices, such as the Dunn index, Silhouette index, Davies–Bouldin index, and the Calinski–Harabasz index. ASOCSA showed superiority over five other existing clustering algorithms in terms of robustness and efficacy.

4.4.9 Cuckoo search (CS)

Goel et al. [84] proposed a cuckoo search clustering algorithm (CSCA). The CSCA was able to group a set of data points into clusters having similar attributes. The algorithm was also able to work in an unsupervised way without having to consider the class of the data points during the clustering process. The Davies–Bouldin (DB) index was used to evaluate the performance of the proposed method. Experimental results showed that the CSCA algorithm demonstrated high accuracy. The authors further applied the CSCA algorithm to a satellite image and used it to extract images of water from a real-time multispectral remote sensing image.

Senthilnath et al. [200] performed a comparative study based on three nature-inspired techniques; the genetic algorithm (GA), particle swarm optimization (PSO), and cuckoo search (CS) were implemented and analyzed for the performance in the clustering problem. The cuckoo search exploits the Lévy flights mechanism, which is heavy-tailed and helps in covering the output domain efficiently. To evaluate the clustering solutions obtained by the three algorithms, the authors employed the classification error percentage (CEP) and the statistical significance test. In the experimental results, compared to the PSO, the GA took less time, but the CS algorithm required considerably less time than did the PSO, with the GA also being more time efficient than the PSO. The heavy-tailed property of the Lévy flights also helped the solutions to converge quickly, thus increasing efficiency.

Bouyer et al. [36] implemented a hybrid approach based on integrating the cuckoo search and differential evolution algorithms for data clustering and called the hybrid HCSDE. The HCSDE algorithm firstly initializes a random population. Then, the cuckoo search uses the Mantegna Lévy distribution to produce new nests and also boost the local search capability. As a validity metric to evaluate the performance HCSDE the authors used the intra-cluster distance measure (for an internal quality measure) and the error rate (ER) (for an external, internal quality measure). Experimental results were compared with those for six other algorithms, which showed that HCSDE was superior in terms of convergence speed, accuracy, and better total within-variance value. However, in some instances, the HCSDE algorithm became trapped in local minima. Hence, the authors proposed this shortfall should be a focus of future research.

4.4.10 Bat algorithm (BA)

Jensi and Jiji [110] implemented a modified bat algorithm, called MBA-LF, for data clustering. In their work, they employed the Lévy flight mechanism to accelerate the movement and foraging abilities of bats in other to enhance the search process. Further, the Lévy flight was used to improve the quality of the clustering results. They used the Euclidean distance to evaluate the distances between the clusters that had been obtained. The computational results, when compared with those from three other existing algorithms, showed that MBA-LF achieved better clusters for the test data objects, it escaped entrapment of local optima, and it effectively explored the search space.

4.4.11 Bee-inspired algorithm (BeeA)

A new encoding scheme based on the bee-inspired algorithm was called cOptBees when presented by Cruz et al. [47]. The proposed algorithm employed an encoding scheme, whereby each bee represented a prototype of the generated clusters. The proposed method was able to generate and maintain the diversity of solutions by finding multiple suboptimal solutions in a single run. Furthermore, the method explored the multimodality feature that is associated with bee colonies. They used entropy, classification error percentage (CEP), purity, and the silhouette index as the fitness functions to evaluate the quality of the clustering solutions obtained and the performance of the algorithm. The test results that were presented showed that when cOptBees was compared to five other algorithms of bi-dimensional and n-dimensional datasets, cOptBees obtained better performance than the other competing algorithms. It was able to find high-quality cluster partitions without needing to know the appropriate number of clusters in the dataset.

4.5 Clustering using the plant-based algorithm

4.5.1 Flower pollination algorithm (FPA)

In the study carried out by Wang et al. [228], a flower pollination algorithm (FPA) with bee pollinators was proposed to solve the clustering problem. The proposed method called BPFPA employed the discard pollen operator and the crossover operator to increase diversity in the population, and it enhanced the local search ability by using the elite-based mutation operator. The pollens represent the centroids of the predefined clusters. These operators were incorporated to address the local

entrapment problem and the poor explorative ability of the basic FPA, thus, enhancing the explorative ability, improving the convergence speed, as well as increasing the diversity in the population. Experimental results were compared with those from six other metaheuristic algorithms and showed the BPFPA's superiority in terms of a higher level of stability, higher accuracy, and faster convergence, indicating it was more competitive in solving clustering problems. The authors suggested that BPFPA could be extended to determine the optimal number of clusters dynamically and that its applicability to higher-dimensional problems should be investigated.

Agarwal and Mehta [12] studied the application of a modified flower pollination algorithm (FPA) to solve the data clustering problem. The objective function of the proposed MFPA-C is to maximize both intra-cluster similarity and inter-cluster dissimilarity. The performance of MFPA-C was compared to that of three other clustering algorithms, which showed that it achieved more promising results with consistent performance than did the others.

4.6 Clustering using breeding-based algorithms

4.6.1 Differential evolution (DE)

Das et al. [50] proposed an improved differential evolution (DE) algorithm for the automatic clustering of unlabeled datasets, called ACDE. The proposed algorithm was able to find the optimal number of clusters automatically and was applicable for high-dimensional datasets. Further, the performance of the proposed method was validated with the CS-index [40] and DB-index [54]. From the statistical analysis of experiments, ACDE outperformed the other compared state-of-the-art algorithms, including the classical DE clustering algorithms, although it did not win in all the instances.

Lee and Chen [136] implemented an improved differential evolution (ACDE-O) algorithm using a crisp number of oscillations for automatic clustering. The oscillation mechanism was used to improve the search possibility of finding more possible clusters in the case where the number of initial clusters was inadequate as a result of bad clusters. Their test results, when compared to those for another clustering algorithm, showed that ACDE-O was better at finding a more suitable number of clusters.

Saha et al. [192] implemented a differential evolution (DE)-based fuzzy clustering for automatic cluster evolution (ADEFC), where they used the Xie–Beni index to assign points to different clusters. The Xie–Beni index was also used as the validity measure for the cluster partitioning, and then, the centers of the clusters were encoded in vectors represented by 0's and 1's. Value 1 of the masker cell determines that the encoded center of the vector can participate in the fuzzy cluster, and otherwise for value 0. The superiority of the proposed technique, as compared with other algorithms, showed that ADEFC consistently performed better than did the other clustering techniques.

Maulik and Saha [150] then extended their study and proposed a new real-coded modified DE-based automatic fuzzy clustering method, called MoDEAFC [149]. The method was implemented to address the issues of proper cluster numbering, as well as good partitioning. It extended the ADEFC by using a fixed-length representation to encode the centroids of each individual and a masker to activate or deactivate a centroid. The Xie–Beni index was also used in assigning proper points to different clusters by minimization, while also considering the Euclidean distance. A new mutation operator, which was used to replace the mutation operator in the classical DE, exponentially decreased within the range of [1, 0.5]. Experimental results showed that MoDEAFC consistently performed better in terms of the accurate number of clusters than did the four other compared algorithms. Further, its application to IRS satellite images of Calcutta and Mumbai showed efficiency in the image segmentation.

Another multi-objective application of DE to automatic fuzzy clustering, called MODE, was presented by Suresh et al. [214]. These authors used a real encoded centroid-based scheme for their search variables, which also contained the variable number of cluster centers. Further, the best solutions from the Pareto optimal set were obtained using a gap statistic. MODE was compared with four other state-of-the-art multi-objective methods, and results showed that it produced better clustering results than did the others.

Kundu et al. [127] implemented GADE, which is an integration of a multi-objective DE-based algorithm with genetic algorithm (GA), for automatic clustering. The proposed model incorporated some operators of the classical genetic algorithm and used the XB and FCM indices as the objective functions to be optimized. Computational experiments showed that GADE, when compared with the results obtained from two other algorithms, achieved the best performance in terms of adjusted Rand index and silhouette index with an equal number of runs for all the generations. Zhong et al. [243] also optimized the XB and FCM indices for multi-objective DE by utilizing a two-layer fuzzy clustering technique, called AFCMDE.

4.6.2 Genetic algorithm (GA)

A genetic algorithm (GA) that was modified to improve the accuracy of classification in cluster analysis was devised by Wang and Wu [229]. Called the chaotic genetic algorithm (CGA), the proposed method adopted the ergodic property of chaotic phenomena to optimize the initial population in

order to speed up the process of selection, crossover, and mutation operators, as well as the convergence property of the genetic algorithm. Experimental results showed that the proposed CGA attained global cluster centroids and greatly improved the amplitude of operation than did the three other existing models. However, CGA was not able to automatically assign clusters.

Dutta et al. [62] implemented a mixed feature multi-objective GA with k-means for data clustering, called MOGA. MOGA addressed the issues of continuous and mixed features that are present in datasets. It simultaneously optimized the intra-cluster distance (homogeneity) and inter-cluster distance (separation) by using a unique distance feature, which was sufficient for both the continuous and mixed features. Experimental results showed that MOGA achieved accurate cluster centroids. The model, however, was not able to deal with unseen data points and missing features and did not obtain the optimum number of clusters.

The earliest attempt at using genetic algorithm for automatic data clustering called CLUSTERING was presented by Tseng and Yang in 2001 [223]. The proposed approach addressed clustering in three ways: firstly, a nearest-neighbor clustering method was used to group together data points that are similar in order for small clusters to be obtained. Secondly, the proposed method merged the set of small clusters into larger ones, which was done by using a weighted difference between the BGS and WGS indices that define the fitness function. Lastly, the appropriate cluster partitioning was done using a heuristic approach. Simulation results showed that CLUSTERING outperformed three other compared algorithms.

An automatic clustering algorithm that used the genetic algorithm, called AGCUK, was implemented by Liu et al. in 2011. A noisy selector and division-absorption mutation operator were used to create a balance between selection pressure and population diversity. The model also adopted a cluster-based representation, whereby an individual represents a real-coded chromosome of variable length, which is randomly selected. Further, they used the DB index to compute the fitness of an individual. AGCUK outperformed four other automatic clustering algorithms in terms of obtaining the optimum number of clusters and lower misclassification rates.

Agusti et al. [16] presented an automatic clustering method based on the grouping genetic algorithm (GGA). In the proposed GGA, they applied a type of partition-based encoding scheme, and the DBI index was used as the objective function. The functions of the GGA were, firstly, the crossover operation between the groups, and offspring was arranged according to the generated groups, then the mutation operator was applied by merging and splitting clusters, and finally, the local search was used to find the local optimum close to the solution. Although GGA was not tested on real-world datasets, the representation scheme, however, resulted in high time complexity for a high volume of data.

Similar to the study in [16], Salcedo-Sanz et al. [195] focused on using a fuzzy version of the DBI index, and their encoding scheme was composed of the membership matrix and group members for their proposed GGA-based method. However, the encoding scheme they adopted also resulted in high time complexity as a result of the data size.

Also, similar to the studies carried out in [16, 195], Raposo et al. implemented an automatic clustering method using a genetic algorithm with new solution encoding and operations called automatic clustering genetic algorithm (ACGA) [184]. The method adopted a new solution encoding-based scheme that had not been tested with the classical GA, and new genetic operators (two new mutations and one new crossover) were developed to ensure that there was high diversity in the population. The CH index was used to test the effectiveness of ACGA; experimental results showed that ACGA outperformed two classical algorithms in terms of better convergence and higher fitness function values.

In 2012, He and Tan proposed a two-stage genetic algorithm for automatic clustering, which they called TGCA. The model employed the selection and mutation operators of the classical genetic algorithm but changed the probabilities of these operators according to the consistency of the number of clusters present in the population. TGCA focused firstly on searching for the best number of clusters and then gradually moved to find global optimal centroids. The model was evaluated using the CH index as the fitness function. The efficiency of TGCA was shown when it was compared with three automatic clustering algorithms. It was evident that TGCA did better in terms of automatically finding the correct number of clusters and clustering accuracy. Two limitations of the method are, firstly, the quality of the final clustering solution may not be good enough due to failure to capture the chromosomes representing all clusters as a result of the random selection, and, secondly, the method does not rearrange the chromosomes before the crossover operation.

To address these limitations, Rahman and Islam [175] proposed a different GA-based approach, called GenClust, that is capable of identifying the right chromosomes using a novel initial population selection approach, chromosomes rearrangement, twin removal operator, and a fitness function and then also finding the right number of clusters automatically. GenClust avoided a user-defined number of clusters while achieving clusters of high quality. The superiority of GenClust over five other existing approaches proved that it was able to rearrange chromosomes with the aid of k-means, which produced better results than that of

He and Tan's algorithm [97]. Further, GenClust's initial selection of a population was based on a deterministic and random process. GenClust, however, was not able to handle datasets of high dimension due to the increased complexity of GA.

4.6.3 Invasive weed optimization (IWO)

The first attempt at using IWO for automatic data clustering was proposed by Chowdhury et al. in 2011. The algorithm made use of a modified Sym-K index, as the fitness function, in order to evaluate the appropriate partitioning of the datasets. The performance of the algorithm was compared with that of three other algorithms, and results showed that IWO partitioned data better than they did, which was evident in the Minkowski scores of the real-life datasets. However, the optimal solutions were derived with a minimum number of populations, which also reduced computational time.

Zhao and Zhou [242] proposed an improved kernel possibilistic fuzzy c-means algorithm for clustering analysis problem called IWO-KPFCM, which was based on the IWO algorithm; the proposed model was designed to handle the issues arising from both the fuzzy c-means and the possibilistic fuzzy c-means algorithms. IWO-KPFCM, at first, uses the basic IWO to find the optimal solutions to the initial centroids, and then, maps the input data from the sample space into the high-dimensional feature space by using the kernel approach. Further, the sample variance is infused into the objective function to measure the degree of data compactness, and then, the improved algorithm clusters the data. Results clearly showed that the proposed method had increased cluster accuracy, faster convergence speed, and a more robust ability to repel noise and outliers than did the two other compared algorithms.

Liu et al. [139] implemented a multi-objective IWO algorithm, called MOIWO, to solve the clustering problem [139]. A feedback-update mechanism was employed to maintain the diversity of the number of clusters during the iteration process. The feedback-update mechanism helps the solution set to accommodate all the types of cluster numbers that are identified. Further, the silhouette index was used to evaluate the efficiency of the proposed algorithm as well as to select the best solution. Experimental results showed good performance of the multi-objective IWO, although it was not able to detect the optimal number of clusters automatically.

4.7 Clustering with the social human behavior-based algorithms

4.7.1 Teaching learning-based optimization (TLBO)

Satapathy and Naik [197] developed a TLBO algorithm, which they used to find the centroids of a user-specified number of clusters. The proposed method made use of two phases of the TLBO algorithm, namely the teacher phase and the learner phase. The teacher phase is when learning from the teacher occurs, while in the learner phase learning comes from the interactions occurring between learners. The learner phase corresponds to the fitness function, while the teacher phase relates to the best solution. The proposed algorithm halted once the user-specified number of iterations was exceeded. Experimental results showed that the proposed TLBO method had more potential to find appropriate centroids to the predefined number of clusters than did the two other compared algorithms.

In another related study, Sahoo and Kumar [193] proposed two different modifications for the TLBO method, to enhance its performance in clustering domains. The modifications were, such that instead of random initialization, a predefined method was previously used to exploit initial centroids. Also, the technique could handle data vectors that had gone out of the boundary conditions. The performance of the proposed modified TLBO was evaluated based on quantization error, intra-cluster distance, and inter-cluster distance. Further, a comparison was made between the modified method and three other algorithms, including the basic TLBO algorithm. From the experimental results, the proposed modified TLBO method showed more accurate results than did the others.

Similar to the study carried out by Das et al. [50] (Sect. 4.4.1), Murty et al. [159] implemented a teaching learning-based optimization (AUTO-TLBO) for automatic data clustering. The effectiveness of their proposed method was evaluated with the CS validity index and compared with that of four other existing algorithms. Results showed that AUTO-TLBO was superior to the other techniques in terms of optimally finding the number of clusters automatically and a fast convergence rate, although their method did not win for all the test instances.

4.7.2 Imperialist competitive algorithm (ICA)

A recent application of the imperialist competitive algorithm was in applying it for the first time to solve automatic data clustering problems, which were carried out by Aliniya and Mirroshandel in 2019. The proposed algorithm, called automatic clustering using an imperialist competitive algorithm (AC-ICA), was based on a novel

combinatorial merge-split method. In the proposed method, the authors introduced a change at the assimilation step of colonies in order to increase the exploration ability of the colonies' movement. Furthermore, a new method was provided to change the number of clusters by combining a random and homogeneity-based merge-split approach. Also, for the re-initialization of empty centroids, an efficient method based on density was adopted. To address the automatic clustering problems, the initialization and imperialist competition steps were changed. AC-ICA used the functions of purity, entropy, Rand index (RI), and adjusted Rand index (ARI), to determine the fitness values and quality of the solutions obtained. Computational results were compared with the imperialist competitive algorithm (ICA) and its three other variants, and they showed that AC-ICA achieved a more accurate number of clusters, better convergence rate, higher accuracy of solutions and better homogeneity. Further, when AC-ICA was applied to face recognition, the results achieved were satisfactory.

4.8 Clustering with physics-based algorithms

4.8.1 Gravitational search algorithm (GSA)

Kumar and Sahoo [126] carried out a review study on the gravitational search algorithm (GSA), which is based on the theory of gravity, and its application to data clustering. The algorithm is able to solve large problems, including optimization problems, because it requires only two parameters to be adjusted and has the ability to find near-global optimum solutions. This ability allows the algorithm to provide better results when compared with other nature-inspired algorithms. The authors went further to discuss the variants of GSA and hybrid methods. As reported, hybrids based on GSA with other algorithms handled a more comprehensive range of problems, thus providing more robust solutions and enhancing the capabilities of GSA. Furthermore, GSA and four of its hybrid-based variants were compared with seven other algorithms, and results showed that two of the hybrid-based GSA algorithms obtained better quality of solutions, better computational time and more efficient convergence. Moreover, it was reported that the GSA was widely applied and reported in many publications in the areas of computer science and computing, as well as civil and mechanical engineering.

Kumar et al., in 2014, implemented a gravitational search algorithm for the automatic evolution of clusters and applied it to image segmentation (Kumar, Chhabra, and Kumar, [124]. The proposed model, called ACGSA, used a variable chromosome representation to encode the cluster centroids of different numbers for clusters. Further, two new operations of threshold setting and weighted cluster centroid were employed to refine the centroids. Experimental results demonstrated that ACGSA outperformed five other existing automatic clustering methods in terms of efficiency, determining the accurate number of clusters, best partition, and effectiveness. Moreover, its application to the automatic segmentation of both grayscale and colored images also showed its efficacy.

4.8.2 Harmony search (HS)

In 2016, Kumar et al. [125] developed a parameter adaptive harmony search algorithm (ACPAHS) for automatic data clustering. The authors used a real-coded variable-length harmony vector, which was able to detect the number of clusters automatically. Furthermore, the assignment of data points to different cluster centers was done using a new approach of weighted Euclidean distance, which was able to detect any type of cluster regardless of its geometric shape. The authors also applied their method to automatic image segmentation and compared it with four other existing clustering techniques. Experimental results showed ACPAHS outperformed other techniques in detecting the number of clusters automatically and gave better clustering result.

4.8.3 Black hole (BH) algorithm

Hatamlou [96] proposed another data clustering algorithm, called BH algorithm, which is also inspired by the black hole phenomenon. His proposed algorithm has two significant advantages over the compared algorithms. First, it has a simple structure and easy implementation. Second, it is free from parameter tuning issues. Further, in the proposed BH algorithm, the best candidate among all the candidates in the search space at each iteration is selected as a black hole, while all the other candidates are generated as the normal stars. The creation of the black hole is not a random process; instead, it involves one of the real candidates of the population. After that, all the candidates then migrate toward the black hole, based on their current location and a random number. The BH algorithm also has the ability for the black hole to absorb the stars that surrounds it. The performance of the BH algorithm is evaluated using the intra-cluster distance measure (for internal quality measure) and the error rate (ER) measure (for external, internal quality measure). Experimental results, when compared with those from four other existing algorithms, showed that the BH algorithm obtained higher-quality clusters than others. For future research on this work, the author suggested that the BH algorithm could be combined with other nature-inspired algorithms for greater effectiveness.

Abdulwahab et al. [2] addressed the issue of exploration capabilities that are associated with the original black hole

algorithm by introducing the Lévy flight mechanism. Their proposed algorithm, called Lévy Flight Black Hole (LBH), was able to optimize the performance of the original black hole algorithm with enhanced global search capacity to avoid being trapped in the local minima. In the proposed LBH, the movement of each star depends mainly on the step size that is generated by the Lévy distribution. Each star explores an area that is far from the current black hole when the step size is big, while, when the step size is small, the star explores an area that is near the current black hole. Further, the Lévy flight resolved the issue of global optimization and efficiently improved the search capability of the stars within the search space. The performance of the LBH algorithm was evaluated using the Euclidean distance measure. Furthermore, the performance of the LBH algorithm was compared with that of nine other clustering algorithms, and results showed that LBH clustered the data objects efficiently, escaped local optima entrapment, and effectively explored the search space more than the other competing algorithms did.

4.9 Clustering with miscellaneous sources of inspiration algorithms

4.9.1 Membrane computing

Peng et al. [173] published an automatic clustering algorithm inspired by membrane computing. In their method, a tissue-like membrane system with fully connected structures was designed to be the computing framework for automatically determining the most appropriate number of clusters as well as the optimal partitioning between clusters. The method also incorporated a modification of the velocity-position model that was developed as evolution rules based on its communication system, and the CS index was used to evaluate the efficiency of the proposed method. Comparison with three other existing algorithms showed that the proposed membrane algorithm effectively determined the most appropriate number of clusters, and displayed more robustness, better scalability, and better clustering effects on high-dimensional datasets than did the other competing algorithms.

4.9.2 Dynamic local search

Liu et al. [140] implemented an automatic clustering algorithm, called DLSIAC, which was based on a dynamic local search. The proposed algorithm was developed to automatically generate the number of clusters as well as proper partitioning for some selected datasets. Two techniques contributed to the performance of the proposed algorithm. Firstly, a dynamic local search strategy was employed to find the correct numbers of clusters and centroids rapidly. Secondly, to improve the accuracy of clustering and convergence rate, a modified clonal mutation scheme was introduced. Further, two mutation strategies were selected to produce new antibodies. The performance of the proposed algorithm was evaluated using the PBM index and was compared with the performance of five other state-of-the-art algorithms. Results showed that DLSIAC provided more consistent results than did the others. Further application of the algorithm to image segmentation showed that it kept the regional consistency and detailed image information. DLSIAC was also able to find the optimal number of clusters.

4.9.3 Action-set learning automata

Recently, Anari et al. [24] proposed continuous action-set learning automata, called ACCALA, for automatic data clustering, which was further applied to image segmentation. The continuous learning automata is an optimization tool that interacts with a dynamic environment and learns the optimal action from the feedback from the environment. The continuous learning automata further defined a suitable action set for each automation, thus significantly impacting the search behavior. Also, the proposed ACCALA was aimed at finding an action set for each automation automatically. The simulation results of ACCALA, in comparison with results from seven well-known automatic clustering methods, showed that it produced a more accurate number of clusters (compact and well-separated clusters) with greater efficiency. ACCALA also performed well in the segmentation of grey-scale and colored images.

4.9.4 Artificial immune system (AIS)

Younsi and Wang [235] published a new artificial immune system (AIS) algorithm for data clustering, which was based on the CLONALG algorithm, although the proposed method did not use the cloning process in CLONALG because it would have introduced intensive matching computation between antibodies and antigens. In the new method, the Euclidean distance was used to measure the distance between the antigens and cells (data vector), as follows. On the one hand, if the Euclidean distance obtained is less than or equal to the network affinity threshold (NAT), the value is selected and retained in the long-term memory. If, on the other hand, the distance measure is more than one, the cell identifies the antigen, and then, the closest cell is selected. Once all the cells have been presented to the antigen, the closest cell that has been selected eliminates the antigen. The cells can identify more than one antigen, and the resulting cells are the representation of the data that is being processed. Further, the

selected cells represent the data that is being compressed. Secondly, the algorithm was used for clustering and data visualization. All the cells that had been selected and stored in the memory were then matched to other similar/identical cells to form clusters. The NAT value in this phase is re-calculated using the memory cells and a new randomly chosen vector. If the value of the Euclidean distance is less than or equal to the NAT value, the cells that are close to each other are linked together to form clusters. Furthermore, the elimination of the antigens reduces the time complexity of repeatedly matching similar cells, which is used as the stopping criteria. The AIS algorithm was efficient enough in the clustering results that were obtained.

4.9.5 Metaheuristics for automatic clustering

José-García and Gómez-Flores [111] presented a survey study on some nature-inspired metaheuristic algorithms that had been used for automatic data clustering. The study also covered some cluster validity indices (CVIs) that were applied to evaluate the quality of automatic clustering algorithms as well as clustering solutions. The authors reviewed a total of 65 automatic clustering approaches, which were based on single-solution, single-objective, and multiobjective metaheuristic techniques. According to José-García and Gómez-Flores, the usage percentages of the three aforementioned techniques were recorded as 3%, 69%, and 28%, respectively. Among their findings, they reported that even though the single-objective clustering algorithms appears to be suitable and efficient for the tasks of grouping linearly separable clusters, many researchers had recently become more inclined to use the multiobjective algorithms to address nonlinearly separable problems.

More recently, Ezugwu [75] carried out a comprehensive study on the major nature-inspired metaheuristic algorithms that had been applied to solve automatic data clustering problems. The publication included a similar comparative study of several modified well-known global metaheuristic algorithms in order to evaluate their suitability for solving automatic clustering problems. Further, Ezugwu implemented several representatives of single and hybrid swarm intelligence and evolutionary algorithms, namely particle swarm differential evolution algorithm, firefly differential evolution algorithm, and invasive weed optimization differential evolution algorithm to deal with the task of automatic data clustering.

In Table 7, we present a summary of the state-of-the-art literature works carried out on the classical and automatic data clustering techniques. The summary presented in Table 7 focuses mainly on the contributing authors, clustering method used in implementing the referenced work, the application area of the implemented approaches discussed in the reviewed work, the type of clustering approach used, and finally the cluster validity index used in evaluating the quality of the proposed clustering algorithms.

5 Clustering similarity measures

One question that comes to mind once a clustering algorithm has classified a dataset is, how well does the classification fit the input data? This is pertinent because no single clustering algorithm is optimal, so given different conditions, different or even the same algorithm would yield different results. This makes it necessary to estimate how well a classification fits the underlying structure of the input dataset. The evaluation criteria for validating results of the clustering algorithm are a fundamental aspect of data clustering. Cluster validation criteria are usually classified as being either internal or external validation. However, there is a third classification called relative validation [31, 137].

5.1 Internal validation criteria

In real-world applications, the underlying structure of the dataset is usually unknown; therefore, there is no way of knowing the correct partitioning of the dataset. The internal validation criteria focus on the partitioned dataset (by the clustering algorithm); it measures the intra-cluster compactness and the inter-cluster separation. There are a variety of criteria that have been proposed, which are outlined below.

5.1.1 Sum of squared error

Sum of squared error (SSE) is one of the most popular cluster evaluation criteria; it is defined as follows:

$$\mathrm{SSE} = \sum_{k=1}^{K} \sum_{\forall x_i \in C_k} \|x_i - \mu_k\|^2 \tag{7}$$

where C_k is the set of all instances in the cluster k and μ_k is the vector mean of k. So, we need to look for a partition with the lowest SSE [93, 222].

5.1.2 Scatter criteria

For any kth cluster, the scatter criteria [59, 188] are given as follows:

$$S_k = \sum_{x \in C_k} (x - \mu_k)(x - \mu_k)^{\mathrm{T}}. \tag{8}$$

Table 7 Summary of some metaheuristics that have been applied to non-automatic clustering and automatic clustering

References	Clustering method	Application area	Clustering type	Cluster validity index (CVI)
Van der Merwe and Engelbrecht [224]	PSO	Cluster analysis	Non-automatic	Intra-cluster and inter-cluster distance
Zhao et al. [241]	PSO	Cluster analysis	Non-automatic	Intra-cluster distance
Chuang et al. [43]	ACPSO	Cluster analysis	Non-automatic	–
Silva Filho et al. [209]	FCM-IDPSO	Cluster analysis	Non-automatic	Rand index (RI)
	FCM2-IDPSO	Cluster analysis	Non-automatic	
Rana et al. [180]	BR-APSO	Cluster analysis	Non-automatic	–
Cura [48]	PSO	Cluster analysis	Non-automatic	–
Lashkari and Moattar [135]	ECPSO	Cluster analysis	Non-automatic	Purity index
Alswaitti et al. [22]	DPSO	Cluster analysis	Non-automatic	Dunn index
Duan et al. [58]	ABCPS	Cluster analysis	Non-automatic	MOC index Fukuyama and Sugeno index Weighted inter–intra-index
Atabay et al. [26]	IKPSO	Cluster analysis	Non-automatic	–
Nayak et al. [161]	GA-IPSO-K	Cluster analysis	Non-automatic	–
Huang et al. [102]	ACOR-PSO	Cluster analysis Continuous optimization problem	Non-automatic	–
Omran et al. [167]	DCPSO	Cluster analysis	Automatic	Dunn index (DI) Turi index
		Image segmentation		S_Dbw index
Masoud et al. [148]	CPSOII	Cluster analysis	Automatic	Variance ratio criterion (VRC)
		Combinatorial optimization problem		Davies–Bouldin (DB) index
Ling et al. [138]	PLDC	Cluster analysis	Automatic	RI
Kuo and Zulvia [131]	ACPSO	Cluster analysis	Automatic	VI index
Nanda and Panda [160]	MOIMPSO	Cluster analysis 3D human models	Automatic	–
Das et al. [51]	MEPSO, Kernel_MEPSO	Cluster analysis	Automatic	
Kao and Chen [113]	PSOAC	Cluster analysis Cell formation	Automatic	–
Abubaker et al. [11]	MOPSOSA	Cluster analysis	Automatic	DB index Symmetry (Symm) index Conn index
Tsai et al. [221]	MPREPSO	Cluster analysis	Non-automatic	Compact-separated (CS) index
		Image segmentation	Non-automatic	

Table 7 (continued)

References	Clustering method	Application area	Clustering type	Cluster validity index (CVI)
Das et al. [50]	ACDE	Cluster analysis	Automatic	DB index
		Image segmentation		CS index
Lee and Chen [136]	ACDE-O	Cluster analysis	Automatic	I index
Saha et al. [192]	ADEFC	Cluster analysis	Automatic	Xie–Beni (XB) index
Maulik and Saha [150]	MoDEAFC	Cluster analysis	Automatic	XB index
		Image segmentation		
Suresh et al. [214]	MODE	Cluster analysis	Automatic	XB index
				FCM index
				RI
				Silhouette index (SI)
Kundu et al. [127]	GADE	Cluster analysis	Automatic	XB index
				FCM index
Zhong et al. [243]	AFCMDE	Cluster analysis	Automatic	XB index
		Remote sensing		J_m index
Wang and Wu [229]	CGA	Cluster analysis	Non-automatic	
Dutta et al. [62]	MOGA	Cluster analysis	Non-automatic	DB index
Tseng and Yang [223]	CLUSTERING	Cluster analysis	Non-automatic	–
Liu et al. [143]	AGCUK	Cluster analysis	Automatic	DB index
He and Tan [97]	TGCA	Cluster analysis	Automatic	Calinski–Harabasz (CH) index
				RI
				Adjusted rand index (ARI)
Rahman and Islam [175]	GenClust	Cluster analysis	Automatic	XB index
Senthilnath et al. [201]	FA	Cluster analysis	Non-automatic	Classification error percentage (CEP)
Hassanzadeh and Meybodi [95]	K-FA	Cluster analysis	Non-automatic	Intra-cluster distance
Bannti and Bajaj [28]	FClust	Cluster analysis	Non-automatic	Trace within criteria (TWR)
				Variance ratio criteria (VRC)
				Run length distribution (RLD)
Kaushik and Arora [120]	FGA	Cluster analysis	Non-automatic	–
Nayak et al. [162]	FAFCM and improved FAFCM	Cluster analysis	Non-automatic	–
Sundararajan and Karthikeyan [213]	Hybrid modified firefly and dynamic k-means algorithm	Cluster analysis	Non-automatic	Inter-cluster distance
				Intra-cluster distance
Karaboga and Ozturk [117]	ABC	Cluster analysis	Non-automatic	CEP
Marinakis et al. [147]	DABC-GRASP	Cluster analysis	Non-automatic	Sigmod function
Tran et al. [220]	EABCK	Cluster analysis	Non-automatic	–
Ozturk et al. [168]	IDisABC	Cluster analysis	Automatic	VI index
				Correct classification percentage (CCP)

Table 7 (continued)

References	Clustering method	Application area	Clustering type	Cluster validity index (CVI)
Kuo et al. [129]	AKC-BCO	Cluster analysis	Automatic	CS_{kernel}
		Medicine (prostate cancer)		VI index
Kuo and Zulvia [128]	iABC	Cluster analysis	Automatic	VI index
		Customer segmentation		
Zhao and Zhou [242]	IWO-KPFCM	Cluster analysis	Non-automatic	–
Liu et al. [139]	MOIWO	Cluster analysis	Non-automatic	SI
				J_m index
				XB index
				ARI
Chowdhury et al. [41]	IWO	Cluster analysis	Automatic	Sym-K index
Satapathy and Naik [197]	TLBO	Cluster analysis	Non-automatic	–
Sahoo and Kumar [193]	TLBO	Cluster analysis	Non-automatic	–
Murthy et al. [159]	AUTO-TLBO	Cluster analysis	Automatic	CS index
Niknam et al. [165]	ACO-SA	Cluster analysis	Non-automatic	–
Wang et al. [228]	BPFPA	Cluster analysis	Non-automatic	–
Agarwal and Mehta [12]	MFPA-C	Cluster analysis	Non-automatic	Fitness function evaluation
Das et al. [52]	ACBEA	Cluster analysis	Automatic	CS index
Peng et al. [173]	Membrane system	Cluster analysis	Automatic	CS index
		Membrane computing		
Kumar et al. [125]	ACPAHS	Cluster analysis	Automatic	Inter–intra-cluster ratio
		Image segmentation		Fitness function evaluation
Liu et al. [140]	DLSIAC	Cluster analysis	Automatic	PMB index
		Image segmentation		
Kumar et al. [124]	ACGSA	Cluster analysis	Automatic	Inter–intra-cluster ratio
		Image segmentation		Fitness function evaluation
Kapoor et al. [114]	GWO	Cluster analysis	Automatic	Inter–intra-cluster ratio
		Image segmentation		DB index
Anari et al. [24]	ACCALA	Cluster analysis	Automatic	S_Dbw index
		Image segmentation		
Zhou et al. [245]	SOS	Cluster analysis	Automatic	Fitness function evaluation
Pacheco et al. [169]	Anthill	Cluster analysis	Automatic	SI
Elaziz et al. [64]	ASOCSA	Cluster analysis	Automatic	Dunn index
				SI
				DB index
				Calinski–Harabasz (CH) index
Agusti et al. [16]	GGA	Cluster analysis	Automatic	DBI index
Salcedo-Sanz et al. [195]	GGA-based model	Cluster analysis	Automatic	DBI index
Raposo et al. [184]	ACGA	Cluster analysis	Automatic	CH index

Table 7 (continued)

References	Clustering method	Application area	Clustering type	Cluster validity index (CVI)
Sharma and Chhabra [202]	HPSOM AHPSOM	Cluster analysis Mobile network data	Automatic	Inter-cluster distance Intra-cluster distance Adjusted Rand index (ARI) F-measure
Aliniya and Mirroshandel [21]	AC-ICA	Cluster analysis Face recognition	Automatic	Purity entropy Rand index (PERI) ARI *F*-measure
Rana et al. [179]	Hybrid sequential	Cluster analysis	Non-automatic	Intra-cluster distance Inter-cluster distance Quantization error
Boryczka [35]	ACA	Cluster analysis	Non-automatic	Cosine measure Gower measure Euclidean distance
Bouyer et al. [36]	HCSDE	Cluster analysis	Non-automatic	Inter-cluster distance Intra-cluster distance Error rate
Senthilnath et al. [200]	PSO, GA, CS	Cluster analysis	Non-automatic	CEP Statistical significance test
Goel et al. [84]	CSCA	Cluster analysis Satellite image	Non-automatic	DB index
Jensi and Jiji [110]	MBA-LF	Cluster analysis	Non-automatic	Euclidean distance
Cruz et al. [47]	cOptBees	Cluster analysis	Non-automatic	Entropy CEP Purity SI
Kumar and Sahoo [126]	GSA	Cluster analysis	Non-automatic	–
Abdulwahab et al. [2]	LBH	Cluster analysis	Non-automatic	Euclidean distance
Hatamlou [96]	BH	Cluster analysis	Non-automatic	Error rate Inter-cluster distance Intra-cluster distance
Younsi and Wang [235]	AIS	Cluster analysis	Non-automatic	Euclidean distance
Ezugwu [75]	FA, DE, PSO, IWO	Cluster analysis	Automatic	DB and CS index
Rajah and Ezugwu [176]	SOS, SOSFA, SOSDE, SOSTLBO, SOSPSO	Cluster analysis	Automatic	DB and CS index
Agbaje et al. [14]	FAPSO	Cluster analysis	Automatic	DB and CS index

5.1.3 Condorcet's criterion

Condorcet [44] proposed an evaluation criterion, which is given as follows:

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} s(x_j, x_k) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} d(x_j, x_k) \tag{9}$$

where $s(x_j, x_k)$ and $d(x_j, x_k)$, respectively, are similarity and distance between the vectors x_j and x_k.

5.1.4 C-criterion

An extension of Condorcet's validity index is given in [80]. The C-criterion is defined as follows:

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} \left(s\left(x_j, x_k\right) - \gamma\right) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} \left(\gamma - s\left(x_j, x_k\right)\right) \tag{10}$$

where γ is a threshold value.

5.1.5 Category utility metric

The category utility metric measures the goodness of fitness of the category [45, 83]. It is the evaluation criterion used in the popular conceptual clustering algorithm called COBWEB [78]. Given a set of entities, the binary feature set of size n is defined as

$$F = \{f_i\}, i = 1, 2, \ldots, n$$

and the binary category $C = \{c, \bar{c}\}$ is defined as follows:

$$\mathrm{CU}(C, F) = \left[p(c) \sum_{i=1}^{n} p(f_i|c) \log p(f_i|c) + p(\bar{c}) \sum_{i=1}^{n} p(f_i|\bar{c}) \log p(f_i|\bar{c})\right] - \sum_{i=1}^{n} p(f_i) \log p(f_i), \tag{11}$$

given that:

$p(c)$ is the prior probability of an entity belonging to the positive category c
$p(f_i|c)$ is the conditional probability of the feature f_i, given that it belongs to the positive category c
$p(f_i|\bar{c})$ is the conditional probability of the feature f_i, given that it belongs to the positive category $\bar{c}$
and $p(f_i)$ is the previous probability of the entity.

5.1.6 Bayesian information criterion (BIC) index

The BIC aims to solve the problem of overfitting of the partitions generated by the algorithm [174] and is defined as follows:

$$\mathrm{BIC} = -\ln(L) + v \ln(n) \tag{12}$$

where n is the number of entities, L is the likelihood of the parameters to generate the data in the model, and v is the number of free parameters in the Gaussian model. Minimizing the BIC is the goal.

5.1.7 Calinski–Harabasz index

The Calinski–Harabasz validity index measures compactness by calculating the distances between the points in a cluster to their centroids, and the separation is calculated by measuring the distance from the centroids to the global centroid [37]. This index is defined as

$$\mathrm{CH} = \frac{\mathrm{trace(SB)}}{\mathrm{trace(Sw)}} \cdot \frac{n_p - 1}{n_p - k} \tag{13}$$

where (SB) is the inter-cluster scatter matrix, (Sw) the intra-cluster scatter matrix, n_p is the number of entities in a cluster, and k the number of clusters.

5.1.8 Davies–Bouldin index (DB)

The DB index measures the average inter-cluster similarity between two clusters and the one that is closest to it [54]. It requires that information of at least two clusters be known. The Davies–Bouldin index is defined as follows:

$$\mathrm{BD} = \frac{1}{c} \sum_{i=1}^{c} \underset{i \neq j}{\mathrm{Max}} \left\{ \frac{d(x_i) + d\left(x_j\right)}{d\left(c_i, c_j\right)} \right\} \tag{14}$$

where c is the number of clusters, i, j are cluster labels, $d(x_i)$ and $d\left(x_j\right)$ are all entities in clusters i and j, $d\left(c_i, c_j\right)$ is the distance between the cluster centroids. Minimizing DB results in a "better" clustering solution.

5.1.9 Silhouette index

The silhouette index measures the compactness and separation of clusters [189]. It requires that information of at least two clusters be known.

Given a cluster, $X_j (j = 1, \ldots c)$, the index assigns to the ith entity of X_j the silhouette width, $s(i) = (i = 1, \ldots m)$. This value gives a degree of likelihood of the ith sample belonging in the cluster X_j. The index is defined as:

$$s(i) = \frac{(b(i) - a(i))}{Max\{a(i), b(i)\}} \tag{15}$$

where $a(i)$ is the average distance between the ith entity in the cluster and the remaining entities of cluster X_j; $b(i)$ is the minimum average distance between the ith s and all of the entities clustered in X_k (k = 1,… c; $k \neq j$).

5.1.10 Dunn index

The focus of the Dunn index is to estimate the ratio between the smallest inter-cluster distance and the largest intra-cluster distance in a partitioning [60]. It requires that information of at least two clusters be known. Several

variants of this index exist in the literature [31, 170]. The Dunn index is defined as follows:

$$\text{Dunn} = \min_{1 \le i \le c} \left\{ \min \left\{ \frac{d(c_i, c_j)}{\max_{1 \le k \le c} d(X_k)} \right\} \right\} \tag{16}$$

where $d(c_i, c_j)$ is the distance between cluster X_i and X_j; $d(X_k)$ represents the distance between members of cluster (X_k) and c is the number of clusters in the dataset. Maximizing the Dunn index gives a good clustering solution. Some setbacks of the Dunn include its time complexity, and it is affected by noise in datasets.

5.1.11 NIVA index

The NIVA validation index [186] is defined as follows:

$$\text{NIVA}(C) = \frac{\text{Compac}(C)}{\text{SepxG}(C)} \tag{17}$$

where Compac (C) is the average compactness of the cluster C and SepxG(C) is the average separability of cluster C.

5.1.12 Gamma index

The gamma index [27] is defined as:

$$G(C) = \frac{\sum_{c_k \in C} \sum_{x_i, x_j \in c_k} dl(x_i, x_j)}{n_w \left(\binom{N}{2} - n_w \right)} \tag{18}$$

where $dl(x_i, x_j)$ is the number of all object pairs in X.

5.1.13 Score function

This index measures the cluster separation by estimating the distance between cluster centroids to the global centroid. The compactness of the clusters is measured by estimating the distance from the points in a cluster to their centroid [194]. The index is defined as follows:

$$\text{SF}(C) = 1 - \frac{1}{e^{e^{\text{bcd}(C)} + \text{wcd}(C)}} \tag{19}$$

where

$$\text{bcd}(C) = \frac{\sum_{c_k \in C} |c_k| d_e(\overline{c_k, X})}{N \times K}$$

and

$$\text{wcd}(C) = \sum_{c_k \in C} \frac{1}{|c_k|} \sum_{x_i \in c_k} d_e(x_i, \overline{c_k}).$$

5.1.14 C-index

The C-index can be used for varying input data; it is easy to compute. It is easily generalized to estimate the cohesion of clusters [49]. The C-index is defined as:

$$\text{CI}(C) = \frac{S(C) - S_{\min}(C)}{S_{\max}(C) - S_{\min}(C)} \tag{20}$$

where

$$S(C) = \sum_{C_k \in C} \sum_{x_i, x_j \in C_k} d_e(x_i x_j)$$
$$S_{\min}(C) = \sum \min(n_w)_{x_i, x_j \in X} \{ d_e(x_i x_j) \}$$
$$S_{\max}(C) = \sum \max(n_w)_{x_i, x_j \in X} \{ d_e(x_i x_j) \}.$$

5.1.15 Sym-index

The theoretical base for the sym-index is the I-index, which is a point symmetry distance measure [29]. It is defined as follows:

$$\text{Sym}(C) = \frac{\max_{C_k, C_l \in} \left\{ d_e(\overline{c_k, c_l}) \right\}}{K \sum_{c_k \in} \sum_{x_i \in c_k} d^*_{ps}(x_i, c_k)} \tag{21}$$

5.1.16 COP index

The compactness or COP index is measured as the distance between the cluster points and their centroids, whereas the separation is a measure of the largest distance between neighbors [25]. It is defined as follows:

$$\text{COP}(C) = \frac{1}{N} \sum_{c_k \in} |c_k| \frac{\frac{1}{|c_k| \sum_{x_i \in c_k} d_e(x_i \overline{c_k})}}{\min_{x_i \notin c_k} \max_{x_i \in c_k} d_e(x_i x_j)}. \tag{22}$$

5.1.17 Negentropy increment

The negentropy index measures the normality rather than the compactness or separation of the clusters [133].

$$\text{NI}(C) = \frac{1}{2} \sum_{c_k \in C} p(c_k) \log \left| \sum_{c_k} \right| - 1/2 \log \left| \sum_{X} \right| - \sum_{c_k \in C} p(c_k) \log p(c_k) \tag{23}$$

5.1.18 SV-index

The separation is a measure of the nearest neighbor, and the compactness is a measure of the border points to the centroids of the cluster [239], as follows:

$$\mathrm{SV}(C) = \frac{\sum_{c_k \in C} \min_{c_l \in C \setminus c_k} \left\{ d_e(\overline{c_k, c_l}) \right\}}{\sum_{c_k \in C} 10/|c_k| \sum \max_{x_i \in c_k} (0.1|c_k|) \{ d_e(x_i \overline{c_k}) \}} \tag{24}$$

5.1.19 OS-index

The OS-index [57] is defined as follows:

$$\mathrm{OS}(C) = \frac{\sum_{c_k \in C} \sum_{x_i \in c_k} \mathrm{OV}(x_i, c_k)}{\sum_{c_k \in C} 10/|c_k| \sum \max_{x_i \in c_k} (0.1|c_k|) \{ d_e(x_i \overline{c_k}) \}} \tag{25}$$

5.1.20 The modified Hubert Γ statistic

This index [218] is defined as follows:

$$\Gamma = \left(\frac{1}{M} \right) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(i,j) \cdot Q(i,j) \tag{26}$$

where N is the dimension of the dataset, $M = N(N-1)/2$, P is the proximity matrix of the dataset and Q is an $N \times N$ matrix.

5.1.21 SD validity index

This index measures the mean intra- and inter-cluster scattering [90]. The definition for this index is as follows:

$$\mathrm{SD}(n_c) = a \cdot \mathrm{Scat}(n_c) + \mathrm{Dis}(n_c) \tag{27}$$

where

$$\mathrm{Scat}(n_c) = \frac{1}{n_c} \sum_{i=1}^{n_c} \sigma(v_i)/\sigma(X)$$

$$\mathrm{Dis}(n_c) = \frac{D_{\max}}{D_{\min}} \sum_{k=1}^{n_c} \left(\sum_{z=1}^{n_c} v_k - v_z \right)^{-1}.$$

5.1.22 S_Dbw validity index

The S_Dbw validity index uses the underlying characteristics of the clusters to measure the validity of the results from the clustering algorithm [88]. It is defined as follows

$$\mathrm{S\_Dbw}(n_c) = \mathrm{Scat}(n_c) + \mathrm{Dens\_bw}(n_c) \tag{28}$$

where

$$\mathrm{Dens\_bw}(n_c) = \frac{1}{n_c \cdot (n_c - 1)} \sum_{i=1}^{n_n} \left(\sum_{\substack{j=1 \\ i \neq j}}^{n_c} \frac{\mathrm{density}(u_{ij})}{\max\{\mathrm{density}(v_i), \mathrm{density}(v_j)\}} \right)$$

where v_i, v_j are the centroids of cluster c_i c_j, and u_{ij} the middle point of the line segment.

5.1.23 Root-mean-square standard deviation (RMSSTD)

The RMSSTD of the new cluster is the square root of the variance of all the attributes used in the clustering process [54]. It is defined as:

$$\mathrm{RMSSTD} = \left[\frac{\sum i = 1 \ldots nc \ \sum_{k=1}^{n_{ij}} (x_k - \overline{x_k})^2}{\sum i = 1 \ldots nc \ (n_{ij} - 1)} \right]_{j=1 \ldots v} \tag{29}$$

5.1.24 R-squared (RS)

RS is a measure of the difference between clusters [203]. This index is defined as follows:

$$\mathrm{RS} = \frac{SS_b}{SS_t} = \frac{SS_t - SS_w}{SS_t}. \tag{30}$$

5.1.25 Compact-separated (CS) index

The CS index is a validity measure [75, 121] that estimates the ratio of the sum of within-cluster scatter to between-cluster separation. A large value of a CS index indicates low compactness or separation, while a lesser value means a better clustering. It has been shown that the CS index offers more efficiency in handling clusters having different dimensions, densities, or sizes. Although it is computationally more expensive than the DB index in terms of execution time, it does, however, produces better quality solutions than does the DB index. Let the within-cluster scatter be denoted as X_i and the between-cluster separation be represented as X_j, such that the distance measure V is given as $V(X_i, X_j)$. Hence, the CS index for a clustering Q is computed as follows [75].

$$\mathrm{CS}(Q,V) = \frac{\frac{1}{P}\sum_{i=1}^{P}\left[\frac{1}{D_n}\sum_{X_i\in Q_i}\max_{X_j\in Q_i}\{V(X_i,X_j)\}\right]}{\frac{1}{P}\sum_{i=1}^{P}\left[\min_{j\in P, j\neq i}\{V(x_i,x_j)\}\right]} = \frac{\sum_{i=1}^{P}\left[\frac{1}{Q_i}\sum_{X_i\in Q}\max_{X_j\in Q}\{V(X_i,X_j)\}\right]}{\sum_{i=1}^{P}\left[\min_{j\in P, j\neq i}\{V(x_i,x_j)\}\right]} \tag{31}$$

where $|D_n|$ represents the number of data points in cluster P, the function $V(X_i, X_j)$ is the distance between within-cluster scatter X_i and between-cluster separation X_j, $V(x_i, x_j)$ is the distance of data points d from their centroids, and P is the number of clusters in Q.

5.2 External quality criteria measures

In this approach, knowledge about the dataset is required (i.e., underlying structure or number of clusters), although in the real world this information is not always available. The basic idea is to match the result of the partition with the predefined structure of the dataset.

5.2.1 Mutual information-based measure

This index is based on the mutual interdependency between the partition result and the underlying structure of the dataset [212].

For m instances in clusters $C = \{C_1, C_2, \ldots, C_g\}$ and target attribute z with domain $\mathrm{dom}(z) = \{c_1, c_2, \ldots, c_k\}$, the index is defined as follows:

$$C = \frac{2}{m}\sum_{i=1}^{g}\sum_{h=1}^{k} m_{i,h}\log_{g\cdot k}\left(\frac{m_{i,h}\cdot m}{m_{.,h}\cdot m_{i,.}}\right) \tag{32}$$

where $m_{i,h}$ is the number of instances in cluster C_i and in class c_h, $m_{.,h}$ denotes the total number of instances in the class c_h, and $m_{l,.}$ denotes the number of instances in cluster C_i.

5.2.2 Rand index

This index shows the similarities between partitions of the clustering algorithm and the underlying structure of the dataset [181]. The index is defined as follows:

$$\mathrm{RAND} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}} \tag{33}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

The index usually is in the range 0 to 1, with a Rand index of 1 indicating a perfect match.

5.2.3 *F*-measure

Equal weighing for the false positives and false negatives usually results in undesirable features. So the F-measure index addresses this by using weighting recall parameter $\eta > 0$ to balance the false negatives [187]. The F-measure is defined as follows:

$$F = \frac{(\eta^2+1)\cdot P\cdot R}{\eta^2\cdot P + R} \tag{34}$$

where P is the precision rate, and R is the recall rate.

The range of recall starts from no effect ($\eta = 0$) to more effect as η increases (indicating a higher clustering quality).

5.2.4 Jaccard index

This index shows the similarities between the two datasets. It is the ratio between the number of entities that belong to both datasets and the number of entities in both datasets [104]. The Jaccard index is defined as follows:

$$J(AB) = \frac{A\cap B}{A\cup B} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{35}$$

if A and B are empty, then $0 \le J(AB) \le 1$.

5.2.5 Fowlkes–Mallows index

This index measures the compactness of clusters obtained from a clustering algorithm. Maximizing the index results in higher similarities [81]. It is defined as

$$\mathrm{FM} = \sqrt{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}\cdot\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}} \tag{36}$$

5.2.6 NMI measure

The normalized mutual information (NMI) is defined as follows:

$$\mathrm{NMI}(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} \tag{37}$$

where $I(X, Y)$ is the mutual information between two random variables X and Y and $H(X)$ denotes the entropy of X, X is the partition by a clustering algorithm, and Y represents the true labels of the dataset [137].

5.2.7 Purity

Purity applies to a set of clusters. The purity for each cluster P_j is defined as:

$$P_j = \frac{1}{n_j}\mathrm{Max}_i\left(n_j^i\right) \tag{38}$$

The purity for the set of clusters is calculated as a weighted sum of the individual purities [123]. This is given as:

$$\text{Purity} = \sum_{j=1}^{m}\frac{n_j}{n}P_j \tag{39}$$

where n_j denote the size of cluster j, m is the number of clusters and n is the total number of entities.

5.2.8 Entropy

Entropy increases as the classification of objects in a cluster become more varied. If all the objects in the cluster belong to one label, then the entropy is 0 [57]. In this case, entropy is defined as follows:

$$E_j = \sum_i p_{ij}\log\left(p_{ij}\right) \tag{40}$$

5.2.9 Relative validation

This scheme tries to validate the partitions of a clustering algorithm by estimating the best clustering scheme possible under certain assumptions and parameters. It tunes the parameters and evaluates or compares the resulting cluster structures produced by the algorithm. The relative validation involves a lot of statistical testing [89]. Assuming the clustering problem is defined as thus:

> Let P_{alg} be the set of parameters associated with a specific clustering algorithm (e.g., the number of clusters n_c). Among the clustering schemes C_i, $i = 1,\ldots,n_c$, defined by a specific algorithm, for different values of the parameters in P_{alg}, choose the one that best fits the data set. [89].

The following cases hold:

- P_{alg} does not contain n_c as a parameter.
  The idea here is to tune the parameter over a wide range of values and run the clustering algorithm and then choose the maximum range for which n_c is constant. Normally, $n_c \ll N$ where N is the number of tuples
- P_{alg} contains n_c as a parameter.
  Define a maximum and minimum range first, then run the algorithm r times over each n_c between the minimum and maximum range, tuning the parameters during each run. Then plot the best values of the index obtained against n_c. The plot may indicate the best cluster.

6 Discussion

Clustering is aimed at ensuring that a set of objects is efficiently classified into their various clusters. The approach taken to achieve the clustering of such objects relies largely on the algorithms designed for the task. Hence, algorithms for clustering need to be very efficient in their approach to classifying or categorizing objects into their most fitting class or category. Moreover, the challenge of managing extensive data, which are frequently generated from social media and other online network-based platforms with high throughput streams, requires that clustering algorithms evolve to allow for efficient deployment. In addition to the challenge of data size, memory usage and management are also a design issue with clustering algorithms, meaning that algorithms that are designed to function well in an environment of limited memory are considered efficient. Clustering algorithms that are aimed at streaming data would demonstrate an interesting performance in this scenario.

The application of clustering techniques and the corresponding array of algorithms, to different domains or fields such as engineering, medicine, and data mining, continue to affect the development and modification of clustering algorithms and their related techniques. The development of a clustering algorithm entails the following phases: feature selection, pattern proximity, cluster formation, and clustering validation [215]. The task of designing and developing clustering algorithms also factors in the automatic and non-automatic pattern of clustering objects the proposed clustering algorithm adopts. Design of automatic clustering algorithms is non-trivial because algorithm design must accommodating the number of clusters being unknown a priori [196], contrasting with non-automatic clustering algorithms which require that such a parameter be known. This, therefore, is one important concept that has shaped the trends in the emergence of new clustering algorithms, which we shall discuss in the following paragraphs.

Clustering algorithms have evolved and are evolving due to the nature of data being classified, the approach for similarity measures (like Euclidean distance), the evaluation criteria, the validity and accuracy of the clusters generated, and the high dimensionality of data, which often leads to increased computational cost. Although evaluation of clusters derived from a clustering algorithm may follow the method of homogeneity, completeness, and V-measure, the models used for the evaluation and performance measures are also worth noting. These include Condorcet's criterion, edge-cut metrics, the sum of squared error, category utility metric, scatter criteria, and C-criterion in the category of internal performance measures, while the likes

of Fowlkes–Mallows index, Rand index, mutual information-based measure, confusion matrix, *F*-measure, and Jaccard index may form the external performance measures. These performance measures may not directly affect the design of the clustering algorithm. They do, however, weigh in on its performance, which invariably influences the trends in design, evolution, and application of such clustering algorithms. Accordingly, we shall focus our discussion on the evolution of popular traditional or classical clustering algorithms, namely BIRCH, DBSCAN, k-Means, Mini-Batch k-Means, Mean Shift, OPTICS, Spectral Clustering, and Mixture of Gaussians. Also, we will discuss trends for the task of data clustering using algorithms that adopt or repurpose approaches based on nature-inspired (NI) algorithms. Meanwhile, in our discussion of the trends in clustering, we will touch on issues like how clustering algorithms are able to handle high dimensionality of data while also managing computational cost, and their effect on the consistency of algorithms, and other relevant issues.

6.1 Recent trends in clustering algorithms

In Sects. 1 and 2, it was noted that clustering techniques might be categorized as either hierarchical or partitional, with some literature including the classifications of grid based, density based or model based. Therefore, the following subsections present trends in clustering algorithms based on their methods, namely hierarchical based, partitional based, grid based, density based and model based.

6.1.1 Hierarchical-based algorithms

Clustering algorithms based on a hierarchical approach exploit the hierarchical structure inherent in the data. An interesting issue about hierarchical clustering algorithms is its non-sensitivity to the chosen distance metric and the automatic nature of the discovery of the number of clusters existing in a dataset. These characteristics, especially the selection of distance metric, are unusual among other categories of clustering algorithms. Some popular clustering algorithms that fit well into this category are the balanced iterative reducing and clustering using hierarchies (BIRCH), clustering using representatives (CURE), as well as ROCK and CHAMELEON.

As clustering algorithms have evolved, the need for robustness in handling outliers and continuous increments in the size of data (high dimensionality of data) when clustering has motivated the use of BIRCH. However, an attempt to further improve the performance of BIRCH in this direction has resulted in the variants named Bubble and Bubble-FM clustering algorithms. In addition to this capability of BIRCH, it is known to achieve and maintain a computational complexity of $O(N)$.

Similarly, CURE, a multi-centroid clustering algorithm, has been shown to be tolerant of outliers and able to handle large-scale databases well. However, CURE archives a computational complexity of $O(N2\log N)$ compared to the low value associated with BIRCH and has an ineffective approach for handling noise. Nevertheless, it performs well in high-dimensional datasets having varying densities of data points, and also it shows the capacity to locate non-spherically shaped and wide variance-sized datasets [86], even though it does not use a distance function. The ability of CURE to support clustering of non-spherically shaped datasets lies in its mechanism for representing clusters using well-scattered points per cluster, thereby yielding more than one point per cluster, which leads to its geometrical flexibility and its ability to shrink to detect outliers.

Although BIRCH and CURE are not purely hierarchical clustering algorithms, they do represent an improved version of hierarchical clustering algorithms; as such, they are two integrated hierarchical clustering algorithms. Sometimes, the use of BIRCH is complemented by other clustering algorithms, which are applied to the result of summaries generated by BIRCH. BIRCH's summarizing nature allows it to minimize memory usage during clustering operation. While CURE performs well on non-spherical data, BIRCH suffers some limitations which are overcome by the improved versions of it: for instance, Link BIRCH (LBIRSCH) leverages on the concept of link, as used in ROCK [87].

The robust clustering algorithm (ROCK) uses links instead of a distance function for the purpose of clustering. It is a clustering algorithm used on categorical datasets and has demonstrated an interesting performance in cluster forming, cluster merging, and other cluster-based operations. It has a computational complexity of $O(N2 + Nm + N2\log N)$, meaning that it also scales poorly in this area. QROCK is a quick version of the ROCK algorithm for clustering of categorical data.

In contrast, the CHAMELEON clustering algorithm is known for handling low-dimensional spaces and allows for merging of clusters using proximity between two clusters. The algorithm's capability of operating on the sparse graph, where nodes denote data items and where edges with weights represent similarities between data objects, makes it perform well as a clustering algorithm. CHAMELEON, despite having a computational complexity of $O(Nm + NlogN + m2logN)$, has been proven to outperform DBSCAN and CURE. For better time complexity, BIRCH may outperform CURE even though CURE effectively handles larger datasets and with a better quality of clustering [199].

6.1.2 Partitional-based algorithm

The partional-based category of clustering algorithms applies the Euclidean distance as the most commonly used criterion data object. Examples of clustering algorithms found in this category include k-means, expectation–maximization (EM), fuzzy c-means, PAM, CLARA, and CLARANS.

One of the best-known and probably most used classical clustering algorithms is the k-means algorithm [199] with a computational complexity of $O(N)$, meaning that it is computationally efficient. Its simplicity has probably won it the attention it has garnered, sometimes making it a benchmark for clustering algorithms. The popular k letter used in its name defines k centroids, one for each cluster such that each point in the dataset is assigned to the closest centroid. However, this makes it sensitive to the initial randomly set k value, thereby limiting its performance on data objects which are only clustered on the spherical shape or are inconsistent [219]. Its method of handling features is similar to the approach that the BIRCH clustering algorithm adopts in dealing with metric attributes.

The fuzzy c-means (FCM) algorithm changes discrete values of the belonging label, $\{0, 1\}$ into the continuous interval $[0, 1]$. As with the k-means clustering algorithm, FCM has a computational complexity of $O(N)$, except for some of its approaches such as the use of fuzzy logic in determining clusters. However, FCM has the drawback of susceptibility to local optima, dependence on its initial partition and sensitivity to noise and outliers. Because FCM often arrives at inexact clustering results, an improved version of the algorithm was proposed, namely the weighted fuzzy c-means (WFCM), which uses a two-stage feature of selection and weighting. Efforts to address the memory and speed issues associated with k-means algorithm resulted in what is known as MapReduce-based k-means (PK-means) demonstrating the distributive nature of its parent algorithm. The PK-means clustering algorithm allows computation tasks to be distributed among participating machines, which thereby improves the performance of k-means because it allows for easy scaling up while speed and dataset sizes are also increased.

To improve on the sensitivity of k-means to outliers, another clustering algorithm, k-medians, was proposed. It uses the median vector of the group to compute the center, although it is slower in cases with larger datasets arising from computing the median vector. Another centroid-based clustering algorithm is the mean shift clustering, which works by updating candidates for center points to be the mean of the points. This clustering algorithm uses a sliding-window-based algorithm to find dense areas among data points. The algorithm outdoes the k-means, based on its mechanism of using mean shift to discover the number of clusters and also the ability to use its windows to eliminate near-duplicates. While k-means and its variants leverage computing means and medians, the expectation–maximization (EM) clustering algorithm uses Gaussian mixture models (GMM) so that data points are Gaussian-distributed. By using this approach of using a mean function, EM clustering algorithm circumvents the assumption that its clusters are circular.

Partitioning around medoids (PAM) is another memory demanding clustering algorithm, which often stores its result of pairwise dissimilarity matrix computation in memory, thereby limiting its application to large datasets. The computational complexity of the algorithm is $O\left(K(N-K)^2\right)$, where K is the number of clusters, and N represents the number of points in the data. To overcome this memory demand of PAM, another clustering algorithm, clustering large applications (CLARA) was proposed. CLARA minimizes the average dissimilarity between objects and objects closest to them. It does this by reducing the search space by searching only a sub-graph prepared by a sampled $O(K)$ data points, although it has the ability to draw multiple samples. This leads to the computational complexity of $O\left(K(40+K)^2+K(N-K)\right)$ while providing users with a clustering algorithm capable of handling large datasets, thereby earning it the rank of best partitioning clustering algorithm considering the output. But to further improve the efficiency of CLARA, a variant clustering algorithm named CLARANS was proposed, which has a computational complexity of $O(KN^2)$. It operates by searching the entire graph while seeking to obtain an optimal local solution. Unlike CLARA, this algorithm achieves its sampling through a dynamic approach that uses iterative operations for the search procedure. This dynamic sampling leads to the efficient performance of CLARANS and also influences the clusters derived from its operation. Nevertheless, a study has revealed that BIRCH outperforms CLARANS [163]. Generally speaking, some of the clustering algorithms in this category are known to present the drawback of being unable to adjust themselves when a merge or split decision has been executed [182].

6.1.3 Grid-based clustering algorithms

The approach of clustering adopted in grid-based algorithms is similar to geometric settings of grid structure; it uses a multi-resolution grid data structure. It does this by quantizing the clustering space into a given number of cells before performing the required operations on the quantized space. Clustering algorithms like STING, Wave-Cluster

and CLIQUE constitute members of this category of clustering technique.

The STING clustering algorithm has been proven relevant in parallel processing because it operates by breaking down available space of objects into cells of rectangular shapes and a hierarchical format. The resulting data in the hierarchical structure is considered as its clusters [232]. The clustering algorithm does this successfully in such a way as to remove the resource burden engaged during clustering and query-based problems. STING is often rated to outperform DBSCAN, BIRCH, and even CLARANS, although the algorithm suffers from slower execution in comparison with those it outperforms, like DBSCAN. Another grid-based clustering algorithm, which is efficient in terms of computational time complexity trade-off, is the so-called Wave-Cluster algorithm, in which detection of arbitrarily shaped clusters is based on wavelet transformations. Its outstanding performance is widely reported to be 30 times better than for CLARANS and 10 times more efficient than the hierarchical-based BIRCH [23]. In another effort to leverage the DNF expression-like approach, another grid-based clustering algorithm, Clustering in QUEst (CLIQUE) follows that approach to generate its clusters. This makes it insensitive to the sequence in which inputs are entered into the algorithm as it searches clusters by exploiting density-based clusters in subspaces. Its support for the detection of clusters in subspaces of high dimensionality makes it different to and better than other clustering algorithms. The assumption is that such highest dimensionality possesses high-density clusters in subspaces.

6.1.4 Density-based clustering algorithms

The approach of grid-based clustering positions for some limitations, as outlined in Table 8, has resulted in a density clustering technique designed to overcome those limitations. In this section, we shall explore trends in some density-based clustering algorithms like DBSCAN, OPTICS, DBCLASD, and DENCLUE.

The first, and presumably the most popular, clustering algorithm in this category is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN is built so that, in any direction, cluster points are closely packed together and as a result yield clusters of different shapes. However, DBSCAN is limited by its nature in that it does not capture various kinds of noise points in clusters of different densities. Meanwhile, a parallel form of DBSCAN was proposed, namely PDBSCAN. Furthermore, to advance the concept of generalization in density-based clustering algorithms, another clustering algorithm (GDBSCAN) was proposed to enable support for cluster objects in their numerical and categorical attribute forms. Recently, another variant of DBSCAN algorithm is the MR-DBSCAN- or MapReduce-based DBSCAN. By exploiting the provision of Hadoop frameworks in MapReduce, the MR-DBSCAN adopts an approach that favors scalability and reduction in computational cost through the use of data partitioning method. Also, to take advantage of GPUs which have thousands of cores that propel their speed and computational power, G-DBSCAN was developed. This variant of DBSCAN algorithm enjoys the merits of a parallel computing environment. Other variants and improved versions of DBSCAN, proposed to overcome its limitations, are LD-BSCA [230], FDBSCAN, VDBSCAN, IDBSCAN, Revised DBSCAN (RDBSCAN capable of easily identifying the borders of objects lying close to adjacent borders) and shared nearest-neighbor algorithm (SNN, which leverages some concepts in ROCK clustering algorithm to produce a density-based clustering algorithm). In related work, a study reported how experimentation involving k-means, k-medoids, fuzzy c-means, DBSCAN, OPTICS, and hierarchical clustering algorithms combined DBSCAN with other algorithms to provide a simple Amplitude Modulation (AMC) algorithm [156]. Advances related to DBSCAN were also reported by Vo-Van et al. [227], who adopted the epsilon radius neighbors used in DBSCAN to identify the number and shape of clusters automatically. Table 9 presents a category based comparison of the performance of clustering algorithms.

DBSCAN is associated with the problems of being unable to detect interesting clusters from datasets presenting varying densities and sensitivity to the radius of the neighborhood and the minimum number of points in a neighborhood. To overcome these, a density-based clustering algorithm, which was aimed at tackling this limitation of DBSCAN, is a connectivity-based algorithm named OPTICS. OPTICS yields more efficiency than does DBSCAN at a computational complexity of $O(n \log n)$, although it can only generate clusters with local-density non-similar clusters. Meanwhile, another density-based clustering proposed is the Distribution-based Clustering of Large Spatial Databases (DBCLASD), which performs well by building clusters from large spatial databases. DBCLASD essentially employs an incremental approach to place points in a cluster.

In a related work to advance the performance and speed of DBSCAN, and even CLARANS clustering algorithms, DENsity-based CLUstEring (DENCLUE) was able to speed up DBSCAN while forming center defined and multi-center defined type of clusters. DENCLUE, based on kernel density estimation, uses strong mathematical models making it capable of working with datasets having noise and modeling arbitrarily shaped clusters in high-dimensional datasets. DENCLUE has also demonstrated good clustering properties and results in datasets with large

Table 8 Performance comparison of clustering algorithms from classical-based to nature-inspired-based algorithms

Clustering algorithm	Description	Complexity of algorithm	Shape of cluster	Scalability	Type of dataset	Suitability for large/ dimensionality data	Advantage	Disadvantage
BIRCH	BIRCH uses a hierarchical data structure called CF-tree for partitioning the incoming data points in an incremental and dynamic way	$O(n)$	Non-convex	High	Numerical	Yes/no	Reasonably fast, it can be used as a more intelligent alternative to data sampling in order to improve the scalability of other clustering algorithms	It may not work well when clusters are not "spherical" because it uses the concept of radius or diameter to control the boundary of a cluster It is order-sensitive as it may generate different clusters for different orders of the same input data
CURE	A constant number of representative points are chosen to represent a cluster	$O(n^2 \log n)$	Varying shapes	High	Numerical	Yes/yes	Can recognize arbitrarily shaped clusters and is robust to the presence of outliers	Ignores the information about the aggregate inter-connectivity of objects in two clusters
ROCK	Using the Jaccard coefficient to measure similarity. It accepts as input the set S of n sampled points to be clustered (that are drawn randomly from the original data set), and the number of desired clusters k	$O(n^2 + nm^3a + n^2 \log n)$	Varying shapes	Middle	Categorical	No/yes	Robust and appropriate for large dataset	Space complexity depends on initialization of local heaps
CHAMELEON		$O(n^2)$	Varying shapes	High	Numerical, categorical, spatial, multivariate and others	No/no	Effective in datasets that contain points in 2D space, and clusters of different shapes, densities, sizes, noise, and artifacts	Low-dimensional spaces, and not applied to high dimensions Time complexity in high dimensions is high
PAM		$O\left(k(n-k)^2\right)$	Non-convex	Low	Numerical	No/no	It is robust to outliers	Number of clusters must be predetermined
CLARA		$O\left(k(40+k)^2 + k(n-k)\right)$	Non-convex	High	Numeric	Yes/no	Can handle large datasets	Not robust to outliers
CLARANS		$O(kn^2)$	Non-convex	Middle	Numeric	Yes/no	Robust to outliers	Has high computational cost
DBSCAN	Locates regions of high density from regions of low density	$O\ (n \log n)$	Varying shapes	Middle	Numeric	Yes/no	Number of clusters not initially known Identifies outliers as noises. Able to find arbitrarily sized and arbitrarily shaped clusters	Fails when clusters are of varying density

Table 8 (continued)

Clustering algorithm	Description	Complexity of algorithm	Shape of cluster	Scalability	Type of dataset	Suitability for large/ dimensionality data	Advantage	Disadvantage
Fuzzy c-means	Similar to k-means algorithm and also membership value	$O(n)$	Non-convex	Middle	Numerical	Yes/no		
K-means	Partitions the set of feature vectors into K disjoint	$O(n)$	Non-convex	Middle	Numerical	Yes/no	Effective in dealing with huge datasets and often terminates at a local optimum	It fails in cases where the clusters are not circular
STING	Forms a hierarchical structure from several levels of rectangular cells	$O(k)$	Varying shapes		Spatial	Yes/no	Allows parallelization and multiresolution	Resulting clusters are all bounded horizontally or vertically and not diagonally
WaveCluster		$O(n)$	Varying shapes		Spatial	Yes/no	No need for an a priori information on number of clusters	Not efficient in high-dimensional space
CLIQUE	Generates the set of two-dimensional cells that might possibly be dense from one-dimensional spaces	$O(Ck + mk)$	Varying shapes	High	Numerical	Yes/yes	Has good scalability as the number of attributes is increased	Prone to high-dimensional clusters
SOM net	Uses two layers of neural network and their neurons as cluster centers	$O(n^2m)$	Non-convex		Multivariate	No/yes	Easily detects outlier and can deal with missing data values	
DENCLUE	Based on the minimization of both the WGS and DB indices to explore and exploit the search space, respectively	$O(\log\|D\|)$	Varying shapes		Large no. of data	Yes/	Has a solid mathematical base and is capable of generalizing various clustering methods like partitioning-, hierarchical-, and density-based methods	The density parameter and the noise threshold need to be selected carefully as it significantly affects the quality of results
DBCLASD	Design good cluster for spatial database	$O(3n^2)$	Varying shapes		Spatial data with uniformly distributed points		Useful when time does not matter but clustering quality is desired	Slow computational procedure

Table 9 A category-based comparison of their performance of clustering algorithms

Category of clustering technique	Computational complexity	Advantages	Disadvantages	Input
Hierarchical	$O(n^2)$	Handle any forms of similarity or distance functions and are applicable to any attributes type	Already created clusters are not revisited for improvement	Radius of cluster, branching factor
Partitioning	$O(n)$	Simple to understand and implement. It takes less time to execute as compared to other techniques	Whenever a point is close to the center of another cluster; it gives poor outcome due to overlapping of data points	Number of clusters required
Grid based	$O(n)$	Possess a quick processing time and is independent of the number of data objects	Adversely affected by the number of cells in each dimension in the quantized space	Size of grid, number of objects in a cell
Density based	$O(n \log\ n)$	Efficiently handles large amount of noise in dataset and does not need a priori specification	Performs poorly when there is high dimensionality data	Threshold, radius

noise. In another variation, the SUBspace CLUstering (SUBCLU) clustering algorithm uses the approach of cluster identification by dense regions being separated from the sparse ones to build clusters using a bottom-up model. Finally, for this section of density-based clustering algorithms, the Fast Density-Based Clustering (FDC) algorithm, uses a density-linked relationship, defined by equivalence.

6.1.5 Model-based clustering algorithm

In this section, we discuss two model-based clustering algorithms, namely COBWEB and SOM. Model-based clustering algorithms are designed to use selected models for representation of clusters. All clustering algorithms in this category are usually categorized into statistical learning method (COBWEB) and neural network learning method (SOM and ART). SOM leverages the presumed existence of topology to translate mappings from a high dimension in the input space to a lower dimension in the output space. SOM uses the Euclidean distance function for its distance measure. On the one hand, the notable performance of the SOM clustering algorithm has been observed even when the number of clusters increases, although with slower performance. However, SOM is sensitive to noise in datasets. On the other hand, COBWEB uses some exploratory criteria to build clusters through classification trees, which translate into a hierarchical clustering.

6.1.6 Modern clustering algorithms

In the previous sections, we have largely centered our discussion on trends in traditional or classical clustering algorithms. The challenge with some such clustering algorithms is that they are easily entrapped within local optima and present difficulties in handling complex datasets. Therefore, we felt it necessary to also dwell more on modern clustering algorithms. This category of clustering algorithm consists of those based on an ensemble of models: quantum theory (e.g., quantum clustering QC and DQC), spectral graph theory (e.g., SM and NJW), affinity propagation (AP) and nature-inspired (NI) oriented clustering algorithms. We shall focus our discussion in this section on several clustering algorithms using nature-inspired metaheuristic algorithms to solve data clustering problems. For instance, metaheuristic algorithms consisting of biotic and abiotic forms, like cuckoo search (CS), firefly algorithm, BAT algorithm, genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), gravitational and Tabu search algorithms, have been well exploited for clustering tasks. We shall therefore divide our discussion of clustering algorithms in this context into three: evolutionary, biotic algorithms together with collective intelligence metaheuristic approaches, and abiotic algorithms

Clustering algorithms resulting from repurposing approaches, based on evolutionary algorithms like evolution strategies (ES) evolutionary programming (EP) genetic algorithm (GA) particle swarm optimization (PSO), differential evolution (DE), and ant colony optimization (ACO), have yielded powerful clustering algorithms. Examples of these clustering algorithms are the EP-based and GA-based clustering algorithms such as evolutionary programming clustering algorithm (EPC) and its improved version GEP-cluster; GA-based clustering algorithm has been leveraged in tackling problems of automatic

clustering, of which CLUSTERING, quantum-based QGA, two staged-based TGCA, multi-objective MOKGA, multi-objective soft subspace-based MOEASSC, VGA-clustering, k-means-based clustering KMQGA, fuzzy-based FVGA-clustering, symmetry-based VGAPS, fuzzy-based VGAPS named FVGAPS, and AGCUK (which does not require the cluster number be specified a priori) algorithms are good examples. The DE-based approach has also been optimized to produce clustering algorithms like the automatic-based ACDE, a corresponding fuzzy-based AFDE, which was further enhanced to use the kernel to produce KFNDE, and an automatic version of AFDE called MoDEAFC. Other examples include multi-objective versions of clustering algorithms like MODE, DEMO, and a fuzzy-based MOMoDEFC.

The biotic approaches which include swarm intelligence (SI), artificial immune systems (AIS), invasive weed optimization (IWO), and simulated annealing (SA) have also been harnessed successfully to develop new clustering algorithms. Some of these clustering algorithms include SI PSO-based DCPSO clustering algorithm, a segmentation-based MEPSO, point-symmetry-based PSOPS, a similitude of AGCUK named CPSO, DCPG resulting from hybridization of PSO and GA, ant-based called Ant-clustering, an improved version of ant-based called ATTA-C, automatic single-objective IWO-clustering, BCO-based automatic AKC-BCO which uses kernels, AIS-based cluster algorithm named GTCSA (which uses clonal selection algorithm), and DLSIAC. Other multi-objective clustering algorithms include IDCMC, SI-based MOPSO, MOIMPSO and MOPSOSA (resulting from hybridization efforts), MOCLONAL, SA-based AMOSA, MOIWO (which is similar to CPSO but does not need to know the number of clusters a priori), Bird Flock Gravitational Search Algorithm (BFGSA), and specifically three-objective-based VAMOSA and GenClustMOO. There are studies that have also attempted to hybridize traditional clustering algorithms with NI-based methods to yield clustering algorithms like k-Means-ALO (ant lion optimization), KMeans-PSO, and KMeans-FA.

Finally, clustering algorithms in the category of the abiotic approach have also been proposed and developed. For instance, a clustering algorithm named GRIN is based on gravity theory in physics and has proved to be effective due to its non-sensitivity to the distribution of the data set [122].

Considerable attention has recently been generated by automatic clustering algorithm based on nature-inspired metaheuristics [111] and their applications, due to the limitations of the single-objective metaheuristics-based clustering algorithm in automatic clustering algorithms, so multi-objective clusterings have generated many clustering approaches, and algorithms are continuously being optimized to solve even nonlinearly separable problems. This is necessary given that single-objective clustering algorithms are good at efficiently grouping linearly separable clusters but suffer from getting entrapped in local regions. Although NI-based approaches in single-objective automatic clustering using evolution strategy (ES), genetic algorithm (GA), evolutionary programming (EP), or differential evolution (DE) have been attained, multiobjective metaheuristics optimize more than one objective function simultaneously.

NI-based clustering algorithms make use of their foundational approaches to demonstrate the ability to learn or adapt to new situations while solving clustering problems even in complex and changing environments [145]. Studies have reported a state-of-the-art performance of such algorithms as seen in an improved firefly algorithm, which was hybridized with particle swarm optimization algorithm to solve automatic data clustering problems [14]. Similarly, an NI-based clustering that uses SOS algorithm was applied to solve clustering problems [244].

6.2 Open challenges and further research directions

In the previous subsection, we chronicled the advances and trends in the evolution of most of the clustering algorithms, while highlighting their strengths and weaknesses. This analysis has, accordingly, revealed the challenges and opportunities inherent in building new clustering algorithms; the gap which researchers can exploit. We assert that some open challenges associated with the clustering algorithms we have reviewed should be clearly outlined in this section so as to provide readers with direction from which concepts and ideas can be generated in building new clustering algorithms. Although other mainstream issues like the evaluation criteria, distance function or approach of measures of similarity may be considered as open opportunities for further research, sophisticated automatic clustering and problems associated with widening fields of applications of such clustering algorithms leave much room for future work. Meanwhile, the proliferation of clustering algorithms also provides users with an array of options from which to select the most appropriate algorithm for their domain. We observed that this is also an avenue where developers of clustering algorithms may attempt to focus algorithm design domain specification rather than generalizing their operations. We argue that such a focus might reveal some latent clustering properties shared by domains of application, thereby highlighting likely areas for fruitful hybridization of clustering concepts. For instance, many application domains have large amounts of high-dimensional datasets, for which most automated clustering algorithms suffer some limitations

when handling this problem and may result in unattractive clusters.

Furthermore, the peculiarity of the categories of clustering techniques and algorithms discussed in Sect. 6.1 presents developers of related algorithms with open challenges for revolutionizing approaches for designing clustering algorithms. For instance, one consideration is that hierarchical-based clustering algorithms are associated with a growing time complexity as the number of instances increases. Another aspect is that grid-based clustering algorithms are limited by their difficulty in discovering clusters of varied shapes or sizes. By contrast, density-based clustering algorithms successfully identify clusters of varied shapes and can effectively handle noise in datasets. Clustering algorithms based on the technique of partitioning require the a priori knowledge about the distribution of data leading to preknowledge of the number of clusters since such information is needed as input to those classes of clustering algorithms. Careful exploitation of the pros and cons of these clustering algorithms holds interesting avenues of exploration for algorithm designers.

Also, considering a widely adopted approach of developing clustering algorithms through efforts aimed at leveraging approaches found in nature-inspired algorithms, most of the problems associated with automatic clustering are being addressed. This research effort has produced interesting clustering algorithms that have proven to be effective and outperformed classical clustering algorithms [111]. However, even clustering algorithms patterned after this approach show evidence of the need for further enhancement. Again, this is another area open for further advancing the design of the clustering algorithm. Meanwhile, other challenges that might have been only partially addressed by some clustering algorithms and so provide opportunities for improvement are the difficulty in extending clusters resulting from low-dimensional cases to high-dimensional cases; the challenge of discovering important parameters for tuning clustering algorithms for effective application across domains; and the difficulty associated with verifying and interpreting clusters of high dimensionality.

Considering all the challenges above, the following list outlines some areas open for advancing design and development of clustering algorithms and their techniques. Although some of them might not appear to be completely new, they do, however, need further improvement.

1. Increase in sources and platforms for data generation continues to present data analysts with challenges associated with the extraction of knowledge from terabytes and petabytes of data. Therefore, clustering algorithms targeted at effectively clustering such data need improvement or a complete redesign of such.
2. The change in focus of design patterns in developing clustering algorithms in the category of non-automatic approach has revealed problems peculiar to the resulting automatic clustering algorithms. In addition to designing clustering algorithms that are robust enough to handle problems of automatic-based algorithms, there is also a need for a sophisticated automatic clustering technique that allows for flexibility and effectiveness in use.
3. The challenge of sophisticated or complex automatic clustering problems may arise from solutions in (2), probably due to the nature of input or the dataset. However, such clustering algorithms may be further improved through the design of mechanisms for discovering the intrinsic nature of the input to allow for choosing between single-objective or multiobjective optimizations.
4. Notwithstanding the advances made through the application of the NI-based approach, clustering algorithms originating from some of these approaches, like the gravitational search algorithm, bacterial foraging and firefly optimizations, still leave room for developing improved versions of such clustering algorithms capable of being used in the automatic clustering task.
5. Widening fields of application for clustering techniques, especially in real-time systems, is pushing for research into advancing clustering algorithms aimed at reduced time complexity for memory or vice versa. Hence, the clustering algorithm design might even be skillfully exploited for improved performance of both parameters. Nevertheless, whatever direction research on this issue takes, it will result in more effective measures for handling large datasets with varied forms of data.
6. Studies have shown that when operators are combined in algorithm design, the tendency has been to poise such algorithm to robustly handle diversity in data or population, thereby improving the quality of result within a short time [191]. Leveraging on this concept, designers of the clustering algorithm have the opportunity to develop algorithms capable of handling input with diverse attributes.
7. Similarly, clustering algorithms can be further improved to learn techniques for adapting to clusters with non-uniform sparsity and size. This adaptation mechanism must make room for handling outliers.
8. The role of the objective function in the clustering algorithm design has largely influenced the creation and advancement of different clustering algorithms, as presented in Sect. 6.1. There is therefore a need

for studies focused on investigating the performance of different objective functions across various classification techniques with the hope of cueing motivation for the design of better clustering algorithms, especially in multi-objective optimization.

9. We mentioned in Sect. 6.1.6 that ensembles of clustering algorithm are recent advances associated with modern clustering techniques. In addition to this ensemble of clustering algorithms, the researcher may consider building clustering algorithms in distributed clustering. This has become necessary due to the limitation of classical algorithms in handling huge amount of data, so that, with a sample size of a petabyte of data, clustering is a challenge.
10. Advances in ensembles of clustering algorithms may also be furthered in the need for hybridization of NI-based clustering algorithms through a reasonable combination of such algorithms in a fashionable and performance enhancement way.
11. Moreover, hybridization arising from the idea in (9) above leads to a new clustering algorithm demonstrating the properties and advantages of two or more metaheuristics-based clustering algorithms. An example of such hybridization concepts is the hybrid clustering algorithm resulting from integrating FCM algorithm with feature weighting by a three-layer NN, and a hybrid clustering approach based on MODE and GA resulted in GADE algorithm.
12. In addition, the combinatorial effect of those clustering algorithm methods has the potential to increase the computational time complexity, resource and efficiency, resulting in clustering algorithms.
13. Another option to those in (9–11) could be to attempt to integrate domain-based requirements into a new and single algorithm. The approach here looks in the direction of application of the clustering algorithm rather than the algorithm procedure itself.
14. Increasing production of computational resources with high-capacity GPU and also exploiting the advantage of parallel computing may help in devising design patterns aimed at delivering better clustering algorithms. Also, it is reported that MapReduce-based clustering algorithms promise to provide scalable and faster clustering algorithms [205].
15. New clustering algorithms can also emerge from designing solutions to some fundamental challenges of non-automatic and automatic clustering.

6.3 Trending application areas of clustering algorithms

Clustering analysis is broadly used in several real-world application areas, such as sport, education, market research, pattern recognition, data analysis, image processing, advertisements (recommender systems), big data analysis, and drug activity prediction. The focus of this section is to briefly introduce to a wider audience or data mining enthusiast some of the trending real-world application areas in which most of the state-of-the-art clustering algorithms have been applied to solve difficult clustering problems. Specifically, those with recent practical applications are of interest to us here. For example, the recent challenging application domain includes big data analysis, satellite image processing, wireless sensor networks, and gene sequence clustering in bioinformatics. However, out of the endless list of useful applications of clustering analysis, a few selected application areas are discussed below.

6.3.1 Identifying fake news

Although social media provides a platform for quick and seamless access to information, the propagation of false information, especially in recent years, has raised some major concerns, given that social media are the primary source of information for much of the world population. The impact of fake news cannot be underestimated. Simply put, fake news items often spread faster than genuine news due to their tendency to manipulate an individual's beliefs, with devastating consequences in a country where such is accepted as the norm. Therefore, one major challenge is to automatically identify false information by categorizing all articles into different types and then to notify users about the credibility of the chosen article or information-shared online. In this case, automatic clustering algorithms can easily be applied to solve the problem. In the study conducted by Hosseinimotlagh and Papalexakis [100], the authors explored the option of fake news identification using tensor decomposition ensembles. The proposed clustering algorithm presented in [100] works in such a way that it accepts as input the content of the possibly fake news article, the corpus, examining the words used in the article and then clustering such words. The clustering processes are what help the technique to distinguish between genuine and fake news. Certain words are found more commonly in sensationalized, "click-bait" articles. When a researcher sees a high percentage of specific terms in an article, it gives a higher probability of the material being fake news.

6.3.2 Spam filter

To an extent, a large portion of our daily life revolves around information sharing that relies heavily on the email system. The importance and usage emails are growing exponentially, despite the evolution of mobile applications and social networks. Data mining and analysis of emails can be conducted for several purposes such as spam detection and classification, or subject classification. Spam or junk emails are typical examples of individuals phishing for people's personal data [17, 18]. There are a record number of machine learning or clustering algorithms that have been developed to avoid receiving spam emails in the main inbox. The purpose of these clustering algorithms is to flag an email as either spam or not. It is interesting to note that a considerable number of nature-inspired clustering algorithms can be used as a powerful tool for email spam filtering.

6.3.3 Identifying fraudulent or criminal activity

Another interesting research area in which nature-inspired metaheuristic clustering algorithms could easily be applied is in crime management. Clustering algorithms can be employed to cluster certain traits associated with specific criminal behaviors committed repeatedly in a designated location [190]. Such criminal behaviors or activities could be in the form of rape, home invasion, serial murder, violent crimes, etc. For comparing criminal records, clustering algorithms are also well suited. However, in a real-life crime scene or fraudulent activity, it is usually very difficult to identify what is true and what is false. Here, the application of clustering algorithms can be very helpful in grouping similar criminal behaviors. Therefore, based on the characteristics of the different groups, it becomes much easier to classify them into clusters of genuine or fraudulent or criminal activities.

6.3.4 Fantasy football and sports

In sport, selecting the right squad to constitute a team is often a very daunting task requiring tough decisions. Similarly, deciding which players are going to perform best for a team and so allow such a team to emerge victorious in the competition depends greatly on the ability to choose the best players. The challenge at the start of the season is that there is always an insufficient or sparse dataset available to facilitate identifying the winning players, thereby helping in the decision-making process. Despite this limitation with dataset availability, there are some instances where k-means and high-profile machine learning clustering algorithms have been used to find similar players based on some of their unique characteristics [134]. One major advantage of using machine learning clustering techniques, in this case, is that one can carefully select a better team more easily at the start of the year, thereby presenting a very high probability or chance of winning a game.

6.3.5 Big data analysis

Another very interesting application area for clustering algorithms is in big data analysis. The serious challenge with big data clustering lies in the need to develop robust, fast, scalable iterative clustering algorithms that converge faster and also give higher performance, better accuracy, and reduced error rate [205]. Most of the traditional clustering techniques, which are susceptible to entrapment into local optima or are problem specific, would find it very challenging to process terabytes or petabytes of data. Therefore, there is a need to develop clustering algorithms that are able to cope well with the high computational cost associated with big data analysis. The nature-inspired clustering algorithms are known to be very scalable and efficient in terms of handling high-dimensional or large-scale datasets and therefore could be considered as more appropriate clustering techniques for processing larger datasets.

6.3.6 Search engines

A clustering algorithm is a backbone behind search engines' technology. The search engines employ clustering strategy to group similar entities into one cluster and dissimilar entities into another cluster, with the overall goal of maintaining perfect separateness relative to compactness and cohesion between dissimilar objects. Distance measures are used to provide results for the searched data according to the nearest similar object, which is clustered around the data to be searched [141]. Those clustering algorithms that require little or no supervision in correlating similarity between similar objects from search queries and stored data objects are considered the best techniques, with greater chances of getting the required result on the front page. The nature-inspired clustering algorithms capable of automatic clustering are even seen as the best fits for developing intelligent web search engines.

6.3.7 Educational data mining

Educational data mining plays a major role in monitoring the progress of students' academic performance in any institution of learning. The quality of student academic evaluation is closely tied to using the best assessment performance metrics and algorithms. However, developing and applying such an efficient algorithm has been a critical issue for the academic community of higher learning. Some

well-known standard clustering algorithms such as the k-means, k-medoids, fuzzy c-means, and expectation maximization algorithms, have been used to develop efficient clustering techniques, which are able to monitor students' academic performance [61]. These algorithms are able to conduct unsupervised learning-based clustering tasks using students' raw academic scores to help classify each student into a well-defined cluster that clearly defines the behavior and learning style of all students in the cluster.

6.3.8 Customer segmentation

The task of customer segmentation requires that producers, business owners and sellers effectively categorize their customers into groups that are intelligible enough to provide sensitive information for improving sales or performance. Traditional approaches require the use of business analytical packages, which have not demonstrated robustness and efficiency in the face of huge data. Hence, the use of clustering techniques has generated interesting performance in this area. Major clustering models and algorithms such as k-means, agglomerative hierarchical, hierarchical and non-hierarchical clustering algorithms have been adopted for different research needs. Segmentation of customers may assume forms that use customer behaviors, demography, psychographics, needs, wants, characteristics and even geographical place, such that each segment comprises customers who share similar market characteristics. Ezenkwu et al. [66] and Kansal et al. [112] are good examples of researchers who have attempted to apply clustering techniques and algorithms to solve this problem of customer segmentation. The first study applied k-means clustering algorithm to the issue of segmenting customers while adopting MATLAB platform for implementing the algorithm, which they trained using a z-score normalized two-feature dataset. The approach was able to generate four clusters of customers from the dataset containing features of customer information as related to the average amount of goods purchased and the average number of customer visits. As a result, they were able to class customers as High-Buyers-Regular-Visitors (HBRV), High-Buyers-Irregular-Visitors (HBIV), Low-Buyers-Regular-Visitors (LBRV) and Low-Buyers-Irregular-Visitors (LBIV) [66]. Similarly, the second study also applied the k-means clustering algorithm in addition to agglomerative, and meanshift clustering-based algorithms in segmenting customers. They also applied their method to a dataset containing features of customers based on their mean amount of shopping and average visits to the shop. As a result, their studies generated five segments of clusters: Careless, Careful, Standard, Target and Sensible customers. Furthermore, the application of mean shift clustering algorithm further revealed other segments of the cluster with characteristics of High Buyers and Frequent Visitors (HBFV) and High Buyers and Occasional Visitors (HBOV) [112].

6.3.9 Recommender system

Recommender system, also known as RS, is built to provide support in automating the process of deriving proposals very close to the user's real interest. Such a system can accurately predict a user's opinion or even ratings on issues, products, or services by leveraging the user's past response or behavior. Clustering techniques have been widely applied to this task of recommending appropriate items or services to users. On the one hand, authors in [154] used the learning-based Gravitational Search Algorithm and Learning Automata (GSA-LA), an approach of single-objective hybrid evolutionary clustering technique for grouping items in the offline collaborative filtering RS. On the other hand, Kużelewska [132] developed a clustering method for identifying similar features existing among users as well as their profiles in order to aid the process of the user's selection of products [132]. Their similarity measurement models were those of Euclidean distance, log-likelihood function, correlation coefficient, and cosine.

6.3.10 Wireless sensor network's based application

The use of wireless sensor networks (WSNs) has gained wide acceptance in different fields of research. Health facilities, home surveillance systems, educational or institutional centers, military establishments, and even industry are areas where WSNs are frequently used. However, the application of clustering technique to the use of WSNs is becoming even more attractive because of the problems of energy efficiency of nodes and lifetime of the network as it relates to WSNs. Hence, clustering algorithms can be adopted to improve the energy of WSNs nodes and as well as the scalability of the nodes. Singh et al. [210] presented the application of clustering algorithms to the challenges listed earlier. Their work outlines different studies that have successfully applied clustering algorithms to this task. They revealed that different approaches of clustering have used centralized, distributed, hybrid, equal and unequal clustering techniques in solving the problem of sensor networks [210]. Furthermore, the authors stated that most of the studies using these approaches leverage the residual energy of nodes and distance to the base station as parameters for selecting cluster heads. Zanjireh et al. [240], by optimizing the distribution of cluster heads across the network, were able to derive a new clustering algorithm for wireless sensor networks, which they claim successfully

reduced the energy consumption of the network, thereby prolonging the lifetime of the node [240].

6.3.11 Drug activity prediction

The application of clustering techniques and their related algorithms have also attracted the interest of pharmaceutical companies. Complex and larger pharmacology networks can be made simpler through the use of clustering algorithms for clustering existing drugs into new groups. These new clusters can form the basis for repositioning or repurposing the drugs for other use. This classification task often takes the approach of first selecting compound, virtual library generation, High-Throughput Screening (HTS), Quantitative Structure–Activity Relationship (QSAR) Analysis and Absorption, Distribution, Metabolism, Elimination and Toxicity (ADMET) prediction. Malhat et al. [146] presented chemoinformatics clustering algorithms by using k-means, bisecting k-means, and Ward clustering algorithms for drug discovery processes. Besides this, the authors also carried out a comparative analysis of the performances of the algorithm for the discovery of the most effective algorithm in this task. Furthermore, they applied the algorithms over homogeneous and heterogeneous chemical datasets, thereby obtaining a result that shows the k-means algorithm is more fit for a small number of clusters while bisecting k-means and Ward algorithms are more fit for large clusters formation for homogeneous and heterogeneous data sets in terms of time and standard deviation. Similarly, Hameed et al. [91] applied two-tiered drug-centric unsupervised clustering algorithms, which they proposed for drug repositioning. Their clustering approach to solving the problem of repositioning drugs was achieved through an integration of information such as drug–chemical, drug–disease, drug–gene, drug–protein and drug–side effects in first clustering drugs by using the Growing Self Organizing Map (GSOM) based on their homogenous information. Secondly, the authors performed clustering of the resulting previous groups using drug–drug relation matrices [91].

7 Summary

In real life, the relevance of clustering analysis cannot be overemphasized, specifically for those application areas where decision making and exploratory pattern analysis are required to be carried out on large-scale datasets. In most cases, extracting the essential information from several millions of data samples with tens of thousands of dimensionalities is a very daunting task to embark on. Therefore, the only methods that have worked very well with data analysis of such magnitude are those unsupervised learning algorithms that are equipped with mechanisms that can efficiently help computing resource to easily explore and understand the detailed structural composition of any data objects. The recent developments in the application of nature-inspired metaheuristic optimization algorithms have paved the way for researchers to develop some of the simplest yet most robust data abstraction and analysis tools, which do not require any prior knowledge of the data to be processed. It is noteworthy to mention that several studies have revealed that nature-inspired metaheuristic clustering algorithms are ideally suited for achieving better, simple and compact representations of data objects in complex and large-scale datasets.

This paper has provided a detailed, state-of-the-art collection in the form of a comprehensive overview and bibliometric analysis of the well-known clustering algorithms. A taxonomy of clustering algorithms is presented and discussed. The focus is centered on clustering algorithms and automatic clustering algorithms in the bibliometric analysis. The publications and citation structures are analyzed from the early 1990s to 2019. A total of 5063 papers were extracted, among which 97.41% were articles. Publication numbers and citation growth have increased significantly every year, which may be attributed to the current era of large datasets in almost all domains. Four papers each had more than 3000 citations. *Pattern Recognition* is the journal with the highest number of publications. As to authors, Bezdek has been the most productive (most number of papers), while Jain is highly influential (most number of citations). China has published the highest number of papers, while those from the USA have been cited several times. Among institutions, the Chinese Academy of Sciences is the most productive and Michigan State University is the most influential. Top-most keywords used by authors are clustering, data mining, clustering algorithms, etc. For each of the bibliometric indicators, we have also presented visualizations using the VOSviewer.

Furthermore, an all-inclusive review of the metaheuristic clustering approaches from the early 1990s to date was presented. The current study has introduced the basic and core idea of some of the commonly used clustering algorithms; specifically, this paper has focused on reviewing nature-inspired metaheuristic clustering techniques with the primary aim of identifying their common inspirational sources, taxonomical classification, advantages and disadvantages of each. However, it is burdensome to present a complete list of all the clustering algorithms due to the intersection of research fields and the diversity in application areas. Therefore, this study has only considered the well-researched and commonly used clustering algorithms, with high practical values. The overall review of the

clustering algorithms presented in this paper has the major goal of providing interested readers with a systematic and clear understanding of the importance of existing data analysis algorithms and their application areas.

In conclusion, the subject of data clustering or analysis is considered to be an interesting, useful, and challenging problem. Moreover, the tasks of clustering involve a very complex sequence of processes that must be carefully sorted and executed in order to obtain any meaningful result from the candidate datasets. It is also interesting to note that literature in the area of clustering is quite diverse, with so much practical potential in real-world application areas such as pattern recognition, marketing and sales research, predictive gaming, web network traffic classification, and document filtering and retrieval. However, there are still some major concerns with the problem of dealing with large-scale datasets, in terms of determining the number of clusters automatically, selection of clustering methods and more efficient automatic clustering algorithms to handle real-world clustering problems. Nevertheless, researches in these areas are still very active within the research community. Finally, considering the relevance of clustering tasks to most real-world problems, it is still possible to explore and exploit further application potential areas with the most efficient data abstraction algorithms, specifically, using state-of-the-art nature-inspired metaheuristic clustering algorithms.

As a way forward, considering the large volume of literature available in clustering and its applications, it is possible that the current study missed out some recently published related clustering methods. Therefore, we recommend the consideration of this specific limitation in any future research. Further, it will also be interesting to consider extending the current literature review to include a more constructive discussion on the merits and demerits of all the reviewed state-of-the-art clustering algorithms that are presented and discussed in this paper.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of the paper.

References

1. Abbasi AA, Younis M (2007) A survey on clustering algorithms for wireless sensor networks. Comput Commun 30(14–15):2826–2841
2. Abdulwahab HA, Noraziah A, Alsewari AA, Salih SQ (2019) An enhanced version of black hole algorithm via levy flight for optimization and data clustering problems. IEEE Access 7:142085–142096
3. Abraham A, Das S, Konar A (2007) Kernel based automatic clustering using modified particle swarm optimization algorithm. In: Proceedings of the 9th annual conference on genetic and evolutionary computation. ACM, pp 2–9
4. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795
5. Abualigah LM, Khader AT, Al-Betar MA (2016) Multi-objectives-based text clustering technique using K-mean algorithm. In: 2016 7th international conference on computer science and information technology (CSIT). IEEE, pp 1–6
6. Abualigah LM, Khader AT, Hanandeh ES (2018) A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering. Intell Decis Technol 12(1):3–14
7. Abualigah LM, Khader AT, Al-Betar MA, Alomari OA (2017) Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Expert Syst Appl 84:24–36
8. Abualigah LM, Khader AT, Al-Betar MA, Awadallah MA (2016) A krill herd algorithm for efficient text documents clustering. In: 2016 IEEE symposium on computer applications & industrial electronics (ISCAIE). IEEE, pp 67–72
9. Abualigah LM, Khader AT, AlBetar MA, Hanandeh ES (2016) A new hybridization strategy for krill herd algorithm and harmony search algorithm applied to improve the data clustering. In: 1st EAI international conference on computer science and engineering. European Alliance for Innovation (EAI), p 54
10. Abualigah LM, Khader AT, AlBetar MA, Hanandeh ES (2017) Unsupervised text feature selection technique based on particle swarm optimization algorithm for improving the text clustering. In: EAI international conference on computer science and engineering
11. Abubaker A, Baharum A, Alrefaei M (2015) Automatic clustering using multi-objective particle swarm and simulated annealing. PLoS One 10(7):e0130995
12. Agarwal P, Mehta S (2016) Enhanced flower pollination algorithm on data clustering. Int J Comput Appl 38(2–3):144–155
13. Agarwal P, Alam MA, Biswas R (2011) Issues, challenges and tools of clustering algorithms. arXiv preprint arXiv:1110.2610
14. Agbaje MB, Ezugwu AE, Els R (2019) Automatic data clustering using hybrid firefly particle swarm optimization algorithm. IEEE Access 7:184963–184984
15. Aggarwal CC (ed) (2014) Data classification: algorithms and applications. CRC Press, Boca Raton
16. Agusti LE, Salcedo-Sanz S, Jiménez-Fernández S, Carro-Calvo L, Del Ser J, Portilla-Figueras JA (2012) A new grouping genetic algorithm for clustering problems. Expert Syst Appl 39(10):9695–9703
17. Akinyelu AA, Ezugwu AE (2019) Nature inspired instance selection techniques for support vector machine speed optimization. IEEE Access 7:154581–154599
18. Akinyelu AA, Ezugwu AE, Adewumi AO (2019) Ant colony optimization edge selection for support vector machine speed optimization. Neural Comput Appl 32:1–33
19. Akyol S, Alatas B (2017) Plant intelligence based metaheuristic optimization algorithms. Artif Intell Rev 47(4):417–462
20. Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180
21. Aliniya Z, Mirroshandel SA (2019) A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm. Expert Syst Appl 117:243–266
22. Alswaitti M, Albughdadi M, Isa NAM (2018) Density-based particle swarm optimization algorithm for data clustering. Expert Syst Appl 91:170–186

23. Anand N, Vikram P (2015) Comprehensive analysis & performance comparison of clustering algorithms for big data. Rev Comput Eng Res 4:54–80
24. Anari B, Torkestani JA, Rahmani AM (2017) Automatic data clustering using continuous action-set learning automata and its application in segmentation of images. Appl Soft Comput 51:253–265
25. Arbelaitz O, Gurrutxaga I, Muguerza J, PéRez JM, Perona I (2013) An extensive comparative study of cluster validity indices. Pattern Recognit 46(1):243–256
26. Atabay HA, Sheikhzadeh MJ, Torshizi M (2016) A clustering algorithm based on integration of K-means and PSO. In: 2016 1st conference on swarm intelligence and evolutionary computation (CSIEC). IEEE, pp 59–63
27. Baker FB, Hubert LJ (1975) Measuring the power of hierarchical cluster analysis. J Am Stat Assoc 70(1975):31–38
28. Banati H, Bajaj M (2013) Performance analysis of firefly algorithm for data clustering. Int J Swarm Intell 1(1):19–35
29. Bandyopadhyay S, Saha S (2008) A point symmetry-based clustering technique for automatic evolution of clusters. IEEE Trans Knowl Data Eng 20(2008):1441–1457
30. Berkhin P (2006) A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping multidimensional data. Springer, Berlin, pp 25–71
31. Bezdek JC, Pal NR (1998) Some new indexes of cluster validity. IEEE Trans Syst Man Cyber Part B 28(3):301–315
32. Bezdek JC (2013) Pattern recognition with fuzzy objective function algorithms. Springer, Berlin
33. Blanco-Mesa F, León-Castro E, Merigó JM (2019) A bibliometric analysis of aggregation operators. Appl Soft Comput 81:105488
34. Blanco-Mesa F, Merigó JM, Gil-Lafuente AM (2017) Fuzzy decision making: a bibliometric-based review. J Intell Fuzzy Syst 32(3):2033–2050
35. Boryczka U (2009) Finding groups in data: cluster analysis with ants. Appl Soft Comput 9(1):61–70
36. Bouyer A, Ghafarzadeh H, Tarkhaneh O (2015) An efficient hybrid algorithm using cuckoo search and differential evolution for data clustering. Indian J Sci Technol 8(24):1–12
37. Calinski T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat 3(1974):1–27
38. Chang DX, Zhang XD, Zheng CW, Zhang DM (2010) A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem. Pattern Recognit 43(4):1346–1360
39. Chang H, Yeung DY (2008) Robust path-based spectral clustering. Pattern Recognit 41(1):191–203
40. Chou CH, Su MC, Lai E (2004) A new cluster validity measure and its application to image compression. Pattern Anal Appl 7(2):205–220
41. Chowdhury A, Bose S, Das S (2011) Automatic clustering based on invasive weed optimization algorithm. In: International conference on swarm, evolutionary, and memetic computing. Springer, Berlin, Heidelberg, pp 105–112
42. Chu Y, Mi H, Liao H, Ji Z, Wu QH (2008) A fast bacterial swarming algorithm for high-dimensional function optimization. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence). IEEE, pp 3135–3140
43. Chuang LY, Hsiao CJ, Yang CH (2011) Chaotic particle swarm optimization for data clustering. Expert Syst Appl 38(12):14555–14563
44. Condorcet MJAN (2014) "Essai sur l"Application de l"Analyse 'a la Probabilité des decisions rendues a la Pluralité des Voix," paris: L"Imprimerie Royale, 1785
45. Corter, J. E. and Gluck, M. A. (1992). "Explaining basic categories: Feature predictability and information," Psychological Bulletin, vol. 111, no. 2, pp 291–303, 1992
46. Cowgill MC, Harvey RJ, Watson LT (1999) A genetic algorithm approach to cluster analysis. Comput Math Appl 37(7):99–108
47. Cruz DPF, Maia RD, de Castro LN (2013) A new encoding scheme for a bee-inspired optimal data clustering algorithm. In: 2013 BRICS congress on computational intelligence and 11th Brazilian congress on computational intelligence. IEEE, pp 136–141
48. Cura T (2012) A particle swarm optimization approach to clustering. Expert Syst Appl 39(1):1582–1588
49. Dalrymple-Alford EC (1970) The measurement of clustering in free recall. Psycol. Bull. 74:32–34
50. Das S, Abraham A, Konar A (2007) Automatic clustering using an improved differential evolution algorithm. IEEE Trans Syst Man Cybern Part A Syst Hum 38(1):218–237
51. Das S, Abraham A, Konar A (2008) Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. Pattern Recognit Lett 29(5):688–699
52. Das S, Chowdhury A, Abraham A (2009) A bacterial evolutionary algorithm for automatic data clustering. In: 2009 IEEE congress on evolutionary computation. IEEE, pp 2403–2410
53. Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution—an updated survey. Swarm Evol Comput 27:1–30
54. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 2:224–227
55. Dorigo M, Birattari M (2010) Ant colony optimization. Springer, Berlin, pp 36–39
56. Dorigo M, Stützle T (2019) Ant colony optimization: overview and recent advances. In: Gendreau M, Potvin JY (eds) Handbook of metaheuristics. Springer, Cham, pp 311–351
57. Drewes B (2005) Some industrial applications of text mining. In: Knowledge mining. StudFuzz, vol 185. Springer, Berlin, Heidelberg, pp 233–238
58. Duan G, Hu W, Zhang Z (2016) A novel data clustering algorithm based on modified adaptive particle swarm optimization. Int J Signal Process Image Process Pattern Recognit 9(3):179–188
59. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York
60. Dunn JC (1973) A Fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cyber 3(1973):32–57
61. Dutt A, Ismail MA, Herawan T (2017) A systematic review on educational data mining. IEEE Access 5:15991–16005
62. Dutta D, Dutta P, Sil J (2012) Data clustering with mixed features by multi objective genetic algorithm. In: 2012 12th international conference on hybrid intelligent systems (HIS). IEEE, pp 336–341
63. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43
64. Elaziz MA, Nabil NEGGAZ, Ewees AA, Lu S (2019) Automatic data clustering based on hybrid atom search optimization and sine–cosine algorithm. In: 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp 2315–2322
65. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol 96, No. 34, pp 226–231
66. Ezenkwu CP, Ozuomba S, Kalu C (2015) Application of K-means algorithm for efficient customer segmentation: a strategy for targeted customer services. Int J Adv Res Artif Intell (IJARAI) 4(10):40–44

67. Ezugwu AE (2019) Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times. Knowl-Based Syst 172:15–32
68. Ezugwu AES, Adewumi AO (2017) Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Syst Appl 87:70–78
69. Ezugwu AE, Adewumi AO (2017) Soft sets based symbiotic organisms search algorithm for resource discovery in cloud computing environment. Future Gener Comput Syst 76:33–50
70. Ezugwu AE, Akutsah F (2018) An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times. IEEE Access 6:54459–54478
71. Ezugwu AE, Prayogo D (2019) Symbiotic organisms search algorithm: theory, recent advances and applications. Expert Syst Appl 119:184–209
72. Ezugwu AE, Adeleke OJ, Viriri S (2018) Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times. PLoS One 13(7):e0200030
73. Ezugwu AE, Adeleke OJ, Akinyelu AA, Viriri S (2019) A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. Neural Comput Appl 32:1–45
74. Ezugwu AE, Akutsah F, Olusanya MO, Adewumi AO (2018) Enhanced intelligent water drops algorithm for multi-depot vehicle routing problem. PLoS One 13(3):e0193751
75. Ezugwu AE (2020) Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. SN Appl Sci 2(2):273
76. Fahad A, Alshatri N, Tari Z, Alamri A, Khalil I, Zomaya AY, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput 2(3):267–279
77. Falkenauer E (1998) Genetic algorithms and grouping problems. Wiley, Chichester
78. Fisher DH (1987) Knowledge acquisition via incremental conceptual clustering. Mach Learn 2(2):139–172. https://doi.org/10.1007/BF00114265
79. Fister I, Fister I Jr, Yang XS, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46
80. Fortier JJ, Solomon H (1966) Clustering procedures. In: Krishnaiah PR (ed) Multivariate analysis, vol 62. Academic Press, New York
81. Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. J Am Stat Assoc 78(383):553–569
82. Gan G, Ma C, Wu J (2007) Data clustering: theory, algorithms, and applications, vol 20. SIAM, Philadelphia
83. Gluck MA, Corter JE (1985) Information, uncertainty, and the utility of categories. In: Program of the 7th annual conference of the cognitive science society, pp 283–287
84. Goel S, Sharma A, Bedi P (2011) Cuckoo search clustering algorithm: a novel strategy of biomimicry. In: 2011 world congress on information and communication technologies. IEEE, pp 916–921
85. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99
86. Guha S, Rastogi R, Shim K (2001) Cure: an efficient clustering algorithm for large databases. Inf Syst 26(1):35–58
87. Guo D, Chen J, Chen Y, Li Z (2018) LBIRCH: an improved BIRCH algorithm based on link. ICMLC 2018:74–79
88. Halkidi M, Vazirgiannis M (2001) Clustering validity assessment: finding the optimal partitioning of a data set. In: Proceedings 2001 IEEE international conference on data mining. IEEE, pp 187–194
89. Halkidi M, Batistakis Y, Vazirgiannis M (2002) Clustering validity checking methods: part II. ACM Sigmod Rec 31(3):19–27
90. Halkidi M, Vazirgiannis M, Batistakis I (2000) Quality scheme assessment in the clustering process. In: Proceedings of PKDD, Lyon, France
91. Hameed PN, Verspoor K, Kusljic S, Halgamuge S (2018) A two-tiered unsupervised clustering approach for drug repositioning through heterogeneous data integration. BMC Bioinform 19:29
92. Hamerly G, Elkan C (2004) Learning the k in k-means. In: Advances in neural information processing systems. MIT Cambridge Press, 2003, pp 281–288
93. Hamilton JD (1994) Time series analysis. Princeton University Press, Princeton, p 159
94. Hartigan JA, Wong MA (1979) Algorithm AS 136: A k-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):100–108
95. Hassanzadeh T, Meybodi MR (2012) A new hybrid approach for data clustering using firefly algorithm and K-means. In: The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012). IEEE, pp 007–011
96. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci 222:175–184
97. He H, Tan Y (2012) A two-stage genetic algorithm for automatic clustering. Neurocomputing 81:49–59
98. He Q, Jin X, Du C, Zhuang F, Shi Z (2014) Clustering in extreme learning machine feature space. Neurocomputing 128:88–95
99. Hoos HH, Stützle T (2004) Stochastic local search: foundations and applications. Elsevier, Amsterdam
100. Hosseinimotlagh S, Papalexakis EE (2018) Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. In: Proceedings of the workshop on misinformation and misbehavior mining on the web (MIS2)
101. Hruschka ER, Campello RJ, Freitas AA (2009) A survey of evolutionary algorithms for clustering. IEEE Trans Syst Man Cybern Part C (Appl Rev) 39(2):133–155
102. Huang CL, Huang WC, Chang HY, Yeh YC, Tsai CY (2013) Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. Appl Soft Comput 13(9):3864–3872
103. Hussain K, Salleh MNM, Cheng S, Shi Y (2019) Metaheuristic research: a comprehensive survey. Artif Intell Rev 52(4):2191–2233
104. Jaccard P (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bulletin de la Société Vaudoise des Sciences Naturelles 37:241–272
105. Jafar OM, Sivakumar R (2010) Ant-based clustering algorithms: a brief survey. Int J Comput Theory Eng 2(5):787
106. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31(8):651–666
107. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323
108. Jakobsson M, Rosenberg NA (2007) CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. Bioinformatics 23(14):1801–1806
109. Janmaijaya M, Shukla AK, Abraham A, Muhuri PK (2018) A scientometric study of neurocomputing publications (1992–2018): an aerial overview of intrinsic structure. Publications 6(3):32
110. Jensi R, Jiji GW (2015) MBA-LF: a new data clustering method using modified BAT algorithm and levy flight. ICTACT J Soft Comput 6(1):1093–1101

111. José-García A, Gómez-Flores W (2016) Automatic clustering using nature-inspired metaheuristics: a survey. Appl Soft Comput 41:192–213
112. Kansal T, Bahuguna S, Singh V, Choudhury T (2018) Customer segmentation using K-means clustering. In: 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)
113. Kao Y, Chen CC (2014) Automatic clustering for generalised cell formation using a hybrid particle swarm optimisation. Int J Prod Res 52(12):3466–3484
114. Kapoor S, Zeya I, Singhal C, Nanda SJ (2017) A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation. Proc Comput Sci 115:415–422
115. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, vol 200. Technical report-tr06. Erciyes Universityersity, Engineering Faculty, Computer Engineering Department, pp 1–10
116. Karaboğa D, Ökdem S (2004) A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turk J Electr Eng Comput Sci 12(1):53–60
117. Karaboga D, Ozturk C (2011) A novel clustering approach: artificial bee colony (ABC) algorithm. Appl Soft Comput 11(1):652–657
118. Karthikeyan M, Aruna P (2013) Probability based document clustering and image clustering using content-based image retrieval. Appl Soft Comput 13(2):959–966
119. Karypis G, Han EH, Chameleon VK (1999) A hierarchical clustering algorithm using dynamic modeling. IEEE Comput 32(8):68–75
120. Kaushik K, Arora V (2015) A hybrid data clustering using firefly algorithm based improved genetic algorithm. Proc Comput Sci 58:249–256
121. Kosters WA, Laros JF (2007) Metrics for mining multisets. In: International conference on innovative techniques and applications of artificial intelligence. Springer, London, pp 293–303
122. Kotsiantis S, Pintelas EP (2004) Recent advances in clustering: a brief survey. WSEAS Trans Inf Sci Appl 1:73–81
123. Kovács F, Ivancsy R (2006) Cluster validity measurement for arbitrary shaped clustering. In: Proceeding of the 5th. WSEAS international conference on artificial, knowledge engineering and data bases, Madrid, Spain, February 15–17, 2006, pp 372–377
124. Kumar V, Chhabra JK, Kumar D (2014) Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. Eng Appl Artif Intell 29:93–103
125. Kumar V, Chhabra JK, Kumar D (2016) Automatic data clustering using parameter adaptive harmony search algorithm and its application to image segmentation. J Intell Syst 25(4):595–610
126. Kumar Y, Sahoo G (2014) A review on gravitational search algorithm and its applications to data clustering & classification. Int J Intell Syst Appl 6(6):79
127. Kundu D, Suresh K, Ghosh S, Das S, Abraham A, Badr Y (2009) Automatic clustering using a synergy of genetic algorithm and multi-objective differential evolution. In: International conference on hybrid artificial intelligence systems. Springer, Berlin, Heidelberg, pp 177–186
128. Kuo RJ, Zulvia FE (2018) Automatic clustering using an improved artificial bee colony optimization for customer segmentation. Knowl Inf Syst 57(2):331–357
129. Kuo RJ, Huang YD, Lin CC, Wu YH, Zulvia FE (2014) Automatic kernel clustering with bee colony optimization algorithm. Inf Sci 283:107–122
130. Kuo RJ, Syu YJ, Chen ZY, Tien FC (2012) Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Inf Sci 195:124–140
131. Kuo R, Zulvia F (2013) Automatic clustering using an improved particle swarm optimization. J Ind Intell Inf 1(1):46–51
132. Kużelewska U (2014) Clustering algorithms in hybrid recommender system on MovieLens data. Stud Log Gramm Rhetor 37(50):125–139
133. Lago-Fernández LF, Corbacho F (2010) Normality-based validation for crisp clustering. Pattern Recognit 43(3):782–795
134. Landers JR, Duperrouzel B (2018) Machine learning approaches to competing in fantasy leagues for the NFL. IEEE Trans Games 11(2):159–172
135. Lashkari M, Moattar MH (2015) The improved K-means clustering algorithm using the proposed extended PSO algorithm. In: 2015 international congress on technology, communication and knowledge (ICTCK). IEEE, pp 429–434
136. Lee WP, Chen SW (2010) Automatic clustering with differential evolution using cluster number oscillation method. In: 2010 2nd international workshop on intelligent systems and applications. IEEE, pp 1–4
137. Legány C, Juhász S, Babos A (2006) Cluster validity measurement techniques. In: Proceeding of the 5th. WSEAS international conference on artificial, knowledge engineering and data bases, Madrid, Spain, February 15–17, 2006, pp 388–393
138. Ling HL, Wu JS, Zhou Y, Zheng WS (2016) How many clusters? A robust PSO-based local density model. Neurocomputing 207:264–275
139. Liu R, Wang X, Li Y, Zhang X (2012) Multi-objective invasive weed optimization algortihm for clustering. In: 2012 IEEE congress on evolutionary computation. IEEE, pp 1–8
140. Liu R, Zhu B, Bian R, Ma Y, Jiao L (2015) Dynamic local search based immune automatic clustering algorithm and its applications. Appl Soft Comput 27:250–268
141. Liu T, Rosenberg C, Rowley HA (2007) Clustering billions of images with large scale nearest neighbor search. In: 2007 IEEE workshop on applications of computer vision (WACV'07). IEEE, p 28
142. Liu X, Fu H (2010) An effective clustering algorithm with ant colony. JCP 5(4):598–605
143. Liu Y, Wu X, Shen Y (2011) Automatic clustering using genetic algorithms. Appl Math Comput 218(4):1267–1279
144. MacQueen J (1967) 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, pp 281–297
145. Majhi SK, Biswal S (2018) Optimal cluster analysis using hybrid K-means and ant lion optimizer. Karbala Int J Mod Sci 4(4):347–360
146. Malhat MG, Mousa HM, El-Sisi AB (2014) Clustering of chemical data sets for drug discovery. In: 2014 9th international conference on informatics and systems
147. Marinakis Y, Marinaki M, Matsatsinis N (2009) A hybrid discrete artificial bee colony-GRASP algorithm for clustering. In: 2009 international conference on computers & industrial engineering. IEEE, pp 548–553
148. Masoud H, Jalili S, Hasheminejad SMH (2013) Dynamic clustering using combinatorial particle swarm optimization. Appl Intell 38(3):289–314
149. Maulik U, Saha I (2009) Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. Pattern Recognit 42(9):2135–2149
150. Maulik U, Saha I (2010) Automatic fuzzy clustering using modified differential evolution for image classification. IEEE Trans Geosci Remote Sens 48(9):3503–3510
151. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1(4):355–366

152. Merigó JM, Cobo MJ, Laengle S, Rivas D, Herrera-Viedma E (2019) Twenty years of soft computing: a bibliometric overview. Soft Comput 23(5):1477–1497
153. Milligan GW, Cooper MC (1987) Methodology review: clustering methods. Appl Psychol Meas 11(4):329–354
154. Mohammadpour T, Bidgoli AM, Enayatifar R, Javadi HH (2019) Efficient clustering in collaborative filtering recommender system: hybrid method based on genetic algorithm and gravitational emulation local search algorithm. Genomics 111(6):1902–1912
155. Molina D, Poyatos J, Del Ser J, García S, Hussain A, Herrera F (2020) Comprehensive taxonomies of nature-and bio-inspired optimization: inspiration versus algorithmic behavior, critical analysis and recommendations. arXiv preprint arXiv:2002.08136
156. Mouton JP, Ferreira M, Helberg SJA (2020) A comparison of clustering algorithms for automatic modulation classification. Expert Syst Appl 151:113317
157. Muhuri PK, Shukla AK, Abraham A (2019) Industry 4.0: a bibliometric analysis and detailed overview. Eng Appl Artif Intell 78:218–235
158. Muhuri PK, Shukla AK, Janmaijaya M, Basu A (2018) Applied soft computing: a bibliometric analysis of the publications and citations during (2004–2016). Appl Soft Comput 69:381–392
159. Murty MR, Naik A, Murthy JVR, Reddy PP, Satapathy SC, Parvathi K (2014) Automatic clustering using teaching learning based optimization. Appl Math 5(08):1202
160. Nanda SJ, Panda G (2013) Automatic clustering algorithm based on multi-objective immunized PSO to classify actions of 3D human models. Eng Appl Artif Intell 26(5–6):1429–1441
161. Nayak J, Kanungo DP, Naik B, Behera HS (2016) Evolutionary improved swarm-based hybrid K-means algorithm for cluster analysis. In: Proceedings of the second international conference on computer and communication technologies. Springer, New Delhi, pp 343–352
162. Nayak J, Nanda M, Nayak K, Naik B, Behera HS (2014) An improved firefly fuzzy c-means (FAFCM) algorithm for clustering real world data sets. In: Advanced computing, networking and informatics, vol 1. Springer, Cham, pp 339–348
163. Nayyar A, Puri V (2017) Comprehensive analysis & performance comparison of clustering algorithms for big data. Rev Comput Eng Res 4(2):54–80
164. Nerurkar P, Shirke A, Chandane M, Bhirud S (2018) Empirical analysis of data clustering algorithms. Proc Comput Sci 125:770–779
165. Niknam T, Olamaie J, Amiri B (2008) A hybrid evolutionary algorithm based on ACO and SA for cluster analysis. J Appl Sci 8(15):2695–2702
166. Niu B, Wang H (2012) Bacterial colony optimization. Discrete Dyn Nat Soc 2012:698057. https://doi.org/10.1155/2012/698057
167. Omran MG, Salman A, Engelbrecht AP (2006) Dynamic clustering using particle swarm optimization with application in image segmentation. Pattern Anal Appl 8(4):332
168. Ozturk C, Hancer E, Karaboga D (2015) Dynamic clustering with improved binary artificial bee colony algorithm. Appl Soft Comput 28:69–80
169. Pacheco TM, Gonçalves LB, Ströele V, Soares SSR (2018) An ant colony optimization for automatic data clustering problem. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
170. Pal NR, Biswas J (1997) Cluster validation using graph theoretic concepts. Pattern Recognit 30(6):847–857
171. Paterlini S, Krink T (2006) Differential evolution and particle swarm optimisation in partitional clustering. Comput Stat Data Anal 50(5):1220–1247
172. Pelleg D (2000) Extending K-means with efficient estimation of the number of clusters in ICML. In: Proceedings of the 17th international conference on machine learning, pp 277–281
173. Peng H, Wang J, Shi P, Riscos-Núñez A, Pérez-Jiménez MJ (2015) An automatic clustering algorithm inspired by membrane computing. Pattern Recognit Lett 68:34–40
174. Raftery A (1986) A note on Bayes factors for log-linear contingency table models with vague prior information. J R Stat Soc 48(2):249–250
175. Rahman MA, Islam MZ (2014) A hybrid clustering technique combining a novel genetic algorithm with K-means. Knowl-Based Syst 71:345–365
176. Rajah V, Ezugwu AE (2020) Hybrid symbiotic organism search algorithms for automatic data clustering. In: 2020 conference on information communications technology and society (ICTAS). IEEE, pp 1–9
177. Rajpurohit J, Sharma TK, Abraham A, Vaishali A (2017) Glossary of metaheuristic algorithms. Int J Comput Inf Syst Ind Manag Appl 9:181–205
178. Ramadas M, Abraham A (2019) Metaheuristics for data clustering and image segmentation. Springer, Berlin
179. Rana S, Jasola S, Kumar R (2010) A hybrid sequential approach for data clustering using K-Means and particle swarm optimization algorithm. Int J Eng Sci Technol 2(6)
180. Rana S, Jasola S, Kumar R (2013) A boundary restricted adaptive particle swarm optimization for data clustering. Int J Mach Learn Cybern 4(4):391–400
181. Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
182. Rani Y, Rohil H (2013) A study of hierarchical clustering algorithm. Int J Inf Comput Technol 3(10):1115–1122
183. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
184. Raposo C, Antunes CH, Barreto JP (2014) Automatic clustering using a genetic algorithm with new solution encoding and operators. In: International conference on computational science and its applications. Springer, Cham, pp 92–103
185. Razmjooy N, Khalilpour M, Ramezani M (2016) A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. J Control Autom Electr Syst 27(4):419–440
186. Rendon LE, Garcia R, Abundez I, Gutierrez C et al (2002) Niva: a robust cluster validity. In: 2th. WSEAS international conference on scientific computation and soft computing, Crete, Greece, pp 209–213
187. Rijsbergen V (1979) Information retrieval. Butterworths, London, p 1979
188. Rokach L (2005) "Clustering methods", data mining and knowledge discovery handbook. Springer, Berlin, pp 331–352
189. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
190. Sabau AS (2012) Survey of clustering based financial fraud detection research. Inform Econ 16(1):110
191. Saemi B, Hosseinabadi AA, Kardgar M, Balas VE (2018) Nature inspired partitioning clustering algorithms: a review and analysis. In: Advances in intelligent systems and computing, pp 96–116
192. Saha I, Maulik U, Bandyopadhyay S (2009) A new differential evolution based fuzzy clustering for automatic cluster evolution. In: 2009 IEEE international advance computing conference. IEEE, pp 706–711
193. Sahoo AJ, Kumar Y (2014) Modified teacher learning based optimization method for data clustering. In: Advances in signal

processing and intelligent recognition systems. Springer, Cham, pp 429–437
194. Saitta S, Raphael B, Smith I (2007) Abounded index for cluster validity. In: Perner P (ed) Machine learning and data mining in pattern recognition, vol 4571. Lecture notes in computer science. Springer, Berlin, pp 174–187
195. Salcedo-Sanz S, Carro-Calvo L, Portilla-Figueras A, Cuadra L, Camacho D (2013) Fuzzy clustering with grouping genetic algorithms. In: International conference on intelligent data engineering and automated learning. Springer, Berlin, Heidelberg, pp 334–341
196. Sarsoh JT, Hashim KM, Miften FS (2009) Comparisons between automatic and non-automatic clustering algorithms. J Coll Educ Pure Sci 4(1):221–227
197. Satapathy SC, Naik A (2011) Data clustering based on teaching-learning-based optimization. In: International conference on swarm, evolutionary, and memetic computing. Springer, Berlin, Heidelberg, pp 148–156
198. Sathappan S, Sridhar S, Tomar DC (2017) A literature study on traditional clustering algorithms for uncertain data. J Adv Math Comput Sci 21:1–21
199. Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A et al (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681
200. Senthilnath J, Das V, Omkar SN, Mani V (2013) Clustering using levy flight cuckoo search. In: Proceedings of seventh international conference on bio-inspired computing: theories and applications (BIC-TA 2012). Springer, India, pp 65–75
201. Senthilnath J, Omkar SN, Mani V (2011) Clustering using firefly algorithm: performance study. Swarm Evol Comput 1(3):164–171
202. Sharma M, Chhabra JK (2019) Sustainable automatic data clustering using hybrid PSO algorithm with mutation. Sustain Comput Inform Syst 23:144–157
203. Sharma SC (1996) Applied multivariate techniques. Wiley, New York
204. Shehab M, Khader AT, Al-Betar MA (2017) A survey on applications and variants of the cuckoo search algorithm. Appl Soft Comput 61:1041–1059
205. Shirkhorshidi AS, Aghabozorgi S, Wah TY, Herawan T (2014) Big data clustering: a review. Lecture notes in computer science. Springer, Cham, pp 707–720
206. Shukla AK, Banshal SK, Seth T, Basu A, John R, Muhuri PK (2020) A bibliometric overview of the field of type-2 fuzzy sets and systems [discussion forum]. IEEE Comput Intell Mag 15(1):89–98
207. Shukla AK, Sharma R, Muhuri PK (2018) A review of the scopes and challenges of the modern real-time operating systems. Int J Embed Real-Time Commun Syst (IJERTCS) 9(1):66–82
208. Shukla N, Merigó JM, Lammers T, Miranda L (2020) Half a century of computer methods and programs in biomedicine: a bibliometric analysis from 1970 to 2017. Comput Methods Programs Biomed 183:105075
209. Silva Filho TM, Pimentel BA, Souza RM, Oliveira AL (2015) Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. Expert Syst Appl 42(17–18):6315–6328
210. Singh J, Kumar R, Mishra AK (2015) Clustering algorithms for wireless sensor networks: a review. In: 2015 2nd international conference on computing for sustainable global development (INDIACom)
211. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
212. Strehl A, Ghosh J (2000) Clustering guidance and quality evaluation using relationship-based visualization. In: Intelligent engineering systems through artificial neural networks, St. Louis, Missouri, USA, pp 483–488
213. Sundararajan S, Karthikeyan S (2014) An efficient hybrid approach for data clustering using dynamic K-means algorithm and firefly algorithm. J Eng Appl Sci 9(8):1348–1353
214. Suresh K, Kundu D, Ghosh S, Das S, Abraham A (2009) Automatic clustering with multi-objective differential evolution algorithms. In 2009 IEEE congress on evolutionary computation. IEEE, pp 2590–2597
215. Taghva K, Sharma M (2007) Comparison of automatic clustering and manual categorization of documents. In: Akhgar B (ed) ICCS
216. Tan PN, Steinbach M, Kumar V (2013) Data mining cluster analysis: basic concepts and algorithms. In: Introduction to data mining, pp 487–533
217. Tang WH, Wu QH (2011) Evolutionary computation. In: Tang WH, Wu QH (eds) Condition monitoring and assessment of power transformers using computational intelligence. Springer, London, pp 15–36
218. Theodoridis S, Koutroubas K (1999) Pattern recognition. Academic Press, Cambridge
219. Thomas MC, Romagnoli J (2016) Extracting knowledge from historical databases for process monitoring using feature extraction and data clustering. In: Proceedings of the 26th European symposium on computer aided process engineering—ESCAPE vol 26, pp 861–864
220. Tran DC, Wu Z, Wang Z, Deng C (2015) A novel hybrid data clustering algorithm based on artificial bee colony algorithm and k-means. Chin J Electron 24(4):694–701
221. Tsai CW, Huang KW, Yang CS, Chiang MC (2015) A fast particle swarm optimization for clustering. Soft Comput 19(2):321–338
222. Tsay RS (2005) Analysis of financial time series. Wiley, New York
223. Tseng LY, Yang SB (2001) A genetic approach to the automatic clustering problem. Pattern Recognit 34(2):415–424
224. Van der Merwe DW, Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: The 2003 congress on evolutionary computation, 2003. CEC'03, vol 1. IEEE, pp 215–220
225. Van Eck NJ, Waltman L (2010) Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics 84(2):523–538
226. Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416
227. Vo-Van T, Nguyen-Hai A, Tat-Hong MV, Nguyen-Trang T (2020) A new clustering algorithm and its application in assessing the quality of underground water. Sci Program 2020:6458576. https://doi.org/10.1155/2020/6458576
228. Wang R, Zhou Y, Qiao S, Huang K (2016) Flower pollination algorithm with bee pollinator for cluster analysis. Inf Process Lett 116(1):1–14
229. Wang S, Wu Y (2010) Clustering analysis based on chaos genetic algorithm. In: 2010 Chinese control and decision conference. IEEE, pp 16–19
230. Wei G, Liu H, Xie M (2009) Clustering large spatial data with local-density and its application. Inf Technol J 8(4):476–485
231. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Philip SY, Zhou ZH (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37
232. Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. Ann Data Sci 2(2):165–193
233. Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678

234. Yang XS (2010) Firefly algorithm, nature inspired metaheuristic algorithms, 2010. LUniversityer Press, Frome
235. Younsi R, Wang W (2004) A new artificial immune system algorithm for clustering. In: International conference on intelligent data engineering and automated learning. Springer, Berlin, Heidelberg, pp 58–64
236. Yu D, Xu Z, Kao Y, Lin CT (2017) The structure and citation landscape of IEEE Transactions on Fuzzy Systems (1994–2015). IEEE Trans Fuzzy Syst 26(2):430–442
237. Yu JY, Chong PHJ (2005) A survey of clustering schemes for mobile ad hoc networks. IEEE Commun Surv Tutor 7(1):32–48
238. Žalik KR (2008) An efficient k′-means clustering algorithm. Pattern Recognit Lett 29(9):1385–1391
239. Žalik KR, Žalik B (2011) Validity index for clusters of different sizes and densities. Pattern Recognit Lett 32(2):221–234
240. Zanjireh MM, Shahrabi A, Larijani H (2013) ANCH: a new clustering algorithm for wireless sensor networks
241. Zhao M, Tang H, Guo J, Sun Y (2014) Data clustering using particle swarm optimization. In: Future information technology. Springer, Berlin, Heidelberg, pp 607–612
242. Zhao XQ, Zhou JH (2015) Improved kernel possibilistic fuzzy clustering algorithm based on invasive weed optimization. J Shanghai Jiaotong Univ (Sci) 20(2):164–170
243. Zhong Y, Zhang S, Zhang L (2013) Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery. IEEE J Sel Top Appl Earth Obs Remote Sens 6(5):2290–2301
244. Zhou Y, Wu H, Luo Q, Abdel-Baset M (2018) Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowl-Based Syst 163:546–557
245. Zhou Y, Wu H, Luo Q, Abdel-Baset M (2019) Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowl-Based Syst 163:546–557
246. Zou F, Chen D, Xu Q (2019) A survey of teaching–learning-based optimization. Neurocomputing 335:366–383