Received May 11, 2021, accepted May 22, 2021, date of publication May 26, 2021, date of current version June 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3084057

Scalable Clustering Algorithms for Big Data: A Review

MAHMOUD A. MAHDI, KHALID M. HOSNY, AND IBRAHIM ELHENAWY
Faculty of Computers and Information, Zagazig University, Zagazig 44519, Egypt
Corresponding author: Khalid M. Hosny (k_hosny@yahoo.com)

ABSTRACT Clustering algorithms have become one of the most critical research areas in multiple domains, especially data mining. However, with the massive growth of big data applications in the cloud world, these applications face many challenges and difficulties. Since Big Data refers to an enormous amount of data, most traditional clustering algorithms come with high computational costs. Hence, the research question is how to handle this volume of data and get accurate results at a critical time. Despite ongoing research work to develop different algorithms to facilitate complex clustering processes, there are still many difficulties that arise while dealing with a large volume of data. In this paper, we review the most relevant clustering algorithms in a categorized manner, provide a comparison of clustering methods for large-scale data and explain the overall challenges based on clustering type. The key idea of the paper is to highlight the main advantages and disadvantages of clustering algorithms for dealing with big data in a scalable approach behind the different other features.

INDEX TERMS Clustering, unsupervised learning, traditional clustering, parallel clustering, stream clustering, high dimensional data, big data, large-scale.

I. INTRODUCTION

Clustering, or cluster analysis or data segmentation [1], commonly defined as the grouping of similar objects into classes called clusters [2] or defined more specifically as an unsupervised learning approach to classification of patterns into groups (clusters) based upon similarity, where a pattern is a representation of features or attributes of an object [3]. Clustering methods are unsupervised because we do not know the classification parameters, characteristics of data, or even the number of cluster groups versus the classification methods. Therefore, the clustering-based techniques try to estimate and learn these parameters from the given data. Usually, there is two way to perform this process: an offline method for a given saved batch of data, and an online for coming data sequentially. The offline methods resulting in better accuracy but not valid for very massive or real-time data.

A. BACKGROUND

Today, clustering is considered one of the vital data-mining tools for analyzing data. There are large standard application fields in which clustering is one of its tools, such as the following:

- Social network analysis
- Collaborative filtering
- Data summarizing
- Multimedia data analysis
- Customer segmentation
- Biological data analysis

These different applications produce different ***data types*** with other characteristics. The most common types of these data are numerical data, categorical data, text data, multimedia data, time-series data, discrete sequences, network data, and uncertain data (Figure 1). Each of these data types requires a special pre-processing or processing before applying any data mining technique.

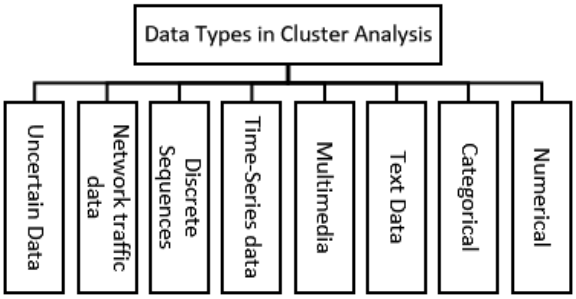

FIGURE 1. Different data types in cluster analysis.

With increase data in size and speed, handling of big data has become inevitable. There are many definitions of

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

Big Data, and one of them is the amount of data just beyond technology's capability to store, manage, and process efficiently [4]. Big data are seen as vast, complex, and growing from multiple or autonomous resources. As a result of the fast improvement of communication, data storage methods, and the high ability to collect data, big data became rapidly growing in all fields and domains such as science, engineering, physical, biological, and biomedical sciences [5]. Also, many new applications can quickly generate vast amounts of data during a short time; for example, social networks provide incredible opportunities for social connections and an enormous volume of data [6].

In the same direction, ***Data streams*** refer to a massive amount of data generated at very high speed, such as network traffic, web click streams, and sensor networks [7]. Hence, Datastream mining has become a hot research topic because of introducing advanced technologies and applications that regularly generate data streams.

Although big data and streaming data add big challenges, the data type significantly impacts the clustering problem. As a result, the kind of data plays the primary role in choosing techniques used for the clustering analysis. Accordingly, there are wide ranges of clustering techniques' models that have different clustering methodology. Figure 2 shows the general clustering techniques classified based on its methods from various studies. Because of the significant growth of data occupies many future challenges and often require specialized techniques [8].

The algorithm **scalability** is the ability of the algorithm to handle a growing amount of input by adding additional resources to the system [9]. In this case, the system can be scaled up or down based on the work size. Nowadays, the scaling the resources has become an essential factor as a result of the cost of adding resources to the system, which is why research has tended towards developing ways to deal with scalable systems, especially in cases of big data and what means real-time versus cost. According to scalability definition, we classify the techniques in this review into two types: traditional and scalable techniques. The traditional techniques consist of clustering algorithms without regard to the system's scalability. In contrast, the scalable techniques consist of the clustering algorithms utilizing the system's scalability.

Due to the limitations of the traditional clustering algorithms either in output speed or in processing data, researches investigated in two directions to face these challenges. The first direction is by trying to improve traditional algorithms to working with large data size and the other orientation by proposed new methodology based on the benefit of new technology such as parallel computing, cloud computing, and map-reduce.

B. CURRENT ISSUES

The significant challenges for data miners and data analyzers come from using the best method to extract useful information from the large dimensional and increasing dataset [10]. Nowadays, Big Data is an exciting area for scientific research. It is becoming a common data source for many business applications, which require a range of data mining operations [11]. However, there are difficulties in applying data mining techniques to big-data because of the new challenges with big data [12]. The scalability, complexity, and the presence of mixed data are the main challenges of big data analysis, and clustering appeared due to [13]. So, parallelism is introduced by many clustering algorithms because it is useful for applying the 'divide and conquer' strategy in algorithms to reduce the time complexity related to big data [14]. It identified that the main challenges while clustering extensive data fall in the following points:

1) Clusters usually have non-uniform shapes,
2) Lack of knowledge and ability to either determine the number of input parameters or choose the better values of it, and
3) Scalability and the incredible size of extensive data.

On the other hand, one of the most critical aspects of the data mining problem is how similarity is defined and measured. The similarity measures affect the clustering and classification of information concerning the type of data. Clustering techniques for categorical data are very different from those for numerical data in terms of the definition of the similarity measure. It is also rare to find the boundaries of clusters and avoid their overlap, which adds a constraint to researchers when choosing the optimal similarity measure is applied to a wide range of data types.

Focus on scalable clustering techniques (Figure 3); the algorithms classified according to how to deal with the data either as a batch or real-time streaming. In streaming clustering techniques, most of the algorithms were coming from the traditional methods with some modification to working with the stream data. Unlike big data techniques, most approaches are based on new algorithms. They classified into two strategies based on the number of machine algorithm can dealing during the process (single or multiple machines).

C. CONTRIBUTIONS

In this paper, we focus on the issues and challenges of scalable clustering techniques implemented to face big data or streaming data. The basic contribution of this paper as follows:

- Present a general overview of different clustering algorithm based on the clustering techniques.
- Reviews the different clustering algorithms and main features according to general features and categorizing them based on new scalability methods.
- Present features comparison between studies algorithms based on clustering techniques.

II. TRADITIONAL CLUSTERING TECHNIQUES

In this part, we review the traditional clustering techniques. Algorithms that are not dealing with system scalability or data size as a metric in clustering processing will be

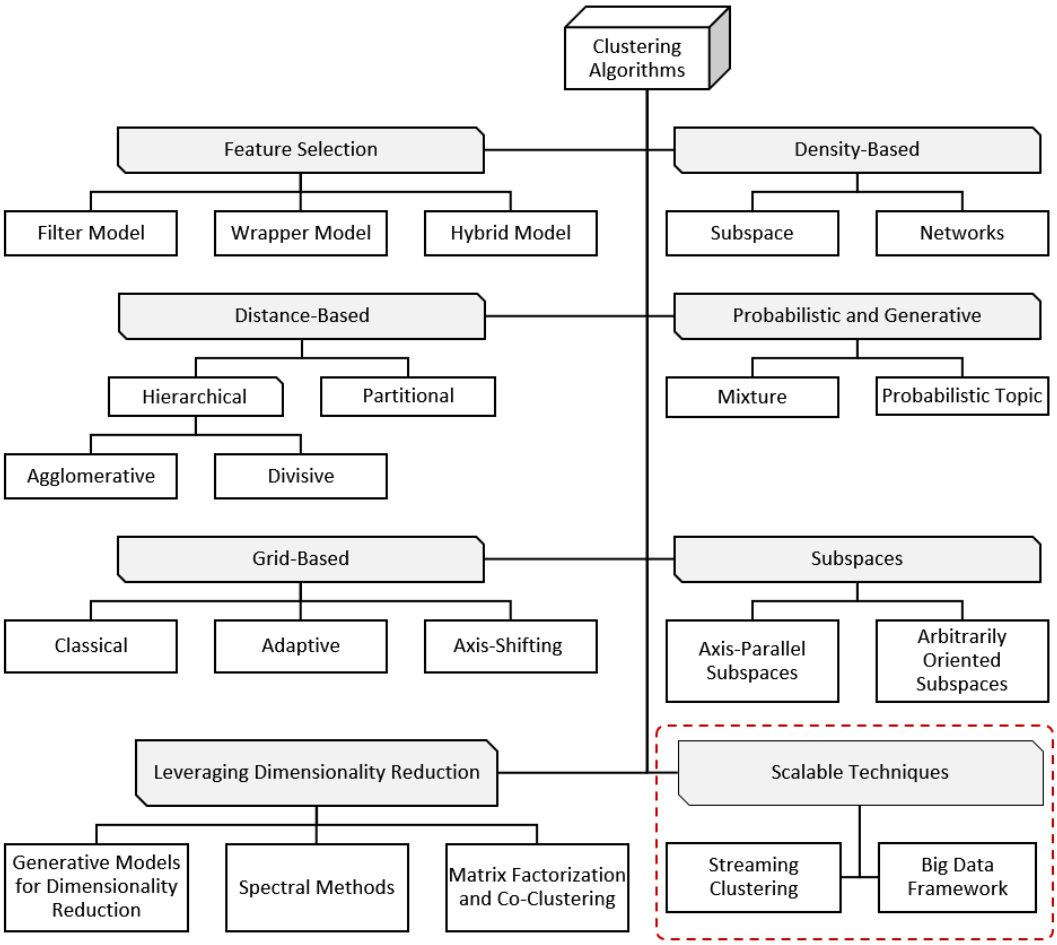

FIGURE 2. Clustering algorithms/techniques model-based classification.

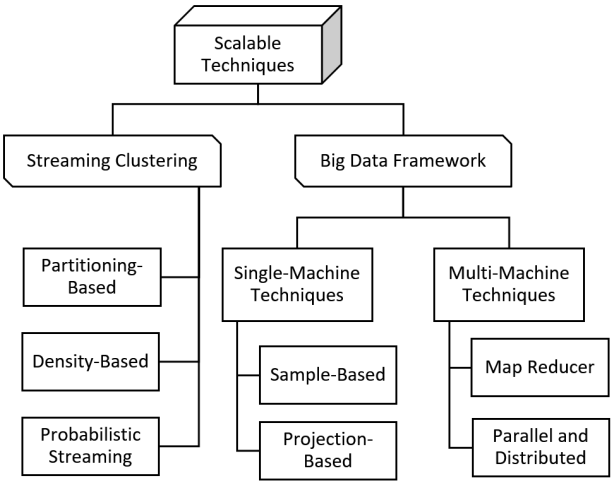

FIGURE 3. Scalable clustering techniques.

covered under the traditional term. The traditional method does not depend on processing the volume of data in terms of distribution methods on devices or division to deal with extensive data.

A. HIERARCHICAL CLUSTERING

In a hierarchical clustering algorithm, cluster data grouped with a sequence of nested partitions, either from separate clusters to a cluster, including all individuals or vice versa. The former is known as agglomerative, and the latter is called divisive. Agglomerative methods: use the 'bottom-up' approach; they begin with each object as a separate cluster and merge them into successively larger clusters. Divisive methods: on the other hand, use the 'top-down' approach; they begin with the whole set of objects in one cluster and proceed to divide it into successively smaller clusters. Figure 4 demonstrates the difference between the two approaches in process direction. In practice, agglomerative techniques are commonly used, while divisive techniques are limited due to their prohibitive computational burden. The output of hierarchical clustering is usually represented as a dendrogram or voroni diagram in visualizing big data (such as figure 5), which clearly describes the proximity among data objects and their clusters and provides good visualization. Although the classical hierarchical clustering methods are conceptually easy to understand, they suffer the disadvantages of high computational complexity.

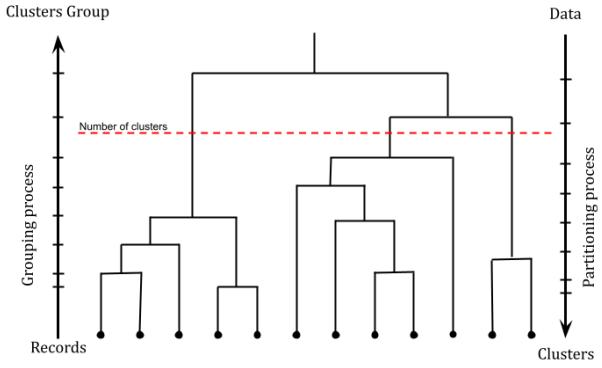

FIGURE 4. Hierarchical clustering technique

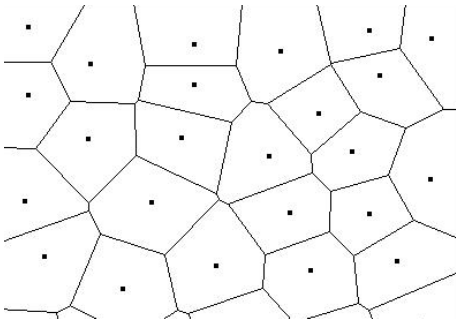

FIGURE 5. Voronoi diagram

This high computational burden limits its application in large-scale data sets [15]. Many algorithms fall in this category such as Chameleon [16], ROCK [17], LIMBO [18], F-Tree [19], MTMDP [20], and MGR [21].

B. PARTITION-BASED CLUSTERING

In the partitioning-based algorithms, all clusters data is recursively divided into some partitions until the partition criterion reaches a specified value, and here each partition represents a cluster. K-means [22], and K-medoids [23] are most famous algorithms based on partitioning. K-means iterative update the centre of the cluster until coverage data. Some versions of K-means has been proposed in the way to improve time complexity as [24]. Other algorithm's fall in this category, such as PAM [25], CLARA [26], CLARANS [27], COOLCAT [28] and Squeezer [29].

C. DENSITY-BASED CLUSTERING

The density clustering aims to discover the shapes of the clusters. In this type, the data are numerical so that they can be grouped based on dimensional distances. Initially, data divided into three types of points: core, boundary, and noise points. The point considered a core point if it has a least $\boldsymbol{m}$ points within distance $\boldsymbol{n}$, the point considered a border if it has at least on core within range; otherwise, the point marked as noise. The algorithms work by grouping these points to form a density of the clusters. Algorithms fall in this category such as CACTUS [30], CLOPE [31], DBDC [32], and EBK-modes [33].

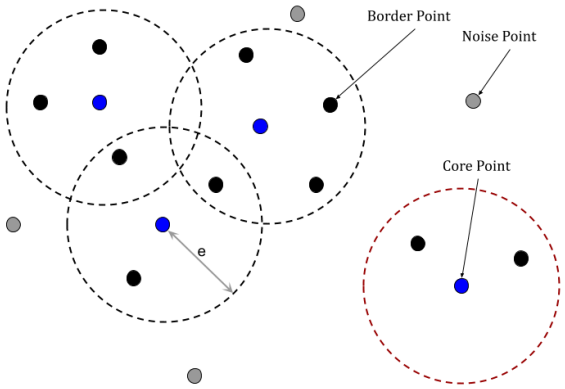

FIGURE 6. Density-based clustering technique

D. PROBABILISTIC AND GENERATIVE CLUSTERING

In the model-based algorithms, data is clustering based on various strategies such as statistical methods and conceptual methods. There are two common ways of model-based algorithms: the neural network approach, and the analytical approach. The neural network approaches are supervised techniques; However, Kohonen's SOM is the model used for clustering [34]. Algorithms fall in this category, such as SVC [35] and Ensemble [36].

E. GRID-BASED CLUSTERING

Grid-based clustering algorithms have shown great interest in their advantages of discovering clusters with different shapes and sizes. Mainly, There are two methods in this type: Fix-up and the adaptive grid partition method. The idea of a fix-up grid partition method is to divide each dimension of the data space into equal lengths, and then they crossed rectangular cells of the same size. Since the points in the same network belong to a group, they are treated as a single object. All groupings run on these grid cells. While the idea of the adaptive grid partition method is to divide data space into non-crossed grid cells of different sizes according to the data distribution feature, the total number of grid cells is significantly reduced compared to those fix-up partition methods. Still, the determination of spitting points required massive computation power [37]. Some algorithms, such as CLIQUE [38], STING [39], WaveCluster [40], and DENCLUE [41], used the fix-up grid method, while some other used the adaptive grid partition method such as OptiGrid [42] and MAFIA [43].

III. SCALABLE CLUSTERING TECHNIQUES

We explore some of scalable clustering techniques. Usually, two different methods developed to handle big data; first, techniques focus on reducing the size of the data either vertically or horizontally (Figure 7), while the other methods focus on speeding the execution depend on multi-physical processors. With this technique, big data can be cut into smaller pieces to processing on different devices simultaneously. The multi-machine clustering algorithm classified into

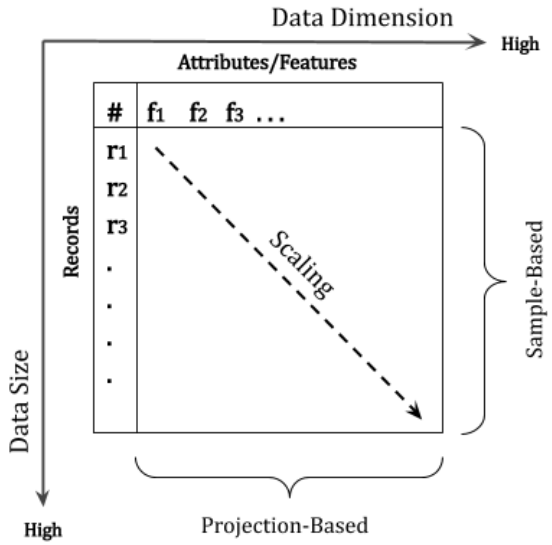

FIGURE 7. Reduction methods vs data scale

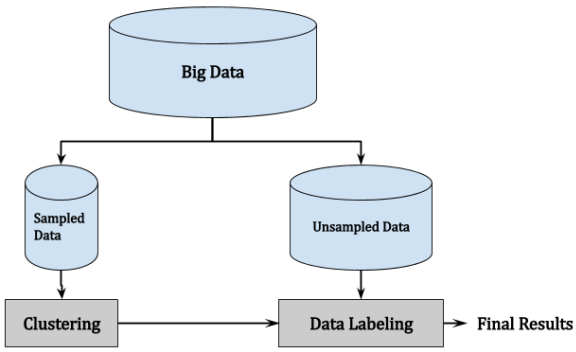

FIGURE 8. Sampling-based clustering technique

two categories according to the technique used to run multiple processes, either map-reduce or parallel, and distribution. Commonly, these methods solve the clustering time challenge of big data. The [44] presented a more details survey on parallel clustering algorithms according to the platform of big data. In common, the scalable clustering algorithm is designed to fit the specific scaled platform. While the [45] presented a more details survey on stream clustering algorithms compared with traditional algorithms. In this section, we review scaling algorithms based on techniques rather than the architecture of the platform across streaming and non-streaming algorithms.

A. SAMPLING-BASED

This method is one of the first ways to try to overcome data volume and operation speed problems; the primary goal is based on clustering data samples and rather than clustering the entire data set (Figure 8). After processing, the results of the clustering are generalized to the whole data set [12]. These methods are one of the ideas that contributed to accelerating techniques by minimizing the data size and thus

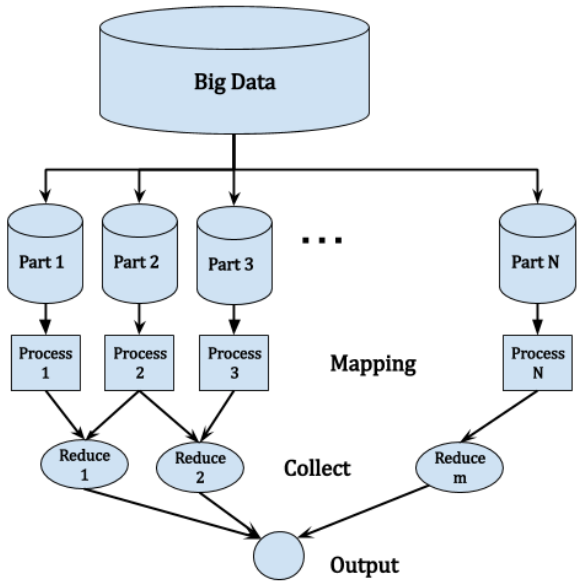

FIGURE 9. Map-reduced based clustering technique

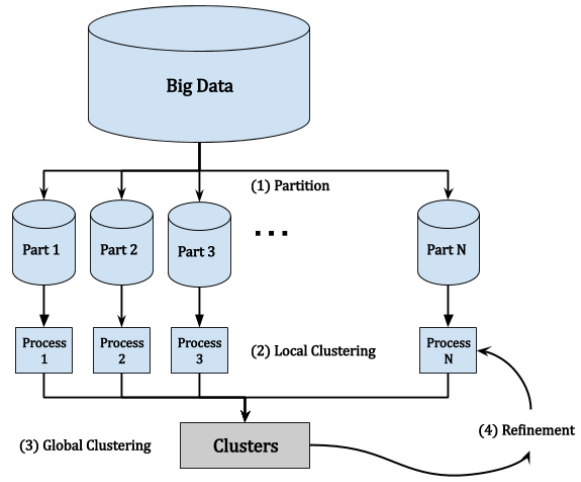

FIGURE 10. Parallel clustering techniques

time and complexity. In contrast, these techniques added time and complexity in pre-processing data required for sampling operations. Moreover, the clustering subset of data not give the same accuracy compared to the entire data. Algorithms fall in this category, such as BIRCH [46], CURE [47], and CLARANS [27].

B. REDUCTION AND PROJECTION-BASED

The high dimensionality of data adds additional challenges to most clustering algorithms, such as the existence of noises features or sparse data [48]. While sampling-based techniques reduce the data size vertically, they are not considered the best solution to a high-dimensional data set in cases. Similarly, the projection-based methods try to reduce the data size horizontally. The current approaches use procedures like feature selection, feature extraction, approximation, and random

TABLE 1. Advantages and limitations of based on general clustering Type.

Type	Conclusion
Density	**Advantages:** • Handling Noise • Good for Streaming data • Good for Spatial data • Handling Arbitrary-shape clusters **Limitations:** • High Time Complexity • High Space Complexity • Depend on data order • Required a large number of parameters
Grid	**Advantages:** • High Scalable • Parallelism • Handling Arbitrary-shape clusters • Handling Large data size • Handling Noise **Limitations:** • Predefned grid size • Hard to handle high dimensional data
Hierarchical	**Advantages:** • Easy to implement **Limitations:** • High Time Complexity • Hard to measure distance on the different data type • Predefined number of clusters • Noise sensitivity • Termination condition has to be specified
Model	**Advantages:** • High Time Complexity • Handling Noise **Limitations:** • Hard to mix different data type • Quality depends on the statistical model • The number of clusters must be predefined • Low Scalability
Partitioning	**Advantages:** • High Scalability • Easy to implement on the various data type • Parallelism **Limitations:** • High Time Complexity • Predefined number of clusters • Wrong results for complex cluster type

TABLE 2. Advantages and limitations of based on general scalable techniques.

Type	Conclusion
Sampling based	**Advantages:** • Dataset Reduction • Faster **Limitations:** • Hard to handle high dimensional data • Sampling-based on data structure • Data analysis before the algorithm
Projection based	**Advantages:** • Handling high dimensional data • Handling Noise **Limitations:** • Results based on a feature selection algorithm
Parallel Based	**Advantages:** • Parallel processing • High Scalable **Limitations:** • Hard to implement
Map-Reduced Based	**Advantages:** • High Scalable **Limitations:** • Implementation required special resources • Hard to implement • Reduction based on the operation

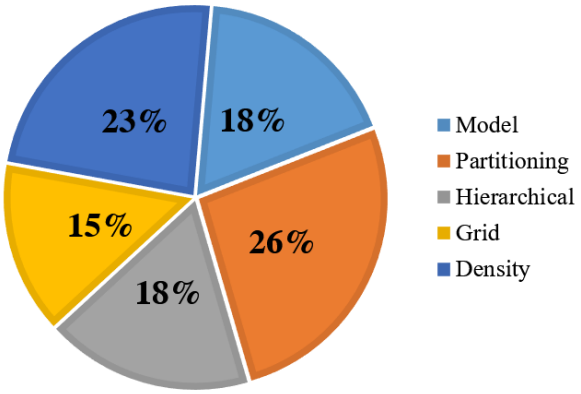

FIGURE 11. Percentage of studied algorithms grouped by clustering type

reduction. These techniques are also required pre-processing data as sampling-based. The feature selection lowers the dimension space by filtering the data attributes based on data dependency, While the feature extraction is constructing new advantage features. Algorithms fall in this category, such as Ensemble [48], Colibri [49], and RP [50].

C. SUBSPACE CLUSTERING

Subspace clustering is a technique that searches for clusters in different subspaces. The basic idea is to discover clusters using a subset of dimensions. Generally, two types of subspace clustering based on the search strategy; bottom-up and top-down. The bottom-up start by finding clustering in lower dimension and iterative merging them to process higher dimension spaces. Top-down start by find clusters in full dimension then evaluates the subspace of each cluster. Generally, Subspace clustering solves the problem of the high dimensional dataset faces most grid-based approaches. Algorithms fall in this category, such as SSSC [51], TNNLS [52] and StructAE [53]

D. MAP-REDUCED BASED

As a single processor with one memory cannot handle extensive data at an adequate speed, it emphasizes the need for algorithms that run on multiple devices. The Map-Reduce framework (as displayed in Figure 9) dropped many concerns necessary to run algorithms on multiple devices such as network connection, data distribution, and load balancing. These advantages allow many researchers to easily applied and improved their algorithms in parallel processing systems. There is a set of proposed research that re-apply or re-implement a better clustering technique using map-reducer, such as the research in [54]. They presented an

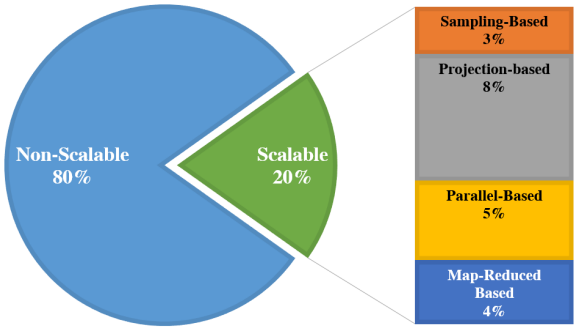

FIGURE 12. Scalable vs. non-scalable techniques of studied algorithms

integrated approach for CURE clustering using the map-reduce method. Other algorithms fall in this category such as DisCo [55], PKMeans [56], BoW [57], wkPCM [58], and KIM [59].

E. PARALLEL AND DISTRIBUTED-BASED

The data-driven path identification approach (DDPI) [60] is a concurrent algorithm representation of k-means with neural network batch training. Instead of implementing a distributed system, the author presented a data-parallel interface to permit the parallel implementation of the k-means algorithm using a neural network; but, it adds a supervised step. The three-parallelization steps of the algorithm are partitioning and distribution data then, computing using distributed data, and lastly, assembling local computational results. The concurrent structure provided a way to reduce the computational demand of neural techniques. The Density based distributed clustering (DBDC) [32] used the distributed techniques to speed up the clustering process on large scale data and based on density partitioning clustering technique (Figure 10). Distributed clustering in this algorithm is operating at two different levels (local and global). At the local level, DBDC uses an independent algorithm for clustering, which carried out the process on partitioned data. On the global level, it uses a density algorithm clustering called DBSCAN to generate the results in the main site. The DBSCAN is used for all kinds of metric data spaces only. While meeting real-time constraints in clustering data streams using parallelization in the cloud, the Cloud DIKW [61] introduced a cloud framework using integrated parallel batch and stream processing.

Moreover, clustering social media data streams are used as a tested application domain. Recently, the [62] presented a parallel implementation of a multi-objective feature selection. The algorithm makes possible use of high dimensional datasets when there are much more features than data items. The feature selection is most helpful in classification.

IV. EVALUATIONS

A. RESULTS AND DISCUSSION

In this paper, we surveyed 101 algorithms in various clustering type, which includes most of the relevant clustering algorithms. We collected a large number of algorithms in different types to figure out the common themes of clustering type (such as scalability, complexity, data type) without focusing on the special considerations for each algorithm (such as application area, and data processing. The main criteria of paper selection was as the following: The main paper is proposed a clustering algorithm, and the algorithm is comparable with competitors in the same clustering area.

Figure 11 shows the percentage of each clustering method in the studied algorithms. The partition-based method had the most significant proportion, with 25% of all algorithms. 20% of the study algorithms applied an expandable method. Figure 12 shows the percentage of scalable processes that are not scalable. Projection-based algorithms were the highest research trend due to the ease with which this method applied against other scalable technologies.

Table 3 shows the clustering algorithm comparison classified according to their type and arranged chronologically within each class. The data are collected, directed from algorithms' reference, and validated from other surveys [14], [63]–[72]. We evaluated the techniques based on the following:

1) **Data Size**: which determines the ability of the algorithm to operate using very large or limited data size
2) **High Dimensions**: It determines the strength of the algorithm work with large dimensions of data.
3) **Streaming**: This defines the way the algorithm uses data, either Batch or stream way.
4) **Spatial Data Processing**: Algorithm's ability to dealing with complex and vital data types such as spatial.
5) **Different data types**: to determines whether the algorithm can work on more than one type of data at the same time.
6) **Handling Noise**: Algorithm's ability to overcome the outliers.
7) **Arbitrary Shape**: The cluster shape output.
8) **Scalable**: determine if the algorithm includes any of the four scalable methods, which are sampling, projection, parallel, and Map-Reduced.
9) **Complexity**: The time complexity of algorithms which classify the complexity of the algorithm into three classes; first, if the algorithm complexity is linear or semi-linear, then complexity class is low; if it is below quadratic, then complexity class is Middle, and if it is quadratic or above then the complexity is high.
10) **No of Parameters**: The number of Parameters the algorithm needs to operate
11) **Data Type**: The type of data that is suitable for a specific algorithm.

B. SUMMARY

Figure 13 presents a summary of the characteristic of the studied algorithms listed in Table 3. The percentage in table represents the average of feature in each algorithm's category. From this statistic, the following observed:

TABLE 3. Classification and characteristic of clustering algorithms.

Clustering Cat.	Algorithm	Publication Year	Large Dataset	High Dimensional	Stream Data	Spatial Data	Handling Noise	Arbitrary Shape	Sequence independence	Scalable Method	Scalability	Parameters	Time Complexity Class	Time Complexity	Data Type	Reference
Hierarchical	SNN	1985	○	○	○	○	○	●	○	○	Low	1	High	$\mathcal{O}(n^2)$	Discrete	[73]
	BIRCH	1996	●	○	○	○	●	○	●	○	High	2	Low	$\mathcal{O}(n)$	Numerical	[46]
	CURE	1998	●	●	○	○	●	●	●	●	High	2	High	$\mathcal{O}(n^2 \log n)$	Numerical	[47]
	CACTUS	1999	○	○	○	○	○	○	●	○	Low	2	Low	$\mathcal{O}(t\,n)$	Discrete	[30]
	Chameleon	1999	○	○	○	○	●	●	●	○	High	3	High	$\mathcal{O}(n^2)$	Discrete	[16]
	CLICK	2000	●	○	○	○	○	○	●	○	High	4	Middle	$\mathcal{O}((d\,n)^{2/3}\,k)$	Discrete	[74]
	ROCK	2000	○	●	○	○	○	●	●	○	Middle	2	High	$\mathcal{O}(n^2 \log n)$	Discrete	[17]
	Spectral	2002	○	○	○	○	○	○	●	○	Low	1	Low	$\mathcal{O}(n)$	Numerical	[75]
	LIMBO	2004	●	○	○	○	●	●	●	○	Middle	1	Middle	$\mathcal{O}(n \log n)$	Discrete	[18]
	CLICKS	2005	●	●	○	○	○	●	●	○	Middle	$*$		$(scaled)$	Discrete	[76]
	DisCo	2008	●	○	○	○	○	●	○	●	High	2	High	$\mathcal{O}(k\,d\,v/p)$	Numerical	[55]
	ECHIDNA	2008	●	○	○	○	○	●	○	○	Low	2		$*$	Mixed	[77]
	GRIDCLUST	2009	○	○	○	●	○	●	○	○	Low	2	High	$\mathcal{O}(n^2/m)$	Numerical	[78]
	ClusTree	2011	●	○	●	○	○	●	○	○	Middle	3	High	$\mathcal{O}(n^3)$	Discrete	[79]
	FTREE	2012	●	●	○	○	○	●	●	○	High	1	High	$\mathcal{O}(n^2\,d)$	Discrete	[19]
	MGR	2014	●	○	○	●	○	●	○	○	Low	$*$	High	$\mathcal{O}(n\,k\,d^2\,k+d)$	Discrete	[21]
	MTMDP	2014	●	○	○	○	○	●	○	○	Low	$*$	High	$\mathcal{O}(n\,k\,d^2)$	Discrete	[20]
	Wards	2014	○	○	○	○	○	●	○	○	Low	0	High	$\mathcal{O}(n^2)$	Numerical	[80]
Density	DBSCAN	1996	●	○	○	●	○	●	●	○	Middle	2	Middle	$\mathcal{O}(n \log n)$	Numerical	[81]
	DBCLASD	1998	●	●	○	●	●	●	●	○	Middle	0	Middle	$\mathcal{O}(n \log n)$	Numerical	[82]
	DENCLUE	1998	●	●	○	●	●	●	○	○	Middle	1	High	$\mathcal{O}(n^2)$	Numerical	[41]
	GDBSCAN	1998	●	○	○	●	●	●	○	○	Middle	2	High	$\mathcal{O}(n^2)$	Numerical	[83]
	OPTICS	1999	●	○	○	●	●	●	●	○	Middle	2	High	$\mathcal{O}(n^2)$	Numerical	[84]
	STIRR	2000	●	○	○	○	○	○	○	○	Low	1	Low	$\mathcal{O}(n)$	Discrete	[85]
	CLOPE	2002	●	○	○	○	○	○	●	○	Low	1	Middle	$\mathcal{O}(k\,d\,n)$	Discrete	[31]
	Mean-shift	2002	○	○	○	●	●	●	●	○	Low	2	High	$(kernel)$	Discrete	[86]
	DBDC	2004	●	○	○	●	●	●	●	●	High	4	High	$\mathcal{O}(n^2)$	Numerical	[32]
	PreDeCon	2004	●	○	○	○	●	●	●	○	Low	4	High	$\mathcal{O}(d\,n^2)$	Numerical	[87]
	SUBCLU	2004	●	●	○	○	○	●	○	○	High	2		$*$	Numerical	[88]
	DenStream	2006	●	○	●	●	●	●	○	○	High	4	Low	$\mathcal{O}(n)$	Numerical	[89]
	D-Stream	2005	●	○	●	○	●	●	○	○	Middle	4	Middle	$\mathcal{O}(p\,n)$	Numerical	[90]
	HIERDENC	2007	○	○	○	●	●	●	●	○	Low	2	Low	$\mathcal{O}(n)$	Discrete	[91]
	MULIC	2007	○	○	○	○	●	●	○	○	Low	2	High	$\mathcal{O}(t\,n^2)$	Discrete	[92]
	C-DenStream	2009	●	○	●	○	●	●	○	○	Low	2	Low	$\mathcal{O}(n)$	Numerical	[93]
	FlockStream	2009	●	○	●	○	●	●	○	○	Low	4	Low	$\mathcal{O}(d)+O(p)$	Numerical	[94]
	HDenStream	2009	●	●	●	○	●	●	○	○	Middle	4	Low	$\mathcal{O}(d\,p)$	Mixed	[95]
	rDenStream	2009	●	○	●	○	●	●	○	○	Low	4	Low	$\mathcal{O}(n)$	Numerical	[96]
	SDStream	2009	●	○	●	○	●	●	○	○	Low	3	Low	$\mathcal{O}(d\,p)$	Numerical	[97]
	DENGRIS-Stream	2012	●	○	●	○	●	●	○	○	Middle	2	Middle	$\mathcal{O}(p\,n)$	Numerical	[98]
	HDDStream	2012	●	●	●	○	●	●	○	●	High	4	Low	$\mathcal{O}(d\,p)$	Numerical	[99]
	PreDenConStream	2012	●	●	●	○	●	●	○	●	High	4	Low	$\mathcal{O}(d\,p)$	Numerical	[100]
	MuDiStream	2016	●	●	●	○	●	●	○	○	Middle	1	Low	$\mathcal{O}(s+m+\log\log n+\log n)$	Mixed	[101]
Grid	BANG	1997	●	●	○	●	●	○	●	○	Low	2	Low	$\mathcal{O}(n)$	Numerical	[102]
	PROCLUS	1997	●	●	○	○	●	○	○	●	Low	2	Low	$\mathcal{O}(n)$	Mixed	[103]
	STING	1997	●	○	○	●	●	●	●	○	Low	1	Low	$\mathcal{O}(n)$	Numerical	[39]
	CLIQUE	1998	●	●	○	●	●	●	●	○	High	2	Low	$\mathcal{O}(n+k^2)$	Numerical	[104]
	WaveCluster	1998	●	○	○	●	●	●	●	○	High	2	Low	$\mathcal{O}(n)$	Numerical	[40]
	ENCLUS	1999	●	○	○	○	●	○	○	○	Low	2	High	$\mathcal{O}(n\,d+m^d)$	Numerical	[105]
	MAFIA	1999	●	○	○	○	●	●	●	●	High	2	Middle	$\mathcal{O}(d\,n)$	Numerical	[106]
	OPTIGRID	1999	●	●	○	○	●	●	○	○	High	3	Middle	$\mathcal{O}(d\,n\,\log n)$	Mixed	[42]
	FC	2000	●	●	○	○	●	●	○	○	High	2	Low	$\mathcal{O}(n)$	Numerical	[107]
	ORCLUS	2000	●	●	○	●	●	●	●	○	Low	2	Middle	$\mathcal{O}(d^3)$	Mixed	[108]
	DUCStream	2005	○	○	●	○	●	○	○	○	Low	1	Low	$\mathcal{O}(n)$	Mixed	[109]
	Colibri	2008	●	○	○	○	○	○	○	●	Low	$*$	Middle	$\mathcal{O}(d\,n+d^2)$	Numerical	[49]
	DDStream	2008	●	○	●	○	●	●	●	○	Middle	4	Middle	$\mathcal{O}(p^2)$	Numerical	[110]
	MR-Stream	2009	●	●	●	○	●	●	●	○	Middle	4	Middle	$\mathcal{O}()$	Numerical	[111]
	PKS-Stream	2011	●	●	●	○	●	●	○	○	Middle	2	Low	$\mathcal{O}(d\,p)$	Numerical	[112]
	EM	1977	●	●	○	○	●	○	○	○	Low	3	Middle	$\mathcal{O}(d\,k\,n)$	Mixed	[113]
	FCM	1984	○	○	○	○	○	○	○	○	Middle	1	Low	$\mathcal{O}(n)$	Numerical	[114]
	CLASSIT	1989	○	○	○	○	○	●	○	○	Low	1	High	$\mathcal{O}(n^2)$	Numerical	[115]
	COBWEB	1989	○	○	○	○	●	●	○	○	Middle	1	High	$\mathcal{O}(d^2\,n)$	Discrete	[115]
	FCS	1992	○	○	○	○	○	●	○	○	Low	6	High	$(kernel)$	Numerical	[116]
	MM	1994	○	○	○	○	●	●	○	○	Low	$*$	Middle	$\mathcal{O}(v^2\,n)$	Numerical	[117]

TABLE 3. ***(Continued.)*** **Classification and characteristic of clustering algorithms.**

Clustering Cat.	Algorithm	Publication Year	Large Dataset	High Dimensional	Stream Data	Spatial Data	Handling Noise	Arbitrary Shape	Sequence independence	Scalable Method	Scalability	Parameters	Time Complexity Class	Time Complexity	Data Type	Reference
Model	AutoClass	1996	○	●	○	●	●	●	●	○	Low	1	High	$\mathcal{O}(d^2\ k\ t\ n)$	Mixed	[118]
	SOMs	1999	○	●	○	○	○	●	○	○	Low	3	High	$\mathcal{O}(n^2\ m)$	Numerical	[119]
	GMM	2000	○	○	○	○	●	●	○	○	Low	0	High	$\mathcal{O}(k\ t\ n^2)$	Numerical	[120]
	SVC	2001	○	○	○	○	●	●	○	○	Low	2	High	$(kernel)$	Numerical	[35]
	FREM	2002	●	●	○	●	●	○	○	○	Middle	5	Middle	$\mathcal{O}(d\ k\ t\ n)$	Mixed	[121]
	MMC	2005	○	○	○	○	●	●	○	○	Low	5	High	$(kernel)$	Numerical	[122]
	MKC	2009	○	○	○	○	○	●	○	○	Low	3	High	$(kernel)$	Numerical	[123]
	SWEM	2009	●	○	●	○	●	●	○	○	Low	4		$*$	Mixed	[124]
	SLINK	2011	●	○	○	○	○	●	○	○	Low	2	High	$\mathcal{O}(n^2)$	Mixed	[125]
	EBK-modes	2014	○	○	○	○	○	●	○	○	Low	2	High	$\mathcal{O}(d^2\ n)$	Discrete	[33]
	NSGA-FMC	2015	●	○	○	○	○	●	○	○	Low	$*$		$*$	Discrete	[126]
Partitioning	K-means	1967	○	○	○	○	○	○	○	○	High	2	Middle	$\mathcal{O}(n\ k\ d\ t)$	Numerical	[22]
	PAM	1987	○	○	○	○	●	○	○	○	Low	1	High	$\mathcal{O}(k^3\ n^2)$	Numerical	[25]
	CLARA	1990	●	○	○	○	●	●	●	●	High	1	Middle	$\mathcal{O}(k\ s^2 + k(n-k))$	Numerical	[26]
	K-Prototype	1997	○	○	○	○	○	○	○	○	Low	1	Low	$\mathcal{O}(n\ k\ t)$	Mixed	[127]
	K-modes	1997	●	●	○	○	○	●	○	○	High	1	Low	$\mathcal{O}(n\ k\ t)$	Discrete	[128]
	ParMETIS	1999	●	○	○	○	○	○	○	●	High	2	Low	$\mathcal{O}(p^2\ \log p)$	Numerical	[129]
	STREAM	2000	●	○	●	○	○	○	○	○	Middle	3	Low	$\mathcal{O}(n\ k\ d)$	Numerical	[130]
	CLARANS	2002	●	○	○	●	●	●	●	●	Middle	2	High	$\mathcal{O}(n^2\ k)$	Numerical	[27]
	COOLCAT	2002	●	○	○	○	○	○	○	○	Middle	1	High	$\mathcal{O}(n^2)$	Discrete	[28]
	Squeezer	2002	●	○	○	○	○	○	○	○	Low	1	Middle	$\mathcal{O}(n\ k\ d\ m)$	Discrete	[29]
	Stream LSerach	2002	●	○	●	○	○	○	○	○	Low	2	Middle	$\mathcal{O}(k\ i\ q\ n)$	Numerical	[131]
	CluStream	2003	●	○	●	○	○	○	○	●	High	9	Middle	$\mathcal{O}(n \log n)$	Numerical	[132]
	DDPI	2003	○	○	○	○	○	●	●	●	Low	2	High	$\mathcal{O}(k\ d\ s)$	Numerical	[60]
	Ensemble	2003	●	●	○	○	○	●	○	●	High	$*$	High	$\mathcal{O}(n^2\ k)$	Numerical	[48]
	MCLUST	2003	○	○	○	●	●	●	●	○	Low	2	High	$\mathcal{O}(d^2\ n)$	Numerical	[133]
	HPStream	2004	●	●	●	○	○	○	○	●	High	0	Low	$\mathcal{O}(n)$	Numerical	[134]
	K-ANMI	2008	○	○	○	○	○	●	○	○	Low	1	High	$\mathcal{O}(d^3\ k^2\ t\ n)$	Discrete	[135]
	SWClustering	2008	●	○	●	○	○	○	○	○	Middle	5	High	$\mathcal{O}(n\ k\ v)$	Numerical	[136]
	K-medoids	2009	○	●	○	○	○	○	○	○	Low	1	High	$\mathcal{O}(k\ (n-k)^2)$	Discrete	[23]
	PKMeans	2009	○	○	○	○	○	○	○	●	High	$*$		$(scaled)$	Numerical	[56]
	Ensemble	2010	●	○	○	○	○	○	○	○	Middle	$*$	High	$\mathcal{O}(d\ t\ n^2)$	Numerical	[36]
	G-ANMI	2010	●	○	○	○	○	●	○	○	Middle	$*$		$*$	Discrete	[137]
	RP	2010	●	○	○	○	○	○	○	●	High	2	Middle	$\mathcal{O}(n\ k\ d)$	Numerical	[50]
	BoW	2011	●	●	○	○	○	●	●	●	High	2		$(scaled)$	Numerical	[57]
	StreamKM++	2012	●	○	●	○	○	○	○	○	High	3	High	$\mathcal{O}(n\ k\ d^2)$	Numerical	[138]
	KIM	2014	●	○	○	○	○	●	○	●	High	$*$		$*$	Discrete	[59]
	Cloud-DIKW	2015	●	●	●	○	○	●	○	●	High	$*$	High	$\mathcal{O}(n^2\ d)$	Mixed	[61]

Complexity Variables: N = num objects, k = num clusters, m = num attributes, d = num features (unique attribute), s = sample size, t = num iterations, v = num vectors, e = num edges, p = num partitions, and q = num micro-clusters.
Feature: ●= included, ○= not included.

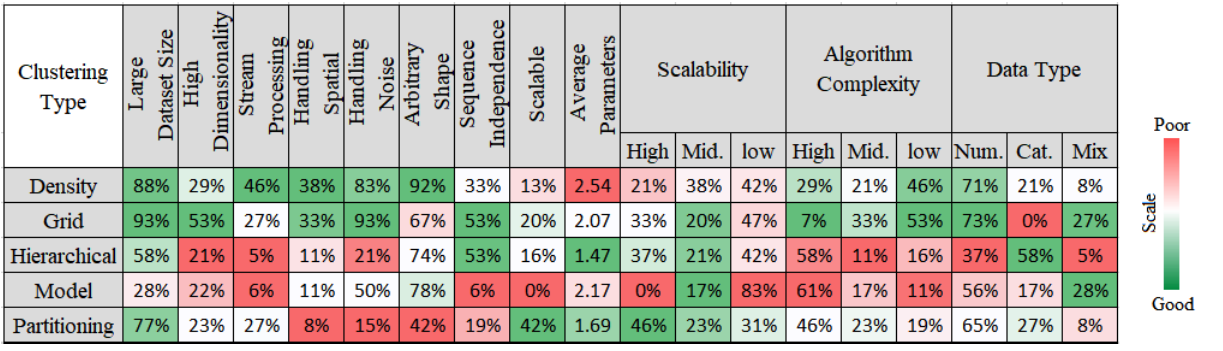

FIGURE 13. **Percentage of surveyed algorithms with corresponding features.**

- Often the grid-based algorithms can handle extensive, high-dimensional data and then density-based algorithms. To manage the stream data, we find most of the algorithms depend on density, grid-based, then partitioning-based, while the hierarchical and model-based among the least researched here.

- Most of the algorithms have high complexity implementation, except grid-based and density-based algorithms; most of their application is between medium and low complexity.
- Partitioning and model-based researches are usually based on numeric data types, and this limits its use in many fields. When datatype is a complex structure, the grid-based methods are more distinct.

In Table 1, we summary most of the advantages and limitations based on the clustering type. While in Table 2, summary most of the pros and restrictions based on a scalable model.

V. CONCLUSION

In this paper, we survey the literature to analyze and evaluate traditional and scalable clustering algorithms on big and large-scale datasets. First, we studying the different types of clustering and summarized the characteristics of the algorithms studies based on the algorithm's characteristics, scalability, handling noise, data type, and complexity. Second, we compare the general characteristics of the clustering types. Finally, we summarized the strengths and weaknesses of each variety of clustering methods and scalable techniques. The main results obtained from the selected studies are:

- The new scalable techniques are decreases due to the new direction of research toward implementing algorithms on the cloud-based infrastructure rather than developing new practical methods.
- The most commonly used method for scalable techniques in the studies was the partitioning-based algorithms then hierarchical-based algorithms.
- The density-based and grid-based algorithms found to be the most common techniques to handle large data size with noise in the literature. However, it observed that very few studies use mixed datatype datasets for evaluating the effectiveness of algorithm results.
- Grid-based algorithms were the fastest techniques to handle high dimensionality data.
- The most commonly used method for handling stream data in the studies was density-based algorithms.

We conclude that large size, high dimensional, speed, noise, shapes, and values integrated to form critical challenges in the data mining and analysis field. Hence, these challenges realized using scalable techniques such as cloud and parallelism. Since there is a wide range of scalable techniques used, the choice and implementation of the best model add additional challenges. The assessment study will give researchers the way to choose the appropriate techniques for choosing or developing a scalable clustering algorithm according to the considerations mentioned earlier.

FUTURE WORK

The large dimensional size of the data, the speed of data processing, data noise, data complexity, are integrated to form critical challenges in the field of data mining and analysis. Hence, these challenges realized using scalable techniques such as cloud and parallelism to handling data challenges. However, there is a wide range of scalable techniques across the current methods are used. The choice and implementation of the best model add additional challenges to researchers. Besides data problems designing new intelligent techniques for auto-adaptive based on data type, then auto partitioning data, distributing toward multiple methods, and collecting or sorting results at the same time is considered an open point to research.

REFERENCES

[1] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. Hoboken, NJ, USA: Wiley, 2009.
[2] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1, no. 10. New York, NY, USA: Springer, 2001.
[3] M. W. Berry and M. Browne, *Lecture Notes in Data Mining*. Singapore: World Scientific, 2006.
[4] S. H. Kaisler, F. J. Armour, A. Espinosa, and W. H. Money, "Big data and analytics challenges and issues," 2014.
[5] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[6] D. Agrawal, C. Budak, A. El Abbadi, T. Georgiou, and X. Yan, "Big data in online social networks: User interaction analysis to model user behavior in social networks," in *Databases in Networked Information Systems*, A. Madaan, S. Kikuchi, and S. Bhalla, Eds. Cham, Switzerland: Springer, 2014, pp. 1–16, doi: 10.1007/978-3-319-05693-7_1.
[7] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," *Knowl. Inf. Syst.*, vol. 45, no. 3, pp. 535–569, Dec. 2015.
[8] C. C. Aggarwal and C. K. Reddy, "Data clustering," in *Algorithms and Application*. Boca Raton, FL, USA: CRC Press, 2014.
[9] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proc. 2nd Int. Workshop Softw. Perform. (WOSP)*, 2000, pp. 195–203.
[10] D. C. Anastasiu, J. Iverson, S. Smith, and G. Karypis, "Big data frequent pattern mining," in *Frequent Pattern Mining*, C. C. Aggarwal and J. Han, Eds. Cham, Switzerland: Springer, 2014, pp. 225–259, doi: 10.1007/978-3-319-07821-2_10.
[11] K. Jaseena and J. M. David, "Issues, challenges, and solutions: Big data mining," in *Proc. NeTCoM, CSIT, GRAPH-HOC, SPTM*, 2014, pp. 131–140.
[12] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: A review," in *Computational Science and Its Applications—ICCSA 2014*. Cham, Switzerland: Springer, 2014, pp. 707–720, doi: 10.1007/978-3-319-09156-3_49.
[13] D. Jayalatchumy, P. Thambidurai, and A. A. Vasumathi, "Parallel processing of big data using power iteration clustering over MapReduce," in *Proc. World Congr. Comput. Commun. Technol.*, Feb. 2014, pp. 176–178.
[14] W. Kim, "Parallel clustering algorithms: Survey," *Parallel Algorithms, Spring*, vol. 34, p. 43, 2009.
[15] R. Xu and D. Wunsch, *Clustering*, vol. 10. Hoboken, NJ, USA: Wiley, 2008.
[16] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
[17] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *Inf. Syst.*, vol. 25, no. 5, pp. 345–366, Jul. 2000.
[18] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevcik, "LIMBO: Scalable clustering of categorical data," in *Advances in Database Technology—EDBT 2004*, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, Eds. Berlin, Germany: Springer, 2004, pp. 123–146, doi: 10.1007/978-3-540-24741-8_9.
[19] R. Bahgat, M. A. Mahdi, S. E. Abdel-Rahman, and I. A. Ismail, "Frequency tree for clustering categorical data with similarity measure based on items' weights," in *Qatar Foundation Annual Research Forum*, vol. 2012, no. 1. Doha, Qatar: Hamad bin Khalifa Univ. Press (HBKU Press), 2012.
[20] M. Li, S. Deng, L. Wang, S. Feng, and J. Fan, "Hierarchical clustering algorithm for categorical data using a probabilistic rough set model," *Knowl.-Based Syst.*, vol. 65, pp. 60–71, Jul. 2014.
[21] H. Qin, X. Ma, T. Herawan, and J. M. Zain, "MGR: An information theory based hierarchical divisive clustering algorithm for categorical data," *Knowl.-Based Syst.*, vol. 67, pp. 401–411, Sep. 2014.

[22] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, 1967, pp. 281–297.
[23] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.
[24] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognit.*, vol. 93, pp. 95–112, Sep. 2019.
[25] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Proc. Stat. Data Anal. Based L1 Norm Conf.*, Neuchâtel, Switzerland. Amsterdam, The Netherlands: Elsevier, 1987, pp. 405–416.
[26] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program pam)," *Finding Groups Data, Introduction Cluster Anal.*, vol. 344, pp. 68–125, Mar. 1990.
[27] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.
[28] D. Barbará, Y. Li, and J. Couto, "COOLCAT: An entropy-based algorithm for categorical clustering," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 582–589.
[29] Z. He, X. Xu, and S. Deng, "Squeezer: An efficient algorithm for clustering categorical data," *J. Comput. Sci. Technol.*, vol. 17, no. 5, pp. 611–624, Sep. 2002.
[30] V. Ganti, J. Gehrke, and R. Ramakrishnan, "CACTUS—Clustering categorical data using summaries," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1999, pp. 73–83.
[31] Y. Yang, X. Guan, and J. You, "CLOPE: A fast and effective clustering algorithm for transactional data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 682–687.
[32] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "DBDC: Density based distributed clustering," in *Advances in Database Technology—EDBT 2004*, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, Eds. Berlin, Germany: Springer, 2004, pp. 88–105, doi: 10.1007/978-3-540-24741-8_7.
[33] J. L. Carbonera and M. Abel, "An entropy-based subspace clustering algorithm for categorical data," in *Proc. IEEE 26th Int. Conf. Tools Artif. Intell.*, Nov. 2014, pp. 272–277.
[34] D. Brugger, M. Bogdan, and W. Rosenstiel, "Automatic cluster detection in Kohonen's SOM," *IEEE Trans. Neural Netw.*, vol. 19, no. 3, pp. 442–459, Mar. 2008.
[35] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, Mar. 2002.
[36] J. Al-Shaqsi and W. Wang, "A clustering ensemble method for clustering mixed data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
[37] B.-Z. Qiu, X.-L. Li, and J.-Y. Shen, "Grid-based clustering algorithm based on intersecting partition and density estimation," in *Emerging Technologies in Knowledge Discovery and Data Mining*, T. Washio, Z.-H. Zhou, J. Z. Huang, X. Hu, J. Li, C. Xie, J. He, D. Zou, K.-C. Li, and M. M. Freire, Eds. Berlin, Germany: Springer, 2007, pp. 368–377, doi: 10.1007/978-3-540-77018-3_37.
[38] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
[39] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. VLDB*, vol. 97, 1997, pp. 186–195.
[40] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," in *Proc. 24th. Int. Conf. Very Large Databases*, vol. 98. Aug. 1998, pp. 428–439.
[41] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: AAAI Press, 1998, pp. 58–65.
[42] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering," in *Proc. 25th Int. Conf. Very Large Databases*, 1999, pp. 506–517.
[43] W. Cheng, W. Wang, and S. Batista, "Grid-based clustering," in *Data Clustering*. London, U.K.: Chapman & Hall, 2018, pp. 128–148.
[44] Z. Dafir, Y. Lamari, and S. C. Slaoui, "A survey on parallel clustering algorithms for big data," *Artif. Intell. Rev.*, vol. 54, no. 4, pp. 2411–2443, 2020.
[45] A. Zubaroğlu and V. Atalay, "Data stream clustering: A review," 2020, *arXiv:2007.10781*. [Online]. Available: http://arxiv.org/abs/2007.10781
[46] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.
[47] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, 1998.
[48] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 186–193.
[49] H. Tong, S. Papadimitriou, J. Sun, P. S. Yu, and C. Faloutsos, "Colibri: Fast mining of large static and dynamic graphs," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 686–694.
[50] C. Boutsidis, A. Zouzias, and P. Drineas, "Random projections for k-means clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 298–306.
[51] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 430–437.
[52] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.
[53] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
[54] S. Maitrey and C. K. Jha, "An integrated approach for CURE clustering using map-reduce technique," *Proc. Elsevier*, pp. 563–571, 2013.
[55] S. Papadimitriou and J. Sun, "DisCo: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 512–521.
[56] W. Zhao, H. Ma, and Q. He, "Parallel K-means clustering based on MapReduce," in *Cloud Computing*, M. G. Jaatun, G. Zhao, and C. Rong, Eds. Berlin, Germany: Springer, 2009, pp. 674–679, doi: 10.1007/978-3-642-10665-1_71.
[57] R. L. F. Cordeiro, C. Traina, A. J. M. Traina, J. López, U. Kang, and C. Faloutsos, "Clustering very large multi-dimensional datasets with MapReduce," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 690–698.
[58] Q. Zhang and Z. Chen, "A weighted kernel possibilistic c-means algorithm based on cloud computing for clustering big data," *Int. J. Commun. Syst.*, vol. 27, no. 9, pp. 1378–1391, 2014.
[59] Y.-I. Kim, Y.-K. Ji, and S. Park, "Big text data clustering using class labels and semantic feature based on Hadoop of cloud computing," *Int. J. Softw. Eng. Appl.*, vol. 8, no. 4, pp. 1–10, 2014.
[60] S. Arjunan, S. Deris, R. M. Illias, and M. S. Mohamad, "A parallelizing interface for K-means type clustering algorithms and neural network batch training," in *Malaysian-Japan Seminar on Artificial Intelligence Applications in Industry*. Kuala Lumpur, Malaysia, Aug. 2003.
[61] X. Gao, E. Ferrara, and J. Qiu, "Parallel clustering of high-dimensional social media data streams," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 323–332.
[62] D. Kimovski, J. Ortega, A. Ortiz, and R. Banos, "Feature selection in high-dimensional EEG data by parallel multi-objective optimization," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2014, pp. 314–322.
[63] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[64] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data: Recent Advances in Clustering*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Berlin, Germany: Springer, 2006, pp. 25–71, doi: 10.1007/3-540-28349-8_2.
[65] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.
[66] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: Finding a match for a biomedical application," *Briefings Bioinf.*, vol. 10, no. 3, pp. 297–314, Dec. 2008.
[67] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 267–279, Sep. 2014.
[68] A. Amini, T. Y. Wah, and H. Saboohi, "On density-based data streams clustering algorithms: A survey," *J. Comput. Sci. Technol.*, vol. 29, no. 1, pp. 116–141, Jan. 2014.

[69] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.
[70] M. Ghesmoune, M. Lebbah, and H. Azzag, "State-of-the-art on clustering data streams," *Big Data Anal.*, vol. 1, no. 1, p. 13, Dec. 2016.
[71] S. Mansalis, E. Ntoutsi, N. Pelekis, and Y. Theodoridis, "An evaluation of data stream clustering algorithms," *Stat. Anal. Data Mining ASA Data Sci. J.*, vol. 11, no. 4, pp. 167–187, 2018.
[72] A. Pérez-Suárez, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "A review of conceptual clustering algorithms," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1267–1296, Aug. 2019.
[73] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[74] R. Sharan and R. Shamir, "Click: A clustering algorithm with applications to gene expression analysis," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 8, no. 307, 2000, p. 16.
[75] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.
[76] M. J. Zaki and M. Peters, "CLICKS: Mining subspace clusters in categorical data via K-partite maximal cliques," in *Proc. 21st Int. Conf. Data Eng. (ICD)*, Apr. 2005, pp. 355–356.
[77] A. N. Mahmood, C. Leckie, and P. Udaya, "An efficient clustering scheme to exploit hierarchical data in network traffic analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 752–767, Jun. 2008.
[78] Q. Cao, B. Bouqata, P. D. Mackenzie, D. Messier, and J. J. Salvo, "A grid-based clustering method for mining frequent trips from large-scale, event-based telematics datasets," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 2996–3001.
[79] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "The ClusTree: Indexing micro-clusters for anytime stream mining," *Knowl. Inf. Syst.*, vol. 29, no. 2, pp. 249–272, Nov. 2011.
[80] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion?" *J. Classification*, vol. 31, no. 3, pp. 274–295, Oct. 2014.
[81] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.
[82] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," in *Proc. 14th Int. Conf. Data Eng.*, Feb. 1998, pp. 324–331.
[83] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 169–194, Jun. 1998.
[84] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999.
[85] D. Gibson, J. Kleinberg, and P. Raghavan, "Clustering categorical data: An approach based on dynamical systems," *VLDB J. Int. J. Very Large Data Bases*, vol. 8, nos. 3–4, pp. 222–236, Feb. 2000.
[86] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[87] C. Bohm, K. Kailing, H.-P. Kriegel, and P. Kroger, "Density connected clustering with local subspace preferences," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 27–34.
[88] K. Kailing, H.-P. Kriegel, and P. Kröger, "Density-connected subspace clustering for high-dimensional data," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2004, pp. 246–256.
[89] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2006, pp. 328–339.
[90] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2007, pp. 133–142.
[91] B. Andreopoulos, A. An, and X. Wang, "Hierarchical density-based clustering of categorical data and a simplification," in *Advances in Knowledge Discovery and Data Mining*, Z.-H. Zhou, H. Li, and Q. Yang, Eds. Berlin, Germany: Springer, 2007, pp. 11–22, doi: 10.1007/978-3-540-71701-0_5.
[92] B. Andreopoulos, A. An, X. Wang, M. Faloutsos, and M. Schroeder, "Clustering by common friends finds locally significant proteins mediating modules," *Bioinformatics*, vol. 23, no. 9, pp. 1124–1131, May 2007.
[93] C. Ruiz, E. Menasalvas, and M. Spiliopoulou, "C-DenStream: Using domain knowledge on a data stream," in *Discovery Science*, J. Gama, V. S. Costa, A. M. Jorge, and P. B. Brazdil, Eds. Berlin, Germany: Springer, 2009, pp. 287–301, doi: 10.1007/978-3-642-04747-3_23.
[94] A. Forestiero, C. Pizzuti, and G. Spezzano, "FlockStream: A bio-inspired algorithm for clustering evolving data streams," in *Proc. 21st IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2009, pp. 1–8.
[95] J. Lin and H. Lin, "A density-based clustering over evolving heterogeneous data stream," in *Proc. ISECS Int. Colloq. Comput., Commun., Control, Manage.*, Aug. 2009, pp. 275–277.
[96] L.-X. Liu, H. Huang, Y.-F. Guo, and F.-C. Chen, "RDenStream, a clustering algorithm over an evolving data stream," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, Dec. 2009, pp. 1–4.
[97] J. Ren and R. Ma, "Density-based data streams clustering over sliding windows," in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 5, Aug. 2009, pp. 248–252.
[98] A. Amini, T. Y. Wah, and Y. W. Teh, "Dengris-stream: A density-grid based clustering algorithm for evolving data streams over sliding window," in *Proc. Int. Conf. Data Mining Comput. Eng.*, Dec. 2012, pp. 206–210.
[99] I. Ntoutsi, A. Zimek, T. Palpanas, P. Kröger, and H.-P. Kriegel, "Density-based projected clustering over high dimensional data streams," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 987–998.
[100] M. Hassani, P. Spaus, M. M. Gaber, and T. Seidl, "Density-based projected clustering of data streams," in *Scalable Uncertainty Management*, E. Hüllermeier, S. Link, T. Fober, and B. Seeger, Eds. Berlin, Germany: Springer, 2012, pp. 311–324, doi: 10.1007/978-3-642-33362-0_24.
[101] A. Amini, H. Saboohi, T. Herawan, and T. Y. Wah, "MuDi-stream: A multi density clustering algorithm for evolving data stream," *J. Netw. Comput. Appl.*, vol. 59, pp. 370–385, Jan. 2016.
[102] E. Schikuta and M. Erhart, "The BANG-clustering system: Grid-based data analysis," in *Advances in Intelligent Data Analysis Reasoning about Data*, X. Liu, P. Cohen, and M. Berthold, Eds. Berlin, Germany: Springer, 1997, pp. 513–524, doi: 10.1007/BFb0052867.
[103] B. L. Milenova and M. M. Campos, "Clustering large databases with numeric and nominal values using orthogonal projections," Oracle Data Mining Technol., Oracle Corp., Boston, MA, USA, Jan. 1997.
[104] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1998, pp. 94–105.
[105] C.-H. Cheng, A. W. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1999, pp. 84–93.
[106] S. Goil, H. Nagesh, and A. Choudhary, "Mafia: Efficient and scalable subspace clustering for very large data sets," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 443. New York, NY, USA: ACM, 1999, p. 452.
[107] D. Barbará and P. Chen, "Using the fractal dimension to cluster datasets," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 260–264.
[108] C. C. Aggarwal and P. S. Yu, "Finding generalized projected clusters in high dimensional spaces," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2000, pp. 70–81.
[109] J. Gao, J. Li, Z. Zhang, and P.-N. Tan, "An incremental data stream clustering algorithm based on dense units detection," in *Advances in Knowledge Discovery and Data Mining*, T. B. Ho, D. Cheung, and H. Liu, Eds. Berlin, Germany: Springer, 2005, pp. 420–425, doi: 10.1007/11430919_49.
[110] C. Jia, C. Tan, and A. Yong, "A grid and density-based clustering algorithm for processing data stream," in *Proc. 2nd Int. Conf. Genetic Evol. Comput.*, Sep. 2008, pp. 517–521.
[111] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," *ACM Trans. Knowl. Discovery from Data*, vol. 3, no. 3, pp. 1–28, Jul. 2009.
[112] J. Ren, B. Cai, and C. Hu, "Clustering over data streams based on grid density and index tree," *J. Converg. Inf. Technol.*, vol. 6, no. 1, pp. 83–93, Jan. 2011.
[113] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.
[114] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, Jan. 1984.

[115] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artif. Intell.*, vol. 40, nos. 1–3, pp. 11–61, Sep. 1989.
[116] R. N. Dave and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 643–662, 1992.
[117] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, Aug. 1994.
[118] J. Stutz and P. Cheeseman, "Autoclass—A Bayesian approach to classification," in *Maximum Entropy and Bayesian Methods*, J. Skilling and S. Sibisi, Eds. Dordrecht, The Netherlands: Springer, 1996, pp. 117–126, doi: 10.1007/978-94-009-0107-0_13.
[119] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 6, pp. 2907–2912, Mar. 1999.
[120] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 554–560.
[121] C. Ordonez and E. Omiecinski, "FREM: Fast and robust EM clustering for large data sets," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 590–599.
[122] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1537–1544.
[123] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2009, pp. 638–649.
[124] X. H. Dang, V. Lee, W. K. Ng, A. Ciptadi, and K. L. Ong, "An EM-based algorithm for clustering data streams in sliding windows," in *Database Systems for Advanced Applications*, X. Zhou, H. Yokota, K. Deng, and Q. Liu, Eds. Berlin, Germany: Springer, 2009, pp. 230–235, doi: 10.1007/978-3-642-00887-0_18.
[125] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[126] C.-L. Yang, R. J. Kuo, C.-H. Chien, and N. T. P. Quyen, "Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering," *Appl. Soft Comput.*, vol. 30, pp. 113–122, May 2015.
[127] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proc. 1st Pacific–Asia Conf. Knowl. Discovery Data Mining, (PAKDD)*. Singapore, 1997, pp. 21–34.
[128] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," *DMKD*, vol. 3, no. 8, pp. 34–39, 1997.
[129] G. Karypis and V. Kumar, "Parallel multilevel series k-Way partitioning scheme for irregular graphs," *SIAM Rev.*, vol. 41, no. 2, pp. 278–300, Jan. 1999.
[130] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in *Proc. 41st Annu. Symp. Found. Comput. Sci.*, Nov. 2000, pp. 359–366.
[131] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering," in *Proc. 18th Int. Conf. Data Eng.*, Feb. 2002, pp. 685–694.
[132] C. C. Aggarwal, S. Y. Philip, J. Han, and J. Wang, "A framework for clustering evolving data streams," in *Proc. VLDB Conf.*, Amsterdam, The Netherlands: Elsevier, 2003, pp. 81–92.
[133] C. Fraley and A. E. Raftery, "Enhanced model-based clustering, density estimation, and discriminant analysis software: Mclust," *J. Classification*, vol. 20, no. 2, pp. 263–286, 2003.
[134] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proc. 13th Int. Conf. Very Large Data Bases*. Toronto, ON, Canada, vol. 30, 2004, pp. 852–863.
[135] Z. He, X. Xu, and S. Deng, "K-ANMI: A mutual information based clustering algorithm for categorical data," *Inf. Fusion*, vol. 9, no. 2, pp. 223–233, Apr. 2008.
[136] A. Zhou, F. Cao, W. Qian, and C. Jin, "Tracking clusters in evolving data streams over sliding windows," *Knowl. Inf. Syst.*, vol. 15, no. 2, pp. 181–214, May 2008.
[137] S. Deng, Z. He, and X. Xu, "G-ANMI: A mutual information based genetic clustering algorithm for categorical data," *Knowl.-Based Syst.*, vol. 23, no. 2, pp. 144–149, Mar. 2010.
[138] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler, "StreamKM++: A clustering algorithm for data streams," *J. Experim. Algorithmics*, vol. 17, pp. 1–2, May 2012.