Received November 10, 2021, accepted December 7, 2021, date of publication December 10, 2021,
date of current version January 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3134704

A Systematic Review of Density Grid-Based Clustering for Data Streams

MUSTAFA TAREQ[1], ELANKOVAN A SUNDARARAJAN[2], (Member, IEEE), AARON HARWOOD[3], (Member, IEEE), AND AZURALIZA ABU BAKAR[4], (Member, IEEE)

[1]Department of Computer Technology Engineering, Al-Hikma University College, Baghdad 10015, Iraq
[2]Centre for Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, Selangor 43600, Malaysia
[3]School for Computing and Information Science, The University of Melbourne, Melbourne, VIC 3010, Australia
[4]Centre for Artificial Intelligence and Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, Selangor 43600, Malaysia

Corresponding author: Elankovan A Sundararajan (elan@ukm.edu.my)

This work was supported in part by the Ministry of Higher Education Malaysia under Grant FRGS/1/2018/ICT04/UKM/02/1, and in part by Research University under Grant DIP-2018-041.

ABSTRACT Various applications, such as electronic business, satellite remote sensing, intrusion discovery, and network traffic monitoring, generate large unbounded data stream sequences at a rapid pace. The clustering of data streams has attracted considerable interest due to the increasing usage of evolving data streams. In particular, evolving data streams affect clustering because they introduce numerous challenges, such as time and memory limits and one-pass clustering. Furthermore, researchers need to be able to determine arbitrarily shaped clusters present in evolving data streams from applications. Due to these characteristics, conventional density grid-based clustering techniques cannot be used. Moreover, the existing density grid-based clustering algorithms have low cluster quality for clustering evolving data streams. This study conducted a systematic literature review (SLR) and noted numerous research-related issues encountered in solving the aforementioned problems. We summarized numerous grid-based clustering algorithms that have been used and determined their distinctive and limited features. We also observed how these algorithms address the challenges affecting the clustering of evolving data streams and studied their advantages and disadvantages. SLR was based on 104 articles published between 2010 and 2021. Numerous challenges remain for grid-based clustering algorithms, particularly in terms of time-limited and high-dimensional data handling. Last, our findings indicated a variety of active studies on density grid-based clustering.

INDEX TERMS Clustering, data stream, grid-based clustering, data stream clustering, density-based clustering.

I. INTRODUCTION

In recent years, rapid growth in the fields of computer intelligence and quarrying data streams has occurred, as mining instruments and specialized extraction devices have increased in popularity among users [1]–[5]. A total of 2.5 quintillion bytes of data are created daily, and over 90% of existing data worldwide have been generated in the past two years alone. In 2007, the amount of information created and collected globally exceeded the available storage capacity for the first time [6]–[9]. An increasing number of data streams demanding excessive storage capacity, real-time monitoring, and high-frequency sampling have been produced, thereby deviating from traditional static data collection [10]–[13]. At present, data streams are expected to be handled as they arrive. Consequently, the mining and real-time analysis of data streams have triggered renewed research focus on identifying and developing improved processes [12], [14]–[16]. Moreover, clustering algorithms are receiving increased consideration with regard to data mining optimization. Typical features of data streams are as follows [17]–[19].

- Data streams are rapidly evolving, collected in real time, and reliant on rapid processing, interpretation, and response.

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

- They are restricted in terms of storage. Therefore, only synopsis data can be saved, and obtaining fundamental data is a relatively challenging task.
- Data streams comprise an unceasing flow of huge data.
- Directly accessing the data stream is virtually impossible. Therefore, sophisticated algorithms must be used for processing streams to generate usable information over time.
- They are multidimensional and require specialized algorithms to enable data streaming.

The data stream mining process involves the challenging process of real-time extraction of valuable trends and patterns from dynamic streaming data using a single scan [12], [20]–[24]. Numerous scientists have investigated data stream clustering because it is a highly effective technique for data mining [6], [15], [21], [25], [26]. Clustering includes steps such as processing data and partitioning objects and information into a few subsets called clusters. This technique groups objects into individual clusters, thereby generating differentiation between the clusters [10], [27]–[31]. Clustering assists in data restructuring by (a) replacing the cluster using one or numerous novel representatives, (b) categorizing similar objects, and (c) finding new patterns. Clustering algorithms, which are used for processing large amounts of data, have previously included advanced filtering processes that could be used in machine learning and data mining processes for recognizing data patterns [1], [32]. For handling a large volume of data contained in hard disks or data streams, streaming access methods achieve better performance than random access methods. Therefore, streaming algorithms are preferable when confronted with a large data volume [33]–[36]. However, the application of conventional density clustering algorithms is inappropriate for evolving data streams because these data streams are inherently constantly evolving. Hence, new and improved density clustering processes should be developed to address this issue.

Density-based approaches create a data density profile for clustering purposes. Density-based clustering can filter noise or outliers, detect arbitrarily shaped clusters, and require only one scan of the raw data [18], [37]. Thus, clusters are regarded as dense areas of data points that are separated in the data space by sparse regions of low density [38]–[40]. One of the most common forms of density clustering methods is the set of density grid-based clustering, which forms another relevant clustering category that has long been considered. The recent advent of using grids in clustering algorithms has been a huge leap forward in the field of data stream mining [1], [27], [29], [36], [41].

Compared with other clustering methods, density grid-based clustering varies in terms of the adoption of a multipurpose grid data structure, in which data objects are mapped into grids and clusters are shaped based on their densities [27], [42], [43]. Thereafter, the data space is partitioned into a certain number of cells to produce a grid, in which the mapping of data records is done to create micro clusters that can eventually be used for the final clustering step [41]–[46]. In this manner, grid cells include synopsis information on the data stream [47]–[50]. However, we should consider several issues associated with density grid-based clustering, such as the majority cannot handle evolving data streams, suffer from the need for high memory, have low processing rates, or are not completely online methods [28], [51]–[54]. Furthermore, low cluster quality related to clustering evolving data streams is associated with the current density grid-based clustering algorithms. The current research provides an SLR of new approaches and techniques developed to address the problem of density grid-based clustering. Several objectives pertaining to density grid-based studies have been identified, and the different mechanisms and approaches involved have been segmented. Moreover, grid algorithms used in clustering data streams have been summarized while discussing limitations and distinctive characteristics, and how such algorithms can solve problems related to clustering data streams has been described, along with their advantages and disadvantages. We also proposed possible challenges and likely directions toward a future solution to deal with evolving data stream clustering that could help guide future research toward new solution mechanisms and undiscussed areas. Last, we provide overviews on density grid-based clustering paradigms and data streams to ensure that this survey is self-contained and enhances reader convenience. The following are the survey's contributions:

1. A comprehensive survey was conducted on the current density grid-based clustering algorithms published from 2010 to 2021.
2. We summarized numerous grid-based clustering
3. algorithms that have been used and determined their distinctive and limited features. We also observed how these algorithms address the challenges that affect the clustering of evolving data streams and studied the benefits and drawbacks of these clustering algorithms.
4. Current open challenges are identified and described to help guide future research directions on evolving data stream clustering.

The remainder of this paper is organized as follows. Section II presents a methodology review on research method objective determinations. Section IV describes data stream clustering. Section V provides the concept of density grid-based clustering. Section VI presents a description of grid-based clustering methods. Section VII discusses the findings. Section VIII provides the conclusions.

II. REVIEW METHOD

A systematic review is described as a process of interpreting and assessing available studies related to a specific research topic and area of interest [55]. Kitchenham described the various reasons why such a study is necessary. The most important reasons are to identify the topics that should be investigated further, synthesize all available studies related to a particular treatment and technology, and formulate a

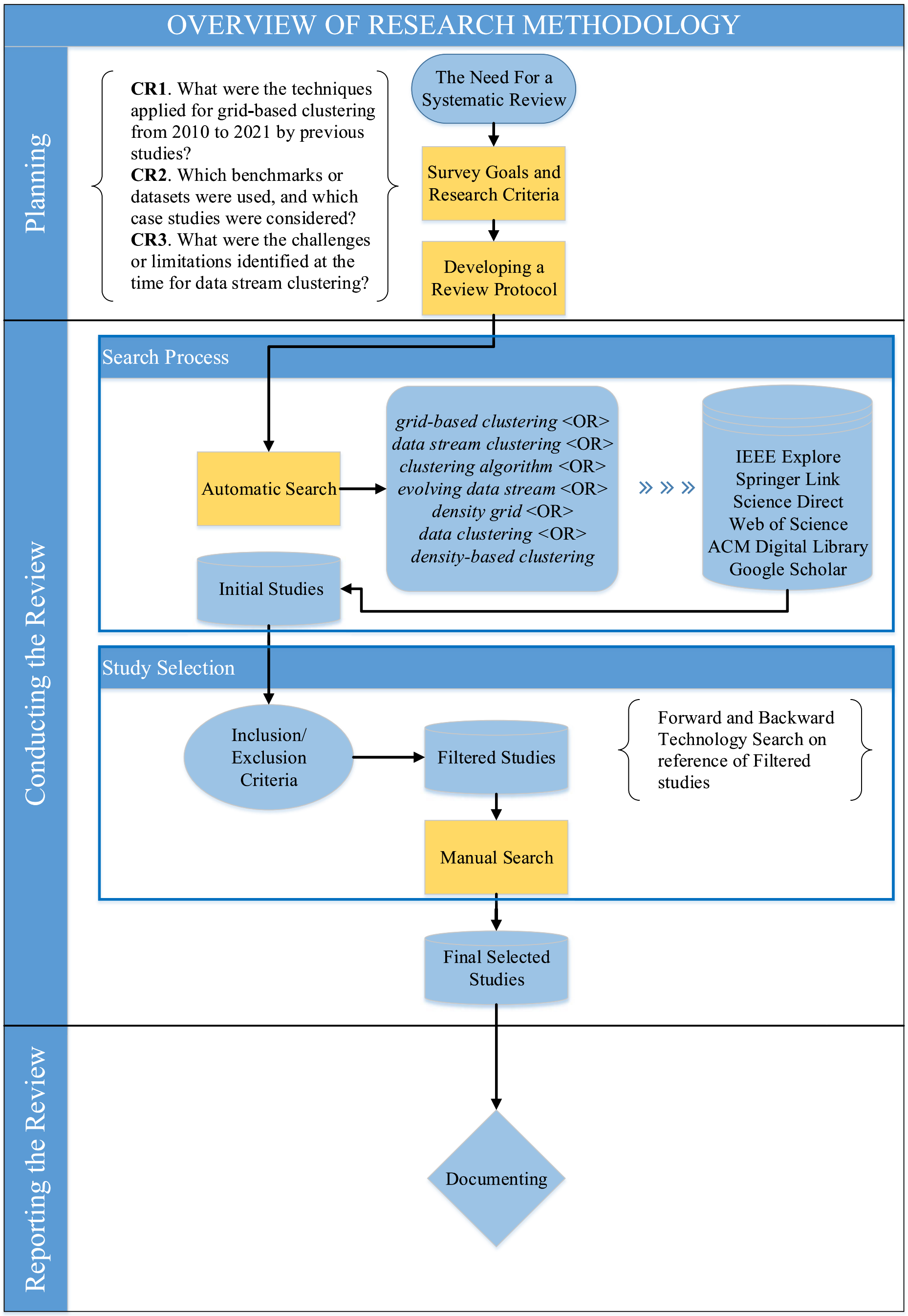

OVERVIEW OF RESEARCH METHODOLOGY
Planning
CR1. What were the techniques applied for grid-based clustering from 2010 to 2021 by previous studies?
CR2. Which benchmarks or datasets were used, and which case studies were considered?
CR3. What were the challenges or limitations identified at the time for data stream clustering?
The Need For a Systematic Review
Survey Goals and Research Criteria
Developing a Review Protocol
Conducting the Review
Search Process
Automatic Search
grid-based clustering <OR> data stream clustering <OR> clustering algorithm <OR> evolving data stream <OR> density grid <OR> data clustering <OR> density-based clustering
IEEE Explore
Springer Link
Science Direct
Web of Science
ACM Digital Library
Google Scholar
Initial Studies
Study Selection
Inclusion/ Exclusion Criteria
Filtered Studies
Forward and Backward Technology Search on reference of Filtered studies
Manual Search
Final Selected Studies
Reporting the Review
Documenting

FIGURE 1. Overview of the research methodology.

specific background for describing the newest studies. This section presents our review protocol, which includes several steps described by Kitchenham [55]. Figure 1 shows a summary of the methodological steps for SLR. The following section explains the details of these steps:

A. NEED FOR A SYSTEMATIC REVIEW

Although numerous techniques have been suggested for data stream clustering, there is still a need for a comprehensive data stream clustering strategy. This study investigates the density grid-based clustering methods developed for data stream clustering in the literature review, apart from those that have attempted to provide an overview of the vast literature on techniques. Numerous techniques for data stream clustering using density grid-based clustering techniques have been developed for a single study area or specific application. This survey significantly expands the discussion in several directions according to the CRs.

B. SURVEY GOALS AND RESEARCH CRITERIA

The current study aims to collect and investigate all effective, available, and credible studies on density grid-based

TABLE 1. List of electronic databases.

Databases	URLs
IEEE Explore	"www.ieeexplore.ieee.org"
SpringerLink	"www.link.springer.com"
ScienceDirect	"www.sciencedirect.com"
Web of Science	"www.webofknowledge.com"
ACM Digital Library	"www.dl.acm.org"
Google Scholar	"www.scholar.google.com"

clustering algorithms. Particular focus is given to the identification of methods from papers, their key features, and descriptions of their respective characteristics. To achieve these goals and identify the selected research methodologies, the case studies considered are only those for which new methodologies are proposed, actual datasets are used, and benchmarks are applied. The following criteria (CRs) are investigated:

CR1. What were the techniques applied by previous studies for grid-based clustering from 2010 to 2021?
CR2. Which benchmarks or datasets were used, and which case studies were considered?
CR3. What were the challenges or limitations identified at the time for data stream clustering?

C. DEVELOPING A REVIEW PROTOCOL

A key step in carrying out SLR is the review protocol, which aids in determining the methods beneficial for systematic review. The review protocol primarily aims to minimize study bias and differentiate SLR versus the traditional methods of the literature review [56]. This review protocol helps classify the "search strategy, review background, extraction of data, development of CRs, data synthesis and study selection criteria." An explanation pertaining to review background and relevant CRs has been previously provided. The next section offers details for describing the other elements.

D. AUTOMATIC SEARCH

This section outlines how each paper in our review was identified. High-level, prestigious international conferences and reference journals were selected from electronic databases using the standard manual search method. Table 1 provides a list of the electronic databases.

Our search string was structured to retrieve as many papers as possible related to density grid-based clustering, which is our subject of concern in this SLR study. We used a mix of keywords containing (*grid-based clustering* OR *data stream clustering* OR *clustering algorithm* OR *evolving data stream* OR *density grid* OR *data clustering* OR *density-based clustering*) and other variations, combined by the "OR" operator.

E. INCLUSION AND EXCLUSION CRS

The inclusion and exclusion CRs were applied to ensure that basic studies on SLR were appropriate and to help answer the research questions presented in SLR. We only selected research papers (from conferences and journals), written in English, and published between 2010 and 2020 on online digital databases. Only papers related to the institutional repository were selected in the study. Papers that were unrelated to the institutional repository domain were not selected in this study. Furthermore, only papers that fulfil any of the research objectives are included in the current research. Given that we could not clearly understand papers written in languages other than English, these articles were excluded from the current study. Table 2 presents the various inclusion and exclusion CRs used in this SLR study.

F. MANUAL SEARCH

By referring to [57], we used a forward and backward search for identifying citations pertaining to primary studies. Google Scholar was used to identify all studies cited in the chosen primary study. A manual search was performed to ensure the relative completeness and comprehensiveness of a systematic review pertaining to the research and guarantee that nothing was missed.

Mendeley (https://www.mendeley.com) was used to manage and sort all studies and eliminate duplicate research.

G. DATA COLLECTION

We obtained articles from ScienceDirect, Springer Link, IEEE-Xplore, ACM Digital Library, and Web of Science. The combined results from all databases were 344 articles. After removing duplicates and papers not written in English, 298 articles remained. Following a review of the titles and keywords of the 298 articles, we obtained 165 articles related to our study. Thereafter, we reviewed the abstracts of these articles and selected 104 articles, which were read thoroughly.

We used the search strategy as a basis in utilizing the PRISMA guidelines shown in Fig. 2. In particular, Fig. 2 highlights the total number of papers identified, screened, and that were eligible and included in this SLR study. The PRISMA flowchart indicates the number of studies extracted from the published databases and statistics on the number of papers included/excluded in this study. All reasons for inclusion/exclusion were defined in the current research. Finally, the PRISMA flow diagram presents the numbers of papers included in the quantitative and qualitative analyses.

III. DATA STREAM CLUSTERING

Data streams are noted in various real-world fields, such as accounting, agriculture, and health care, and they generate large amounts of data daily [58], [59]. Moreover, data streams constantly evolve, and the amount of data becomes unlimited after a certain period. Researchers face numerous challenges when dealing with this issue because of the physical

TABLE 2. Inclusion and exclusion CRs.

Inclusion CRs	Exclusion CRs
Articles are written in English and polished	Written in a language other than English
Published within 2010–2021	Falls outside the selected period
Addresses a specific technique, such as the density grid method	Focuses on other techniques
Directly or indirectly addresses CRs	Does not match the inclusion CRs

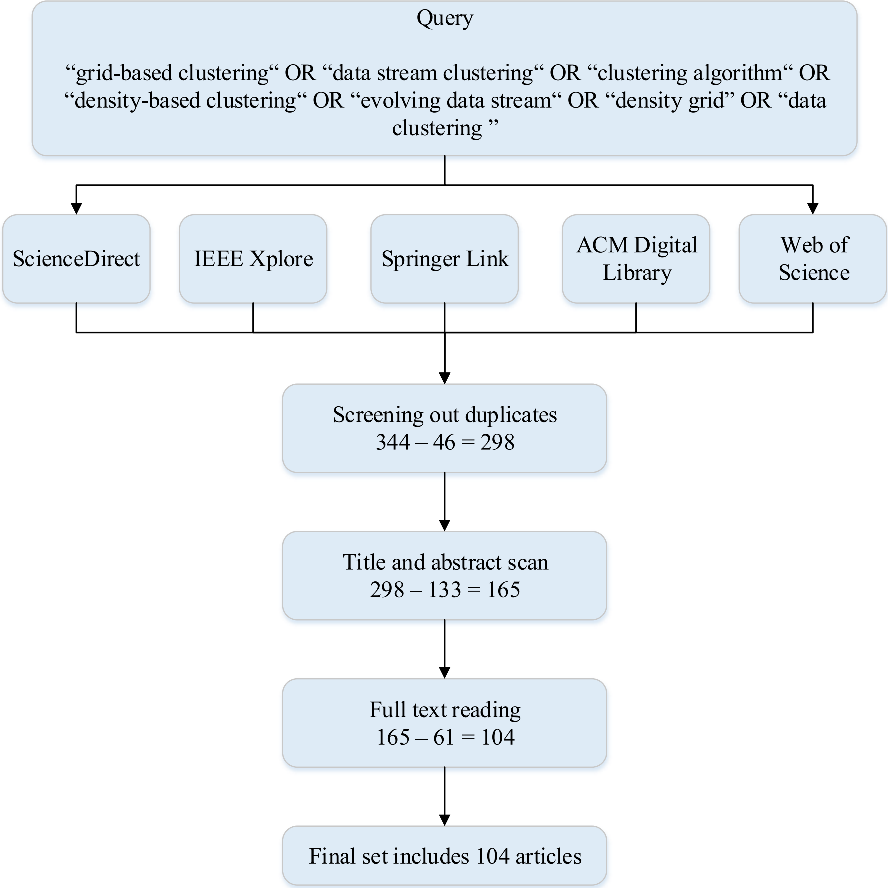

Query
“grid-based clustering“ OR “data stream clustering“ OR “clustering algorithm“ OR “density-based clustering“ OR “evolving data stream“ OR “density grid” OR “data clustering ”
ScienceDirect
IEEE Xplore
Springer Link
ACM Digital Library
Web of Science
Screening out duplicates
344 – 46 = 298
Title and abstract scan
298 – 133 = 165
Full text reading
165 – 61 = 104
Final set includes 104 articles

FIGURE 2. PRISMA flowchart.

limitations of existing computational resources [60], [61]. In the past 10 years, researchers have shown higher interest than ever in managing these unbounded and massive data sequences, which are created at rapid rates [22], [62]–[64]. A data stream S is defined as a massive sequence of data objects, $X^1, X^2, \ldots, X^N$, where $S = \left\{X^i\right\}_{i=1}^{N}$ is potentially unbounded ($N \to \infty$). Every data object 1 can be described using the n-dimensional attribute vector $X^i = \left[x_j^i\right]_{j=1}^{n}$, which belongs to the attribute space Ω. Moreover, Ω can be categorical, continuous, or mixed [22]. However, major issues with data streams are their management, analysis, storage, and recovery [10], [22], [59], [65].

Clustering is a data mining task used for clustering large datasets to make data objects in a group similar to each other and differentiate them from the points of other clusters [66]–[68]. Clustering of data streams raises new challenges, such as noisy data, restricted memory, evolving data, limited time, single-pass clustering, and multi-dimensionality [6], [11], [22], [69]–[72]. Data streaming requires real-time processing to manage the large arrival rates of data, and interpreted results are expected within a short timeframe. Moreover, maintaining an entire data stream in dynamic memory is often impossible due to the unlimited quantity of data being transmitted. The data stream also passes only once, making multiple active scans impossible to perform [14], [33], [73]–[76]. Two significant issues in all real-time data collection methods are the processing of continuously evolving data and the storage of raw data objects for later analysis. Several methods exist for accomplishing these tasks, such as online-offline processing, single-pass processing, online processing, and evolving; and various techniques for summarizing data streams. A brief explanation of these methods is provided as follows.

A. PROCESSING

1) EVOLVING

Clusters are measured in a single-pass approach through the entire data stream. Data streams are continuous and evolve continuously over time. Therefore, clustering outcomes change significantly over time [10], [73], [77], [78]. In the ''evolving'' approach, the behavior of a stream is considered a process that evolves and is processed in the form of several window models. Various approaches have been developed based on this method [14], [20], [29], [52], [73], [75], [78]–[81].

2) ONLINE–OFFLINE

Clustering algorithms for data streams occasionally need to validate clusters in different portions of the stream. To track the evolving behaviors of data streams, a variable time window model is required, but the dynamic clustering of data streams across all time horizons cannot be achieved. Therefore, the ''online-offline'' method was created [73], in which the online phase keeps summaries of data streams, whereas the offline phase provides comprehension of clusters [10], [27], [73], [77]. The majority of clustering methods use a two-phase approach designed to evolve data streams [10], [27], [43], [52], [73], [79], [81]–[84].

3) ONLINE

The ''online'' technique provides a clustered data stream that is continuously updated with every data point added [85]. Fully online unsupervised learning can detect anomalies as they occur in real time, regardless of how the ''normal'' stream behavior or nature of the anomalies changes. As such, the online technique uses an evolving data stream clustering algorithm [29], [30], [85]–[87].

4) SINGLE-PASS

For the ''single-pass'' technique, data streams are clustered by scanning only once. The assumption is that the data arrive in k-cluster chunks, are generated through a K-means clustering method and are used as a data stream. STREAM is another well-known algorithm that uses a local search algorithm to partition the input stream into chunks and generate a cluster [14], [75].

B. SUMMARIZATION

Large data volumes impose some time and space constraints on the computational phase of algorithms. Furthermore, the algorithms cannot retain the complete data record because of the large size of the data stream. Hence, a synopsis was constructed using data items in the data stream [10]. The design and selection of a specific synopsis approach is dependent on the issue that must be solved [10], [27]. Hence, the following techniques are used for summarization.

1) SAMPLING METHODS

Instead of obtaining a full stream of data, which is not feasible, a data sample can be collected from a data stream. Reservoir sampling refers to a sampling technique that can be used to select random data points. This technique is helpful for data streams [88].

2) WAVELETS

Wavelets are a common summarization technique used for data streams. They can be utilized to summarize and represent data for processing different types of signals and images [89]. The wavelet technique can be used for processing different types of signals and images. They can be used with an input signal for multi-resolution hierarchical systems. Furthermore, wavelet-based histograms are interpreted dynamically over a long period [62], [90]–[94].

3) HISTOGRAMS

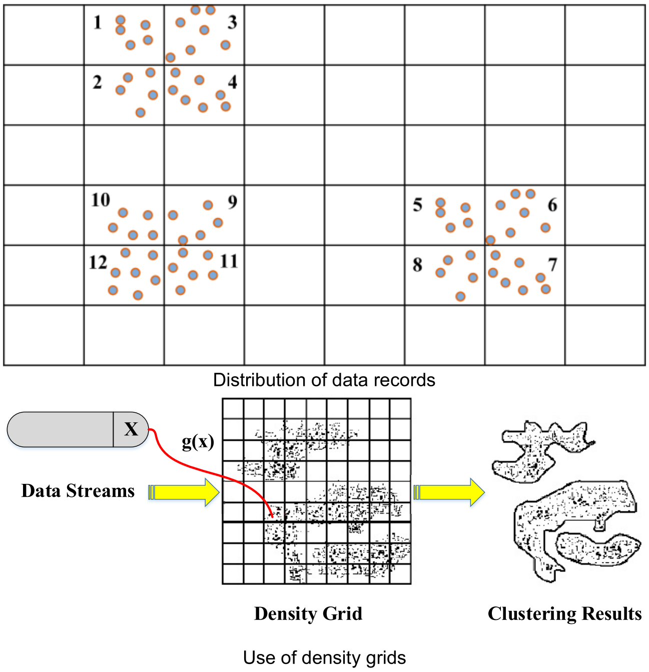
A histogram refers to a synopsis, which helps in the accurate approximation of a primary data distribution and offers a graphical representation of data. Histograms are commonly used to compute the aggregate statistics, approximate query answers, query optimization, and selectivity estimation [95]. Numerous studies have proposed using histograms for data summarization. The current study discusses developments in the synopsis structure. In particular, researchers have proposed the adaptive cumulative windows model (ACWM) algorithm, which summarizes data streams using histograms [96].

4) SKETCHES

Sketches are a data summarization technique that helps in analyzing data streams. The sketch-based approach is derived from the wavelet process. A sketch refers to a randomized linear projection of the primary data vector, which is regarded as a randomized version of the wavelet process. Other techniques highlight a small section of data, although sketches summarize the complete dataset at numerous detailed levels [65]. The current researchers introduce a count-min sketch for data stream summarization [97].

5) GRID

Grids are considered a popular technique for summarizing data from data streams. In this technique, the data space is partitioned into small sections known as grids [36]. Thereafter, objects in the data stream are plotted into these grids. Every grid includes a vector of characteristics that summarizes data points mapped to it. Clusters are formed based on a grid density [10], [27], [83].



Distribution of data records

Use of density grids

FIGURE 3. Framework for data distribution and density grid clustering.

6) MICRO CLUSTER

This important technique is used in data stream clustering to effectively compress data streams. This technique can perform adjustments effectively after the evolution of primary data streams [27], [29], [98].

IV. THE CONCEPT OF DENSITY GRID-BASED CLUSTERING

Clustering is an imperative undertaking in collecting a data stream to ensure that the most representative dataset possible is preserved according to the capabilities of the system used [1], [14]. Numerous clustering approaches have been suggested for use with data streams [3], [14], [18], [75]. Previous data stream clustering approaches have been applied in a "single-stage model" that handles data stream clustering through a continuous type of fixed and stagnant data clustering [41], [50], [99]–[102].

By merging "grid-based clustering" with "density-based clustering algorithms," researchers have developed different algorithms, such as [10], [27], [41]–[43], [46], [47], [50], [80], [102]–[106]. In density grid-based clustering, data points are plotted on a grid, and clusters are formed thereafter based on grid densities, as shown in Fig. 3.

The first step of the grid-based algorithm involves dividing the area such that all data points are mapped on the grid structure. Grid size refers to the parameter at runtime, in which each dimension has a similar grid spacing, thereby resulting in a uniform grid. Thereafter, the algorithm analyzes every data point (a highly parallelizable step) and determines which grid the point belongs to [42], [50], [107]. The density measure in each grid square is increased by 1, and the point is located in the same grid square [27], [42], [50], [102], [107].

As described in Eqs. (1, 2, and 3), every grid cell is defined by an n-dimensional value vector. Each value denotes the offset of the grid cell from the original cell in the same dimension (G_i represents the offset in the i_{th} dimension). In each dimension, the offset value can be derived by extracting the value of the dimension from its data point and dividing it thereafter using the grid size and noting the result [50], [107]–[109]. This algorithm groups several data points and counts thereafter the number belonging to each group; it is intended to work with high-dimensionality data [50], [66], [103], [107], [108], [110].

$$\mathrm{GridCell} = [G_1, G_2, \ldots, G_{n-1}, G_n] \qquad \text{Eq. (1)}$$

$$DataPoint = [D_1, D_2, \ldots, D_{n-1}, D_n] \qquad \text{Eq. (2)}$$

$$G_i = D_i \big/ GridSize \qquad \text{Eq. (3)}$$

V. DENSITY GRID-BASED CLUSTERING ALGORITHMS

This section reviews the use of "density grid-based clustering" techniques on data streams. Most of the revised algorithms' features show that the majority of these algorithms are based on a "CluStream" structure [73]. The algorithms possess online-offline stages. The online stage encapsulates the algorithm's data records, whereas the offline stage performs the clustering process according to the summary information. Data stream clustering algorithms are applied using two approaches: the "one-pass approach" and the "evolving approach." The one-pass approach groups data streams according to one-time skimming, whereas the evolving approach may be modified or adjusted for different data streams. Fig. 4 illustrates the various options used when creating grid-based clustering.

A. IGDCL

A novel irregular grid-based clustering (IGDCL) algorithm [41] has been developed for high-dimensional data streams. The IGDCL algorithm showed better performance than previous algorithms because it splits the d-dimensional space into grids. Thereafter, every data object is mapped on the adaptive grid, decreasing the effects of the sizes of grid cells and borders to overcome the sparsity of a high-dimensional data stream. When a data stream enters the algorithm, the irregular grid structure can be incrementally updated, and a final cluster is generated that groups the dense grids into subspaces. The IGDCL algorithm uses a density fading function for every data point, reflecting the data stream's evolution. However, it does not present a simple plan for eliminating intermittent grids. A novel partition method based on the data's distribution information has also been used to generate an irregular grid structure. In this method, subspaces composed of dimensions related to the subsequent clusters are developed. This algorithm is similar to CluStream [73] because it includes the average value of the sum of squares distance (SSQ).

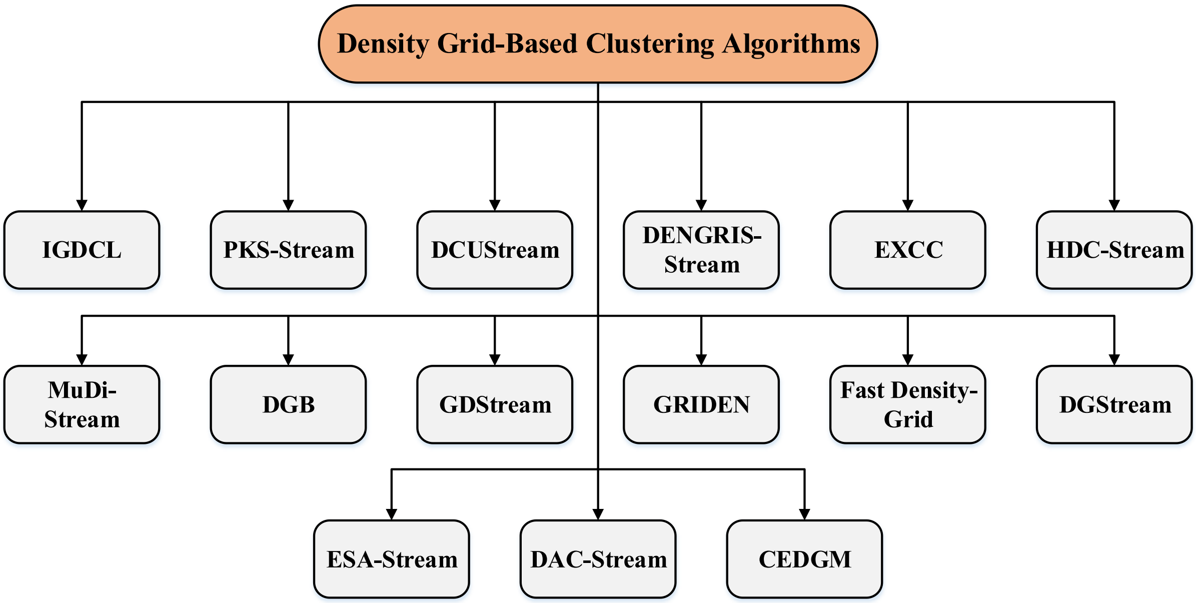


FIGURE 4. Categorization of density grid-based clustering algorithms.

B. PKS-STREAM

Many current "density-grid clustering" algorithms are unable to control high-dimensional data streams. For this reason, [111] favored the "PKS-Stream" algorithm, which is founded on the "Pks-tree" and "density grid," in which the storage facility and indexing effectiveness are enhanced. The grid-based clustering approach leaves several cells vacant, particularly for high-dimensional data; when all grids are preserved, the algorithm's time complexity increases. However, in instances where only occupied grids are saved, the algorithm overlooks the connections between grids. Therefore, the Pks-tree algorithm is preferred when recording dense cells and connections between grids because it possesses online and offline stages.

In the PKS-Stream algorithm's online stage, data streams are continuously read and captured onto connected Pks-tree grid cells. In the offline stage, clustering begins with the new-level dense cells of the Pks-tree leaf. The algorithm checks the grid's density and produces a different cluster when the cell density is above the threshold. To improve the algorithm's effectiveness, vacant grid cells are detached using the "K-cover" concept, which displays the number of dense grids adjacent to the leaf node grids. PKS-Stream is a "density grid-based clustering" that handles high-dimensional data. Nevertheless, after mapping a new point to any of the tree cells, no pruning is performed on the tree [10]. PKS-Stream is influenced by K in terms of the clustering result and by "K-cover," which drives the cluster determination process.

C. DCUSTREAM

"DCUStream" [112] is a novel dynamic algorithm to address uncertainty in data streams and was proposed by Yang *et al.* in 2012. They indicated that for every data point in the stream, the probability of its occurrence and its appearance time is measured. Every data point is drawn onto the grid. The algorithm reflects the uncertainty weighting of every measured data point derived from advanced data stream characteristics and its probability of occurrence. DCUStream considers the main grid to be a dense grid with neighbors, where sparse and dense grids are described by considering the uncertainty threshold of data density. For clustering, DCUStream scrutinizes all grids to identify the main dense grid and applies the "depth-first search algorithm" to identify neighboring dense grids; the progression continues for all unmarked dense grids. All sparse grids are known as noise. This algorithm enhances the effectiveness of "density-based" clustering in undefined data stream environments. The weakness of DCUStream is that mining for density grids and their neighbors is an extremely time-consuming process [59].

D. DENGRIS-STREAM

The "density grid-based using a sliding window" (DENGRIS-Stream) algorithm has been proposed for clustering data streams [47]. This process uses the "density grid-based clustering" technique for a data stream by using a "sliding window." This algorithm distributes data points on a grid, calculates each grid's density value, and groups density grids into time window units. Dissemination of the latest records using the sliding window is also noted and is required in various data stream applications. The DENGRIS-Stream algorithm generates a final grid model and identifies and eliminates grids with no timestamps in the sliding window. Moreover, this algorithm eliminates terminated grids before data points are migrated to the grid, where timestamps and memory are preserved. However, current researchers did not compare the performance of this algorithm to those of other algorithms.

E. ExCC

The ''exclusive and complete clustering'' (ExCC) algorithm [43] possesses online and offline phases [52], [80], [83], [113]. The online stage retains a summary of a grid's data, whereas the offline stage produces macro clusters. ExCC is a widely used algorithm because it trims according to the data streaming speed instead of using window dispersion. This fashionable algorithm uses a grid approach for data dissemination and handles noise in the offline stage through a ''watch and wait for policy.'' ExCC utilizes a density factor to isolate sparse and dense grids and uses group density and cell density thresholds to remove noise. This algorithm also calculates thresholds depending on data measurements, the typical number of objects in each grid, and grid granularity. To produce clusters, ExCC considers recent and dense grids before selecting a dense grid, as well as considering its eight closest neighbors, while ensuring that measurements of grid qualities are introduced. The algorithm's constraint is that it requires more processing time than other algorithms to increase its memory capacity [20], [59].

F. HDC-STREAM

Amini *et al.* (2014) suggested a density-based clustering algorithm for the Internet of Things (IoT). The ''hybrid density-based clustering for data streams'' (HDC-Stream) algorithm [10] requires low processing time, making it appropriate for real-time applications with IoT devices. HDC-Stream applies a hybrid method and carries out clustering in three different stages while capitalizing on the benefits of the density grid-based and micro clustering methods. This algorithm can identify randomly shaped groups and treat them as outliers in the online and offline stages. The online stage of ''HDC-Stream'' continuously reads new incoming data records and either supplements them to the present mini cluster or draws them onto the grid. HDC-Stream occasionally removes outliers during the trimming process. Thereafter, final clusters are produced by request in the offline stage. The current authors have shown that this algorithm rapidly generates high-quality results for clustering data streams but cannot efficiently store and cluster multi density data [20], [27].

G. MuDi-STREAM

Other researchers have recommended the ''multi density data stream'' (MuDi-Stream) algorithm for cases in which the variety of data leads to unreliable clustering results [27]. This situation is viewed as an ''online-offline'' algorithm with four parts [43], [52], [80], [83], [113]. In the online step, this algorithm retains the data synopsis depending on the development of multi density data streams as major mini-clusters. In the offline step, MuDi-Stream produces final groups using the ''modified density-based'' algorithm. Researchers have used a hybrid technique that utilizes micro clustering and grid-based algorithms to derive the synopsis information of a given data point. A grid technique is utilized to develop mini-clusters and determine outliers with diverse ranges, which helps improve the efficiency of density data, thereby significantly decreasing merging time. The MuDi-Stream algorithm consists of four major factors: mapping data components, generating mini-cluster components, pruning grids and mini-cluster components, and generating macro cluster components. The first three components are utilized in the online step, while the offline step uses the fourth component. Each component carries out a vital function during the overall process, including assimilation, drawing and generating mini-clusters, trimming mini-clusters or grids, and generating the final clusters. Another ''density-based'' clustering approach (i.e., M-DBSCAN) was recommended for offline phases. The MuDi-Stream algorithm does not present the obtained data synopsis because it cannot handle high-dimensional data streams owing to an increase in the number of vacant grids, thereby decreasing the processing time.

H. DGB

Wu and Wilamowski (2016) proposed the ''density and grid-based'' (DGB) [42] technique for classifying data using arbitrary shapes or noise. They calculated the distance between a few grid nodes instead of estimating the Euclidean distance of mutual patterns. Three steps are included in this algorithm. Step 1 involves scaling and normalization of the primary dataset into the standard grid, in which the original dataset is normalized to [0, 1] in every dimension and scaled thereafter onto the $[1, N_{grid}]$ range grid (i.e., N_{grid} denotes the range of a grid in every dimension). This approach simplifies the calculation of the local density of a grid. Step 2 involves computing the local density of nodes with the help of a fuzzy type approximation and substituting it for the local density of the patterns. In this technique, nodes are determined in grids with their integer coordinates. Finally, Step 3 includes finding mountain ridges, in which the outlook of the nodes' local density represents mountains with different heights. This process helps redefine the clustering task as the determination of mountain ridges.

The DGB technique determines clusters instead of setting the desired number of nodes and can detect nonwhite and white noise by defining the density threshold as noise. These aspects significantly decrease computational complexity because the algorithm does not calculate the Euclidean mutual distance between patterns. Given that this algorithm does not count the number of patterns in a cell or compute density contribution patterns, the authors applied the fuzzy approximation technique for computing the nodal density. Experimental results obtained using real datasets have confirmed that the DGB technique decreases processing time because it eliminates the estimation of Euclidean distances between all mutual patterns. However, the DGB cannot accommodate high-dimensional data streams owing to increases in the number of vacant grids, thereby decreasing the processing time.

I. GDSTREAM

Wang and Li (2017) proposed a new ‘‘grid and density for a data stream’’ (GDStream) approach [50]. This approach processes marginal points by dividing the data space and applying the data object to address the impact coefficients of neighboring grid components and enhance the algorithm’s effectiveness and correctness. Under GDStream, some connected relationships are well defined, and elementary models are called ‘‘density threshold MinPts’’ and ‘‘domain radius.’’ GDStream draws each streamed data point onto the matching grid cell and applies the effective number of data points to the neighboring cells’ center of mass to obtain the grid cell density, thereby effectively managing the occurrences of peripheral points in the grid cells. This process enables rapid, accurate, and predictable identification of clusters but comes at the expense of processing speed when real-time data are used.

J. GRIDEN

In 2018, a new density-based and grid-based clustering algorithm called GRIDEN was proposed by Chao and Jinwei [114]. This algorithm was designed by considering a newly proposed concept pertaining to neighboring cells to develop the aforementioned properties. GRIDEN can be described based on five serial steps that support parallel computing: (1) calculate the neighbor, (2) construct the grid, (3) merge the core cells, (4) find the core cells, and (5) find the border cells. GRIDEN’s basic concept pertains to learning from the grid-based clustering mechanism’s effectiveness and density-based clustering mechanism’s precision, incorporating their merits, and dealing with their deficiencies.

The GRIDEN algorithm enables the transfer of the basic concept with regard to the neighbor from points to cells and uses neighbor cells to execute the entire clustering process. The grid can be defined simply as a logical account book possessing a multi dimensional index, which enables rapid queries on the information of each cell. Thus, mapping of all data points should be done just once in the book instead of partitioning the memory. Compared with the grid-based DBSCAN algorithm, GRIDEN uses a new clustering mechanism that allows segmenting each grid cell to a set of symmetric hypersquare cells. This process results in the formation of a minimum hypersquare spatial zone in a manner that a cell’s neighbor could exist to minimize query. Thereafter, the set is used to identify spatially separated dense cells, which are merged to improve the density reachable and density connected clustering functions. The GRIDEN results indicate that clustering quality is reliable. In terms of runtime, GRIDEN is executed as a grid-based algorithm and needs computational time to be exclusively linear to N. Additional computing cores could be used to continuously accelerate clustering speed. However, an increase in parameter K and dimensionality D results in a rapid increase in the grid cell magnitude. Moreover, querying nonempty cells could result in running out of memory and low efficiency.

K. FAST DENSITY-GRID

Brown *et al.* (2019) developed a clustering algorithm called the ‘‘fast density-grid’’ [103], which achieves good performance on large datasets and decreases the running time required for clustering. This algorithm is also scalable for parallel versions and executed in three steps. Step 1 involves dividing an area into spaces in which data exist in the grid structure. Step 2 involves examining these spaces and investigating all neighboring spaces to determine the densest neighbor in each space. In existing techniques, neighbors for every space are initially determined. Last, Step 3 searches each neighbor to determine if it displays higher densities than space; if so, then it becomes the densest space. Every space is assigned a densest neighbor or represented as its own densest neighbor after every stage. After the densest neighbor in every grid space is assigned, the algorithm starts the cluster generation step. The algorithm’s limitation is that it leaves sparse grids and merges them as an outlier cluster for further consideration.

L. DGSTREAM

The DGStream algorithm, which consists of an offline and online processing framework [115], is a novel density grid-based clustering technique. In this algorithm, the online phase uses feature vectors represented with a micro-cluster for every grid, thereby helping dynamically maintain all necessary information on the constantly arriving data stream. In the offline phase, DGStream implements the DBSCAN algorithm to benefit from its speed and improve its run time. Moreover, DGStream depends on grids to decrease the time complexity and further accelerate the speed. The DGStream algorithm uses a decay function to accurately describe the stream evolution procedure. Moreover, it uses a process for deleting sparse grids and continues the processing using a few dense grids.

Furthermore, where the grid densities are below the specified threshold, they can be detected due to the input data’s small size. These grids can be removed after. In the algorithm, only dense grids are considered during the storage and processing stages, thereby saving the system memory and time. The DGStream algorithm uses a mechanism to eliminate noise and handle outliers. The implementation of the DGStream technique was evaluated using various simulated databases and different parameter settings. For this purpose, the researchers considered various concept drifts, cluster numbers, evolving data, and outlier detection.

M. ESA-STREAM

A previous study proposed a completely online and lightweight stream clustering algorithm called the ‘‘efficient self-adaptive stream’’ (ESA-Stream) [116]. ESA-Stream is used to address the issue pertaining to real-time processing and manually fixed parameters. In a self-adaptive manner, the main parameters are learned/generated online. The grid density centroid-based method accelerates dimensionality reduction and makes the clustering process considerably accurate and efficient. To build this technique, the authors provided three main contributions. First, they utilized fully online techniques that did not require the traditional time-consuming offline stage, as characterized in the DCDGA [61], CEDAS [29], CEC [117], CODAS [30], and CEDGM [36] algorithms. Second, an efficient self-adaptive technique was proposed to dynamically learn parameter settings during clustering to match evolving or drifting data streams. Third, a grid feature vector and grid density centroid were introduced to accelerate dimensionality reduction and enable the clustering of data streams efficiently and in a fully online mode. ESA-Stream is characterized by good clustering quality and rapid processing time. Even though ESA-Stream has proven its effectiveness in clustering real-time and high-dimensional evolving data streams, certain gaps continue to exist for this algorithm. For example, several Chebyshev distance function calls are needed to compute the distance between two adjacent grids when using high-dimensional data. Moreover, due to the large number of vacant squares, excellent memory efficiency is required when working with high-dimensional data.

N. CEDGM

A previous study proposed a novel online method called clustering evolving data streams via a density grid-based method (CEDGM) [118]. CEDGM's primary objectives are to decrease the number of calls to a distance function, improve cluster quality, discover noise, and understand all data points' properties in the developing data stream. Moreover, CEDGM uses information on data points to formulate core micro-clusters (CMCs). This technique is also used online and includes two major phases. Phase 1 generates CMCs, while Phase 2 combines all CMCs into macro-clusters. The grid-based technique is utilized as an outlier buffer to address multi-density and noise data. This study on CEDGM is considered the first to propose a new clustering process for determining the evolving nature of clusters after including grid granularity for reducing data, simplifying calculations, and eliminating the effects of fine data that lack a vital role in clustering. This mechanism forms grids after dividing the data space into small portions. Thereafter, an illustrative neighbor search is conducted on all grids to group them into cluster grids. After comparing this technique with other clustering algorithms, CEDGM was noted to offer the best computational time because it does not depend on the size of data points but the number of cells. This technique is beneficial for multi-density data and resilient against noise and specific, arbitrarily shaped clusters.

In the CEDGM case, a new data point from a data stream was observed to fall into one of three regions. First, when the data point lies in empty spaces of the grid granularity, it generated a novel outlier. Second, when the data point lies in CMC's shell area, it is assigned to a cluster. Thus, the CMC center and cluster counts were recursively updated. Third, the data point was allotted to CMC, and the cluster count was updated if the data point fell into the kernel area. Thereafter, the modified or generated CMC was examined to determine if the cluster density was above the minimum threshold. Researchers can examine this CMC to determine whether it newly overlaps with some other CMCs. If new overlapping is noted, then CMCs are linked to one macro-cluster. The connected CMCs possess one final cluster and create an arbitrarily shaped cluster. Different times and sample speeds were used to analyze the efficiency of the CEDGM algorithm. These results indicated that CEDGM could substantially improve the clustering results compared with other clustering algorithms.

Table 3 summarizes various grid-based clustering algorithms. Table 4 presents a comparison of the advantages and disadvantages of all grid-based algorithms.

VI. DISCUSSION OF FINDING

A. APPLIED APPROACHES

Section VIII discusses the methods used to provide solutions to the challenges and fulfil CR1 (approach and methodology) in detail. Each of the papers reviewed and its treatment of the researchers' challenges are tabulated and summarized in Tables 3 and 4. Note that substantial focus has been given to handling evolving data and limited time (50%) and noisy data (93%) compared with any other challenges, as described in Table 5 and Fig. 5(a) and 5(b).

B. INVESTIGATED DATASETS

Addressing CR2 (i.e., number of datasets used and case studies considered) has been proven difficult. The use of various datasets, each of which supports a variety of parameters and a predefined composition problem, is beneficial and essential for evaluating the proposed approaches and analyzing their performances. Unfortunately, the number of datasets available in the research domain is extremely low and limited to five key datasets: KDD CUP'99 [27]–[29], [41], [49], [52], [73], [80], [83], [86], [106], [111], [113], [36] and Forest Covertype [102], [112], Iris [103], [119], Airlines [2], [29], Traffic [116], and Electricity [2]. In rare cases, some researchers have relied on synthetically generated datasets to solve this problem.

C. CHALLENGES AND LIMITATIONS OF DATA STREAM CLUSTERING

This section aims to address CR3 (i.e., challenges and limitations of data stream clustering) and discuss numerous vital algorithms using density grid-based methods for data stream clustering. The reviewed methods partition a data space into small grid segments, map data streams onto the grids, and cluster data streams thereafter by applying the density-based method. The main challenges of data stream clustering are time limitations, handling noisy data, memory space limitations, handling high-dimensional data, and handling evolving data. To solve

TABLE 3. Comparison of various grid-based clustering algorithms.

References	Names	Year	Input Parameters	Objectives	Results
[41]	IGDCL	2010	*d*-dimensional data, dense grid threshold, minimum density thresh value, decay factor, radius	Clustering high-dimensional data	Reduced the influence of the sizes and borders of grid cells and overcome the sparsity of high-dimensional data streams
[111]	PKS-Stream	2011	PKS-tree, density threshold	Clustering high-dimensional data	Arbitrarily shaped clusters
[112]	DCUStream	2012	Data stream, density threshold	Clustering uncertain data	Arbitrarily shaped clusters
[47]	DENGRIS-Stream	2012	Data stream, sliding window size	Clustering over a sliding window	Arbitrarily shaped clusters
[43]	EXCC	2013	Grid granularity	Clustering heterogeneous data streams	Arbitrarily shaped clusters
[10]	HDC-Stream	2014	MinPts, data stream, radius	Improvement of computational time and quality	Arbitrarily shaped clusters
[27]	MuDi-Stream	2016	Sample speed, decay, MinPts, grid granularity, radius	Improvement of computational time and quality	Arbitrarily shaped clusters
[42]	DGB	2016	Eps (radius), MinPts, grid granularity	Decreasing the number of computations of Euclidean distances between mutual patterns and using the standard grid and sparse matrix technique to reduce processing time	Arbitrarily shaped clusters
[50]	GDStream	2017	MinPts, data stream, radius, decay	Identifying clusters rapidly and accurately	Arbitrarily shaped clusters
[114]	GRIDEN	2018	Data stream, MinPts, ε, k, Grid1 and Grid2	Achieving effectiveness for massive spatial data, credible quality, good robustness, ability to discover multi-density clusters, good flexibility, and supports parallel computing	Reliable clustering quality and fast runtime
[103]	Fast Density-Grid	2019	MinPts, grid granularity, radius	Improving the quality of clusters and reducing runtime	Arbitrarily shaped clusters
[115]	DGStream	2020	Undefined	Handling outliers and noisy data	Arbitrarily shaped clusters
[36]	CEDGM	2020	Radius, MinThreshold, decay, grid granularity, sample speed	Reducing computational time and improving cluster quality	Low computational time and arbitrarily shaped clusters
[116]	ESA-Stream	2020	Data stream X, decay factor λ, grid's length len	Solving the problem of real-time processing and manually fixed parameters	Fast processing time and good clustering quality

TABLE 4. Advantages and disadvantages of various grid-based clustering.

References	Names	Year	Advantages	Disadvantages
[41]	IGDCL	2010	- Improves cluster quality for uncertain data streams - Handles high-dimensional data	- No straightforward solution for removing sporadic grids - Cannot handle limits to memory space and computational time
[111]	PKS-Stream	2011	- Handles high-dimensional data - Handles noisy and evolving data	- Does not perform pruning on the tree after adding a new data point to any of the cells of the tree - Cannot handle limits to memory space and computational time
[112]	DCUStream	2012	- Improves the efficiency of clusters in unpredictable data streams	- Time-consuming when searching for a core dense grid and its neighbors
[47]	DENGRIS-Stream	2012	- Handles evolving data streams	- No analysis proving its effectiveness compared with alternative progressive algorithms
[43]	EXCC	2013	- Handles noisy and evolving data - Covers data streams with categorical and numerical attributes	- Cannot handle high-dimensional data - Use of a pool for keeping dense grids requires more time than other methods for processing and more memory to keep the grids
[10]	HDC-Stream	2014	- Low time complexity with high cluster quality	- Unable to cluster multi-density data
[27]	MuDi-Stream	2016	- Useful in multi-density environments	- Cannot handle high-dimensional data streams and long computational times
[42]	DGB	2016	- Outperforms other methods in terms of processing time required for 2D datasets	- Unable to handle high-dimensional data
[50]	GDStream	2017	- Detecting clusters rapidly and accurately	- Slow capturing of massive real-time data - Cannot handle limits to memory and computational time
[114]	GRIDEN	2018	- Reliable cluster quality and fast runtime	- Magnitude of grid cells increasing rapidly, possibly leading to out of memory and low efficiency when querying nonempty cells
[103]	Fast Density-Grid	2019	- Detecting clusters rapidly and accurately	- Retains and merges sparse grids for use as an outlier cluster
[115]	DGStream	2020	- Removing noise data and handling outliers	- Low cluster quality when dealing with high-dimensional datasets
[36]	CEDGM	2020	- Low computational complexity and time with high efficiency for the clustering of high-dimensional data	- Fails to merge when dealing with evolving datasets
[116]	ESA-Stream	2020	- Fast processing time and reliable clustering quality	- Requires a number of Chebyshev distance function calls - Requires high memory efficiency

these problems, several algorithms have been proposed, but they are unable to solve these problems. The following solutions in some of the algorithms address the identified challenges.

1) TIME LIMITATIONS

The data stream is constantly inward bound, and data require a real-time response [83], [100], [120]. Hence, a clustering algorithm is required to treat the data speed. HDC-Stream

TABLE 5. Density-based clustering algorithms and desired objectives in the investigated studies.

References	Approaches	Limited time	Handling noisy data	High-dimensional data	Limited memory space	Processing evolving data
[41]	IGDCL	√	√	√		
[111]	PKS-Stream		√	√		
[112]	DCUStream		√		√	√
[47]	DENGRIS-Stream		√		√	√
[43]	EXCC		√			√
[10]	HDC-Stream	√	√	√		√
[27]	MuDi-Stream	√	√		√	√
[42]	DGB	√	√			
[50]	GDStream				√	
[114]	GRIDEN	√	√			
[103]	Fast Density-Grid		√	√		
[115]	DGStream	√	√		√	√
[36]	CEDGM		√	√	√	√
[116]	ESA-Stream	√	√	√		

is one such algorithm and is characterized by low time complexity and high data stream clustering quality. As an alternative to the use of a searching list to find suitable outliers, this algorithm records new data objects into grid cells and saves a considerable amount of computational time [20], [59], [121], [94].

2) HANDLING NOISY DATA

Managing noisy data is another challenge. Clustering algorithms should be capable of processing random noise within data streams to minimize outliers' impact on cluster formation. The majority of the reviewed clustering methods can discover and delete interrupted grids plotted by outliers. Algorithms periodically verify grid density to ensure that thresholds are met; otherwise, the current grid is deleted from the list [10], [59], [121], [94].

3) HANDLING HIGH-DIMENSIONAL DATA

The main problem in the clustering data stream is the handling of high dimensionality to improve the scalability of the algorithms investigated; only the PKS-Stream algorithm is capable of dealing with high-dimensional data. Although several empty grids are present during high-dimensional data grid-based clustering, PKS-Stream improves storage efficiency by merely storing nonempty grid cells. The other reviewed approaches presume that the majority of the grids contain minimal data or are empty and do not propose any particular method for handling high-dimensional data [20], [59], [121], [94].

4) MEMORY SPACE LIMITATIONS

Given that vast amounts of data streams are produced in real time, substantial amounts of memory are required [83], [112]. Hence, clustering algorithms must operate efficiently and use limited memory. The reviewed methods divide the data area into small segments called grids and plot data records into the equivalent grid. They record the summary information for data records in each grid. For recording synopsis data, the reviewed grid-based clustering uses different summarization methods [59], [94], [121], [122].

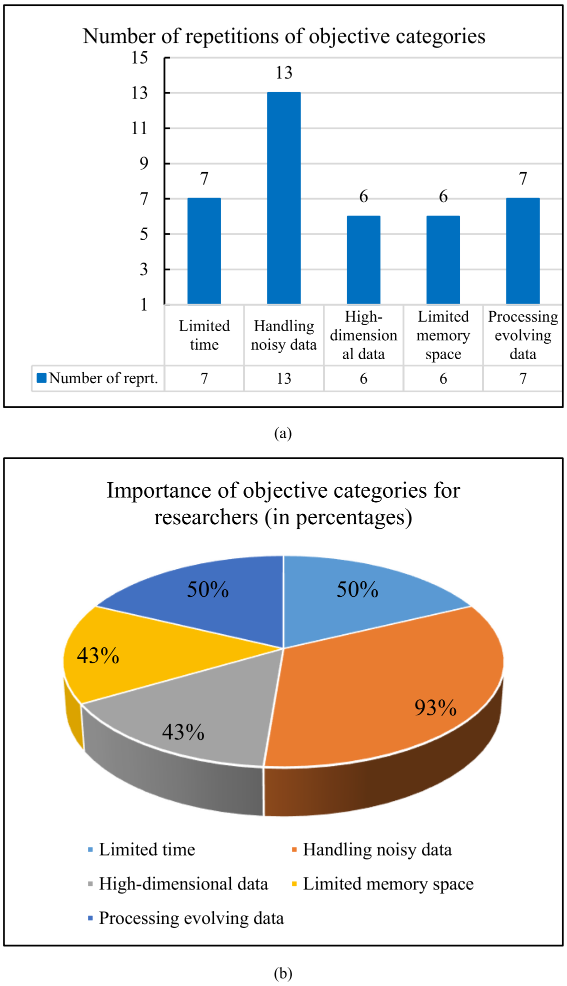


FIGURE 5. **(a) Number of repetitions of objective categories and (b) importance of objective categories (in percentages).**

5) HANDLING EVOLVING DATA STREAMS

A clustering algorithm must consider that data streams continuously evolve. The reviewed algorithms view a data stream's behavior as an evolving process over time using the "fading window" model. Each data record's density, which is determined by the "fading window" model, is assigned a decay factor. This decay factor places more weight on recent data than on older data while continuously keeping historical information [20], [59], [94], [122].

6) HANDLING COMPUTATIONAL TIME

The main issue in high-dimensional data stream clustering is the length of computational time. Only the CEDGM algorithm [119] is capable of dealing with this issue. The other proposed algorithms have high computational times resulting from numerous distance function calls between data points and the center of clusters.

VII. CONCLUSION

This study reviewed some remarkable density grid-based clustering algorithms developed for use with data streams. The reviewed clustering algorithms partition a data space into a limited number of cells, which form a grid. An infinite number of data records are plotted on the grid to create micro-clusters, which are used for final clustering based on their densities. The advantages and disadvantages of the reviewed grid-based clustering methods are presented in Table 1. The fundamental challenges in clustering data streams are memory limitations, time limitations, handling high-dimensional data, noisy data, and evolving data. No single reviewed algorithm can simultaneously address all challenging issues. Numerous challenges remain for grid-based clustering algorithms, particularly in terms of time-limited and high-dimensional data handling. Evidently, the preceding discussion on various types of density grid-based clustering algorithms can claim that the field of data stream clustering is open for researchers. Some possible research focus areas suggested for the future are listed as follows.

- Grid-based clustering and micro-clustering have their respective advantages. Hence, the development of algorithms using a hybrid method combining grid-based clustering and micro-clustering warrants further investigation.
- We observed that the majority of the algorithms in their offline phases use DBSCAN, which requires the setting of various parameters. Investigations into the use of other types of approaches focused on density for data stream clustering should be conducted.
- The majority of clustering algorithms are unable to handle high-dimensional data streams. The number of grids increases as the dimensionality of space increases. They achieve poor performances on extremely high-dimensional data. Accordingly, further research should be conducted to improve the performance of such algorithms.
- These algorithms have been developed to handle data streams containing clusters of different sizes, shapes, and densities. Nevertheless, only a few of them can handle challenging clustering tasks.
- The evaluation of density-based clustering algorithms that use actual datasets, such as Synapse and clinical trials, is another possible research topic.
- No existing algorithm can simultaneously address all issues. Hence, an effective algorithm with the ability to solve all the identified problems should be developed.
- Most of the approaches are nonautonomous, which means they need manual tuning of their internal parameters. Unfortunately, the tuning process requires considerable effort, and the parameter results might become invalid after a particular time due to statistical changes in data or concept drift characteristics.
- The dynamical metrics used by the existing clustering algorithms ignore handling the problem of false merging of clusters thatoccurs when two or more clusters overlap on top of each other. This leads to false prediction in the clustering analysis caused by the evolving natures of the clusters and therefore reduces thecluster quality.

REFERENCES

[1] M. Mousavi, A. A. Bakar, and M. Vakilian, "Data stream clustering algorithms: A review," *Int. J. Adv. Soft Comput. Appl.*, vol. 7, no. 3, p. 13, 2015.

[2] J. P. Barddal, H. M. Gomes, and F. Enembreck, "SNCStream: A social network-based data stream clustering algorithm," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 935–940.

[3] M. Mousavi and A. A. Bakar, "Improved density based algorithm for data stream clustering," *J. Teknologi*, vol. 77, no. 18, pp. 73–77, Nov. 2015.

[4] A. Algergawy, M. Mesiti, R. Nayak, and G. Saake, "XML data clustering: An overview," *ACM Comput. Surv.*, vol. 43, no. 4, pp. 1–41, Oct. 2011.

[5] Y. Yogita and D. Toshniwal, "Clustering techniques for streaming data—A survey," in *Proc. 3rd IEEE Int. Adv. Comput. Conf. (IACC)*, Feb. 2013, pp. 951–956.

[6] 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, 2015.

[7] V. Moustaka, A. Vakali, and L. G. Anthopoulos, "A systematic review for smart city data analytics," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–41, Jan. 2019.

[8] F. H. Kuwil, Ü. Atila, R. Abu-Issa, and F. Murtagh, "A novel data clustering algorithm based on gravity center methodology," *Expert Syst. Appl.*, vol. 156, Oct. 2020, Art. no. 113435.

[9] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, 2015.

[10] A. Amini, H. Saboohi, T. Y. Wah, and T. Herawan, "A fast density-based clustering algorithm for real-time Internet of Things stream," *Sci. World J.*, vol. 2014, pp. 1–11, Jan. 2014.

[11] M. Khalilian and N. Mustapha, "Data stream clustering: Challenges and issues," 2010, *arXiv:1006.5261*.

[12] B. Krawczyk, L. L. Minkub, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017.

[13] K. Wankhade, T. Hasan, and R. Thool, "A survey: Approaches for handling evolving data streams," in *Proc. Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2013, pp. 621–625.

[14] 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.*, 2002, pp. 685–694.

[15] D. S. Suresh and W. Vinod, "Survey paper on clustering data streams based on shared density between micro-clusters," *Int. Res. J. Eng. Technol.*, pp. 440–445, 2017.

[16] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, 2015.

[17] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: Recent achievements and new challenges," *Inf. Fusion*, vol. 28, pp. 45–59, Mar. 2016.

[18] M. Hahsler and M. Bolaños, "Clustering data streams based on shared density between micro-clusters," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1449–1461, Jun. 2016.

[19] S. Askari, "Fuzzy C-means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113856.

[20] A. Amini, "An adaptive density-based method for clustering evolving data streams," Fakulti Sains Komputer dan Teknologi Maklumat, Univ. Malaya, Kuala Lumpur, Malaysia, 2014.

[21] S. Ding, F. Wu, J. Qian, H. Jia, and F. Jin, "Research on data stream clustering algorithms," *Artif. Intell. Rev.*, vol. 43, no. 4, pp. 593–600, Apr. 2015.

[22] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. D. Carvalho, and J. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–31, 2013.

[23] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi, S. Benrernou, I. Brandic, A. Kertész, M. Parkin, and M. Carro, "A survey on service quality description," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–58, Oct. 2013.

[24] P. Goyal, J. S. Challa, S. Shrivastava, and N. Goyal, "Anytime frequent itemset mining of transactional data streams," *Big Data Res.*, vol. 21, Sep. 2020, Art. no. 100146.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[26] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Sep. 2018.

[27] 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.

[28] J.-Y. Chen and H.-H. He, "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data," *Inf. Sci.*, vol. 345, pp. 271–293, Jun. 2016.

[29] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," *Inf. Sci.*, vols. 382–383, pp. 96–114, Mar. 2017.

[30] R. Hyde and P. Angelov, "A new online clustering approach for data in arbitrary shaped clusters," in *Proc. IEEE 2nd Int. Conf. Cybern. (CYBCONF)*, Jun. 2015, pp. 228–233.

[31] A. Madraky, Z. A. Othman, and A. Hamdan, "Analytic methods for spatio-temporal data in a nature-inspired data model," *Int. Rev. Comput. Softw.*, vol. 9, no. 3, pp. 547–556, 2014.

[32] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel, and A. Küpper, "Variations on the clustering algorithm BIRCH," *Big Data Res.*, vol. 11, pp. 44–53, Mar. 2018.

[33] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler, "StreamKM++: A clustering algorithm for data streams," *J. Exp. Algorithmics*, vol. 17, pp. 2.1–2.30, May 2012.

[34] A. A. Abin and V.-V. Vu, "A density-based approach for querying informative constraints for clustering," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113690.

[35] M. Carnein and H. Trautmann, "EvoStream—Evolutionary stream clustering utilizing idle times," *Big Data Res.*, vol. 14, pp. 101–111, Dec. 2018.

[36] M. Tareq, E. A. Sundararajan, M. Mohd, and N. S. Sani, "Online clustering of evolving data streams using a density grid-based method," *IEEE Access*, vol. 8, pp. 166472–166490, 2020.

[37] 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.

[38] E. Güngör and A. Özmen, "Distance and density based clustering algorithm using Gaussian kernel," *Expert Syst. Appl.*, vol. 69, pp. 10–20, Mar. 2017.

[39] F. Ros and S. Guillaume, "DENDIS: A new density-based sampling for clustering algorithm," *Expert Syst. Appl.*, vol. 56, pp. 349–359, Sep. 2016.

[40] X. Chen, "A new clustering algorithm based on near neighbor influence," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7746–7758, Nov. 2015.

[41] G. Hou, R. Yao, J. Ren, and C. Hu, "Irregular grid-based clustering over high-dimensional data streams," in *Proc. 1st Int. Conf. Pervasive Comput., Signal Process. Appl.*, Sep. 2010, pp. 783–786.

[42] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1620–1628, Aug. 2017.

[43] V. Bhatnagar, S. Kaur, and S. Chakravarthy, "Clustering data streams using grid-based synopsis," *Knowl. Inf. Syst.*, vol. 41, no. 1, pp. 127–152, Oct. 2014.

[44] N. H. Park and W. S. Lee, "Statistical grid-based clustering over data streams," *ACM SIGMOD Rec.*, vol. 33, no. 1, pp. 32–37, Mar. 2004.

[45] A. Zhou, F. Cao, Y. Yan, C. Sha, and X. He, "Distributed data stream clustering: A fast EM-based approach," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 736–745.

[46] Z. Zhishui, "A kind of data stream clustering algorithm based on grid-density," in *Proc. Int. Conf. Comput. Sci., Environ., Ecoinformatics, Educ.* Berlin, Germany: Springer, 2011, pp. 418–423.

[47] 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.*, 2012, pp. 206–210.

[48] F. Gouineau, T. Landry, and T. Triplet, "PatchWork, a scalable density-grid clustering algorithm," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, Apr. 2016, pp. 824–831.

[49] W. Liu and J. OuYang, "Clustering algorithm for high dimensional data stream over sliding windows," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 1537–1542.

[50] L. Wang and H. Li, "Clustering algorithm based on grid and density for data stream," *AIP Conf. Proc.*, vol. 1839, no. 1, 2017, Art. no. 020202.

[51] R. Greenlaw and S. Kantabutra, "Survey of clustering: Algorithms and applications," *Int. J. Inf. Retr. Res.*, vol. 3, no. 2, pp. 1–29, 2013.

[52] 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 Data*, vol. 3, no. 3, pp. 1–28, Jul. 2009.

[53] J. J. Jung, "Semantic preprocessing for mining sensor streams from heterogeneous environments," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6107–6111, May 2011.

[54] D. Brown and Y. Shi, "A distributed density-grid clustering algorithm for multi-dimensional data," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 1–8.

[55] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE Tech. Rep. ver. 2.3, 2007, vol. 5.

[56] B. Kitchenham, "Procedures for performing systematic reviews," Keele Univ., Keele, U.K., 2004, pp. 1–26, vol. 33, no. 2004.

[57] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quart.*, vol. 26, no. 2, pp. 13–23, 2002.

[58] Q. Quan, C. K. Xiao, and R. Zhang, "Grid-based data stream clustering for intrusion detection," *Int. J. Netw. Secur.*, vol. 15, no. 1, pp. 1–8, Jan. 2013.

[59] K. S. S. Reddy and C. S. Bindu, "A review on density-based clustering algorithms for big data analysis," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud) (I-SMAC)*, Feb. 2017, pp. 123–130.

[60] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evolving Syst.*, vol. 9, no. 1, pp. 1–23, 2018.

[61] M. Tareq and E. A. Sundararajan, "A new density-based method for clustering data stream using genetic algorithm," *Technol. Rep. Kansai Univ.*, vol. 62, no. 11, p. 16, 2020.

[62] C. C. Aggarwal and S. Y. Philip, "A survey of synopsis construction in data streams," in *Data Streams*. Boston, MA, USA: Springer, 2007, pp. 169–207.

[63] J. Gama and M. M. Gaber, *Learning From Data Streams: Processing Techniques in Sensor Networks*. Berlin, Germany: Springer, 2007.

[64] J. Gama, *Knowledge Discovery From Data Streams*. Boca Raton, FL, USA: CRC Press, 2010.

[65] C. C. Aggarwal, *Data Streams: Models and Algorithms*. NY, USA: Springer, 2007.

[66] M. Ghesmoune, M. Lebbah, and H. Azzag, "State-of-the-art on clustering data streams," *Big Data Anal.*, vol. 1, no. 1, p. 13, 2016.

[67] B. Aubaidan, M. Mohd, and M. Albared, "Comparative study of k-means and k-means++ clustering algorithms on crime domain," *J. Comput. Sci.*, vol. 10, no. 7, pp. 1197–1206, 2014.

[68] V. Sankardoss, S. Sabberwal, and K. Rajambal, "Experimental design of capacitance required for self-excited induction generator," *J. Theor. Appl. Inf. Technol.*, vol. 45, no. 1, pp. 1–7, 2012.

[69] J. de Andrade Silva, E. R. Hruschka, and J. Gama, "An evolutionary algorithm for clustering data streams with a variable number of clusters," *Expert Syst. Appl.*, vol. 67, pp. 228–238, Jan. 2017.

[70] M. D. da Assunção, A. da Silva Veith, and R. Buyya, "Distributed data stream processing and edge computing: A survey on resource elasticity and future directions," *J. Netw. Comput. Appl.*, vol. 103, pp. 1–17, Feb. 2018.

[71] G. Krempl, I. Žliobaite, D. Brzezinski, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, and M. Spiliopoulou, "Open challenges for data stream mining research," *ACM SIGKDD Explor. Newslett.*, vol. 16, no. 1, pp. 1–10, 2014.

[72] M. H. U. Rehman, C. S. Liew, T. Y. Wah, and M. K. Khan, "Towards next-generation heterogeneous mobile data stream mining applications: Opportunities, challenges, and future research directions," *J. Netw. Comput. Appl.*, vol. 79, pp. 1–24, Feb. 2017.

[73] 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.

[74] D. Barbará, "Requirements for clustering data streams," *ACM SIGKDD Explor. Newslett.*, vol. 3, no. 2, pp. 23–27, Jan. 2002.

[75] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams: Theory and practice," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 3, pp. 515–528, May/Jun. 2003.

[76] M. Aljibawi, M. Z. A. Nazri, and Z. Othman, "A survey on clustering density based data stream algorithms," *Int. J. Eng. Technol.*, vol. 7, no. 36, pp. 147–153, 2018.

[77] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "On high dimensional projected clustering of data streams," *Data Mining Knowl. Discovery*, vol. 10, no. 3, pp. 251–273, May 2005.

[78] M. Charikar, L. O'Callaghan, and R. Panigrahy, "Better streaming algorithms for clustering problems," in *Proc. 35th ACM Symp. Theory Comput.*, 2003, pp. 30–39.

[79] 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*, vol. 30, 2004, pp. 852–863.

[80] L. Tu and Y. Chen, "Stream data clustering based on grid density and attraction," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 3, pp. 1–27, Jul. 2009.

[81] 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, 2008.

[82] F. Ansarifar and A. Ahmadi, "A novel algorithm for adaptive data stream clustering," in *Proc. Iranian Conf. Electr. Eng. (ICEE)*, May 2018, pp. 1542–1546.

[83] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 133–142.

[84] Z. Xiong, R. Chen, Y. Zhang, and X. Zhang, "Multi-density DBSCAN algorithm based on density levels partitioning," *J. Inf. Comput. Sci.*, vol. 9, no. 10, pp. 2739–2749, 2012.

[85] R. D. Baruah and P. Angelov, "DEC: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1619–1631, Sep. 2014.

[86] R. D. Baruah, P. Angelov, and D. Baruah, "Dynamically evolving clustering for data streams," in *Proc. IEEE Conf. Evolving Adapt. Intell. Syst. (EAIS)*, Jun. 2014, pp. 1–6.

[87] C. Isaksson, M. H. Dunham, and M. Hahsler, "SOStream: Self organizing density-based clustering over data stream," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, 2012, pp. 264–278.

[88] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.

[89] M. N. Garofalakis, *Wavelets on Streams*. Princeton, NJ, USA: Citeseer, 2009.

[90] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: Survey summary," in *Proc. Companion Web Conf. Web Conf.* Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 507–511.

[91] H.-H. Chen, B.-L. Shi, J.-B. Qian, and Y.-F. Chen, "Wavelet synopsis based clustering of parallel data streams," *J. Softw.*, vol. 21, no. 4, pp. 644–658, Mar. 2010.

[92] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, 2002, pp. 1–16.

[93] M. Garofalakis, "Wavelets on streams," in *Encyclopedia of Database Systems*. 2009, pp. 3446–3451.

[94] 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.

[95] M. Garofalakis, J. Gehrke, and R. Rastogi, "Querying and mining data streams: You only get one look a tutorial," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2002, p. 635.

[96] R. Sebastião, J. Gama, and T. Mendonça, "Fading histograms in detecting distribution and concept changes," *Int. J. Data Sci. Anal.*, vol. 3, no. 3, pp. 183–212, May 2017.

[97] G. Cormode and S. Muthukrishnan, "An improved data stream summary: The count-min sketch and its applications," *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[98] J. Mao, Q. Song, C. Jin, Z. Zhang, and A. Zhou, "Online clustering of streaming trajectories," *Frontiers Comput. Sci.*, vol. 12, no. 2, pp. 245–263, Apr. 2018.

[99] Y. Lu, Y. Sun, G. Xu, and G. Liu, "A grid-based clustering algorithm for high-dimensional data streams," in *Proc. Int. Conf. Adv. Data Mining Appl.* Berlin, Germany: Springer, 2005, pp. 824–831.

[100] L. Tu and Y. Chen, "Stream data clustering based on grid density and attraction," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 3, p. 12, Jul. 2009.

[101] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[102] S. Dong, J. Liu, Y. Liu, L. Zeng, C. Xu, and T. Zhou, "Clustering based on grid and local density with priority-based expansion for multi-density data," *Inf. Sci.*, vol. 468, pp. 103–116, Nov. 2018.

[103] D. Brown, A. Japa, and Y. Shi, "A fast density-grid based clustering method," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 48–54.

[104] C.-M. Chao and G.-L. Chao, "Resource-aware density-and-grid-based clustering in ubiquitous data streams," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 1203–1208.

[105] W. Hu, M. Cheng, G. Wu, and L. Wu, "Research on parallel data stream clustering algorithm based on grid and density," in *Proc. Int. Conf. Comput. Sci. Mech. Autom. (CSMA)*, Oct. 2015, pp. 70–75.

[106] Y.-H. Lu and Y. Huang, "Mining data streams using clustering," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 4, 2005, pp. 2079–2083.

[107] A. H. Pilevar and M. Sukumar, "GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 999–1010, May 2005.

[108] B.-Z. Qiu, X.-L. Li, and J.-Y. Shen, "Grid-based clustering algorithm based on intersecting partition and density estimation," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2007, pp. 368–377.

[109] M. Tareq and E. A. Sundararajan, "An evolving approach to data streams clustering based on Chebychev with false merging," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 9, pp. 1–11, 2021.

[110] G. Esfandani, M. Sayyadi, and A. Namadchian, "GDCLU: A new grid-density based CLUstring algorithm," in *Proc. 13th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Aug. 2012, pp. 102–107.

[111] 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.

[112] Y. Yang, Z. Liu, J.-P. Zhang, and J. Yang, "Dynamic density-based clustering algorithm over uncertain data streams," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, May 2012, pp. 2664–2670.

[113] J. Gao, J. Li, Z. Zhang, and P.-N. Tan, "An incremental data stream clustering algorithm based on dense units detection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2005, pp. 420–425.

[114] C. Deng, J. Song, R. Sun, S. Cai, and Y. Shi, "GRIDEN: An effective grid-based and density-based spatial clustering algorithm to support parallel computing," *Pattern Recognit. Lett.*, vol. 109, pp. 81–88, Jul. 2018.

[115] R. Ahmed, G. Dalkılıç, and Y. Erten, "DGStream: High quality and efficiency stream clustering algorithm," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112947.

[116] Y. Li, H. Li, Z. Wang, B. Liu, J. Cui, and H. Fei, "ESA-Stream: Efficient self-adaptive online data stream clustering," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 23, 2020, doi: 10.1109/TKDE.2020.2990196.

[117] M. Tareq, E. A. Sundararajan, and M. Mohd, "Online clustering of evolving data stream based on adaptive Chebychev distance," in *Proc. 281st Int. Conf. IIER*, 2020, pp. 41–46.

[118] C. Jia, C. Tan, and A. Yong, "A grid and density-based clustering algorithm for processing data stream," in *Proc. 2nd Int. Conf. Genet. Evol. Comput.*, Sep. 2008, pp. 517–521.

[119] D. Zhang, H. Tian, Y. Sang, Y. Li, Y. Wu, J. Wu, and H. Shen, "A clustering algorithm based on density-grid for stream data," in *Proc. 13th Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, Dec. 2012, pp. 398–403.

[120] A. Amini and T. Y. Wah, "Density micro-clustering algorithms on data streams: A review," in *Proc. World Congr. Eng.*, vol. 2188. London, U.K.: International Association of Engineers, 2010, pp. 410–414.

[121] A. Amini, T. Y. Wah, M. R. Saybani, and S. R. A. S. Yazdi, "A study of density-grid based clustering algorithms on data streams," in *Proc. 8th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Jul. 2011, pp. 1652–1656.

[122] A. S. Sabau, "Clustering data streams using mass estimation," in *Proc. 15th Int. Symp. Symbolic Numeric Algorithms Sci. Comput.*, Sep. 2013, pp. 289–295.

MUSTAFA TAREQ received the B.Sc. degree in computer science from AL-Turath University College, Iraq, in 2012, the M.Sc. degree in network technology from the National University of Malaysia (UKM), Malaysia, in 2016, and the Ph.D. degree in computer science from the Centre of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). He is currently a Lecturer with the Department of Computer Technology Engineering, Al-Hikma University College, Iraq. His research interests include the Internet of Things, data mining, clustering algorithms, evolving data streams, and computer networks.

ELANKOVAN A SUNDARARAJAN (Member, IEEE) received the B.Sc. degree (Hons.) and the M.Sc. degree in computer science from the National University of Malaysia, in 1995 and 1997, respectively, and the Ph.D. degree in computer science from The University of Melbourne, Australia, in 2008. He is currently an Associate Professor with the National University of Malaysia (UKM). His current research interests include high-performance computing, big data, cloud computing, and computer networks.

AARON HARWOOD (Member, IEEE) received the BIT, B.Eng. (ME), and Ph.D. degrees from Griffith University, Brisbane, Australia, in 1998 and 2002, respectively. He is currently a Senior Lecturer with the Department of Computer Science and Software Engineering, The University of Melbourne, Melbourne. He is also a Founding Member of the Peer-to-Peer Networks and Applications Research Group. His research interests include the performance of large-scale, decentralized, autonomous systems. He has been a member of the ACM and the IEEE Computer Society, since 2003.

AZURALIZA ABU BAKAR (Member, IEEE) received the Ph.D. degree in artificial intelligence from Universiti Putra Malaysia (UPM), in 2002. She has been a Professor in data mining with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), since 2011. Her research interests include data analytics and artificial intelligence specifically in rough set theory, feature selection algorithms, nature-inspired computing, and sentiment analysis. She served as an Advisor for the Data Mining and Optimization Laboratory and a member of the Sentiment Analysis Laboratory. She has led 13 research projects, including three in progress and is a member of 42 research projects. She is a member of the IEEE Computational Intelligence Society. She has also served on approximately 30 conference and workshop program committees and has served as the Program Chair.